S922X Datasheet

Revision: 01

Release Date: 2019-05-05

Amlogic, Ltd.

DISCLAIMER

Amlogic Ltd. may make improvements and/or changes in this document or in the product described in this document at any time.

This product is not intended for use in medical, life saving, or life sustaining applications.

Circuit diagrams and other information relating to products of Amlogic Ltd. are included as a means of illustrating typical applications. Consequently, complete information sufficient for production design is not necessarily given. Amlogic makes no representations or warranties with respect to the accuracy or completeness of the contents presented in this document.

CONTACT INFORMATION

Amlogic, Ltd.

2518 Mission College Blvd, Suite 120

Santa Clara, CA 95054

U.S.A.

www.amlogic.com

Change History

Issue 01 (2019-05-05)

This is the 01 version.

Compared to *S922X Datasheet (0.1)*, the following topics are changed.

Section	Change Description
3.3	Added a note that some GPIOAO pins can be reset by watchdog.
4	Modified the chapter title.
4.3	Added a note of ripple voltage after the table
4.4	Modified thermo resistance value.
4.5.1	Added two notes of min and max value of R_{PD} explaining the GPIO pin voltage in test condition.
4.7.1	Added note after Table 4-2 that driving strength of OD pin is not adjustabe.
4.7.7	• Modified the specs of tHC, tLC and tRC in Table 4-14 and Table 4-15. • Updated timing figures from Figure 4-16 to Figure 4-19.
5 and 6	Changed these two sections into chapters.
6	Updated the power on sequence figure.
7	Updated the dimension figure.
14.2	Removed spec of PCIE.
14.3.3.2	Modified register description of ETH_PHY_CNTL1.
14.6.3	Corrected a typo.

Contents

1. About This Documentation

1.1 Documentation Overview

This documentation is an overall description of Amlogic advanced AI application processor S922X designed for hybrid OTT/IP Set Top Box(STB) and high-end media box applications, providing programmers instructions from the following aspects: system, video path, audio path, memory interfaces, I/O interfaces and system interfaces.

1.2 Acronyms and Abbreviations

Table 1-1 Acronyms and Abbreviations

A		
AHB	AMBA High-speed Bus	A bus protocal designed for the connection and management of functional blocks in system-on-a-chip (SoC) designs
APB	Advanced Peripheral Bus	A bus protocal designed for low bandwidth control accesses
D		
DMAC	Direct Memory Access Controller	An engine connected to the DDR controller for the purposes of moving data to/from DDR memory.
DMC	DDR memory controller	An engine controls DDR
E		
eMMC	Embedded multimedia card	An architecture consisting of an embedded stortage solution with MMC interface, flash memory and controler
H		
HDMI	High-Definition Multimedian Interface	A compact audio/video interface used for transmitting uncompressed digital data
I		
I2C	Inter-Integrated Circuit	An multi-master, multi-slave, single-ended, serial computer bus used for attaching lower-speed peripheral ICs to processors and microcontrollers.
I2S	Inter IC sound	An electrical serial bus interface stand used for connnecting digital audio device together
P		
PCM	Pulse Code Modulation	A method used to digitally represent sampled analog signals
PDM	Pulse Density Modulation	A mothed used to represent an analog signal with digital signal
S		
SPI	Synchronous Peripheral Interface	A synchronous serial data link standard operating in sull duplex mode, devices communicate in master/slave mode where the master device initiates the date frame

2. General Information

2.1 Overview

S922X is an advanced application processor designed for hybrid OTT/IP Set Top Box(STB) and high-end media box applications. It integrates a powerful CPU, GPU subsystem, a secured 4K video CODEC engine and a best-in-class HDR image processing pipeline with all major peripherals to form the ultimate high-performance multimedia AP.

The main system CPU is based on Big.Little architecture which integrates a quad-core ARM Cortex-A73 CPU cluster and a dual-core Cortex-A53 cluster with united L2 cache to improve system performance. Each CPU core includes the separate NEON SIMD co-processor to improve software media processing capability.

The graphic subsystem consists of two graphic engines and a flexible video/graphic output pipeline. The ARM Mali™-G52-MP4(6EE) GPU handles all OpenGL ES 3.2 Vulkan 1.0 and OpenCL 2.0 graphic programs, while the 2.5D graphics processor handles additional scaling, alpha, rotation and color space conversion operations. Together, the CPU and GPU handle all operating system, networking, user-interface and gaming related tasks. The video output pipeline includes Dolby Vision[optional], advanced HDR10, HDR10+, HLG and PRIME HDR processing, REC709/BT2020 processing, motion adaptive edge enhancing de-interlacing, flexible programmable scalar, and many picture enhancement filters before passing the enhanced image to the video output ports.

Amlogic Video Engine (AVE-10) offloads the Cortex-A53 CPUs from all video CODEC processing. It includes dedicated hardware video decoder and encoder. AVE-10 is capable of decoding 4Kx2K resolution video at 75fps with complete Trusted Video Path (TVP) for secure applications and supports full formats including MVC, MPEG-1/2/4, VC-1/WMV, AVS, AVS+, AVS2 RealVideo, MJPEG streams, H.264, H265-10, VP9 and also JPEG pictures with no size limitation. The independent encoder is able to encode in JPEG or H.265/H.264 up to 1080p at 60fps.

S922X integrates all standard audio/video input/output interfaces including a HDMI2.1 transmitter with 3D, Dynamic HDR, CEC and HDCP 2.2 support, stereo audio DAC, a CVBS output, 4-lane MIPI DSI interface, multiple TDM, PCM, I2S and SPDIF digital audio input/output interfaces, 8 channel far-field PDM digital microphone (DMIC) inputs and a DVP camera interface.

S922X also integrates a set of functional blocks for digital TV broadcasting streams. The built-in two demux can process the TV streams from the serial and parallel transport stream input interface, which can connect to external tuner/demodulator.

The processor has rich advanced network and peripheral interfaces, including a 10/100/1000M Ethernet MAC with RGMII, 10/100M Ethernet PHY, one USB XHCI OTG 2.0 port, one USB3.0 and PCIe 2.0 combo interface and multiple SDIO/SD card controllers, UART, I2C, high-speed SPI and PWMs.

Standard development environment utilizing GNU/GCC Android tool chain is supported. Please contact your AMLOGIC sales representative for more information.

2.2 Features

CPU Sub-system

- Quad core ARM Cortex-A73 and dual core ARM Cortex-A53 CPU
- ARMv8-A architecture with Neon and Crypto extensions
- Unified system L2 cache
- Build-in Cortex-M4 core for always on processing
- Advanced TrustZone security system
- Application based traffic optimization using internal QoS-based switching fabrics

3D Graphics Processing Unit

- ARM G52 MP4(6EE) GPU support max to 800MHz
- 8-wide warps, 2x dual texture pipe, 6x 8-wide execution engines (EE)
- Concurrent multi-core processing
- OpenGL ES 3.2, Vulkan 1.0 and OpenCL 2.0 support

2.5D Graphics Processor

- Fast bitblt engine with dual inputs and single output
- Programmable raster operations (ROP)
- Programmable polyphase scaling filter
- Supports multiple video formats 4:2:0, 4:2:2 and 4:4:4 and multiple pixel formats (8/16/24/32 bits graphics layer)
- Fast color space conversion
- Advanced anti-flickering filter

Crypto Engine

- AES/ block cipher with 128/256 bits keys, standard 16 bytes block size and streaming ECB, CBC and CTR modes
- TDES block cipher with ECB and CBC modes supporting 64 bits key for DES and 192 bits key for 3DES
- Hardware crypto key-ladder operation and DVB-CSA for transport stream encryption
- Built-in hardware True Random Number Generator (TRNG), CRC and SHA-1/SHA-2/HMAC SHA engine

Video/Picture CODEC

- Amlogic Video Engine (AVE) with dedicated hardware decoders and encoders
- Support multi-video decoder up to 4Kx2K@60fps+1x1080P@60fps
- Supports multiple “secured” video decoding sessions and simultaneous decoding and encoding
- Video/Picture Decoding
    - VP9 Profile-2 up to 4Kx2K@60fps
    - H.265 HEVC MP-10@L5.1 up to 4Kx2K@60fps
    - AVS2-P2 Profile up to 4Kx2K@60fps
    - H.264 AVC HP@L5.1 up to 4Kx2K@30fps
    - H.264 MVC up to 1080P@60fps
    - MPEG-4 ASP@L5 up to 1080P@60fps (ISO-14496)
    - WMV/VC-1 SP/MP/AP up to 1080P@60fps
    - AVS-P16(AVS+) /AVS-P2 JiZhun Profile up to 1080P@60fps
    - MPEG-2 MP/HL up to 1080P@60fps (ISO-13818)
    - MPEG-1 MP/HL up to 1080P@60fps (ISO-11172)

  - RealVideo 8/9/10 up to 1080P@60fps
  - Multiple language and multiple format sub-title video support
  - MJPEG and JPEG unlimited pixel resolution decoding (ISO/IEC-10918)
  - Supports JPEG thumbnail, scaling, rotation and transition effects
  - Supports *.mkv,*.wmv,*.mpg, *.mpeg, *.dat, *.avi, *.mov, *.iso, *.mp4, *.rm and *.jpg file formats

- Video/Picture Encoding
  - Independent JPEG and H.265/H.264 encoder with configurable performance/bit-rate
  - JPEG image encoding
  - H.265/H.264 video encoding up to 1080P@60fps with low latency
  - DVP (ITU 601/656) camera interface
  - Supports YUV or RGB camera input formats

9th Generation Advanced Amlogic TruLife Image Engine

- Supports Dolby Vision[Optional], HDR10, HDR10+, HLG HDR and Prime HDR processing
- Motion compensated noise reduction and 3D digital noise reduction for random noise
- Block noise, mosquito noise, spatial noise, contour noise reduction
- Motion compensated and motion adaptive de-interlacer
- Edge interpolation with low angle protection and processing
- 3:2/2:2 pulldown and Video on Film (VOF) detection and processing
- Smart sharpness with SuperScaler technology including de-contouring, de-ring, LTI, CTI, de-jaggy, peaking
- Dynamic non-Linear contrast enhancement
- 3D LUTs with 17x17x17 nodes, provide 4913 different control points,which is competent for matching calibrated displays to a target colorspace
- High precision HSL color space-based color management with low saturation protection, independent luma/hue/saturation adjustment to achieve blue/green extension, fresh tone correction, and wider gamut for video
- 2 video planes and 3 graphics planes
- Independent HDR re-mapping of video and graphic layer

Camera Interface

- DVP(ITU 601/656) camera interface with 4 lanes
- Supports RAW, YUV or RGB camera input formats

Video Output

- Built-in HDMI 2.1 transmitter including both controller and PHY with CEC, Dynamic HDR and HDCP 2.2, 4Kx2K@60 max resolution output
- CVBS 480i/576i standard definition output
- Supports all standard SD/HD/FHD video output formats: 480i/p, 576i/p, 720p, 1080i/p and 4Kx2K
- 4-lane MIPI DSI interface, resolution up to 1920*1080 with rotation and panel calibration

Audio Decoder and Input/Output

- Supports MP3, AAC, WMA, RM, FLAC, Ogg and programmable with 7.1/5.1 down-mixing
- Built-in serial digital audio SPDIF/IEC958 input/output and PCM input/output, up to 192KHz x 32bit x ch2
- 3 built-in TDM/PCM/I2S ports with TDM/PCM mode up to 384kHz x32bits x 8ch or 96kHz x 32bits x 32ch and I2S mode up to 384kHz x 32bits x 8ch

- Digital microphone PDM voice input with programmable CIC, LPF & HPF, support up to 8 DMICs
- Built-in stereo audio DAC
- Supports concurrent dual audio stereo channel output with combination of analog+PCM or I2S+PCM

Memory and Storage Interface

- 32-bit DRAM memory interface with dual ranks and max 4GB total address space
- Compatible with JEDEC standard DDR3-2133 /DDR3L-2133 /DDR4-2666 /LPDDR3-2133 /LPDDR4-3200 SDRAM
- Supports SLC/MLC/TLC NAND Flash with 60-bit ECC, compatible to Toshiba toggle mode in addition to ONFI 2.2
- SDSC/SDHC/SDXC card and SDIO interface with 1-bit and 4-bit data bus width supporting spec version 2.x/3.x/4.x DS/HS modes up to UHS-I SDR104
- eMMC and MMC card interface with 1/4/8-bit data bus width fully supporting spec version 5.0 HS200
- Supports serial 1, 2 or 4-bit NOR Flash via SPI interface
- Built-in 4k bits One-Time-Programming memory for key storage

Network

- Integrated IEEE 802.3 10/100/1000M Ethernet MAC with RGMII interface
- Integrated 10[Note1]/100M Ethernet PHY interface
- WiFi/IEEE802.11 & Bluetooth supporting via PCIE/SDIO /USB/UART/PCM
- Network interface optimized for mixed WIFI and BT traffic

Digital Television Interface

- One serial and one parallel Transport stream (TS) input interface with built-in demux processor for connecting to external digital TV tuner/demodulator
- Built-in PWM, I2C and SPI interfaces to control tuner and demodulator
- Integrated ISO 7816 smart card controller

Integrated I/O Controllers and Interfaces

- One USB XHCI OTG 2.0 port
- One USB SS and PCIe 2.0 combo interface up to 5Gbps, supporting 2 configurations:
    - 1 USB 2.0 Host + 1 PCIe
    - 1 USB3.0 (No PCIe)
- Multiple PWM, UART, I2C and SPI interface with slave select
- Programmable IR remote input/output controllers
- Built-in 10bit SAR ADC with 4 input channels
- A set of General Purpose IOs with built-in pull up and pull down

System, Peripherals and Misc. Interfaces

- Integrated general purpose timers, counters, DMA controllers
- 24 MHz crystal input
- Embedded debug interface using ICE/JTAG

Power Management

- Multiple internal power domains controlled by software
- Multiple sleep modes for CPU, system, DRAM, etc.
- Multiple internal PLLs to adjust the operating frequencies
- Multi-voltage I/O design for 1.8V and 3.3V

- Power management auxiliary processor in a dedicated always-on (AO) power domain for system stand-by

Security

- Trustzone based Trusted Execution Environment (TEE)
- Secured boot, encrypted OTP, encrypted DRAM with memory integrity checker, hardware key ladder and internal control buses and storage
- Separated secure/non-secure Entropy true RNG
- Pre-region/ID memory security control and electric fence
- Hardware based Trusted Video Path (TVP), video watermarking and secured contents (needs SecureOS software)
- Secured IO and secured clock

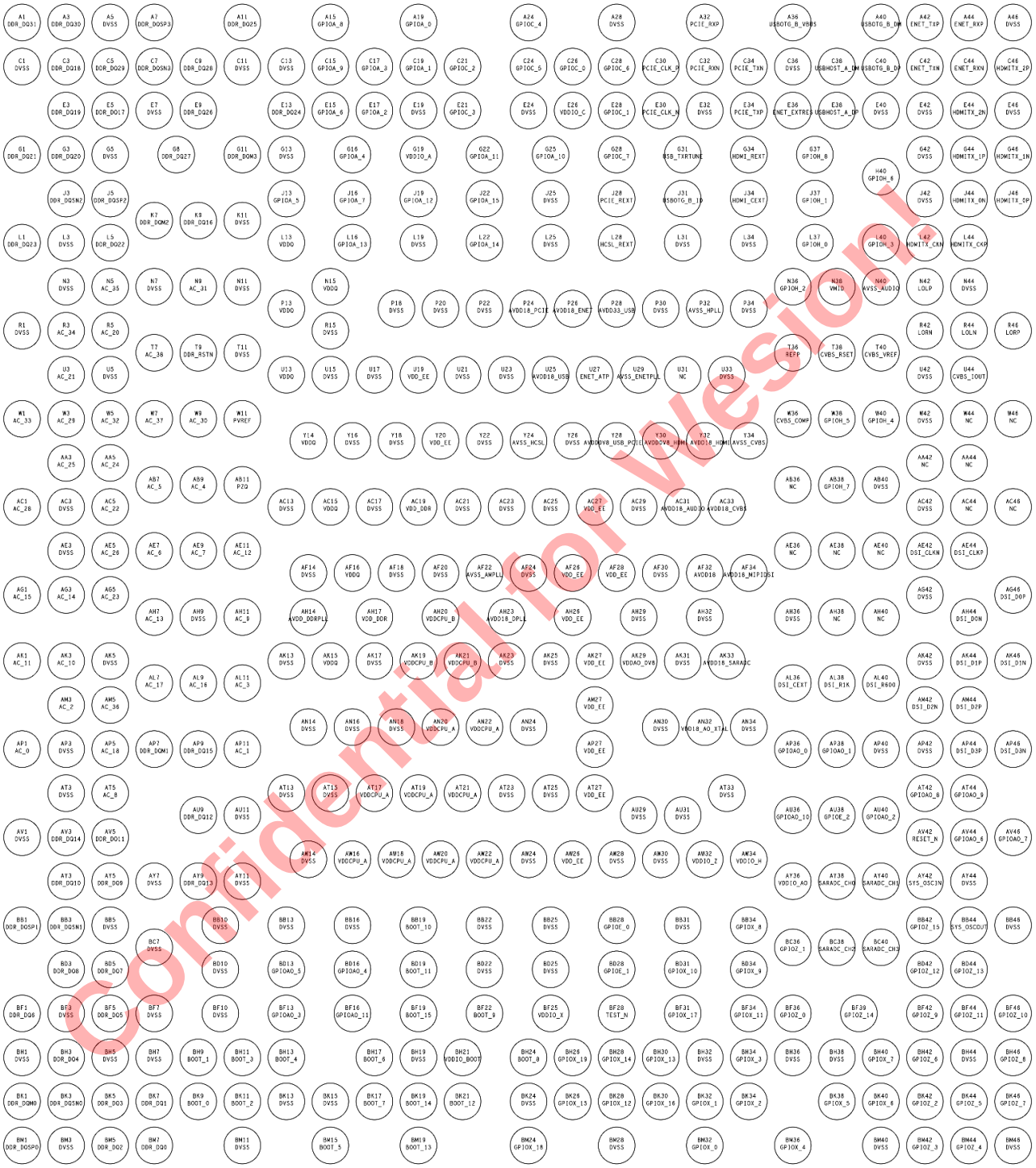
Package

- FCBGA, 16.1mmx14.3mm, 0.6mm ball pitch, RoHS compliant

Note1: 10Base-T not support CTS mode

3. Pin Out Specification

3.1 Pin-Out Diagram (top view)



3.2 Pin Order

Table 3-1. Pin Order

BALL #	NET NAME	BALL #	NET NAME	BALL #	NET NAME
A1	DDR_DQ31	E15	GPIOA_6	J34	HDMI_CEXT
A3	DDR_DQ30	E17	GPIOA_2	J37	GPIOH_1
A5	DVSS	E19	DVSS	J42	DVSS
A7	DDR_DQSP3	E21	GPIOC_3	J44	HDMITX_0N
A11	DDR_DQ25	E24	DVSS	J46	HDMITX_0P
A15	GPIOA_8	E26	VDDIO_C	K7	DDR_DQM2
A19	GPIOA_0	E28	GPIOC_1	K9	DDR_DQ16
A24	GPIOC_4	E30	PCIe_CLK_n	K11	DVSS
A28	DVSS	E32	DVSS	L1	DDR_DQ23
A32	PCIe_RXP	E34	PCIe_TXP	L3	DVSS
A36	USBOTG_B_VBUS	E36	ENET_EXTRES	L5	DDR_DQ22
A40	USBOTG_B_DM	E38	USBHOST_A_DP	L13	VDDQ
A42	ENET_TXP	E40	DVSS	L16	GPIOA_13
A44	ENET_RXP	E42	DVSS	L19	DVSS
A46	DVSS	E44	HDMITX_2N	L22	GPIOA_14
C1	DVSS	E46	DVSS	L25	DVSS
C3	DDR_DQ18	G1	DDR_DQ21	L28	HCSL_REXT
C5	DDR_DQ29	G3	DDR_DQ20	L31	DVSS
C7	DDR_DQSN3	G5	DVSS	L34	DVSS
C9	DDR_DQ28	G8	DDR_DQ27	L37	GPIOH_0
C11	DVSS	G11	DDR_DQM3	L40	GPIOH_3
C13	DVSS	G13	DVSS	L42	HDMITX_CKN
C15	GPIOA_9	G16	GPIOA_4	L44	HDMITX_CKP
C17	GPIOA_3	G19	VDDIO_A	N3	DVSS
C19	GPIOA_1	G22	GPIOA_11	N5	AC_35

BALL #	NET NAME	BALL #	NET NAME	BALL #	NET NAME
C21	GPIOC_2	G25	GPIOA_10	N7	DVSS
C24	GPIOC_5	G28	GPIOC_7	N9	AC_31
C26	GPIOC_0	G31	USB_TXRTUNE	N11	DVSS
C28	GPIOC_6	G34	HDMI_REXT	N15	VDDQ
C30	PCIe_CLK_p	G37	GPIOH_8	N36	GPIOH_2
C32	PCIe_RXN	G42	DVSS	N38	VMID
C34	PCIe_TXN	G44	HDMITX_1P	N40	AVSS_AUDIO
C36	DVSS	G46	HDMITX_1N	N42	LOLP
C38	USBHOST_A_DM	H40	GPIOH_6	N44	DVSS
C40	USBOTG_B_DP	J3	DDR_DQSN2	P13	VDDQ
C42	ENET_TXN	J5	DDR_DQSP2	P18	DVSS
C44	ENET_RXN	J13	GPIOA_5	P20	DVSS
C46	HDMITX_2P	J16	GPIOA_7	P22	DVSS
E3	DDR_DQ19	J19	GPIOA_12	P24	AVDD18_PCIe
E5	DDR_DQ17	J22	GPIOA_15	P26	AVDD18_ENET
E7	DVSS	J25	DVSS	P28	AVDD33_USB
E9	DDR_DQ26	J28	PCIe_REXT	P30	DVSS
E13	DDR_DQ24	J31	USBOTG_B_ID	P32	AVSS_HPLL
P34	DVSS	Y24	AVSS_HCSL	AF20	DVSS
R1	DVSS	Y26	DVSS	AF22	AVSS_AMPLL
R3	AC_34	Y28	AVDD0V8_USB_PCIe	AF24	DVSS
R5	AC_20	Y30	AVDD0V8_HDMI	AF26	VDD_EE
R15	DVSS	Y32	AVDD18_HDMI	AF28	VDD_EE
R42	LORN	Y34	AVSS_CVBS	AF30	DVSS
R44	LOLN	AA3	AC_25	AF32	AVDD18
R46	LORP	AA5	AC_24	AF34	AVDD18_MIPIDSI
T7	AC_38	AA42	NC	AG1	AC_15

BALL #	NET NAME	BALL #	NET NAME	BALL #	NET NAME
T9	DDR_RSTn	AA44	NC	AG3	AC_14
T11	DVSS	AB7	AC_5	AG5	AC_23
T36	REFP	AB9	AC_4	AG42	DVSS
T38	CVBS_RSET	AB11	PZQ	AG46	DSI_D0P
T40	CVBS_VREF	AB36	NC	AH7	AC_13
U3	AC_21	AB38	GPIOH_7	AH9	DVSS
U5	DVSS	AB40	DVSS	AH11	AC_9
U13	VDDQ	AC1	AC_28	AH14	AVDD_DDRPLL
U15	DVSS	AC3	DVSS	AH17	VDD_DDR
U17	DVSS	AC5	AC_22	AH20	VDDCPU_B
U19	VDD_EE	AC13	DVSS	AH23	AVDD18_DPLL
U21	DVSS	AC15	VDDQ	AH26	VDD_EE
U23	DVSS	AC17	DVSS	AH29	DVSS
U25	AVDD18_USB	AC19	VDD_DDR	AH32	DVSS
U27	ENET_ATP	AC21	DVSS	AH36	DVSS
U29	AVSS_ENETPLL	AC23	DVSS	AH38	NC
U31	NC	AC25	DVSS	AH40	NC
U33	DVSS	AC27	VDD_EE	AH44	DSI_D0N
U42	DVSS	AC29	DVSS	AK1	AC_11
U44	CVBS_IOUT	AC31	AVDD18_AUDIO	AK3	AC_10
W1	AC_33	AC33	AVDD18_CVBS	AK5	DVSS
W3	AC_29	AC42	DVSS	AK13	DVSS
W5	AC_32	AC44	NC	AK15	VDDQ
W7	AC_37	AC46	NC	AK17	DVSS
W9	AC_30	AE3	DVSS	AK19	VDDCPU_B
W11	PVREF	AE5	AC_26	AK21	VDDCPU_B
W36	CVBS_COMP	AE7	AC_6	AK23	DVSS

BALL #	NET NAME	BALL #	NET NAME	BALL #	NET NAME
W38	GPIOH_5	AE9	AC_7	AK25	DVSS
W40	GPIOH_4	AE11	AC_12	AK27	VDD_EE
W42	DVSS	AE36	NC	AK29	VDDAO_0V8
W44	NC	AE38	NC	AK31	DVSS
W46	NC	AE40	NC	AK33	AVDD18_SARADC
Y14	VDDQ	AE42	DSI_CLKN	AK42	DVSS
Y16	DVSS	AE44	DSI_CLKP	AK44	DSI_D1P
Y18	DVSS	AF14	DVSS	AK46	DSI_D1N
Y20	VDD_EE	AF16	VDDQ	AL7	AC_17
Y22	DVSS	AF18	DVSS	AL9	AC_16
AL11	AC_3	AU29	DVSS	BB46	DVSS
AL36	DSI_CEXT	AU31	DVSS	BC7	DVSS
AL38	DSI_R1K	AU36	GPIOAO_10	BC36	GPIOZ_1
AL40	DSI_R600	AU38	GPIOE_2	BC38	SARADC_CH2
AM3	AC_2	AU40	GPIOAO_2	BC40	SARADC_CH3
AM5	AC_36	AV1	DVSS	BD3	DDR_DQ8
AM27	VDD_EE	AV3	DDR_DQ14	BD5	DDR_DQ7
AM42	DSI_D2N	AV5	DDR_DQ11	BD10	DVSS
AM44	DSI_D2P	AV42	RESET_N	BD13	GPIOAO_5
AN14	DVSS	AV44	GPIOAO_6	BD16	GPIOAO_4
AN16	DVSS	AV46	GPIOAO_7	BD19	BOOT_11
AN18	DVSS	AW14	DVSS	BD22	DVSS
AN20	VDDCPU_A	AW16	VDDCPU_A	BD25	DVSS
AN22	VDDCPU_A	AW18	VDDCPU_A	BD28	GPIOE_1
AN24	DVSS	AW20	VDDCPU_A	BD31	GPIOX_10
AN30	DVSS	AW22	VDDCPU_A	BD34	GPIOX_9
AN32	VDD18_AO_XTAL	AW24	DVSS	BD42	GPIOZ_12

BALL #	NET NAME	BALL #	NET NAME	BALL #	NET NAME
AN34	DVSS	AW26	VDD_EE	BD44	GPIOZ_13
AP1	AC_0	AW28	DVSS	BF1	DDR_DQ6
AP3	DVSS	AW30	DVSS	BF3	DVSS
AP5	AC_18	AW32	VDDIO_Z	BF5	DDR_DQ5
AP7	DDR_DQM1	AW34	VDDIO_H	BF7	DVSS
AP9	DDR_DQ15	AY3	DDR_DQ10	BF10	DVSS
AP11	AC_1	AY5	DDR_DQ9	BF13	GPIOAO_3
AP27	VDD_EE	AY7	DVSS	BF16	GPIOAO_11
AP36	GPIOAO_0	AY9	DDR_DQ13	BF19	BOOT_15
AP38	GPIOAO_1	AY11	DVSS	BF22	BOOT_9
AP40	DVSS	AY36	VDDIO_AO	BF25	VDDIO_X
AP42	DVSS	AY38	SARADC_CH0	BF28	TEST_N
AP44	DSI_D3P	AY40	SARADC_CH1	BF31	GPIOX_17
AP46	DSI_D3N	AY42	SYS_OSCIN	BF34	GPIOX_11
AT3	DVSS	AY44	DVSS	BF36	GPIOZ_0
AT5	AC_8	BB1	DDR_DQSP1	BF39	GPIOZ_14
AT13	DVSS	BB3	DDR_DQSN1	BF42	GPIOZ_9
AT15	DVSS	BB5	DVSS	BF44	GPIOZ_11
AT17	VDDCPU_A	BB10	DVSS	BF46	GPIOZ_10
AT19	VDDCPU_A	BB13	DVSS	BH1	DVSS
AT21	VDDCPU_A	BB16	DVSS	BH3	DDR_DQ4
AT23	DVSS	BB19	BOOT_10	BH5	DVSS
AT25	DVSS	BB22	DVSS	BH7	DVSS
AT27	VDD_EE	BB25	DVSS	BH9	BOOT_1
AT33	DVSS	BB28	GPIOE_0	BH11	BOOT_3
AT42	GPIOAO_8	BB31	DVSS	BH13	BOOT_4
AT44	GPIOAO_9	BB34	GPIOX_8	BH17	BOOT_6

BALL #	NET NAME	BALL #	NET NAME	BALL #	NET NAME
AU9	DDR_DQ12	BB42	GPIOZ_15	BH19	DVSS
AU11	DVSS	BB44	SYS_OSCOUT	BH21	VDDIO_BOOT
BH24	BOOT_8	BK11	BOOT_2	BM1	DDR_DQSP0
BH26	GPIOX_19	BK13	DVSS	BM3	DVSS
BH28	GPIOX_14	BK15	DVSS	BM5	DDR_DQ2
BH30	GPIOX_13	BK17	BOOT_7	BM7	DDR_DQ0
BH32	DVSS	BK19	BOOT_14	BM11	DVSS
BH34	GPIOX_3	BK21	BOOT_12	BM15	BOOT_5
BH36	DVSS	BK24	DVSS	BM19	BOOT_13
BH38	DVSS	BK26	GPIOX_15	BM24	GPIOX_18
BH40	GPIOX_7	BK28	GPIOX_12	BM28	DVSS
BH42	GPIOZ_6	BK30	GPIOX_16	BM32	GPIOX_0
BH44	DVSS	BK32	GPIOX_1	BM36	GPIOX_4
BH46	GPIOZ_8	BK34	GPIOX_2	BM40	DVSS
BK1	DDR_DQM0	BK38	GPIOX_5	BM42	GPIOZ_3
BK3	DDR_DQSN0	BK40	GPIOX_6	BM44	GPIOZ_4
BK5	DDR_DQ3	BK42	GPIOZ_2	BM46	DVSS
BK7	DDR_DQ1	BK44	GPIOZ_5	-	-
BK9	BOOT_0	BK46	GPIOZ_7	-	-

3.3 Pin Description

The S922X application processor pin assignment is described in the following table.

Table 3-2. Pin Name assignments

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
GPIOZ - Refer to Table 3-3 for functional multiplex information.					
GPIOZ_0	DIO	Up	General purpose input/output bank Z signal 0	VDDIO_Z	NC
GPIOZ_1	DIO	Up	General purpose input/output bank Z signal 1	VDDIO_Z	NC
GPIOZ_2	DIO	Up	General purpose input/output bank Z signal 2	VDDIO_Z	NC
GPIOZ_3	DIO	Up	General purpose input/output bank Z signal 3	VDDIO_Z	NC
GPIOZ_4	DIO	Up	General purpose input/output bank Z signal 4	VDDIO_Z	NC
GPIOZ_5	DIO	Up	General purpose input/output bank Z signal 5	VDDIO_Z	NC
GPIOZ_6	DIO	Up	General purpose input/output bank Z signal 6	VDDIO_Z	NC
GPIOZ_7	DIO	Up	General purpose input/output bank Z signal 7	VDDIO_Z	NC
GPIOZ_8	DIO	Up	General purpose input/output bank Z signal 8	VDDIO_Z	NC
GPIOZ_9	DIO	Down	General purpose input/output bank Z signal 9	VDDIO_Z	NC
GPIOZ_10	DIO	Down	General purpose input/output bank Z signal 10	VDDIO_Z	NC
GPIOZ_11	DIO	Down	General purpose input/output bank Z signal 11	VDDIO_Z	NC
GPIOZ_12	DIO	Down	General purpose input/output bank Z signal 12	VDDIO_Z	NC
GPIOZ_13	DIO	Down	General purpose input/output bank Z signal 13	VDDIO_Z	NC
GPIOZ_14	OD 3.3V	Z	OD input/output bank Z signal 14	VDDIO_Z	NC
GPIOZ_15	OD 3.3V	Z	OD input/output bank Z signal 15	VDDIO_Z	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
VDDIO_Z	P	-	Power supply for GPIO bank Z	-	NC
GPIOA - Refer to Table 3-4 for functional multiplex information.					
GPIOA_0	DIO	Down	General purpose input/output bank A signal 0	VDDIO_A	NC
GPIOA_1	DIO	Down	General purpose input/output bank A signal 1	VDDIO_A	NC
GPIOA_2	DIO	Down	General purpose input/output bank A signal 2	VDDIO_A	NC
GPIOA_3	DIO	Down	General purpose input/output bank A signal 3	VDDIO_A	NC
GPIOA_4	DIO	Down	General purpose input/output bank A signal 4	VDDIO_A	NC
GPIOA_5	DIO	Down	General purpose input/output bank A signal 5	VDDIO_A	NC
GPIOA_6	DIO	Down	General purpose input/output bank A signal 6	VDDIO_A	NC
GPIOA_7	DIO	Down	General purpose input/output bank A signal 7	VDDIO_A	NC
GPIOA_8	DIO	Down	General purpose input/output bank A signal 8	VDDIO_A	NC
GPIOA_9	DIO	Down	General purpose input/output bank A signal 9	VDDIO_A	NC
GPIOA_10	DIO	Down	General purpose input/output bank A signal 10	VDDIO_A	NC
GPIOA_11	DIO	Down	General purpose input/output bank A signal 11	VDDIO_A	NC
GPIOA_12	DIO	Down	General purpose input/output bank A signal 12	VDDIO_A	NC
GPIOA_13	DIO	Down	General purpose input/output bank A signal 13	VDDIO_A	NC
GPIOA_14	DIO	Up	General purpose input/output bank A signal 14	VDDIO_A	NC
GPIOA_15	DIO	Up	General purpose input/output bank A signal 15	VDDIO_A	NC
VDDIO_A	P	-	Power supply for GPIO bank A	-	NC
BOOT - Refer to Table 3-5 for functional multiplex information.					

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
BOOT_0	DIO	UP	General purpose input/output bank BOOT signal 0	VDDIO_BOOT	NC
BOOT_1	DIO	UP	General purpose input/output bank BOOT signal 1	VDDIO_BOOT	NC
BOOT_2	DIO	UP	General purpose input/output bank BOOT signal 2	VDDIO_BOOT	NC
BOOT_3	DIO	UP	General purpose input/output bank BOOT signal 3	VDDIO_BOOT	NC
BOOT_4	DIO	UP	General purpose input/output bank BOOT signal 4	VDDIO_BOOT	NC
BOOT_5	DIO	UP	General purpose input/output bank BOOT signal 5	VDDIO_BOOT	NC
BOOT_6	DIO	UP	General purpose input/output bank BOOT signal 6	VDDIO_BOOT	NC
BOOT_7	DIO	UP	General purpose input/output bank BOOT signal 7	VDDIO_BOOT	NC
BOOT_8	DIO	UP	General purpose input/output bank BOOT signal 8	VDDIO_BOOT	NC
BOOT_9	DIO	UP	General purpose input/output bank BOOT signal 9	VDDIO_BOOT	NC
BOOT_10	DIO	UP	General purpose input/output bank BOOT signal 10	VDDIO_BOOT	NC
BOOT_11	DIO	UP	General purpose input/output bank BOOT signal 11	VDDIO_BOOT	NC
BOOT_12	DIO	DOWN	General purpose input/output bank BOOT signal 12	VDDIO_BOOT	NC
BOOT_13	DIO	DOWN	General purpose input/output bank BOOT signal 13	VDDIO_BOOT	NC
BOOT_14	DIO	UP	General purpose input/output bank BOOT signal 14	VDDIO_BOOT	NC
BOOT_15	DIO	UP	General purpose input/output bank BOOT signal 15	VDDIO_BOOT	NC
VDDIO_BOOT	P	-	Power supply for GPIO bank BOOT	-	To VDDIO_BOOT
GPIOC - Refer to Table 3-6 for functional multiplex information.					
GPIOC_0	DIO	UP	General purpose input/output bank C signal 0	VDDIO_C	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
GPIOC_1	DIO	UP	General purpose input/output bank C signal 1	VDDIO_C	NC
GPIOC_2	DIO	UP	General purpose input/output bank C signal 2	VDDIO_C	NC
GPIOC_3	DIO	UP	General purpose input/output bank C signal 3	VDDIO_C	NC
GPIOC_4	DIO	UP	General purpose input/output bank C signal 4	VDDIO_C	NC
GPIOC_5	DIO	UP	General purpose input/output bank C signal 5	VDDIO_C	NC
GPIOC_6	DIO	UP	General purpose input/output bank C signal 6	VDDIO_C	NC
GPIOC_7	OD3.3V	Z	OD input/output bank C signal 7	VDDIO_C	NC
VDDIO_C	P	-	Power supply for GPIO bank C	-	NC
GPIOX - Refer to Table 3-7 for functional multiplex information.					
GPIOX_0	DIO	Up	General purpose input/output bank X signal 0	VDDIO_X	NC
GPIOX_1	DIO	Up	General purpose input/output bank X signal 1	VDDIO_X	NC
GPIOX_2	DIO	Up	General purpose input/output bank X signal 2	VDDIO_X	NC
GPIOX_3	DIO	Up	General purpose input/output bank X signal 3	VDDIO_X	NC
GPIOX_4	DIO	Up	General purpose input/output bank X signal 4	VDDIO_X	NC
GPIOX_5	DIO	Up	General purpose input/output bank X signal 5	VDDIO_X	NC
GPIOX_6	DIO	Down	General purpose input/output bank X signal 6	VDDIO_X	NC
GPIOX_7	DIO	Up	General purpose input/output bank X signal 7	VDDIO_X	NC
GPIOX_8	DIO	Up	General purpose input/output bank X signal 8	VDDIO_X	NC
GPIOX_9	DIO	Up	General purpose input/output bank X signal 9	VDDIO_X	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
GPIOX_10	DIO	Up	General purpose input/output bank X signal 10	VDDIO_X	NC
GPIOX_11	DIO	Up	General purpose input/output bank X signal 11	VDDIO_X	NC
GPIOX_12	DIO	Up	General purpose input/output bank X signal 12	VDDIO_X	NC
GPIOX_13	DIO	Up	General purpose input/output bank X signal 13	VDDIO_X	NC
GPIOX_14	DIO	Up	General purpose input/output bank X signal 14	VDDIO_X	NC
GPIOX_15	DIO	Up	General purpose input/output bank X signal 15	VDDIO_X	NC
GPIOX_16	DIO	Up	General purpose input/output bank X signal 16	VDDIO_X	NC
GPIOX_17	DIO	Down	General purpose input/output bank X signal 17	VDDIO_X	NC
GPIOX_18	DIO	Up	General purpose input/output bank X signal 18	VDDIO_X	NC
GPIOX_19	DIO	Z	General purpose input/output bank X signal 19	VDDIO_X	NC
VDDIO_X	P	-	Power supply for GPIO bank X	-	NC
GPIOH - Refer to Table 3-8 for functional multiplex information.					
GPIOH_0	OD5V	Z	OD input/output bank H signal 0	VDDIO_H	NC
GPIOH_1	OD5V	Z	OD input/output bank H signal 1	VDDIO_H	NC
GPIOH_2	OD5V	Z	OD input/output bank H signal 2	VDDIO_H	NC
GPIOH_3	OD5V	Z	OD input/output bank H signal 3	VDDIO_H	NC
GPIOH_4	DIO	DOWN	General purpose input/output bank H signal 4	VDDIO_H	NC
GPIOH_5	DIO	DOWN	General purpose input/output bank H signal 5	VDDIO_H	NC
GPIOH_6	DIO	DOWN	General purpose input/output bank H signal 6	VDDIO_H	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
GPIOH_7	DIO	DOWN	General purpose input/output bank H signal 7	VDDIO_H	NC
GPIOH_8	OD5V	Z	OD input/output bank H signal 8	VDDIO_H	NC
VDDIO_H	P	-	Power supply for GPIO bank H	-	NC
GPIOAO - Refer to Table 3-9 for functional multiplex information. The following pins can be reset by watchdog: GPIOAO_0, GPIOAO_1, GPIOAO_2, GPIOAO_4, GPIOAO_6, GPIOAO_7, GPIOAO_10, TEST_N (only output reg and pull up/down).					
GPIOAO_0	DIO	Up	General purpose input/output bank AO signal 0	VDDIO_AO	NC
GPIOAO_1	DIO	Up	General purpose input/output bank AO signal 1	VDDIO_AO	NC
GPIOAO_2	DIO	Down	General purpose input/output bank AO signal 2	VDDIO_AO	NC
GPIOAO_3	DIO	Up	General purpose input/output bank AO signal 3	VDDIO_AO	NC
GPIOAO_4	DIO	Down	General purpose input/output bank AO signal 4	VDDIO_AO	NC
GPIOAO_5	DIO	Up	General purpose input/output bank AO signal 5	VDDIO_AO	NC
GPIOAO_6	DIO	Down	General purpose input/output bank AO signal 6	VDDIO_AO	NC
GPIOAO_7	DIO	Up	General purpose input/output bank AO signal 7	VDDIO_AO	NC
GPIOAO_8	DIO	Up	General purpose input/output bank AO signal 8	VDDIO_AO	NC
GPIOAO_9	DIO	Down	General purpose input/output bank AO signal 9	VDDIO_AO	NC
GPIOAO_10	DIO	Up	General purpose input/output bank AO signal 10	VDDIO_AO	NC
GPIOAO_11	DIO	Down	General purpose input/output bank AO signal 11	VDDIO_AO	NC
TEST_N	DIO	UP	SOC test pin and general purpose input/output bank AO signal 12. Should be pulled up during normal power-on.	VDDIO_AO	NC
RESET_N	Input	DOWN	System reset input	VDDIO_AO	To RESET_N

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
VDDIO_AO	P	-	Power supply for GPIO bank AO	VDDIO_AO	To VDDIO_AO
GPIOE- Refer to Table 3-10 for functional multiplex information.					
GPIOE_0	DIO	Z	General purpose input/output bank E signal 0	VDD18_AO_XTAL	NC
GPIOE_1	DIO	Z	General purpose input/output bank E signal 1	VDD18_AO_XTAL	NC
GPIOE_2	DIO	Z	General purpose input/output bank E signal 2	VDD18_AO_XTAL	NC
VDD18_AO_XTAL	P	-	Power supply for GPIO bank E and XTAL, IOVREF	-	To VDD18_AO_XTAL
SARADC					
SARADC_CH0	AI	-	ADC channel 0 input	AVDD18_SARADC	NC
SARADC_CH1	AI	-	ADC channel 1 input	AVDD18_SARADC	NC
SARADC_CH2	AI	-	ADC channel 2 input	AVDD18_SARADC	NC
SARADC_CH3	AI	-	ADC channel 3 input	AVDD18_SARADC	NC
AVDD18_SARADC	P	-	Analog power supply for SARADC	-	To 1.8V
CVBS OUT					
CVBS_COMP	A	-	CVBS external compensation capacitor connection	AVDD18_CVBS	NC
CVBS_IOUT	AO	-	Video DAC output	AVDD18_CVBS	NC
CVBS_RSET	A	-	CVBS output strength setting resistor	AVDD18_CVBS	NC
CVBS_VREF	A	-	CVBS reference voltage filter cap	AVDD18_CVBS	NC
AVDD18_CVBS	P	-	1.8 V Analog power supply for CVBS_OUT	-	To 1.8V
HDMI TX					
HDMITX_0P	AO	-	HDMI TMDS data0 positive output	NC	NC
HDMITX_0N	AO	-	HDMI TMDS data0 negative output	NC	NC
HDMITX_1P	AO	-	HDMI TMDS data1 positive output	NC	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
HDMITX_1N	AO	-	HDMI TMDS data1 negative output	NC	NC
HDMITX_2P	AO	-	HDMI TMDS data2 positive output	NC	NC
HDMITX_2N	AO	-	HDMI TMDS data2 negative output	NC	NC
HDMITX_CKP	AO	-	HDMI TMDS clock positive output	NC	NC
HDMITX_CKN	AO	-	HDMI TMDS clock negative output	NC	NC
HDMI_REXT	A	-	HDMI output strength setting resistor	HDMI_AVDD18	NC
HDMI_CEXT	A	-	HDMI TX external filter cap	HDMI_AVDD18	NC
AVDD18_HDMI	P	-	Analog power supply 1.8V for HDMI	-	To 1.8V
AVDD0V8_HDMI	P	-	Power supply 0.8V for HDMI	-	To VDD_EE
DRAM- Refer to Table 3-11 for functional multiplex information.					
AC_0	DO	-	DDR PHY address/command/control signal bit 0	VDDQ	NC
AC_1	DO	-	DDR PHY address/command/control signal bit 1	VDDQ	NC
AC_2	DO	-	DDR PHY address/command/control signal bit 2	VDDQ	NC
AC_3	DO	-	DDR PHY address/command/control signal bit 3	VDDQ	NC
AC_4	DO	-	DDR PHY address/command/control signal bit 4	VDDQ	NC
AC_5	DO	-	DDR PHY address/command/control signal bit 5	VDDQ	NC
AC_6	DO	-	DDR PHY address/command/control signal bit 6	VDDQ	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
AC_7	DO	-	DDR PHY address/command/control signal bit 7	VDDQ	NC
AC_8	DO	-	DDR PHY address/command/control signal bit 8	VDDQ	NC
AC_9	DO	-	DDR PHY address/command/control signal bit 9	VDDQ	NC
AC_10	DO	-	DDR PHY address/command/control signal bit 10	VDDQ	NC
AC_11	DO	-	DDR PHY address/command/control signal bit 11	VDDQ	NC
AC_12	DO	-	DDR PHY address/command/control signal bit 12	VDDQ	NC
AC_13	DO	-	DDR PHY address/command/control signal bit 13	VDDQ	NC
AC_14	DO	-	DDR PHY address/command/control signal bit 14	VDDQ	NC
AC_15	DO	-	DDR PHY address/command/control signal bit 15	VDDQ	NC
AC_16	DO	-	DDR PHY address/command/control signal bit 16	VDDQ	NC
AC_17	DO	-	DDR PHY address/command/control signal bit 17	VDDQ	NC
AC_18	DO	-	DDR PHY address/command/control signal bit 18	VDDQ	NC
AC_20	DO	-	DDR PHY address/command/control signal bit 20	VDDQ	NC
AC_21	DO	-	DDR PHY address/command/control signal bit 21	VDDQ	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
AC_22	DO	-	DDR PHY address/command/control signal bit 22	VDDQ	NC
AC_23	DO	-	DDR PHY address/command/control signal bit 23	VDDQ	NC
AC_24	DO	-	DDR PHY address/command/control signal bit 24	VDDQ	NC
AC_25	DO	-	DDR PHY address/command/control signal bit 25	VDDQ	NC
AC_26	DO	-	DDR PHY address/command/control signal bit 26	VDDQ	NC
AC_28	DO	-	DDR PHY address/command/control signal bit 28	VDDQ	NC
AC_29	DO	-	DDR PHY address/command/control signal bit 29	VDDQ	NC
AC_30	DO	-	DDR PHY address/command/control signal bit 30	VDDQ	NC
AC_31	DO	-	DDR PHY address/command/control signal bit 31	VDDQ	NC
AC_32	DO	-	DDR PHY address/command/control signal bit 32	VDDQ	NC
AC_33	DO	-	DDR PHY address/command/control signal bit 33	VDDQ	NC
AC_34	DO	-	DDR PHY address/command/control signal bit 34	VDDQ	NC
AC_35	DO	-	DDR PHY address/command/control signal bit 35	VDDQ	NC
AC_36	DO	-	DDR PHY address/command/control signal bit 36	VDDQ	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
AC_37	DO	-	DDR PHY address/command/control signal bit 37	VDDQ	NC
AC_38	DO	-	DDR PHY address/command/control signal bit 38	VDDQ	NC
DDR_RSTn	DO	-	DDR3/DDR4/LPDDR4 RSTn	VDDQ	NC
DDR_DQ0	DIO	-	DRAM data bus bit 0	VDDQ	To DRAM
DDR_DQ1	DIO	-	DRAM data bus bit 1	VDDQ	To DRAM
DDR_DQ2	DIO	-	DRAM data bus bit 2	VDDQ	To DRAM
DDR_DQ3	DIO	-	DRAM data bus bit 3	VDDQ	To DRAM
DDR_DQ4	DIO	-	DRAM data bus bit 4	VDDQ	To DRAM
DDR_DQ5	DIO	-	DRAM data bus bit 5	VDDQ	To DRAM
DDR_DQ6	DIO	-	DRAM data bus bit 6	VDDQ	To DRAM
DDR_DQ7	DIO	-	DRAM data bus bit 7	VDDQ	To DRAM
DDR_DQ8	DIO	-	DRAM data bus bit 8	VDDQ	To DRAM
DDR_DQ9	DIO	-	DRAM data bus bit 9	VDDQ	To DRAM
DDR_DQ10	DIO	-	DRAM data bus bit 10	VDDQ	To DRAM
DDR_DQ11	DIO	-	DRAM data bus bit 11	VDDQ	To DRAM
DDR_DQ12	DIO	-	DRAM data bus bit 12	VDDQ	To DRAM
DDR_DQ13	DIO	-	DRAM data bus bit 13	VDDQ	To DRAM
DDR_DQ14	DIO	-	DRAM data bus bit 14	VDDQ	To DRAM
DDR_DQ15	DIO	-	DRAM data bus bit 15	VDDQ	To DRAM
DDR_DQ16	DIO	-	DRAM data bus bit 16	VDDQ	NC
DDR_DQ17	DIO	-	DRAM data bus bit 17	VDDQ	NC
DDR_DQ18	DIO	-	DRAM data bus bit 18	VDDQ	NC
DDR_DQ19	DIO	-	DRAM data bus bit 19	VDDQ	NC
DDR_DQ20	DIO	-	DRAM data bus bit 20	VDDQ	NC
DDR_DQ21	DIO	-	DRAM data bus bit 21	VDDQ	NC
DDR_DQ22	DIO	-	DRAM data bus bit 22	VDDQ	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
DDR_DQ23	DIO	-	DRAM data bus bit 23	VDDQ	NC
DDR_DQ24	DIO	-	DRAM data bus bit 24	VDDQ	NC
DDR_DQ25	DIO	-	DRAM data bus bit 25	VDDQ	NC
DDR_DQ26	DIO	-	DRAM data bus bit 26	VDDQ	NC
DDR_DQ27	DIO	-	DRAM data bus bit 27	VDDQ	NC
DDR_DQ28	DIO	-	DRAM data bus bit 28	VDDQ	NC
DDR_DQ29	DIO	-	DRAM data bus bit 29	VDDQ	NC
DDR_DQ30	DIO	-	DRAM data bus bit 30	VDDQ	NC
DDR_DQ31	DIO	-	DRAM data bus bit 31	VDDQ	NC
DDR_DQM0	DIO	-	DRAM data mask 0	VDDQ	To DRAM
DDR_DQM1	DIO	-	DRAM data mask 1	VDDQ	To DRAM
DDR_DQM2	DIO	-	DRAM data mask 2	VDDQ	NC
DDR_DQM3	DIO	-	DRAM data mask 3	VDDQ	NC
DDR_DQSP0	DIO	-	DRAM data strobe 0	VDDQ	To DRAM
DDR_DQSN0	DIO	-	DRAM data strobe 0 complementary	VDDQ	To DRAM
DDR_DQSP1	DIO	-	DRAM data strobe 1	VDDQ	To DRAM
DDR_DQSN1	DIO	-	DRAM data strobe 1 complementary	VDDQ	To DRAM
DDR_DQSP2	DIO	-	DRAM data strobe 2	VDDQ	NC
DDR_DQSN2	DIO	-	DRAM data strobe 2 complementary	VDDQ	NC
DDR_DQSP3	DIO	-	DRAM data strobe 3	VDDQ	NC
DDR_DQSN3	DIO	-	DRAM data strobe 3 complementary	VDDQ	NC
PZQ	A	-	DRAM reference pin for ZQ calibration, to GND by 240ohm.	VDDQ	To GND by 240ohm
PVREF	A	-	DRAM reference voltage	VDDQ	To GND by capacitor
AVDD_DDRPLL	P		Analog power supply for DDRPLL	-	To DDR VDDQ

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
USB					
USBHOST_A_DP	AIO	-	USB 2.0 Port A positive data signal (Host only)	AVDD33_USB	NC
USBHOST_A_DM	AIO	-	USB 2.0 Port A negative data signal (Host only)	AVDD33_USB	NC
USBOTG_B_DP	AIO	-	USB 2.0 Port B positive data signal (OTG)	AVDD33_USB	NC
USBOTG_B_DM	AIO	-	USB 2.0 Port B negative data signal (OTG)	AVDD33_USB	NC
USBOTG_B_ID	AIO	-	USB OTG mini-receptacle identifier (Internal 12.8KΩ pull-up resistor to AVDD18)	AVDD18_USB	NC
USBOTG_B_VBUS	AIO	-	USB OTG cable power detection	AVDD18_USB	NC
USB_TXRTUNE	AIO	-	USB 2.0 Port A B host output strength setting resistor	AVDD18_USB	NC
AVDD33_USB	P	-	3.3V Power supply for USB	-	To 3.3V
AVDD18_USB	P	-	1.8V Power supply for USB	-	To 1.8V
Ethernet					
ENET_ATP	AIO	-	Ethernet PHY analog test pin	AVDD18_NET	NC
ENET_EXTRES	A	-	Ethernet PHY external resistor connection	AVDD18_NET	NC
ENET_RXN	AIO	-	Ethernet PHY receive date negative input	AVDD18_NET	NC
ENET_RXP	AIO	-	Ethernet PHY receive data positive input	AVDD18_NET	NC
ENET_TXN	AIO	-	Ethernet PHY transmit data negative output	AVDD18_NET	NC
ENET_TXP	AIO	-	Ethernet PHY transmit data positive output	AVDD18_NET	NC
AVDD18_ENET	AP	-	Analog 1.8V power supply for Ethernet module	-	To 1.8V
DSI					
DSI_CEXT	A	-	MIPI DSI external filter capacitor	AVDD18_MIPIDSI	NC
DSI_CLKN	AO	-	MIPI DSI clock negative output	AVDD18_MIPIDSI	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
DSI_CLKP	AO	-	MIPI DSI clock positive output	AVDD18_MIPIDSI	NC
DSI_D0N	AIO	-	MIPI DSI data0 negative output or Bidirectional in LP mode	AVDD18_MIPIDSI	NC
DSI_D0P	AIO	-	MIPI DSI data0 positive output or Bidirectional in LP mode	AVDD18_MIPIDSI	NC
DSI_D1N	AO	-	MIPI DSI data1 negative output	AVDD18_MIPIDSI	NC
DSI_D1P	AO	-	MIPI DSI data1 positive output	AVDD18_MIPIDSI	NC
DSI_D2N	AO	-	MIPI DSI data2 negative output	AVDD18_MIPIDSI	NC
DSI_D2P	AO	-	MIPI DSI data2 positive output	AVDD18_MIPIDSI	NC
DSI_D3N	AO	-	MIPI DSI data3 negative output	AVDD18_MIPIDSI	NC
DSI_D3P	AO	-	MIPI DSI data3 positive output	AVDD18_MIPIDSI	NC
DSI_R1K	A	-	MIPI DSI reference current setting resistor with 1K ohm	AVDD18_MIPIDSI	NC
DSI_R600	A	-	MIPI DSI reference voltage setting resistor with 604 ohm	AVDD18_MIPIDSI	NC
AVDD18_MIPIDSI	AP	-	MIPI-DSI power supply	-	To 1.8V
Audio DAC					
LOLN	AO	-	Audio DAC line-out left channel negative signal	AVDD18_Audio	NC
LOLP	AO	-	Audio DAC line-out left channel positive signal	AVDD18_Audio	NC
LORN	AO	-	Audio DAC line-out right channel negative signal	AVDD18_Audio	NC
LORP	AO	-	Audio DAC line-out right channel positive signal	AVDD18_Audio	NC
REFP	A	-	Audio DAC positive reference voltage	AVDD18_Audio	NC

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
VMID	A	-	Audio DAC external filter cap connection	AVDD18_Audio	NC
AVDD18_AUDIO	AP	-	Analog 1.8V for Audio DAC	-	To 1.8V
AVSS_Audio	AP	-	Analog power ground for Audio DAC	-	To VSS
PCIE					
PCIE_CLK_n	AO	-	PCIE reference clock negative signal	AVDD18_PCIE	NC
PCIE_CLK_p	AO	-	PCIE reference clock positive signal	AVDD18_PCIE	NC
PCIE_REXT	AIO	-	PCIE output strength setting resistor	AVDD18_PCIE	NC
PCIE_RXN	AI	-	PCIE or USB3.0input negative signal	AVDD18_PCIE	NC
PCIE_RXP	AI	-	PCIE or USB3.0input positive signal	AVDD18_PCIE	NC
PCIE_TXN	AO	-	PCIE or USB3.0 output negative signal	AVDD18_PCIE	NC
PCIE_TXP	AO	-	PCIE or USB3.0output positive signal	AVDD18_PCIE	NC
AVDD0V8_USB_PCIE	AP	-	Analog 0.8V power supply for USB and PCIE	-	To VDD_EE
AVDD18_PCIE	AP	-	Analog 1.8V power supply for PCIE	-	To 1.8V
HCSL_REXT	AIO	-	PCIE reference clk output strength setting resistor	AVDD18_PCIE	NC
AVSS_HCSL	AP	-	Analog ground for PCIE reference module clock	-	To VSS
System Clock & PLL					
SYS_OSCIN	AI	-	24MHz crystal oscillator input	VDD18_AO_XTAL	To XTAL
SYS_OSCOUT	AO	-	24MHz crystal oscillator output	VDD18_AO_XTAL	To XTAL
Analog Power					
AVDD18_DPLL	AP	-	Analog power of System PLL	-	To 1.8V
AVSS_ENETPLL	AP	-	Ground of Ethernet PLL		To GND

Net Name	Type	Default Pull UP/DN	Description	Power Domain	If Unused
AVSS_AMPLL	AP	-	Ground of DDR AM_ PLL	-	To GND
AVSS_HPLL	AP	-	Ground of HDMI PLL	-	To GND
AVSS_PLL	AP	-	Ground of System PLL	-	To GND
AVSS_CVBS	AP		Ground of CVBS digital-analog converter	-	To GND
Digital Power					
VDDCPU_A	P	-	Power supply for CPU (Cortex A73)	-	To VDDCPU_A
VDDCPU_B	P	-	Power supply for CPU (Cortex A53)	-	To VDDCPU_B
VDDQ	P	-	DDR IO Power supply for DDR PHY	-	To VDDQ
VDD_DDR	P	-	Core Power supply for DDR PHY	-	To VDD_EE
VDD_EE	P	-	Power supply for GPU and core logic	-	To VDD_EE
VDD18_AO_XTAL	P	-	1.8V Power supply for Always On Domain	-	To VDD18_AO_XTAL
VDDAO_0V8	P	-	0.8V power supply for AO, XTAL and M3/M4 CPU	-	To VDDAO_0V8
VDDIO_A	P	-	Power supply for GPIO bank A	-	NC
VDDIO_BOOT	P		Power supply for GPIO bank BOOT	-	NC
VDDIO_C	P		Power supply for GPIO bank C	-	NC
VDDIO_H	P		Power supply for GPIO bank H	-	NC
VDDIO_X	P		Power supply for GPIO bank X	-	NC
VDDIO_Z	P	-	Power supply for GPIO bank Z	-	NC
Digital Ground					
DVSS	P	-	Digital power ground	-	To GND

Abbreviations:

- DI = Digital input pin
- DO = Digital output pin
- DIO = Digital input/output pin
- OD3.3V = 3.3V input tolerant open drain (OD) output pin, need external pull up
- OD5V = 5V input tolerant open drain (OD) output pin, need external pull up
- A = Analog setting or filtering pin
- AI = Analog input pin
- AO = Analog output pin
- AIO = Analog input/output pin
- P = Power pin
- AP = Analog power pin
- NC = No connection
- UP = Pull-Up
- DOWN = Pull-down
- Z = High-Z

3.4 Pin Multiplexing Tables

Multiple usage pins are used to conserve pin consumption for different features. The S922X devices can be used in many different applications but each application will not utilize all the on chip features. As a result, some of the features share the same pin. Most of the multiple usage pins can be used as a GPIO pin as well.

Table 3-3. GPIOZ_x Multi-Function Pin

Pin Name	Fun1	Func2	Func3	Func4	Func5	Func6	Func7
GPIOZ_0	ETH_MDIO	BT656_A_VS	ISO7816_CLK	I2C_EE_M0_SDA	PWM_B	-	-
GPIOZ_1	ETH_MDC	BT656_A_HS	ISO7816_DATA	I2C_EE_M0_SCL	PWM_C	-	-
GPIOZ_2	ETH_RGMII_RX_CLK	PWM_D	TSIN_B_VALID	TDMC_D0	SDCARD_D0	TDMC_DIN0	PDM_DIN0
GPIOZ_3	ETH_RX_DV	BT656_A_CLK	TSIN_B_SOP	TDMC_D1	SDCARD_D1	TDMC_DIN1	PDM_DIN1
GPIOZ_4	ETH_RXD0	BT656_A_DIN0	TSIN_B_DIN0	TDMC_D2	SDCARD_D2	TDMC_DIN2	PDM_DIN2
GPIOZ_5	ETH_RXD1	BT656_A_DIN1	TSIN_B_CLK	TDMC_D3	SDCARD_D3	TDMC_DIN3	PDM_DIN3
GPIOZ_6	ETH_RXD2_RGMII	BT656_A_DIN2	TSIN_B_FAIL	TDMC_FS	SDCARD_CLK	TDMC_SLV_FS	PDM_DCLK
GPIOZ_7	ETH_RXD3_RGMII	BT656_A_DIN3	TSIN_B_DIN1	TDMC_SCLK	SDCARD_CMD	TDMC_SLV_SCLK	I2C_EE_M0_SDA
GPIOZ_8	ETH_RGMII_TX_CLK	BT656_A_DIN4	TSIN_B_DIN2	MCLK_1	-	-	I2C_EE_M0_SCL
GPIOZ_9	ETH_TXEN	BT656_A_DIN5	TSIN_B_DIN3	-	-	-	-
GPIOZ_10	ETH_TXD0	BT656_A_DIN6	TSIN_B_DIN4	-	IR_REMOTE_OUT		-
GPIOZ_11	ETH_TXD1	BT656_A_DIN7	TSIN_B_DIN5	-	-		-
GPIOZ_12	ETH_TXD2_RGMII	-	TSIN_B_DIN6	-	PWM_F		-
GPIOZ_13	ETH_TXD3_RGMII	CLK12_24	TSIN_B_DIN7	-	PWM_B		GEN_CLK_EE
GPIOZ_14	ETH_LINK_LED	-	I2C_EE_M2_SDA	-	-	-	-
GPIOZ_15	ETH_ACT_LED	-	I2C_EE_M2_SCL	-	-	-	-

Table 3-4. GPIOA_x Multi-Function Pin

Pin Name	Func1	Func2	Func3
GPIOA_0	MCLK_0	-	-
GPIOA_1	TDMB_SCLK	TDMB_SLV_SCLK	-
GPIOA_2	TDMB_FS	TDMB_SLV_FS	-
GPIOA_3	TDMB_D0	TDMB_DIN0	-
GPIOA_4	TDMB_D1	TDMB_DIN1	PWM_D
GPIOA_5	PDM_DIN3	TDMB_DIN2	TDMB_D2
GPIOA_6	PDM_DIN2	TDMB_DIN3	TDMB_D3
GPIOA_7	PDM_DCLK	TDMC_D3	TDMC_DIN3
GPIOA_8	PDM_DIN0	TDMC_D2	TDMC_DIN2
GPIOA_9	PDM_DIN1	TDMC_D1	TDMC_DIN1
GPIOA_10	SPDIF_IN	TDMC_D0	TDMC_DIN0
GPIOA_11	SPDIF_OUT	MCLK_1	PWM_F
GPIOA_12	SPDIF_IN	TDMC_SCLK	TDMC_SLV_SCLK
GPIOA_13	SPDIF_OUT	TDMC_FS	TDMC_SLV_FS
GPIOA_14	WORLD_SYNC	I2C_EE_M3_SDA	-
GPIOA_15	IR_REMOTE_IN	I2C_EE_M3_SCL	-

Table 3-5. BOOT_x Multi-Function Pin

Pin Name	Func1	Func2	Func3
BOOT_0	EMMC_D0	-	-
BOOT_1	EMMC_D1	-	-
BOOT_2	EMMC_D2	-	-
BOOT_3	EMMC_D3	-	NOR_HOLD
BOOT_4	EMMC_D4	-	NOR_D
BOOT_5	EMMC_D5	-	NOR_Q
BOOT_6	EMMC_D6	-	NOR_C
BOOT_7	EMMC_D7	-	NOR_WP
BOOT_8	EMMC_CLK	NAND_WEN_CLK	-
BOOT_9	-	NAND_ALE	-
BOOT_10	EMMC_CMD	NAND_CLE	-
BOOT_11	-	NAND_CE0	-
BOOT_12	-	NAND_REN_WR	-
BOOT_13	EMMC_NAND_DQS	-	-
BOOT_14	-	NAND_RB0	NOR_CS
BOOT_15	-	NAND_CE1	-

Table 3-6. GPIOC_x Multi-Function Pin

Pin Name	Func1	Func2	Func3	Func4	Func5
GPIOC_0	SDCARD_D0	JTAG_B_TDO	-	PDM_DIN0	SPI_A_MOSI
GPIOC_1	SDCARD_D1	JTAG_B_TDI	-	PDM_DIN1	SPI_A_MISO
GPIOC_2	SDCARD_D2	UART_AO_A_RX	-	PDM_DIN2	SPI_A_SS0
GPIOC_3	SDCARD_D3	UART_AO_A_TX	-	PDM_DIN3	SPI_A_SCLK
GPIOC_4	SDCARD_CLK	JTAG_B_CLK	-	PDM_DCLK	PWM_C
GPIOC_5	SDCARD_CMD	JTAG_B_TMS	I2C_EE_M0_SDA	-	ISO7816_CLK
GPIOC_6	-	-	I2C_EE_M0_SCL	-	ISO7816_DATA
GPIOC_7	PCIECK_REQN	WORLD_SYNC	-	-	-

Table 3-7. GPIOX_x Multi-Function Pin

Pin Name	Func1	Func2	Func3	Func4	Func5	Func6	Func7
GPIOX_0	SDIO_D0	PDM_DIN0	TSIN_A_DIN0	-	-	-	-
GPIOX_1	SDIO_D1	PDM_DIN1	TSIN_A_SOP	-	-	-	-
GPIOX_2	SDIO_D2	PDM_DIN2	TSIN_A_VALID	-	-	-	-
GPIOX_3	SDIO_D3	PDM_DIN3	TSIN_A_CLK	PWM_D	-	-	-
GPIOX_4	SDIO_CLK	PDM_DCLK	-	-	-	-	-
GPIOX_5	SDIO_CMD	MCLK_1	-	PWM_C	-	-	-
GPIOX_6	PWM_A	UART_EE_B_TX	-	PWM_D	-	-	-
GPIOX_7	PWM_F	UART_EE_B_RX	-	PWM_B	-	-	-
GPIOX_8	TDMA_D1	TDMA_DIN1	TSIN_B_SOP	SPI_A_MOSI	PWM_C	ISO7816_CLK	-
GPIOX_9	TDMA_D0	TDMA_DIN0	TSIN_B_VALID	SPI_A_MISO	-	ISO7816_DATA	-
GPIOX_10	TDMA_FS	TDMA_SLV_FS	TSIN_B_DIN0	SPI_A_SS0	I2C_EE_M1_SDA	-	-
GPIOX_11	TDMA_SCLK	TDMA_SLV_SCLK	TSIN_B_CLK	SPI_A_SCLK	I2C_EE_M1_SCL	-	-
GPIOX_12	UART_EE_A_TX	-	-	-	-	-	-
GPIOX_13	UART_EE_A_RX	-	-	-	-	-	-
GPIOX_14	UART_EE_A_CTS	-	-	-	-	-	-
GPIOX_15	UART_EE_A_RTS	-	-	-	-	-	-
GPIOX_16	PWM_E	-	-	-	-	-	-
GPIOX_17	I2C_EE_M2_SDA	-	-	-	-	-	-
GPIOX_18	I2C_EE_M2_SCL	-	-	-	-	-	-

Pin Name	Func1	Func2	Func3	Func4	Func5	Func6	Func7
GPIOX_19	PWM_B	WORLD_SYNC	-	-	-	-	GEN_CLK_EE

Table 3-8. GPIOH_x Multi-Function Pin

Pin Name	Func1	Func2	Func3	Func4	Func5	Func6
GPIOH_0	HDMITX_SDA	I2C_EE_M3_SDA	-	-	-	-
GPIOH_1	HDMITX_SCL	I2C_EE_M3_SCL	-	-	-	-
GPIOH_2	HDMITX_HPD_IN	I2C_EE_M1_SDA	-	-	-	-
GPIOH_3	-	I2C_EE_M1_SCL	-	AO_CEC_A	AO_CEC_B	-
GPIOH_4	SPDIF_OUT	UART_EE_C_RTS	SPI_B_MOSI	-	-	-
GPIOH_5	SPDIF_IN	UART_EE_C_CTS	SPI_B_MISO	PWM_F	TDMB_D3	TDMB_DIN3
GPIOH_6	ISO7816_CLK	UART_EE_C_RX	SPI_B_SS0	I2C_EE_M1_SDA	IR_REMOTE_OUT	-
GPIOH_7	ISO7816_DATA	UART_EE_C_TX	SPI_B_SCLK	I2C_EE_M1_SCL	PWM_B	-
GPIOH_8	-	-	-	-	-	-

Table 3-9. GPIOAO_x Multi-Function Pin

Pin Name	Func1	Func2	Func3	Func4	Func5	Func6	Func7
GPIOAO_0	UART_AO_A_TX	-	-	-	-	-	-
GPIOAO_1	UART_AO_A_RX	-	-	-	-	-	-
GPIOAO_2	I2C_AO_M0_SCL	UART_AO_B_TX	I2C_AO_S0_SCL	-	-	-	-
GPIOAO_3	I2C_AO_M0_SDA	UART_AO_B_RX	I2C_AO_S0_SDA	-	-	-	-
GPIOAO_4	IR_REMOTE_OUT	CLK_32K_IN	PWMAO_C	PWMAO_C_HIZ	TDMB_D0	TDMB_DIN0	-

Pin Name	Func1	Func2	Func3	Func4	Func5	Func6	Func7
GPIOAO_5	IR_REMOTE_IN	-	PWMAO_D	-	-	-	-
GPIOAO_6	JTAG_A_CLK	-	PWMAO_C	TSIN_A_SOP	TDMB_D2	TDMB_DIN2	-
GPIOAO_7	JTAG_A_TMS	-	-	TSIN_A_DIN0	TDMB_FS	TDMB_SLV_FS	-
GPIOAO_8	JTAG_A_TDI	-	UART_AO_B_TX	TSIN_A_CLK	TDMB_SCLK	TDMB_SLV_SCLK	-
GPIOAO_9	JTAG_A_TDO	IR_REMOTE_OUT	UART_AO_B_RX	TSIN_A_VALID	MCLK_0	-	-
GPIOAO_10	AO_CEC_A	AO_CEC_B	PWMAO_D	SPDIF_OUT	TDMB_D1	TDMB_DIN1	CLK12_24
GPIOAO_11	-	PWMAO_A_HIZ	PWMAO_A	GEN_CLK_EE	GEN_CLK_AO	-	-

Table 3-10. GPIOE_x Multi-Function Pin

Pin Name	Func1	Func2	Func3	Func4
GPIOE_0	UART_AO_A_CTS	UART_AO_B_CTS	PWMAO_B	I2C_AO_M0_SCL
GPIOE_1	UART_AO_A_RTS	UART_AO_B_RTS	PWMAO_D	I2C_AO_M0_SDA
GPIOE_2	CLK12_24	CLK25_EE	-	-

Table 3-11 DDR AC Multi-Function Pin

Pin Name	LPDDR3	LPDDR4	DDR3	DDR4
AC_0	CKEA0	CKEA0	CKE0	CKE0
AC_1	CKEA1	CKEA1	CKE1	CKE1
AC_2	CSA0	CSA0	CS_N0	CS_N0
AC_3	CSA1	CSA1	NC	NC
AC_4	CLKA_T	CLKA_T	CAS_N	A6
AC_5	CLKA_C	CLKA_C	BA2	A8
AC_6	NC	NC	A7	A2
AC_7	NC	NC	A5	A11
AC_8	CAA2	CAA2	A10	A10
AC_9	CAA7	CAA3	WE_N	BG1

Pin Name	LPDDR3	LPDDR4	DDR3	DDR4
AC_10	CAA1	CAA1	A0	A3
AC_11	CAA4	CAA0	A2	A12
AC_12	CAA5	CAA5	A9	A0
AC_13	CAA6	CAA4	A13	A4
AC_14	CAA0	NC	A14	A13
AC_15	CAA3	NC	A11	A9
AC_16	CAA9	NC	CLK0_T	CLK0_T
AC_17	CAA8	NC	CLK0_C	CLK0_C
AC_18	ODTA	NC	NC	NC
AC_20	NC	CKEB0	CLK1_T	CLK1_T
AC_21	NC	CKEB1	CLK1_C	CLK1_C
AC_22	NC	CSB1	NC	NC
AC_23	NC	CSB0	NC	NC
AC_24	NC	CLKB_T	A6	A5
AC_25	NC	CLKB_C	A4	BA1
AC_26	NC	NC	A1	A1
AC_28	NC	CAB1	A8	A7
AC_29	NC	CAB3	BA1	RAS_N
AC_30	NC	CAB5	A15	ACT_N
AC_31	NC	CAB2	RAS_N	WE_N
AC_32	NC	CAB4	NC	NC
AC_33	NC	CAB0	A12	CAS_N
AC_34	NC	NC	A3	BA0
AC_35	NC	NC	BA0	BG0
AC_36	NC	NC	ODT0	ODT0
AC_37	NC	NC	ODT1	ODT1
AC_38	NC	NC	CS_N1	CS_N1
DDR_RSTn	NC	RESET_N	RESET_N	RESET_N
PVREF	PVREF	PVREF	PVREF	PVREF

Pin Name	LPDDR3	LPDDR4	DDR3	DDR4
PZQ	PZQ	PZQ	PZQ	PZQ

Table 3-12. PCIE IO Multi-Function Pin

Pin Name	Func1	Func2
PCIE_RXN	PCIE_RXN	USB3.0_RXN
PCIE_RXP	PCIE_RXP	USB3.0_RXP
PCIE_TXN	PCIE_TXN	USB3.0_TXN
PCIE_TXP	PCIE_TXP	USB3.0_TXP

3.5 Signal Descriptions

Table 3-13. SD Card Interface Signal Description

Signal Name	Type	Description
SDCARD_D0	DIO	SD Card data bus bit 0 signal
SDCARD_D1	DIO	SD Card data bus bit 1 signal
SDCARD_D2	DIO	SD Card data bus bit 2 signal
SDCARD_D3	DIO	SD Card data bus bit 3 signal
SDCARD_CLK	DO	SD Card clock signal
SDCARD_CMD	DIO	SD Card command signal

Table 3-14. SDIO Interface Signal Description

Signal Name	Type	Description
SDIO_D0	DIO	SDIO data bus bit 0 signal
SDIO_D1	DIO	SDIO data bus bit 1 signal
SDIO_D2	DIO	SDIO data bus bit 2 signal
SDIO_D3	DIO	SDIO data bus bit 3 signal
SDIO_CLK	DO	SDIO clock signal
SDIO_CMD	DIO	SDIO command signal

Table 3-15. Clock Interface Signal Description

Signal Name	Type	Description
CLK_32K_IN	DI	32KHz clock input
CLK12_24	DO	12MHz/24MHZ clock output
CLK25_EE	DO	25MHz clock output

Table 3-16. UART Interface Signal Description

Signal Name	Type	Description
UART_AO_A _ TX	DO	UART Port A data output in AO domain
UART_AO_A_RX	DI	UART Port A data input in AO domain
UART_AO_ A_ CTS	DI	UART Port A Clear To Send Signal in AO domain
UART_AO_A _ RTS	DO	UART Port A Ready To Send Signal in AO domain
UART_AO_B _ TX	DO	UART Port B data output in AO domain
UART_AO_B _ RX	DI	UART Port B data input in AO domain
UART_AO_B _ CTS	DI	UART Port B Clear To Send Signal in AO domain
UART_AO_B _ RTS	DO	UART Port B Ready To Send Signal in AO domain
UART_EE_A _TX	DO	UART Port A data output in EE domain
UART_EE _A_RX	DI	UART Port A data input in EE domain
UART_EE _A_CTS	DI	UART Port A Clear To Send Signal in EE domain
UART_EE _A_RTS	DO	UART Port A Ready To Send Signal in EE domain
UART_EE _B_TX	DO	UART Port B data output in EE domain
UART_EE _B_RX	DI	UART Port B data input in EE domain
UART_EE _B_CTS	DI	UART Port B Clear To Send Signal in EE domain
UART_EE _B_RTS	DO	UART Port B Ready To Send Signal in EE domain
UART_EE _C_TX	DO	UART Port C data output in EE domain
UART_EE _C_RX	DI	UART Port C data input in EE domain
UART_EE _C_CTS	DI	UART Port C Clear To Send Signal in EE domain
UART_EE _C_RTS	DO	UART Port C Ready To Send Signal in EE domain

Table 3-17. ISO7816 Interface Signal Description

Signal Name	Type	Description
ISO7816_DATA	DIO	ISO7816 data signal
ISO7816_CLK	DO	ISO7816 clock signal

Table 3-18. TS In Interface Signal Description

Signal Name	Type	Description

TSIN_A_DIN0	DI	Serial TS input port A data
TSIN_A_CLK	DI	TS input port A clock
TSIN_A_SOP	DI	TS input port A start of stream signal
TSIN_A_VALID	DI	TS input port A date valid signal
TSIN_B_DIN0	DI	Serial/Parallel TS input port B data 0
TSIN_B_DIN1	DI	Parallel TS input port B data 1
TSIN_B_DIN2	DI	Parallel TS input port B data 2
TSIN_B_DIN3	DI	Parallel TS input port B data 3
TSIN_B_DIN4	DI	Parallel TS input port B data 4
TSIN_B_DIN5	DI	Parallel TS input port B data 5
TSIN_B_DIN6	DI	Parallel TS input port B data 6
TSIN_B_DIN7	DI	Parallel TS input port B data 7
TSIN_B_FAIL	DI	TS input port B fail signal
TSIN_B_CLK	DI	TS input port B clock
TSIN_B_SOP	DI	TS input port B start of stream signal
TSIN_B_VALID	DI	TS input port B date valid signal

Table 3-19. PWM Interface Signal Description

Signal Name	Type	Description
PWM_A	DO	PWM channel A output signal
PWM_B	DO	PWM channel B output signal
PWM_C	DO	PWM channel C output signal
PWM_D	DO	PWM channel D output signal
PWM_E	DO	PWM channel E output signal
PWM_F	DO	PWM channel F output signal
PWMAO_A / PWMAO_A_HIZ	DO	PWM A output signal in Always On domain, or extended HiZ function of PWMAO_A
PWMAO_B	DO	PWM B output signal in Always On domain
PWMAO_C / PWMAO_C_HIZ	DO	PWM C output signal in Always On domain, or extended HiZ function of PWMAO_C
PWMAO_D	DO	PWM D output signal in Always On domain

Table 3-20. I2C Interface Signal Description

Signal Name	Type	Description
I2C_AO_ M0_ SCL	DO	I2C bus port 0 clock output, Master mode, in AO domain
I2C_AO _M0_SDA	DIO	I2C bus port 0 data input/output, Master mode, in AO domain
I2C_AO_S0_SCL	DI	I2C bus port 0 clock input, Slave mode, in AO domain
I2C_AO_S0_SDA	DIO	I2C bus port 0 data input/output, Slave mode, in AO domain
I2C_EE_M0_SCL	DO	I2C bus port 0 clock output, Master mode, in EE domain
I2C_EE _M0_SDA	DIO	I2C bus port 0 data input/output, Master mode, in EE domain
I2C_EE_M1_SCL	DO	I2C bus port 1 clock output, Master mode, in EE domain
I2C_EE _M1_SDA	DIO	I2C bus port 1 data input/output, Master mode, in EE domain
I2C_EE_M2_SCL	DO	I2C bus port 2 clock output, Master mode, in EE domain
I2C_EE _M2_SDA	DIO	I2C bus port 2 data input/output, Master mode, in EE domain
I2C_EE_M3_SCL	DO	I2C bus port 3 clock output, Master mode, in EE domain
I2C_EE_M3_SDA	DIO	I2C bus port 3 data input/output, Master mode, in EE domain

Table 3-21. eMMC Interface Signal Description

Signal Name	Type	Description
EMMC_D0	DIO	eMMC/NAND data bus bit 0 signal
EMMC_D1	DIO	eMMC/NAND data bus bit 1 signal
EMMC_D2	DIO	eMMC/NAND data bus bit 2 signal
EMMC_D3	DIO	eMMC/NAND data bus bit 3 signal
EMMC_D4	DIO	eMMC/NAND data bus bit 4 signal
EMMC_D5	DIO	eMMC/NAND data bus bit 5 signal
EMMC_D6	DIO	eMMC/NAND data bus bit 6 signal
EMMC_D7	DIO	eMMC/NAND data bus bit 7 signal
EMMC_CLK	DO	eMMC clock signal
EMMC_CMD	DIO	eMMC command signal

Signal Name	Type	Description
EMMC_NAND_DQS	DIO	eMMC/NAND data strobe

Table 3-22. NAND Signal Description

Signal Name	Type	Description
NAND_RB0	DI	NAND ready/busy
NAND_ALE	DO	NAND address latch enable
NAND_CE0	DO	NAND chip enable 0
NAND_CE1	DO	NAND chip enable 1
NAND_CLE	DO	NAND command latch enable
NAND_REN_WR	DO	NAND read enable or write/read
NAND_WEN_CLK	DO	NAND write enable or clock

Table 3-23. NOR Interface Signal Description

Signal Name	Type	Description
NOR_CS	DO	SPI NOR chip select
NOR_C	DO	SPI NOR Serial Clock
NOR_D	DIO	SPI NOR 1bit mode Output, 2/4 bit mode data I/O 0
NOR_Q	DIO	SPI NOR 1bit mode Input, 2/4 bit mode data I/O 1
NOR_WP	DIO	SPI NOR Write protection output, 4 bit mode data I/O 2
NOR_HOLD	DIO	SPI bus hold output, 4 bit mode data I/O 3

Table 3-24. HDMI Interface Signal Description

Signal Name	Type	Description
HDMITX_SDA	DIO	HDMI TX DDC_I2C interface data signal
HDMITX_SCL	DO	HDMI TX DDC_I2C interface clock signal
HDMITX_HPD_IN	DI	HDMI TX hot-plug in signal input
AO_CEC_A	DIO	Customer Electronics Control signal in AO domain
AO_CEC_B	DIO	2nd pin of Customer Electronics Control signal in AO domain

Table 3-25. SPDIF Interface Signal Description

Signal Name	Type	Description
SPDIF_IN	DI	SPDIF input signal
SPDIF_OUT	DO	SPDIF output signal

Table 3-26. PCIE Interface Signal Description

Signal Name	Type	Description
PCIECK_REQN	DI	PCIE clock request input

Table 3-27. SPI Interface Signal Description

Signal Name	Type	Description
SPI_A_MOSI	DIO	SPI master output, slave input A
SPI_A_MISO	DIO	SPI master input, slave output A
SPI_A_SCLK	DIO	SPI clock A
SPI_A_SS0	DIO	SPI slave select 0 A
SPI_B_MOSI	DIO	SPI master output, slave input B
SPI_B_MISO	DIO	SPI master input, slave output B
SPI_B_SCLK	DIO	SPI clock B
SPI_B_SS0	DIO	SPI slave select 0 B

Table 3-28. Remote Interface Signal Description

Signal Name	Type	Description
IR_REMOTE_IN	DI	IR remote control input
IR_REMOTE_OUT	DO	IR remote control output

Table 3-29. Time Division Multiplexing Signal Description

Signal Name	Type	Description
MCLK_0	DO	Master clock output 0, for I2S master mode
MCLK_1	DO	Master clock output 1, for I2S master mode
TDMA_DIN0	DI	Data input 0 of TDM port A

Signal Name	Type	Description
TDMA_DIN1	DI	Data input 1 of TDM port A
TDMA_D0	DIO	Data input/output 0 of TDM port A
TDMA_D1	DIO	Data input/output 1 of TDM port A
TDMA_SCLK	DO	Bit clock output of TDM port A
TDMA_FS	DO	Frame sync output of TDM port A (Word clock of I2S)
TDMA_SLV_SCLK	DI	Bit clock input of TDM port A
TDMA_SLV_FS	DI	Frame sync input of TDM port A (Word clock of I2S)
TDMB_DIN0	DI	Data input 0 of TDM port B
TDMB_DIN1	DI	Data input 1 of TDM port B
TDMB_DIN2	DI	Data input 2 of TDM port B
TDMB_DIN3	DI	Data input 3 of TDM port B
TDMB_D0	DIO	Data input/output 0 of TDM port B
TDMB_D1	DIO	Data input/output 1 of TDM port B
TDMB_D2	DIO	Data input/output 2 of TDM port B
TDMB_D3	DIO	Data input/output 3 of TDM port B
TDMB_SCLK	DO	Bit clock output of TDM port B
TDMB_FS	DO	Frame sync output of TDM port B (Word clock of I2S)
TDMB_SLV_SCLK	DI	Bit clock input of TDM port B
TDMB_SLV_FS	DI	Frame sync input of TDM port B (Word clock of I2S)
TDMC_DIN0	DI	Data input 0 of TDM port C
TDMC_DIN1	DI	Data input 1 of TDM port C
TDMC_DIN2	DI	Data input 2 of TDM port C
TDMC_DIN3	DI	Data input 3 of TDM port C
TDMC_D0	DIO	Data input/output 0 of TDM port C
TDMC_D1	DIO	Data input/output 1 of TDM port C
TDMC_D2	DIO	Data input/output 2 of TDM port C
TDMC_D3	DIO	Data input/output 3 of TDM port C
TDMC_SCLK	DO	Bit clock output of TDM port C
TDMC_FS	DO	Frame sync output of TDM port C (Word clock of I2S)

Signal Name	Type	Description
TDMC_SLV_SCLK	DI	Bit clock input of TDM port C
TDMC_SLV_FS	DI	Frame sync input of TDM port C (Word clock of I2S)

Table 3-30. PDM Signal Description

Signal Name	Type	Description
PDM_DIN0	DI	PDM input data 0 signal
PDM_DIN1	DI	PDM input data 1 signal
PDM_DIN2	DI	PDM input data 2 signal
PDM_DIN3	DI	PDM input data 3 signal
PDM_DCLK	DO	PDM output clock signal

Table 3-31. JTAG Interface Signal Description

Signal Name	Type	Description
JTAG_A_TDO	DO	JTAG data output channel A
JTAG_A _TDI	DI	JTAG data input channel A
JTAG_A_TMS	DI	JTAG Test mode select input channel A
JTAG_A_CLK	DI	JTAG Test clock input channel A
JTAG_B_TDO	DO	JTAG data output channel B
JTAG_B _TDI	DI	JTAG data input channel B
JTAG_B_TMS	DI	JTAG Test mode select input channel B
JTAG_B _CLK	DI	JTAG Test clock input channel B

Table 3-32. BT656 Interface Signal Description

Signal Name	Type	Description
BT656_A_DIN0	DI	BT656 input data bus bit 0
BT656_A_DIN1	DI	BT656 input data bus bit 1
BT656_A_DIN2	DI	BT656 input data bus bit 2
BT656_A_DIN3	DI	BT656 input data bus bit 3
BT656_A_DIN4	DI	BT656 input data bus bit 4
BT656_A_DIN5	DI	BT656 input data bus bit 5

BT656_A_DIN6	DI	BT656 input data bus bit 6
BT656_A_DIN7	DI	BT656 input data bus bit 7
BT656_A_CLK	DI	BT656 input Clock
BT656_A_HS	DI	BT656 input HSYNC Signal
BT656_A_VS	DI	BT656 input VSYNC Signal

Table 3-33. Ethernet Interface Signal Description

Signal Name	Type	Description
ETH_LINK_LED	DO	Ethernet link LED indicator
ETH_ACT_LED	DO	Ethernet active LED indicator
ETH_RGMII_RX_CLK	DI	Ethernet RGMII interface receive clock input
ETH_RGMII_TX_CLK	DO	Ethernet RGMII transmit clock
ETH_TX_EN	DO	Ethernet RMII/RGMII Interface transmit enable
ETH_TXD3	DO	Ethernet RGMII interface transmit data 3
ETH_TXD2	DO	Ethernet RGMII interface transmit data 2
ETH_TXD1	DO	Ethernet RMII/RGMII interface transmit data 1
ETH_TXD0	DO	Ethernet RMII/RGMII interface transmit data 0
ETH_RX_DV	DI	Ethernet RMII/RGMII interface receive data valid signal
ETH_RXD3	DI	Ethernet RGMII interface receive data 3
ETH_RXD2	DI	Ethernet RGMII interface receive data 2
ETH_RXD1	DI	Ethernet RMII/RGMII interface receive data 1
ETH_RXD0	DI	Ethernet RMII/RGMII interface receive data 0
ETH_MDIO	DIO	Ethernet SMI interface management data input/output
ETH_MDC	DO	Ethernet SMI interface management clock

Table 3-34. Other Signal Description

Signal Name	Type	Description
WORLD_SYNC	DI	World clock sync input, to sync clock of multi devices
GEN_CLK_EE	DO	General clock output for EE domain clock, for debug
GEN_CLK_AO	DO	General clock output for AO domain clock, for debug

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

The table below gives the absolute maximum ratings. Exposure to stresses beyond those listed in this table may result in permanent device damage, unreliability or both.

Characteristic	Value	Unit
VDDCPU_A/B Supply Voltage	1.1	V
VDD_EE Supply Voltage	1.0	V
VDDQ Supply Voltage	1.7	V
1.8V Supply Voltage	1.98	V
3.3V Supply Voltage	3.63	V
Input voltage, V_I	-0.3 ~ VDDIO+0.3	V
Junction Temperature	125	°C

4.2 Recommended Operating Conditions

Note:

All voltage specs listed in this part are applicable at the pins of the chip, not the output of the DC-DC.

Symbol	Parameter	Min.	Typ.	Max.	Unit
VDDCPU_A/B	Voltage for Cortex A73/A53 CPU	0.68[1]	-	1.03[2]	V
VDD_EE and other 0.8V domain	Voltage for GPU & core logic	0.77	0.8	0.9[2]	V
VDDQ	DDR3/DDR3L/DDR4/LPDDR/LPDDR3/LPDDR4 IO Supply Voltage	1.05	-	1.6	V
AVDD18	1.8V AVDD for HDMI, USB, SARADC, PCIE, CVBS, AUDIO, MIPI_DSI and ETHERNET phy.	1.71	1.80	1.89	V
VDD18_AO_XTAL	1.8V VDD for XTAL, and IOVREF	1.71	1.80	1.89	V
AVDD_DDRPLL	Analog power supply for DDRPLL module	1.05	-	1.89	V
AVDD33	3.3V AVDD for USB	3.15	3.3	3.45	V
VDDIO	LV mode	1.71	1.80	1.89	V
	HV mode	3.0[3]	3.3	3.45	V
T_J	Operating Junction Temperature	0	-	105[4]	°C
T_A	Operating Ambient Temperature	0	-	70	°C

Note:

1) Minimal VDDCPU_A/B voltage is for sleep mode while system runs at very low speed. Higher clock will need higher voltage. Considering the power supply may have 3% deviation, the minimal voltage in actual application should not be set to lower than min spec plus 0.02V.
2) Likewise, maximum VDDCPU_A/B voltage in actual application should not be higher than max spec minus 0.03V. Voltage of VDDCPU_A/B will affect CPU speed. Use lower voltage when CPU runs on lower speed to save power. Recommend to use +/-1.5% or higher precision DCDC.
3) GPIO cannot work if VDDIO voltage is out of the spec of LV / HV mode. GPIO output at HV mode will be weaker & max operating speed will be lower if VDDIO are design to 3.0V. Do not design VDDIO to lower than 3.0V in HV mode, recommend to use +/-1.5% or higher precision DCDC to supply power for VDDIO, actual voltage supplies to VDDIO (HV mode) should not be lower than 2.9V.
4) For operating temperature, good heat sink may be needed to guarantee Tj < max spec.

4.3 Ripple Voltage Specification

Please check below table for ripple votage specification.

Power	Max Ripple	Unit	Test state
VDDCPU_A	40	+/-mV	Run APK StabilityTest
VDDCPU_B	40	+/-mV	Run APK StabilityTest
VDD_EE and other 0.8V domain	40	+/-mV	Run APK Basemark ES 2.0 Taiji
DDR3 VDDQ and AVDD_DDRPLL	60	+/-mV	Kernel boot
DDR3L VDDQ and AVDD_DDRPLL	60	+/-mV	Kernel boot
LPDDR3 VDDQ and AVDD_DDRPLL	40	+/-mV	Kernel boot
DDR4 VDDQ and AVDD_DDRPLL	40	+/-mV	Kernel boot
LPDDR4 VDDQ and AVDD_DDRPLL	40	+/-mV	Kernel boot
AVDD18	30	+/-mV	Kernel boot
VDD18_AO_XTAL	30	+/-mV	Kernel boot
AVDD33	50	+/-mV	WIFI SCAN
VDDIO LV	60	+/-mV	Kernel boot
VDDIO HV	60	+/-mV	WIFI SCAN

Note:

Ripple specification is only a reference spec, customer should run stress/performance/reliability test (high/low temperature test , damp and hot test , function test , etc...) on their product to confirm the system stability

4.4 Thermal Resistance

Jedec 2P2S board 101.5mm*114.5mm, natural convection, ambient temperature 25°C.

Symbol	Parameter	Value (°C/Watt)	Air Flow (m/s)
Θja	Package junction-to- ambiance thermal resistance in nature convection	15.0521	0
Θjb	Package junction-to-pcb thermal resistance in nature convection	6.26571	0
Θjc	Package junction-to-case thermal resistance in nature convection	5.57433	0

Note:

1.) Due to the thinness of the SOC, DRAM or capacitors placed close to SOC may prevent heatsink touching SOC top side. A special convex shape heatsink is recommended.

2.) These measurement were conducted on a JEDEC defined 2S2P system. For more information, check below JEDEC standards:

- JESD51-2A： Integrated Circuits Thermal Test Method Environmental Conditions - Natural Convection (Still Air)
- JESD51-8：Integrated Circuit Thermal Test Method Environmental Conditions — Junction-to-Board
- JESD51-12：Guidelines for Reporting and Using Electronic Package Thermal Information

3.) m/s = meters per second

4.5 DC Electrical Characteristics

Note:

All voltage specs listed in this part are applicable at the pins of the chip, not the output of the DC-DC.

4.5.1 Normal GPIO Specifications (For DIO)

Symbol	Parameter	Min.	Typ.	Max.	Unit
$V_{iH(VDDIO=3.3V)}$[3]	High-level input voltage	IOVREF+0.37	-	VDDIO+0.3	V
$V_{iL(VDDIO=3.3V)}$[3]	Low-level input voltage	-0.3	-	IOVREF-0.23	V
$V_{iH(VDDIO=1.8V)}$[3]	High-level input voltage	IOVREF/2+0.3	-	VDDIO+0.3	V
$V_{iL(VDDIO=1.8V)}$[3]	Low-level input voltage	-0.3	-	IOVREF/2-0.3	V
R_{PU}	Built-in pull up resistor	50K	60K	70K	ohm
R_{PD}	Built-in pull down resistor	50K[5]	60K	130K[6]	ohm
IoL/IoH(DS=0)[1,4]	GPIO driving capability	0.5	-	-	mA
IoL/IoH(DS=1)[1]	GPIO driving capability	2.5	-	-	mA
IoL/IoH(DS=2)[1]	GPIO driving capability	3	-	-	mA
IoL/IoH(DS=3)[1]	GPIO driving capability	4[2]	-	-	mA
VOH	Output high level with IoL/IoH loading	VDDIO-0.5	-	-	V
VOL	Output low level with IoL/IoH loading	-	-	0.4	V

Note:

1. With Minimal IoL/IoH driving capability loading, IO is garanteed to meet Vol < 0.4V or VOH > (VDDIO-0.5V) spec.
2. Maximal GPIO loading is 6mA for application such as driving LED, which does not care about Vol/Voh spec. Please set DS=3 for such application.
3. VDD18_AO supplies power to IOVREF.

4. Do not use this setting, it's too weak for most applications.
5. Test condition, GPIO pin voltage close to 0 V.
6. Test condition, GPIO pin voltage close to VDDIO(3.3 V).

4.5.2 Open Drain GPIO Specifications (For DIO_OD)

Symbol	Parameter	Min.	Typ.	Max.	Unit
$V_{iH(OD5V)}$	High-level input voltage	1.5		5.5	V
$V_{iL(OD5V)}$	Low-level input voltage	-0.3		0.8	V
$V_{iH(OD3.3V)}$	High-level input voltage	1.5		3.6	V
$V_{iL(OD3.3V)}$	Low-level input voltage	-0.3		0.8	V
$R_{PU/PD}$	No built-in pull up/down resistor on OD IO	-	-	-	ohm
Io	OD IO driving low capability	4		6	mA
VOL	Output low level with min Io loading			0.4	V

Note:

1. With Minimal IoL driving capability loading, IO is garanteed to meet Vol<0.4V spec
2. Maximal GPIO loading is 6mA for application such as driving LED, which does not care about Vol spec.
3. The Vil/Vih of OD PAD is irrelevant to VDDIO voltage

4.5.3 DDR3/DDR3L/DDR4/LPDDR3/LPDDR4 SDRAM Specifications

Recommended Operating Conditions

Symbol	Parameter	Min.	Typ.	Max.	Unit
VDDQ	IO supply voltage(DDR3)	1.425	1.50	1.57	V
VDDQ	IO supply voltage(DDR3L)	1.283	1.35	1.45	V
VDDQ	IO supply voltage(DDR4)	1.14	1.20	1.30	V
VDDQ	IO supply voltage(LPDDR3)	1.14	1.2	1.30	V
VDDQ	IO supply voltage(LPDDR4)	1.06	1.1	1.17	V
Vref	Input reference supply voltage	0.49*VDDQ	0.5*VDDQ	0.51*VDDQ	V

Note: The minimal VDDQ voltage in sleep mode is defined by memory.

DC specifications - DDR3/DDR3L mode

Symbol	Parameter	Min.	Typ.	Max.	Unit
VIH	DC input voltage high	Vref + 0.100		VDDQ	V
VIL	DC input voltage low	VSSQ		Vref-0.100	V
VOH	DC output logic high	0.8*VDDQ			V
VOL	DC output logic low			0.2*VDDQ	V
RTT	Input termination resistance to VDDQ/2	100 54 36	120 60 40	140 66 44	ohm

DC specifications – DDR4 mode

Symbol	Parameter	Min.	Typ.	Max.	Unit
VdlVW_total	Rx Mask voltage-p-p total			136	mv
VOH	DC output logic high	0.9*VDDQ			V
VOL	DC output logic low			0.1*VDDQ	V
RTT	Input termination resistance to VDDQ	200	240	280	ohm
		100	120	140	
		67	80	93	
		50	60	70	
		42	48	56	
		34	40	46	
		28	34	40	

DC Specifications – LPDDR3 mode

Symbol	Parameter	Min.	Typ.	Max.	Unit
VIH	DC input voltage high	Vref + 0.100		VDDQ	V
VIL	DC input voltage low	VSSQ		Vref-0.100	V
VOH	DC output logic high	0.9*VDDQ			V
VOL	DC output logic low			0.1*VDDQ	V
RTT	Input termination resistance to VDDQ	100 200	120 240	140 280	ohm

DC Specifications – LPDDR4 mode

Symbol	Parameter	Min.	Typ.	Max.	Unit
VOH	DC output logic high	0.9*VDDQ	-	-	V
VOL	DC output logic low	-	-	0.1*VDDQ	V
RTT	Input termination resistance to VDDQ	216	240	264	ohm
		108	120	132	
		72	80	88	
		54	60	66	
		43.2	48	52.8	
		36	40	44	

4.6 Recommended Oscillator Electrical Characteristics

S922X requires the 24MHz oscillator for generating the main clock source.

Table 4-1. 24MHz Oscillator Specification

Symbol	Description	Min.	Typ.	Max.	Unit	Notes
F_o	Nominal Frequency		24		MHz	
$\Delta f/f_o$	Frequency Tolerance	-30		+30	ppm	At 25 ˚C
		-50		+50	ppm	At -20~85 ˚C
C_L	Load Capacitance	7.5	12	12.5	pF	
ESR	Equivalent Series Resistance			100	oHm	

Note:

1. 10ppm Tolerance is preferred if 24MHz XTAL is also driving WIFI module.
2. For user external clock source，Please connect input clock output to SYS_OSCIN , let SYS_OSCOUT floating.
3. The threshold of Xin inverter is around 0.9V (Xin range: -0.3V to +2.1V). Therefore, Following suggestion for input clock.

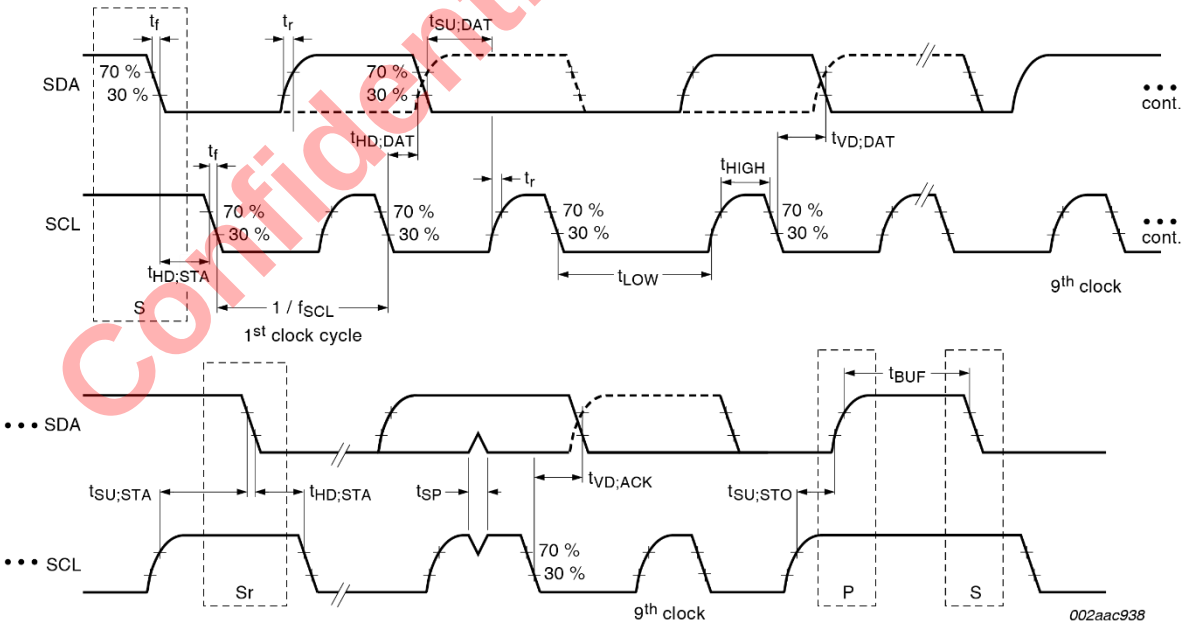
- Suggestion 1: Without DC blocking capacitor, use a higher Vpp output TCXO. The high voltage should be higher than 1.35V (VSWING >1.35V, 0V to >1.35V).
- Suggestion 2: With DC blocking capacitor, re-bias the middle voltage at 0.9V, VSWING >2*0.45V;

4.7 Timing Information

4.7.1 I2C Timing Specification

The I2C master interface Fast/Standard mode timing specifications are shown below.

Figure 4-1. I2C Interface Timing Diagram, FS mode



$V_{IL} = 0.3V_{DD}$

$V_{IH} = 0.7V_{DD}$

Table 4-2. I2C Interface Timing Specification, SF mode

Symbol	Parameter	Standard-mode		Fast-mode		Unit
		Min.	Max	Min	Max	
tR	Rise time of SDA and SCL signals	—	1000	—	300	ns
tF	Fall time of SDA and SCL signals	—	300	—	300	ns
fSCL	SCL clock frequency	—	100	—	400	KHz
tLOW	LOW period of the SCL clock	4.7	—	1.3	—	µs
tHIGH	HIGH period of the SCL clock	4.0	—	0.6	—	µs
tSu;STA	Setup time for START	4.7	—	0.6	—	µs
tSu;DAT	Setup time for SDA	250	—	100	—	ns
tSu;STO	Setup time for STOP	4.0	—	0.6	—	µs
tHd;STA	Hold time for START	4.0	—	0.6	—	µs
tHd;DAT	Hold time for SDA	0	3.45	0	0.9	µs
tBuf	Bus free time between stop and start	4.7	—	1.3	—	µs

Note:

Driving strength of OD pin is not adjustable.

4.7.2 EMMC/SDIO Timing Specification

Timing specification for EMMC and SD are shown as below.

Figure 4-2. EMMC HS200 Data Output Timing

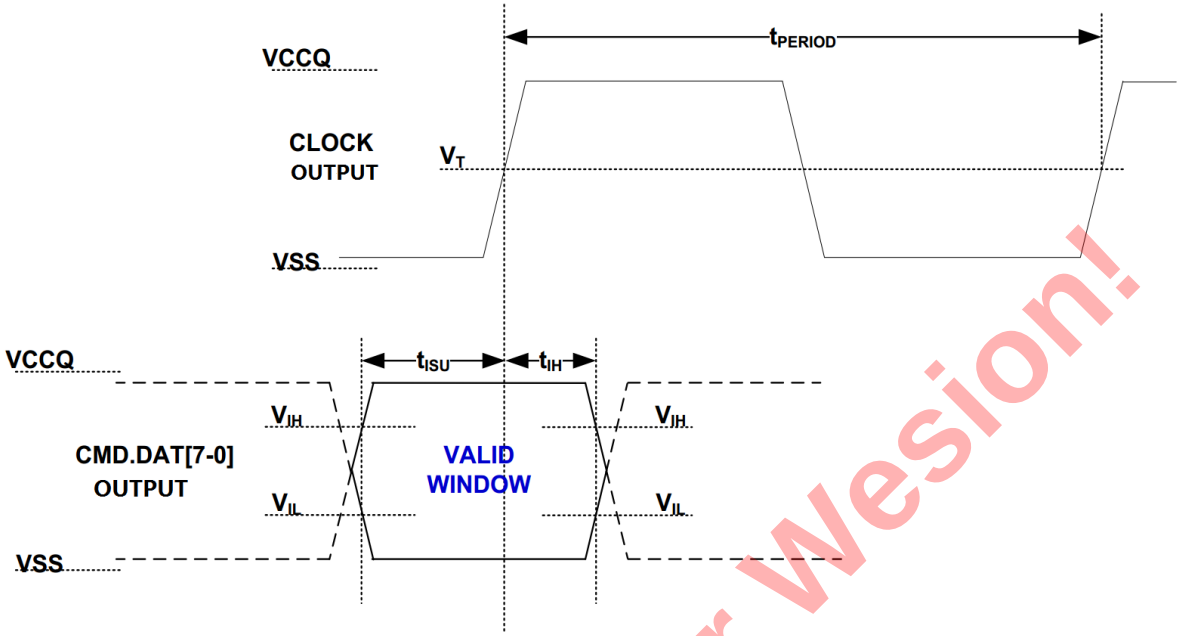


Table 4-3 HS200 Timing Specification

Symbol	Parameter	Min	Max	Unit
tPERIOD	Cycle time data transfer mode	5	-	ns
tISU	output set-up time	1.4	-	ns
tIH	output hold time	0.8	-	ns

Figure 4-3. EMMC HS200 Data Input Timing

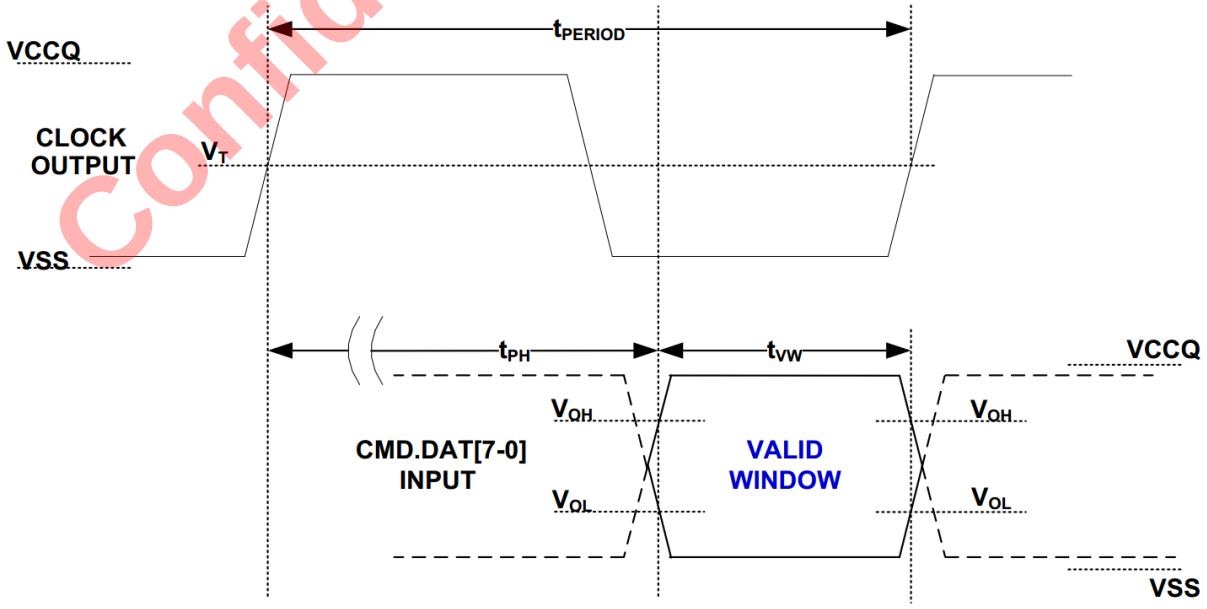


Table 4-4 HS200 Timing Specification

Symbol	Parameter	Min	Max	Unit
tPH	Device output momentary phase from CLK input to CMD or DAT line output. Does not include a longterm temperature drift.	0	2	UI
ΔTPH	Delay variation due to temperature change after tuning. Total allowable shift of output valid window (TVW) from last system Tuning procedure Δ TPH is 2600ps for ΔT from -25 °C to 125 °C during operation.	-350(ΔT=-20deg.C)	1550(ΔT=90deg.C)	ps
tVW	Valid Data Simple window	0.575	-	UI

Figure 4-4. SDIO(SDR104) Clock Signal Timing Diagram

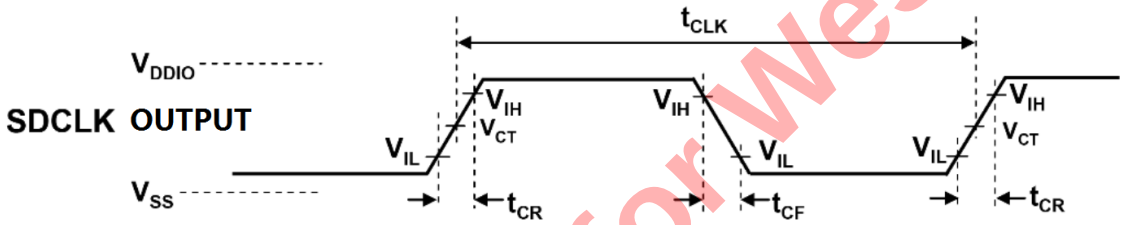


Table 4-5 SDIO(SDR104) Clock Timing Specification

Symbol	Parameter (SDR104 Mode)	Min	Max	Unit
tCLK	clock period Data Transfer Mode(PP)	4.8	-	ns
Duty	Clock Duty	30	70	%
tCR	clock rise time	-	0.96	ns
tCF	clock fall time	-	0.96	ns

Figure 4-5. SDIO(SDR104) Output Timing Diagram

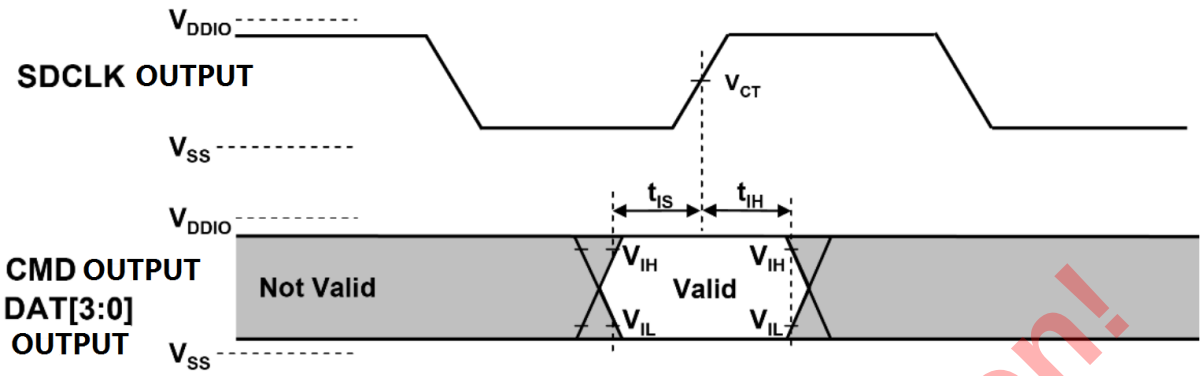


Table 4-6 SDIO(SDR104) Timing Specification

Inputs CMD, DAT (referenced to CLK)				
Symbol	**Parameter**	**Min**	**Max**	**Unit**
tIS	input set-up time	1.4	-	ns
tIH	input hold time	0.8	-	ns

Note:SD card interface uses SDIO protocol

4.7.3 NAND Timing Specification

Nand timing specifications are shown as below.

Figure 4-6 Async Waveform for Command/Address/Data Output Timing

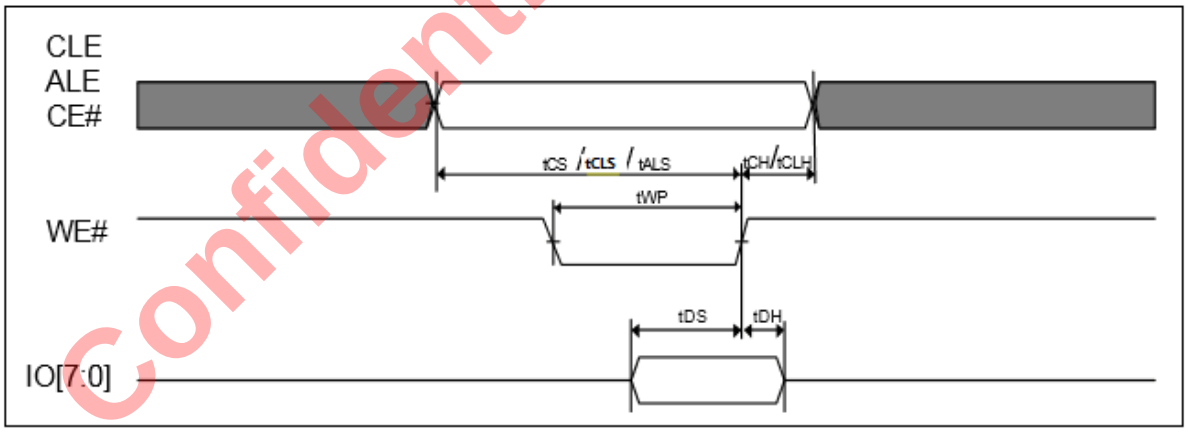


Figure 4-7 Async Waveform for Address Output Cycle

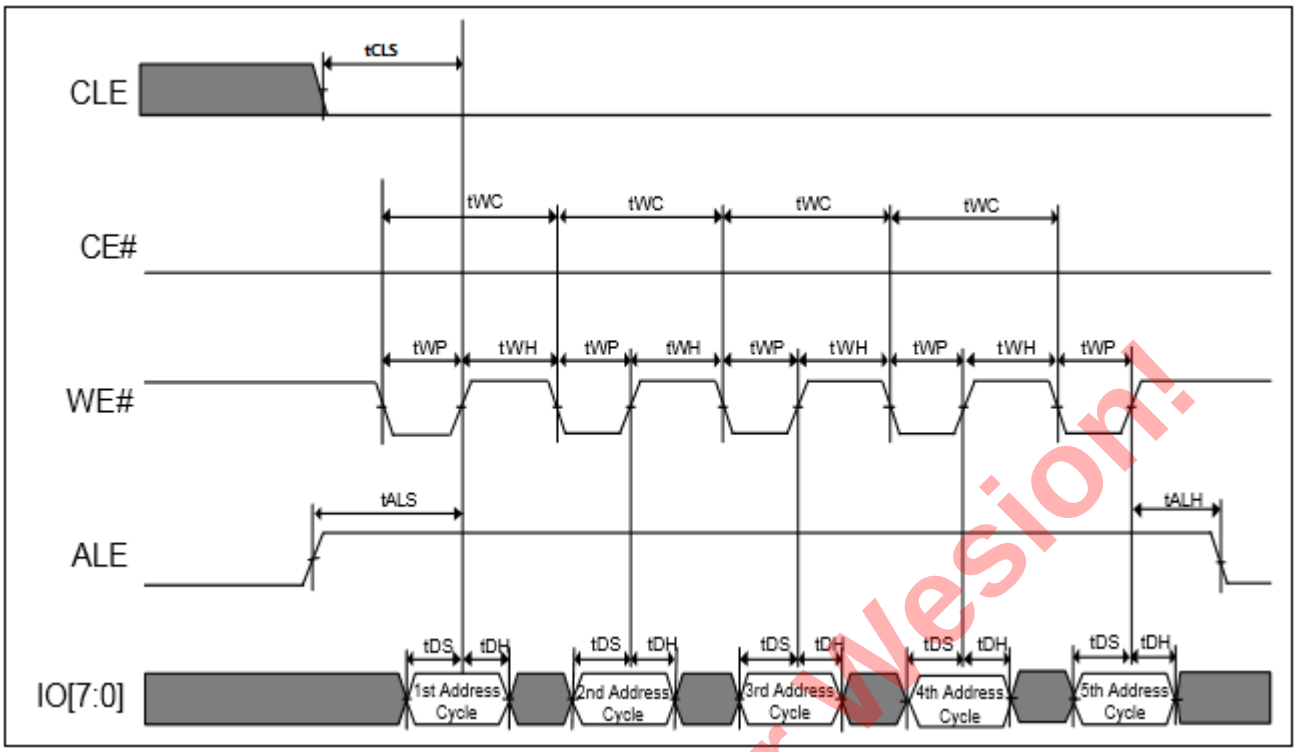


Figure 4-8 Async Waveform for Sequential Data Read Cycle(After Read)-EOD Mode

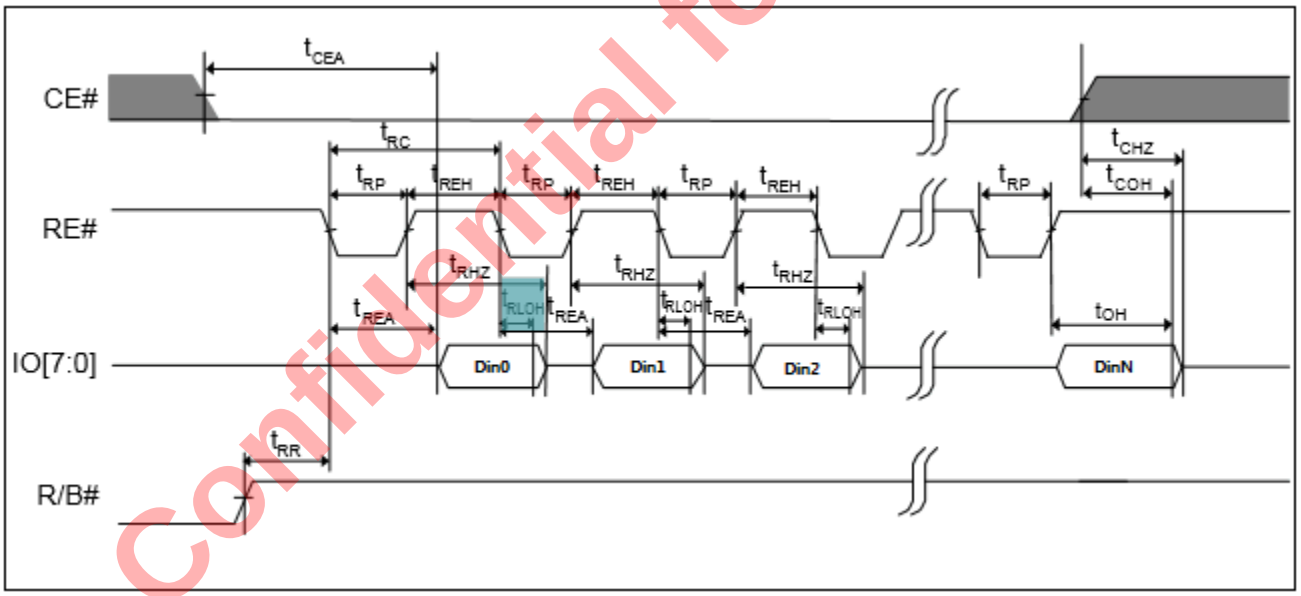


Table 4-7 Nand Timing Specifications

Symbol	Parameter(Asynchronous) (mode 5)	Min	Max	Unit
tCLS	CLE setup time	**10**	**-**	ns
tCLH	CLE hold time	**5**	**-**	ns
tALS	ALE setup	**10**	**-**	ns
tALH	ALE hold	**5**	**-**	ns
tDS	Data setup time	**7**	**-**	ns
tDH	Data hold time	**5**	**-**	ns
tWC	WE# cycle time	**20**	**-**	ns
tWP	WE# pulse width	**10**	**-**	ns
tWH	WE# high lold time	**7**	**-**	ns
tREA	RE# access time	**-**	**16**	ns
tOH	Data output hold time	**15**	**-**	ns
tRLOH	RE#-low to data hold time (EDO)	**5**	**-**	ns
tRP	RE# pulse width	**10**	**-**	ns
tREH	RE# high hold time	**7**	**-**	ns
tRC	RE# cycle time	**20**	**-**	ns

4.7.4 SPICC Timing Specification

Figure 4-9. SPICC Timing Diagram

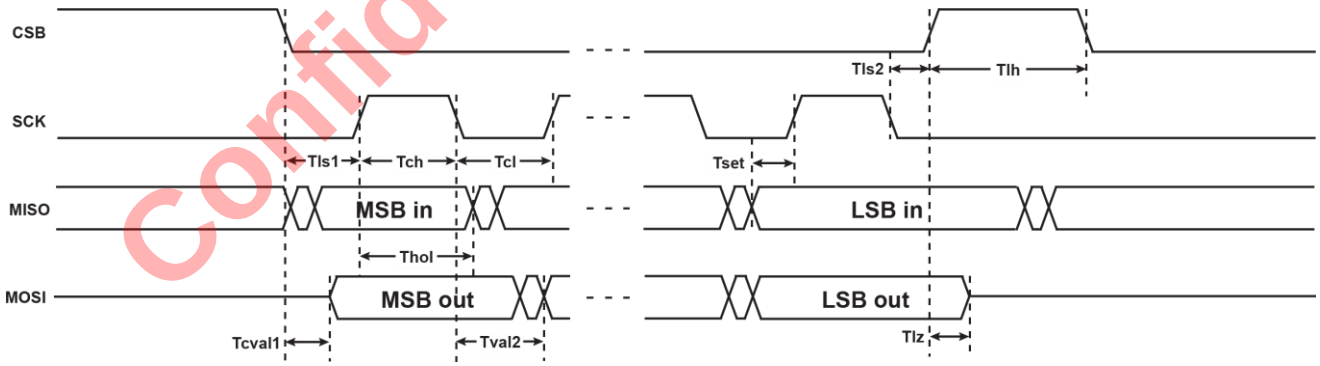


Table 4-8. SPICC Master Timing Specification

Symbol	Description	Min.	Max.	Unit
fCLK	Clock Frequency	1	80	MHz
TCH	Clock high time	5		ns
TCL	Clock low time	5		ns
TLS1	CS fall to First Rising CLK Edge	50		ns
TSET	Data input Setup Time	4		ns
THOL	Data input Hold Time	4		ns
TLH	Minimum idling time between transfers(minimum ss high time)	5		ns

4.7.5 SPIFC Timing Specification

Figure 4-10. SPIFC Serial Input Timing Diagram

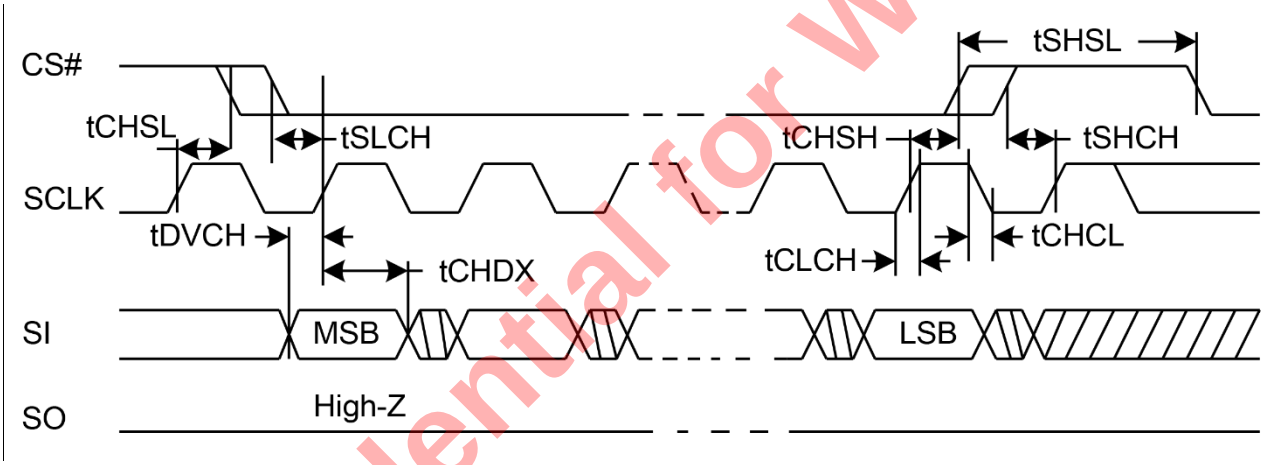


Figure 4-11. SPIFC Out Timing Diagram

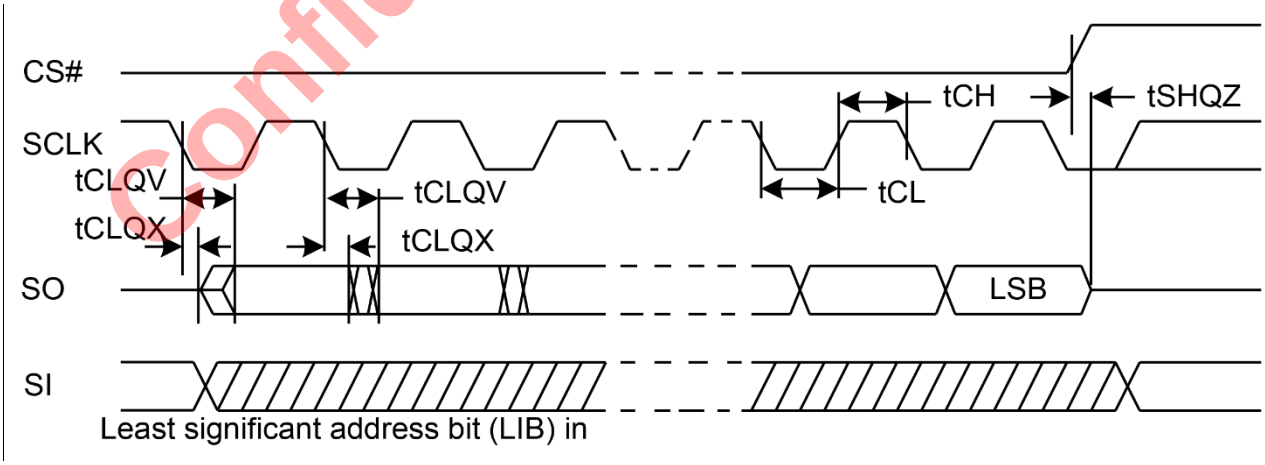


Table 4-9. SPIFC Master Timing Specification

Symbol	Parameter(Clock 41.7MHz)	Min	Max	Unit
fRSCLK	Clock Frequency for READ instructions		50	Mhz
tCH	Clock High Time	8		ns
tCL	Clock Low Time	8		ns
tCLCH	Clock Rise Time (peak to peak)	0.1		V/ns
tCHCL	Clock Fall Time (peak to peak)	0.1		V/ns
tSLCH	CS# Active Setup Time (relative to SCLK)	4	-	ns
tCHSH	CS# Active Hold Time (relative to SCLK)	4	-	ns
tDVCH	Data In Setup Time	2	-	ns
tCHDX	Data In Hold Time	3	-	ns
tSHQZ	Output Disable Time (relative to CS#)		8	ns
tCLQV	Clock Low to Output Valid		6	ns
tCLQX	Output Hold Time	1		ns

4.7.6 Ethernet Timing Specification

Figure 4-12. Management Data Timing Diagram

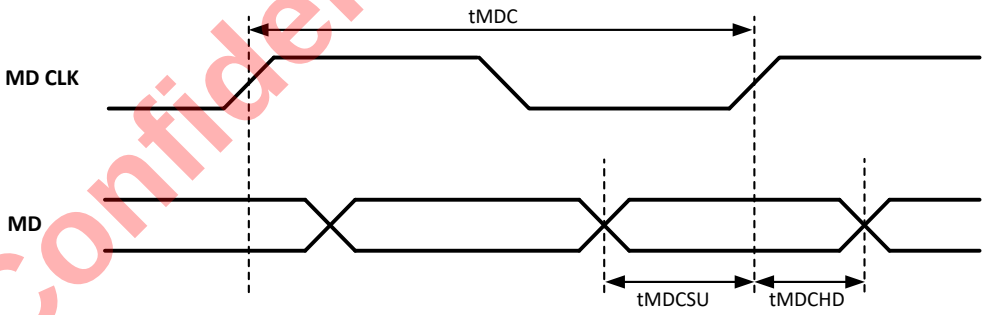


Table 4-10. Management Data Timing Specification

Symbol	Description	Min.	Typ.	Max.	Unit	Notes
tMDC	MDC clock Period	400	500		ns	From MAC
tMDCSU	Setup time to rising edge of MDC	10			ns	
tMDCHD	Hold time to rising edge of MDC	10			ns	

Figure 4-13. RMII Timing Diagram

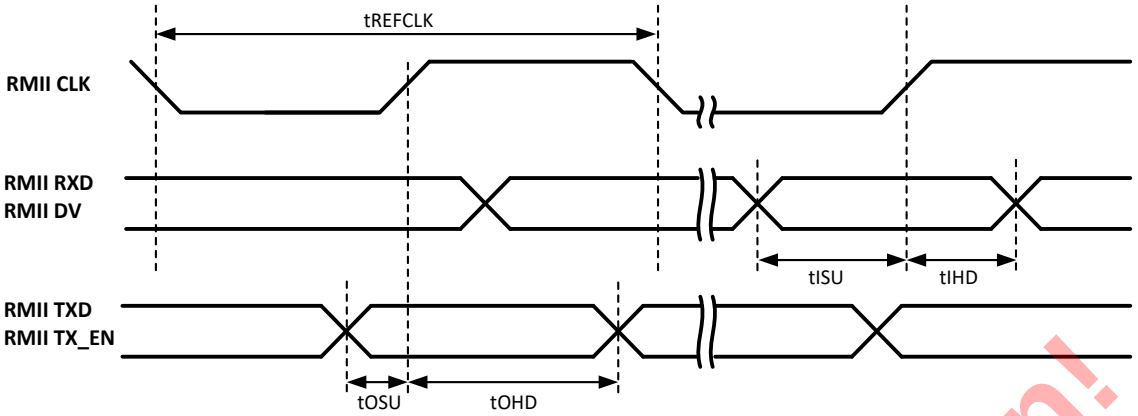


Table 4-11. RMII Timing Specification

Symbol	Description	Min.	Typ.	Max	Unit	Notes
tREFCLK	RMII clock period		20		ns	50MHz from PHY
tOSU	TXD & TX_EN setup time to rising edge of RMII clock	1.8	10		ns	To PHY
tOHD	TXD & TX_EN hold time to rising edge of RMII clock	1.4	10		ns	To PHY
tISU	RXD & DV setup time to rising edge of RMII clock	1.0	10		ns	From PHY
tIHD	RXD & DV hold time to rising edge of RMII clock	1.0	10		ns	From PHY

Figure 4-14. RGMII Receive Timing Diagram

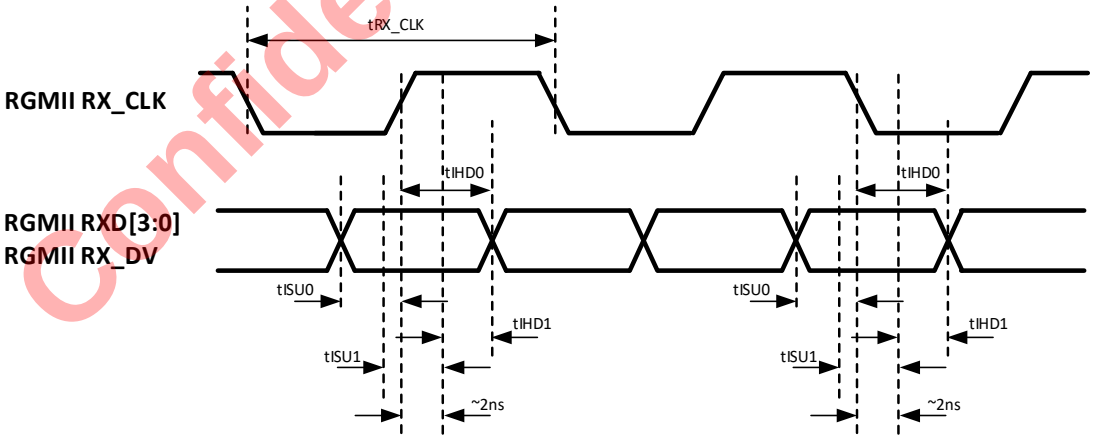


Table 4-12. RGMII Receive Timing Specification

Symbol	Description	Min.	Typ.	Max	Unit	Notes
tRX_CLK	RGMII RX_CLK clock period		8		ns	125MHz from PHY

tSETUP	RXD[3:0] & RX_DV setup time (PHY internal delay enabled)	1.2			ns	From PHY
tHOLD	RXD[3:0] & RX_DV hold time (PHY internal delay enabled)	1.2			ns	From PHY
tSKEW	RXD[3:0] & RX_DV skew between these 5 signals (PHY internal delay disabled)	-0.5		0.5	ns	From PHY

When PHY internal delay is enabled, check setup/hold timing.

When PHY internal delay is disabled, check signal skew.

Figure 4-15. RGMII Transmit Timing Diagram

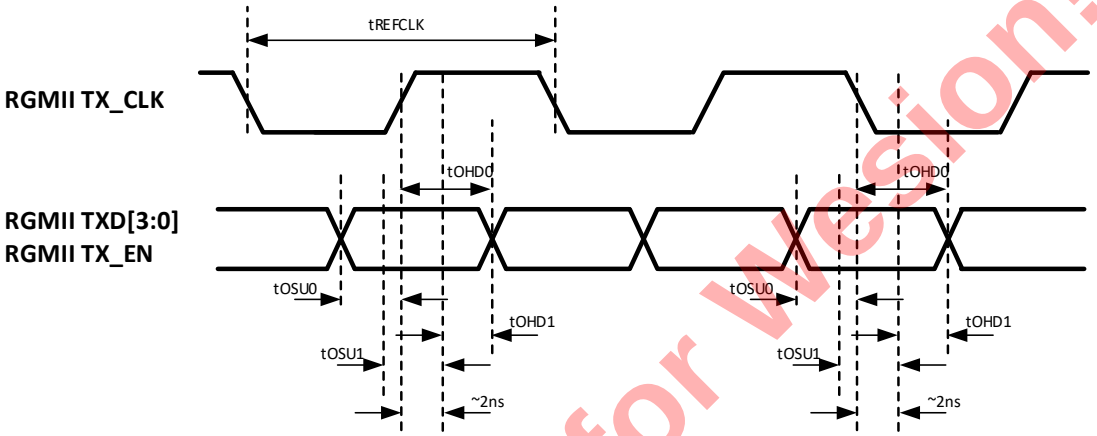


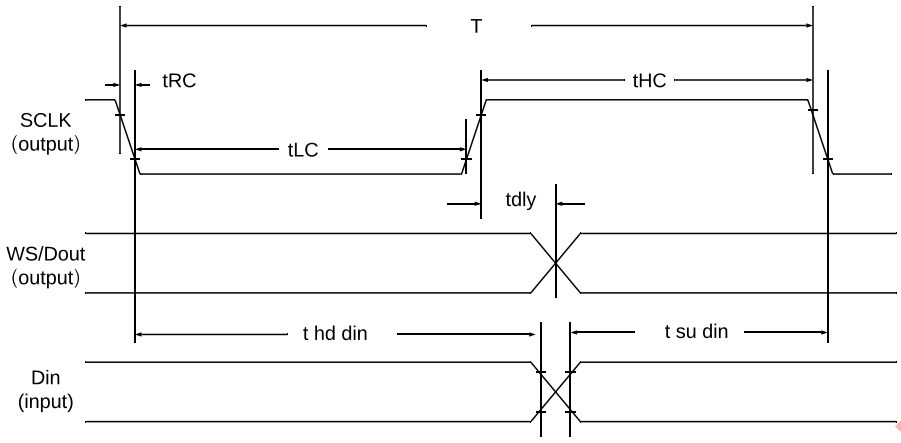
Table 4-13. RGMII Transmit Timing Specification

Symbol	Description	Min.	Typ.	Max	Unit	Notes
tTX_CLK	RGMII TX_CLK clock period		8		ns	125MHz to PHY
tOSU	TXD & TX_EN setup time to rising edge of RGMII clock (no clock delay added)	1			ns	From PHY
	TXD & TX_EN setup time to rising edge of RGMII clock (clock delay added)	-0.9			ns	From PHY
tOHD	RXD & DV hold time to rising edge of RGMII clock (no clock delay added)	0.8			ns	From PHY
	RXD & DV hold time to rising edge of RGMII clock (clock delay added)	2.7			ns	From PHY

4.7.7 Audio Timing Specification

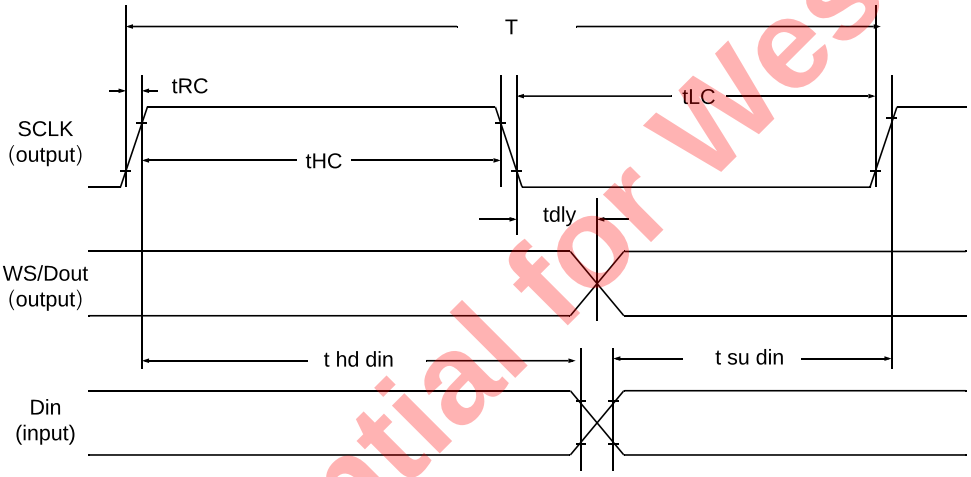
There are two modes for the audio I2S/TDM interface: Master mode and Slave mode, as shown below.

Figure 4-16 Output Data of SCLK Rising Edge TDM/I2S Timing, Master Mode



Note：Input data was sampled on SCLK rising/falling edge via software setting.

Figure 4-17 Output Data of SCLK Falling Edge TDM/I2S Timing, Master Mode



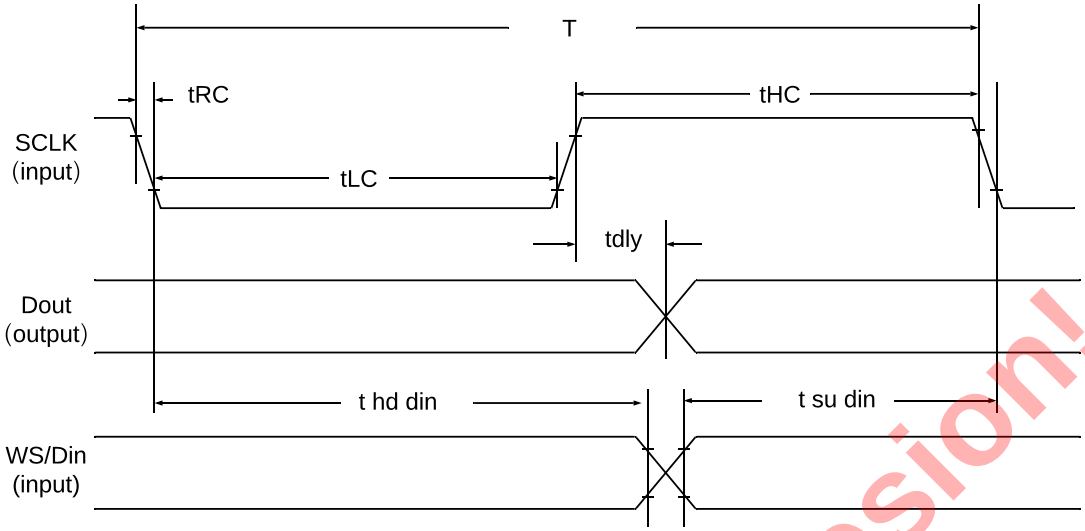
Note：Input data was sampled on SCLK rising/falling edge via software setting.

Table 4-14 Audio I2S/TDM Timing Specification, Transmitter, Master Mode

Transmitter (master mode)					
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**Unit**
T	Clock period	10			ns
tHC	High level of SCLK	0.35			T
tLC	Low level of SCLK	0.35			T
tRC	Edge time of SCLK			0.15	T
tdly	Delay from SCLK to WS	-2	3	5	
Tsu din	Setup time of Din	4			ns
Thd din	Hold time of Din	4			ns

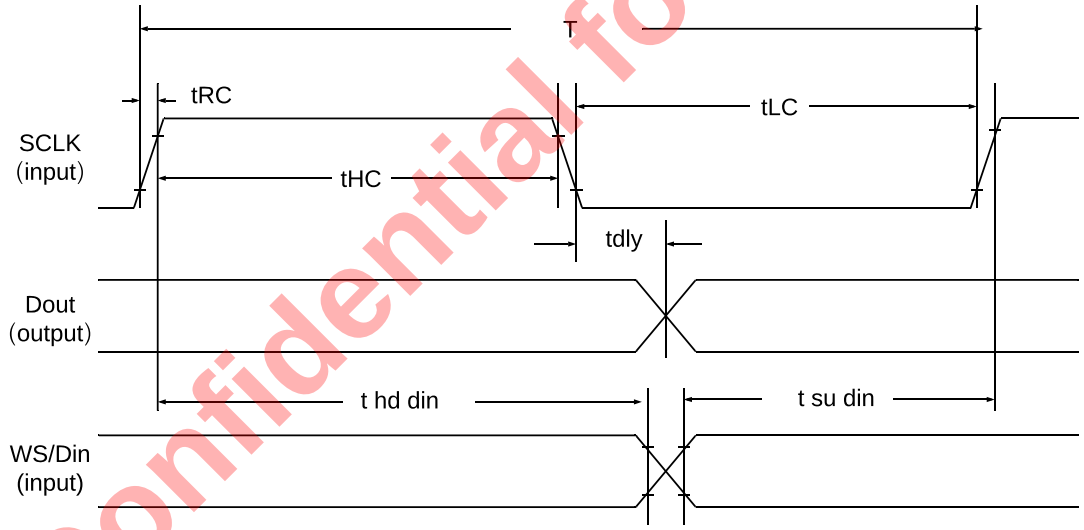
Note: Measure point refer to parameter Vih, ViL of Normal GPIO specifications.

Figure 4-18 Output Data of SCLK Rising Edge TDM/I2S Timing, Slave Mode



Note：Input data was sampled on SCLK rising/falling edge via software setting.

Figure 4-19 Output Data of SCLK Falling Edge TDM/I2S Timing, Slave Mode



Note：Input data was sampled on SCLK rising/falling edge via software setting.

Table 4-15 Audio I2S/TDM Timing Specification, Transmitter, Slave Mode

Transmitter (slave mode)					
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**unit**
T(out)	Clock period	40			ns
T(in)	Clock period	10			ns
tHC	High level of SCLK	0.35			T

Transmitter (slave mode)					
Symbol	**Parameter**	**Min**	**Typ**	**Max**	**unit**
tLC	Low level of SCLK	0.35			T
tRC	Edge time of SCLK			0.15	T
tsu din	Setup time of WS/Din	4			ns
thd din	Hold time of WS/Din	4			ns
tdly	Delay between SCLK and Dout	2	12	15	ns

Note: Measure point refer to parameter Vih, ViL of Normal GPIO specifications.

4.7.8 PDM Timing Specification

Figure 4-20 PDM Timing Diagram

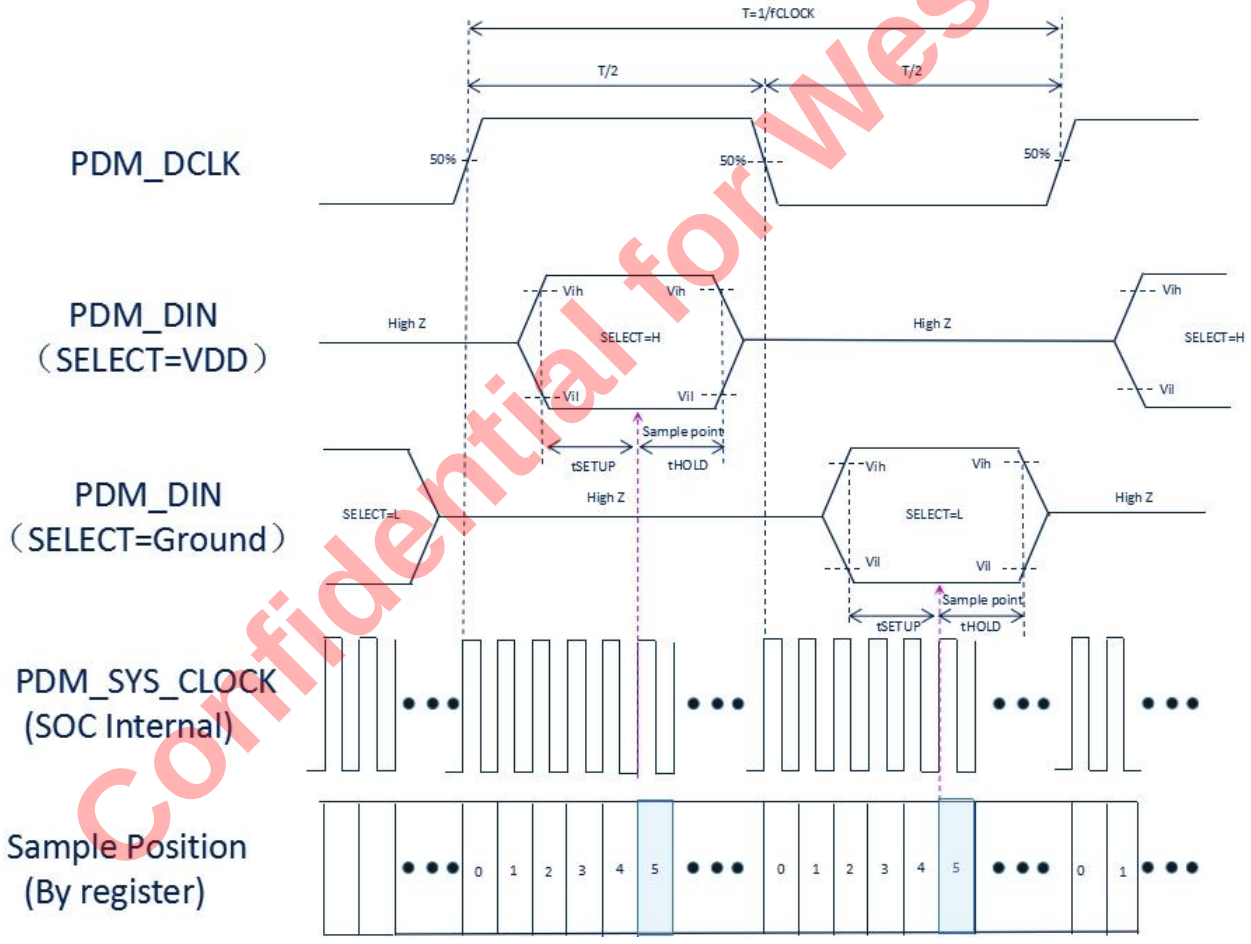


Table 4-16 PDM Timing Specification

Parameter	Symbol	Min.	Typ.	Max.	Units.
PDM clock period	tDCLK	200			ns
PDM clock duty cycle	tHIGH/tLOW	48%		52%	tDCLK

PDM Data setup time	tSETUP	20			ns
PDM Data hold time	tHOLD	20			ns
Sys clock period	tSYSCLK	5	7.5		ns

Note:

1.Default PDM_SYS_CLOCK=133MHz

2.For Sample position，please refer to PDM register PDM_CHAN_CTRL,PDM_CHAN_CTRL1

4.7.9 UART Timing Specification

Figure 4-21 Figure UART Timing Diagram

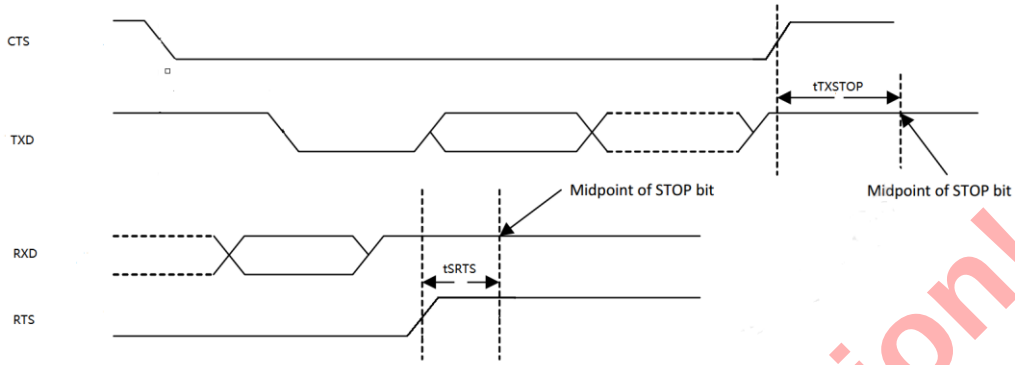


Table 4-17 UART Timing Specification

Parameter	Symbole	Min.	Max.	Units.
Delay time ,CTS high before midpoint of stop bit	tTXSTOP	-	0.5	Bit Periods
Delay time ,midpoint of stop bit to RTS hgih	tSRTS	-	0.5	Bit Periods

4.8 Power Consumption

Note：Value listed here is estimated typical max value tested. Enough margin in circuit needs to be reserved.

Symbol	Maximum Current	Note
VDDCPU_A	6 A	-
VDDCPU_B	1.2A	
VDD_EE	4 A	-
VDD_DDR	400 mA	
VDDQ	600 mA	Note: Peak SOC + DRAM VDDQ current is up to 1.5A with 2 ranks DDR3.

Symbol	Typical current	Maximum current	Note
VDD18_AO_XTAL	1 mA	-	EFUSE: Max 100 mA when programing EFUSE.
VDDAO_0V8	22 mA	-	-
AVDD0V8_USB_PCIE	58 mA	-	
AVDD0V8_HDMI	23 mA	-	At 6 Gbps mode
AVDD_DDRPLL	6 mA	-	
AVDD18_ENET	40 mA	-	-
AVDD18_AUDIO	4 mA	6.6 mA	-
AVDD18_PCIE	45mA	-	At 5 Gbps mode
AVDD18_HDMI	3.3 mA	-	-
AVDD18_SARADC	1.2 mA		
AVDD18_CVBS		48 mA	
AVDD18_MIPIDSI	40 mA		
AVDD18_USB	13.8 mA	17 mA	Per channel
AVDD33_USB	7 mA	-	Per channel
AVDD18_DPLL	35 mA	--	
VDDIO	-	-	Note

Note:

VDDIO=1.8V, DS=3, output 200MHz clock:

1) IO pad itself consumes about 1.4mA.

2) Driving a 55ohm trace with length of 50mm and width of 0.1mm will consumes about 2.8mA additional current (low impedance trace consumes more power)
3) Base #2, add 5pF cap will consumes about 1.8mA additional current, total about 6mA
4) When VDDIO=3.3V, GPIO consumes about 70% higher current, about 13mA
5) Internal & external pull down resistor consumes additional current

4.9 Storage and Baking Conditions

The processor is moisture-sensitive device of MSL level 3, defined by IPC/JEDEC J-STD-020. Please follow the storage and backing guidelines.

1) Calculated shelf life in sealed bag: 12 months at <40°C and <90% relative humidity (RH).
2) After bag is opened, devices that will be subjected to reflow solder or other high temperature process must be
   a) Mounted with 168 hours of factory conditions ≤30°C/60% RH, or
   b) Stored per J-STD-033
3) Devices require bake, before mounting, if Humidity Indicator Card reads >10%.
4) If baking is required, refer IPC/JEDEC J-STB-033 for baking process.

5. Power On Config

3 Boot pins are used as power on config (POC) pins, to set the booting sequence.

POC setting is latched at the rising edge of reset signal.

3 POC pins are all pull high internal, CPU will try to boot from nand/eMMC first, if fails than try to boot from SD CARD, still fails then try to boot from USB (PC).

External 4.7K ohm pull down resistors can be used to change the POC setting. The resistors should be placed on right location, avoid stubs on high speed signals.

S922X's Power On Configuration is listed as following:

Table 5-1. Power On Configuration Pin Table

POC	Boot Pin	Name	Pull low	Pull high
POC0	Boot [4]	SPI NAND First	SPI NAND boot first	Default sequence
POC1	Boot [5]	USB First	USB boot first	Default sequence
POC2	Boot [6]	SPI NOR First	SPI NOR first	Default sequence

Booting Sequence Diagram

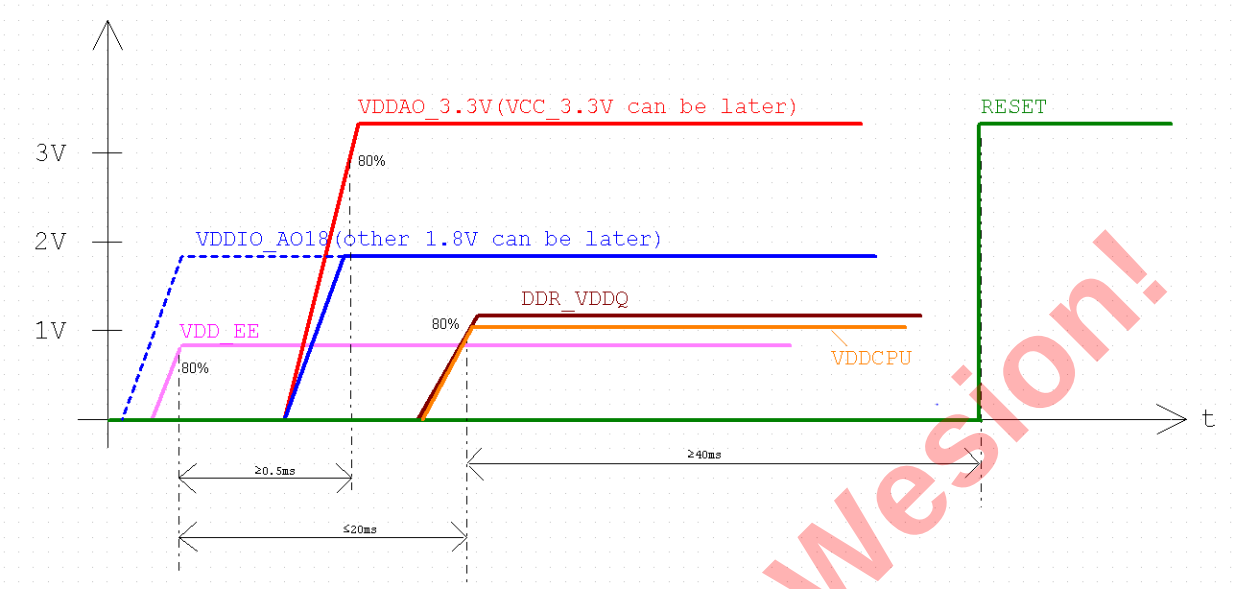
No.	POC_0 (SPI_NAND)	POC_1 (USB_BOOT)	POC_2 (SPI_NOR, eMMC/NAND)	1st Boot device	2nd Boot device	3rd Boot device	4th Boot device
1	0	0	0	USB (short delay)	SPI_NOR	NAND/eMMC	SD Card
2	0	0	1	USB (short delay)	NAND/eMMC	SD Card	-
3	0	1	0	SPI_NOR	NAND/eMMC	SD Card	USB
4	0	1	1	SPI_NAND	NAND/eMMC	USB	-
5	1	0	0	USB (short delay)	SPI_NOR	NAND/eMMC	SD Card
6	1	0	1	USB (short delay)	NAND/eMMC	SD Card	-
7	1	1	0	SPI_NOR	NAND/eMMC	SD Card	USB
8	1	1	1	NAND/eMMC	SD Card	USB	-

Note:

If GPIOC is not work as SDIO port, please do not pull CARD_DET(GPIOC_6) low when system booting up, to avoid romcode trying to boot from SD CARD.

6. Recommended Power on/down Sequence

Example power on sequence:

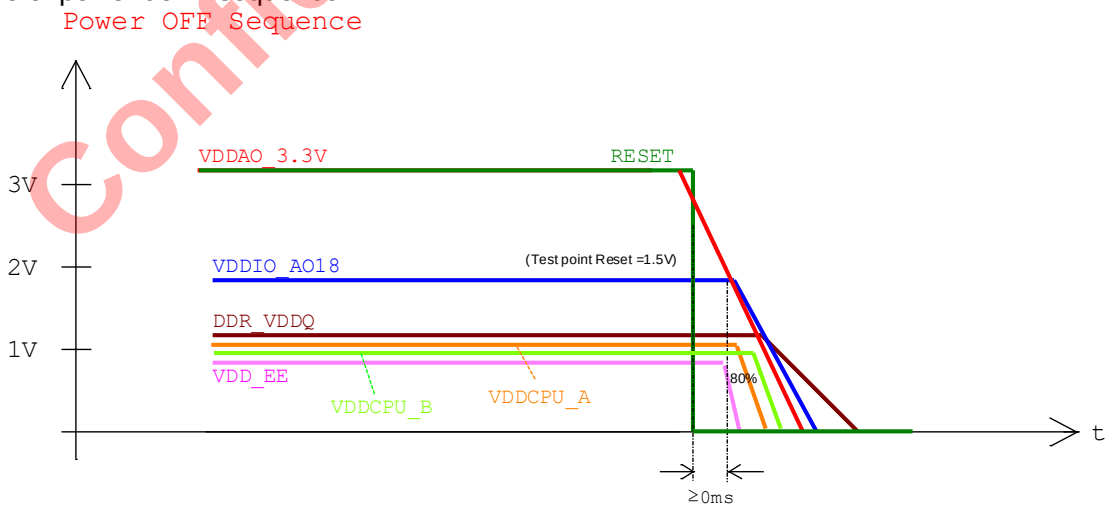


Note:

1) All test values refer to 80% of typical power voltage.
2) VDDAO_3.3V & VCC3.3V should ramp up > 0.5ms later than VDD_EE.
3) All power sources should get stable within 20ms (except for DDR_VDDQ).
4) No sequence requirement between VDD18_AO and VDD_EE. No sequence requirement between VDDCPU_A & VDDCPU_B & DDR_VDDQ and other power source.
5) VDDIO_AO18 should ramps up earlier or at the same time with VDDAO_3.3V & VCC3.3V, VDDAO_3.3V & VCC3.3V should never be 2.5V higher than VDD18_AO.
6) RESET_n should keep low for at least 40ms after power up (except DDR_VDDQ).

Please refer to reference schematics.

Example of power down sequence:



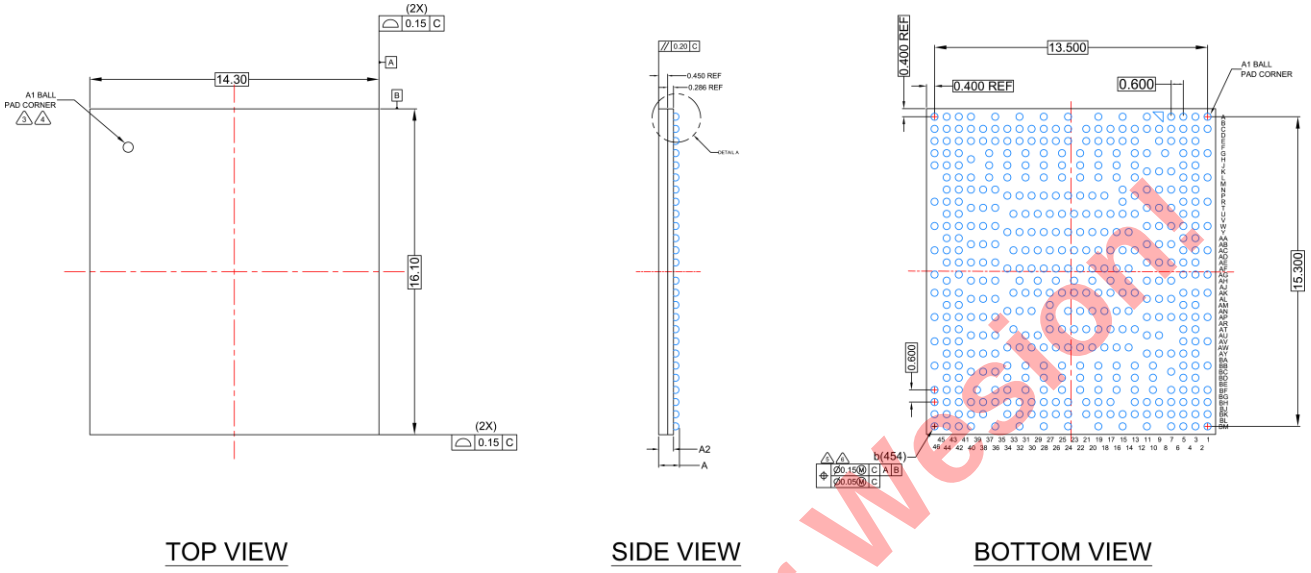
Note:

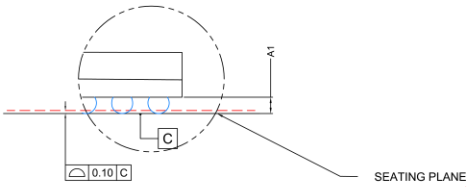
A reset IC should monitor VDDIO3.3V, output lower (lower than 1.5V) before other power rails turning off. There is no power off sequence between other power rails.

7. Mechanical Dimensions

The S922X processor comes in a 52x46 ball matrix FCBGA RoHS package. The mechanical dimensions are given in millimeters as the following figures.

Figure 7-1 Dimensions





DETAIL A
ROTATED 90°

DIMENSION	MINIMUM	NOMINAL	MAXIMUM
A	0.926	1.006	1.086
A1	0.230	0.270	0.310
A2	0.686	0.736	0.786
b	0.300	0.350	0.400
NUMBER OF BALL 454			

8. System

This part describes the S922X system architecture from the following aspects:

- MEMORY MAP
- POWER DOMAIN
- CPU and GPU SUBSYSTEM
- CLOCK AND RESET UNIT
- SYSTEM BOOT
- GENERAL PURPOSE INPUT/OUTPUT (GPIO)
- INTERRUPT CONTROLLER
- DIRECT MEMORY ACCESS CONTROLLER (DMAC)
- TIMER
- Crypto

8.1 Memory Map

S922X's memory map is listed as following in Table 8-1.

Table 8-1 S922X Memory Map

START	END	REGION (NORMAL)
FFFF0000	FFFFFFFF	A53_ROM
FFFE8000	FFFEFFFF	
FFFA0000	FFFE7FFF	AHB SRAM
FFE80000	FFF9FFFF	
FFE40000	FFE7FFFF	mali
FFE0D000	FFE3FFFF	
FFE0B000	FFE0CFFF	
FFE09000	FFE0AFFF	USBCTRL
FFE07000	FFE08FFF	EMMCC/NAND
FFE05000	FFE06FFF	EMMCB
FFE03000	FFE04FFF	EMMCA
FFE02000	FFE02FFF	BT656
FFE01000	FFE01FFF	HTX_HDCP22
FFE00000	FFE00FFF	
FFD26000	FFDFFFFF	
FFD25000	FFD25FFF	SC
FFD24000	FFD24FFF	EE_UART_0
FFD23000	FFD23FFF	EE_UART_1
FFD22000	FFD22FFF	EE_UART_2
FFD21000	FFD21FFF	
FFD20000	FFD20FFF	
FFD1F000	FFD1FFFF	EE_I2C_M0
FFD1E000	FFD1EFFF	EE_I2C_M1
FFD1D000	FFD1DFFF	EE_I2C_M2
FFD1C000	FFD1CFFF	EE_I2C_M3
FFD1B000	FFD1BFFF	PWM_AB

START	END	REGION (NORMAL)
FFD1A000	FFD1AFFF	PWM_CD
FFD19000	FFD19FFF	PWM_EF
FFD18000	FFD18FFF	MSR_CLK
FFD17000	FFD17FFF	
FFD16000	FFD16FFF	
FFD15000	FFD15FFF	SPICC_1
FFD14000	FFD14FFF	SPIFC
FFD13000	FFD13FFF	SPICC_0
FFD12000	FFD12FFF	
FFD11000	FFD11FFF	
FFD10000	FFD10FFF	
FFD0F000	FFD0FFFF	ISA
FFD0E000	FFD0EFFF	PARSER
FFD0D000	FFD0DFFF	
FFD0C000	FFD0CFFF	SANA
FFD0B000	FFD0BFFF	STREAM
FFD0A000	FFD0AFFF	ASYNC_FIFO
FFD09000	FFD09FFF	ASYNC_FIFO2
FFD08000	FFD08FFF	ASSIST
FFD07000	FFD07FFF	MIPI_DSI_HOST
FFD06000	FFD06FFF	STB
FFD05000	FFD05FFF	AIFIFO
FFD04000	FFD04FFF	
FFD03000	FFD03FFF	
FFD02000	FFD02FFF	
FFD01000	FFD01FFF	RESET
FFD00000	FFD00FFF	
FFC08000	FFCFFFFF	
FFC00000	FFC07FFF	GIC

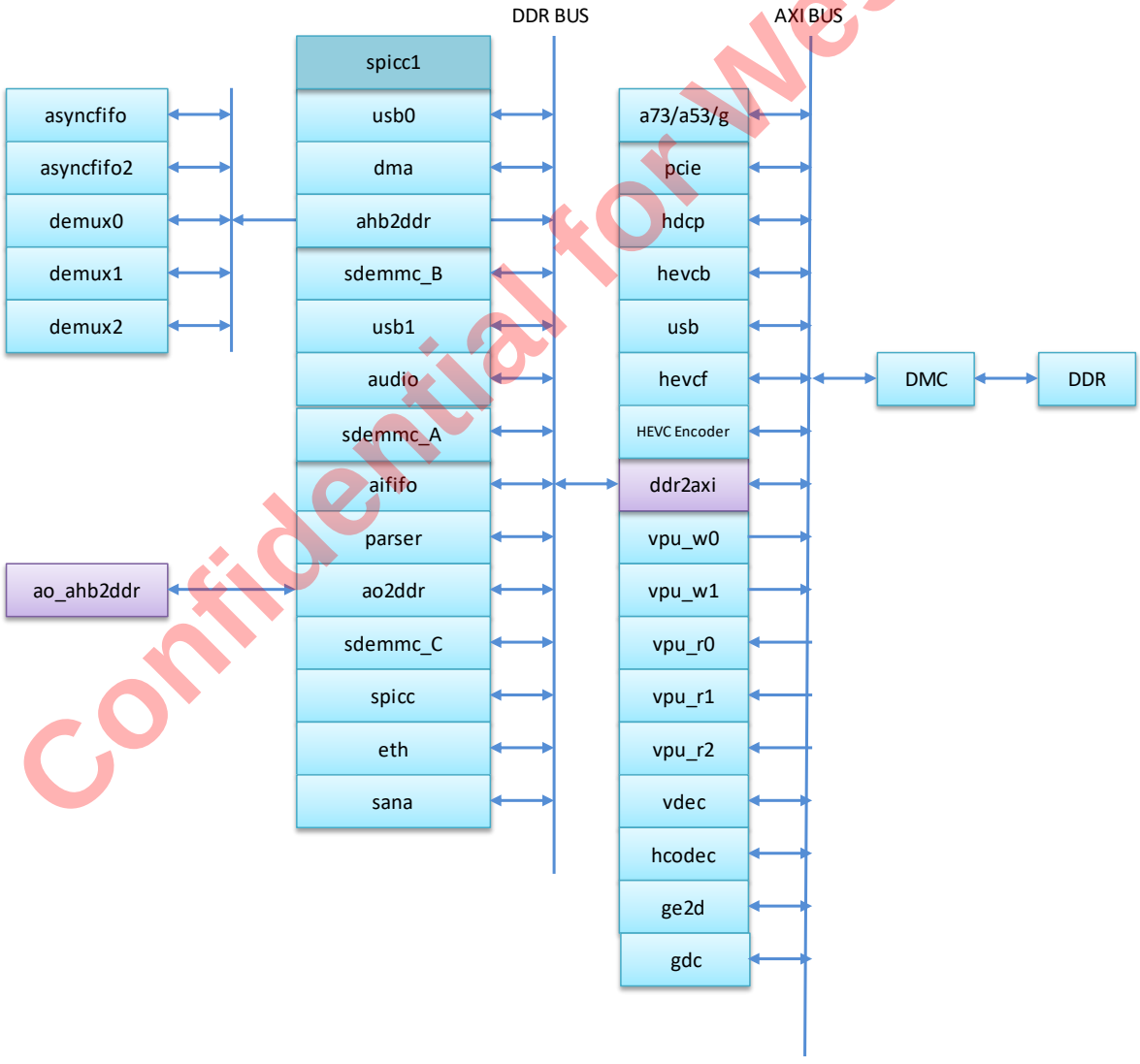
START	END	REGION (NORMAL)
FFB00000	FFBFFFFF	GPV
FF950000	FFAFFFFF	
FF940000	FF94FFFF	GE2D
FF900000	FF93FFFF	VPU
FF80B000	FF8FFFFF	
FF80A000	FF80AFFF	AO_MAILBOX
FF809000	FF809FFF	SAR_ADC
FF808000	FF808FFF	IR_DEC
FF807000	FF807FFF	PWM_AB
FF806000	FF806FFF	AO_I2C_S0
FF805000	FF805FFF	AO_I2C_M0
FF804000	FF804FFF	AO_UART2
FF803000	FF803FFF	AO_UART
FF802000	FF802FFF	PWM_CD
FF801000	FF801FFF	
FF800000	FF800FFF	RTI
FF660000	FF7FFFFF	
FF65E000	FF65FFFF	
FF65C000	FF65DFFF	
FF65A000	FF65BFFF	
FF658000	FF659FFF	
FF656000	FF657FFF	EQ_DRC
FF654000	FF655FFF	MIPI_adapter
FF652000	FF653FFF	
FF650000	FF651FFF	
FF64E000	FF64FFFF	RESET_SEC
FF64C000	FF64DFFF	ETH_PHY
FF64A000	FF64BFFF	AUDIO_LOCKER
FF648000	FF649FFF	PCIE_A

START	END	REGION (NORMAL)
FF646000	FF647FFF	PCIE_PHY
FF644000	FF645FFF	MIPI_DSI_PHY
FF642000	FF643FFF	AUDIO
FF640000	FF641FFF	PDM
FF63E000	FF63FFFF	DMA
FF63C000	FF63DFFF	HIU
FF63A000	FF63BFFF	USBPHY21
FF638000	FF639FFF	DMC
FF636000	FF637FFF	USBPHY20
FF635000	FF635FFF	
FF634C00	FF634FFF	TS_DDR
FF634800	FF634BFF	TS_PLL
FF634400	FF6347FF	PERIPHS_REG
FF634000	FF6343FF	
FF632000	FF633FFF	ACODEC
FF630000	FF631FFF	EFUSE
FF620000	FF62FFFF	DOS
FF610000	FF61FFFF	WAVEL
FF600000	FF60FFFF	HDMITX
FF500000	FF5FFFFF	USB0
FF400000	FF4FFFFF	USB1
FF3F0000	FF3FFFFF	ETH
FF3E0000	FF3EFFFF	CCI
FF180000	FF3DFFFF	
FF000000	FF3EFFFF	
FF140000	FF17FFFF	
FF100000	FF130000	
FF000000	FF0FFFFF	AXI_SRAM
FE000000	FEFFFFFF	DDR_CTRL

START	END	REGION (NORMAL)
FC000000	FDFFFFFF	PCIE_AXI
FA000000	FBFFFFFF	
F6000000	F9FFFFFF	FLASH
F5800000	F5FFFFFF	A53_DBG
F5000000	F57FFFFF	A73_DBG
0	F4FFFFFF	DDR

Below is the DDR bus.

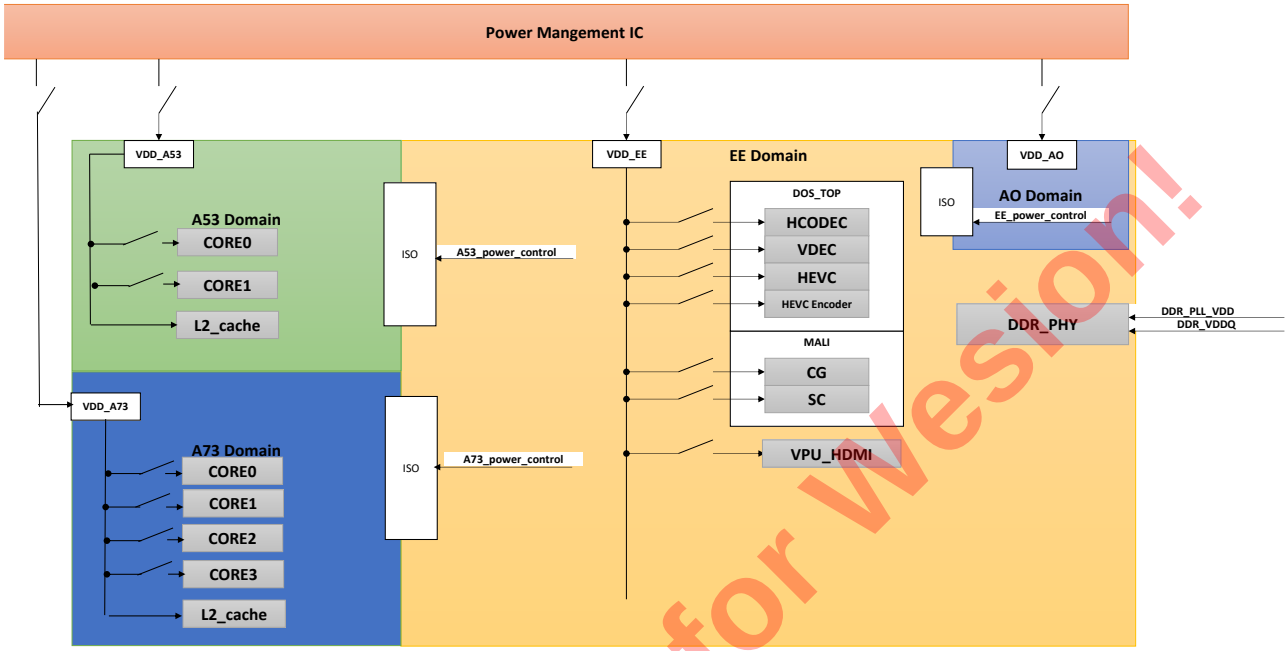
Figure 8-1 S922X DDR Bus



8.2 Power Domain

The following figure shows the power domain of S922X.

Figure 8-2 Power Domain



8.2.1 Top Level Power Domains

The power supplies for the different domains must follow a specific power supply order: The A53/A73 can't be powered without the EE domain. The EE domain can't be powered up without the AO domain. If you read Table 8-2 left to right then right to left, that's essentially the power up/down sequence for the entire chip.

Table 8-2 Power on Sequence of Different Power Domains

	Always On	**EE Domain**		**A73 Domain**				**A53 Domain**		
	Logic	EE Logic	Mali and DOS and VPU	L2 Cache	CPU0	CPU1	CPU2/3	L2 Cache	CPU0	CPU1
STATE 0 All off	Off	OFF		The A73 domain must be off if the EE domain is OFF				OFF	OFF	OFF
STATE 1 AO ON only	ON	Off	OFF	The A73 domain must be off if the EE domain is OFF				OFF	OFF	OFF
STATE 2 A53/A73 OFF only	On	On	ON/OFF	Off	Off	Off	Off	OFF	OFF	OFF
STATE 3 L2C ON only	On	On	ON/OFF	ON	Off		Off	ON	OFF	OFF
STATE 4 SMP - example	On	On	ON/OFF	ON	On		Off	ON	ON	OFF
STATE 5 SMP - example	On	On	ON/OFF	ON	On		On	ON	ON	ON

8.2.2 A53/A73 Big Little Power Modes

The A53 and a73 domain is the last to power up and the first to power down. The A73 domain itself consists of a quad (4) CPU, an L2 cache controller and an SCU. The A53 domain itself consists of a dual (2) CPU, an L2 cache controller and an SCU. The A53 CPU boots with the SCU/L2 powered and CPU0 powered. After CPU0 boots, subsequent CPU's can be enabled and disabled independently of one another using the control bits described below. A73 domain is default shut down. Before power on CPUs inside A73, should boot up SCU/L2. Then we can power on any cores of a73 domain.The flow-diagram below illustrates the transition to each legal state.

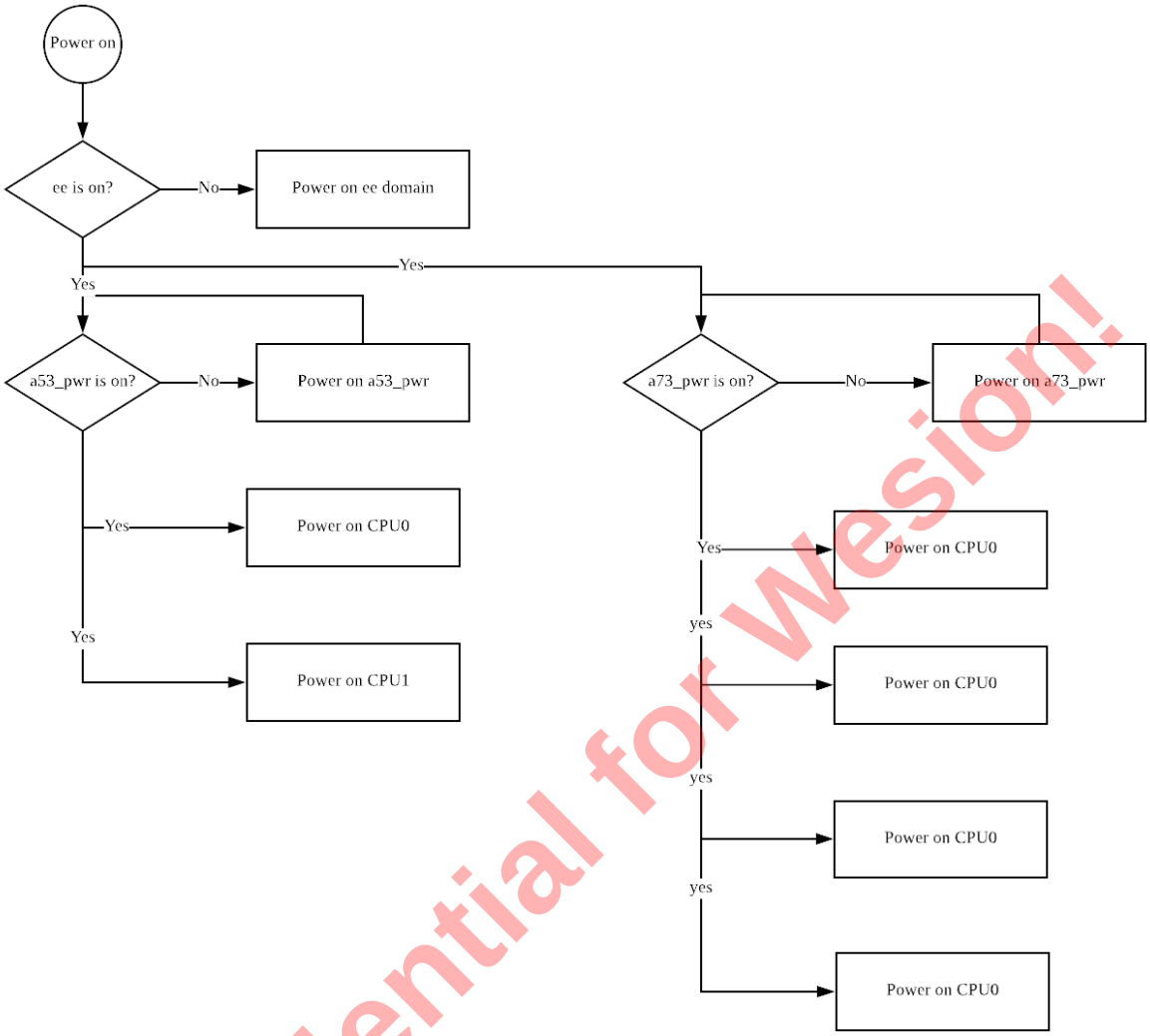
Table 8-3 Power On Sequence of A53

A53	CPU0	CPU1	A53_pwr
Domain Power Sleep bit (1 = power off)	Bit[2] 0xff8000E4	Bit[4] 0xff8000E4	
Domain Power Acknowledge bit	Bit[16] 0xff8000E4	Bit[17] 0xff8000E4	
Input Signal isolation bit (1 = isolated)	Bit[0] 0xff8000E0	Bit[1] 0xff8000E0	Bit[12] 0xff8000E0
Output signal isolation bit (1 = isolated)	Bit[0] 0xff8000E0	Bit[1] 0xff8000E0	Bit[12] 0xff8000E0

Table 8-4 Power On Sequence of A73

A73	CPU0	CPU1	CPU2	CPU3	A73_pwr
Domain Power Sleep bit (1 = power off)	Bit[2] 0xff80034C	Bit[4] 0xff80034C	Bit[6] 0xff80034C	Bit[8] 0xff80034C	
Domain Power Acknowledge bit	Bit[16] 0xff80034C	Bit[17] 0xff80034C	Bit[18] 0xff80034C	Bit[19] 0xff80034C	
Input Signal isolation bit (1 = isolated)	Bit[0] 0xff800348	Bit[1] 0xff800348	Bit[2] 0xff800348	Bit[3] 0xff800348	Bit[12] 0xff800348
Output signal isolation bit (1 = isolated)	Bit[0] 0xff800348	Bit[1] 0xff800348	Bit[2] 0xff800348	Bit[3] 0xff800348	Bit[12] 0xff800348

Figure 8-3 Power Seuquence of A53/A73



8.2.3 EE Top Level Power Modes

EE domain is powered off by DC-DC. VDD_EE also share the same VDD with VDDCE of A53/A73 memory arrays. So if EE domains is shut off, A53/A73 memory is also shut off. That does not matter. Before EE power domain is shut off, A53/A73 should be shut off at first.

Table 8-5 Power Sequence of DOS

DOS	VDEC	HEVC	HCODEC	HEVC Encoder
Domain Power Sleep bit (1 = power off)	Bit[3:2] 0XFF8000E8	Bit[7:6] 0XFF8000E8	Bit[1:0] 0XFF8000E8	Bit[25:24] 0XFF8000E8
Domain Power Acknowledge bit	Bit[1] 0XFF8000F0	Bit[3] 0XFF8000F0	Bit[0] 0XFF8000F0	Bit[12] 0XFF8000F0
Input Signal isolation bit (1 = isolated)	Bit[6] 0XFF8000EC	Bit[10] 0XFF8000EC	Bit[4] 0XFF8000EC	Bit[12] 0XFF8000EC
Output signal isolation bit (1 = isolated)	Bit[7] 0XFF8000EC	Bit[11] 0XFF8000EC	Bit[5] 0XFF8000EC	Bit[13] 0XFF8000EC
Memory Power Down/Up 0 = powered on1 = powered off	0xff62fcc0	0xff62fccc	0xff62fcc8	0xff62fce4
Reset_n	0xff62fc00	0xff62fcd0	0xff62fc1c	0xff62fcdc

Table 8-6 Power Sequence of VPU

VPU	VDEC
Domain Power Sleep bit (1 = power off)	Bit[8] 0XFF8000E8
Domain Power Acknowledge bit	Bit[9:8] 0XFF8000F0
Input Signal isolation bit (1 = isolated)	Bit[9] 0XFF8000E8
Memory Power Down/Up 0x00000000 = powered on 0xFFFFFFFF= powered off	Bit[31:0] 0XFF63C104 Bit[31:0] 0XFF63C108 Bit[31:0] 0XFF63C134 Bit[15:8] 0XFF63C100

8.2.4 Mali Power Modes

The G52 block sits within the EE domain and the G52 module itself has 4 distinct power domains:

- GL(Always ON[EE])
- CG(PD0)
- SC0(PD1)

- SC1 (PD2)

Power on sequence: GL→CG→(SC0,SC1)

Power off sequence: (SC0,SC1)→ CG → GL

8.2.5 Power/Isolation/Memory Power Down Register Summary

Below lists the registers related to power domain.

Each register final address = 0xFF800000 + offset * 4

AO_RTI_PWR_CNTL_REG0 0x04

Bit(s)	R/W	Default	Description
31-24	R	0	Unused
17-16	R/W	0	Unused
15	R/W	0	Unused
14	R/W	1	Set to 1 to enable the crystal clock throughout the AO domain. Set to 0 to disable the crystal clock (and therefore save power)
13	R/W	0	Reserved
12-10	R/W	0	ALT_32K_INPUT_SEL: 000 = XTAL generate 32k 001 = gpio pad
9	R/W	1	EE_RESET_N: This bit is OR'd with the RESET_N signal from the pad to generate a new signal called reset_n_to_ee. Reset_n_to_ee is the general reset for the entire Everything Else domain. Therefore, if EE_RESET_N is set low, the entire Everything Else domain is reset.
8	R/W	0	RTC_OSCIN_SEL: Set this bit to 1 to mux the RTC 32khz crystal in place of the clk81 signal that comes from the Everything Else domain to drive the Always On domain.
7	R/W	0	Unused
6	R/W	0	Unused
5	R/W	0	Unused
4	R/W	0	Ee_iso_in_en
3	R/W	0	Ee_iso_out_en
2	R/W	0	Unused
1	R/W	0	Reset_iso_en
0	R/W	0	Unused

AO_RTI_PWR_SYS_CPU_CNTL0 0x38

Bit(s)	R/W	Default	Description
31-30	R	-	A53_pwrctli3 (readback)

Bit(s)	R/W	Default	Description
29-28	R	-	A53_pwrctli2 (readback)
27-26	R	-	A53_pwrctli1 (readback)
25-24	R	-	A53_pwrctli0 (readback)
23-22	R/W	0x3	A53_pwrctli3
21-20	R/W	0x3	A53_pwrctli2
19-18	R/W	0x3	A53_pwrctli1
17-16	R/W	0	A53_pwrctli0
15-0	R/W	0	unused

AO_RTI_PWR_SYS_CPU_CNTL1 0x39

Bit(s)	R/W	Default	Description
31-24	R	0	unused
23-16	R	-	A53 sleep status
15-10	R/W	0	Unused
9-8	R/W	3	CPU3_SLEEP
7-6	R/W	3	CPU2_SLEEP
5-4	R/W	3	CPU1_SLEEP
3-2	R/W	0	CPU0_SLEEP
1	R/W	0	Unused
0	R/W	0	Unused

AO_RTI_GEN_PWR_SLEEP0 0x3a

Bit(s)	R/W	Default	Description
31-8	R/W	0	unused
7	R/W	1	DOS_HEVC_D1_PWR_OFF
6	R/W	1	DOS_HEVC_PWR_OFF
5	R/W	1	DOS_VDEC2_D1_PWR_OFF
4	R/W	1	DOS_VDEC2_PWR_OFF
3	R/W	1	DOS_VDEC1_D1_PWR_OFF
2	R/W	1	DOS_VDEC1_PWR_OFF
1	R/W	1	DOS_HCODEC_D1_PWR_OFF

Bit(s)	R/W	Default	Description
0	R/W	1	DOS_HCODEC_PWR_OFF

AO_RTI_GEN_PWR_ISO0 0x3b

Bit(s)	R/W	Default	Description
31-12	R/W	0	unused
11	R/W	1	DOS_HEVC_OUT_EN
10	R/W	1	DOS_HEVC_IN_EN
9	R/W	1	DOS_VDEC2_ISO_OUT_EN
8	R/W	1	DOS_VDEC2_ISO_IN_EN
7	R/W	1	DOS_VDEC1_ISO_OUT_EN
6	R/W	1	DOS_VDEC1_ISO_IN_EN
5	R/W	1	DOS_HCODEC_ISO_OUT_EN
4	R/W	1	DOS_HCODEC_ISO_IN_EN
3	R/W	1	GPU_ISO_OUT_EN
2	R/W	1	GPU_ISO_IN_EN
1	R/W	1	Unused
0	R/W	1	Unused

AO_RTI_GEN_PWR_ACK0 0x3c

Bit(s)	R/W	Default	Description
31-10	R	0	Reserved
9	R/W	1	VPU_ACK
8	R/W	1	HDMI_ACK
7	R/W	1	MALI_PWRUP_ACK[3]
6	R/W	1	MALI_PWRUP_ACK[3]
5	R/W	1	MALI_PWRUP_ACK[3]
4	R/W	1	MALI_PWRUP_ACK[3]
3	R/W	1	DOS_HEVC
2	R/W	1	DOS_VDEC2
1	R/W	1	DOS_VDEC1
0	R/W	1	DOS_HCODEC

AO_RTI_GEN_PWR_SYS_CPU_MEM_PD0 0x3d

Bit(s)	R/W	Default	Description
31-0	R/W	0xFFF00000	Unused

AO_RTI_GEN_PWR_SYS_CPU_MEM_PD1 0x3e

Bit(s)	R/W	Default	Description
31~8	-	0	0
7~0	R/W	0x03f	Reserved

AO_RTI_PWR_SYS_CPUB_CNTL0 0xd2

Bit(s)	R/W	Default	Description
31-30	R	-	A73_pwrctli3 (readback)
29-28	R	-	A73_pwrctli2 (readback)
27-26	R	-	A73_pwrctli1 (readback)
25-24	R	-	A73_pwrctli0 (readback)
23-22	R/W	0x3	A73_pwrctli3
21-20	R/W	0x3	A73_pwrctli2
19-18	R/W	0x3	A73_pwrctli1
17-16	R/W	0x3	A73_pwrctli0
15-0	R/W	0	unused

AO_RTI_PWR_SYS_CPUB_CNTL1 0xd3

Bit(s)	R/W	Default	Description
31-24	R	0	unused
23-16	R	-	A73 sleep status
15-10	R/W	0	Unused
9-8	R/W	3	CPU3_SLEEP
7-6	R/W	3	CPU2_SLEEP
5-4	R/W	3	CPU1_SLEEP
3-2	R/W	3	CPU0_SLEEP
1	R/W	0	Unused
0	R/W	0	Unused

AO_RTI_GEN_PWR_SYS_CPUB_MEM_PD0 0xd4

Bit(s)	R/W	Default	Description
31-0	R/W	0xFFF3FFFF	Unused

AO_RTI_GEN_PWR_SYS_CPUB_MEM_PD1 0xd5

Bit(s)	R/W	Default	Description
31-8	R	0	0
7-0	R/W	0x01f	Unused

Below are DOS power domain related registers.

DOS_MEM_PD_VDEC 0xff62fcc0

Bit(s)	R/W	Default	Description
17-16	R/W	0x3	vdec/pre_arb
15-14	R/W	0x3	vdec/pic_dc
13-12	R/W	0x3	vdec/pscale
11-10	R/W	0x3	vdec/mcrcc
9-8	R/W	0x3	vdec/dblk
7-6	R/W	0x3	vdec/mc
5-4	R/W	0x3	vdec/iqidct
3-2	R/W	0x3	vdec/vld
1-0	R/W	0x3	vdec/vcpu

DOS_MEM_PD_HCODEC 0xff62fcc8

Bit(s)	R/W	Default	Description
17-16	R/W	0x3	hcodec/pre_arb
15-14	R/W	0x3	hcodec/pic_dc
13-12	R/W	0x3	hcodec/mfdin
11-10	R/W	0x3	hcodec/mcrcc
9-8	R/W	0x3	hcodec/dblk
7-6	R/W	0x3	hcodec/mc
5-4	R/W	0x3	hcodec/qdct
3-2	R/W	0x3	hcodec/vlc
1-0	R/W	0x3	hcodec/vcpu

DOS_MEM_PD_HEVC 0xff62fccc

Bit(s)	R/W	Default	Description
15-14	R/W	0x3	hevc/ddr
13-12	R/W	0x3	hevc/sao
11-10	R/W	0x3	hevc/dblk
9-8	R/W	0x3	hevc/ipp
7-6	R/W	0x3	hevc/mpred
5-4	R/W	0x3	hevc/iqit
3-2	R/W	0x3	hevc/parser
1-0	R/W	0x3	hevc/vcpu

DOS_MEM_PD_HEVC Encoder 0xff62fce4

Bit(s)	R/W	Default	Description
31-2	R/W	0x3fffffff	HEVC Encoder/vcore
1-0	R/W	0x3	HEVC Encoder /vcpu

DOS_SW_RESET0 0xff62fc00

Bit(s)	R/W	Default	Description
14	R/W	0	vdec/afifo
13	R/W	0	vdec/ddr
12	R/W	0	vdec/ccpu
11	R/W	0	vdec/mcpu
10	R/W	0	vdec/psc
9	R/W	0	vdec/pic_dc (cmpif_bridge if compress mode)
8	R/W	0	vdec/dblk
7	R/W	0	vdec/mc
6	R/W	0	vdec/iqidct
5	R/W	0	vdec/vififo
4	R/W	0	vdec/vld_part
3	R/W	0	vdec/vld
2	R/W	0	vdec/assist
1	R/W	0	vdec/dos_reg internal

DOS_SW_RESET1 0xff62fc1c

Bit(s)	R/W	Default	Description
17	R/W	0	hcodec/qdct
16	R/W	0	hcodec/vlc
14	R/W	0	hcodec/afifo
13	R/W	0	hcodec/ddr
12	R/W	0	hcodec/ccpu
11	R/W	0	hcodec/mcpu
10	R/W	0	hcodec/psc
9	R/W	0	hcodec/pic_dc
8	R/W	0	hcodec/dblk
7	R/W	0	hcodec/mc
6	R/W	0	hcodec/iqidct
5	R/W	0	hcodec/vififo
4	R/W	0	hcodec/vld_part
3	R/W	0	hcodec/vld
2	R/W	0	hcodec/assist

DOS_SW_RESET2

Reserved

DOS_SW_RESET3 0xff62fcd0

Bit(s)	R/W	Default	Description
24	R/W	0	hevc/afifo
26	R/W	0	hevc/mmu
19	R/W	0	hevc/sao
18	R/W	0	hevc/mpred
17	R/W	0	hevc/qdct
15	R/W	0	hevc/ipp
14	R/W	0	hevc/iqit
13	R/W	0	hevc/ddr
12	R/W	0	hevc/ccpu
11	R/W	0	hevc/mcpu
8	R/W	0	hevc/dblk
4	R/W	0	hevc/parser_state
3	R/W	0	hevc/parser
2	R/W	0	hevc/assist

DOS_SW_RESET4 0xff62fcdc

Bit(s)	R/W	Default	Description
11	R/W	0	HEVC Encoder/preset_n
10	R/W	0	HEVC Encoder /areset_n
9	R/W	0	HEVC Encoder /creset_n
8	R/W	0	HEVC Encoder /breset_n

8.3 System Booting

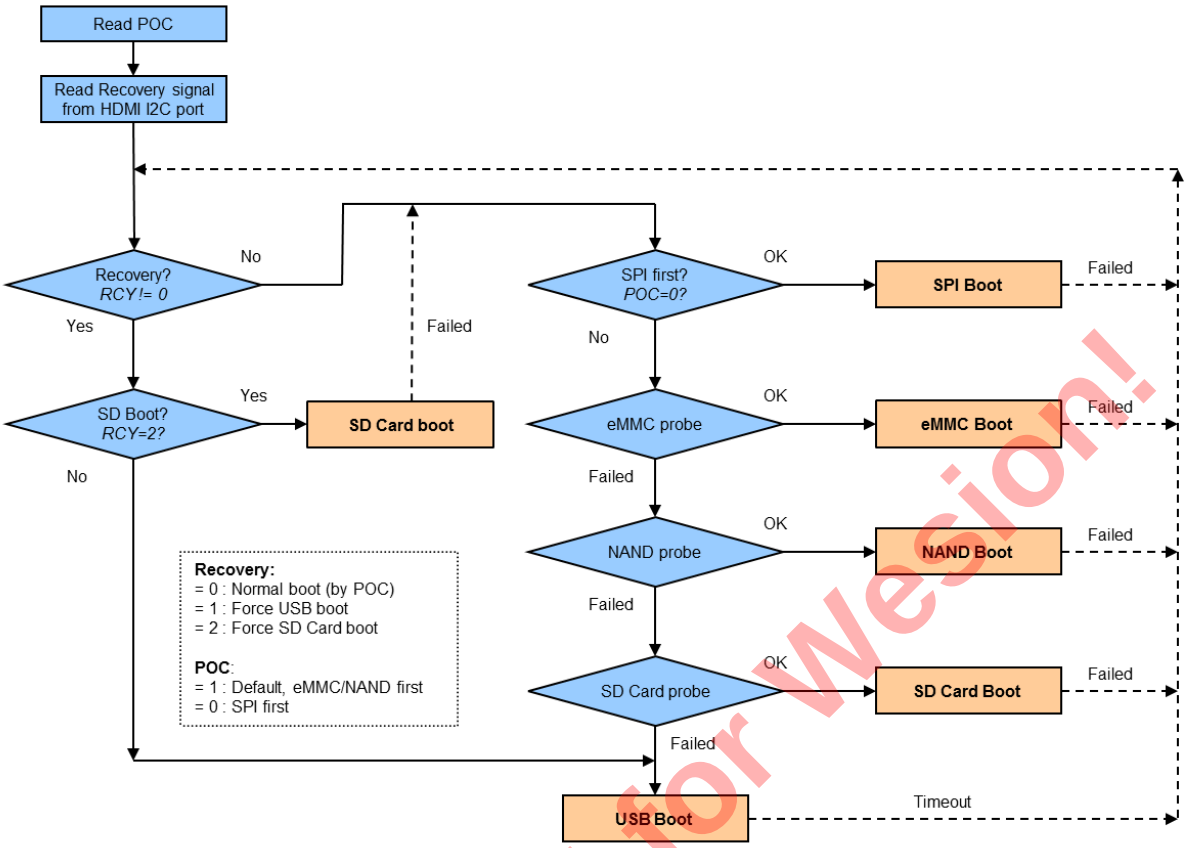
8.3.1 Overview

The part describes the power–on mode configuration of S922X, which include two portions: Cortex-M3 for security control and A53 for others.

8.3.2 Power-on Flow Chart

Figure 8-4 illustrates S922X’s power on sequence.

Figure 8-4 Power-on Flow Chart



8.4 CPU

The Cortex™-A53 MP subsystem of the chip is a high-performance, low-power, ARM macrocell with an L1 cache subsystem and an L2 cache subsystem that provide full virtual memory capabilities. The Cortex-A53 processor implements the ARMv8 architecture and runs32-bit ARM instructions and 64 bit ARMv8 instructions. The developers can follow the ARM official reference documents for programming details.

The Cortex-A53 processor features are:

- in-order pipeline with dynamic branch prediction
- ARM, Thumb, and ThumbEE instruction set support
- TrustZone security extensions
- Harvard level 1 memory system with a Memory Management Unit (MMU)
- 128-bit AXI master interface
- ARM CoreSight debug architecture
- trace support through an Embedded Trace Macrocell (ETMv4) interface
- Intelligent Energy Manager (IEM) support with
  - asynchronous AXI wrappers
  - two voltage domains
- Media Processing Engine (MPE) with NEON technology
- Supports FPU
- Supports Hardware Virtualization

The quad core Cortex™-A73 processor is paired with A53 processor in a big.Little configuration, with each core has L1 instruction and data chaches, together with a single shared L2 unified cache with A53. The developers can follow the ARM official reference documents for programming details.

The Cortex-A73 processor features are:

- Armv8-A Architecture
- 1-4x Symmetrical Multiprocessing (SMP) within a single processor cluster, and multiple coherent SMP processor clusters through AMBA 4 ACE technology
- AArch32 for full backward compatibility with Armv7
- AArch64 for 64-bit support and new architectural features
- TrustZone security technology
- NEON advanced SIMD
- DSP & SIMD extensions
- VFPv4 floating point
- Supports Hardware virtualization
- trace support through CoreSight SoC-400 interface

8.5 GPU

The G52 MP4 (6EE) GPU is a hardware accelerator for 2D and 3D graphics system which compatible with the following graphics standards: OpenGL ES 3.2 Vulkan 1.0 and OpenCL 2.0. The developers can follow the ARM and Khronos official reference documents for programming details. GPU supports max of 800MHz.

8.6 Clock

8.6.1 Overview

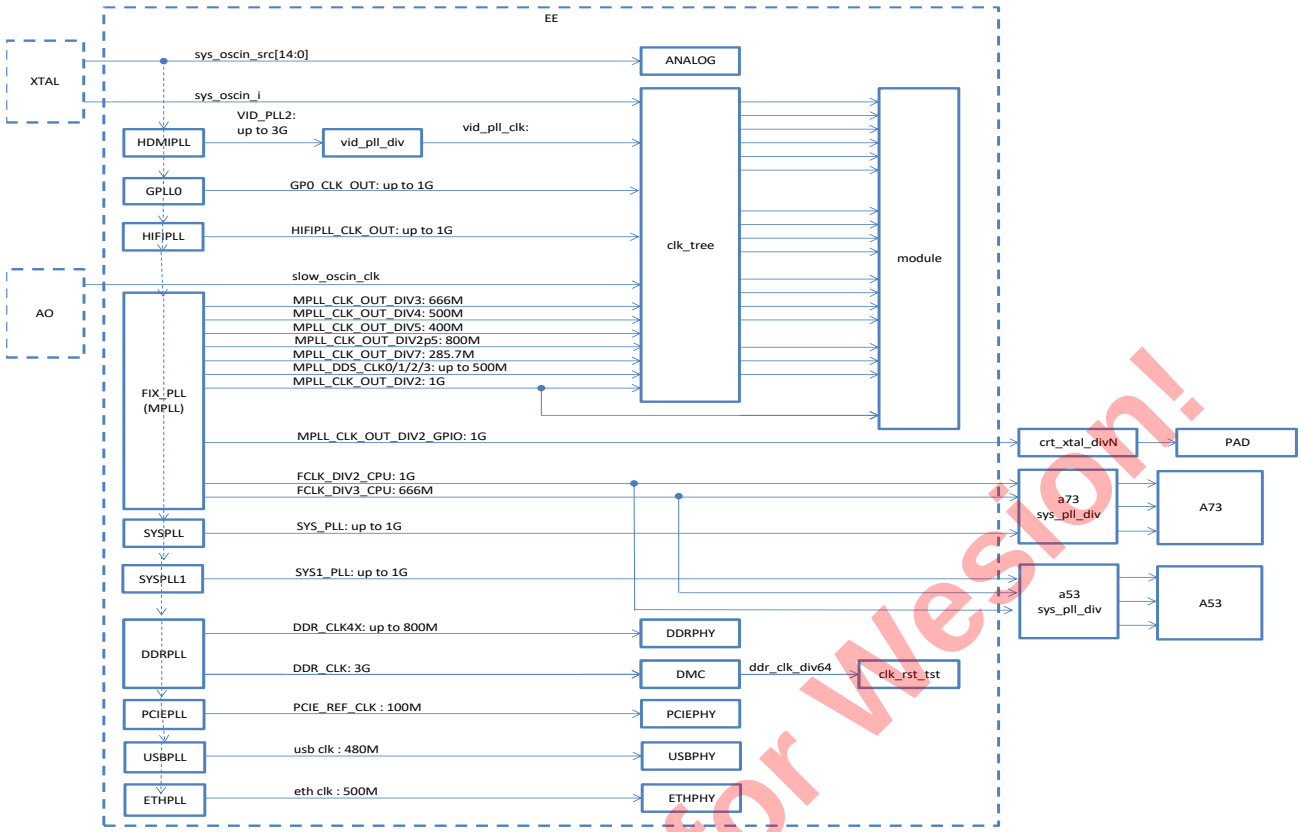
The clock and reset unit is an APB slave module that is designed for generating all of the internal and system clocks, resets of chip. S922X uses an external 24MHz crystal; there are 10 PLLs: MPLL(FIXPLL), SYS_PLL, SYS1_PLL, HDMI_PLL, GP0_PLL, DDR_PLL, HIFI_PLL, PCIE_PLL, ETHPHY_PLL and USBPHY_PLL, as shown in the following table.

Table 8-7 S922X PLLs

Type	Output Clock	Integer/Fractional	MAX FREQ	Spread Spectrum
SYS_PLL	SYS_CLK_OUT	Fractional	6G	yes
SYS1_PLL	SYS1_CLK_OUT	Fractional	6G	yes
GP0_PLL	GP0_CLK_OUT	Fractional	6G	yes
HIFI_PLL	HIFI_CLK_OUT	Fractional	6G	yes
PCIE_PLL	PCIE_REF_CLK_N/P		100M(fixed)	yes
ETHPHY_PLL	ETH_CLK		500M(fixed)	No
USBPHY_PLL	USB_CLK		480M(fixed)	No
MPLL	FCLK_DIV2_CPU		1G(fixed)	No
	FCLK_DIV3_CPU		666M(fixed)	No
	MPLL_CLK_OUT_DIV2_GPIO		1G(fixed)	No
	MPLL_CLK_OUT_DIV2		1G(fixed)	No
	MPLL_CLK_OUT_DIV2p5		800M(fixed)	No
	MPLL_CLK_OUT_DIV3		666M(fixed)	No
	MPLL_CLK_OUT_DIV4		500M(fixed)	No
	MPLL_CLK_OUT_DIV5		400M(fixed)	No
	MPLL_CLK_OUT_DIV7		285.7M(fixed)	No
	MPLL_DDS_CLK0	Fractional	500M	No
	MPLL_DDS_CLK1	Fractional	500M	No
	MPLL_DDS_CLK2	Fractional	500M	yes
	MPLL_DDS_CLK3	Fractional	500M	yes
HDMI_PLL	HDMI_CLK_OUT	Fractional	6G	yes
	HDMI_CLK_OUT2	Fractional	6G	yes
DDR_PLL	DDR_CLK4X_OUT	Fractional	4.8G	yes
	DDR_CLK_OUT	Fractional	4.8G	No

Clock Trees Figure 8-5 shows the clock connections of S922X. In this part, we will discuss A53/A73 clock tree, AO clock tree, HDMI clock tree and EE clock tree in detail.

Figure 8-5 Clock Connections

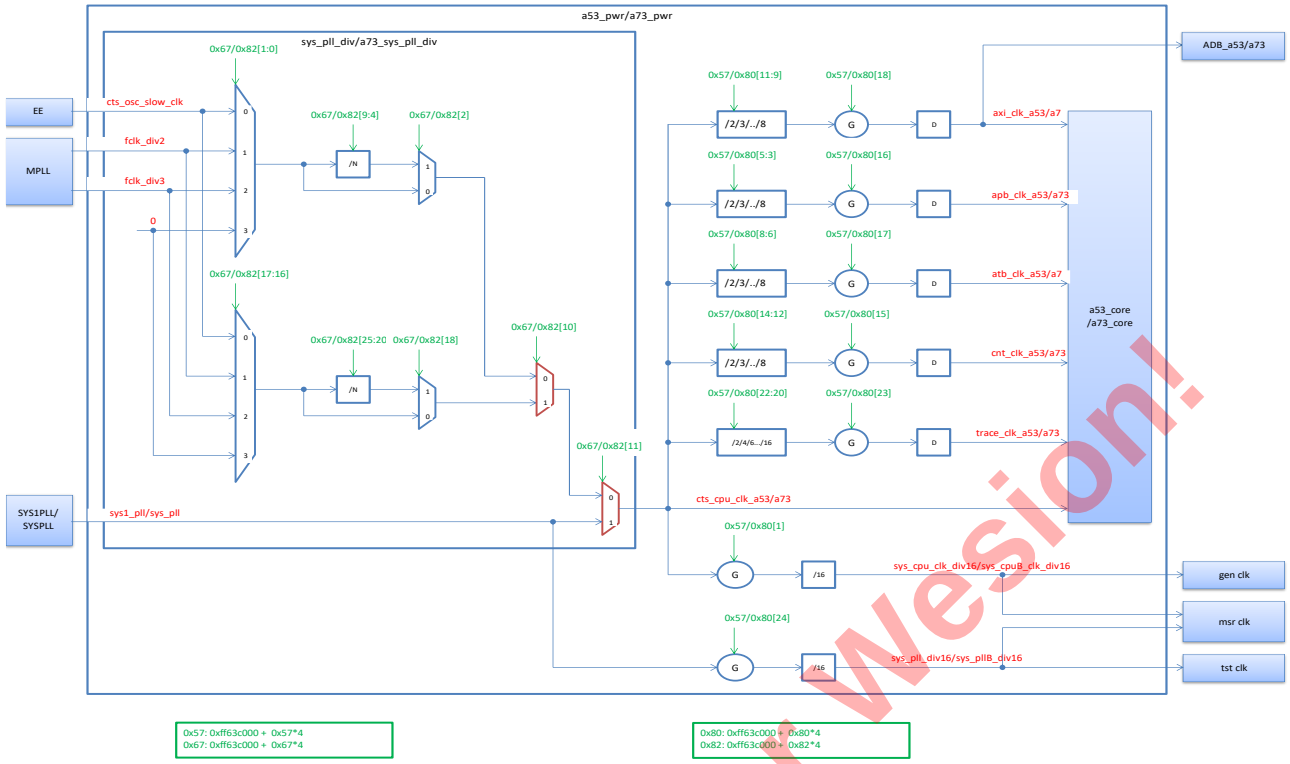


8.6.1.1 A53/A73 Clock Tree

A53/A73 has 4 clock source, as shown in the following figure, among which,

4. cts_osc_slow_clk is for low power and debug,
5. fclk_div2 and fclk_div3 are for frequencies lower than 1G.
6. sys_pll is for frequencies higher than 1G.

Figure 8-6 Mutil Phase PLLS of A53/A73



To avoid glitch when change frequencies, there are 2 specially designed dynamic muxes, labeled by red in the above figure. When frequencies changes, the dynamic muxes will first stop the first frequency, then start the second so there will be no mixing of 2 different frequencies thus generate no frequency glitch.

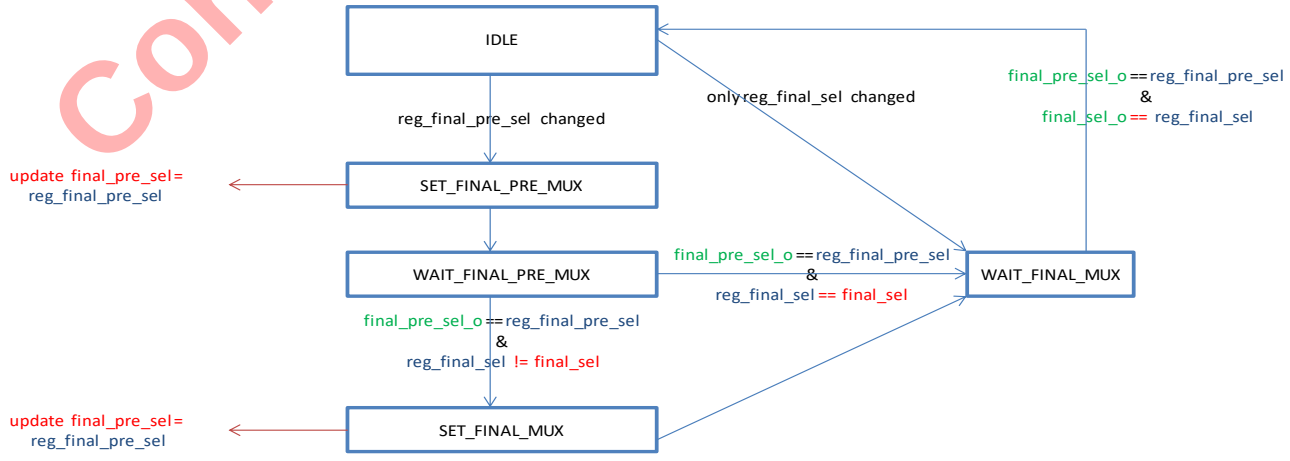
It is possible to do the following switch without glitch:

7. Between any 2 frequencies lower than 1GHz;
8. From a frequency lower than 1GHz to a frequency higher than 1GHz;

If the user want to switch between 2 frequencies both higher than 1GHz, it is strongly recommended to change to frequencies lower than 1GHz first.

The diagram of specially designed dynamic mux is shown in the following diagram:

Figure 8-7 Diagram of Dynamic Mux



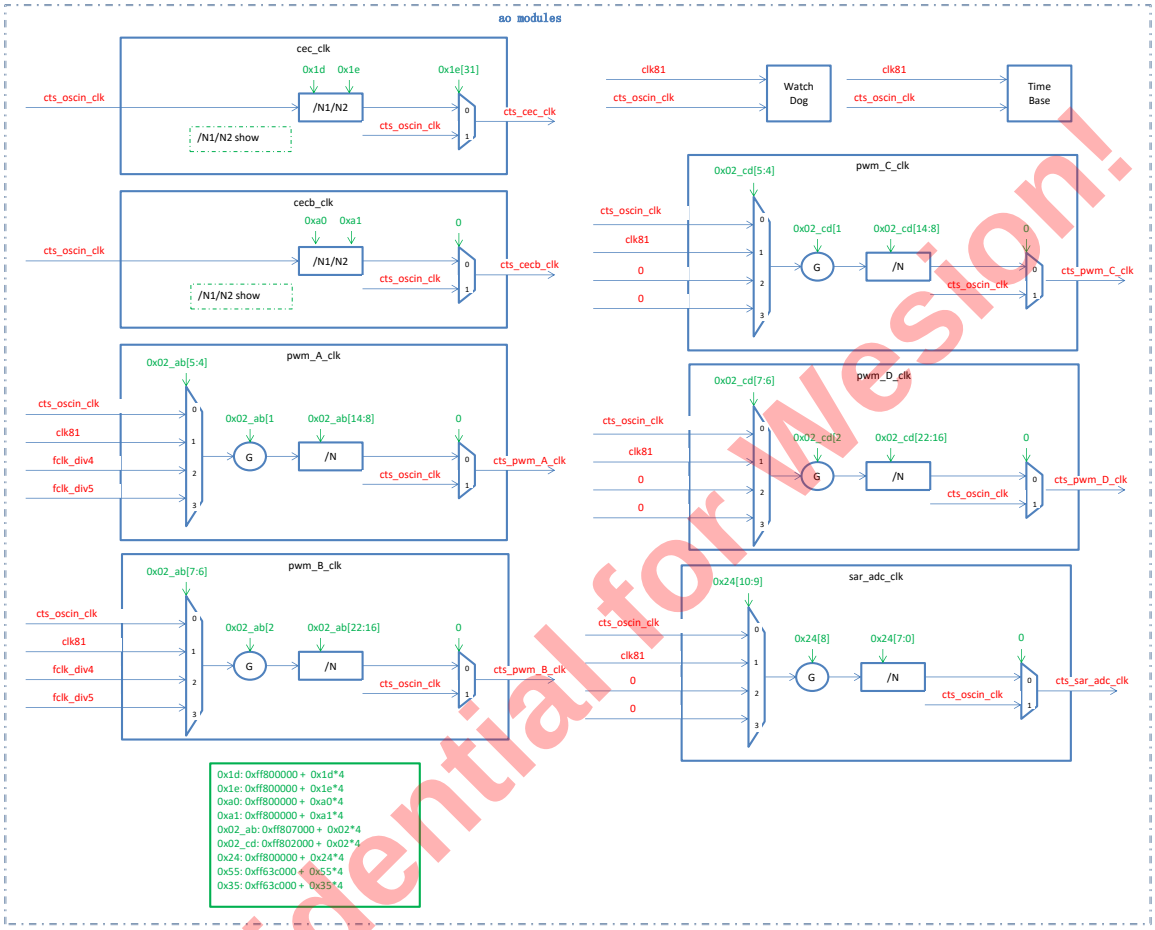
Note:

When use these 2 dynamic muxes, both control bits of these muxes have to be configured at the same time, otherwise they will not function correctly.

8.6.1.2 AO Clock Tree

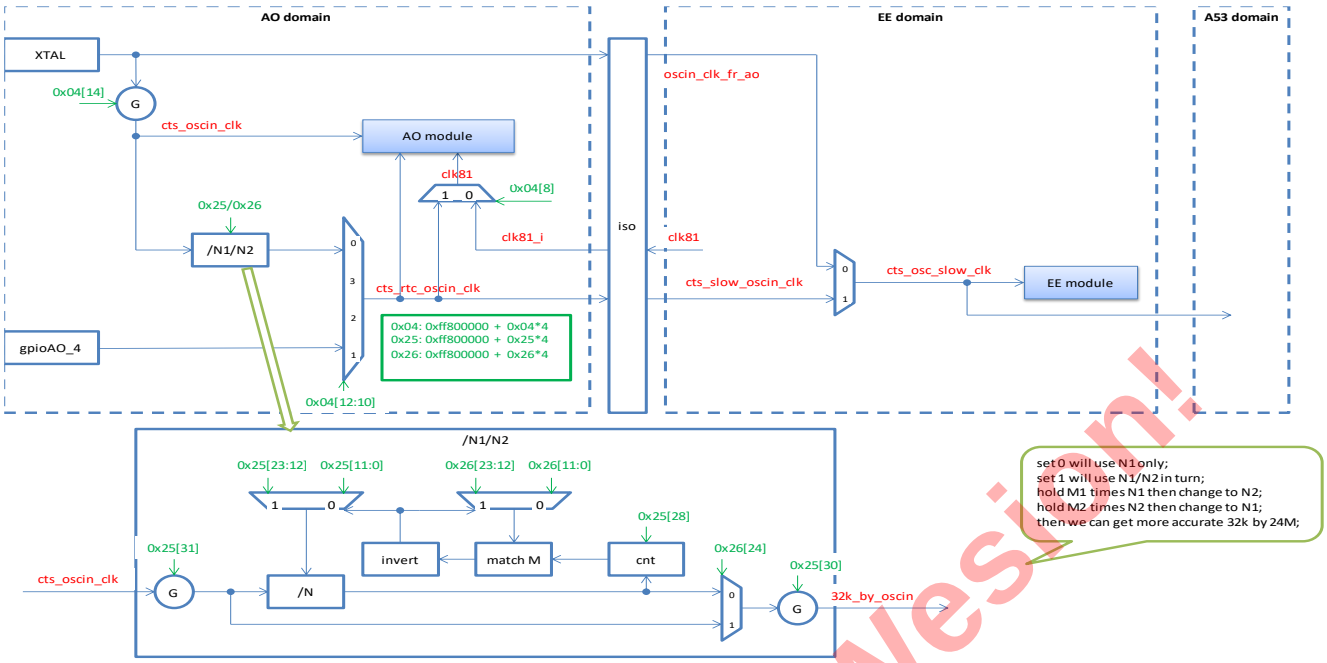
The following figure shows the clock source of AO modules:

Figure 8-8 AO Clock Sources



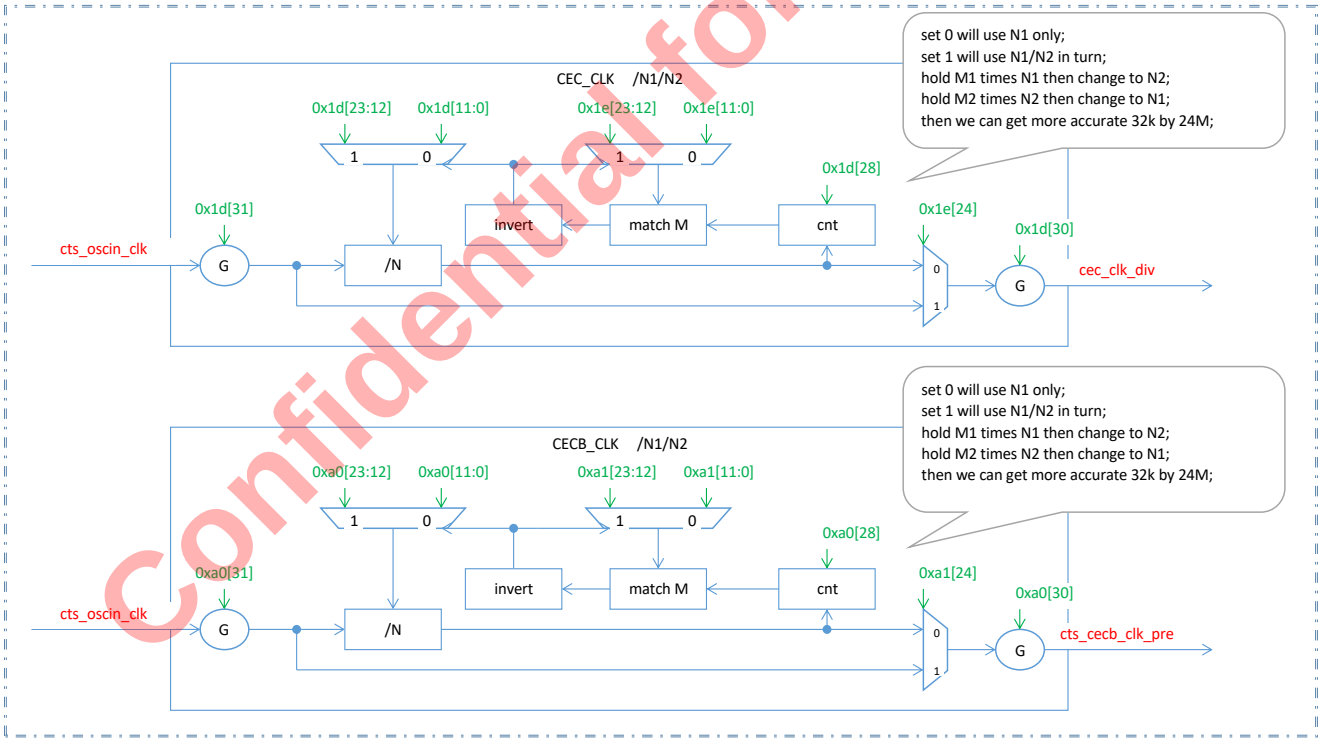
For low power mode, 32KHz clock is needed, and it is generated in AO domain as shown in the following diagram:

Figure 8-9 How to generate 32KHz Clock



To generate exact 32768Hz frequency, please check below diagram.

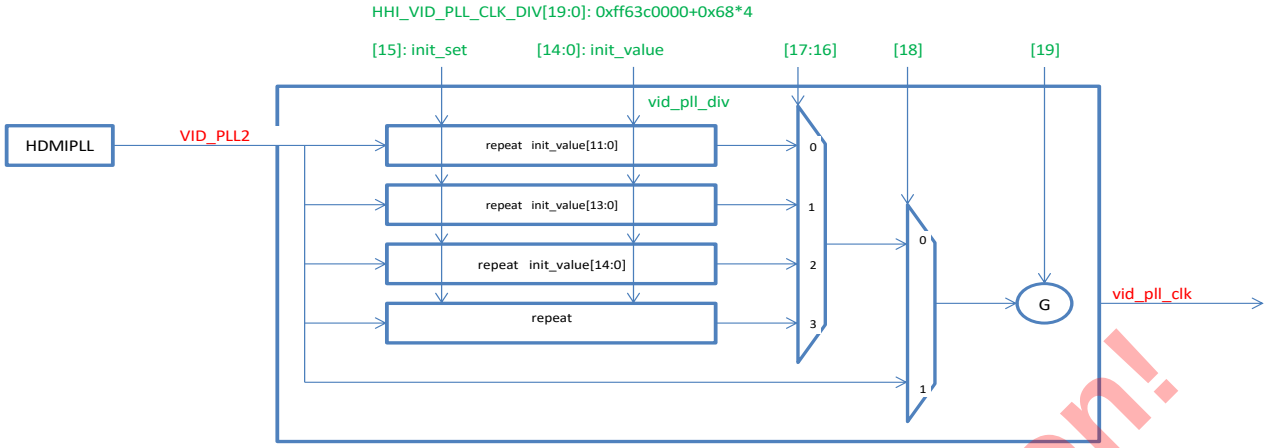
Figure 8-10 How to generate 32786Hz Clock



8.6.1.3 HDMI Clock Tree

The HDMI_PLL goes through vid_pll_div to generate new clock. The HDMI clock tree is shown in the following figure:

Figure 8-11 HDMI Clock Tree



8.6.1.4 EE Clock Tree

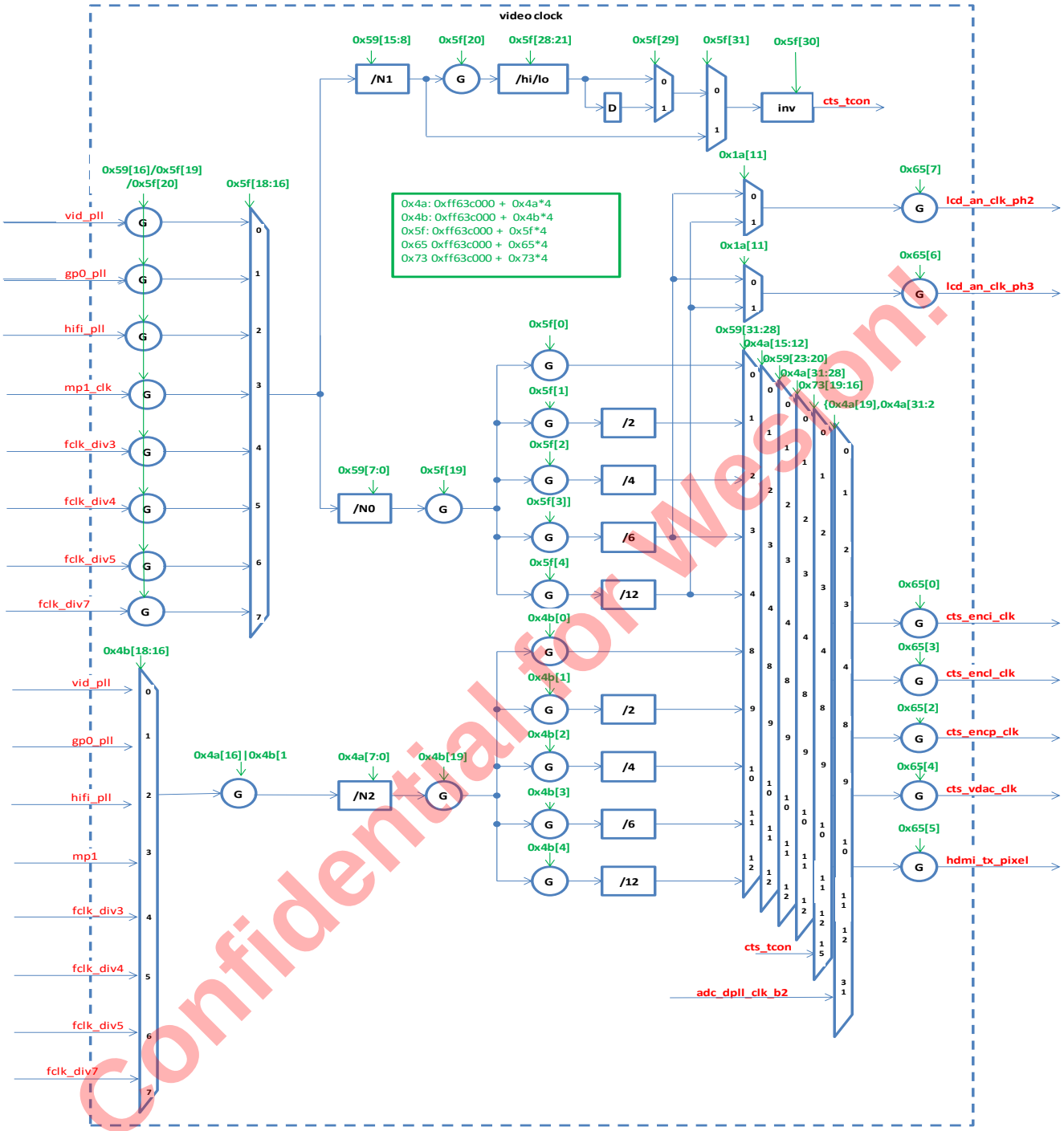
All EE clocks except video clock are listed in the following table:

Table 8-8 EE Clock

name	destination	gate	N	sel	src0	src1	src2	src3	src4	src5	src6	src7	base
cts_slow_oscin_clk	a73/a53			0x5d[9]	xtal	cts_rtc_oscin_clk							ff63c000
clk81_en_o	ao			0x5d[8]	cts_slow_oscin_clk	mpeg_pll_clk_final							ff63c000
clk81	all_module			0x5d[8]	cts_slow_oscin_clk	mpeg_pll_clk_final							ff63c000
mpeg_pll_clk_final		0x5d[7]	0x5d[31:0]	0x5d[14:12]	cts_slow_oscin_clk	fclk_div7	mp1	mp2	fclk_div4	fclk_div3	fclk_div5	ff63c000	
if 0x7a[31] = 0, use 0x78; if 0x7a[31] = 1, use 0x7a;	cts_hcodec_clk	dos	0x78[24]	0x78[22:16]	0x78[27:25]	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	hifi	gp0	xtal	ff63c000
if 0x7b[31] = 0, use 0x79; if 0x7b[31] = 1, use 0x7b;	cts_hevcb_clk	dos	0x7a[24]	0x7a[22:16]	0x7a[27:25]	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	hifi	gp0	xtal	ff63c000
if 0x7a[15] = 0, use 0x78; if 0x7a[15] = 1, use 0x7a;	cts_vdec_clk	dos	0x78[8]	0x78[6:0]	0x78[11:9]	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	hifi	gp0	xtal	ff63c000
if 0x7b[15] = 0, use 0x79; if 0x7b[15] = 1, use 0x7b;	cts_hevcf_clk	dos	0x78[8]	0x79[6:0]	0x79[11:9]	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	hifi	gp0	xtal	ff63c000
	cts_gp2d_clk	gp2d/dmc	0x7d[30]			cts_vapbclk							ff63c000
if 0x7d[31] = 0, use 0x7d[11:0]; if 0x7d[31] = 1, use 0x7d[27:16];	cts_vapbclk	vapb	0x7d[8]	0x7d[6:0]	0x7d[11:9]	fclk_div4	fclk_div3	fclk_div5	fclk_div7	mp1	vid_pll	mp2	fclk_div2p5	ff63c000
if 0x6c[31] = 0, use 0x6c[11:0]; if 0x6c[31] = 1, use 0x6c[27:16]	cts_mali_clk	mali	0x6c[8]	0x6c[6:0]	0x6c[11:9]	xtal	gp0	hifi	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	ff63c000
	cts_msr_clk	periphs	0x6001[19]	0x6002[7:0]	0x6001[26:20]	see clkmsr							ffd00000	
	cts_msr_clk_hs	periphs	0x6002[28]			fclk_div5							ffd00000	
	cts_pwm_A_clk	periphs	0x1c02[15]	0x1c02[14:8]	0x1c02[5:4]	xtal	vid_pll	fclk_div4	fclk_div3				ffd00000	
	cts_pwm_B_clk	periphs	0x1c02[23]	0x1c02[22:16]	0x1c02[7:6]	xtal	vid_pll	fclk_div4	fclk_div3				ffd00000	
	cts_pwm_C_clk	periphs	0x8031[15]	0x8031[14:8]	0x8031[5:4]	xtal	vid_pll	fclk_div4	fclk_div3				ffd00000	
	cts_pwm_D_clk	periphs	0x8031[23]	0x8031[22:16]	0x8031[7:6]	xtal	vid_pll	fclk_div4	fclk_div3				ffd00000	
	cts_pwm_E_clk	periphs	0x6402[15]	0x6402[14:8]	0x6402[5:4]	xtal	vid_pll	fclk_div4	fclk_div3				ffd00000	
	cts_pwm_F_clk	periphs	0x6402[23]	0x6402[22:16]	0x6402[7:6]	xtal	vid_pll	fclk_div4	fclk_div3				ffd00000	
	sc_clk	periphs	0x9400[24]	0x9402[19:12]	0x9402[31:30]	fclk_div4	fclk_div3	fclk_div5	xtal				ffd00000	
	cts_sd_emmc_clk_A	sd_emmc_A	0x99[7]	0x99[6:0]	0x99[11:9]	xtal	fclk_div2	fclk_div3	fclk_div5	fclk_div2	mp2	mp3	gp0	ff63c000
	cts_sd_emmc_clk_B	sd_emmc_B	0x99[23]	0x99[22:16]	0x99[27:25]	xtal	fclk_div2	fclk_div3	fclk_div5	fclk_div7	mp2	mp3	gp0	ff63c000
	cts_sd_emmc_clk_C	sd_emmc_C	0x97[7]	0x97[6:0]	0x97[11:9]	xtal	fclk_div2	fclk_div3	fclk_div5	fclk_div7	mp2	mp3	gp0	ff63c000
	cts_emcl_clk	vpu				vid_pll	gp0	hifi	mp1	fclk_div3	fclk_div4	fclk_div5	fclk_div7	
	cts_encl_clk	vpu				vid_pll	gp0	hifi	mp1	fclk_div3	fclk_div4	fclk_div5	fclk_div7	
	cts_encp_clk	vpu				vid_pll	gp0	hifi	mp1	fclk_div3	fclk_div4	fclk_div5	fclk_div7	
see video	cts_hdmi_tx_pixel_clk	vpu	see video			vid_pll	gp0	hifi	mp1	fclk_div3	fclk_div4	fclk_div5	fclk_div7	
	cts_vdac_clk	vpu				vid_pll	gp0	hifi	mp1	fclk_div3	fclk_div4	fclk_div5	fclk_div7	
	lcd_an_clk_ph2	vpu				vid_pll	gp0	hifi	mp1	fclk_div3	fclk_div4	fclk_div5	fclk_div7	
	lcd_an_clk_ph3	vpu				vid_pll	gp0	hifi	mp1	fclk_div3	fclk_div4	fclk_div5	fclk_div7	
	hcm_clkc	vpu				vid_pll	gp0	hifi	mp1	fclk_div3	fclk_div4	fclk_div5	fclk_div7	
	cts_hdcp22_skpclk	hdmirx	0x7c[24]	0x7c[22:16]	0x7c[26:25]	xtal	fclk_div4	fclk_div3	fclk_div5					ff63c000
	cts_vid_mean_clk	vpu	0x94[8]	0x94[6:0]	0x94[11:9]	xtal	fclk_div4	fclk_div3	fclk_div5					ff63c000
	cts_vid_lock_clk	vpu	0xf2[7]	0xf2[6:0]	0xf2[9:8]	xtal	cts_encl_clk	cts_encl_clk	cts_encp_clk					ff63c000
if 0x6f[31] = 0, use 0x6f[11:0]; if 0x6f[31] = 1, use 0x6f[27:16];	cts_vpu_clk	vpu	0x6f[8]	0x6f[6:0]	0x6f[11:9]	fclk_div3	fclk_div4	fclk_div5	fclk_div7	mp1	vid_pll	hifi	gp0	ff63c000
	cts_hdcp22_esmclk	vpu/dmc/hdmirx	0x7c[8]	0x7c[6:0]	0x7c[10:9]	fclk_div7	fclk_div4	fclk_div3	fclk_div5					ff63c000
	cts_oscin_clk	watchdog/periphs/audio_locker				xtal								
if0x83[9] = 0, 0x83 effective immediately if0x83[9] = 1, 0x83 effective after vpu_clkb_pulse	cts_vpu_clkb	vpu	0x83[8]	0x83[7:0]		cts_vpu_clkb_tmp								ff63c000
	cts_vpu_clkb_tmp	no_use	0x83[24]	0x83[19:16]	0x83[21:20]	cts_vpu_clk	fclk_div4	fclk_div5	fclk_div7					ff63c000
	cts_hdmitx_sys_clk	vpu	0x73[8]	0x73[6:0]	0x73[9:9]	xtal	fclk_div4	fclk_div2	fclk_div3					ff63c000
all fixed div64	tst_clk_out[15:0]	gpio	/64			sys_pll_div16	ddr_dpll_pt_clk	vid_pll	mp1	mp2	mp3	fclk_div5		
	cts_cdac_clk_c	CVBS	0x05[20]	0x05[15:0]	0x05[17:16]	xtal	fclk_div5							ff63c000
	cts_bt656		0x75[7]	0x75[6:0]	0x75[10:9]	fclk_div2	fclk_div3	fclk_div5	fclk_div7					ff63c000
	cts_eth_clk_rmii	ETH	0x76[8]	0x76[6:0]	0x76[11:9]	fclk_div2								ff63c000
	cts_eth_clk125M		0x76[7]	8		fclk_div2								ff63c000
	cts_wave420_aclk	wave420	0x9b[8]	0x9b[6:0]	0x9b[11:9]	xtal	fclk_div4	fclk_div3	fclk_div5	fclk_div7	mp2	gp1	gp0	ff63c000
	cts_wave420_bclk		0x9a[8]	0x9a[6:0]	0x9a[11:9]	xtal	fclk_div4	fclk_div3	fclk_div5	fclk_div7	mp2	gp1	gp0	ff63c000
	cts_wave420_cclk		0x9a[24]	0x9a[22:16]	0x9a[27:25]	xtal	fclk_div4	fclk_div3	fclk_div5	fclk_div7	mp2	gp1	gp0	ff63c000
if 0x6d[31] = 0, use 0x6d[11:0]; if 0x6d[31] = 1, use 0x6d[27:16];	cts_vpu_clkc	vpu	0x6d[8]	0x6d[6:0]	0x6d[11:9]	fclk_div4	fclk_div3	fclk_div5	fclk_div7	mp1	vid_pll	mp2	gp0	ff63c000
	cts_dsi_phy_clk	mipidsi	0x95[8]	0x95[6:0]	0x95[14:12]	vid_pll	gp0	hifi	mp1	fclk_div2	fclk_div2p5	fclk_div3	fclk_div7	ff63c000
	cts_dsi_meas_clk		0x94[20]	0x94[18:12]	0x94[23:21]	xtal	fclk_div4	fclk_div3	fclk_div5	vid_pll	gp0	fclk_div2	fclk_div7	ff63c000
	cts_spicc0_clk	spicc0	0xf7[6]	0xf7[5:0]	0xf7[9:7]	xtal	clk81	fclk_div4	fclk_div3	fclk_div2	fclk_div5	fclk_div7	gp0	ff63c000
	cts_spicc1_clk		0xf7[22]	0xf7[21:16]	0xf7[25:23]	xtal	clk81	fclk_div4	fclk_div3	fclk_div2	fclk_div5	fclk_div7	gp0	ff63c000
if 0x6b[31] = 0, use 0x6d[11:0]; if 0x6b[31] = 1, use 0x6d[27:16];	cts_nic_clk	nic/ccl	0x6b[8]	0x6b[6:0]	0x6b[11:9]	xtal	gp0	hifi	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	ff63c000
	cts_axi_sram	axi_sram	0x6b[24]	0x6b[22:16]	0x6b[27:25]	xtal	gp0	hifi	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	ff63c000
	cts_vpuanoq_clk	vpuanoq	0x72[8]	0x72[6:0]	0x72[11:9]	xtal	gp0	hifi	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	ff63c000
	cts_vpuanoq_clkb		0x72[24]	0x72[22:16]	0x72[27:25]	xtal	gp0	hifi	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	ff63c000
	cts_mipi_isp_clk	mipi_isp	0x70[8]	0x70[6:0]	0x70[11:9]	xtal	gp0	hifi	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	ff63c000
	cts_gdc_clk	gdc	0x5b[8]	0x5b[6:0]	0x5b[11:9]	xtal	gp0	hifi	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	ff63c000
	cts_gdc_axi_clk		0x5b[24]	0x5b[22:16]	0x5b[27:25]	xtal	gp0	hifi	fclk_div2p5	fclk_div3	fclk_div4	fclk_div5	fclk_div7	ff63c000
if 0x0d[31] = 0, use 0x0d[11:0]; if 0x0d[31] = 1, use 0x0d[27:16];	cts_mipi_csi_phy_clk	mipi_csi_phy	0x0d[8]	0x0d[6:0]	0x0d[11:9]	xtal	gp0	mp1	mp2	fclk_div3	fclk_div4	fclk_div5	fclk_div7	ff63c000

The video clock tree is shown as following:

Figure 8-12 Video Clock Tree



8.6.2 Frequency Calculation and Setting

The PLL frequecy tuning step is reference clock/2^{17}.

The general frequency tuning formula is shown as below.

$$\text{Target Frequency} = 24\text{MHz} \cdot \frac{DPLL\_M + \frac{DIV\_FRAC}{2^{17}}}{DPLL\_N} \cdot \frac{1}{OD}$$

The PLL fractional value weight table(if not other stated) is shown below:

Table 8-9 S922X Fractional Value Weight Table

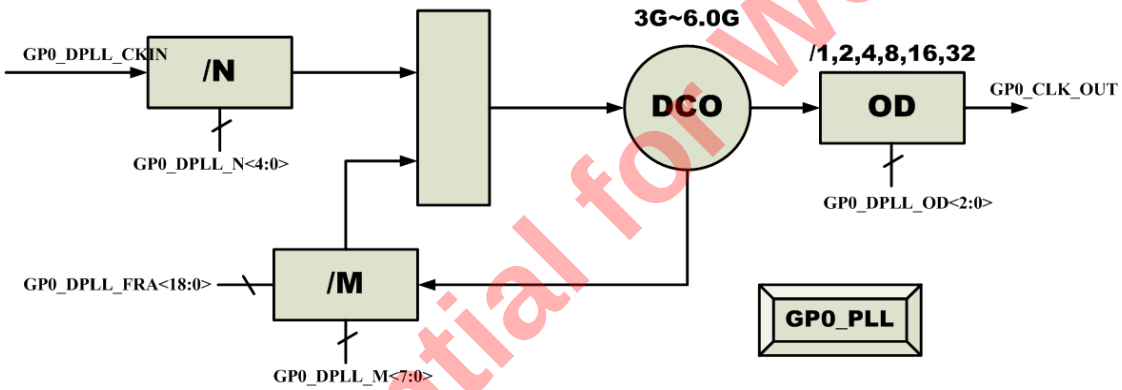
	Bit18	Bit17	Bit16	Bit15	..	..	..	..	..	..	..	Bit0
Weight	+/-	1	1/2	1/2^2	..	..	..	..	..	..	..	1/2^17

For example, to get a fractional value 0.875, you can set CNTL1[18:0] = 19'h1c000

8.6.2.1 SYS_PLL/SYS1_PLL

SYS_PLL diagram is shown as below:

Figure 8-13 SYS_PLL Diagram



DCO frequency is calculated with the following equation:

$$f_{DCO} = f_{REF} \cdot (M + frac) / N$$

OD control table is as following:

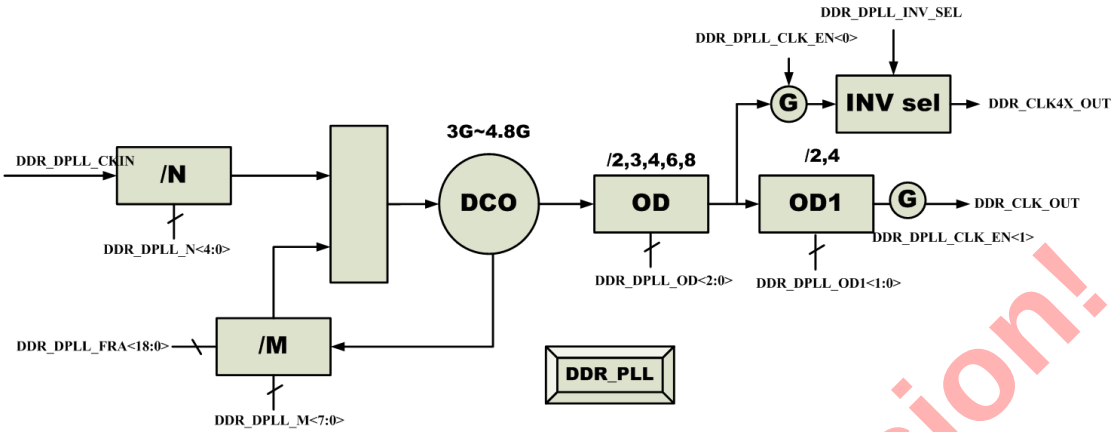
Table 8-10 S922X SYS_PLL OD Control Table

Block	Register	Function
OD	SYS_DPLL_OD<2:0>	000: /1
		001:/2
		010:/4
		011:/8
		100:/16
		101:/32

8.6.2.2 DDR_PLL

DDR_PLL diagram is shown as below:

Figure 8-14 DDR_PLL Diagram



DCO frequency is calculated with the following equation:

$$f_{DCO} = f_{REF} \cdot (M + frac) / N$$

OD control table is as following:

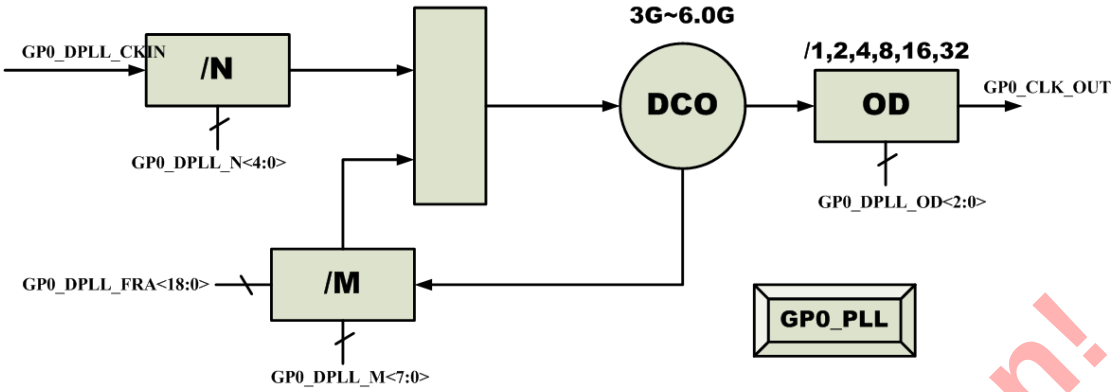
Table 8-11 S922X DDR_PLL Control Table

Block	Register	Function
OD	DDR_DPLL_OD<2:0>	000: /2
		001:/3
		010:/4
		011:/6
		100:/8
OD1	DDR_DPLL_OD1<1:0>	*0:/2
		*1:/4

8.6.2.3 GP0 PLL

GP0 PLL diagram is shown as below:

Figure 8-15 GP0 PLL Diagram



DCO frequency is calculated with the following equation:

$$f_{DCO} = f_{REF} \cdot (M + frac) / N$$

OD control table is as following:

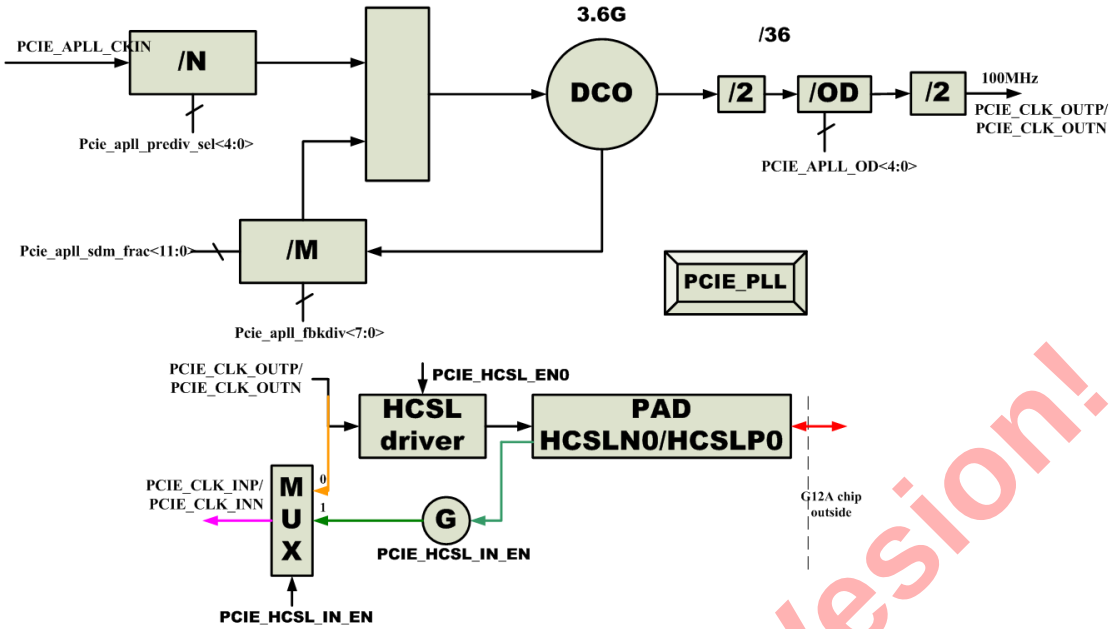
Table 8-12 S922X GP0 PLL OD Control Table

Block	Register	Function
OD	GP0_DPLL_OD<2:0>	000: /1
		001:/2
		010:/4
		011:/8
		100:/16
		101:/32

8.6.2.4 PCIE PLL

PCIE PLL diagram is shown as below.

Figure 8-16 PCIE PLL Diagram



DCO frequency is calculated with the following equation:

$$f_{DCO} = f_{REF} \cdot (M + frac) / N$$

OD control table is as following:

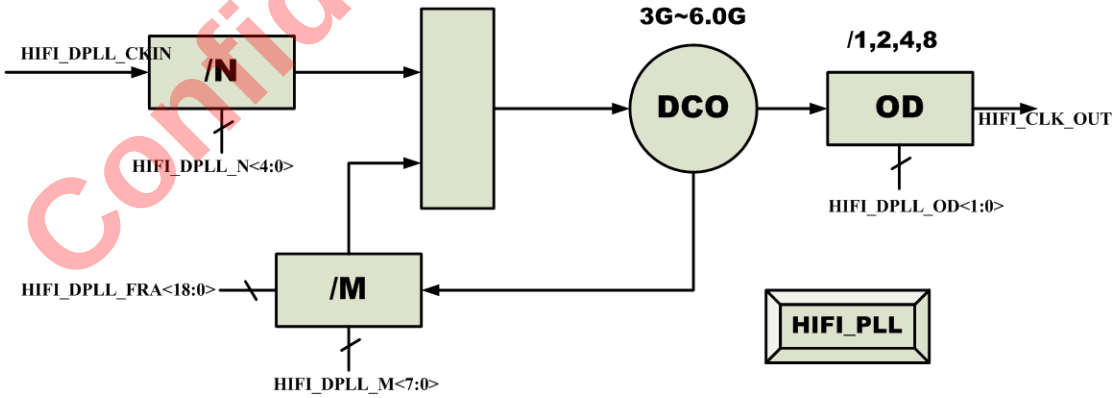
Table 8-13 S922X PCIE PLL OD Control Table

Block	Register	Function
OD	PCIE_APLL_OD<4:0>	OD<4:0>=5'h09 : 1/9

8.6.2.5 HIFI PLL

HIFI PLL diagram is shown as below:

Figure 8-17 HIFI PLL Diagram



DCO frequency is calculated with the following equation:

$$f_{DCO} = f_{REF} \cdot (M + frac) / N$$

OD control table is as following:

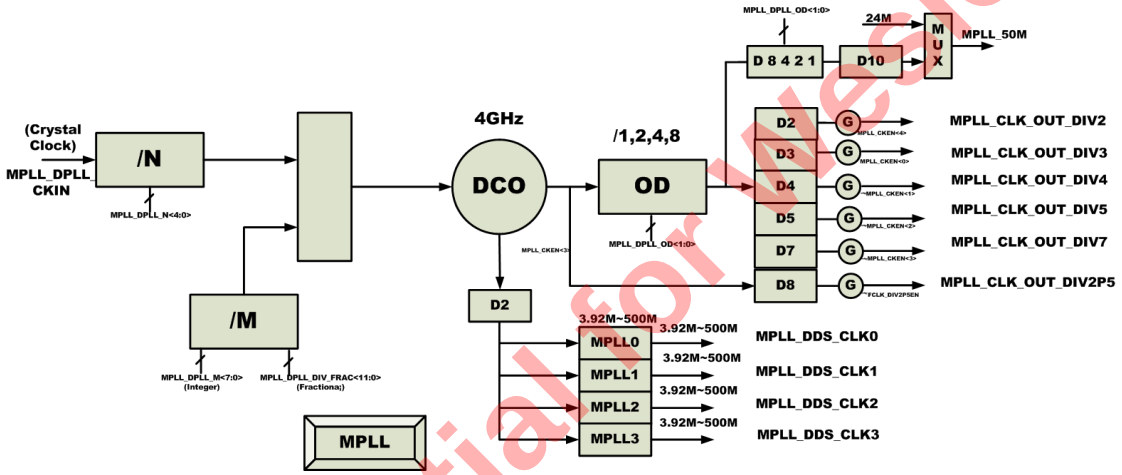
Table 8-14 S922X HIFI PLL OD Control Table

Block	Register	Function
OD	HIFI_DPLL_OD<1:0>	00: /1
		01:/2
		10:/4
		10:/8

8.6.2.6 MPLL(Fixed PLL)

MPLL diagram is shown as below:

Figure 8-18 MPLL Diagram



The output frequency calculation equations are shown as following(if N>3. MPLL_DDS_CLK2 and MPLL_DDS_CLK3 has SSG function):

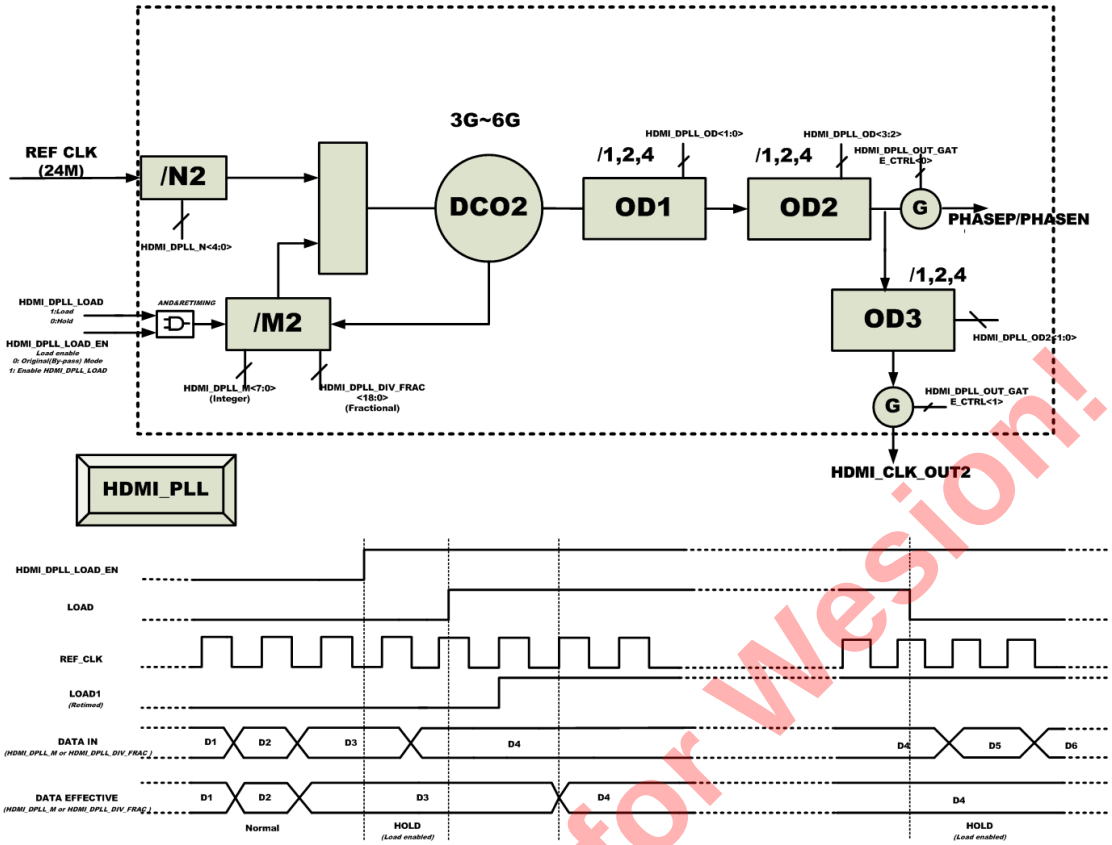
Fout_2G=Fref*M*OD_FB/N

$$MPLL\_CLK\_OUT2=f(N2\_integer, SDM\_IN) = \left(\frac{2Ghz}{(N2\_integer + \frac{SDM\_IN}{16384})}\right)$$

8.6.2.7 HDMI PLL

HDMI PLL diagram is shown as below.

Figure 8-19 HDMI PLL Diagram



OD control table is as following:

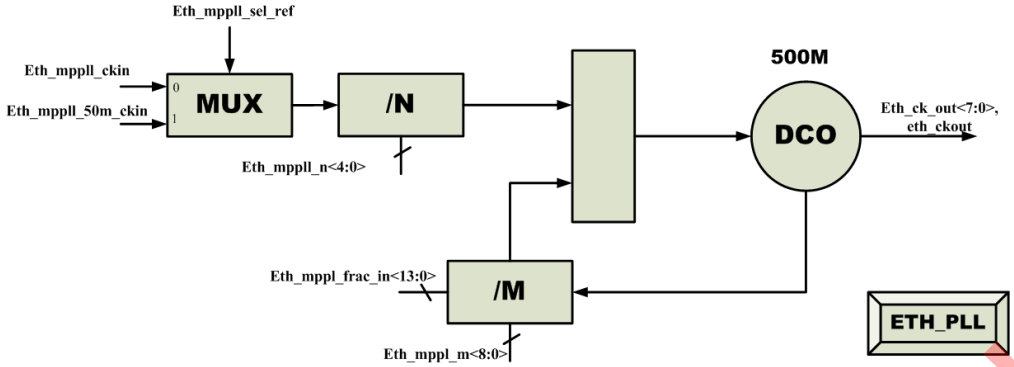
Table 8-15 S922X HDMI PLL OD Control Table

Block	Register	Function
OD1	HDMI_DPLL_OD<1:0>	00: /1
		01:/2
		10:/4
OD2	HDMI_DPLL_OD<3:2>	00: /1
		01:/2
		10:/4
OD3	HDMI_DPLL_OD2<1:0>	00: /1
		01:/2
		10:/4

8.6.2.8 ETHPHY PLL

ETH PLL diagram is shown as below:

Figure 8-20 ETH PLL Diagram



DCO frequency is 500M.

The Eth_PLL fractional value weight table is shown below:

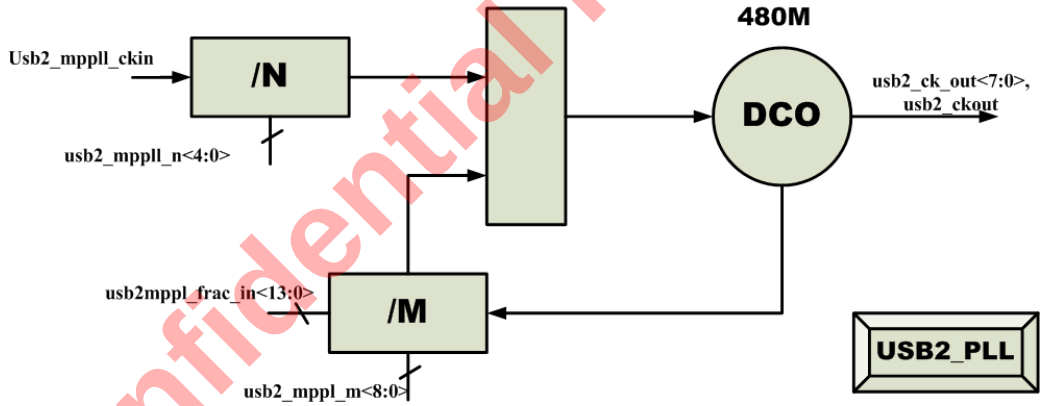
Table 8-16 ETHPHY_PLL Fractional Value Weight Table

	Bit13	Bit12	Bit11	Bit10	..	..	..	..	..	..	..	Bit0
Weight	1/2	1/2^2	1/2^3	1/2^4	..	..	..	..	..	..	..	1/2^14

8.6.2.9 USBPHY PLL

USB PLL diagram is shown as below:

Figure 8-21 USB PLL Diagram



DCO frequency is 500M.

The USB_PLL fractional value weight table is shown below:

Table 8-17 USBPHY_PLL Fractional Value Weight Table

	Bit13	Bit12	Bit11	Bit10	..	..	..	..	..	..	..	Bit0
Weight	1/2	1/2^2	1/2^3	1/2^4	..	..	..	..	..	..	..	1/2^14

8.6.3 Clock Gating

Modules and sub-modules within the chip can be disabled by shutting off the clock. The control for these clocks comes from six CBUS registers that collectively make up a 64-bit register that controls the MPEG_DOMAIN and a 32-bit register that controls the OTHER_ DOMAIN. The table below indicates the Bits associated with either the MPEG_DOMAIN and OTHER_DOMAIN gated clock enables. The table is organized by function rather than by bit order because it makes it easier to determine how to turn on/off

a particular function within the chip. If a bit is set high, the clock is enabled. If a bit is set low, the clock is turned off and the module is disabled.

Table 8-18 AO Domain Clock Gating CBUS 0x1055 control Bits

Address	Bit(s)	Module Description
0xff80004C	31:10	Unused
	9	AO2DDR_AHB
	8	SARADC
	7	IR_OUT
	6	AO_UART2
	5	PROD_I2C
	4	AO_UART2
	3	AO_I2C_S0
	2	AO_I2C_M0
	1	IR_IN
	0	AHB
0xff800050	31:6	Unused
	5	M4_HCLK
	4	M4_FCLK
	3	RTI
	2	AHB_SRAM
	1	M3
	0	MAILBOX

Table 8-19 EE Domain Clock Gating (clk81)

Address	Bit(s)	Module Description
0xff63c140 0xff63c0c0	31	
	30	
	29	
	28	ACODEC

Address	Bit(s)	Module Description
	27	
	26	EMMC_C
	25	EMMC_B
	24	EMMC_A
	23	ASSIST_MISC
	22	
	21	
	20	MIPI_DSI_PHY
	19	HIU Registers
	18	
	17	
	16	
	15	
	14	SPICC_1
	13	UART_EE_A
	12	RANDOM64
	11	SMARTCARD
	10	SANA
	9	I2C
	8	SPICC_0
	7	PERIPHS module top level (there are separate register bits for internal blocks)
	6	PL310 (AXI Matrix) to CBUS
	5	ISA module
	4	ETH_PHY
	3	MIPI_DSI_HOST
	2	ALOCKER
	1	U_DOS_TOP()
	0	DMC/AM2AXI_ARB

Address	Bit(s)	Module Description
0xff63c144 0xff63c0c4	31	
	30	
	29	AHB ARB0
	28	
	27	PCIEPHY
	26	USB General
	25	U_parser_top()
	24	PCIECOMB
	23	RESET
	22	
	21	
	20	General 2D Graphics Engine
	19	
	18	
	17	
	16	EE_UART1
	15	
	14	
	13	ADC
	12	
	11	AIFIFO
	10	
	9	
	8	
	7	
	6	
	5	
	4	Set top box demux module u_stb_top.clk
	3	Ethernet core logic

Address	Bit(s)	Module Description
	2	
	1	
	0	AUDIO
0xff63c148 0xff63c0c8	31	
	30	GIC
	29	
	28	
	27	
	26	
	25	VPU Interrupt
	24	
	23	
	22	
	21	
	20	
	19	
	18	
	17	
	16	GDC
	15	EE_UART2
	14	
	13	
	12	
	11	MMC PCLK
	10	
	9	
	8	USB1 to DDR bridge
	7	
	6	BT656

Address	Bit(s)	Module Description
	5	
	4	HDMITX_TOP HDMITX_PCLK
	3	HDMITX_TOP HTX_HDCP22_PCLK
	2	AHB control bus
	1	
	0	
0xff63c150 0xff63c0cc	31	
	30	
	29	
	28	
	27	
	26	VCLK2_OTHER
	25	VCLK2_VENCL
	24	VCLK2_VENCL MMC Clock All
	23	VCLK2_ENCL
	22	VCLK2_ENCT
	21	Random Number Generator
	20	ENC480P
	19	
	18	
	17	
	16	IEC958_GATE
	15	
	14	AOCLK_GATE
	13	
	12	
	11	
	10	DAC_CLK

Address	Bit(s)	Module Description
	9	VCLK2_ENCP
	8	VCLK2_ENCI
	7	VCLK2_OTHER
	6	VCLK2_VENCT
	5	VCLK2_VENCT
	4	VCLK2_VENCP
	3	VCLK2_VENCP
	2	VCLK2_VENCI
	1	VCLK2_VENCI
	0	
0xff63c154	31	
	30	
	29	
	28	
	27	
	26	
	25	
	24	
	23	
	22	
	21	
	20	
	19	
	18	
	17	
	16	
	15	
	14	
	13	

Address	Bit(s)	Module Description
	12	
	11	
	10	
	9	
	8	
	7	
	6	
	5	
	4	sec_ahb_apb3
	3	reset_sec
	2	rom_boot
	1	efuse
	0	dma

8.6.4 Clock Measure

The chip contains a module that can measure the frequency of internal clock. The frequency measurement is a simple counter measurement in which the counter (driven by MCLK) is enabled for a programmable amount of time. The sources are listed in the following table.

Table 8-20 Clock Measure Source

No	Source
127	1'b0
126	Reserved
125	Reserved
124	cts_gdc_core_clk
123	cts_gdc_axi_clk
122	mod_audio_pdm_dclk_o
121	audio_spdifin_mst_clk
120	audio_spdifout_mst_clk
119	audio_spdifout_b_mst_clk
118	audio_pdm_sysclk

No	Source
117	audio_resample_clk
116	audio_tdmin_a_sclk
115	audio_tdmin_b_sclk
114	audio_tdmin_c_sclk
113	audio_tdmin_lb_sclk
112	audio_tdmout_a_sclk
111	audio_tdmout_b_sclk
110	audio_tdmout_c_sclk
109	c_alocker_out_clk
108	c_alocker_in_clk
107	au_dac_clk_g128x
106	ephy_test_clk
105	am_ring_osc_clk_out_ee[9]
104	am_ring_osc_clk_out_ee[8]
103	am_ring_osc_clk_out_ee[7]
102	am_ring_osc_clk_out_ee[6]
101	am_ring_osc_clk_out_ee[5]
100	am_ring_osc_clk_out_ee[4]
99	am_ring_osc_clk_out_ee[3]
98	cts_ts_clk
97	cts_vpu_clkb_tmp
96	cts_vpu_clkb
95	eth_phy_plltxclk
94	eth_phy_rxclk
93	1'b0
92	sys_pllB_div16
91	sys_cpuB_clk_div16
90	cts_hdmitx_sys_clk
89	HDMI_CLK_TODIG

No	Source
88	Reserved
87	1'b0
86	cts_ML_core_clk
85	cts_ML_axi_clk
84	co_tx_clk
83	co_rx_clk
82	cts_ge2d_clk
81	cts_vapbclk
80	rng_ring_osc_clk[3]
79	rng_ring_osc_clk[2]
78	rng_ring_osc_clk[1]
77	rng_ring_osc_clk[0]
76	cts_cci_clk
75	cts_hevcf_clk
74	Reserved
73	cts_pwm_C_clk
72	cts_pwm_D_clk
71	cts_pwm_E_clk
70	cts_pwm_F_clk
69	cts_hdcp22_skpclk
68	cts_hdcp22_esmclk
67	cts_dsi_phy_clk
66	cts_vid_lock_clk
65	cts_spicc_0_clk
64	cts_spicc_1_clk
63	cts_dsi_meas_clk
62	cts_hevcb_clk
61	gpio_clk_msr
60	1'b0

No	Source
59	cts_hcodec_clk
58	cts_HEVC Encoder _bclk
57	cts_HEVC Encoder _cclk
56	cts_HEVC Encoder _aclk
55	vid_pll_div_clk_out
54	cts_vpu_clkc
53	cts_sd_emmc_clk_A
52	cts_sd_emmc_clk_B
51	cts_sd_emmc_clk_C
50	mp3_clk_out
49	mp2_clk_out
48	mp1_clk_out
47	ddr_dpll_pt_clk
46	cts_vpu_clk
45	cts_pwm_A_clk
44	cts_pwm_B_clk
43	fclk_div5
42	mp0_clk_out
41	mac_eth_rx_clk_rmii
40	1'b0
39	cts_bt656_clk0
38	cts_vdin_meas_clk
37	cts_cdac_clk_c
36	cts_hdmi_tx_pixel_clk
35	cts_Dvalin_clk
34	eth_mppll_50m_ckout
33	sys_cpu_ring_osc_clk[1]
32	cts_vdec_clk
31	mpll_clk_test_out

No	Source
30	pcie_clk_inn
29	pcie_clk_inp
28	cts_sar_adc_clk
27	co_clkin_to_mac
26	sc_clk_int
25	cts_eth_clk_rmii
24	cts_eth_clk125Mhz
23	mpll_clk_50m
22	mac_eth_phy_ref_clk
21	lcd_an_clk_ph3
20	rtc_osc_clk_out
19	lcd_an_clk_ph2
18	sys_cpu_clk_div16
17	sys_pll_div16
16	cts_FEC_CLK_2
15	cts_FEC_CLK_1
14	cts_FEC_CLK_0
13	mod_tcon_clko
12	hifi_pll_clk
11	mac_eth_tx_clk
10	cts_vdac_clk
9	cts_encl_clk
8	cts_encp_clk
7	clk81
6	cts_enci_clk
5	1'b0
4	gp0_pll_clk
3	sys_cpu_ring_osc_clk[0]
2	am_ring_osc_clk_out_ee[2]

No	Source
1	am_ring_osc_clk_out_ee[1]
0	am_ring_osc_clk_out_ee[0]

The sources are controlled by register MSR_CLK_REG0.

8.6.5 Register Description

HIU(base:32'hFF63C000)

Each register final address = 0xFF63C000 + offset * 4

HHI_MIPI_CNTL0 0x00

Bit(s)	R/W	Default	Description
31:16	R/W	0	REGISTER CONTROL B<15:10>:common ref gen control B9:Vref select 0=VR 1=Vbg B8:Iref select 0=I_rout 1=Ibg B7:enable Ibg B6/5:unused B<4:0>:VR trimming control 10mV/step Work mode:1010,0100,1000,0111
15:0	R/W	0	REGISTER CONTROL B3:VR gen from Ibg enable, B2:VR gen by avdd18 enable Other bits unused set 0 Work mode:0000,0000,0000,1000

HHI_MIPI_CNTL1 0x01

Bit(s)	R/W	Default	Description
19	R/W	0	dsi_edp_aux_tx_en AUX TX ENABLE(useless)
18	R/W	0	dsi_prbs_clk_in CLOCK FROM CORE FOR TEST.(max frequency 1GHZ)
17	R/W	0	dsi_edp_aux_tx AUX TX INPUT(useless)
16	R/W	0	dsi_vbg_en Bandgap Enable signal

Bit(s)	R/W	Default	Description
15:0	R/W	0	REGISTER CONTROL B<15:12>:unused B<11:8>:test output select b7:RX output inv select B6:unused B<5:4>:LPTX slew-rate select B3:LRTX leak-current enable B2:input inv select B1:input select 0=prbs 1=normal input B0:prbs7 enable Work mode:0000,0000,0010,1110

HHI_MIPI_CNTL2 0x02

Bit(s)	R/W	Default	Description
31:16	R/W	0	REGISTER CONTROL B<15:9>:CH cell power up B<8:3>:CH control B2/1:unused B0:LPULPS enable Work mode:0010,0110,1000,0000
15:0	R/W	0	REGISTER CONTROL B<15:11>: CH<0:4> power up B10:RX Ibias enable B9:unused B<8:0>:LPRX reference voltage control Work mode:1111,1100,0101,1010

HHI_MIPI_STS 0x03

Bit(s)	R/W	Default	Description
8	R	0	AUX RX OUTPUT(useless)
7-0	R	0	LPRX and LPCD receiver primitive state data <LPCD_OUTN_L,LPCD_OUTN_L,LPCD_OUTP_ H,LPCD_OUTP_L,LP_OUTN_H,LP_OUTN_L,LP _OUTP_H,LP_OUTP_L>

HHI_CHECK_CLK_RESULT 0x04

Bit(s)	R/W	Default	Description
31	R	0	sys_cpu_clk check_result
30	R	0	sys_cpuB_clk check_result
3	R	0	ML_check_result
2	R	0	encp_check_result
1	R	0	hevcb_clk_check_result
0	R	0	mali_clk_check_result

SCR System Clock Reference 0x0B

Bit(s)	R/W	Default	Description
31-0	R/W	0	System clock reference high: bits 31:16

TIMEOUT_VALUE 0x0F

Bit(s)	R/W	Default	Description
15-12	R	0	Unused
11-0	R/W	0	Program timer

Increased by 1 every 900 cycles. Triggers timer interrupt to CPU when it expires.

HHI_GP0_PLL_CNTL0 0x10

Bit(s)	R/W	Default	Description
[31]	R	0	gp0_dpll_lock
[30]	R	0	gp0_dpll_lock_a
[29]	R/W	1	gp0_dpll_reset
[28]	R/W	0	gp0_dpll_en
[18:16]	R/W	0	gp0_dpll_od
[14:10]	R/W	0	gp0_dpll_N
[7:0]	R/W	0	gp0_dpll_M

HHI_GP0_PLL_CNTL1 0x11

Bit(s)	R/W	Default	Description
[18:0]	R/W	0	gp0_dpll_frac

HHI_GP0_PLL_CNTL2 0x12

Bit(s)	R/W	Default	Description
[22:20]	R/W	0	gp0_dpll_fref_sel
[17:16]	R/W	0	gp0_dpll_os_ssc
[15:12]	R/W	0	gp0_dpll_ssc_str_m
[8]	R/W	0	gp0_dpll_ssc_en
[7:4]	R/W	0	gp0_dpll_ssc_dep_sel
[1:0]	R/W	0	gp0_dpll_ss_mode

HHI_GP0_PLL_CNTL3 0x13

Bit(s)	R/W	Default	Description
31	R/W	0	gp0_dpll_afc_bypass
30	R/W	0	gp0_dpll_afc_clk_sel
29	R/W	0	gp0_dpll_code_new
28	R/W	0	gp0_dpll_dco_m_en
27	R/W	0	gp0_dpll_dco_sdm_en
26	R/W	0	gp0_dpll_div2
25	R/W	0	gp0_dpll_div_mode
24	R/W	0	gp0_dpll_fast_lock
23	R/W	0	gp0_dpll_fb_pre_div
22	R/W	0	gp0_dpll_filter_mode
21	R/W	0	gp0_dpll_fix_en
20	R/W	0	gp0_dpll_freq_shift_en
19	R/W	0	gp0_dpll_load
18	R/W	0	gp0_dpll_load_en
17	R/W	0	gp0_dpll_lock_f
16	R/W	0	gp0_dpll_pulse_width_en
15	R/W	0	gp0_dpll_sdmnc_en
14	R/W	0	gp0_dpll_sdmnc_mode
13	R/W	0	gp0_dpll_sdmnc_range
12	R/W	0	gp0_dpll_tdc_en
11	R/W	0	gp0_dpll_tdc_mode_sel
10	R/W	0	gp0_dpll_wait_en

HHI_GP0_PLL_CNTL4 0x14

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	gp0_dpll_alpha
[26:24]	R/W	0	gp0_dpll_rou
[22:20]	R/W	0	gp0_dpll_lambda1
[18:16]	R/W	0	gp0_dpll_lambda0
[13:12]	R/W	0	gp0_dpll_acq_gain
[11:8]	R/W	0	gp0_dpll_filter_pvt2
[7:4]	R/W	0	gp0_dpll_filter_pvt1
[1:0]	R/W	0	gp0_dpll_pfd_gain

HHI_GP0_PLL_CNTL5 0x15

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	gp0_dpll_adj_vco_ldo
[27:24]	R/W	0	gp0_dpll_lm_w
[21:16]	R/W	0	gp0_dpll_lm_s
[15:0]	R/W	0	gp0_dpll_reve

HHI_GP0_PLL_CNTL6 0x16

Bit(s)	R/W	Default	Description
[31:30]	R/W	0	gp0_dpll_afc_hold_t
[29:28]	R/W	0	gp0_dpll_lkw_sel
[27:26]	R/W	0	gp0_dpll_dco_sdm_clk_sel
[25:24]	R/W	0	gp0_dpll_afc_in
[23:22]	R/W	0	gp0_dpll_afc_nt
[21:20]	R/W	0	gp0_dpll_vc_in
[19:18]	R/W	0	gp0_dpll_lock_long
[17:16]	R/W	0	gp0_dpll_freq_shift_v
[14:12]	R/W	0	gp0_dpll_data_sel
[10:8]	R/W	0	gp0_dpll_sdmnc_ulms
[6:0]	R/W	0	gp0_dpll_sdmnc_power

HHI_GP0_PLL_STS 0x17

Bit(s)	R/W	Default	Description
[31]	R	0	gp0_dpll_lock
[30]	R	0	gp0_dpll_lock_a
[29]	R	0	gp0_dpll_afc_done
[22:16]	R	0	gp0_dpll_sdmnc_monitor
[9:0]	R	0	gp0_dpll_out_rsv

HHI_PCIE_PLL_CNTL0 0x26

Bit(s)	R/W	Default	Description
[31]	R	0	pcie_apll_lock
[30]	R	0	pcie_hcsl_cal_done
[29]	R/W	0	pcie_apll_reset
[28]	R/W	0	pcie_apll_en
[27]	R/W	0	pcie_apll_vco_div_sel
[26]	R/W	0	pcie_apll_afc_start
[20:16]	R/W	0	pcie_apll_od
[14:10]	R/W	0	pcie_apll_prediv_sel
[7:0]	R/W	0	pcie_apll_fbkdiv

HHI_PCIE_PLL_CNTL1 0x27

Bit(s)	R/W	Default	Description
12	R/W	0	pcie_apll_sdm_en
[11:0]	R/W	0	pcie_apll_sdm_frac

HHI_PCIE_PLL_CNTL2 0x28

Bit(s)	R/W	Default	Description
[31:28]	R/W	0	pcie_apll_ssc_dep_sel
[25:24]	R/W	0	pcie_apll_ssc_fref_sel
[23:22]	R/W	0	pcie_apll_ssc_mode
[21:20]	R/W	0	pcie_apll_ssc_offset
[19:18]	R/W	0	pcie_apll_str_m
[15:0]	R/W	0	pcie_apll_reserve

HHI_PCIE_PLL_CNTL3 0x29

Bit(s)	R/W	Default	Description
[31]	R/W	0	pcie_apll_afc_bypass_en
[29:28]	R/W	0	pcie_apll_afc_hold_t
[26:20]	R/W	0	pcie_apll_afc_in
[19]	R/W	0	pcie_apll_afc_nt
[18:17]	R/W	0	pcie_apll_afc_div
[16]	R/W	0	pcie_apll_bias_lpf_en
[15:12]	R/W	0	pcie_apll_cp_icap
[11:8]	R/W	0	pcie_apll_cp_ires
[5:4]	R/W	0	pcie_apll_cpi

HHI_PCIE_PLL_CNTL4 0x2A

Bit(s)	R/W	Default	Description
[31]	R/W	0	pcie_apll_shift_en
[27:26]	R/W	0	pcie_apll_shift_t
[25:24]	R/W	0	pcie_apll_shift_v
[23]	R/W	0	pcie_apll_vctrl_mon_en
[21:20]	R/W	0	pcie_apll_lpf_cap
[19:16]	R/W	0	pcie_apll_lpf_capadj
[13:12]	R/W	0	pcie_apll_lpf_res
[11]	R/W	0	pcie_apll_lpf_sf
[10]	R/W	0	pcie_apll_lvr_od_en
[9]	R/W	0	pcie_apll_refclk_mon_en
[8]	R/W	0	pcie_apll_fbkclk_mon_en
[7]	R/W	0	pcie_apll_load
[6]	R/W	0	pcie_apll_load_en

HHI_PCIE_PLL_CNTL5 0x2B

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	pcie_hcsl_adj_ldo
[27]	R/W	0	pcie_hcsl_bgp_en

[24:20]	R/W	0	pcie_hcsl_bgr_adj
[19]	R/W	0	pcie_hcsl_bgr_start
[16:12]	R/W	0	pcie_hcsl_bgr_vref
[11:8]	R/W	0	pcie_hcsl_by_imp_in
[7]	R/W	0	pcie_hcsl_by_imp
[6]	R/W	0	pcie_hcsl_cal_en
[5]	R/W	0	pcie_hcsl_cal_rstn
[4]	R/W	0	pcie_hcsl_edgedrv_en
[3]	R/W	0	pcie_hcsl_en0
[2]	R/W	0	pcie_hcsl_in_en
[1]	R/W	0	pcie_hcsl_sel_pw
[0]	R/W	0	pcie_hcsl_sel_str

HHI_ PCIE _PLL_STS 0x2E

Bit(s)	R/W	Default	Description
[31]	R	0	pcie_apll_lock
[30]	R	0	pcie_hcsl_cal_done
[29]	R	0	pcie_apll_afc_done

HHI_XTAL_DIVN_CNTL 0x2f

Bit(s)	R/W	Default	Description
12	R/W	0	crt_clk25_en
11	R/W	0	crt_clk24_en
10	R/W	0	clk24_div2_en
7:0	R/W	0	clk25_div

HHI_GCLK2_MPEG0 0x30

Bit(s)	R/W	Default	Description
31-0	R/W	All 0	Bits [31:0] of the composite MPEG clock gating register

HHI_GCLK2_MPEG1 0x31

Bit(s)	R/W	Default	Description
31-0	R/W	All 0	Bits [63:32] of the composite MPEG clock gating register

HHI_GCLK2_MPEG2 0x32

Bit(s)	R/W	Default	Description
31-0	R/W	All 0	Bits [63:32] of the composite MPEG clock gating register

HHI_GCLK2_OTHER 0x34

Bit(s)	R/W	Default	Description
31-0	R/W	All 0	Bits [31:0] of the composite Other clock gating register

HHI_HIFI_PLL_CNTL0 0x36

Bit(s)	R/W	Default	Description
[31]	R	0	hifi_dpll_lock
[30]	R	0	hifi_dpll_lock_a
[29]	R/W	1	hifi_dpll_reset
[28]	R/W	0	hifi_dpll_en
[17:16]	R/W	0	hifi_dpll_od
[14:10]	R/W	0	hifi_dpll_ref_div_n
[8:0]	R/W	0	hifi_dpll_int_num

HHI_HIFI_PLL_CNTL1 0x37

Bit(s)	R/W	Default	Description
[18:0]	R	0	hifi_dpll_frac

HHI_HIFI_PLL_CNTL2 0x38

Bit(s)	R/W	Default	Description
[22:20]	R/W	0	hifi_dpll_fref_sel
[17:16]	R/W	0	hifi_dpll_os_ssc
[15:12]	R/W	0	hifi_dpll_ssc_str_m
[8]	R/W	0	hifi_dpll_ssc_en
[7:4]	R/W	0	hifi_dpll_ssc_dep_sel
[1:0]	R/W	0	hifi_dpll_ss_mode

HHI_HIFI_PLL_CNTL3 0x39

Bit(s)	R/W	Default	Description
31	R/W	0	hifi_dpll_afc_bypass
30	R/W	0	hifi_dpll_afc_clk_sel

Bit(s)	R/W	Default	Description
29	R/W	0	hifi_dpll_code_new
28	R/W	0	hifi_dpll_dco_m_en
27	R/W	0	hifi_dpll_dco_sdm_en
26	R/W	0	hifi_dpll_div2
25	R/W	0	hifi_dpll_div_mode
24	R/W	0	hifi_dpll_fast_lock
23	R/W	0	hifi_dpll_fb_pre_div
22	R/W	0	hifi_dpll_filter_mode
21	R/W	0	hifi_dpll_fix_en
20	R/W	0	hifi_dpll_freq_shift_en
19	R/W	0	hifi_dpll_load
18	R/W	0	hifi_dpll_load_en
17	R/W	0	hifi_dpll_lock_f
16	R/W	0	hifi_dpll_pulse_width_en
15	R/W	0	hifi_dpll_sdmnc_en
14	R/W	0	hifi_dpll_sdmnc_mode
13	R/W	0	hifi_dpll_sdmnc_range
12	R/W	0	hifi_dpll_tdc_en
11	R/W	0	hifi_dpll_tdc_mode_sel
10	R/W	0	hifi_dpll_wait_en

HHI_HIFI_PLL_CNTL4 0x3A

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	hifi_dpll_alpha
[26:24]	R/W	0	hifi_dpll_rou
[22:20]	R/W	0	hifi_dpll_lambda1
[18:16]	R/W	0	hifi_dpll_lambda0
[13:12]	R/W	0	hifi_dpll_acq_gain
[11:8]	R/W	0	hifi_dpll_filter_pvt2
[7:4]	R/W	0	hifi_dpll_filter_pvt1

[1:0]	R/W	0	hifi_dpll_pfd_gain

HHI_HIFI_PLL_CNTL5 0x3B

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	hifi_dpll_adj_vco_ldo
[27:24]	R/W	0	hifi_dpll_lm_w
[21:16]	R/W	0	hifi_dpll_lm_s
[15:0]	R/W	0	hifi_dpll_reve

HHI_HIFI_PLL_CNTL6 0x3C

Bit(s)	R/W	Default	Description
[31:30]	R/W	0	hifi_dpll_afc_hold_t
[29:28]	R/W	0	hifi_dpll_lkw_sel
[27:26]	R/W	0	hifi_dpll_dco_sdm_clk_sel
[25:24]	R/W	0	hifi_dpll_afc_in
[23:22]	R/W	0	hifi_dpll_afc_nt
[21:20]	R/W	0	hifi_dpll_vc_in
[19:18]	R/W	0	hifi_dpll_lock_long
[17:16]	R/W	0	hifi_dpll_freq_shift_v
[14:12]	R/W	0	hifi_dpll_data_sel
[10:8]	R/W	0	hifi_dpll_sdmnc_ulms
[6:0]	R/W	0	hifi_dpll_sdmnc_power

HHI_HIFI_PLL_STS 0x3D

Bit(s)	R/W	Default	Description
[31]	R	0	hifi_dpll_lock
[30]	R	0	hifi_dpll_lock_a
[29]	R	0	hifi_dpll_afc_done
[22:16]	R	0	hifi_dpll_sdmnc_monitor
[9:0]	R	0	hifi_dpll_out_rsv

HHI_TIMER90K 0x3F

Bit(s)	R/W	Default	Description
31-16	R/W	0	Unused

15-0	R/W	0x384	90khz divider

HHI_MEM_PD_REG0 0x40

Bit(s)	R/W	Default	Description
31~22	R/W	0xFFFF	Reserved
21~20	R/W	0x3	Axi srame memory PD
19~18	R/W	0x3	Apical GDC memory PD
15~8	R/W	0xFF	HDMI memory PD
5~4	R/W	0xF	audio memory PD
3~2	R/W	0x3	Ethernet memory PD
1~0	R/W	0x3	Reserved

HHI_VPU_MEM_PD_REG0 0x41

Bit(s)	R/W	Default	Description
31~30	R/W	0x3	sharp
29~28	R/W	0x3	Deinterlacer – di_post: 11 = power down. 00 = normal operation
27~26	R/W	0x3	Deinterlacer – di_pre
25~24	R/W	0x3	Vi_di_scaler
23~22	R/W	0x3	afbc_dec1
21~20	R/W	0x3	Srscl super scaler
19~18	R/W	0x3	Vdin1 memory
17~16	R/W	0x3	Vdin0 memory
15~14	R/W	0x3	Osd_scaler memory
13~12	R/W	0x3	Scaler memory
11~10	R/W	0x3	Vpp output fifo
9~8	R/W	0x3	Color management module
7~6	R/W	0x3	Vd2 memory
5~4	R/W	0x3	Vd1 memory
3~2	R/W	0x3	Osd2 memory
1~0	R/W	0x3	Osd1 memory

HHI_VPU_MEM_PD_REG1 0x42

Bit(s)	R/W	Default	Description
31~30	R/W	0x3	atv demod / xvycc_lut
29~28	R/W	0x3	tv decoder cvd2 / ldim_stts
27~26	R/W	0x3	
25~24	R/W	0x3	CVBS inci interface
23~22	R/W	0x3	Panel encl top
21~20	R/W	0x3	Hdmi encp interface
19~18	R/W	0x3	osd afbcd
17~16	R/W	0x3	Afbc dec0
15~14	R/W	0x3	Vpu arb
13~12	R/W	0x3	dolby1b
11~10	R/W	0x3	dolby1a
9~8	R/W	0x3	dolby0
7~6	R/W	0x3	dolby core3
5~4	R/W	0x3	vkstone
3~2	R/W	0x3	VIU2_OFIFO
1~0	R/W	0x3	VIU2 OSD1

HHI_VIID_CLK_DIV 0x4a

Bit(s)	R/W	Default	Description
31-28	R/W	0	DAC0_CLK_SEL
27-24	R/W	0	DAC1_CLK_SEL
23-20	R/W	0	DAC2_CLK_SEL
19	R/W	0	Select adc_pll_clk_b2 to be cts_clk_vdac
18	R/W	0	Unused
17	R/W	0	V2_cntl_clk_div_reset
16	R/W	0	V2_cntl_clk_div_en
15-12	R/W	0	Encl_clk_sel
14-8	R/W	0	Unused
7-0	R/W	0	V2_cntl_xd0

HHI_VIID_CLK_CNTL 0x4b

Bit(s)	R/W	Default	Description
31-20	R/W	0	Unused
19	R/W	0	V2_cntl_clk_en0
18-16	R/W	0	V2_cntl_clk_in_sel
15	R/W	0	V2_cntl_soft_reset
14-5	R/W	0	Unused
4	R/W	0	V2_cntl_div12_en
3	R/W	0	V2_cntl_div6_en
2	R/W	0	V2_cntl_div4_en
1	R/W	0	V2_cntl_div2_en
0	R/W	0	V2_cntl_div1_en

HHI_VPU_MEM_PD_REG2 0x4d

Bit(s)	R/W	Default	Description
31-18	R/W	3	Unused
17-16	R/W	3	Vd2_ofifo
15-14	R/W	3	Prime_dolby_ram
13-12	R/W	3	Osd_bld34
11-10	R/W	3	Vd1_scaler
9-8	R/W	3	Mem_pd_mail_afbcd
7-6	R/W	3	Mem_pd_vi_osd4
5-4	R/W	3	Mem_pd_vi_osd3
3-2	R/W	3	Mem_pd_tcon
1-0	R/W	3	vpp water mark

HHI_GCLK_LOCK 0x4F

Write	Read
SP/SCP sec only	all access

Bit(s)	R/W	Default	Description
31	R/W	0	enable error return if write gclk after lock
8	R/W	0	lock gclk2_other
7	R/W	0	lock gclk2_mpeg2
6	R/W	0	lock gclk2_mpeg1
5	R/W	0	lock gclk2_mpeg0
4	R/W	0	lock gclk_sp_mpeg
3	R/W	0	lock gclk1_other
2	R/W	0	lock gclk1_mpeg2
1	R/W	0	lock gclk1_mpeg1
0	R/W	0	lock gclk1_mpeg0

HHI_GCLK_MPEG0 0x50

Bit(s)	R/W	Default	Description
31-0	R/W	All 1	Bits [31:0] of the composite MPEG clock gating register

HHI_GCLK_MPEG1 0x51

Bit(s)	R/W	Default	Description
31-0	R/W	All 1	Bits [63:32] of the composite MPEG clock gating register

HHI_GCLK_MPEG2 0x52

Bit(s)	R/W	Default	Description
31-0	R/W	All 1	Bits [63:32] of the composite MPEG clock gating register

HHI_GCLK_OTHER 0x54

Bit(s)	R/W	Default	Description
31-0	R/W	All 1	Bits [31:0] of the composite Other clock gating register

HHI_GCLK_SP_MPEG 0x55

Write	Read
SP sec only	SP sec only

Bit(s)	R/W	Default	Description
31-0	R/W	0	bits[31:0] of secure module clock gating.

HHI_SYS_CPU_CLK_CNTL1 0x57

Bit(s)	R/W	Default	Description
31-25	R/W	0	Reserved
24	R/W	0	Sys_pll_div16_en
23	R/W	0	A53_trace_clk _DIS: Set to 1 to manually disable the A53_trace_clk when changing the mux selection. Typically this bit is set to 0 since the clock muxes can switch without glithces. This is a “just in case” bit
22:20	R/W	5	A53_trace_clk: A53 clock divided by 2 A53 clock divided by 3 A53 clock divided by 4 A53 clock divided by 5 A53 clock divided by 6 A53 clock divided by 7 A53 clock divided by 8
19	R/W	0	Timestamp CNTCLKEN_dis
18	R/W	0	AXI_CLK_DIS: Set to 1 to manually disable the AXI clock when changing the mux selection. Typically this bit is set to 0 since the clock muxes can switch without glithces. This is a “just in case” bit
17	R/W	0	ATCLK_dis
16	R/W	0	APB_CLK_DIS: Set to 1 to manually disable the APB clock when changing the mux selection. Typically this bit is set to 0 since the clock muxes can switch without glithces. This is a “just in case” bit
15	R/W	0	Timestamp CNTCLKEN
14~12	R/W	0	Timestamp cntclk mux(not used)
11~9	R/W	1	AXI_CLK _MUX: A53 clock divided by 2 A53 clock divided by 3 A53 clock divided by 4 A53 clock divided by 5 A53 clock divided by 6 A53 clock divided by 7 A53 clock divided by 8
8~6	R/W	2	atCLK _MUX: A53 clock divided by 2 A53 clock divided by 3 A53 clock divided by 4 A53 clock divided by 5 A53 clock divided by 6

Bit(s)	R/W	Default	Description
			A53 clock divided by 7 A53 clock divided by 8
5~3	R/W	4	APB_CLK _MUX: A53 clock divided by 2 A53 clock divided by 3 A53 clock divided by 4 A53 clock divided by 5 A53 clock divided by 6 A53 clock divided by 7 A53 clock divided by 8
2	R/W	0	Soft_reset
1	R/W	0	Sys_cpu_clk_div16_en
0	R/W	0	Pclk_en_dbg

HHI_SYS_CPU_RESET_CNTL 0x58

Write	Read
SP/SCP/AP sec only	all access

Bit(s)	R/W	Default	Description
10	R/W	0	Cpu_axi_reset
9	R/W	0	nPRESETDBG
8	R/W	0	nL2RESET
7-4	R/W	0	nCORERESET[3:0]
3-0	R/W	0	Cpu_soft_reset[3:0]

HHI_VID_CLK_DIV 0x59

Bit(s)	R/W	Default	Description
31-28	R/W	0	ENCI_CLK_SEL
27-24	R/W	0	ENCP_CLK_SEL
23-20	R/W	0	ENCT_CLK_SEL
19-18	R/W	0	UNUSED
17	R/W	0	CLK_DIV_RESET
16	R/W	0	CLK_DIV_EN

Bit(s)	R/W	Default	Description
15-8	R/W	1	XD1
7-0	R/W	1	XD0

HHI_APICALGDC_CNTL 0x5a

Bit(s)	R/W	Default	Description
27:25	R/W	0	Cts_gds_axi_clk:clk div
24	R/W	0	Cts_gds_axi_clk:clk_en
22:16	R/W	0	Cts_gds_axi_clk:clk_sel
11:9	R/W	0	Cts_gds_core_clk:clk div
8	R/W	0	Cts_gds_core_clk:clk_en
6:0	R/W	0	Cts_gds_core_clk:clk_sel

HHI_MPEG_CLK_CNTL 0x5d

Bit(s)	R/W	Default	Description
31	R/W	0	NEW_DIV_EN: If this bit is set to 1, then bits[30:16] make up the clk81 divider. If this bit is 0, then bits[6:0] dictate the divider value. This is a new feature that allows clk81 to be divided down to a very slow frequency.
30~16	R/W	0	NEW_DIV: New divider value if bit[31] = 1
15	R/W	0	Production clock enable
14-12	R.W	6	MPEG_CLK_SEL (See clock document)
11-10	R/W	0	unused
9	R.W	0	RTC Oscillator Enable: Set this bit to 1 to connect the RTC 32khz oscillator output as the XTAL input for the divider above
8	R/W	0	Divider Mux: 0 = the ARC clock and the MPEG system clock are connected to the 27Mhz crystal. 1 = the ARC clock and the MPEG system clock are connected to the MPEG PLL divider
7	R/W	1	PLL Mux: 0 = all circuits associated with the MPEG PLL are connected to 27Mhz. 1 = all circuits associated with the MPEG PLL are connected to the MPEG PLL
6-0	R/W	0	PLL Output divider. The MPEG System clock equals the video PLL clock frequency divided by (N+1). Note: N must be odd (1,3,5,….) so that the MPEG clock is divided by an even number to generate a 50% duty cycle.

HHI_VID_CLK_CNTL0x5f

Bit(s)	R/W	Default	Description
31-21	R/W	0	TCON_CLK0_CTRL

Bit(s)	R/W	Default	Description
20	R/W	0	CLK_EN1
19	R/W	0	CLK_EN0
18-16	R/W	0	CLK_IN_SEL
15	R/W	0	SOFT_RESET
14	R/W	0	PH23_ENABLE
13	R/W	0	DIV12_PH23
12-5	R/W	0	UNUSED
4	R/W	0	DIV12_EN
3	R/W	0	DIV6_EN
2	R/W	0	DIV4_EN
1	R/W	0	DIV2_EN
0	R/W	0	DIV1_EN

HHI_TS_CLK_CNTL 0x64

Bit(s)	R/W	Default	Description
8	R/W	0	ts_clk clk_en
7:0	R/W	0	ts_clk clk_div

HHI_VID_CLK_CNTL2 0x65

Bit(s)	R/W	Default	Description
31-16	R	0	
15-9	R/W	0	Reserved
8	R/W	0	Atv demod vdac gated clock control
7	R/W	1	LCD_AN_CLK_PHY2 gated clock control. 1 = enable
6	R/W	1	LCD_AN_CLK_PH3 gated clock control
5	R/W	1	HDMI_TX_PIXEL_CLK gated clock control
4	R/W	1	VDAC_clk gated clock control
3	R/W	1	ENCL gated clock control
2	R/W	1	ENCP gated clock control
1	R/W	1	ENCT gated clock control
0	R/W	1	ENCI gated clock control

HHI_SYS_CPU_CLK_CNTL0 0x67

Bit(s)	R/W	Default	Description
31	R	0	Final_mux_sel
30	R	0	Final_dyn_mux_sel
29	R	0	Busy_cnt
28	R	0	busy
26	R/W	0	Dyn_enable
25-20	R/W	0	Mux1_divn_tcnt
18	R/W	0	Postmux1
17-16	R/W	0	Premux1
15	R/W	0	Manual_mux_mode
14	R/W	0	Manual_mode_post
13	R/W	0	Manual_mode_pre
12	R/W	0	Force_update_t
11	R/W	0	Final_mux_sel
10	R/W	0	Final_dyn_mux_sel
9-4	R/W	0	mux0_divn_tcnt
3	R/W	0	Rev
2	R/W	0	Postmux0
1-0	R/W	0	Premux0

HHI_VID_PLL_CLK_DIV 0x68

Bit(s)	R/W	Default	Description
31~24	R	0	RESERVED
23~20	R/W	0	Reserved
19	R/W	0	CLK_FINAL_EN
18	R/W	0	CLK_DIV1
17~16	R/W	0	CLK_SEL
15	R/W	0	SET_PRESET
14-0	R/W	0	SHIFT_PRESET

HHI_CCI_CLK_CNTL 0x6B

Bit(s)	R/W	Default	Description
31	R/W	0	Cts_cci_clk :Final_mux_sel
27:25	R/W	0	Cts_cci_clk Mux1_sel: 0:oscin; 1:gp0pll; 2:hifi_pll; 3:fclk_div2p5; 4:fclk_div3; 5:fclk_div4; 6:fclk_div5; 7:fclk_div7;
24	R/W	0	Cts_cci_clk Mux1_en
22:16	R/W	0	Cts_cci_clk Mux1_div
11:9	R/W	0	Cts_cci_clk Mux0_sel: 0:oscin; 1:gp0pll; 2:hifi_pll; 3:fclk_div2p5; 4:fclk_div3; 5:fclk_div4; 6:fclk_div5; 7:fclk_div7;
8	R/W	0	Cts_cci_clk Mux0_en
6:0	R/W	0	Cts_cci_clk Mux0_div

HHI_DVALIN_CLK_CNTL0x6c

Bit(s)	R/W	Default	Description
31	R/W	0	0: Dvalin_clk_cntl[14:0]; 1: Dvalin_clk_cntl[30:16];
27~25	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_Dvalin _clk 0:oscin; 1:gp0pll; 2:hifi_pll; 3:fclk_div2p5; 4:fclk_div3; 5:fclk_div4;

Bit(s)	R/W	Default	Description
			6:fclk_div5; 7:fclk_div7;
24	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_ Dvalin _clk
23	R/W	0	Reserved
22~16	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_ Dvalin _clk
15~12			
11~9	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_Dvalin _clk 0:oscin; 1:gp0pll; 2:mpll2; 3:mpll1; 4:fclk_div7; 5:fclk_div4; 6:fclk_div3; 7:fclk_div5;
8	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_ Dvalin _clk
7	R/W	0	Reserved
6-0	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_ Dvalin _clk

HHI_VPU_CLKC_CNTL 0x6D

Bit(s)	R/W	Default	Description
31	R/W	0	Final mux sel
30-29	R/W	0	Reserved
27~25	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_vpu_clkc 0:fclk_div4; 1:fclk_div3; 2:fclk_div5; 3:fclk_div7; 4:mpll1; 5:vid_pll; 6:mpll2; 7:gp0pll;
24	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_vpu_clkc
23	R/W	0	Reserved

Bit(s)	R/W	Default	Description
22~16	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_vpu_clkc
15~12			
11~9	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_vpu_clkc 0:fclk_div4; 1:fclk_div3; 2:fclk_div5; 3:fclk_div7; 4:mpll1; 5:vid_pll; 6:mpll2; 7:gp0pll;
8	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_vpu_clkc
7	R/W	0	Reserved
6-0	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_vpu_clkc

HHI_VPU_CLK_CNTL 0x6F

Bit(s)	R/W	Default	Description
31	R/W	0	Final mux sel
30-29	R/W	0	Reserved
27~25	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_vpu_clk 0:fclk_div3; 1:fclk_div4; 2:fclk_div5; 3:fclk_div7; 4:mpll1; 5:vid_pll; 6:hifi_pll; 7:gp0_pll;
24	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_vpu_clk
23	R/W	0	Reserved
22~16	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_vpu_clk
15~12			
11~9	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_vpu_clk 0:fclk_div3; 1:fclk_div4;

Bit(s)	R/W	Default	Description
			2:fclk_div5; 3:fclk_div7; 4:mpll1; 5:vid_pll; 6:hifi_pll; 7:gp0_pll;
8	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_vpu_clk
7	R/W	0	Reserved
6-0	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_vpu_clk

HHI_HDMI_CLK_CNTL 0x73

Bit(s)	R/W	Default	Description
31-20	R/W	0	Reserved
19~16	R/W	0	crt_hdmi_pixel_clk_sel
15~11	R/W	0	Reserved
10~9	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_hdmi_sys_clk 0:oscin; 1:fclk_div4; 2:fclk_div3; 3:fclk_div5;
8	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_hdmi_sys_clk
7	R/W	0	Reserved
6-0	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_hdmi_sys_clk

HHI_ETH_CLK_CNTL 0x76

Bit(s)	R/W	Default	Description
17	R/W	0	eth_mac_speed; 0:div 20; 1: div2;
16	R/W	0	invert clk_rmii_pad_i
13	R/W	0	invert clk_rmii_pad_o
12	R/W	0	clk_rmii_pad_o; 0: rmii_clk; 1: rmii_div;
11:9	R/W	0	rmii_clk; clk_sel; 0: fclk_div2; 7: clk_rmii_pad_i;

Bit(s)	R/W	Default	Description
8	R/W	0	rmii_clk; clk_en
7	R/W	0	eth_clk125M; clk_en;
6:0	R/W	0	rmii_clk; clk_div

HHI_VDEC_CLK_CNTL 0x78

Bit(s)	R/W	Default	Description
31-28	R/W	0	Reserved
27~25	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_hcodec_clk 0:fclk_div2p5; 1:fclk_div3; 2:fclk_div4; 3:fclk_div5; 4:fclk_div7; 5:hifi_pll; 6:gp0pll; 7:oscin;
24	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_hcodec_clk
23	R/W	0	Reserved
22~16	R/W	0	CLK_DIV: See the Clock Tree document for information related to ctshcodec_clk
15~12	R/W	0	Reserved
11~9	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_vdec_clk 0:fclk_div2p5; 1:fclk_div3; 2:fclk_div4; 3:fclk_div5; 4:fclk_div7; 5:hifi_pll; 6:gp0pll; 7:oscin;
8	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_vdec_clk
7	R/W	0	Reserved
6-0	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_vdec_clk

HHI_VDEC2_CLK_CNTL 0x79

Bit(s)	R/W	Default	Description
31-28	R/W	0	Reserved
27~25	R/W	0	HEVCB_CLK_SEL: See the Clock Tree document for information related to cts_hevcb_clk 0:fclk_div2p5; 1:fclk_div3; 2:fclk_div4; 3:fclk_div5; 4:fclk_div7; 5:hifi_pll; 6:gp0pll; 7:oscin;
24	R/W	0	HEVCB_CLK_EN: See the Clock Tree document for information related to cts_hevcb_clk
23	R/W	0	Reserved
22~16	R/W	0	HEVCB_CLK_DIV: See the Clock Tree document for information related to cts_hevcb_clk
15~12			
11~9	R/W	0	HEVCF_CLK_SEL: See the Clock Tree document for information related to cts_hevcf_clk 0:fclk_div2p5; 1:fclk_div3; 2:fclk_div4; 3:fclk_div5; 4:fclk_div7; 5:hifi_pll; 6:gp0pll; 7:oscin;
8	R/W	0	HEVCF_CLK_EN: See the Clock Tree document for information related to cts_hevcf_clk
7	R/W	0	Reserved
6-0	R/W	0	HEVCF_CLK_DIV: See the Clock Tree document for information related to cts_hevcf_clk

HHI_VDEC3_CLK_CNTL 0x7a

Bit(s)	R/W	Default	Description
31	R/W	0	cts_hcodec_clk; 0: use VDEC_CLK_CNTL; 1: use VDEC3_CLK_CNTL;
27~25	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_hcodec_clk 0:fclk_div2p5; 1:fclk_div3;

Bit(s)	R/W	Default	Description
			2:fclk_div4; 3:fclk_div5; 4:fclk_div7; 5:hifi_pll; 6:gp0pll; 7:oscin;
24	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_hcodec_clk
23	R/W	0	Reserved
22~16	R/W	0	CLK_DIV: See the Clock Tree document for information related to ctshcodec_clk
15	R/W	0	cts_vdec_clk; 0: use VDEC_CLK_CNTL; 1: use VDEC3_CLK_CNTL;
11~9	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_vdec_clk 0:fclk_div2p5; 1:fclk_div3; 2:fclk_div4; 3:fclk_div5; 4:fclk_div7; 5:hifi_pll; 6:gp0pll; 7:oscin;
8	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_vdec_clk
7	R/W	0	Reserved
6-0	R/W	0	CLK_DIV: See the Clock Tree document for information related to cts_vdec_clk

HHI_VDEC4_CLK_CNTL 0x7b

Bit(s)	R/W	Default	Description
31	R/W	0	HEVCB_CLK; 0: use VDEC2_CLK_CNTL; 1: use VDEC4_CLK_CNTL;
27~25	R/W	0	HEVCB_CLK_SEL: See the Clock Tree document for information related to cts_hevcb_clk 0:fclk_div2p5; 1:fclk_div3; 2:fclk_div4; 3:fclk_div5; 4:fclk_div7; 5:hifi_pll;

Bit(s)	R/W	Default	Description
			6:gp0pll; 7:oscin;
24	R/W	0	HEVCB_CLK_EN: See the Clock Tree document for information related to cts_hevcb_clk
23	R/W	0	Reserved
22~16	R/W	0	HEVCB_CLK_DIV: See the Clock Tree document for information related to cts_hevcb_clk
15	R/W	0	HEVCF_CLK; 0: use VDEC2_CLK_CNTL; 1: use VDEC4_CLK_CNTL;
11~9	R/W	0	HEVCF_CLK_SEL: See the Clock Tree document for information related to cts_hevcf_clk 0:fclk_div2p5; 1:fclk_div3; 2:fclk_div4; 3:fclk_div5; 4:fclk_div7; 5:hifi_pll; 6:gp0pll; 7:oscin;
8	R/W	0	HEVCF_CLK_EN: See the Clock Tree document for information related to cts_hevcf_clk
7	R/W	0	Reserved
6-0	R/W	0	HEVCF_CLK_DIV: See the Clock Tree document for information related to cts_hevcf_clk

HHI_HDCP22_CLK_CNTL 0x7c

Bit(s)	R/W	Default	Description
31-0	R/W	0	Reserved
26-25	R/W	0	Clk_sel: 0:cts_oscin_clk 1:fclk_div4 2:fclk_div3 3:fclk_div5
24	R/W	0	Clk_en
23-0	R/W	0	Clk_div

HHI_VAPBCLK_CNTL 0x7d

Bit(s)	R/W	Default	Description
31	R/W	0	Final_mux_sel

Bit(s)	R/W	Default	Description
30	R/W	0	Enable
29-28	R/W	0	Reserved
27-25	R/W	0	Mux1_sel: 0:fclk_div4 1:fclk_div3 2:fclk_div5 3:fclk_div7 4:mp1_clk_out 5:vid_pll_clk 6:mp2_clk_out 7:fclk_div2p5
24	R/W	0	Mux1_en
23	R/W	0	Reserved
22-16	R/W	0	Mux1_div
15-12	R/W	0	Reserved
11-9	R/W	0	Mux0_sel,as mux1_sel
8	R/W	0	Mux0_en
7	R/W	0	Reserved
6-0	R/W	0	Mux0_div

HHI_SYS_CPUB_CLK_CNTL1 0x80

Bit(s)	R/W	Default	Description
24	R/W	0	Sys_pll_div16_en
23	R/W	0	Trace_clk_dis
22:20	R/W	0x5	Trace_clk_mux
19	R/W	0	CNTCLKEN_clk_dis
18	R/W	0	ACLKM_clk_dis
17	R/W	0	ATCLK_clk_dis
16	R/W	0	PCLK_dis
15	R/W	0	CNTCLKEN
14:12	R/W	0	CNTCLKEN_clk_mux
11:9	R/W	0x1	ACLKM_clk_mux

Bit(s)	R/W	Default	Description
8:6	R/W	0x2	ATCLK_clk_mux
5:3	R/W	0x4	PCLK_mux
2	R/W	0	Soft_reset
1	R/W	0	Sys_clk_div16_en
0	R/W	0	PCLKENDBG

HHI_SYS_CPUB_RESET_CNTL 0x81

Bit(s)	R/W	Default	Description
10	R/W	0	Cpu_axi_reset
9	R/W	0	nPRESETDBG
8	R/W	0	nL2RESET
7-4	R/W	0	nCORERESET[3:0]
3-0	R/W	0	Cpu_soft_reset[3:0]

HHI_SYS_CPUB_CLK_CNTL 0x82

Bit(s)	R/W	Default	Description
31	R	0	Final_mux_sel
30	R	0	Final_dyn_mux_sel
29	R	0	Busy_cnt
28	R	0	busy
26	R/W	0	Dyn_enable
25-20	R/W	0	Mux1_divn_tcnt
18	R/W	0	Postmux1
17-16	R/W	0	Premux1
15	R/W	0	Manual_mux_mode
14	R/W	0	Manual_mode_post
13	R/W	0	Manual_mode_pre
12	R/W	0	Force_update_t
11	R/W	0	Final_mux_sel
10	R/W	0	Final_dyn_mux_sel
9-4	R/W	0	mux0_divn_tcnt

Bit(s)	R/W	Default	Description
3	R/W	0	Rev
2	R/W	0	Postmux0
1-0	R/W	0	Premux0

HHI_VPU_CLKB_CNTL 0x83

Bit(s)	R/W	Default	Description
31-25	R/W	0	Reserved
24	R/W	0	Cts_vpu_clkb_tmp en
21-20	R/W	0	Cts_vpu_clkb_tmp sel 0:cts_vpu_clk 1:fclk_div4 2:fclk_div5 3:fclk_div7
19-16	R/W	0	Cts_vpu_clkb_tmp div
9	R/W	0	Vpu_clkb_latch_en
8	R/W	0	Vpu_clkb_en
7-0	R/W	0	Vpu_clkb_div

HHI_GEN_CLK_CNTL 0x8a

Bit(s)	R/W	Default	Description
31-17	R/W	0	Reserved
16~12	R/W	0	CLK_SEL: 0:cts_oscin_clk 1:rtc_oscin_i 2:sys_cpu_clk_div16 3:ddr_dpll_pt_clk 4:vid_pll_clk 5:gp0_pll_clk 7:hifi_pll_clk 8:pcie_clk_in_n 9:pcie_clk_in_p 12:cts_msr_clk 14:sys_pllB_div16 15:sys_cpuB_clk_div16 16:acodec_dac_clk 17:sys_cpu_clk_div16;

Bit(s)	R/W	Default	Description
			19:fclk_div2 20:fclk_div2p5; 20:fclk_div2; 21:fclk_div3; 22:fclk_div4; 23:fclk_div5; 24:fclk_div7; 25:mpll0_clk; 26:mpll1_clk; 27:mpll2_clk; 28:mpll3_clk; 29:Reserved 30:Reserved
11	R/W	0	CLK_EN: See the Clock Tree document for information related to gen_clk_out
10~0	R/W	0	CLK_DIV: See the Clock Tree document for information related to gen_clk_out

HHI_VDIN_MEAS_CLK_CNTL 0x94

Clock control for cts_vdin_meas_clk

Bit(s)	R/W	Default	Description
31-24	R/W	0	unused
23-21	R/W	0	Cts_dsi_meas_clk :clk_sel 0:oscin; 1:fclk_div4; 2:fclk_div3; 3:fclk_div5; 4:vid_pll; 5:gp0_pll; 6:fclk_div2; 7:fclk_div7;
20	R/W	0	Cts_dsi_meas_clk: clk_en
18:12	R/W	0	Cts_dsi_meas_clk: clk_div
11-9	R/W	0	CLK_SEL: See the Clock Tree document for information related to cts_vdin_meas_clk 0:oscin; 1:fclk_div4; 2:fclk_div3; 3:fclk_div5; 4:vid_pll; 5:gp0_pll;

Bit(s)	R/W	Default	Description
			6:fclk_div2; 7:fclk_div7;
8	R/W	0	CLK_EN: See the Clock Tree document for information related to cts_vdin_meas_clk
6-0	R/W	48	CLK_DIV: See the Clock Tree document for information related to cts_vdin_meas_clk

HHI_MIPIDSI_PHY_CLK_CNTL 0x95

Bit(s)	R/W	Default	Description
14:12	R/W	0	mipi_dsi_phy_clk clk_sel; 0:vid_pll; 1:gp0_pll; 2:hifi_pll; 3:mpll1; 4:fclk_div2; 5:fclk_div2p5; 6:fclk_div3; 7:fclk_div7;
8	R/W	0	mipi_dsi_phy_clk clk_en
6:0	R/W	0	mipi_dsi_phy_clk clk_div

HHI_NAND_CLK_CNTL 0x97

Bit(s)	R/W	Default	Description
31-12	R/W	0	unused
11-9	R/W	0	CLK_SEL: 0:cts_oscin_clk 1:fclk_div2 2:fclk_div3 3:fclk_div5 4:fclk_div7 5:mp2_clk_out 6:mp3_clk_out 7:gp0_pll_clk
8			Reserved
7	R/W	1	CLK_EN:
6-0	R/W	0	CLK_DIV

HHI_SD_EMMC_CLK_CNTL 0x99

Bit(s)	R/W	Default	Description
31-28	R/W	0	unused
27-25	R/W	0	Sd_emmc_B_CLK_SEL: 0:cts_oscin_clk 1:fclk_div2 2:fclk_div3 3:fclk_div5 4:fclk_div7 5:mp2_clk_out 6:mp3_clk_out 7:gp0_pll_clk
24			Reserved
23	R/W	1	Sd_emmc_B_CLK_EN:
22-16	R/W	0	Sd_emmc_B_CLK_DIV
15-12	R/W	0	reseverd
11-9	R/W	0	Sd_emmc_A_CLK_SEL: 0:cts_oscin_clk 1:fclk_div2 2:fclk_div3 3:fclk_div5 4:fclk_div7 5:mp2_clk_out 6:mp3_clk_out 7:gp0_pll_clk
8			Reserved
7	R/W	1	Sd_emmc_A_CLK_EN:
6-0	R/W	0	Sd_emmc_A_CLK_DIV

HHI_HEVC Encoder_CLK_CNTL 0x9A

Bit(s)	R/W	Default	Description
27:25	R/W	0	HEVC Encoder_cclk clk_sel; 0:oscin; 1:fclk_div4; 2:fclk_div3; 3:fclk_div5; 4:fclk_div7;

Bit(s)	R/W	Default	Description
			5:mpll2; 6:mpll3; 7:gp0pll;
24	R/W	0	HEVC Encoder_cclk clk_en
22:16	R/W	0	HEVC Encoder_cclk clk_div
11:9	R/W	0	HEVC Encoder_bclk clk_sel; 0:oscin; 1:fclk_div4; 2:fclk_div3; 3:fclk_div5; 4:fclk_div7; 5:mpll2; 6:mpll3; 7:gp0pll;
8	R/W	0	HEVC Encoder_bclk clk_en
6:0	R/W	0	HEVC Encoder_bclk clk_div

HHI_HEVC Encoder_CLK_CNTL2 0x9B

Bit(s)	R/W	Default	Description
11:9	R/W	0	HEVC Encoder_aclk clk_sel; 0:oscin; 1:fclk_div4; 2:fclk_div3; 3:fclk_div5; 4:fclk_div7; 5:mpll2; 6:mpll3; 7:gp0pll;
8	R/W	0	HEVC Encoder _aclk clk_en
6:0	R/W	0	HEVC Encoder_aclk clk_div

HHI_MPLL_CNTL0 0x9E

Bit(s)	R/W	Default	Description
[21:20]	R/W	0	mpll_sys_dpll_exldo
[18:16]	R/W	0	mpll_test_c
[15:12]	R/W	0	mpll_dds_reve

[11:10]	R/W	0	mpll_dds_vc_vdd
[9:8]	R/W	0	mpll_dds_vr_fb1
[7:6]	R/W	0	mpll_dds_vr_fb2
[1]	R/W	0	mpll_dds_enldo
[0]	R/W	0	mpll_dds_ldo_rupsel

HHI_MPLL_CNTL1 0x9F

Bit(s)	R/W	Default	Description
[31]	R/W	0	mpll_dds0_en
[30]	R/W	0	mpll_dds0_sdm_en
[29]	R/W	0	mpll_dds0_ss_en
[28:20]	R/W	0	mpll_dds0_n_in
[13:0]	R/W	0	mpll_dds0_sdm_in

HHI_MPLL_CNTL2 0xA0

Bit(s)	R/W	Default	Description
[31]	R/W	0	mpll_dds0_ir_bypass
[30]	R/W	0	mpll_dds0_load
[29]	R/W	0	mpll_dds0_lp_en
[28]	R/W	0	mpll_dds0_modesel
[20:16]	R/W	0	mpll_dds0_f_set
[15:12]	R/W	0	mpll_dds0_ir_byin
[10:8]	R/W	0	mpll_dds0_p_set
[7:4]	R/W	0	mpll_dds0_vref_cf
[3:0]	R/W	0	mpll_dds0_vref_cs

HHI_MPLL_CNTL3 0xA1

Bit(s)	R/W	Default	Description
[31]	R/W	0	mpll_dds1_en
[30]	R/W	0	mpll_dds1_sdm_en
[29]	R/W	0	mpll_dds1_ss_en
[28:20]	R/W	0	mpll_dds1_n_in

[13:0]	R/W	0	mpll_dds1_sdm_in

HHI_MPLL_CNTL4 0xA2

Bit(s)	R/W	Default	Description
[31]	R/W	0	mpll_dds1_ir_bypass
[30]	R/W	0	mpll_dds1_load
[29]	R/W	0	mpll_dds1_lp_en
[28]	R/W	0	mpll_dds1_modesel
[20:16]	R/W	0	mpll_dds1_f_set
[15:12]	R/W	0	mpll_dds1_ir_byin
[10:8]	R/W	0	mpll_dds1_p_set
[7:4]	R/W	0	mpll_dds1_vref_cf
[3:0]	R/W	0	mpll_dds1_vref_cs

HHI_MPLL_CNTL5 0xA3

Bit(s)	R/W	Default	Description
[31]	R/W	0	mpll_dds2_en
[30]	R/W	0	mpll_dds2_sdm_en
[29]	R/W	0	mpll_dds2_ss_en
[28:20]	R/W	0	mpll_dds2_n_in
[13:0]	R/W	0	mpll_dds2_sdm_in

HHI_MPLL_CNTL6 0xA4

Bit(s)	R/W	Default	Description
[31]	R/W	0	mpll_dds2_ir_bypass
[30]	R/W	0	mpll_dds2_load
[29]	R/W	0	mpll_dds2_lp_en
[28]	R/W	0	mpll_dds2_modesel
[20:16]	R/W	0	mpll_dds2_f_set
[15:12]	R/W	0	mpll_dds2_ir_byin
[10:8]	R/W	0	mpll_dds2_p_set
[7:4]	R/W	0	mpll_dds2_vref_cf
[3:0]	R/W	0	mpll_dds2_vref_cs

HHI_MPLL_CNTL7 0xA5

Bit(s)	R/W	Default	Description
[31]	R/W	0	mpll_dds3_en
[30]	R/W	0	mpll_dds3_sdm_en
[29]	R/W	0	mpll_dds3_ss_en
[28:20]	R/W	0	mpll_dds3_n_in
[13:0]	R/W	0	mpll_dds3_sdm_in

HHI_MPLL_CNTL8 0xA6

Bit(s)	R/W	Default	Description
[31]	R/W	0	mpll_dds3_ir_bypass
[30]	R/W	0	mpll_dds3_load
[29]	R/W	0	mpll_dds3_lp_en
[28]	R/W	0	mpll_dds3_modesel
[20:16]	R/W	0	mpll_dds3_f_set
[15:12]	R/W	0	mpll_dds3_ir_byin
[10:8]	R/W	0	mpll_dds3_p_set
[7:4]	R/W	0	mpll_dds3_vref_cf
[3:0]	R/W	0	mpll_dds3_vref_cs

HHI_MPLL_STS 0xA7

Bit(s)	R/W	Default	Description
[28]	R	0	mpll_dds0_ir_done
[27:24]	R	0	mpll_dds0_ir_out
[20]	R	0	mpll_dds1_ir_done
[19:16]	R	0	mpll_dds1_ir_out
[12]	R	0	mpll_dds2_ir_done
[11:8]	R	0	mpll_dds2_ir_out
[4]	R	0	mpll_dds3_ir_done
[3:0]	R	0	mpll_dds3_ir_out

HHI_FIX_PLL_CNTL0 0xA8

Bit(s)	R/W	Default	Description

[31]	R	0	fix_dpll_lock
[30]	R	0	fix_dpll_lock_a
[29]	R/W	1	fix_dpll_reset
[28]	R/W	0	fix_dpll_en
[17:16]	R/W	0	fix_dpll_od
[14:10]	R/W	0	fix_dpll_N
[7:0]	R/W	0	fix_dpll_M

HHI_FIX_PLL_CNTL1 0xA9

Bit(s)	R/W	Default	Description
[25]	R/W	0	fix_dpll_div2p5_en
[24:20]	R/W	0	fixpll_dpll_clken
[18:0]	R/W	0	fix_dpll_frac

HHI_FIX_PLL_CNTL2 0xAA

Bit(s)	R/W	Default	Description
[22:20]	R/W	0	fix_dpll_fref_sel
[17:16]	R/W	0	fix_dpll_os_ssc
[15:12]	R/W	0	fix_dpll_ssc_str_m
[8]	R/W	0	fix_dpll_ssc_en
[7:4]	R/W	0	fix_dpll_ssc_dep_sel
[1:0]	R/W	0	fix_dpll_ss_mode

HHI_FIX_PLL_CNTL3 0xAB

Bit(s)	R/W	Default	Description
31	R/W	0	fix_dpll_afc_bypass
30	R/W	0	fix_dpll_afc_clk_sel
29	R/W	0	fix_dpll_code_new
28	R/W	0	fix_dpll_dco_m_en
27	R/W	0	fix_dpll_dco_sdm_en
26	R/W	0	fix_dpll_div2
25	R/W	0	fix_dpll_div_mode
24	R/W	0	fix_dpll_fast_lock

Bit(s)	R/W	Default	Description
23	R/W	0	fix_dpll_fb_pre_div
22	R/W	0	fix_dpll_filter_mode
21	R/W	0	fix_dpll_fix_en
20	R/W	0	fix_dpll_freq_shift_en
19	R/W	0	fix_dpll_load
18	R/W	0	fix_dpll_load_en
17	R/W	0	fix_dpll_lock_f
16	R/W	0	fix_dpll_pulse_width_en
15	R/W	0	fix_dpll_sdmnc_en
14	R/W	0	fix_dpll_sdmnc_mode
13	R/W	0	fix_dpll_sdmnc_range
12	R/W	0	fix_dpll_tdc_en
11	R/W	0	fix_dpll_tdc_mode_sel
10	R/W	0	fix_dpll_wait_en
7	R/W	0	fix_dpll_ssclk_sel
6	R/W	0	fix_dpll_clk_irsel
5	R/W	0	fix_dpll_clk50m_en

HHI_FIX_PLL_CNTL4 0xAC

Bit(s)	R/W	Default	Description
[31:28]	R/W	0	fix_dpll_alpha
[27:24]	R/W	0	fix_dpll_rou
[18:16]	R/W	0	fix_dpll_lambda0
[13:12]	R/W	0	fix_dpll_acq_gain
[11:8]	R/W	0	fix_dpll_filter_pvt2
[7:4]	R/W	0	fix_dpll_filter_pvt1
[1:0]	R/W	0	fix_dpll_pfd_gain

HHI_FIX_PLL_CNTL5 0xAD

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	fix_dpll_adj_vco_ldo

Bit(s)	R/W	Default	Description
[27:24]	R/W	0	fix_dpll_lm_w
[21:16]	R/W	0	fix_dpll_lm_s
[15:0]	R/W	0	fix_dpll_reve

HHI_FIX_PLL_CNTL6 0xAE

Bit(s)	R/W	Default	Description
[31:30]	R/W	0	fix_dpll_afc_hold_t
[29:28]	R/W	0	fix_dpll_lkw_sel
[27:26]	R/W	0	fix_dpll_dco_sdm_clk_sel
[25:24]	R/W	0	fix_dpll_afc_in
[23:22]	R/W	0	fix_dpll_afc_nt
[21:20]	R/W	0	fix_dpll_vc_in
[19:18]	R/W	0	fix_dpll_lock_long
[17:16]	R/W	0	fix_dpll_freq_shift_v
[14:12]	R/W	0	fix_dpll_data_sel
[10:8]	R/W	0	fix_dpll_sdmnc_ulms
[6:0]	R/W	0	fix_dpll_sdmnc_power

HHI_FIX_PLL_STS 0xAF

Bit(s)	R/W	Default	Description
[31]	R	0	fix_dpll_lock
[30]	R	0	fix_dpll_lock_a
[29]	R	0	fix_dpll_afc_done
[22:16]	R	0	fix_dpll_sdmnc_monitor
[9:0]	R	0	fix_dpll_out_rsv

HHI_VDAC_CNTL0 0xBB

Bit(s)	R/W	Default	Description
27	R/W	0	CDAC_BIAS_C
26	R/W	0	CDAC_EXT_VREF_EN
25	R/W	0	CDAC_DRIVER_ADJ
24	R/W	0	CDAC_CLK_PHASE_SEL

Bit(s)	R/W	Default	Description
23~21	R/W	0	CDAC_RL_ADJ
20~16	R/W	0	CDAC_VREF_ADJ
15~8	R/W	0	CDAC_CTRL_RESV2
7~0	R/W	0	CDAC_CTRL_RESV1

HHI_VDAC_CNTL1 0xBC

Bit(s)	R/W	Default	Description
23~16	R	0	CDAC_DIG_OUT_RESV
15~4	R	0	Reserved
3	R/W	0	Cdac_pwd
2~0	R/W	0	CDAC_GSW

HHI_SYS_PLL_CNTL0 0xBD

Bit(s)	R/W	Default	Description
[31]	R	0	sys_dpll_lock
[30]	R	0	sys_dpll_lock_a
[29]	R/W	1	sys_dpll_reset
[28]	R/W	0	sys_dpll_en
[18:16]	R/W	0	sys_dpll_od
[14:10]	R/W	0	sys_dpll_N
[7:0]	R/W	0	sys_dpll_M

HHI_SYS_PLL_CNTL1 0xBE

Bit(s)	R/W	Default	
[18:0]	R/W	0	sys_dpll_frac

HHI_SYS_PLL_CNTL2 0xBF

Bit(s)	R/W	Default	Description
[22:20]	R/W	0x0	sys_dpll_fref_sel
[17:16]	R/W	0x0	sys_dpll_os_ssc
[15:12]	R/W	0x0	sys_dpll_ssc_str_m
[8]	R/W	0x0	sys_dpll_ssc_en
[7:4]	R/W	0x0	sys_dpll_ssc_dep_sel

Bit(s)	R/W	Default	Description
[1:0]	R/W	0x0	sys_dpll_ss_mode

HHI_SYS_PLL_CNTL3 0xC0

Bit(s)	R/W	Default	Description
31	R/W	0x0	sys_dpll_afc_bypass
30	R/W	0x0	sys_dpll_afc_clk_sel
29	R/W	0x0	sys_dpll_code_new
28	R/W	0x0	sys_dpll_dco_m_en
27	R/W	0x0	sys_dpll_dco_sdm_en
26	R/W	0x0	sys_dpll_div2
25	R/W	0x0	sys_dpll_div_mode
24	R/W	0x0	sys_dpll_fast_lock
23	R/W	0x0	sys_dpll_fb_pre_div
22	R/W	0x0	sys_dpll_filter_mode
21	R/W	0x0	sys_dpll_fix_en
20	R/W	0x0	sys_dpll_freq_shift_en
19	R/W	0x0	sys_dpll_load
18	R/W	0x0	sys_dpll_load_en
17	R/W	0x0	sys_dpll_lock_f
16	R/W	0x0	sys_dpll_pulse_width_en
15	R/W	0x0	sys_dpll_sdmnc_en
14	R/W	0x0	sys_dpll_sdmnc_mode
13	R/W	0x0	sys_dpll_sdmnc_range
12	R/W	0x0	sys_dpll_tdc_en
11	R/W	0x0	sys_dpll_tdc_mode_sel
10	R/W	0x0	sys_dpll_wait_en

HHI_SYS_PLL_CNTL4 0xC1

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	sys_dpll_alpha
[26:24]	R/W	0	sys_dpll_rou

[22:20]	R/W	0	sys_dpll_lambda1
[18:16]	R/W	0	sys_dpll_lambda0
[13:12]	R/W	0	sys_dpll_acq_gain
[11:8]	R/W	0	sys_dpll_filter_pvt2
[7:4]	R/W	0	sys_dpll_filter_pvt1
[1:0]	R/W	0	sys_dpll_pfd_gain

HHI_SYS_PLL_CNTL5 0xC2

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	sys_dpll_adj_vco_ldo
[27:24]	R/W	0	sys_dpll_lm_w
[21:16]	R/W	0	sys_dpll_lm_s
[15:0]	R/W	0	sys_dpll_reve

HHI_SYS_PLL_CNTL6 0xC3

Bit(s)	R/W	Default	Description
[31:30]	R/W	0	sys_dpll_afc_hold_t
[29:28]	R/W	0	sys_dpll_lkw_sel
[27:26]	R/W	0	sys_dpll_dco_sdm_clk_sel
[25:24]	R/W	0	sys_dpll_afc_in
[23:22]	R/W	0	sys_dpll_afc_nt
[21:20]	R/W	0	sys_dpll_vc_in
[19:18]	R/W	0	sys_dpll_lock_long
[17:16]	R/W	0	sys_dpll_freq_shift_v
[14:12]	R/W	0	sys_dpll_data_sel
[10:8]	R/W	0	sys_dpll_sdmnc_ulms
[6:0]	R/W	0	sys_dpll_sdmnc_power

HHI_SYS_PLL_STS 0xC4

Bit(s)	R/W	Default	Description
[31]	R	0	sys_dpll_lock
[30]	R	0	sys_dpll_lock_a
[29]	R	0	sys_dpll_afc_done

[22:16]	R	0	sys_dpll_sdmnc_monitor
[9:0]	R	0	sys_dpll_out_rsv

HHI_HDMI_PLL_CNTL0 0xC8

There is some internal muxing controlled by “VLOCK_CNTL_EN” bit[20] of HHI_HDMI_PLL_CNTL6.

Bit(s)	R/W	Default	Description
[31]	R	0	hdmi_dpll_lock
[30]	R	0	hdmi_dpll_lock_a
[29]	R/W	1	hdmi_dpll_reset
[28]	R/W	0	hdmi_dpll_en
[25:24]	R/W	0	hdmi_dpll_out_gate_ctrl
[21:20]	R/W	0	hdmi_dpll_od2
[19:16]	R/W	0	hdmi_dpll_od
[14:10]	R/W	0	hdmi_dpll_N
[7:0]	R/W	0	hdmi_dpll_M

HHI_HDMI_PLL_CNTL1 0xC9

There is some internal muxing controlled by “VLOCK_CNTL_EN” bit[20] of HHI_HDMI_PLL_CNTL6.

Bit(s)	R/W	Default	Description
[18:0]	R/W	0	hdmi_dpll_frac

HHI_HDMI_PLL_CNTL2 0xCA

There is some internal muxing controlled by “VLOCK_CNTL_EN” bit[20] of HHI_HDMI_PLL_CNTL6.

Bit(s)	R/W	Default	Description
[22:20]	R/W	0	hdmi_dpll_fref_sel
[17:16]	R/W	0	hdmi_dpll_os_ssc
[15:12]	R/W	0	hdmi_dpll_ssc_str_m
[8]	R/W	0	hdmi_dpll_ssc_en
[7:4]	R/W	0	hdmi_dpll_ssc_dep_sel
[1:0]	R/W	0	hdmi_dpll_ss_mode

HHI_HDMI_PLL_CNTL3 0xCB

There is some internal muxing controlled by “VLOCK_CNTL_EN” bit[20] of HHI_HDMI_PLL_CNTL6.

Bit(s)	R/W	Default	Description
31	R/W	0	hdmi_dpll_afc_bypass
30	R/W	0	hdmi_dpll_afc_clk_sel
29	R/W	0	hdmi_dpll_code_new
28	R/W	0	hdmi_dpll_dco_m_en
27	R/W	0	hdmi_dpll_dco_sdm_en
26	R/W	0	hdmi_dpll_div2
25	R/W	0	hdmi_dpll_div_mode
24	R/W	0	hdmi_dpll_fast_lock
23	R/W	0	hdmi_dpll_fb_pre_div
22	R/W	0	hdmi_dpll_filter_mode
21	R/W	0	hdmi_dpll_fix_en
20	R/W	0	hdmi_dpll_freq_shift_en
19	R/W	0	hdmi_dpll_load
18	R/W	0	hdmi_dpll_load_en
17	R/W	0	hdmi_dpll_lock_f
16	R/W	0	hdmi_dpll_pulse_width_en
15	R/W	0	hdmi_dpll_sdmnc_en
14	R/W	0	hdmi_dpll_sdmnc_mode
13	R/W	0	hdmi_dpll_sdmnc_range
12	R/W	0	hdmi_dpll_tdc_en
11	R/W	0	hdmi_dpll_tdc_mode_sel
10	R/W	0	hdmi_dpll_wait_en

HHI_HDMI_PLL_CNTL4 0xCC

There is some internal muxing controlled by “VLOCK_CNTL_EN” bit[20] of HHI_HDMI_PLL_CNTL6.

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	hdmi_dpll_alpha
[26:24]	R/W	0	hdmi_dpll_rou
[22:20]	R/W	0	hdmi_dpll_lambda1
[18:16]	R/W	0	hdmi_dpll_lambda0

[13:12]	R/W	0	hdmi_dpll_acq_gain
[11:8]	R/W	0	hdmi_dpll_filter_pvt2
[7:4]	R/W	0	hdmi_dpll_filter_pvt1
[1:0]	R/W	0	hdmi_dpll_pfd_gain

HHI_HDMI_PLL_CNTL5 0xCD

There is some internal muxing controlled by “VLOCK_CNTL_EN” bit[20] of HHI_HDMI_PLL_CNTL6.

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	hdmi_dpll_adj_vco_ldo
[27:24]	R/W	0	hdmi_dpll_lm_w
[21:16]	R/W	0	hdmi_dpll_lm_s
[15:0]	R/W	0	hdmi_dpll_reve

HHI_HDMI_PLL_CNTL6 0xCE

There is some internal muxing controlled by “VLOCK_CNTL_EN” bit[20] of HHI_HDMI_PLL_CNTL6.

Bit(s)	R/W	Default	Description
[31:30]	R/W	0	hdmi_dpll_afc_hold_t
[29:28]	R/W	0	hdmi_dpll_lkw_sel
[27:26]	R/W	0	hdmi_dpll_dco_sdm_clk_sel
[25:24]	R/W	0	hdmi_dpll_afc_in
[23:22]	R/W	0	hdmi_dpll_afc_nt
[21:20]	R/W	0	hdmi_dpll_vc_in
[19:18]	R/W	0	hdmi_dpll_lock_long
[17:16]	R/W	0	hdmi_dpll_freq_shift_v
[14:12]	R/W	0	hdmi_dpll_data_sel
[10:8]	R/W	0	hdmi_dpll_sdmnc_ulms
[6:0]	R/W	0	hdmi_dpll_sdmnc_power

HHI_HDMI_PLL_STS 0xCF

There is some internal muxing controlled by “VLOCK_CNTL_EN” bit[20] of HHI_HDMI_PLL_CNTL6.

Bit(s)	R/W	Default	Description
[31]	R	0	hdmi_dpll_lock
[30]	R	0	hdmi_dpll_lock_a

[29]	R	0	hdmi_dpll_afc_done
[22:16]	R	0	hdmi_dpll_sdmnc_monitor
[9:0]	R	0	hdmi_dpll_out_rsv

HHI_HDMI_PLL_VLOCK_CNTL 0xD1

Bit(s)	R/W	Default	Description
1	R/W	0	Vlock_adj_en_to_pll sel: 1:vloc k_adj_en_pll 0:hi_hdmi_pll_cntl3[19]
0	R/W	0	Hdmi_pll_vlock_cntl_en

HHI_SYS1_PLL_CNTL0 0xE0

Bit(s)	R/W	Default	Description
[31]	R	0	sys1_dpll_lock
[30]	R	0	sys1_dpll_lock_a
[29]	R/W	1	sys1_dpll_reset
[28]	R/W	0	sys1_dpll_en
[18:16]	R/W	0	sys1_dpll_od
[14:10]	R/W	0	sys1_dpll_N
[8:0]	R/W	0	sys1_dpll_int_num

HHI_SYS1_PLL_CNTL1 0xE1

Bit(s)	R/W	Default	Description
[18:0]	R/W	0	sys1_dpll_frac

HHI_SYS1_PLL_CNTL2 0xE2

Bit(s)	R/W	Default	Description
[22:20]	R/W	0x0	sys1_dpll_fref_sel
[17:16]	R/W	0x0	sys1_dpll_os_ssc
[15:12]	R/W	0x0	sys1_dpll_ssc_str_m
[8]	R/W	0x0	sys1_dpll_ssc_en
[7:4]	R/W	0x0	sys1_dpll_ssc_dep_sel
[1:0]	R/W	0x0	sys1_dpll_ss_mode

HHI_SYS1_PLL_CNTL3 0xE3

Bit(s)	R/W	Default	Description
31	R/W	0x0	sys1_dpll_afc_bypass
30	R/W	0x0	sys1_dpll_afc_clk_sel
29	R/W	0x0	sys1_dpll_code_new
28	R/W	0x0	sys1_dpll_dco_m_en
27	R/W	0x0	sys1_dpll_dco_sdm_en
26	R/W	0x0	sys1_dpll_div2
25	R/W	0x0	sys1_dpll_div_mode
24	R/W	0x0	sys1_dpll_fast_lock
23	R/W	0x0	sys1_dpll_fb_pre_div
22	R/W	0x0	sys1_dpll_filter_mode
21	R/W	0x0	sys1_dpll_fix_en
20	R/W	0x0	sys1_dpll_freq_shift_en
19	R/W	0x0	sys1_dpll_load
18	R/W	0x0	sys1_dpll_load_en
17	R/W	0x0	sys1_dpll_lock_f
16	R/W	0x0	sys1_dpll_pulse_width_en
15	R/W	0x0	sys1_dpll_sdmnc_en
14	R/W	0x0	sys1_dpll_sdmnc_mode
13	R/W	0x0	sys1_dpll_sdmnc_range
12	R/W	0x0	sys1_dpll_tdc_en
11	R/W	0x0	sys1_dpll_tdc_mode_sel
10	R/W	0x0	sys1_dpll_wait_en

HHI_SYS1_PLL_CNTL4 0xE4

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	sys1_dpll_alpha
[26:24]	R/W	0	sys1_dpll_rou
[18:16]	R/W	0	sys1_dpll_lambda0
[13:12]	R/W	0	sys1_dpll_acq_gain

[11:8]	R/W	0	sys1_dpll_filter_pvt2
[7:4]	R/W	0	sys1_dpll_filter_pvt1
[1:0]	R/W	0	sys1_dpll_pfd_gain

HHI_SYS1_PLL_CNTL5 0xE5

Bit(s)	R/W	Default	Description
[30:28]	R/W	0	sys1_dpll_adj_vco_ldo
[27:24]	R/W	0	sys1_dpll_lm_w
[21:16]	R/W	0	sys1_dpll_lm_s
[15:0]	R/W	0	sys1_dpll_reve

HHI_SYS1_PLL_CNTL6 0xE6

Bit(s)	R/W	Default	Description
[31:30]	R/W	0	sys1_dpll_afc_hold_t
[29:28]	R/W	0	sys1_dpll_lkw_sel
[27:26]	R/W	0	sys1_dpll_dco_sdm_clk_sel
[25:24]	R/W	0	sys1_dpll_afc_in
[23:22]	R/W	0	sys1_dpll_afc_nt
[21:20]	R/W	0	sys1_dpll_vc_in
[19:18]	R/W	0	sys1_dpll_lock_long
[17:16]	R/W	0	sys1_dpll_freq_shift_v
[14:12]	R/W	0	sys1_dpll_data_sel
[10:8]	R/W	0	sys1_dpll_sdmnc_ulms
[6:0]	R/W	0	sys1_dpll_sdmnc_power

HHI_SYS1_PLL_STS0xE7

Bit(s)	R/W	Default	Description
[31]	R	0	sys1_dpll_lock
[30]	R	0	sys1_dpll_lock_a
[29]	R	0	sys1_dpll_afc_done
[22:16]	R	0	sys1_dpll_sdmnc_monitor
[9:0]	R	0	sys1_dpll_out_rsv

HHI_HDMI_PHY_CNTL0 0xE8

Bit(s)	R/W	Default	Description
31~16	R/W	0	HDMI_CTL1
15~0	R/W	0	HDMI_CTL0

HHI_HDMI_PHY_CNTL1 0xE9

Bit(s)	R/W	Default	Description
31:30	R/W	0	New_prbs_mode
29:28	R/W	0	New_prbs_prbsmode
27	R/W	0	New_prbs_sel
26	R/W	0	New_prbs_en
25:24	R/W	3	Ch3_swap: 0:ch0/1:ch1/2:ch2/3:ch3
23:22	R/W	2	Ch2_swap: 0:ch0/1:ch1/2:ch2/3:ch3
21:20	R/W	1	Ch1_swap: 0:ch0/1:ch1/2:ch2/3:ch3
19:18	R/W	0	Ch0_swap: 0:ch0/1:ch1/2:ch2/3:ch3
17	R/W	0	BIT_INVERT
16	R/W	0	MSB_LSB_SWAP
15	R/W	0	Capture_add1
14	R/W	0	CAPTURE_CLK_GATE_EN
13	R/W	0	HDMI_TX_PRBS_EN: Set to 1 to enable the PRBS engine
12	R/W	0	HDMI_TX_PRBS_ERR_EN: Set to 1 to enable the error flag detector. Set to 0 to reset the error detection logic
11~8	R/W	0	HDMI_TX_SET_HIGH: Set each bit to 1 to set the HDMI pin high
7~4	R/W	0	HDMI_TX_SET_LOW: Set each bit to 0 to set the HDMI Pins low
3	R/W	0	HDMI_FIFO_WR_ENALBE
2	R/W	0	HDMI_FIFO_ENABLE
1	R/W	0	HDMI_TX_PHY_CLK_EN: Set to 1 to enable the HDMI TX PHY
0	R/W	0	HDMI_TX_PHY_SOFT_RESET: Set to 1 to reset the HDMI TX PHY

HHI_HDMI_PHY_CNTL2 0xEA

Bit(s)	R/W	Default	Description
31~9	R	0	Reserved
8	R	0	Test error

7~0	R	0	HDMI_REGRD

HHI_HDMI_PHY_CNTL3 0xEB

Bit(s)	R/W	Default	Description
31~0	R/W	0	RSV

HHI_HDMI_PHY_CNTL4 0xEC

Bit(s)	R/W	Default	Description
[31:24]	R/W	0	New_prbs_err_thr
[21:20]	R/W	0	Dtest_sel
[19]	R/W	0	New_prbs_clr_ber_meter
[17]	R/W	0	New_prbs_freez_ber
[16]	R/W	0	New_prbs_inverse_in
[15:14]	R/W	0	New_prbs_mode
[13:12]	R/W	0	New_prbs_prbs_mode
[11:0]	R/W	0	New_prbs_time_window

HHI_HDMI_PHY_CNTL4 0xED

Bit(s)	R/W	Default	Description
[31:24]	R/W	0	New_prbs_err_thr
[21:20]	R/W	0	Dtest_sel
[19]	R/W	0	New_prbs_clr_ber_meter
[17]	R/W	0	New_prbs_freez_ber
[16]	R/W	0	New_prbs_inverse_in
[15:14]	R/W	0	New_prbs_mode

HHI_HDMI_PHY_STATUS 0xEE

Bit(s)	R/W	Default	Description
[29]	R		Prbs_enable
[28]	R		Test_err
[24]	R		New_prbs_pattern_nok
[20]	R		New_prbs_lock
[19:0]	R		New_prbs_ber_meter

HHI_AXI_PIPEL_CNTL2 0xf1

Bit(s)	R/W	Default	Description
31	R	-	vpu_dmc_axi_pipel_12 : chan_idle
30	R/W	1	vpu_dmc_axi_pipel_12 :req_en
29	R/W	1	vpu_dmc_axi_pipel_12 :auto_gclk_en
28	R/W	0	vpu_dmc_axi_pipel_12 :disable_gclk
27	R	-	vdec_pipel : chan_idle
26	R/W	1	vdec_pipel :req_en
25	R/W	1	vdec_pipel :auto_gclk_en
24	R/W	0	vdec_pipel :disable_gclk
23	R	-	Reserved
22	R/W	1	Reserved
21	R/W	1	Reserved
20	R/W	0	Reserved
3	R	-	dmc_pipel_hdcp22 : chan_idle
2	R/W	1	dmc_pipel_hdcp22 :req_en
1	R/W	1	dmc_pipel_hdcp22 :auto_gclk_en
0	R/W	0	dmc_pipel_hdcp22 :disable_gclk

HHI_VID_LOCK_CLK_CNTL 0xF2

Bit(s)	R/W	Default	Description
31~10	R/W	0	reserved
9-8	R/W	0	Clk_sel: 0:cts_oscin_clk 1:cts_encl_clk 2:cts_enci_clk 3:cts_encp_clk
7	R/W	0	Clk_en
6-0	R/W	0	Clk_div

HHI_AXI_PIPEL_CNTL1 0xf3

Bit(s)	R/W	Default	Description
31	R	-	Reserved

Bit(s)	R/W	Default	Description
30	R/W	1	Reserved
29	R/W	1	Reserved
28	R/W	0	Reserved
27	R	-	Reserved
26	R/W	1	Reserved
25	R/W	1	Reserved
24	R/W	0	Reserved
23	R	-	axi_pipel_HEVC Encoder_1 : chan_idle
22	R/W	1	axi_pipel_HEVC Encoder_1 :req_en
21	R/W	1	axi_pipel_HEVC Encoder_1 :auto_gclk_en
20	R/W	0	axi_pipel_HEVC Encoder_1 :disable_gclk
19	R	-	axi_pipel_HEVC Encoder: chan_idle
18	R/W	1	axi_pipel_HEVC Encoder:req_en
17	R/W	1	axi_pipel_HEVC Encoder:auto_gclk_en
16	R/W	0	axi_pipel_HEVC Encoder:disable_gclk
15	R	-	hevc_f_pipel : chan_idle
14	R/W	1	hevc_f_pipel :req_en
13	R/W	1	hevc_f_pipel :auto_gclk_en
12	R/W	0	hevc_f_pipel :disable_gclk
11	R	-	hevc_pipel : chan_idle
10	R/W	1	hevc_pipel :req_en
9	R/W	1	hevc_pipel :auto_gclk_en
8	R/W	0	hevc_pipel :disable_gclk
7	R	-	vpu_dmc_pipel10 : chan_idle
6	R/W	1	vpu_dmc_pipel10 :req_en
5	R/W	1	vpu_dmc_pipel10 :auto_gclk_en
4	R/W	0	vpu_dmc_pipel10 :disable_gclk
3	R	-	vpu_dmc_pipel11 : chan_idle
2	R/W	1	vpu_dmc_pipel11 :req_en

Bit(s)	R/W	Default	Description
1	R/W	1	vpu_dmc_pipel11 :auto_gclk_en
0	R/W	0	vpu_dmc_pipel11 :disable_gclk

HHI_AXI_PIPEL_CNTL 0xF4

Bit(s)	R/W	Default	Description
27	R	-	hevcb_dmc_pipel : chan_idle
26	R/W	1	hevcb_dmc_pipel :req_en
25	R/W	1	hevcb_dmc_pipel :auto_gclk_en
24	R/W	0	hevcb_dmc_pipel :disable_gclk
23	R	-	hevcf_dmc_pipel : chan_idle
22	R/W	1	hevcf_dmc_pipel :req_en
21	R/W	1	hevcf_dmc_pipel :auto_gclk_en
20	R/W	0	hevcf_dmc_pipel :disable_gclk
15	R	-	vpu_dmc_pipel02 : chan_idle
14	R/W	1	vpu_dmc_pipel02 :req_en
13	R/W	1	vpu_dmc_pipel02 :auto_gclk_en
12	R/W	0	vpu_dmc_pipel02 :disable_gclk
11	R	-	vpu_dmc_pipel01 : chan_idle
10	R/W	1	vpu_dmc_pipel01 :req_en
9	R/W	1	vpu_dmc_pipel01 :auto_gclk_en
8	R/W	0	vpu_dmc_pipel01 :disable_gclk
7	R	-	vpu_dmc_pipel00 : chan_idle
6	R/W	1	vpu_dmc_pipel00 :req_en
5	R/W	1	vpu_dmc_pipel00 :auto_gclk_en
4	R/W	0	vpu_dmc_pipel00 :disable_gclk
3	R	-	hcodec_pipel : chan_idle
2	R/W	1	hcodec_pipel :req_en
1	R/W	1	hcodec_pipel :auto_gclk_en
0	R/W	0	hcodec_pipel :disable_gclk

HHI_AXI_PIPEL_CNTL 0xF4

Bit(s)	R/W	Default	Description
31	R	-	DVALIN DMC Pipeline control: CHAN_IDLE
30	R/W	1	DVALIN DMC Pipeline control: REQ_EN
29	R/W	1	DVALIN DMC Pipeline control: AUTO_GCLK_EN
28	R/W	0	DVALIN DMC Pipeline control: DISABLE_GCLK
27	R	-	HEVCF DMC Pipeline control: CHAN_IDLE
26	R/W	1	HEVCF DMC Pipeline control: REQ_EN
25	R/W	1	HEVCF DMC Pipeline control: AUTO_GCLK_EN
24	R/W	0	HEVCF DMC Pipeline control: DISABLE_GCLK
23	R	-	HEVCB DMC Pipeline control: CHAN_IDLE
22	R/W	1	HEVCB DMC Pipeline control: REQ_EN
21	R/W	1	HEVCB DMC Pipeline control: AUTO_GCLK_EN
20	R/W	0	HEVCB DMC Pipeline control: DISABLE_GCLK

HHI_BT656_CLK_CNTL 0xF5

Bit(s)	R/W	Default	Description
10-9	R/W	0	Bt656_1_clk_sel 0:fclk_div2 1:fclk_div3 2:fclk_div5 3:fclk_div7
8	R/W	0	Reserved
7	R/W	0	Bt656_1_clk_en
6-0	R/W	0	Bt656_1_clk_div

HHI_CDAC_CLK_CNTL 0xF6

Bit(s)	R/W	Default	Description
20	R/W	0	vdac_clk_c clk_en
17:16	R/W	0	vdac_clk_c clk_sel 0:osc; 1:fclk_div5;
15:0	R/W	0	vdac_clk_c: clk_div

HHI_SPICC_CLK_CNTL 0xF7

Bit(s)	R/W	Default	Description
25:23	R/W	0	spicc_1_clk clk_sel 0:osc; 1:clk81; 2:fclk_div4; 3:fclk_div3; 4:fclk_div2; 5:fclk_div5; 6:fclk_div7; 7:gp0_pll;
22	R/W	0	spicc_1_clk clk_en
21:16	R/W	0	spicc_1_clk clk_div
9:7	R/W	0	spicc_0_clk clk_sel 0:osc; 1:clk81; 2:fclk_div4; 3:fclk_div3; 4:fclk_div2; 5:fclk_div5; 6:fclk_div7; 7:gp0_pll;
6	R/W	0	spicc_0_clk clk_en
5:0	R/W	0	spicc_0_clk clk_div

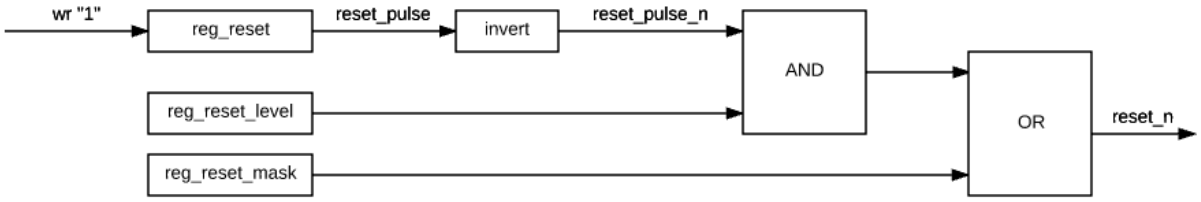
8.7 Reset

8.7.1 Overview

As shown in figure below, S922X reset are designed as following:

Each reset includes 3 control registers, reg_reset, reg_reset_level, and reg_reset_mask, when write “1” to reg_reset, it will generate a signal: reset_pulse; most of module’s reset is negative, so reset_pulse will invert to reset_pulse_n; the usage of reg_reset_level is hold reset_pulse_n to “0”; the usage of reg_reset_mask is hold reset_pulse_n to “1”; ao module has special soft reset control; A53 has special soft reset control.

Figure 8-22 S922X Reset Design Diagram



8.7.2 Register Description

8.7.2.1 EE Reset

RESET0_REGISTER: 0xFFD01004

Bit(s)	R/W	Default	Description
31-27	R/W	0	Reserved
26	R/W	0	vcbus
25	R/W	0	ahb_data
24	R/W	0	ahb_cntl
23	R/W	0	cbus_capb3
22	R/W	0	Reserved
21	R/W	0	dos_capb3
20	R/W	0	Dvalin_capb3
19	R/W	0	hdmitx_capb3
18	R/W	0	Reserved
17	R/W	0	decode_capb3
16	R/W	0	gic
15	R/W	0	pcie_apb
14	R/W	0	pcie_phy
13	R/W	0	vcbus
12	R/W	0	pcie_ctrl_A
11	R/W	0	assist
10	R/W	0	venc
9:8	R/W	0	Reserved

Bit(s)	R/W	Default	Description
7	R/W	0	vid_pll_div
6	R/W	0	afifo
5	R/W	0	viu
4	R/W	0	Reserved
3	R/W	0	ddr
2	R/W	0	dos
1	R/W	0	Reserved
0	R/W	0	hiu

RESET1_REGISTER: 0xFFD01008

Bit(s)	R/W	Default	Description
31-30	R/W	0	Reserved
29	R/W	0	Audio codec
28:15	R/W	0	Reserved
14	R/W	0	sd_emmcC
13	R/W	0	sd_emmcB
12	R/W	0	sd_emmcA
11	R/W	0	eth
10	R/W	0	isa
9	R/W	0	Reserved
8	R/W	0	parser
7	R/W	0	Reserved
6	R/W	0	ahb_sram
5	R/W	0	bt656
4	R/W	0	Reserved
3	R/W	0	ddr
2	R/W	0	usb
1	R/W	0	demux
0	R/W	0	Reserved

RESET2_REGISTER: 0xFFD0100C

Bit(s)	R/W	Default	Description
15	R/W	0	hdmitx
14	R/W	0	Dvalin
13	R/W	0	Reserved
12	R/W	0	Reserved
11	R/W	0	Reserved
10	R/W	0	parser_top
9	R/W	0	parser_ctl
8	R/W	0	parser_fetch
7	R/W	0	parser_reg
6	R/W	0	ge2d
5	R/W	0	alocker
4	R/W	0	mipi_dsi_host
3	R/W	0	Reserved
2	R/W	0	hdmi_tx
1	R/W	0	audio
0	R/W	0	Reserved

RESET3_REGISTER: 0xFFD01010

Bit(s)	R/W	Default	Description
15	R/W	0	demux_2
14	R/W	0	demux_1
13	R/W	0	demux_0
12	R/W	0	demux_s2p_1
11	R/W	0	demux_s2p_0
10	R/W	0	demux_des_pl
9	R/W	0	demux_top
8:0	R/W	0	Reserved

RESET4_REGISTER: 0xFFD01014

Bit(s)	R/W	Default	Description
15	R/W	0	i2c_m2
14	R/W	0	i2c_m1
13	R/W	0	vencl
12	R/W	0	vdi6
11:10	R/W	0	Reserved
9	R/W	0	vdac
8	R/W	0	Reserved
7	R/W	0	vencp
6	R/W	0	venci
5	R/W	0	rdma
4:3	R/W	0	Reserved
2	R/W	0	mipi_dsiphy
1:0	R/W	0	Reserved

RESET6_REGISTER: 0xFFD0101c

Bit(s)	R/W	Default	Description
14	R/W	0	i2c_m3
13	R/W	0	spifc0
12	R/W	0	async1
11	R/W	0	async0
10	R/W	0	uart1_2
9	R/W	0	UART_EE_A
8	R/W	0	ts_cpu
7	R/W	0	stream
6	R/W	0	spicc1
5	R/W	0	ts_pll
4	R/W	0	i2c_m0
3	R/W	0	sana_3
2	R/W	0	sc

Bit(s)	R/W	Default	Description
1	R/W	0	spicc0
0	R/W	0	gen

RESET7_REGISTER: 0xFFD01020

Bit(s)	R/W	Default	Description
13	R/W	0	hevcf_dmc_pipl
12	R/W	0	wave420_dmc_pipl
11	R/W	0	hcodec_dmc_pipl
10	R/W	0	ge2d_dmc_pipl
9	R/W	0	dmc_vpu_pipl
8	R/W	0	nic_dmc_pipl
7	R/W	0	vid_lock
6	R/W	0	Dvalin_dmc_pipl
5	R/W	0	device_mmc_arb
4	R/W	0	ts_gpu
3	R/W	0	usb_ddr3
2	R/W	0	usb_ddr2
1	R/W	0	usb_ddr1
0	R/W	0	usb_ddr0

RESET0_MASK: 0xFFD01040

The Bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

RESET1_MASK: 0xFFD01044

The Bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

RESET2_MASK: 0xFFD01048

The Bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

RESET3_MASK: 0xFFD0104c

The Bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

RESET4_MASK: 0xFFD01050

The Bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

RESET6_MASK: 0xFFD01058

The Bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

RESET7_MASK: 0xFFD0105c

The Bits of this register correspond to the RESET[n] REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

RESET0_LEVEL: 0xFFD01080

The bits of this register correspond to the RESET[n] REGISTERs above. The default of this register is 0xFFFFFFFF. Setting any bit to 0, forces the corresponding RESET LOW. This registers allows the software to “Hold” a reset LOW (in a reset condition).

RESET1_LEVEL: 0xFFD01084

The bits of this register correspond to the RESET[n] REGISTERs above. The default of this register is 0xFFFFFFFF. Setting any bit to 0, forces the corresponding RESET LOW. This registers allows the software to “Hold” a reset LOW (in a reset condition).

RESET2_LEVEL: 0xFFD01088

The bits of this register correspond to the RESET[n] REGISTERs above. The default of this register is 0xFFFFFFFF. Setting any bit to 0, forces the corresponding RESET LOW. This registers allows the software to “Hold” a reset LOW (in a reset condition).

RESET3_LEVEL: 0xFFD0108c

The bits of this register correspond to the RESET[n] REGISTERs above. The default of this register is 0xFFFFFFFF. Setting any bit to 0, forces the corresponding RESET LOW. This registers allows the software to “Hold” a reset LOW (in a reset condition).

RESET4_LEVEL: 0xFFD01090

The bits of this register correspond to the RESET[n] REGISTERs above. The default of this register is 0xFFFFFFFF. Setting any bit to 0, forces the corresponding RESET LOW. This registers allows the software to “Hold” a reset LOW (in a reset condition).

RESET6_LEVEL: 0xFFD01098

The bits of this register correspond to the RESET[n] REGISTERs above. The default of this register is 0xFFFFFFFF. Setting any bit to 0, forces the corresponding RESET LOW. This registers allows the software to “Hold” a reset LOW (in a reset condition).

RESET7_LEVEL: 0xFFD0109c

The bits of this register correspond to the RESET[n] REGISTERs above. The default of this register is 0xFFFFFFFF. Setting any bit to 0, forces the corresponding RESET LOW. This registers allows the software to “Hold” a reset LOW (in a reset condition).

8.7.2.2 SEC Reset

Reset registers in EE domain are on cbus, so the final address = base_address(0xFF64E000) + address *4;

RESET0_SEC 0x00

Bit(s)	R/W	Default	Description
31-7	R/W	0	Reserved

Bit(s)	R/W	Default	Description
6	R/W	0	ao_mailbox
5	R/W	0	ao
4	R/W	0	dma
3	R/W	0	efuse
2	R/W	0	am_ring_osc
1	R/W	0	m4_cpu
0	R/W	0	ao_cpu

RESET1_SEC 0x01

Bit(s)	R/W	Default	Description
15	R/W	0	sys_core3
14	R/W	0	sys_core2
13	R/W	0	sys_core1
12	R/W	0	sys_core0
11	R/W	0	sys_core3
10	R/W	0	sys_core2
9	R/W	0	sys_core1
8	R/W	0	sys_core0
7	R/W	0	sys_pll_div
6	R/W	0	presetn_sys_cpu_capb3
5	R/W	0	rom_boot
4	R/W	0	syscpu_axi
3	R/W	0	syscpu_l2
2	R/W	0	syscpu_preset
1	R/W	0	syscpu_mbist
0	R/W	0	sys_cpu_cpu

RESET0_SEC_LEVEL: 0xFF64E040

The bits of this register correspond to the RESET[n]_SEC REGISTERs above. The default of this register is 0xFFFFFFFF. Setting any bit to 0, forces the corresponding RESET LOW. This registers allows the software to “Hold” a reset LOW (in a reset condition).

RESET1_SEC_LEVEL: 0xFF64E044

The bits of this register correspond to the RESET[n]_SEC REGISTERs above. The default of this register is 0xFFFFFFFF. Setting any bit to 0, forces the corresponding RESET LOW. This registers allows the software to “Hold” a reset LOW (in a reset condition).

RESET0__SEC_MASK 0xFF64E080

The Bits of this register correspond to the RESET[n]_SEC REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

RESET1_SEC_MASK 0xFF64E084

The Bits of this register correspond to the RESET[n]_SEC REGISTERs above. If a bit is set in this register, then when the watchdog timer fires, that particular module will NOT be reset.

AO Soft Reset

AO_SOFT_RESET 0XFF800040

Bit(s)	R/W	Default	Description
31:24	R/W	0	Reserved
23	R/W	0	ir_out
22	R/W	0	uart2
21	R/W	0	Reserved
20	R/W	0	sar_adc
19	R/W	0	i2c_s
18	R/W	0	i2c_m
17	R/W	0	uart
16	R/W	0	ir_in
15:0	R/W	0	Reserved

A53_SOFT_RESET 0xFF63C160

Bit(s)	R/W	Default	Description
10	R/W	0	syscpu_axi
9	R/W	0	syscpu_preset
8	R/W	0	syscpu_l2
7	R/W	0	sys_core3
6	R/W	0	sys_core2
5	R/W	0	sys_core1
4	R/W	0	sys_core0
3	R/W	0	sys_core3_por

Bit(s)	R/W	Default	Description
2	R/W	0	sys_core2_por
1	R/W	0	sys_core1_por
0	R/W	0	sys_core0_por

A73_SOFT_RESET 0xFF63C204

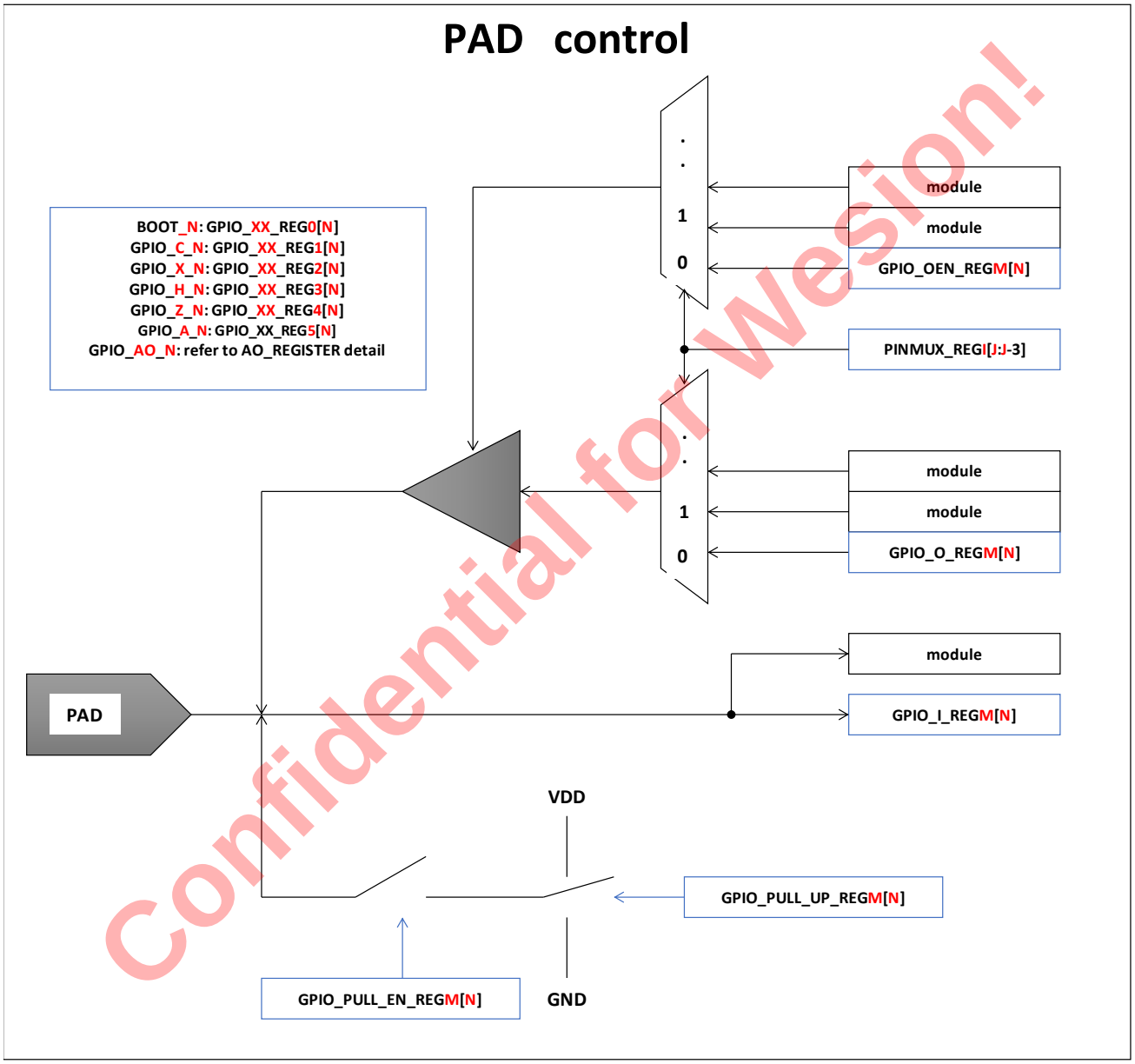
Bit(s)	R/W	Default	Description
10	R/W	0	syscpu_axi
9	R/W	0	syscpu_preset
8	R/W	0	syscpu_l2
7	R/W	0	sys_core3
6	R/W	0	sys_core2
5	R/W	0	sys_core1
4	R/W	0	sys_core0
3	R/W	0	sys_core3_por
2	R/W	0	sys_core2_por
1	R/W	0	sys_core1_por
0	R/W	0	sys_core0_por

8.8 GPIO

8.8.1 Overview

The SOC has a number of multi-function digital I/O pads that can be multiplexed to a number of internal resources (e.g. PWM generators, SDIO controllers). When a digital I/O is not being used for any specific purpose, it is converted to a general purpose GPIO pin. A GPIO pin can be statically set to high/low logical levels. The structure of a GPIO is given below.

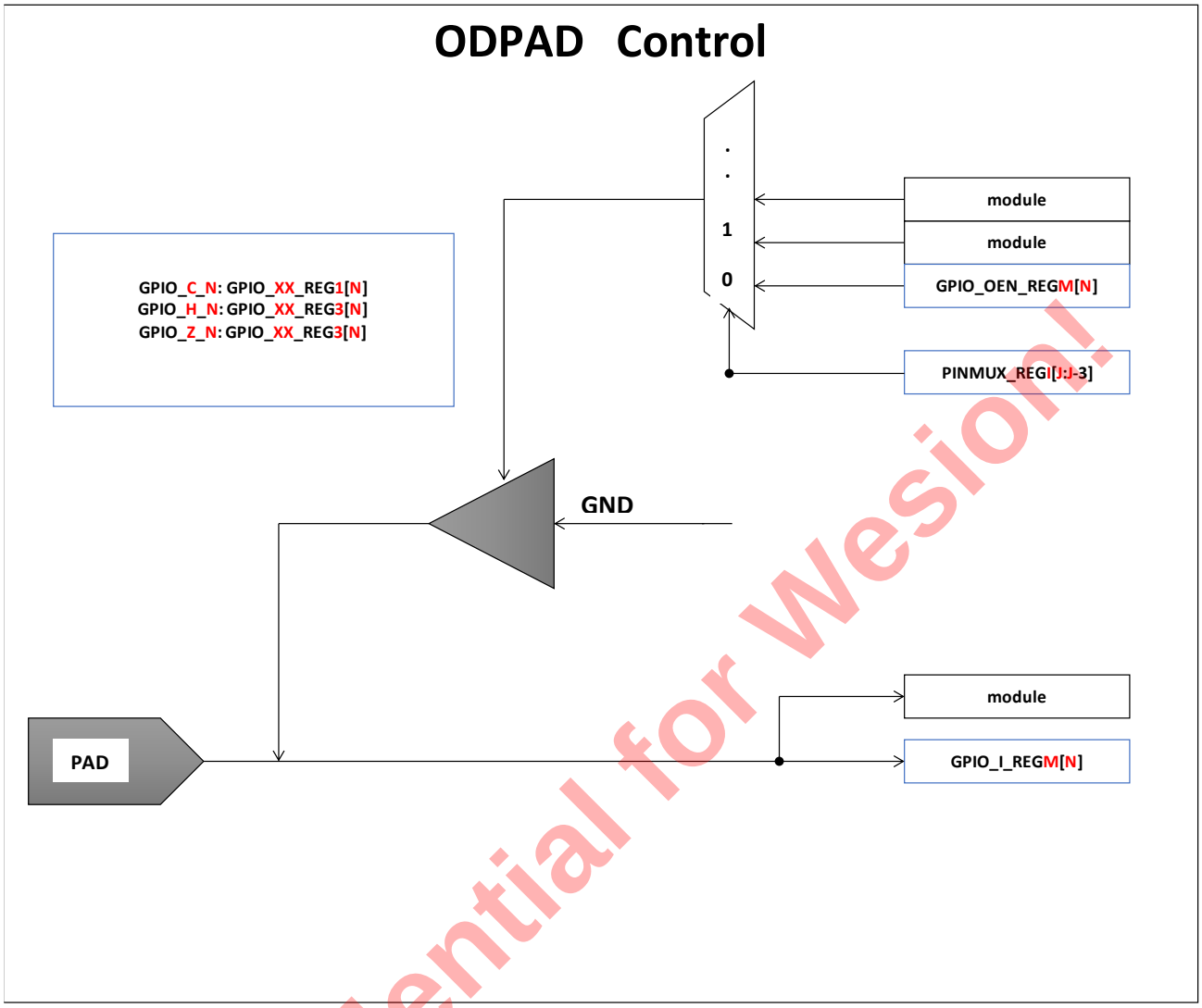
Figure 8-23 GPIO Structure



Each GPIO pad has 6 related registers, among which GPIO_O_REG is used to control the output of the pad, GPIO_I_REG is used to store the input value of the pad; GPIO_OEN_REG is used to enable GPIO output function, PINMUX _REG is used to define the function, GPIO_PULL_EN_REG is used to enable the pull-up function of GPIO pad, and GPIO_PULL_UP_REG is used to set the GPIO pull up/down. For AO GPIO pads, refer to AO GPIO registers.

There are OD pins that can note be used as output, with difference structure, shown as following.

Figure 8-24 OD Pad Structure



8.8.2 GPIO Multiplex Function

The GPIO multiplex functions are shown in the sections below, where the RegNN[MM] corresponds to CBUS registers defined in Table 8-21. The base address is 0xff634400 for PERIPHS_PIN_MUX_X, PREG_PAD_GPIO_EN_N, PREG_PAD_GPIO_O, PREG_PAD_GPIO_I, PAD_PULL_UP_EN_REG, PAD_PULL_UP_REG; and 0xff800000 for AO_RTI_PIN_MUX_REG, the final address is calculated as: final address = base address + offset*4.

Table 8-21 Pin Mux Registers

Pin Mux Registers	Offset
AO_GPIO_I	0x0a
AO_GPIO_O	0x0d
AO_GPIO_O_EN_N	0x09
AO_RTI_PULL_UP_REG	0x0b
AO_RTI_PULL_UP_EN_REG	0x0c

Pin Mux Registers	Offset
AO_PAD_DS_A	0x07
AO_PAD_DS_B	0x08
AO_RTI_PINMUX_REG0	0x05
AO_RTI_PINMUX_REG1	0x06
PERIPHS_PIN_MUX_0	0xb0
PERIPHS_PIN_MUX_1	0xb1
PERIPHS_PIN_MUX_2	0xb2
PERIPHS_PIN_MUX_3	0xb3
PERIPHS_PIN_MUX_4	0xb4
PERIPHS_PIN_MUX_5	0xb5
PERIPHS_PIN_MUX_6	0xb6
PERIPHS_PIN_MUX_7	0xb7
PERIPHS_PIN_MUX_8	0xb8
PERIPHS_PIN_MUX_9	0xb9
PERIPHS_PIN_MUX_A	0xba
PERIPHS_PIN_MUX_B	0xbb
PERIPHS_PIN_MUX_C	0xbc
PERIPHS_PIN_MUX_D	0xbd
PERIPHS_PIN_MUX_E	0xbe
PERIPHS_PIN_MUX_F	0xbf
PAD_DS_REG0A	0xd0
PAD_DS_REG1A	0xd1
PAD_DS_REG2A	0xd2
PAD_DS_REG2B	0xd3
PAD_DS_REG3A	0xd4
PAD_DS_REG4A	0xd5
PAD_DS_REG5A	0xd6
PAD_PULL_UP_REG0	0x3a
PAD_PULL_UP_REG1	0x3b

Pin Mux Registers	Offset
PAD_PULL_UP_REG2	0x3c
PAD_PULL_UP_REG3	0x3d
PAD_PULL_UP_REG4	0x3e
PAD_PULL_UP_REG5	0x3f
PAD_PULL_UP_EN_REG0	0x48
PAD_PULL_UP_EN_REG1	0x49
PAD_PULL_UP_EN_REG2	0x4a
PAD_PULL_UP_EN_REG3	0x4b
PAD_PULL_UP_EN_REG4	0x4c
PAD_PULL_UP_EN_REG5	0x4d
PREG_PAD_GPIO0_EN_N	0x10
PREG_PAD_GPIO0_O	0x11
PREG_PAD_GPIO0_I	0x12
PREG_PAD_GPIO1_EN_N	0x13
PREG_PAD_GPIO1_O	0x14
PREG_PAD_GPIO1_I	0x15
PREG_PAD_GPIO2_EN_N	0x16
PREG_PAD_GPIO2_O	0x17
PREG_PAD_GPIO2_I	0x18
PREG_PAD_GPIO3_EN_N	0x19
PREG_PAD_GPIO3_O	0x1a
PREG_PAD_GPIO3_I	0x1b
PREG_PAD_GPIO4_EN_N	0x1c
PREG_PAD_GPIO4_O	0x1d
PREG_PAD_GPIO4_I	0x1e
PREG_PAD_GPIO5_EN_N	0x20
PREG_PAD_GPIO5_O	0x21
PREG_PAD_GPIO5_I	0x22

Table 8-22 GPIO Bank Z Pin Multiplexing Table

Pin Name	Control Register	Control Bit	Fun1	Func2	Func3	Func4	Func5	Func6	Func7
GPIOZ_0	PERIPHS_PIN_MUX_6 0xff6346d8	[3:0]	ETH_MDIO	BT656_A_VS	ISO7816_CLK	I2C_EE_M0_SDA	PWM_B	-	-
GPIOZ_1		[7:4]	ETH_MDC	BT656_A_HS	ISO7816_DATA	I2C_EE_M0_SCL	PWM_C	-	-
GPIOZ_2		[11:8]	ETH_RGMII_RX_CLK	PWM_D	TSIN_B_VALID	TDMC_D0	SDCARD_D0	TDMC_DIN0	PDM_DIN0
GPIOZ_3		[15:12]	ETH_RX_DV	BT656_A_CLK	TSIN_B_SOP	TDMC_D1	SDCARD_D1	TDMC_DIN1	PDM_DIN1
GPIOZ_4		[19:16]	ETH_RXD0	BT656_A_DIN0	TSIN_B_DIN0	TDMC_D2	SDCARD_D2	TDMC_DIN2	PDM_DIN2
GPIOZ_5		[23:20]	ETH_RXD1	BT656_A_DIN1	TSIN_B_CLK	TDMC_D3	SDCARD_D3	TDMC_DIN3	PDM_DIN3
GPIOZ_6		[27:24]	ETH_RXD2_RGMII	BT656_A_DIN2	TSIN_B_FAIL	TDMC_FS	SDCARD_CLK	TDMC_SLV_FS	PDM_DCLK
GPIOZ_7		[31:28]	ETH_RXD3_RGMII	BT656_A_DIN3	TSIN_B_DIN1	TDMC_SCLK	SDCARD_CMD	TDMC_SLV_SCLK	I2C_EE_M0_SDA
GPIOZ_8	PERIPHS_PIN_MUX_7 0xff6346dc	[3:0]	ETH_RGMII_TX_CLK	BT656_A_DIN4	TSIN_B_DIN2	MCLK_1	-	-	I2C_EE_M0_SCL
GPIOZ_9		[7:4]	ETH_TXEN	BT656_A_DIN5	TSIN_B_DIN3	-	-	-	-
GPIOZ_10		[11:8]	ETH_TXD0	BT656_A_DIN6	TSIN_B_DIN4	-	IR_REMOTE_OUT	-	-
GPIOZ_11		[15:12]	ETH_TXD1	BT656_A_DIN7	TSIN_B_DIN5	-	-	-	-
GPIOZ_12		[19:16]	ETH_TXD2_RGMII	-	TSIN_B_DIN6	-	PWM_F	-	-
GPIOZ_13		[23:20]	ETH_TXD3_RGMII	CLK12_24	TSIN_B_DIN7	-	PWM_B	-	GEN_CLK_EE
GPIOZ_14		[27:24]	ETH_LINK_LED	-	I2C_EE_M2_SDA	-	-	-	-
GPIOZ_15		[31:28]	ETH_ACT_LED	-	I2C_EE_M2_SCL	-	-	-	-

Table 8-23 GPIOA_x Multi-Function Pin

Pin Name	Control Register	Control Bit	Func1	Func2	Func3
GPIOA_0	PERIPHS_PIN_MUX_D 0xff6346f4	[3:0]	MCLK_0	-	-
GPIOA_1		[7:4]	TDMB_SCLK	TDMB_SLV_SCLK	-
GPIOA_2		[11:8]	TDMB_FS	TDMB_SLV_FS	-
GPIOA_3		[15:12]	TDMB_D0	TDMB_DIN0	-
GPIOA_4		[19:16]	TDMB_D1	TDMB_DIN1	PWM_D
GPIOA_5		[23:20]	PDM_DIN3	TDMB_DIN2	TDMB_D2
GPIOA_6		[27:24]	PDM_DIN2	TDMB_DIN3	TDMB_D3
GPIOA_7		[31:28]	PDM_DCLK	TDMC_D3	TDMC_DIN3
GPIOA_8	PERIPHS_PIN_MUX_E 0xff6346f4	[3:0]	PDM_DIN0	TDMC_D2	TDMC_DIN2
GPIOA_9		[7:4]	PDM_DIN1	TDMC_D1	TDMC_DIN1
GPIOA_10		[11:8]	SPDIF_IN	TDMC_D0	TDMC_DIN0
GPIOA_11		[15:12]	SPDIF_OUT	MCLK_1	PWM_F
GPIOA_12		[19:16]	SPDIF_IN	TDMC_SCLK	TDMC_SLV_SCLK
GPIOA_13		[23:20]	SPDIF_OUT	TDMC_FS	TDMC_SLV_FS
GPIOA_14		[27:24]	WORLD_SYNC	I2C_EE_M3_SDA	-
GPIOA_15		[31:28]	IR_REMOTE_IN	I2C_EE_M3_SCL	-

Table 8-24 BOOT_x Multi-Function Pin

Pin Name	Control Register	Control Bit	Func1	Func2	Func3
BOOT_0	PERIPHS_PIN_MUX_0 0xff6346c0	[3:0]	EMMC_D0	-	-
BOOT_1		[7:4]	EMMC_D1	-	-
BOOT_2		[11:8]	EMMC_D2	-	-
BOOT_3		[15:12]	EMMC_D3	-	NOR_HOLD
BOOT_4		[19:16]	EMMC_D4	-	NOR_D
BOOT_5		[23:20]	EMMC_D5	-	NOR_Q
BOOT_6		[27:24]	EMMC_D6	-	NOR_C
BOOT_7		[31:28]	EMMC_D7	-	NOR_WP
BOOT_8	PERIPHS_PIN_MUX_1	[3:0]	EMMC_CLK	NAND_WEN_CLK	-

BOOT_9	0xff6346c4	[7:4]	-	NAND_ALE	-
BOOT_10		[11:8]	EMMC_CMD	NAND_CLE	-
BOOT_11		[15:12]	-	NAND_CE0	-
BOOT_12		[19:16]	-	NAND_REN_WR	-
BOOT_13		[23:20]	EMMC_NAND_DQS	-	-
BOOT_14		[27:24]	-	NAND_RB0	NOR_CS
BOOT_15		[31:28]	-	NAND_CE1	-

Table 8-25 GPIOC_x Multi-Function Pin

Pin Name	Control Register	Control Bit	Func1	Func2	Func3	Func4	Func5
GPIOC_0	PERIPHS_PIN_MUX_9 0xff6346e4	[3:0]	SDCARD_D0	JTAG_B_TDO	-	PDM_DIN0	SPI_A_MOSI
GPIOC_1		[7:4]	SDCARD_D1	JTAG_B_TDI	-	PDM_DIN1	SPI_A_MISO
GPIOC_2		[11:8]	SDCARD_D2	UART_AO_A_RX	-	PDM_DIN2	SPI_A_SS0
GPIOC_3		[15:12]	SDCARD_D3	UART_AO_A_TX	-	PDM_DIN3	SPI_A_SCLK
GPIOC_4		[19:16]	SDCARD_CLK	JTAG_B_CLK	-	PDM_DCLK	PWM_C
GPIOC_5		[23:20]	SDCARD_CMD	JTAG_B_TMS	I2C_EE_M0_SDA	-	ISO7816_CLK
GPIOC_6		[27:24]	-	-	I2C_EE_M0_SCL	-	ISO7816_DATA
GPIOC_7		[31:28]	PCIECK_REQN	WORLD_SYNC		-	-

Table 8-26 GPIOX_x Multi-Function Pin

Pin Name	Control Register	Control Bit	Func1	Func2	Func3	Func4	Func5	Func6	Func7
GPIOX_0	PERIPHS_PIN_MUX_3 0xff6346cc	[3:0]	SDIO_D0	PDM_DIN0	TSIN_A_DIN0	-	-	-	-
GPIOX_1		[7:4]	SDIO_D1	PDM_DIN1	TSIN_A_SOP	-	-	-	-
GPIOX_2		[11:8]	SDIO_D2	PDM_DIN2	TSIN_A_VALID	-	-	-	-

Pin Name	Control Register	Control Bit	Func1	Func2	Func3	Func4	Func5	Func6	Func7
GPIOX_3		[15:12]	SDIO_D3	PDM_DIN3	TSIN_A_CLK	PWM_D	-	-	-
GPIOX_4		[19:16]	SDIO_CLK	PDM_DCLK	-	-	-	-	-
GPIOX_5		[23:20]	SDIO_CMD	MCLK_1	-	PWM_C	-	-	-
GPIOX_6		[27:24]	PWM_A	UART_EE_B_TX	-	PWM_D	-	-	-
GPIOX_7		[31:28]	PWM_F	UART_EE_B_RX	-	PWM_B	-	-	-
GPIOX_8	PERIPHS_PIN_MUX_4 0xff6346d0	[3:0]	TDMA_D1	TDMA_DIN1	TSIN_B_SOP	SPI_A_MOSI	PWM_C	ISO7816_CLK	-
GPIOX_9		[7:4]	TDMA_D0	TDMA_DIN0	TSIN_B_VALID	SPI_A_MISO		ISO7816_DATA	-
GPIOX_10		[11:8]	TDMA_FS	TDMA_SLV_FS	TSIN_B_DIN0	SPI_A_SS0	I2C_EE_M1_SDA	-	-
GPIOX_11		[15:12]	TDMA_SCLK	TDMA_SLV_SCLK	TSIN_B_CLK	SPI_A_SCLK	I2C_EE_M1_SCL	-	-
GPIOX_12		[19:16]	UART_EE_A_TX	-	-	-	-	-	-
GPIOX_13		[23:20]	UART_EE_A_RX	-	-	-	-	-	-
GPIOX_14		[27:24]	UART_EE_A_CTS	-	-	-	-	-	-
GPIOX_15		[31:28]	UART_EE_A_RTS	-	-	-	-	-	-
GPIOX_16	PERIPHS_PIN_MUX_5 0xff6346d4	[3:0]	PWM_E	-	-	-	-	-	-
GPIOX_17		[7:4]	I2C_EE_M2_SDA	-	-	-	-	-	-
GPIOX_18		[11:8]	I2C_EE_M2_SCL	-	-	-	-	-	-
GPIOX_19		[15:12]	PWM_B	WORLD_SYNC	-	-	-	-	GEN_CLK_EE

Table 8-27 GPIOH_x Multi-Function Pin

Pin Name	Control Register	Control Bit	Func1	Func2	Func3	Func4	Func5	Func6
GPIOH_0	PERIPHS_PIN_MUX_B 0xff6346ec	[3:0]	HDMITX_SDA	I2C_EE_M3_SDA	-	-	-	-
GPIOH_1		[7:4]	HDMITX_SCL	I2C_EE_M3_SCL	-	-	-	-
GPIOH_2		[11:8]	HDMITX_HPD_IN	I2C_EE_M1_SDA	-	-	-	-
GPIOH_3		[15:12]	-	I2C_EE_M1_SCL	-	AO_CEC_A	AO_CEC_B	-
GPIOH_4		[19:16]	SPDIF_OUT	UART_EE_C_RTS	SPI_B_MOSI	-	-	-
GPIOH_5		[23:20]	SPDIF_IN	UART_EE_C_CTS	SPI_B_MISO	PWM_F	TDMB_D3	TDMB_DIN3
GPIOH_6		[27:24]	ISO7816_CLK	UART_EE_C_RX	SPI_B_SS0	I2C_EE_M1_SDA	IR_REMOTE_OUT	-
GPIOH_7		[31:28]	ISO7816_DATA	UART_EE_C_TX	SPI_B_SCLK	I2C_EE_M1_SCL	PWM_B	-
GPIOH_8	PERIPHS_PIN_MUX_C 0xff6346f0	[3:0]	-	-	-	-	-	-

Table 8-28 GPIOAO_x Multi-Function Pin

Pin Name	Control Register	Control Bit	Func1	Func2	Func3	Func4	Func5	Func6	Func 7
GPIOAO_0	AO_RTI_PIN MUX_REG0 0xff800014	[3:0]	UART_AO_A_TX	-	-	-	-	-	-
GPIOAO_1		[7:4]	UART_AO_A_RX	-	-	-	-	-	-
GPIOAO_2		[11:8]	I2C_AO_M0_SCL	UART_AO_B_TX	I2C_AO_S0_SCL	-	-	-	-
GPIOAO_3		[15:12]	I2C_AO_M0_SDA	UART_AO_B_RX	I2C_AO_S0_SDA	-	-	-	-
GPIOAO_4		[19:16]	IR_REMOTE_OUT	CLK_32K_IN	PWMAO_C	PWMAO_C_HIZ	TDMB_D0	TDMB_DIN0	-
GPIOAO_5		[23:20]	IR_REMOTE_IN	-	PWMAO_D	-	-	-	-
GPIOAO_6		[27:24]	JTAG_A_CLK	-	PWMAO_C	TSIN_A_SOP	TDMB_D2	TDMB_DIN2	-

Pin Name	Control Register	Control Bit	Func1	Func2	Func3	Func4	Func5	Func6	Func 7
GPIOAO_7		[31:28]	JTAG_A_TMS	-	-	TSIN_A_DIN0	TDMB_FS	TDMB_SLV_FS	-
GPIOAO_8	AO_RTI_PINMUX_REG1 0xff800018	[3:0]	JTAG_A_TDI	-	UART_AO_B_TX	TSIN_A_CLK	TDMB_SCLK	TDMB_SLV_SCLK	-
GPIOAO_9		[7:4]	JTAG_A_TDO	IR_REMOTE_OUT	UART_AO_B_RX	TSIN_A_VALID	MCLK_0	-	-
GPIOAO_10		[11:8]	AO_CEC_A	AO_CEC_B	PWMAO_D	SPDIF_OUT	TDMB_D1	TDMB_DIN1	CLK12_24
GPIOAO_11		[15:12]	-	PWMAO_A_HIZ	PWMAO_A	GEN_CLK_EE	GEN_CLK_AO	-	-

Table 8-29 GPIOE_x Multi-Function Pin

Pin Name	Control Register	Control Bit	Func1	Func2	Func3	Func4
GPIOE_0	AO_RTI_PINMUX_REG1 0xff800018	[19:16]	UART_AO_A_CTS	UART_AO_B_CTS	PWMAO_B	I2C_AO_M0_SCL
GPIOE_1		[23:20]	UART_AO_A_RTS	UART_AO_B_RTS	PWMAO_D	I2C_AO_M0_SDA
GPIOE_2		[27:24]	CLK12_24	CLK25_EE	PWM_A	-

8.8.3 Register Description

Table 8-30 shows the mapping information of GPIO I/O Registers and Nets.

Table 8-30 GPIO Register and Nets

Register	Offset	Net Name
PREG_PAD_GPIO0_EN_N	0x10	BOOT
PREG_PAD_GPIO0_O	0x11	
PREG_PAD_GPIO0_I	0x12	
PAD_PULL_UP_EN_REG0	0x48	
PAD_PULL_UP_REG0	0x3a	
PAD_DS_REG0A	0xd0	
PREG_PAD_GPIO1_EN_N	0x13	GPIOC
PREG_PAD_GPIO1_O	0x14	
PREG_PAD_GPIO1_I	0x15	

Register	Offset	Net Name
PAD_PULL_UP_EN_REG1	0x49	
PAD_PULL_UP_REG1	0x3b	
PAD_DS_REG1A	0xd1	
PREG_PAD_GPIO2_EN_N	0x16	GPIOX
PREG_PAD_GPIO2_O	0x17	
PREG_PAD_GPIO2_I	0x18	
PAD_PULL_UP_EN_REG2	0x4a	
PAD_PULL_UP_REG2	0x3c	
PAD_DS_REG2A	0xd2	
PAD_DS_REG2B	0xd3	
PREG_PAD_GPIO3_EN_N	0x19	GPIOH
PREG_PAD_GPIO3_O	0x1a	
PREG_PAD_GPIO3_I	0x1b	
PAD_PULL_UP_EN_REG3	0x4b	
PAD_PULL_UP_REG3	0x3d	
PAD_DS_REG3A	0xd4	
PREG_PAD_GPIO4_EN_N	0x1c	GPIOZ
PREG_PAD_GPIO4_O	0x1d	
PREG_PAD_GPIO4_I	0x1e	
PAD_PULL_UP_EN_REG4	0x4c	
PAD_PULL_UP_REG4	0x3e	
PAD_DS_REG4A	0xd5	
PREG_PAD_GPIO5_EN_N	0x20	GPIOA
PREG_PAD_GPIO5_O	0x21	
PREG_PAD_GPIO5_I	0x22	
PAD_PULL_UP_EN_REG5	0x4d	
PAD_PULL_UP_REG5	0x3f	
PAD_DS_REG5A	0xd6	

8.8.3.1 Pin MUX Registers

PERIPHS_PIN_MUX_0 0xb0

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for BOOT_0
[7:4]	R/W	0	pin mux select bits for BOOT_1
[11:8]	R/W	0	pin mux select bits for BOOT_2
[15:12]	R/W	0	pin mux select bits for BOOT_3
[19:16]	R/W	0	pin mux select bits for BOOT_4
[23:20]	R/W	0	pin mux select bits for BOOT_5
[27:24]	R/W	0	pin mux select bits for BOOT_6
[31:28]	R/W	0	pin mux select bits for BOOT_7

PERIPHS_PIN_MUX_1 0xb1

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for BOOT_8
[7:4]	R/W	0	pin mux select bits for BOOT_9
[11:8]	R/W	0	pin mux select bits for BOOT_10
[15:12]	R/W	0	pin mux select bits for BOOT_11
[19:16]	R/W	0	pin mux select bits for BOOT_12
[23:20]	R/W	0	pin mux select bits for BOOT_13
[27:24]	R/W	0	pin mux select bits for BOOT_14
[31:28]	R/W	0	pin mux select bits for BOOT_15

PERIPHS_PIN_MUX_2 0xb2

Reserved.

PERIPHS_PIN_MUX_3 0xb3

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOX_0
[7:4]	R/W	0	pin mux select bits for GPIOX_1
[11:8]	R/W	0	pin mux select bits for GPIOX_2
[15:12]	R/W	0	pin mux select bits for GPIOX_3
[19:16]	R/W	0	pin mux select bits for GPIOX_4

[23:20]	R/W	0	pin mux select bits for GPIOX_5
[27:24]	R/W	0	pin mux select bits for GPIOX_6
[31:28]	R/W	0	pin mux select bits for GPIOX_7

PERIPHS_PIN_MUX_4 0xb4

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOX_8
[7:4]	R/W	0	pin mux select bits for GPIOX_9
[11:8]	R/W	0	pin mux select bits for GPIOX_10
[15:12]	R/W	0	pin mux select bits for GPIOX_11
[19:16]	R/W	0	pin mux select bits for GPIOX_12
[23:20]	R/W	0	pin mux select bits for GPIOX_13
[27:24]	R/W	0	pin mux select bits for GPIOX_14
[31:28]	R/W	0	pin mux select bits for GPIOX_15

PERIPHS_PIN_MUX_5 0xb5

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOX_16
[7:4]	R/W	0	pin mux select bits for GPIOX_17
[11:8]	R/W	0	pin mux select bits for GPIOX_18
[15:12]	R/W	0	pin mux select bits for GPIOX_19

PERIPHS_PIN_MUX_6 0xb6

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOZ_0
[7:4]	R/W	0	pin mux select bits for GPIOZ_1
[11:8]	R/W	0	pin mux select bits for GPIOZ_2
[15:12]	R/W	0	pin mux select bits for GPIOZ_3
[19:16]	R/W	0	pin mux select bits for GPIOZ_4
[23:20]	R/W	0	pin mux select bits for GPIOZ_5
[27:24]	R/W	0	pin mux select bits for GPIOZ_6
[31:28]	R/W	0	pin mux select bits for GPIOZ_7

PERIPHS_PIN_MUX_7 0xb7

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOZ_8
[7:4]	R/W	0	pin mux select bits for GPIOZ_9
[11:8]	R/W	0	pin mux select bits for GPIOZ_10
[15:12]	R/W	0	pin mux select bits for GPIOZ_11
[19:16]	R/W	0	pin mux select bits for GPIOZ_12
[23:20]	R/W	0	pin mux select bits for GPIOZ_13
[27:24]	R/W	0	pin mux select bits for GPIOZ_14
[31:28]	R/W	0	pin mux select bits for GPIOZ_15

PERIPHS_PIN_MUX_8 0xb8

Reserved.

PERIPHS_PIN_MUX_9 0xb9

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOC_0
[7:4]	R/W	0	pin mux select bits for GPIOC_1
[11:8]	R/W	0	pin mux select bits for GPIOC_2
[15:12]	R/W	0	pin mux select bits for GPIOC_3
[19:16]	R/W	0	pin mux select bits for GPIOC_4
[23:20]	R/W	0	pin mux select bits for GPIOC_5
[27:24]	R/W	0	pin mux select bits for GPIOC_6
[31:28]	R/W	0	pin mux select bits for GPIOC_7

PERIPHS_PIN_MUX_A 0xba

Reserved.

PERIPHS_PIN_MUX_B 0xbb

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOH_0
[7:4]	R/W	0	pin mux select bits for GPIOH_1
[11:8]	R/W	0	pin mux select bits for GPIOH_2
[15:12]	R/W	0	pin mux select bits for GPIOH_3
[19:16]	R/W	0	pin mux select bits for GPIOH_4

[23:20]	R/W	0	pin mux select bits for GPIOH_5
[27:24]	R/W	0	pin mux select bits for GPIOH_6
[31:28]	R/W	0	pin mux select bits for GPIOH_7

PERIPHS_PIN_MUX_C 0xbc

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOH_8

PERIPHS_PIN_MUX_D 0xbd

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOA_0
[7:4]	R/W	0	pin mux select bits for GPIOA_1
[11:8]	R/W	0	pin mux select bits for GPIOA_2
[15:12]	R/W	0	pin mux select bits for GPIOA_3
[19:16]	R/W	0	pin mux select bits for GPIOA_4
[23:20]	R/W	0	pin mux select bits for GPIOA_5
[27:24]	R/W	0	pin mux select bits for GPIOA_6
[31:28]	R/W	0	pin mux select bits for GPIOA_7

PERIPHS_PIN_MUX_E 0xbe

Bit(s)	R/W	Default	Description
[3:0]	R/W	0	pin mux select bits for GPIOA_8
[7:4]	R/W	0	pin mux select bits for GPIOA_9
[11:8]	R/W	0	pin mux select bits for GPIOA_10
[15:12]	R/W	0	pin mux select bits for GPIOA_11
[19:16]	R/W	0	pin mux select bits for GPIOA_12
[23:20]	R/W	0	pin mux select bits for GPIOA_13
[27:24]	R/W	0	pin mux select bits for GPIOA_14
[31:28]	R/W	0	pin mux select bits for GPIOA_15

PERIPHS_PIN_MUX_F 0xbf

Reserved.

8.8.3.2 Pad pull-up/down Direction

The I/O pads contain both a pull-up and a pull-down. If a bit is set to 1 in the registers below, then the pull-up is enabled. If a bit is set to 0, then the pull-down is enabled.

NOTE: There are separate pull-up “enables” that must also be set to 1 for the pull-up/down direction to function. If an “enable” is set to 0, then the pull-up/down feature is disabled, and the bits below are ignored (on a per pad basis).

PAD_PULL_UP_REG0 0x3a

Bit(s)	R/W	Default	Description
31~16	R/W	0xcfff	Unused
15~0	R/W		Boot[15:0] 1 = pull up. 0 = pull down

PAD_PULL_UP_REG1 0x3b

Bit(s)	R/W	Default	Description
31:7	R/W	0xff	Unused
6~0	R/W		gpioC[6:0] 1 = pull up. 0 = pull down

PAD_PULL_UP_REG2 0x3c

Bit(s)	R/W	Default	Description
31~20	R/W	0x5ffbf	Reserved
19~0	R/W		gpioX[19:0] 1 = pull up. 0 = pull down

PAD_PULL_UP_REG3 0x3d

Bit(s)	R/W	Default	Description
31~8	R/W	0x10f	Reserved
7~4	R/W		gpioH[7:4] 1 = pull up. 0 = pull down
3~0	R/W		Reserved

PAD_PULL_UP_REG4 0x3e

Bit(s)	R/W	Default	Description
31~14	R/W	0xc1ff	Unused
13~0	R/W		gpioZ[13:0] 1 = pull up. 0 = pull down

PAD_PULL_UP_REG5 0x3f

Bit(s)	R/W	Default	Description
31~16	R/W	0xc000	Unused
15~0	R/W		gpioA[15:0] 1 = pull up. 0 = pull down

8.8.3.3 Pad Pull-Up/Down Enables

Each I/O pad has a selectable pull-up or pull-down resistor. For the pull-up direction (up or down) to be operational, the appropriate bit below must be set in order to enable the pull-up/down function.

PAD_PULL_UP_EN_REG0 0x48

Bit(s)	R/W	Default	Description
31~16	R/W	0xffff	Unused
15~0	R/W		boot[15:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down

PAD_PULL_UP_EN_REG1 0x49

Bit(s)	R/W	Default	Description
31~7	R/W	0xff	Unused
6~0	R/W		gpioC[6:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down

PAD_PULL_UP_EN_REG2 0x4a

Bit(s)	R/W	Default	Description
31~20	R/W	0x7ffff	Reserved
19~0	R/W		gpioX[19:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down

PAD_PULL_UP_EN_REG3 0x4b

Bit(s)	R/W	Default	Description
31~8	R/W	0x1ff	reserved
7~4	R/W		gpioH[7:4] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down
3~0	R/W		reserved

PAD_PULL_UP_EN_REG4 0x4c

Bit(s)	R/W	Default	Description
31~14	R/W	0x3fff	reserved
13~0	R/W		gpioZ[13:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down

PAD_PULL_UP_EN_REG5 0x4d

Bit(s)	R/W	Default	Description
31~16	R/W	0xffff	reserved
15~0	R/W		gpioA[15:0] pullup-enable: 1 = pullup or pull-down enabled. 0 = no-pull-up or pull-down

8.8.3.4 PREG_PAD_GPIO_I Registers

PREG_PAD_GPIO0_I Registers 0x12

Bit(s)	R/W	Default	Description
0	R/W	0	BOOT_0 select
1	R/W	0	BOOT_1 select
2	R/W	0	BOOT_2 select
3	R/W	0	BOOT_3 select
4	R/W	0	BOOT_4 select
5	R/W	0	BOOT_5 select
6	R/W	0	BOOT_6 select
7	R/W	0	BOOT_7 select
8	R/W	0	BOOT_8 select
9	R/W	0	BOOT_9 select
10	R/W	0	BOOT_10 select
11	R/W	0	BOOT_11 select
12	R/W	0	BOOT_12 select
13	R/W	0	BOOT_13 select
14	R/W	0	BOOT_14 select
15	R/W	0	BOOT_15 select

PREG_PAD_GPIO1_I Registers 0x15

Bit(s)	R/W	Default	Description
0	R/W	0	GPIOC_0 select
1	R/W	0	GPIOC_1 select
2	R/W	0	GPIOC_2 select
3	R/W	0	GPIOC_3 select
4	R/W	0	GPIOC_4 select
5	R/W	0	GPIOC_5 select
6	R/W	0	GPIOC_6 select
7	R/W	0	GPIOC_7 select

PREG_PAD_GPIO2_I Registers 0x18

Bit(s)	R/W	Default	Description
0	R/W	0	GPIOX_0 select
1	R/W	0	GPIOX_1 select
2	R/W	0	GPIOX_2 select
3	R/W	0	GPIOX_3 select
4	R/W	0	GPIOX_4 select
5	R/W	0	GPIOX_5 select
6	R/W	0	GPIOX_6 select
7	R/W	0	GPIOX_7 select
8	R/W	0	GPIOX_8 select
9	R/W	0	GPIOX_9 select
10	R/W	0	GPIOX_10 select
11	R/W	0	GPIOX_11 select
12	R/W	0	GPIOX_12 select
13	R/W	0	GPIOX_13 select
14	R/W	0	GPIOX_14 select
15	R/W	0	GPIOX_15 select
16	R/W	0	GPIOX_16 select
17	R/W	0	GPIOX_17 select
18	R/W	0	GPIOX_18 select
19	R/W	0	GPIOX_19 select

PREG_PAD_GPIO3_I Registers 0x1b

Bit(s)	R/W	Default	Description
0	R/W	0	GPIOH_0 select
1	R/W	0	GPIOH_1 select
2	R/W	0	GPIOH_2 select
3	R/W	0	GPIOH_3 select
4	R/W	0	GPIOH_4 select
5	R/W	0	GPIOH_5 select
6	R/W	0	GPIOH_6 select
7	R/W	0	GPIOH_7 select
8	R/W	0	GPIOH_8 select

PREG_PAD_GPIO4_I Registers 0x1e

Bit(s)	R/W	Default	Description
0	R/W	0	GPIOZ_0 select
1	R/W	0	GPIOZ_1 select
2	R/W	0	GPIOZ_2 select
3	R/W	0	GPIOZ_3 select
4	R/W	0	GPIOZ_4 select
5	R/W	0	GPIOZ_5 select
6	R/W	0	GPIOZ_6 select
7	R/W	0	GPIOZ_7 select
8	R/W	0	GPIOZ_8 select
9	R/W	0	GPIOZ_9 select
10	R/W	0	GPIOZ_10 select
11	R/W	0	GPIOZ_11 select
12	R/W	0	GPIOZ_12 select
13	R/W	0	GPIOZ_13 select
14	R/W	0	GPIOZ_14 select
15	R/W	0	GPIOZ_15 select

PREG_PAD_GPIO5_I Registers 0x22

Bit(s)	R/W	Default	Description
0	R/W	0	GPIOA_0 select
1	R/W	0	GPIOA_1 select
2	R/W	0	GPIOA_2 select
3	R/W	0	GPIOA_3 select
4	R/W	0	GPIOA_4 select
5	R/W	0	GPIOA_5 select
6	R/W	0	GPIOA_6 select
7	R/W	0	GPIOA_7 select
8	R/W	0	GPIOA_8 select
9	R/W	0	GPIOA_9 select
10	R/W	0	GPIOA_10 select
11	R/W	0	GPIOA_11 select
12	R/W	0	GPIOA_12 select
13	R/W	0	GPIOA_13 select
14	R/W	0	GPIOA_14 select
15	R/W	0	GPIOA_15 select

8.8.3.5 PREG_PAD_GPIO_O Registers

PREG_PAD_GPIO0_O Registers 0x11

Bit(s)	R/W	Default	Description
0	R/W	1	BOOT_0 select
1	R/W	1	BOOT_1 select
2	R/W	1	BOOT_2 select
3	R/W	1	BOOT_3 select
4	R/W	1	BOOT_4 select
5	R/W	1	BOOT_5 select
6	R/W	1	BOOT_6 select
7	R/W	1	BOOT_7 select
8	R/W	1	BOOT_8 select

Bit(s)	R/W	Default	Description
9	R/W	1	BOOT_9 select
10	R/W	1	BOOT_10 select
11	R/W	1	BOOT_11 select
12	R/W	1	BOOT_12 select
13	R/W	1	BOOT_13 select
14	R/W	1	BOOT_14 select
15	R/W	1	BOOT_15 select

PREG_PAD_GPIO1_O Registers 0x14

Bit(s)	R/W	Default	Description
0	R/W	1	GPIOC_0 select
1	R/W	1	GPIOC_1 select
2	R/W	1	GPIOC_2 select
3	R/W	1	GPIOC_3 select
4	R/W	1	GPIOC_4 select
5	R/W	1	GPIOC_5 select
6	R/W	1	GPIOC_6 select

PREG_PAD_GPIO2_O Registers 0x17

Bit(s)	R/W	Default	Description
0	R/W	1	GPIOX_0 select
1	R/W	1	GPIOX_1 select
2	R/W	1	GPIOX_2 select
3	R/W	1	GPIOX_3 select
4	R/W	1	GPIOX_4 select
5	R/W	1	GPIOX_5 select
6	R/W	1	GPIOX_6 select
7	R/W	1	GPIOX_7 select
8	R/W	1	GPIOX_8 select
9	R/W	1	GPIOX_9 select
10	R/W	1	GPIOX_10 select

Bit(s)	R/W	Default	Description
11	R/W	1	GPIOX_11 select
12	R/W	1	GPIOX_12 select
13	R/W	1	GPIOX_13 select
14	R/W	1	GPIOX_14 select
15	R/W	1	GPIOX_15 select
16	R/W	1	GPIOX_16 select
17	R/W	1	GPIOX_17 select
18	R/W	1	GPIOX_18 select
19	R/W	1	GPIOX_19 select

PREG_PAD_GPIO3_O Registers 0x1a

Bit(s)	R/W	Default	Description
4	R/W	1	GPIOH_4 select
5	R/W	1	GPIOH_5 select
6	R/W	1	GPIOH_6 select
7	R/W	1	GPIOH_7 select

PREG_PAD_GPIO4_O Registers 0x1d

Bit(s)	R/W	Default	Description
0	R/W	1	GPIOZ_0 select
1	R/W	1	GPIOZ_1 select
2	R/W	1	GPIOZ_2 select
3	R/W	1	GPIOZ_3 select
4	R/W	1	GPIOZ_4 select
5	R/W	1	GPIOZ_5 select
6	R/W	1	GPIOZ_6 select
7	R/W	1	GPIOZ_7 select
8	R/W	1	GPIOZ_8 select
9	R/W	1	GPIOZ_9 select
10	R/W	1	GPIOZ_10 select
11	R/W	1	GPIOZ_11 select

Bit(s)	R/W	Default	Description
12	R/W	1	GPIOZ_12 select
13	R/W	1	GPIOZ_13 select

PREG_PAD_GPIO5_O Registers 0x21

Bit(s)	R/W	Default	Description
0	R/W	1	GPIOA_0 select
1	R/W	1	GPIOA_1 select
2	R/W	1	GPIOA_2 select
3	R/W	1	GPIOA_3 select
4	R/W	1	GPIOA_4 select
5	R/W	1	GPIOA_5 select
6	R/W	1	GPIOA_6 select
7	R/W	1	GPIOA_7 select
8	R/W	1	GPIOA_8 select
9	R/W	1	GPIOA_9 select
10	R/W	1	GPIOA_10 select
11	R/W	1	GPIOA_11 select
12	R/W	1	GPIOA_12 select
13	R/W	1	GPIOA_13 select
14	R/W	1	GPIOA_14 select
15	R/W	1	GPIOA_15 select

8.8.3.6 PREG PAD_GPIO_EN_N Registers

PREG_PAD_GPIO0_EN_N 0x10

Bit(s)	R/W	Default	Description
0	R/W	1	BOOT_0 select
1	R/W	1	BOOT_1 select
2	R/W	1	BOOT_2 select
3	R/W	1	BOOT_3 select
4	R/W	1	BOOT_4 select

Bit(s)	R/W	Default	Description
5	R/W	1	BOOT_5 select
6	R/W	1	BOOT_6 select
7	R/W	1	BOOT_7 select
8	R/W	1	BOOT_8 select
9	R/W	1	BOOT_9 select
10	R/W	1	BOOT_10 select
11	R/W	1	BOOT_11 select
12	R/W	1	BOOT_12 select
13	R/W	1	BOOT_13 select
14	R/W	1	BOOT_14 select
15	R/W	1	BOOT_15 select

PREG_PAD_GPIO1_EN_N 0x13

Bit(s)	R/W	Default	Description
0	R/W	1	GPIOC_0 select
1	R/W	1	GPIOC_1 select
2	R/W	1	GPIOC_2 select
3	R/W	1	GPIOC_3 select
4	R/W	1	GPIOC_4 select
5	R/W	1	GPIOC_5 select
6	R/W	1	GPIOC_6 select
7	R/W	1	GPIOC_7 select

PREG_PAD_GPIO2_EN_N 0x16

Bit(s)	R/W	Default	Description
0	R/W	1	GPIOX_0 select
1	R/W	1	GPIOX_1 select
2	R/W	1	GPIOX_2 select
3	R/W	1	GPIOX_3 select
4	R/W	1	GPIOX_4 select
5	R/W	1	GPIOX_5 select

Bit(s)	R/W	Default	Description
6	R/W	1	GPIOX_6 select
7	R/W	1	GPIOX_7 select
8	R/W	1	GPIOX_8 select
9	R/W	1	GPIOX_9 select
10	R/W	1	GPIOX_10 select
11	R/W	1	GPIOX_11 select
12	R/W	1	GPIOX_12 select
13	R/W	1	GPIOX_13 select
14	R/W	1	GPIOX_14 select
15	R/W	1	GPIOX_15 select
16	R/W	1	GPIOX_16 select
17	R/W	1	GPIOX_17 select
18	R/W	1	GPIOX_18 select
19	R/W	1	GPIOX_19 select

PREG_PAD_GPIO3_EN_N 0x19

Bit(s)	R/W	Default	Description
0	R/W	1	GPIOH_0 select
1	R/W	1	GPIOH_1 select
2	R/W	1	GPIOH_2 select
3	R/W	1	GPIOH_3 select
4	R/W	1	GPIOH_4 select
5	R/W	1	GPIOH_5 select
6	R/W	1	GPIOH_6 select
7	R/W	1	GPIOH_7 select
8	R/W	1	GPIOH_8 select

PREG_PAD_GPIO4_EN_N 0x1c

Bit(s)	R/W	Default	Description
0	R/W	1	GPIOZ_0 select
1	R/W	1	GPIOZ_1 select

Bit(s)	R/W	Default	Description
2	R/W	1	GPIOZ_2 select
3	R/W	1	GPIOZ_3 select
4	R/W	1	GPIOZ_4 select
5	R/W	1	GPIOZ_5 select
6	R/W	1	GPIOZ_6 select
7	R/W	1	GPIOZ_7 select
8	R/W	1	GPIOZ_8 select
9	R/W	1	GPIOZ_9 select
10	R/W	1	GPIOZ_10 select
11	R/W	1	GPIOZ_11 select
12	R/W	1	GPIOZ_12 select
13	R/W	1	GPIOZ_13 select
14	R/W	1	GPIOZ_14 select
15	R/W	1	GPIOZ_15 select

PREG_PAD_GPIO5_EN_N 0x20

Bit(s)	R/W	Default	Description
0	R/W	1	GPIOA_0 select
1	R/W	1	GPIOA_1 select
2	R/W	1	GPIOA_2 select
3	R/W	1	GPIOA_3 select
4	R/W	1	GPIOA_4 select
5	R/W	1	GPIOA_5 select
6	R/W	1	GPIOA_6 select
7	R/W	1	GPIOA_7 select
8	R/W	1	GPIOA_8 select
9	R/W	1	GPIOA_9 select
10	R/W	1	GPIOA_10 select
11	R/W	1	GPIOA_11 select
12	R/W	1	GPIOA_12 select

Bit(s)	R/W	Default	Description
13	R/W	1	GPIOA_13 select
14	R/W	1	GPIOA_14 select
15	R/W	1	GPIOA_15 select

8.8.3.7 PAD_DS Registers

PAD_DS_REG0A 0xd0

Bit(s)	R/W	Default	Description
[1:0]	R/W	0xaaaaaaaa	BOOT_0 select
[3:2]	R/W		BOOT_1 select
[5:4]	R/W		BOOT_2 select
[7:6]	R/W		BOOT_3 select
[9:8]	R/W		BOOT_4 select
[11:10]	R/W		BOOT_5 select
[13:12]	R/W		BOOT_6 select
[15:14]	R/W		BOOT_7 select
[17:16]	R/W		BOOT_8 select
[19:18]	R/W		BOOT_9 select
[21:20]	R/W		BOOT_10 select
[23:22]	R/W		BOOT_11 select
[25:24]	R/W		BOOT_12 select
[27:26]	R/W		BOOT_13 select
[29:28]	R/W		BOOT_14 select
[31:30]	R/W		BOOT_15 select

PAD_DS_REG1A 0xd1

Bit(s)	R/W	Default	Description
[1:0]	R/W	0xaaaa9aaa	GPIOC_0 select
[3:2]	R/W		GPIOC_1 select
[5:4]	R/W		GPIOC_2 select
[7:6]	R/W		GPIOC_3 select
[9:8]	R/W		GPIOC_4 select

Bit(s)	R/W	Default	Description
[11:10]	R/W		GPIOC_5 select
[13:12]	R/W		GPIOC_6 select

PAD_DS_REG2A 0xd2

Bit(s)	R/W	Default	Description
[1:0]	R/W	0x55955aaa	GPIOX_0 select
[3:2]	R/W		GPIOX_1 select
[5:4]	R/W		GPIOX_2 select
[7:6]	R/W		GPIOX_3 select
[9:8]	R/W		GPIOX_4 select
[11:10]	R/W		GPIOX_5 select
[13:12]	R/W		GPIOX_6 select
[15:14]	R/W		GPIOX_7 select
[17:16]	R/W		GPIOX_8 select
[19:18]	R/W		GPIOX_9 select
[21:20]	R/W		GPIOX_10 select
[23:22]	R/W		GPIOX_11 select
[25:24]	R/W		GPIOX_12 select
[27:26]	R/W		GPIOX_13 select
[29:28]	R/W		GPIOX_14 select
[31:30]	R/W		GPIOX_15 select

PAD_DS_REG2B 0xd3

Bit(s)	R/W	Default	Description
[1:0]	R/W	0xaaaaaa55	GPIOX_16 select
[3:2]	R/W		GPIOX_17 select
[5:4]	R/W		GPIOX_18 select
[7:6]	R/W		GPIOX_19 select

PAD_DS_REG3A 0xd4

Bit(s)	R/W	Default	Description
[9:8]	R/W	0xaaaa55aa	GPIOH_4 select
[11:10]	R/W		GPIOH_5 select
[13:12]	R/W		GPIOH_6 select
[15:14]	R/W		GPIOH_7 select

PAD_DS_REG4A 0xd5

Bit(s)	R/W	Default	Description
[1:0]	R/W	0xaaaaaaa5	GPIOZ_0 select
[3:2]	R/W		GPIOZ_1 select
[5:4]	R/W		GPIOZ_2 select
[7:6]	R/W		GPIOZ_3 select
[9:8]	R/W		GPIOZ_4 select
[11:10]	R/W		GPIOZ_5 select
[13:12]	R/W		GPIOZ_6 select
[15:14]	R/W		GPIOZ_7 select
[17:16]	R/W		GPIOZ_8 select
[19:18]	R/W		GPIOZ_9 select
[21:20]	R/W		GPIOZ_10 select
[23:22]	R/W		GPIOZ_11 select
[25:24]	R/W		GPIOZ_12 select
[27:26]	R/W		GPIOZ_13 select

PAD_DS_REG5A 0xd6

Bit(s)	R/W	Default	Description
[1:0]	R/W	0x5695555a	GPIOA_0 select
[3:2]	R/W		GPIOA_1 select
[5:4]	R/W		GPIOA_2 select
[7:6]	R/W		GPIOA_3 select
[9:8]	R/W		GPIOA_4 select
[11:10]	R/W		GPIOA_5 select

Bit(s)	R/W	Default	Description
[13:12]	R/W		GPIOA_6 select
[15:14]	R/W		GPIOA_7 select
[17:16]	R/W		GPIOA_8 select
[19:18]	R/W		GPIOA_9 select
[21:20]	R/W		GPIOA_10 select
[23:22]	R/W		GPIOA_11 select
[25:24]	R/W		GPIOA_12 select
[27:26]	R/W		GPIOA_13 select
[29:28]	R/W		GPIOA_14 select
[31:30]	R/W		GPIOA_15 select

8.8.3.8 GPIOAO Registers

AO_GPIO_I 0x0a

Always On GPIO Input levels

Bit(s)	R/W	Default	Description
31	R	0	TEST_N input level
18:16	R	0	INPUT_LEVELS: These bits correspond to the INPUT levels on the gpioE[2:0] pins
11:0	R	0	INPUT_LEVELS: These bits correspond to the INPUT levels on the gpioAO[11:0] pins

AO_GPIO_O0x0d

Always On GPIO controls.

NOTE: This register is NOT reset during a watchdog event.

Bit(s)	R/W	Default	Description
31	R/W	0xffffffff	TEST_N_OUTPUT_LEVEL: This bit controls the output level of the test_n pin when TEST_N_GPIO_EN_N is set to 0.
18:16	R/W		OUTPUT_LEVEL: These bits correspond to the output levels on the gpioE[2:0] pins when in GPIO mode.
11:0	R/W		OUTPUT_LEVEL: These bits correspond to the output levels on the gpioAO[11:0] pins when in GPIO mode.

AO_RTI_PINMUX_REG0 0x05

Bit(s)	R/W	Default	Description
31:28	R/W	0	gpioAO_7 select

27:24	R/W	0	gpioAO_6 select
23:20	R/W	0	gpioAO_5 select
19:16	R/W	0	gpioAO_4 select
15:12	R/W	0	gpioAO_3 select
11:8	R/W	0	gpioAO_2 select
7:4	R/W	0	gpioAO_1 select
3:0	R/W	0	gpioAO_0 select

AO_RTI_PINMUX_REG1 0x06

Bit(s)	R/W	Default	Description
31:28	R/W	0	gpioAO TEST_N
27:24	R/W	0	gpioE_2 select
[23:20]	R/W	0	gpioE_1 select
[19:16]	R/W	0	gpioE_0 select
[15:12]	R/W	0	gpioAO_11 select
[11:8]	R/W	0	gpioAO_10 select
[7:4]	R/W	0	gpioAO_9 select
[3:0]	R/W	0	gpioAO_8 select

AO_GPIO_O_EN_N 0x09

Always On GPIO controls.

NOTE: This register is NOT reset during a watchdog event.

Bit(s)	R/W	Default	Description
30-19	R	0xffffffff	Reserved
18-16	R/W		OUTPUT_ENABLE: These bits correspond to the output levels on the gpioE[2:0] pins when in GPIO mode. A ‘0’ sets the gpioE pin to be an output.
15-12	R		Reserved
11-0	R/W		OUTPUT_ENABLE: These bits correspond to the output levels on the gpioAO[11:0] pins when in GPIO mode. A ‘0’ sets the gpioAO pin to be an output.

AO_GPIO_I 0x0a

Always On GPIO Input levels

Bit(s)	R/W	Default	Description
31	R	0	TEST_N input level

18:16	R	0	INPUT_LEVELS: These bits correspond to the INPUT levels on the gpioE[2:0] pins
11-0	R	0	INPUT_LEVELS: These bits correspond to the INPUT levels on the gpioAO[11:0] pins

AO_RTI_PULL_UP_REG 0x0b

Bit(s)	R/W	Default	Description
31	R/W	0x800005ab	TEST_N pull up/down. 1 = pull-up 0 = pull-down
18-16	R/W		gpioE[2:0] pull-up/down. 1 = pull-up 0 = pull-down
11-0	R/W		gpioAO[11:0] pull up/down. 1 = pull-up 0 = pull-down

AO_RTI_PULL_UP_EN_REG 0x0c

Always On GPIO controls.

NOTE: This register is NOT reset during a watchdog event.

Bit(s)	R/W	Default	Description
31	R/W	0x80040fff	TEST_N pull-up/down enable. 1 = pull-up/down enable, 0 = pull-up/down disable
18-16	R/W		gpioE[2:0] pull-up/down enable. 1 = pull-up/down enable, 0 = pull-up/down disable
11-0	R/W		gpioAO[11:0] pull-up/down enable. 1 = pull-up/down enable, 0 = pull-up/down disable

AO_PAD_DS_A 0x07

Pad ds0/ds1

Bit(s)	R/W	Default	Description
23:22	R/W	0xaaaaaaaa	gpioAO_11{ds1 ds0}
21:20	R/W		gpioAO_10{ds1 ds0}
19:18	R/W		gpioAO_9{ds1 ds0}
17:16	R/W		gpioAO_8{ds1 ds0}
15:14	R/W		gpioAO_7{ds1 ds0}
13:12	R/W		gpioAO_6{ds1 ds0}
11:10	R/W		gpioAO_5{ds1 ds0}
9:8	R/W		gpioAO_4{ds1 ds0}
7:6	R/W		gpioAO_3{ds1 ds0}
5:4	R/W		gpioAO_2{ds1 ds0}
3:2	R/W		gpioAO_1{ds1 ds0}
1:0	R/W		gpioAO_0 {ds1 ds0}

AO_PAD_DS_B 0x08

Pad ds0/ds1

Bit(s)	R/W	Default	Description
31:30	R/W	0xaaaaaaaa	reset_n {ds1, ds0}
29:28	R/W		test_n {ds1, ds0}
5:4	R/W		gpioE_2 {ds1 ds0}
3:2	R/W		gpioE_1 {ds1 ds0}
1:0	R/W		gpioE_0 {ds1 ds0}

AO_PINMUX_LOCK 0x17

Bit(s)	R/W	Default	Description
31	R/W	0	1: generate error when write to each locked register;
7	R/W	0	lock ao gpio_o
6	R/W	0	lock ao gpio pull en
5	R/W	0	lock ao gpio pull up
4	R/W	0	lock ao gpio_o_en
1	R/W	0	lock ao pin_mux_reg1
0	R/W	0	lock ao pin_mux_reg0

8.9 Interrupt Control

8.9.1 Overview

Generic Interrupt Controller (GIC) is a centralized resource that supports and manages interrupts in a system. For more details about GIC, please refer to the ARM GIC Architecture Specification V2.0.

8.9.2 Interrupt Source

There are 10 interrupts that are routed to all CPUs (A53/SCP..). The tables below outline the interrupts associated with each interrupt group:

There are 224 interrupt sources in the chip. All of the interrupts are connected to the integrated GIC in Cortex-A53 while the AO-CPU see a sub-set of the interrupts. The control Bits of AO-CPU interrupt are listed in the following table.

Table 8-31 EE Interrupt Source

A53 GIC Bit	Interrupt Source	Description
255	pcie_A_7	
254	pcie_A_6	

A53 GIC Bit	Interrupt Source	Description
253	pcie_A_5	
252	pcie_A_4	
251	pcie_A_3	
250	pcie_A_2	
249	pcie_A_1	
248	pcie_A_0	
247	m_i2c_2_irq	
246	m_i2c_1_irq	
245	mbox_irq_send5	
244	mbox_irq_send4	
243	mbox_irq_send3	
242	mbox_irq_receiv2	
241	mbox_irq_receiv1	
240	mbox_irq_receiv0	
239	ao_gpio_irq1	
238	ao_gpio_irq0	
237	ao_timerB_irq	
236	ao_timerA_irq	
235	cecb_irq	
234	ao_watchdog_irq	
233	ao_m_i2c_to_irq	
232	sar_adc_irq	
231	ao_cec_irq	
230	ao_ir_blaster_irq	
229	ao_uart2_irq	
228	ao_ir_dec_irq	
227	ao_i2c_m_irq	
226	ao_i2c_s_irq	
225	ao_uart_irq	
224	1'b0	
223	SD_EMMC_C	
222	SD_EMMC_B	

A53 GIC Bit	Interrupt Source	Description
221	SD_EMMC_A	
220	HEVC Encoder_VPU_IDLE	
219	HEVC Encoder	
218		
217	dma_irq[5]	
216	dma_irq[4]	
215	dma_irq[3]	
214	dma_irq[2]	
213	dma_irq[1]	
212	dma_irq[0]	
211	eth_phy_irq8	
210	eth_phy_irq7	
209	eth_phy_irq6	
208	eth_phy_irq5	
207	eth_phy_irq4	
206	eth_phy_irq3	
205	eth_phy_irq2	
204	eth_phy_irq1	
203	eth_phy_irq0	
202	mali_irq_pp3	reserved
201	mali_irq_ppmmu2	reserved
200	mali_irq_pp2	reserved
199	mali_irq_ppmmu1	reserved
198	mali_irq_pp1	reserved
197	mali_irq_ppmmu0	reserved
196	mali_irq_pp0	reserved
195	mali_irq_pmu	
194	mali_irq_pp	
193	mali_irq_gpmmu	
192	mali_irq_gp	
191	PCIE_A_EDMA_RD	
190	PCIE_A_EDMA_WR	

A53 GIC Bit	Interrupt Source	Description
189	AUDIO_IRQ9	reserved
188	AUDIO_IRQ8	toram
187	AUDIO_IRQ7	power_detect
186	AUDIO_IRQ6	frddr_C
185	AUDIO_IRQ5	frddr_B
184	AUDIO_IRQ4	frddr_A
183	AUDIO_IRQ3	spdifin
182	AUDIO_IRQ2	toddr_C
181	AUDIO_IRQ1	toddr_B
180	AUDIO_IRQ0	toddr_A
179	1'b0	
178	ge2d	
177	cusad	
176	1'b0	
175	viu1_wm_int	
174	1'b0	
173	A53irq[4]	EXTERRIRQ_a
172	A53irq[3]	|CTIIRQ[3:0]
171	A53irq[2]	|VCPUMNTIRQ_a[3:0]
170	A53irq[1]	|COMMIRQ_a[3:0]
169	A53irq[0]	|PMUIRQ_a[3:0]
168	mbox_rec7	
167	mbox_sed6	
166	mipi_dsi_tear	
165	mipi_dsi_err	
164	viu1_line	
163	asssit_mbox_irq3	
162	asssit_mbox_irq2	
161	asssit_mbox_irq1	
160	asssit_mbox_irq0	
127	m_i2c_3_TO	
126	m_i2c_2_TO	

A53 GIC Bit	Interrupt Source	Description
125	uart2_irq	
124	m_i2c_1_TO	
123	m_i2c_0_TO	
122	spicc_1_int	
121	rdma_done_int	
120	1'b0	
119	1'b0	
118	vid1_wr_irq	
117	vdin1_vsync_int	
116	vdin1_hsync_int	
115	vdin0_vsync_int	
114	vdin0_hsync_int	
113	spicc_0_int	
112	spi_int	
111	vid0_wr_irq	
110	venc_vx1_int	
109	viu1_dolby_int	
108	viu1_mail_afbc	
107	uart1_irq	
106	1'b0	
105	1'b0	
104	1'b0	
103	gpio_irq[7]	
102	gpio_irq[6]	
101	gpio_irq[5]	
100	gpio_irq[4]	
99	gpio_irq[3]	
98	gpio_irq[2]	
97	gpio_irq[1]	
96	gpio_irq[0]	
95	TimerI	TimerI
94	TimerH	TimerH

A53 GIC Bit	Interrupt Source	Description
93	TimerG	TimerG
92	TimerF	TimerF
91	viu2_line	
90	htx_hdcp22_intr	
89	hdmitx_interrupt	
88	viu2_vsync	
87	viu2_hsync	
86	dmc_test_irq	
85	demux_int_2	
84	dmc_irq	
83	dmc_sec_irq	
82	1'b0	
81	1'b0	
80	1'b0	
79	1'b0	
78	di_pre	Reserved for Deinterlacer
77	dos_mbox_slow_irq[2]	DOS Mailbox 2
76	dos_mbox_slow_irq[1]	DOS Mailbox 1
75	dos_mbox_slow_irq[0]	DOS Mailbox 0
74	1'b0	
73	1'b0	
72	di_post	
71	m_i2c_3_irq	I2C Master #3
70	1'b0	
69	smartcard_irq	
68	ts_ddr	
67	ts_pll	
66	nand_irq	
65	viff_empty_int_cpu	
64	parser_int_cpu	
63	U2d_interrupt	USB
62	U3h_interrupt	USB

A53 GIC Bit	Interrupt Source	Description
61	Timer D	Timer D
60	bus_mon1_fast_irq	
59	bus_mon0_fast_irq	
58	Uart_EE_A_irq	
57	async_fifo2_flush_irq	
56	async_fifo2_fill_irq	
55	demux_int	
54	encif_irq	
53	m_i2c_0_irq	
52	bt656	
51	async_fifo_flush_irq	
50	async_fifo_fill_irq	
49	1'b0	
48	usb_iddig_irq	
47	1'b0	unused
46	eth_lip_intro_o	
45	1'b0	
44	mipi_dsi_phy	
43	Timer B	Timer B
42	Timer A	Timer A
41	eth_phy_irq_or	
40	eth_gmac_int	
39	1'b0	
38	Timer C	Timer C
37	demux_int_1	
36	eth_pmt_intr_o	
35	viu1_vsync_int	VSYNC
34	viu1_hsync_int	HSYNC
33	audio_locker	
32	ee_wd_irq	Watchdog Timer

GPIO Interrupt

There are 10 GPIO interrupts, SCP can respond to all of them, while A53 can respond to 8 of them(listed in Table 8-32, GIC bit 96~103). The figure below outline the interrupts associated with each interrupt group:

Figure 8-25 S922X Interrrupt

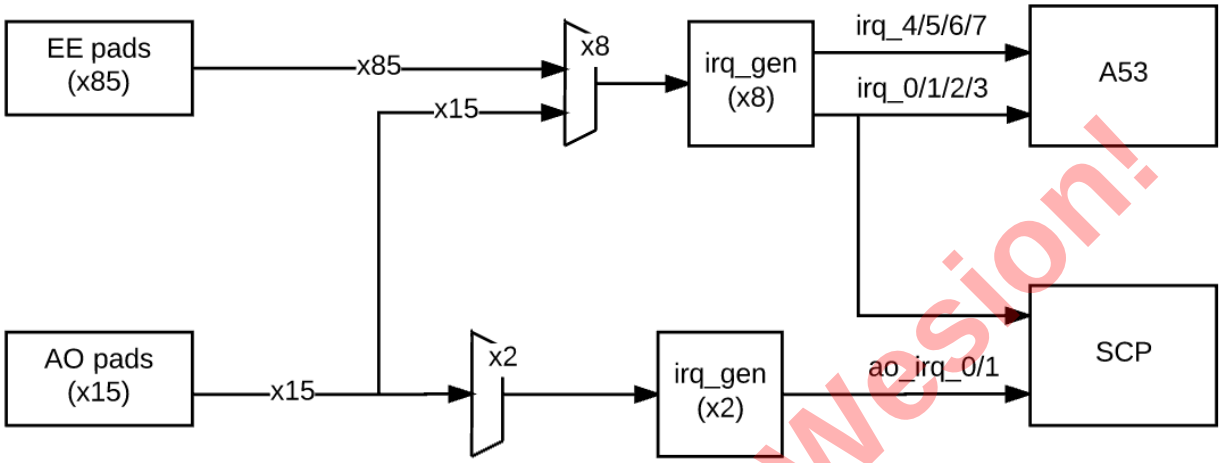
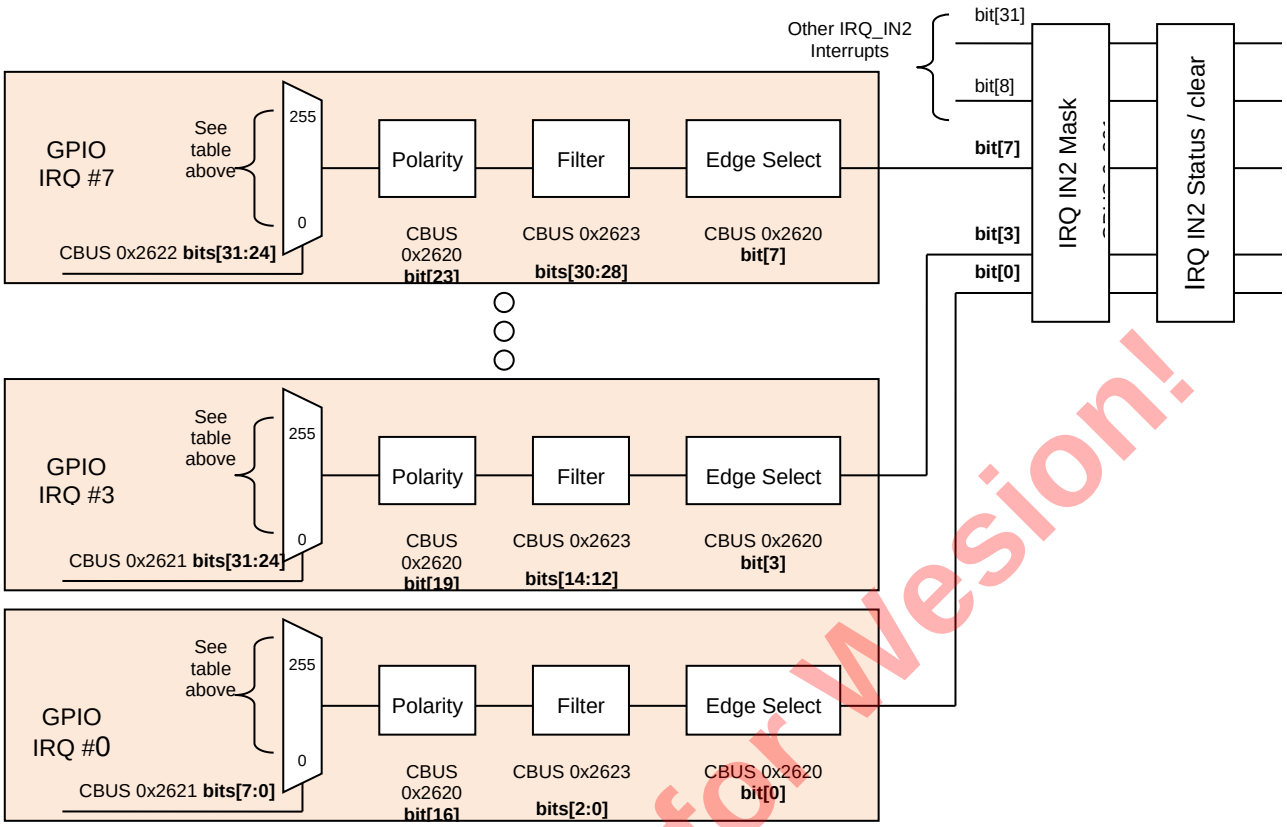


Table 8-32 GPIO Interrupt Sources

Input Mux Location **CBUS registers 0x2621 and 0x2622**	**Description**
[223:100]	Undefined(no interrupt)
[99:97]	gpioE[2:0]
[96:77]	gpioX[19:0]
[76:61]	gpioA[15:0]
[60:53]	gpioC[7:0]
[52:37]	Boot[15:0]
[36:28]	gpioH[8:0]
[27:12]	gpioZ[15:0]
[11:0]	gpioAO[11:0]

The diagram below illustrates the path a GPIO takes to become an interrupt. The eight GPIO interrupts respond to the MASK, STATUS and STATUS/CLEAR registers just like any other interrupt in the chip. The difference for the GPIO interrupts is that they can be filtered and conditioned.

Figure 8-26 GPIO Interrupt Path



NOTE: The input for the GPIO interrupt module (the input into the 256:1 mux) comes directly from the I/O pad of the chip. Therefore, if a pad (say gpioA_11) is configured as a UART TX pin, then in theory, the UART TX pin can be a GPIO interrupt since the TX pin will drive gpioA_11 which in turn can drive the GPIO interrupt module.

8.9.3 Register Description

Each register final address = 0xffd00000 + address * 4

AO_IRQ_GPIO_REG 0x21

Interrupt masking and FIQ select

Bit(s)	R/W	Default	Description
31-20	R	0	Unused
19	R/W	0	GPIO_1_IRQ_EDGE: If this bit is 1, then GPIO_1_IRQ generates an interrupt on the rising (or falling edge if GPIO_1_POL is set) of the selected interrupt.
18	R/W	0	GPIO_0_IRQ_EDGE: If this bit is 1, then GPIO_0_IRQ generates an interrupt on the rising (or falling edge if GPIO_0_POL is set) of the selected interrupt.
17	R/W	0	GPIO_1_POL: This bit controls the polarity of the GPIO muxed into the GPIO_1 interrupt filter module
16	R/W	0	GPIO_0_POL: This bit controls the polarity of the GPIO muxed into the GPIO_0 interrupt filter module

Bit(s)	R/W	Default	Description
15	R/W	0	GPIO_1_FILTER_USE_CLK: Setting this bit to 1 connects the input filter to the system clock (the clock used by the media CPU). If this bit is 0, then the filter uses a 125nS clock to filter the GPIO_1 input
14-12	R/W	0	GPIO_1_FILTER_SEL: 0 = no filter, 1 = max filter (about 2.6uS)
11	R/W	0	GPIO_0_FILTER_USE_CLK
10-8	R/W	0	GPIO_0_FILTER_SEL: 0 = no filter, 1 = max filter (about 2.6uS)
7-4	R/W	0	GPIO_1_INPUT_SEL: These bits select which gpioAO[11:0] pin is connected to the GPIO_1 interrupt filter module.
3-0	R/W	0	GPIO_0_INPUT_SEL: These bits select which gpioAO[11:0] pin is connected to the GPIO_1 interrupt filter module.

GPIO Interrupt EDGE and Polarity 0x3c20

This register controls the polarity of the GPIO interrupts and whether or not the interrupts are level or edge triggered. There are 8 GPIO interrupts. These 8 GPIO interrupts can be assigned to any one of up to 256 pins on the chip.

Bit(s)	R/W	Default	Description
31-24	R	0	unused
23			GPIO_POLARITY_PATH_7: If a bitin this field is 1, then the GPIO signal for GPIO interrupt path 7 is inverted.
22			GPIO_POLARITY_PATH_6:
21			GPIO_POLARITY_PATH_5:
20			GPIO_POLARITY_PATH_4:
19			GPIO_POLARITY_PATH_3:
18			GPIO_POLARITY_PATH_2:
17			GPIO_POLARITY_PATH_1:
16	R/W	0	GPIO_POLARITY_PATH_0:
15-8	R	0	Unused
7	R/W		GPIO_EDGE_SEL_PATH_7: If a bit is set to 1, then the GPIO interrupt for GPIO path 7 is configured to be an edge generated interrupt. If the polarity (above) is 0, then the interrupt is generated on the rising edge. If the polarity is 1, then the interrupt is generated on the falling edge of the GPIO. If a bit is this field is 0, then the GPIO is a level interrupt.
6	R/W		GPIO_EDGE_SEL_PATH_6
5	R/W		GPIO_EDGE_SEL_PATH_5
4	R/W		GPIO_EDGE_SEL_PATH_4
3	R/W		GPIO_EDGE_SEL_PATH_3

Bit(s)	R/W	Default	Description
2	R/W		GPIO_EDGE_SEL_PATH_2
1	R/W		GPIO_EDGE_SEL_PATH_1
0	R/W	0	GPIO_EDGE_SEL_PATH_0

GPIO 0 ~ 3 Pin Select 0x3c21

Each GPIO interrupt can select from any number of up to 256 GPIO pins on the chip. The Bits below control the pin selection for GPIO interrupts 0 ~3.

Bit(s)	R/W	Default	Description
31-24	R/W	0	GPIO_PIN_SEL3: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 3
23-16	R/W	0	GPIO_PIN_SEL2: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 2
15-8	R/W	0	GPIO_PIN_SEL1: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 1
7-0	R/W	0	GPIO_PIN_SEL0: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 0

GPIO 4 ~ 7 Pin Select 0x3c22

Bit(s)	R/W	Default	Description
31-24	R/W	0	GPIO_PIN_SEL7: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 7
23-16	R/W	0	GPIO_PIN_SEL6: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 6
15-8	R/W	0	GPIO_PIN_SEL5: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 5
7-0	R/W	0	GPIO_PIN_SEL4: This value select which of up to 256 pins on the chip can be mapped to GPIO interrupt 4

GPIO Filter Select (interrupts 0~7) 0x3c23

Bit(s)	R/W	Default	Description
31	R/W	0	unused
30-28	R/W	0	FILTER_SEL7: (see FILTER_SEL0)
27	R/W	0	Unused
26-24	R/W	0	FILTER_SEL6: (see FILTER_SEL0)

Bit(s)	R/W	Default	Description
23	R/W	0	Unused
22-20	R/W	0	FILTER_SEL5: (see FILTER_SEL0)
19	R/W	0	Unused
18-16	R/W	0	FILTER_SEL4: (see FILTER_SEL0)
15	R/W	0	Unused
14-12	R/W	0	FILTER_SEL3: (see FILTER_SEL0)
11	R/W	0	Unused
10-8	R/W	0	FILTER_SEL2: (see FILTER_SEL0)
7	R/W	0	Unused
6-4	R/W	0	FILTER_SEL1: (see FILTER_SEL0)
3	R/W	0	unused
2-0	R/W	0	FILTER_SEL0: This value sets the filter selection for GPIO interrupt 0. A value of 0 = no filtering. A value of 7 corresponds to 7 x 3 x (111nS) of filtering.

8.10 TIMER

8.10.1 Overview

The SOC contains 15 general purpose timers and 2 watchdog timers.

8.10.2 General-Purpose Timer

The SOC contains a number of general-purpose timers that can be used as general counters or interrupt generators. Each counter (except TIMER E) can be configured as a periodic counter (for generating periodic interrupts) or a simple count-down and stop counter. Additionally, the timers have a programmable count rate ranging from 1uS to 1mS. The table below outlines the general-purpose timers available in the chip.

Table 8-33 General-Purpose Timer

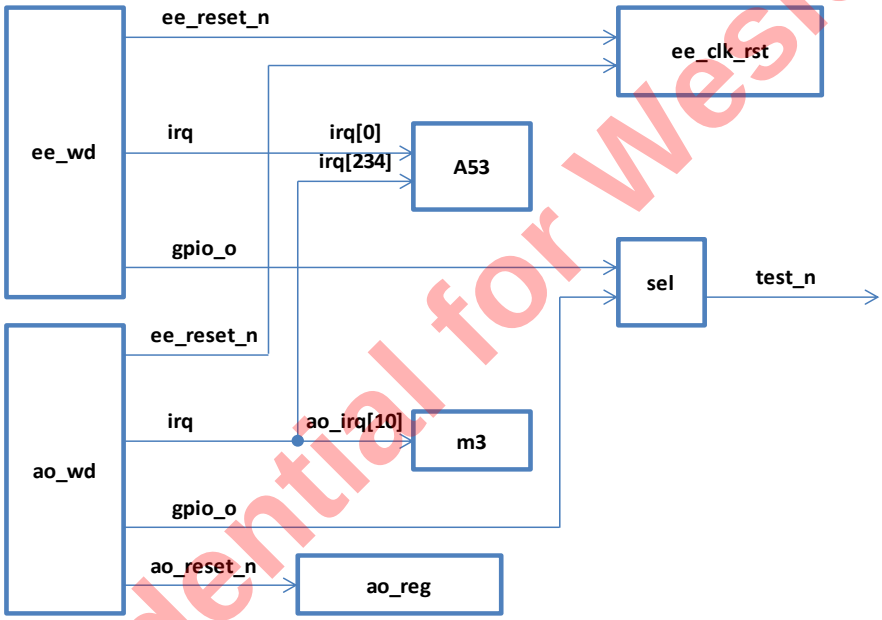
Timer	Timebase Options	Counter size	Comment
Timer A	1uS, 10uS, 100uS, 1mS	16-bits	The 16-bit counter allows the timer to generate interrupts as infrequent as every 65.535 Seconds
Timer B	1uS, 10uS, 100uS, 1mS	16-bits	
Timer C	1uS, 10uS, 100uS, 1mS	16-bits	
Timer D	1uS, 10uS, 100uS, 1mS	16-bits	
Timer E	System clock, 1uS, 10uS, 100uS, 1mS	64-bits	Doesn't generate an interrupt. This is a count up counter that counts from 0 to 0xFFFFFFFF. The counter can be written at any time to reset the value to 0.
Timer F	1uS, 10uS, 100uS, 1mS	16-bits	
Timer G	1uS, 10uS, 100uS, 1mS	16-bits	
Timer H	1uS, 10uS, 100uS, 1mS	16-bits	
Timer I	1uS, 10uS, 100uS, 1mS	16-bits	
Timer A-AO	System clock, 1uS, 10uS, 100uS	32-bits	Used in the Always On domain to generate interrupts for the AO-CPU
Timer B-AO	System clock, 1uS, 10uS, 100uS	32-bits	Used in the Always On domain to generate interrupts for the AO-CPU
Timer C-AO	System clock, 1uS, 10uS, 100uS	32-bits	Used in the Always On domain to generate interrupts for the AO-CPU
Timer E-AO	System clock	64-bits	This Always On counter doesn't generate an interrupt. Instead it simply counts up from 0 to 0xFFFFFFFF. The counter can be written at any time to reset the value to 0.

Timer	Timebase Options	Counter size	Comment
Timer F-AO	System clock	64-bits	This Always On counter doesn't generate an interrupt. Instead it simply counts up from 0 to 0xFFFFFFFF. The counter can be written at any time to reset the value to 0.
Timer G-AO	System clock	64-bits	This Always On counter doesn't generate an interrupt. Instead it simply counts up from 0 to 0xFFFFFFFF. The counter can be written at any time to reset the value to 0.

8.10.3 Watchdog Timer

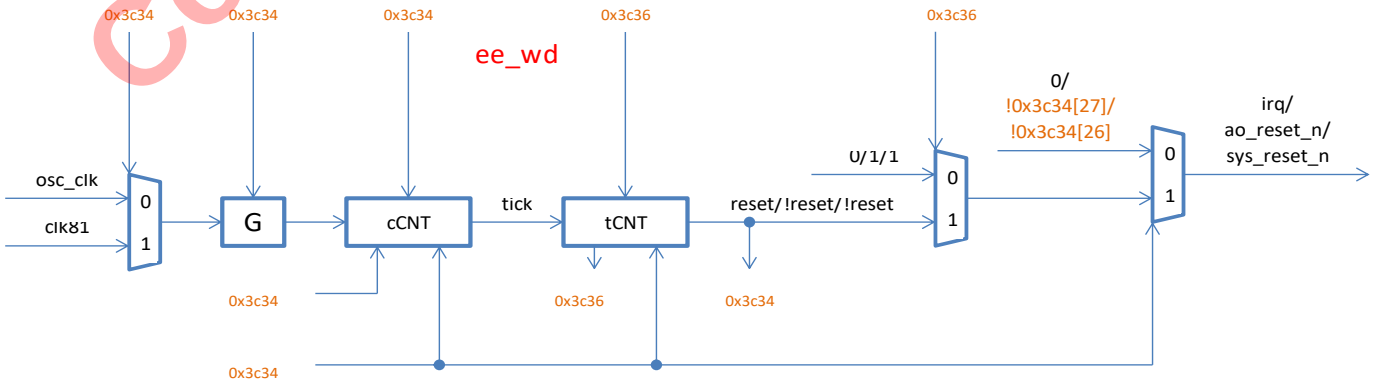
There are also two watchdog timers, one in AO and the other in EE domain, illustrated as following:

Figure 8-27 Watchdog Timer



AO domain watchdog timer and EE domain watchdog timer have the same design, as illustrated in the following figure:

Figure 8-28 EE domain Watchdog Timer Degisn



The AO Domain watchdog timer is driven from the system clock (typically 157Mhz). It is a 16-bit counter that is periodically reset by either the AO CPU or the System CPU (A53). This AO-watchdog timer can be used to generate an interrupt of the AO domain. Additionally, the AO-watchdog timer can be used to "enable" a delay generator that can toggle a GPIO pin (currently the TEST_n I/O pad). The "delay generator" allows an interrupt to first be acknowledged by the AO-CPU before the TEST_N pad is toggled. The "delay generator" is programmable from 1 to 65535 system clocks (typically 417uS).

It should be noted that the AO watchdog timer can also be used to reset the AO domain but this feature is only used when operating in a suspend mode (only the AO-domain is powered). As long as the system periodically resets the AO-watchdog timer the WD_GPIO_CNT (delay generator) will not be enabled and the I/O pad will not toggle.

NOTE:
The maximum delay between two AO-watchdog periodic resets is about 100mS (assuming a 157Mhz system clock).

The EE Domain watchdog timer is driven by the 24Mhz crystal clock and can be used to generate an interrupt to the system CPU (the A53) or optionally, the watchdog timer can completely reset the chip (causing a cold boot). There are a few registers that are not affected by watchdog timer. These registers are only reset by the external RESET_n I/O pad and can be used to store information related to a possible watchdog event. As long as the system CPU periodically resets the EE-watchdog timer, it will never timeout and cause an interrupt or system reset.

NOTE:
The maximum delay between two EE-watchdog periodic resets is about 8.3 Seconds. This time is independent of the system clocks and is driven by the external 24Mhz crystal.

8.10.4 Register Description

Each register final address = 0xffd00000 + address * 4

ISA_TIMER_MUX 0x3c50

Bit(s)	R/W	Default	Description
31-20	R	0	unused
19	R/W	1	TIMERD_ EN: Set to 1 to enable Timer D
18	R/W	1	TIMERC_ EN: Set to 1 to enable Timer C
17	R/W	1	TIMERB_ EN: Set to 1 to enable Timer B
16	R/W	1	TIMERA_EN: Set to 1 to enable Timer A
15	R/W	0	TIMERD_MODE: If This bit is set to 1, then timerD is a periodic . 0 = one-shot timer
14	R/W	0	TIMERC_MODE: If This bit is set to 1, then timerC is a periodic . 0 = one-shot timer
13	R/W	1	TIMERB_MODE: If This bit is set to 1, then timerB is a periodic . 0 = one-shot timer
12	R/W	1	TIMERA_MODE: If This bit is set to 1, then timerA is a periodic . 0 = one-shot timer
11	R	0	unused

Bit(s)	R/W	Default	Description
10-8	R/W	0x1	TIMER E input clock selection: 000: System clock 001: 1uS Timebase resolution 010: 10uS Timebase resolution 011: 100uS Timebase resolution 100: 1mS timebase NOTE: The mux selection for Timer E is different from timer A, B, C and D
7-6	R/W	0x0	TIMER D input clock selection: See TIMER A below
5-4	R/W	0x0	TIMER C input clock selection: See TIMER A below
2-3	R/W	0x0	TIMER B input clock selection: See TIMER A below
1-0	R/W	0x0	TIMER A Input clock selection: These Bits select the input timebase for the counters for TimerA 00: 1uS Timebase resolution 01: 10uS Timebase resolution 10: 100uS Timebase resolution 11: 1mS Timebase resolution

ISA_TIMERA 0x3c51

Timer A is a 16 bit count DOWN counter driven by the clock selected in register 0x01000530. TIMER A will count down from some value to zero, generate an interrupt and then re-load the original start count value. This timer can be used to generate a periodic interrupt (e.g. interrupt every 22 uS).

Bit(s)	R/W	Default	Description
31-16	R	-	Current Count value
15-0	R/W	0x0	Starting count value. Write this value to start TIMER A.

ISA_TIMERB 0x3c52

Timer B is just like Timer A.

Bit(s)	R/W	Default	Description
31-16	R	-	Current Count value
15-0	R/W	0x0	Starting count value. Write this value to start TIMER B

ISA_TIMERC 0x3c53

Timer C is just like Timer A.

Bit(s)	R/W	Default	Description
31-16	R	0	unused

15-0	R/W	0x0	Starting count value. Write this value to start TIMER C

ISA_TIMERD 0x3c54

Timer D is identical to Timer A.

Bit(s)	R/W	Default	Description
31-16	R	0	unused
15-0	R/W	0x0	Starting count value. Write this value to start TIMER D

ISA_TIMERE 0x3c62

Timer E is simply a32-bit counter that increments at a rate set by register 0x2650. To reset the counter to zero, simply write this register with any value. The value below is a read-only value that reflects the current count of the internal counter. This register can be used by software to simply provide a polling delay loop based on a programmable timebase.

Bit(s)	R/W	Default	Description
31-0	R	-	Current value of Timer E. Write this register with any value to clear the counter.

ISA_TIMERE _HI 0x3c63

Bit(s)	R/W	Default	Description
31-0	R	-	Current value of Timer E[63:32]. Need read ISA_TIMERE first.

ISA_TIMER_MUX1 0x3c64

Bit(s)	R/W	Default	Description
31-20	R	0	unused
19	R/W	1	TIMERI_ EN: Set to 1 to enable Timer I
18	R/W	1	TIMERH_ EN: Set to 1 to enable Timer H
17	R/W	1	TIMERG_ EN: Set to 1 to enable Timer G
16	R/W	1	TIMERF_EN: Set to 1 to enable Timer F
15	R/W	0	TIMERI_MODE: If This bit is set to 1, then timerI is a periodic . 0 = one-shot timer
14	R/W	0	TIMERH_MODE: If This bit is set to 1, then timerH is a periodic . 0 = one-shot timer
13	R/W	1	TIMERG_MODE: If This bit is set to 1, then timerG is a periodic. 0 = one-shot timer
12	R/W	1	TIMERF_MODE: If This bit is set to 1, then timerF is a periodic. 0 = one-shot timer
11	R	0	unused

Bit(s)	R/W	Default	Description
10-8	R/W	0x1	TIMER J input clock selection: 000: System clock 001: 1uS Timebase resolution 010: 10uS Timebase resolution 011: 100uS Timebase resolution 100: 1mS timebase NOTE: The mux selection for Timer J is different from timer F, G, H and I
7-6	R/W	0x0	TIMER I input clock selection: See TIMER F below
5-4	R/W	0x0	TIMER H input clock selection: See TIMER F below
2-3	R/W	0x0	TIMER G input clock selection: See TIMER F below
1-0	R/W	0x0	TIMER F Input clock selection: These Bits select the input timebase for the counters for TimerF 00: 1uS Timebase resolution 01: 10uS Timebase resolution 10: 100uS Timebase resolution 11: 1mS Timebase resolution

ISA_TIMERF 0x3c65

Timer F is a 16 bit count DOWN counter driven by the clock selected in register 0x01000530. TIMER A will count down from some value to zero, generate an interrupt and then re-load the original start count value. This timer can be used to generate a periodic interrupt (e.g. interrupt every 22 uS).

Bit(s)	R/W	Default	Description
31-16	R	-	Current Count value
15-0	R/W	0x0	Starting count value. Write this value to start TIMER A.

ISA_TIMERG 0x3c66

Timer G is just like Timer F.

Bit(s)	R/W	Default	Description
31-16	R	-	Current Count value
15-0	R/W	0x0	Starting count value. Write this value to start TIMER B

ISA_TIMER_H 0x3c67

Timer H is just like Timer F.

Bit(s)	R/W	Default	Description
31-16	R	0	unused
15-0	R/W	0x0	Starting count value. Write this value to start TIMER C

ISA_TIMER_I 0x3c68

Timer I is just like Timer F.

Bit(s)	R/W	Default	Description
31-16	R	0	unused
15-0	R/W	0x0	Starting count value. Write this value to start TIMER I

WATCHDOG_CNTL 0x3c34

Bit(s)	R/W	Default	Description
31	R	0	Watchdog_reset
30-28	R/W	0	Reserved
27	R/W	0	Ao_reset_n_now, if watchdog_en =0, output ao_reset_n = ! ao_reset_n_now
26	R/W	0	Sys_reset_n_now, if watch_dog_en = 0, output sys_reset_n = !sys_reset_n_now.
25	R/W	0	Clk_div_en: 0: no tick; 1: generate tick;
24	R/W	0	Clk_en: 0: no clk; 1: clk work;
23	R/W	0	Interrupt_en: 0: no irq out; 1: irq = watchdog_reset;
22	R/W	0	Ao_reset_n_en: 0: output ao_reset_n = 1; 1: output ao_reset_n = ! watchdog_reset;
21	R/W	0	Sys_reset_n_en 0: output sys_reset_n = 1; 1: output sys_reset_n = ! watchdog_reset;
20	R/W	0	Reserved
19	R/W	0	Clk_sel: 0:osc_clk; 1:clk_81
18	R/W	0	Watchdog_en 0: no watchdog reset 1: gen watchdog reset
17-0	R/W	0	Clk_div_tcnt, when clk_div_en is 1, generate a tick each tcnt clock

WATCHDOG_CNTL1 0x3c35

Bit(s)	R/W	Default	Description
31-18	R	0	Reserved
17	R/W	0	Gpio_pulse 0:level reset 1:pulse reset
16	R/W	0	Gpio_polarity 0: 1 is reset; 1: 0 is reset.
15-0	R/W	0	Gpio_pulse_tcnt If gpio_pulse is 1, level reset will hold tcnt clock.

WATCHDOG_TCNT 0x3c36

Bit(s)	R/W	Default	Description
31-16	R	0	The cnt of tick.
15-0	R/W	5000	If watchdog_en is 1, when tick cnt reached “5000”, generate watchdog_reset.

WATCHDOG_RESET0x3c37

Bit(s)	R/W	Default	Description
31-0	W	0	When write any value(include 0), watchdog module will be reset.

Below registers have the same base address: 0xFF800000, register address: 0xFF800000 + offset * 4, same for secure and non-secure access

AO_TIMER_CTRL 0xf0

Timer controls.

Bit(s)	R/W	Default	Description
31-7	R/W	0	Unused
6-5	R/W	0	TIMERE_CLK_MUX: 00 = Timer E clock = Media CPU clock 01 = Timer E clock = 1uS ticks 10 = Timer E clock = 10uS ticks 11 = Timer E clock = 100uS ticks
4	R/W	0	TIMER_E_EN
3	R/W	0	TIMER_A_EN
2	R/W	0	TIMER_A_MODE: 1 = periodic, 0 = one-shot

1-0	R/W	0	TIMERA_CLK_MUX: 00 = TimerA clock = Media CPU clock 01 = Timer A clock = 1uS ticks 10 = Timer A clock = 10uS ticks 11 = Timer A clock = 100uS ticks

AO_TIMER_SEC_SCP_CTRL 0xf1

Timer controls.

Bit(s)	R/W	Default	Description
31-7	R/W	0	Unused
6-5	R/W	0	TIMER F_CLK_MUX: 00 = Timer F clock = Media CPU clock 01 = Timer F clock = 1uS ticks 10 = Timer F clock = 10uS ticks 11 = Timer F clock = 100uS ticks
4	R/W	0	TIMER_F_EN
3	R/W	0	TIMER_B_EN
2	R/W	0	TIMER_B_MODE: 1 = periodic, 0 = one-shot
1-0	R/W	0	TIMERB_CLK_MUX: 00 = TimerB clock = Media CPU clock 01 = Timer B clock = 1uS ticks 10 = Timer B clock = 10uS ticks 11 = Timer B clock = 100uS ticks

AO_TIMER_SEC_SP_CTRL 0xf2

Timer controls.

Bit(s)	R/W	Default	Description
31-7	R/W	0	Unused
6-5	R/W	0	TIMER G_CLK_MUX: 00 = Timer G clock = Media CPU clock 01 = Timer G clock = 1uS ticks 10 = Timer G clock = 10uS ticks 11 = Timer G clock = 100uS ticks
4	R/W	0	TIMER_G_EN
3	R/W	0	TIMER_C_EN
2	R/W	0	TIMER_C_MODE: 1 = periodic, 0 = one-shot

1-0	R/W	0	TIMERC_CLK_MUX: 00 = Timer C clock = Media CPU clock 01 = Timer C clock = 1uS ticks 10 = Timer C clock = 10uS ticks 11 = Timer C clock = 100uS ticks

AO_TIMERA_REG 0xf3

Timer A starts at a non-zero value and decrements to 0. When timer A reaches a count of 0 it will re-load with the TIMER_A_TCNT value.

Bit(s)	R/W	Default	Description
31-0	R/W	0	timer A set value; if mode = 0, will clr timer to 0;

AO_TIMERA_CUR_REG 0xf4

Bit(s)	R/W	Default	Description
31-0	R	0	TIMER A current count.

AO_TIMERB_REG 0xf5

Bit(s)	R/W	Default	Description
31-0	R/W	0	timer B set value; if mode = 0, will clr timer to 0;

AO_TIMERB_CUR_REG 0xf6

Bit(s)	R/W	Default	Description
31-0	R	0	TIMER B current count.

AO_TIMERC_REG 0xf7

Bit(s)	R/W	Default	Description
31-0	R/W	0	timer C set value; if mode = 0, will clr timer to 0;

AO_TIMERC_CUR_REG 0xf8

Bit(s)	R/W	Default	Description
31-0	R	0	TIMER C current count.

AO_TIMERE_REG 0xf9

If this register is written (with any value), then Timer E is reset to 0. Immediately after being cleared, timer E will start incrementing at a clock rate equal to the clock used for the Media CPU..

Bit(s)	R/W	Default	Description
31-0	R/W	0	TIMER E current Count[31:0]

AO_TIMERE_HI_REG 0xfa

Bit(s)	R/W	Default	Description
31-0	R/W	0	TIMER E current Count[63:32], latched by read timerE[31:0];

AO_TIMERF_REG 0xfb

If this register is written (with any value), then Timer F is reset to 0. Immediately after being cleared, timer F will start incrementing at a clock rate equal to the clock used for the Media CPU..

Bit(s)	R/W	Default	Description
31-0	R/W	0	TIMER F current Count[31:0]

AO_TIMERF_HI_REG 0xfc

Bit(s)	R/W	Default	Description
31-0	R/W	0	TIMER F current Count[63:32], latched by read timerF[31:0];

AO_TIMERG_REG 0xfd

If this register is written (with any value), then Timer G is reset to 0. Immediately after being cleared, timer G will start incrementing at a clock rate equal to the clock used for the Media CPU..

Bit(s)	R/W	Default	Description
31-0	R/W	0	TIMER G current Count[31:0]

AO_TIMERG_HI_REG 0xfe

Bit(s)	R/W	Default	Description
31-0	R/W	0	TIMER G current Count[63:32], latched by read timerG[31:0];

AO_WATCHDOG_CNTL 0x48

A53/Secure	A53/Non Secure	M3/Secure	M3/Non Secure	Reset by watchdog
R/W	R/W	R/W	R/W	Yes

Bit(s)	R/W	Default	Description
31-28	R	0	unused
27	R/W	0	Ao_reset_n_now
26	R/W	0	Sys_reset_n_now
25	R/W	0	Clk_div_en
24	R/W	1	Clk_en
23	R/W	0	Interrupt_en

22	R/W	0	Ao_reset_n_en
21	R/W	0	Sys_reset_n_en
20	R/W	0	Noused
19	R/W	0	Clk_sel
18	R/W	0	Watchdog_en
17-0	R/W	23999	Clk_div_tcnt

AO_WATCHDOG_CNTL1 0x49

A53/Secure	A53/Non Secure	M3/Secure	M3/Non Secure	Reset by watchdog
R/W	R/W	R/W	R/W	Yes

Bit(s)	R/W	Default	Description
31-17	R	0	unused
17	R/W	0	Gpio_pulse
16	R/W	0	Gpio_polarity
15-0	R/W	0	Gpio_pulse_tcnt

AO_WATCHDOG_TCNT 0x4a

A53/Secure	A53/Non Secure	M3/Secure	M3/Non Secure	Reset by watchdog
R/W	R/W	R/W	R/W	Yes

Bit(s)	R/W	Default	Description
31-16	R	0	Watchdog count read
15-0	R/W	0	Watchdog_count set

AO_WATCHDOG_RESET 0x4b

A53/Secure	A53/Non Secure	M3/Secure	M3/Non Secure	Reset by watchdog
R/W	R/W	R/W	R/W	Yes

Write any value can reset watchdog.

Bit(s)	R/W	Default	Description
31-0	W	0	Watchdog soft reset

8.11 Crypto

8.11.1 Overview

The crypto engine is one encrypt/decrypt function accelerator. It supports both encryption/decryption and signature/verification. Crypto engine supports 4 different modes, i.e. A53 secure, A53 non secure, M3 secure and M3 non secure. The crypto engine has special internal DMA controller to transfer data.

It has the following features:

- AES block cipher with 128/192/256 Bits keys, standard 16 bytes block size and streaming ECB, CBC and CTR modes
- DES/TDES block cipher with ECB and CBC modes supporting 64 Bits key for DES and 192 Bits key for 3DES
- Hardware key-ladder operation and DVB-CSA for transport stream encryption
- Built-in hardware True Random Number Generator (TRNG), CRC and SHA-1/SHA-2 engine

8.11.2 Key Ladder

The Key Ladder is a series of TDES / AES crypto processes that iterates on different user supplied and OTP keys. The key ladder module uses a single AES / TDES crypto module and iterates using internal storage to hold temporary states.

8.11.3 RNG

Functionality the Random Number Generator (RNG) contains two main modules: True Random Number Generator (TRNG) and Deterministic Random Number Generator (DRNG).

True Random Number Generator (TRNG): this TRNG is realized by using metastability and jitter for random bit generation based on four free running ring oscillator

Deterministic Random Number Generator (DRNG): this DRNG, which has32-bit random number generator, is mainly designed to increase the throughput and do post-processing of the Digital TRNG, which will need hundreds cycles to collect entropy.

8.11.4 EFUSE

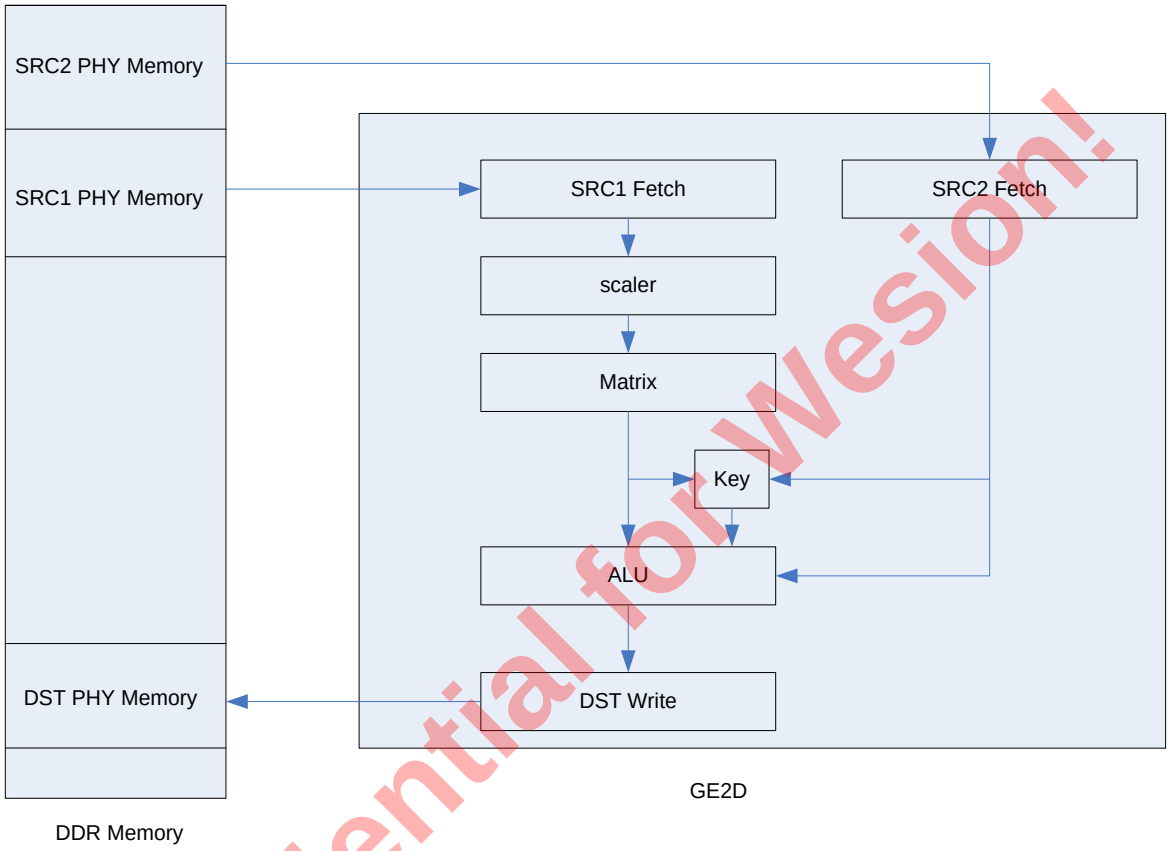
The EFUSE consists of a 4kbit One Time Programmable (OTP) memory that is broken up into 32, 128-bit blocks. Data is always read/written in 128-bit blocks using the APB bus (software) or by the Key-ladder which is integrated with EFUSE block.

9. GE2D

9.1 Overview

The basic structure of GE2D is shown in Fig below.

Figure 9-1 GE2D Structure



9.2 Register Description

GE2D_GEN_CTRL0 0x8A0

Bit(s)	R/W	Default	Description
31	R/W	0	DST_BYTEMASK_ONLY: Applicable only if DST_BITMASK_EN=1.
30	R/W	0	DST_BITMASK_EN: destination bitmask enable. 0: disable; 1: enable.
29	R/W	0	SRC2_KEY_EN: source2 key enable. 0: disable; 1: enable.
28	R/W	0	SRC2_KEY_MODE: source2 key mode.

Bit(s)	R/W	Default	Description
			0: mask data when match; 1: mask data when unmatch.
27	R/W	0	SRC1_KEY_EN: source1 key enable. 0: disable; 1: enable.
26	R/W	0	SRC1_KEY_MODE: source1 key mode. 0: mask data when match; 1: mask data when unmatch.
25-24	R/W	0	DST1_8B_MODE_SEL: Destination 1's 8-bit mode component selection. 0: Select Y or R; 1: Select Cb or G; 2: Select Cr or B; 3: Select Alpha.
23	R/W	0	DST_CLIP_MODE: 0: Write inside clip window; 1: Write outside clip window.
22-17	R	0	Unused
16-15	R/W	0	SRC2_8B_MODE_SEL: Applicable only when SRC2_FORMAT=0, define the property of the 8-bit output data. 0: The 8-bit output data is for Y or R; 1: The 8-bit output data is for Cb or G; 2: The 8-bit output data is for Cr or B; 3: The 8-bit output data is for Alpha.
14	R/W	0	SRC2_FILL_MODE: When the display window is outside the boundary of the clipping window, this field defines how to fill the area outside the clipping window. 0: Fill with the pixels at the boundary of the clipping window; 1: Fill with the default color defined by register GE2D_SRC2_DEF_COLOR.
13-12	R/W	0	SRC2_PIC_STRUCT: Define how source2 reads the picture stored in DDR memory. 0: Read all lines; 1: Reserved; 2: Read even lines only; 3: Read odd lines only.
11	R/W	0	SRC2_X_YC_RATIO: Source2 x direction yc ratio. 0: 1:1; 1: 2:1.

Bit(s)	R/W	Default	Description
9-7	R	0	Unused
6-5	R/W	0	SRC1_8B_MODE_SEL: Applicable only when SRC1_SEP_EN =0, SRC1_FORMAT=0 and SRC1_LUT_EN=0, define the property of the 8-bit output data. 0: the 8-bit output data is for Y or R; 1: the 8-bit output data is for Cb or G; 2: the 8-bit output data is for Cr or B; 3: the 8-bit output data is for Alpha.
4	R/W	0	SRC1_FILL_MODE: When the display window is outside the boundary of the clipping window, this field defines how to fill the area outside the clipping window. 0: Fill with the pixels at the boundary of the clipping window; 1: Fill with the default color defined by register GE2D_SRC1_DEF_COLOR.
3	R/W	0	SRC1_LUT_EN: Applicable only when SRC1_SEP_EN =0 and SRC1_FORMAT= 0, define whether to enable 8-bit input data to look up a 32-bit pixel for output. 0: Disable; 1: Enable.
2-1	R/W	0	SRC1_PIC_STRUCT: Define how source1 reads the picture stored in DDR memory. 0: Read all lines; 1: Reserved; 2: Read even lines only; 3: Read odd lines only.

GE2D_GEN_CTRL1 0x8A1

Bit(s)	R/W	Default	Description
31	R/W	0	SOFT_RST: If true, reset GE2D.
30	R/W	0	dst write response counter reset
29	R/W	0	disable adding dst write response count to busy bit
28-27	R	0	Unused
26	R/W	0	COLOR_CONVERSION_MODE[1] in alu. Mode[1:0] 3 : color_out = color; 2 : color_out = (color != 255) ? color : color + 1; 1 : color_out = (color < 128) ? color : color + 1; 0 : color_out = (color == 0) ? color : color + 1.
25-24	R/W	0	INTERRUPT_CTRL: If bit[0] true, generate interrupt when one command done; if bit[1] true, generate interrupt when ge2d change from busy to not busy.

Bit(s)	R/W	Default	Description
23-22	R/W	0x3	SRC2_BURST_SIZE_CTRL: Source2 DDR request burst size control. (Note: data to source2 are stored together in one DDR memory block.) 0: Burst size = 24 x 64-bit; 1: Burst size = 32 x 64-bit; 2: Burst size = 48 x 64-bit; 3: Burst size = 64 x 64-bit.
21-16	R/W	0x3F	SRC1_BURST_SIZE_CTRL: Source1 DDR request burst size control. Bit[21:20] control Y burst size, bit[19:18] control Cb burst size, bit[17:16] control Cr burst size. Each 2-bit is decoded as below: 0: Burst size = 24 x 64-bit; 1: Burst size = 32 x 64-bit; 2: Burst size = 48 x 64-bit; 3: Burst size = 64 x 64-bit.
15-14	R/W	0	DST1_PIC_STRUCT: Define how destination 1 write the picture to DDR memory. 0: Write all lines (whole frame); 1: Reserved; 2: Write even lines only (top); 3: Write odd lines only (bottom).
13-12	R/W	0	SRC_RD_CTRL: Bit[13] if true, force read src1, bit[12] if true, force read src2.
11	R/W	0	DST2_URGENT_EN: Destination 2 DDR request urgent enable.
10	R/W	0	SRC1_URGENT_EN: Source1 DDR request urgent enable.
9	R/W	0	SRC2_URGENT_EN: Source2 DDR request urgent enable.
8	R/W	0	DST1_URGENT_EN: Destination 1 DDR request urgent enable.
7-0	R/W	0	SRC1_GB_ALPHA: Source1 global alpha.

GE2D_GEN_CTRL2 0x8A2

Bit(s)	R/W	Default	Description
31	R/W	0	ALPHA_CONVERSION_MODE[0] in alu. Mode[1:0]: 2,3 : alpha_out = (alpha!=255) ? alpha : alpha + 1; 1 : alpha_out = (alpha < 128) ? alpha : alpha + 1; 0 : alpha_out = (alpha == 0) ? alpha : alpha + 1.
30	R/W	0	COLOR_CONVERSION_MODE[0] in alu. Mode[1:0] 3 : color_out = color;

Bit(s)	R/W	Default	Description
			2 : color_out = (color != 255) ? color : color + 1; 1 : color_out = (color < 128) ? color : color + 1; 0 : color_out = (color == 0) ? color : color + 1.
29	R/W	0	SRC1_GB_ALPHA_EN in alu. As = src1_gb_alpha_en ? Asr * Ag: Asr
28	R/W	0	DST1_COLOR_ROUND_MODE. 1 = Truncate the full bit color components to required output bit width; 0 = Round (+ 0.5) the full bit color components to required output bit width.
27	R/W	0	SRC2_COLOR_EXPAND_MODE. 1 = Expand the color components to 8-bit by padding LSBs with MSBs. E.g. If the input is 5'b11000, the output is expanded to 8'b11000110; 0 = Expand the color components to 8-bit by padding LSBs with 0.
26	R/W	0	SRC2_ALPHA_EXPAND_MODE. 1 = Expand alpha value to 8-bit by padding LSBs with MSBs. E.g. If the input is 5'b11000, the output is expanded to 8'b11000110; 0 = If input alpha value is all 1, then expand the value to 8-bit by padding LSBs with 1; otherwise, pad LSBs with 0.
25	R/W	0	SRC1_COLOR_EXPAND_MODE. 1 = Expand the color components to 8-bit by padding LSBs with MSBs. E.g. If the input is 5'b11000, the output is expanded to 8'b11000110; 0 = Expand the color components to 8-bit by padding LSBs with 0.
24	R/W	0	SRC1_ALPHA_EXPAND_MODE. 1 = Expand alpha value to 8-bit by padding LSBs with MSBs. E.g. If the input is 5'b11000, the output is expanded to 8'b11000110; 0 = If input alpha value is all 1, then expand the value to 8-bit by padding LSBs with 1; otherwise, pad LSBs with 0.
23	R/W	0	DST_LITTLE_ENDIAN: define the endianess of SRC2 input data. 1 = Little endian; 0 = Big endian.
22-19	R/W	0	DST1_COLOR_MAP: Applicable to 16-bit, 24-bit and 32-bit pixel, defines the bit-field allocation of the pixel data. For whether to truncate or round full 8-bit to output, refer to GE2D_GEN_CTRL2.DST1_COLOR_ROUND_MODE.

Bit(s)	R/W	Default	Description
			For 16-bit mode (DST1_FORMAT=1): 0 = Unused; 1 = 6:5:5 format. Bit[15:10] is Y[7:2] or R[7:2], bit[9:5] is Cb[7:3] or G[7:3], bit[4:0] is Cr[7:3] or B[7:3]; 2 = 8:4:4 format. Bit[15:8] is Y or R, bit[7:4] is Cb[7:4] or G[7:4], bit[3:0] is Cr[7:4] or B[7:4]]; 3 = 6:4:4:2 format. Bit[15:10] is Y[7:2] or R[7:2], bit[9:6] is Cb[7:4] or G[7:4], bit[5:2] is Cr[7:4] or B[7:4], bit[1:0] is Alpha[7:6]; 4 = 4:4:4:4 format. Bit[15:12] is Y[7:4] or R[7:4], bit[11:8] is Cb[7:4] or G[7:4], bit[7:4] is Cr[7:4] or B[7:4], bit[3:0] is Alpha[7:4]; 5 = 5:6:5 format. Bit[15:11] is Y[7:3] or R[7:3], bit[10:5] is Cb[7:2] or G[7:2], bit[4:0] is Cr[7:3] or B[7:3]; 6 = 4:4:4:4 format. Bit[15:12] is Alpha[7:4], bit[11:8] is Y[7:4] or R[7:4], bit[7:4] is Cb[7:4] or G[7:4], bit[3:0] is Cr[7:4] or B[7:4]; 7 = 1:5:5:5 format. Bit[15] is Alpha[7], bit[14:10] is Y[7:3] or R[7:3], bit[9:5] is Cb[7:3] or G[7:3], bit[4:0] is Cr[7:3] or B[7:3]; 8 = 4:6:4:2 format. Bit[15:12] is Y[7:4] or R[7:4], bit[11:6] is Cb[7:2] or G[7:2], bit[5:2] is Cr[7:4] or B[7:4], bit[1:0] is Alpha[1:0]. 9 = CbCr format. Bit[15:8] is Cb, bit[7:0] is Cr; 10 = CrCb format. Bit[15:8] is Cr, bit[7:0] is Cb. For 24-bit mode (DST1_FORMAT=2): 0 = RGB 8:8:8 mode. Bit[23:16] is Y or R, bit[15:8] is Cb or G, bit[7:0] is Cr or B; 1 = RGBA 5:6:5:8 mode. Bit[23:19] is Y[7:3] or R[7:3], bit[18:13] is Cb[7:2] or G[7:2], bit[12:8] is Cr[7:3] or B[7:3], bit[7:0] is Alpha; 2 = ARGB 8:5:6:5 mode. Bit[23:16] is Alpha, bit[15:11] is Y[7:3] or R[7:3], bit[10:5] is Cb[7:2] or G[7:2], bit[4:0] is Cr[7:3] or B[7:3]; 3 = RGBA 6:6:6:6 mode. Bit[23:18] is Y[7:2] or R[7:2], bit[17:12] is Cb[7:2] or G[7:2], bit[11:6] is Cr[7:2] or B[7:2], bit[5:0] is Alpha[7:2]; 4 = ARGB 6:6:6:6 mode. Bit[23:18] is Alpha[7:2];, bit[17:12] is Y[7:2] or R[7:2], bit[11:6] is Cb[7:2] or G[7:2], bit[5:0] is Cr[7:2] or B[7:2]; 5 = BGR 8:8:8 mode. Bit[23:16] is Cr or B, bit[15:8] is Cb or G, bit[7:0] is Y or R. For 32-bit mode (DST1_FORMAT=3): 0 = RGBA 8:8:8:8 format. Bit[31:24] is Y or R, bit[23:16] is Cb or G; bit[15:8] is Cr or B; bit[7:0] is Alpha;

Bit(s)	R/W	Default	Description
			1 = ARGB 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Y or R; bit[15:8] is Cb or G; bit[7:0] is Cr or B; 2 = ABGR 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Cr or B; bit[15:8] is Cb or G; bit[7:0] is Y or R; 3 = BGRA 8:8:8:8 format. Bit[31:24] is Cr or B, bit[23:16] is Cb or G; bit[15:8] is Y or R; bit[7:0] is Alpha.
18	R/W	0	ALU_MULT_MODE: 1: mult result rounding else truncation
17-16	R/W	0	DST1_FORMAT: define output pixel byte-width. 0: Output pixel is 1-byte (8-bit) color component; 1: Output pixel is 2-byte (16-bit), refer to GE2D_GEN_CTRL2.DST1_COLOR_MAP for further pixel color mapping; 2: Output pixel is 3-byte (24-bit), refer to GE2D_GEN_CTRL2.DST1_COLOR_MAP for further pixel color mapping; 3: Output pixel is 4-byte (32-bit), refer to GE2D_GEN_CTRL2.DST1_COLOR_MAP for further pixel color mapping.
15	R/W	0	SRC2_LITTLE_ENDIAN: define the endianess of SRC2 input data. 1 = Little endian; 0 = Big endian.
14-11	R/W	0	SRC2_COLOR_MAP: Applicable to 16-bit, 24-bit and 32-bit pixel, defines the bit-field allocation of the pixel data. For expanding the bit-fields to full 8-bit, refer to GE2D_GEN_CTRL2.SRC2_COLOR_EXPAND_MODE and GE2D_GEN_CTRL2.SRC2_ALPHA_EXPAND_MODE. For 16-bit mode (SRC2_FORMAT=1): 0 = Unused; 1 = 6:5:5 format. Bit[15:10] is Y[7:2] or R[7:2], bit[9:5] is Cb[7:3] or G[7:3], bit[4:0] is Cr[7:3] or B[7:3], for Alpha value refer to ???; 2 = 8:4:4 format. Bit[15:8] is Y or R, bit[7:4] is Cb[7:4] or G[7:4], bit[3:0] is Cr[7:4] or B[7:4]], for Alpha value refer to ???; 3 = 6:4:4:2 format. Bit[15:10] is Y[7:2] or R[7:2], bit[9:6] is Cb[7:4] or G[7:4], bit[5:2] is Cr[7:4] or B[7:4], bit[1:0] is Alpha[7:6]; 4 = 4:4:4:4 format. Bit[15:12] is Y[7:4] or R[7:4], bit[11:8] is Cb[7:4] or G[7:4], bit[7:4] is Cr[7:4] or B[7:4], bit[3:0] is Alpha[7:4]; 5 = 5:6:5 format. Bit[15:11] is Y[7:3] or R[7:3], bit[10:5] is Cb[7:2] or G[7:2], bit[4:0] is Cr[7:3] or B[7:3], for Alpha value refer to ???;

Bit(s)	R/W	Default	Description
			6 = 4:4:4:4 format. Bit[15:12] is Alpha[7:4], bit[11:8] is Y[7:4] or R[7:4], bit[7:4] is Cb[7:4] or G[7:4], bit[3:0] is Cr[7:4] or B[7:4]; 7 = 1:5:5:5 format. Bit[15] is Alpha[7], bit[14:10] is Y[7:3] or R[7:3], bit[9:5] is Cb[7:3] or G[7:3], bit[4:0] is Cr[7:3] or B[7:3]; 8 = 4:6:4:2 format. Bit[15:12] is Y[7:4] or R[7:4], bit[11:6] is Cb[7:2] or G[7:2], bit[5:2] is Cr[7:4] or B[7:4], bit[1:0] is Alpha[1:0]. For 24-bit mode (SRC2_FORMAT=2): 0 = RGB 8:8:8 mode. Bit[23:16] is Y or R, bit[15:8] is Cb or G, bit[7:0] is Cr or B, for Alpha value refer to ???; 1 = RGBA 5:6:5:8 mode. Bit[23:19] is Y[7:3] or R[7:3], bit[18:13] is Cb[7:2] or G[7:2], bit[12:8] is Cr[7:3] or B[7:3], bit[7:0] is Alpha; 2 = ARGB 8:5:6:5 mode. Bit[23:16] is Alpha, bit[15:11] is Y[7:3] or R[7:3], bit[10:5] is Cb[7:2] or G[7:2], bit[4:0] is Cr[7:3] or B[7:3]; 3 = RGBA 6:6:6:6 mode. Bit[23:18] is Y[7:2] or R[7:2], bit[17:12] is Cb[7:2] or G[7:2], bit[11:6] is Cr[7:2] or B[7:2], bit[5:0] is Alpha[7:2]; 4 = ARGB 6:6:6:6 mode. Bit[23:18] is Alpha[7:2];, bit[17:12] is Y[7:2] or R[7:2], bit[11:6] is Cb[7:2] or G[7:2], bit[5:0] is Cr[7:2] or B[7:2]; 5 = BGR 8:8:8 mode. Bit[23:16] is Cr or B, bit[15:8] is Cb or G, bit[7:0] is Y or R, for Alpha value refer to ???. For 32-bit mode (SRC2_FORMAT=3): 0 = RGBA 8:8:8:8 format. Bit[31:24] is Y or R, bit[23:16] is Cb or G; bit[15:8] is Cr or B; bit[7:0] is Alpha; 1 = ARGB 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Y or R; bit[15:8] is Cb or G; bit[7:0] is Cr or B; 2 = ABGR 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Cr or B; bit[15:8] is Cb or G; bit[7:0] is Y or R; 3 = BGRA 8:8:8:8 format. Bit[31:24] is Cr or B, bit[23:16] is Cb or G; bit[15:8] is Y or R; bit[7:0] is Alpha.
10	R/W	0	ALPHA_CONVERSION_MODE[1] in alu. Mode[1:0]: 2,3 : alpha_out = (alpha!=255) ? alpha : alpha + 1; 1 : alpha_out = (alpha < 128) ? alpha : alpha + 1; 0 : alpha_out = (alpha == 0) ? alpha : alpha + 1.
9-8	R/W	0	SRC2_FORMAT: define input pixel byte-width.

Bit(s)	R/W	Default	Description
			0: Input pixel is 1-byte (8-bit) color component; 1: Input pixel is 2-byte (16-bit), refer to GE2D_GEN_CTRL2.SRC2_COLOR_MAP for further pixel color mapping; 2: Input pixel is 3-byte (24-bit), refer to GE2D_GEN_CTRL2.SRC2_COLOR_MAP for further pixel color mapping; 3: Input pixel is 4-byte (32-bit), refer to GE2D_GEN_CTRL2.SRC2_COLOR_MAP for further pixel color mapping.
7	R/W	0	SRC1_LITTLE_ENDIAN: define the endianess of SRC1 input data. 1 = Little endian; 0 = Big endian.
6-3	R/W	0	SRC1_COLOR_MAP: Note: If SRC1_DEEPCOLOR=0, the SRC1_COLOR_MAP's definitions is as below. If SRC1_DEEPCOLOR=1, please refer to SRC1_DEEPCOLOR entry for new meaning. Applicable to 16-bit, 24-bit and 32-bit pixel, defines the bit-field allocation of the pixel data. For expanding the bit-fields to full 8-bit, refer to GE2D_GEN_CTRL2.SRC1_COLOR_EXPAND_MODE and GE2D_GEN_CTRL2.SRC1_ALPHA_EXPAND_MODE. For 16-bit mode (SRC1_FORMAT=1): 0 = 4:2:2 format (Y0Cb0Y1Cr0); 1 = 6:5:5 format. Bit[15:10] is Y[7:2] or R[7:2], bit[9:5] is Cb[7:3] or G[7:3], bit[4:0] is Cr[7:3] or B[7:3], for Alpha value refer to ???; 2 = 8:4:4 format. Bit[15:8] is Y or R, bit[7:4] is Cb[7:4] or G[7:4], bit[3:0] is Cr[7:4] or B[7:4]], for Alpha value refer to ???; 3 = 6:4:4:2 format. Bit[15:10] is Y[7:2] or R[7:2], bit[9:6] is Cb[7:4] or G[7:4], bit[5:2] is Cr[7:4] or B[7:4], bit[1:0] is Alpha[7:6]; 4 = 4:4:4:4 format. Bit[15:12] is Y[7:4] or R[7:4], bit[11:8] is Cb[7:4] or G[7:4], bit[7:4] is Cr[7:4] or B[7:4], bit[3:0] is Alpha[7:4]; 5 = 5:6:5 format. Bit[15:11] is Y[7:3] or R[7:3], bit[10:5] is Cb[7:2] or G[7:2], bit[4:0] is Cr[7:3] or B[7:3], for Alpha value refer to ???; 6 = 4:4:4:4 format. Bit[15:12] is Alpha[7:4], bit[11:8] is Y[7:4] or R[7:4], bit[7:4] is Cb[7:4] or G[7:4], bit[3:0] is Cr[7:4] or B[7:4]; 7 = 1:5:5:5 format. Bit[15] is Alpha[7], bit[14:10] is Y[7:3] or R[7:3], bit[9:5] is Cb[7:3] or G[7:3], bit[4:0] is Cr[7:3] or B[7:3]; 8 = 4:6:4:2 format. Bit[15:12] is Y[7:4] or R[7:4], bit[11:6] is Cb[7:2] or G[7:2], bit[5:2] is Cr[7:4] or B[7:4], bit[1:0] is Alpha[1:0].

Bit(s)	R/W	Default	Description
			For 24-bit mode (SRC1_FORMAT=2): 0 = RGB 8:8:8 mode. Bit[23:16] is Y or R, bit[15:8] is Cb or G, bit[7:0] is Cr or B, for Alpha value refer to ???; 1 = RGBA 5:6:5:8 mode. Bit[23:19] is Y[7:3] or R[7:3], bit[18:13] is Cb[7:2] or G[7:2], bit[12:8] is Cr[7:3] or B[7:3], bit[7:0] is Alpha; 2 = ARGB 8:5:6:5 mode. Bit[23:16] is Alpha, bit[15:11] is Y[7:3] or R[7:3], bit[10:5] is Cb[7:2] or G[7:2], bit[4:0] is Cr[7:3] or B[7:3]; 3 = RGBA 6:6:6:6 mode. Bit[23:18] is Y[7:2] or R[7:2], bit[17:12] is Cb[7:2] or G[7:2], bit[11:6] is Cr[7:2] or B[7:2], bit[5:0] is Alpha[7:2]; 4 = ARGB 6:6:6:6 mode. Bit[23:18] is Alpha[7:2];, bit[17:12] is Y[7:2] or R[7:2], bit[11:6] is Cb[7:2] or G[7:2], bit[5:0] is Cr[7:2] or B[7:2]; 5 = BGR 8:8:8 mode. Bit[23:16] is Cr or B, bit[15:8] is Cb or G, bit[7:0] is Y or R, for Alpha value refer to ???. 14=NV12 format. 8-bit Y and 16-bit CbCr; 15=NV21 format. 8-bit Y and 16-bit CrCb; For 32-bit mode (SRC1_FORMAT=3): 0 = RGBA 8:8:8:8 format. Bit[31:24] is Y or R, bit[23:16] is Cb or G; bit[15:8] is Cr or B; bit[7:0] is Alpha; 1 = ARGB 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Y or R; bit[15:8] is Cb or G; bit[7:0] is Cr or B; 2 = ABGR 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Cr or B; bit[15:8] is Cb or G; bit[7:0] is Y or R; 3 = BGRA 8:8:8:8 format. Bit[31:24] is Cr or B, bit[23:16] is Cb or G; bit[15:8] is Y or R; bit[7:0] is Alpha.
2	R/W	0	SRC1_DEEPCOLOR: 1 = Enable deepcolor formats support, the formats are defined by SRC1_FORMAT and SRC1_COLOR_MAP; 0 = Disable deepcolor. The supported deepcolor formats are as below: src1_deep_color=1, src1_format=2'b01, src1_color_map=4'b0000 or 4'b0001: 10-bit 422 in one canvas -- 10-bit Y + 10-bit C = 20-bit per pixel in canvas. If src1_color_map=4'b0000, the sequence is Y0Cb0, Y1Cr0, ... If src1_color_map=4'b0001, the sequence is Y0Cr0, Y1Cb0, ...

Bit(s)	R/W	Default	Description
			src1_deep_color=1, src1_format=2'b01, src1_color_map=4'b1000 or 4'b1001: 12-bit 422 in one canvas -- 12-bit Y + 12-bit C = 24-bit per pixel in canvas. If src1_color_map=4'b1000, the sequence is Y0Cb0, Y1Cr0, ... If src1_color_map=4'b1001, the sequence is Y0Cr0, Y1Cb0, ... src1_deep_color=1, src1_format=2'b10: 10-bit 444 in one canvas -- 10-bit Y + 10-bit Cb + 10-bit Cr + 2-bit stuffing = 32-bit per pixel in canvas. If src1_color_map=4'b0000, the sequence is Y0Cb0Cr0, Y1Cb1Cr1, ... If src1_color_map=other value, the sequence is Cr0Cb0Y0, Cr1Cb1Y1, ...
1-0	R/W	0	SRC1_FORMAT: define input pixel byte-width. Note: If SRC1_DEEPCOLOR=0, the SRC1_FORMAT's definitions is as below. If SRC1_DEEPCOLOR=1, please refer to SRC1_DEEPCOLOR entry for new meaning. 0: Input pixel is 1-byte (8-bit), it is either an 8-bit color component or 8-bit address to look up a 32-bit pixel, refer to GE2D_GEN_CTRL0. SRC1_LUT_EN; 1: Input pixel is 2-byte (16-bit), refer to GE2D_GEN_CTRL2.SRC1_COLOR_MAP for further pixel color mapping; 2: Input pixel is 3-byte (24-bit), refer to GE2D_GEN_CTRL2.SRC1_COLOR_MAP for further pixel color mapping; 3: Input pixel is 4-byte (32-bit), refer to GE2D_GEN_CTRL2.SRC1_COLOR_MAP for further pixel color mapping.

GE2D_CMD_CTRL 0x8A3

Bit(s)	R/W	Default	Description
31-10	R	0	unused
9	R/W	0	SRC2_FILL_COLOR_EN: if true, all src2 data use default color.
8	R/W	0	SRC1_FILL_COLOR_EN: if true, all src1 data use default color.
7	R/W	0	DST_XY_SWAP: if true, dst x/y swap.
6	R/W	0	DST_X_REV: if true, dst x direction reversely read.
5	R/W	0	DST_Y_REV: if true, dst y direction reversely read.
4	R/W	0	SRC2_X_REV: if true, src2 x direction reversely read.
3	R/W	0	SRC2_Y_REV: if true, src2 y direction reversely read.

Bit(s)	R/W	Default	Description
2	R/W	0	SRC1_X_REV: if true, src1 x direction reversely read.
1	R/W	0	SRC1_Y_REV: if true, src1 y direction reversely read.
0	R/W	0	CBUS_CMD_WR: If true, generate a pulse to validate a GE2D command, the command is described by the rest of the field of this register.

GE2D_STATUS0 0x8A4

Bit(s)	R/W	Default	Description
31-29	R	0	unused
28-17	R	0	dst write response counter, for debug only.
16-7	R	0	DP_STATUS: ge2d_dp status, for debug only.
6	R	0	R1CMD_RDY: read src2 cmd ready.
5	R	0	R2CMD_RDY: read src2 cmd ready.
4	R	0	PDPCMD_V: pre dpcmd ready.
3	R	0	DPCMD_RDY: GE2D dpcmd ready.
2	R	0	BUF_CMD_V: GE2D buffer command valid.
1	R	0	CURR_CMD_V: GE2D current command valid.
0	R	0	GE2D_BUSY: GE2D busy.

GE2D_STATUS1 0x8A5

Bit(s)	R/W	Default	Description
31-30	R	0	unused
29-16	R	0	ge2d_dst1_status, for debug only.
15	R	1	ge2d_rd_src2 core.fifo_empty.
14	R	0	ge2d_rd_src2 core.fifo_overflow.
13-12	R	0	ge2d_rd_src2 core.req_st. Same as req_st_y.
11	R	0	ge2d_rd_src2 cmd_if.cmd_err, true if cmd_format=1.
10	R	0	ge2d_rd_src2 cmd_if.cmd_st, 0=IDLE state, 1=BUSY state.
9	R	1	ge2d_rd_src1 luma_core(chroma_core).fifo_empty.
8	R	0	ge2d_rd_src1 luma_core(chroma_core).fifo_overflow.
7-6	R	0	ge2d_rd_src1 chroma_core.req_st_cr. Same as req_st_y.
5-4	R	0	ge2d_rd_src1 chroma_core.req_st_cb. Same as req_st_y.

Bit(s)	R/W	Default	Description
3-2	R	0	ge2d_rd_src1 luma_core.req_st_y. 0: IDLE; 1: WAIT_FIFO_ROOM; 2: REQUEST; 3: WAIT_FINISH.
1	R	0	ge2d_rd_src1 cmd_if.stat_read_window_err, 1=reading/clipping window setting exceed limit.
0	R	0	ge2d_rd_src1 cmd_if.cmd_st, 0=IDLE state, 1=BUSY state.

GE2D_SRC1_DEF_COLOR 0x8A6

Bit(s)	R/W	Default	Description
31-24	R/W	0	Default Y or R.
23-16	R/W	0x80	Default Cb or G.
15-8	R/W	0x80	Default Cr or B.
7-0	R/W	0	Default Alpha.

GE2D_SRC1_CLIPX_START_END 0x8A7

Bit(s)	R/W	Default	Description
31	R/W	0	SRC1 clip x start extra, if true, one more data is read for chroma.
30-29	R	0	Unused.
28-16	R/W	0	SRC1 clip x start.
15	R/W	0	SRC1 clip x end extra, if true, one more data is read for chroma.
14-13	R	0	Unused.
12-0	R/W	0x1FFF	SRC1 clip x end.

GE2D_SRC1_CLIPY_START_END 0x8A8

Bit(s)	R/W	Default	Description
30-29	R	0	Unused.
28-16	R/W	0	SRC1 clip y start.
14-13	R	0	Unused.
12-0	R/W	0x1FFF	SRC1 clip y end.

GE2D_SRC1_CANVAS 0x8A9

Bit(s)	R/W	Default	Description
31-24	R/W	0	SRC1 canvas address0, for Y only or Y/Cb/Cr stored together.
7-0	R	0	Unused.

GE2D_SRC1_X_START_END 0x8AA

Bit(s)	R/W	Default	Description
31	R/W	0	SRC1 x start extra bit1, if true, one more chroma data is read for x even start chroma data when y/c ratio = 2 or x even/odd start chroma extra data when y/c ratio = 1
30	R/W	0	SRC1 x start extra bit0, if true, one more chroma data is read for x odd start chroma data when y/c ratio = 2
29-16	R/W	0	SRC1 x start, signed data
15	R/W	0	SRC1 x end extra bit1, if true, one more chroma data is read for x odd end chroma data when y/c ratio = 2 or x even/odd end chroma extra data when y/c ratio = 1
14	R/W	0	SRC1 x end extra bit0, if true, one more chroma data is read for x even end chroma data when y/c ratio = 2
13-0	R/W	0	SRC1 x end, signed data.

GE2D_SRC1_Y_START_END 0x8AB

Bit(s)	R/W	Default	Description
31	R/W	0	SRC1 y start extra bit1, if true, one more chroma line is read for y even start chroma data when y/c ratio = 2 or x even/odd start chroma extra data when y/c ratio = 1
30	R/W	0	SRC1 y start extra bit0, if true, one more chroma line is read for y odd start chroma data when y/c ratio = 2
29-16	R/W	0	SRC1 y start, signed data
15	R/W	0	SRC1 y end extra bit1, if true, one more chroma line is read for y odd end chroma data when y/c ratio = 2 or y even/odd end chroma extra data when y/c ratio = 1
14	R/W	0	SRC1 y end extra bit0, if true, one more chroma line is read for y even end chroma data when y/c ratio = 2
13-0	R/W	0	SRC1 y end, signed data.

GE2D_SRC1_LUT_ADDR 0x8AC

Bit(s)	R/W	Default	Description
31-9	R	0	Unused.
8	R/W	1	0 = Write LUT, 1 = Read LUT.
7-0	R/W	0	LUT_ADDR: The initial read or write address of the look-up table

GE2D_SRC1_LUT_DAT 0x8AD

Bit(s)	R/W	Default	Description
31-24	R/W	0	Current LUT entry's Y or R
23-16	R/W	0	Current LUT entry's Cb or G
15-8	R/W	0	Current LUT entry's Cr or B
7-0	R/W	0	Current LUT entry's Alpha.

GE2D_SRC1_FMT_CTRL 0x8AE

Bit(s)	R/W	Default	Description
31-20	R	0	Unused.
19	R/W	0	SRC1_CHFMT_RPT_PIX: if true, horizontal formatter using repeat to get the pixel, otherwise using interpolation.
18	R/W	0	SRC1_CHFMT_EN: horizontal formatter enable.
17	R/W	0	SRC1_CVFMT_RPT_PIX: if true, vertical formatter using repeat to get the pixel, otherwise using interpolation.
16	R/W	0	SRC1_CVFMT_EN: vertical formatter enable.
15-8	R/W	0	SRC1_X_CHR_PHASE: X direction chroma phase, Bit[15:12] for x direction even start/end chroma phase when y/c ratio = 2 or start/end even/odd chroma phase when y/c ratio = 1; Bit[11:8] for x direction odd start/end chroma phase only when y/c ration = 2.
7-0	R/W	0	SRC1_Y_CHR_PHASE:Y direction chroma phase. Bit[7:4] for y direction even start/end chroma phase when y/c ratio = 2 or start/end even/odd chroma phase when y/c ratio = 1; Bit[3:0] for y direction odd start/end chroma phase only when y/c ration = 2.

GE2D_SRC2_DEF_COLOR 0x8AF

Bit(s)	R/W	Default	Description
31-24	R/W	0	Default Y or R.
23-16	R/W	0x80	Default Cb or G.
15-8	R/W	0x80	Default Cr or B.
7-0	R/W	0	Default Alpha.

GE2D_SRC2_CLIPX_START_END 0x8B0

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	SRC2 clip x start.

Bit(s)	R/W	Default	Description
15-13	R	0	Unused.
12-0	R/W	0x1FFF	SRC2 clip x end.

GE2D_SRC2_CLIPY_START_END 0x8B1

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	SRC2 clip y start.
15-13	R	0	Unused.
12-0	R/W	0x1FFF	SRC2 clip y end.

GE2D_SRC2_X_START_END 0x8B2

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	SRC2 x start.
15-13	R	0	Unused.
12-0	R/W	0	SRC2 x end.

GE2D_SRC2_Y_START_END 0x8B3

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	SRC2 y start.
15-13	R	0	Unused.
12-0	R/W	0	SRC2 y end.

GE2D_DST_CLIPX_START_END 0x8B4

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	DST clip x start.
15-13	R	0	Unused.
12-0	R/W	0x1FFF	DST clip x end.

GE2D_DST_CLIPY_START_END 0x8B5

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.

28-16	R/W	0	DST clip y start.
15-13	R	0	Unused.
12-0	R/W	0x1FFF	DST clip y end.

GE2D_DST_X_START_END 0x8B6

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	DST x start.
15-13	R	0	Unused.
12-0	R/W	0	DST x end.

GE2D_DST_Y_START_END 0x8B7

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	DST y start.
15-13	R	0	Unused.
12-0	R/W	0	DST y end.

GE2D_SRC2_DST_CANVAS 0x8B8

Bit(s)	R/W	Default	Description
31-24	R	0	Unused.
23-16	R/W	0	DST2 canvas address.
15-8	R/W	0	SRC2 canvas address.
7-0	R/W	0	DST1 canvas address.

GE2D_VSC_START_PHASE_STEP 0x8B9

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-0	R/W	0x01000000	5.24 format.

GE2D_VSC_PHASE_SLOPE 0x8BA

Bit(s)	R/W	Default	Description
31-25	R	0	Unused.
24-0	R/W	0	Signed data.

GE2D_VSC_INI_CTRL 0x8BB

Bit(s)	R/W	Default	Description
31	R	0	Unused.
30-29	R/W	0	vertical repeat line0 number.
28-24	R	0	Unused.
23-0	R/W	0	vertical scaler initial phase.

GE2D_HSC_START_PHASE_STEP 0x8BC

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-0	R/W	0x01000000	5.24 format.

GE2D_HSC_PHASE_SLOPE 0x8BD

Bit(s)	R/W	Default	Description
31-25	R	0	Unused.
24-0	R/W	0	Signed data.

GE2D_HSC_INI_CTRL 0x8BE

Bit(s)	R/W	Default	Description
31	R	0	Unused.
30-29	R/W	0	horizontal repeat line0 number.
28-24	R	0	Unused.
23-0	R/W	0	horizontal scaler initial phase.

GE2D_HSC_ADV_CTRL 0x8BF

Bit(s)	R/W	Default	Description
31-24	R/W	0	advance number in this round, if horizontal scaler is working on dividing mode.
23-0	R/W	0	horizontal scaler advance phase in this round, if horizontal scaler is working on dividing mode.

GE2D_SC_MISC_CTRL 0x8C0

Bit(s)	R/W	Default	Description
31	R	0	Unused.
30	R/W	0	VSC_NEAREST_EN: vertical nearest mode enable, must set vt_bank_length = 4.
29	R/W	0	HSC_NEAREST_EN: horizontal nearest mode enable, must set hz_bank_length = 4.

28	R/W	0	HSC_DIV_EN: horizontal scaler dividing mode enable.
27-15	R/W	0	HSC_DIV_LENGTH: horizontal dividing length, if bit 25 is enable.
14	R/W	0	pre horizontal scaler enable.
13	R/W	0	pre vertical scale enable.
12	R/W	0	vertical scale enable.
11	R/W	0	horizontal scaler enable.
10	R	0	Unused.
9	R/W	0	HSC_RPT_CTRL: if true, treat horizontal repeat line number(GE2D_HSC_INI_CTRL bit 30:29) as repeating line, otherwise using treat horizontal repeat line number as minus line number.
8	R/W	0	VSC_RPT_CTRL: if true, treat vertical repeat line number(GE2D_VSC_INI_CTRL bit 30:29) as repeating line, otherwise using treat vertical repeat line number as minus line number.
7	R/W	0	VSC_PHASE0_ALWAYS_EN: if true, always use phase0 in vertical scaler.
6-4	R/W	2	VSC_BANK_LENGTH: vertical scaler bank length.
3	R/W	0	HSC_PHASE0_ALWAYS_EN: if true, always use phase0 in horizontal scaler.
2-0	R/W	2	HSC_BANK_LENGTH: horizontal scaler bank length.

GE2D_VSC_NRND_POINT 0x8C1

Bit(s)	R/W	Default	Description
31-14	R	0	Unused.
13-0	R	0	vertical scaler next round integer pixel pointer, signed data.

GE2D_VSC_NRND_PHASE 0x8C2

Bit(s)	R/W	Default	Description
31-24	R	0	Unused.
23-0	R	0	vertical scaler next round phase.

GE2D_HSC_NRND_POINT 0x8C3

Bit(s)	R/W	Default	Description
31-14	R	0	Unused.
13-0	R	0	horizontal scaler next round integer pixel pointer, signed data.

GE2D_HSC_NRND_PHASE 0x8C4

Bit(s)	R/W	Default	Description
31-24	R	0	Unused.

23-0	R	0	horizontal scaler next round phase.

GE2D_MATRIX_PRE_OFFSET 0x8C5

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-20	R/W	0	pre_offset0.
19	R	0	Unused.
18-10	R/W	0	pre_offset1.
9	R	0	Unused.
8-0	R/W	0	pre_offset2.

GE2D_MATRIX_COEF00_01 0x8C6

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	coef00.
15-13	R	0	Unused.
12-0	R/W	0	coef01.

GE2D_MATRIX_COEF02_10 0x8C7

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	Coef02.
15-13	R	0	Unused.
12-0	R/W	0	Coef10.

GE2D_MATRIX_COEF11_12 0x8C8

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	Coef11.
15-13	R	0	Unused.
12-0	R/W	0	Coef12.

GE2D_MATRIX_COEF20_21 0x8C9

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.

28-16	R/W	0	Coef20.
15-13	R	0	Unused.
12-0	R/W	0	Coef21.

GE2D_MATRIX_COEF22_CTRL 0x8CA

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-16	R/W	0	coef22.
15-8	R	0	Unused.
7	R/W	0	input y/cb/cr saturation enable.
6-1	R	0	Unused.
0	R/W	0	conversion matrix enable.

GE2D_MATRIX_OFFSET 0x8CB

Bit(s)	R/W	Default	Description
31-29	R	0	Unused.
28-20	R/W	0	offset0.
19	R	0	Unused.
18-10	R/W	0	offset1.
9	R	0	Unused.
8-0	R/W	0	offset2.

GE2D_ALU_OP_CTRL 0x8CC

Bit(s)	R/W	Default	Description
31-27	R	0	Unused.
26-25	R/W	0	SRC1 color multiplier alpha selection. if 00, Cs = Csr if 01, Cs = Csr * Asr * Ag (if source is not premultiplied) if 10, Cs = Csr * Ag (if source is premultipied).
24	R/W	0	SRC2 color multiplier alpha selection. if 0, no multiplier, Cd = Cdr, otherwise, Cd = Cdr * Ad.
23	R	0	Unused.
22-12	R/W	0x010	ALU color operation. Bit[22:20] Blending Mode Parameter. 3'b000: ADD Cs*Fs + Cd*Fd

Bit(s)	R/W	Default	Description
			3'b001: SUBTRACT Cs*Fs - Cd*Fd 3'b010: REVERSE SUBTRACT Cd*Fd - Cs*Fs 3'b011: MIN min(Cs*Fs, Cd*Fd) 3'b100: MAX max(Cs*Fs, Cd*Fd) 3'b101: LOGIC OP Cs op Cd .... reserved Bit[19:16] Source Color Blending Factor CFs. 4'b0000: ZERO 0 4'b0001: ONE 1 4'b0010: SRC_COLOR Cs(RGBs) 4'b0011: ONE_MINUS_SRC_COLOR 1 - Cs(RGBs) 4'b0100: DST_COLOR Cd(RGBd) 4'b0101: ONE_MINUS_DST_COLOR 1 - Cd(RGBd) 4'b0110: SRC_ALPHA As 4'b0111: ONE_MINUS_SRC_ALPHA 1 – As 4'b1000: DST_ALPHA Ad 4'b1001: ONE_MINUS_DST_ALPHA 1 – Ad 4'b1010: CONST_COLOR Cc(RGBc) 4'b1011: ONE_MINUS_CONST_COLOR 1 - Cc(RGBc) 4'b1100: CONST_ALPHA Ac 4'b1101: ONE_MINUS_CONST_ALPHA 1 – Ac 4'b1110: SRC_ALPHA_SATURATE min(As,1-Ad) .... reserved Bit[15:12] dest Color Blending Factor CFd, when bit[22:20]!= LOGIC OP. 4'b0000: ZERO 0 4'b0001: ONE 1 4'b0010: SRC_COLOR Cs(RGBs) 4'b0011: ONE_MINUS_SRC_COLOR 1 - Cs(RGBs) 4'b0100: DST_COLOR Cd(RGBd) 4'b0101: ONE_MINUS_DST_COLOR 1 - Cd(RGBd) 4'b0110: SRC_ALPHA As 4'b0111: ONE_MINUS_SRC_ALPHA 1 – As 4'b1000: DST_ALPHA Ad 4'b1001: ONE_MINUS_DST_ALPHA 1 – Ad 4'b1010: CONST_COLOR Cc(RGBc) 4'b1011: ONE_MINUS_CONST_COLOR 1 - Cc(RGBc) 4'b1100: CONST_ALPHA Ac 4'b1101: ONE_MINUS_CONST_ALPHA 1 – Ac 4'b1110: SRC_ALPHA_SATURATE min(As,1-Ad) .... reserved Bit[15:12] logic operations, when bit[22:20]== LOGIC OP. 4'b0000: CLEAR 0 4'b0001: COPY s 4'b0010: NOOP d 4'b0011: SET 1 4'b0100: COPY_INVERT ~s 4'b0101: INVERT ~d 4'b0110: AND_REVERSE s & ~d 4'b0111: OR_REVERSE s | ~d 4'b1000: AND s & d 4'b1001: OR s | d 4'b1010: NAND ~(s & d) 4'b1011: NOR ~(s | d) 4'b1100: XOR s ^ d 4'b1101: EQUIV ~(s ^ d) 4'b1110: AND_INVERTED ~s & d 4'b1111: OR_INVERTED ~s | d
11	R	0	Unused.
10-0	R/W	0x010	ALU alpha operation.

Bit(s)	R/W	Default	Description
			Bit[10:8] Blending Equation Math Operation. 3'b000: ADD As*Fs + Ad*Fd 3'b001: SUBTRACT As*Fs - Ad*Fd 3'b010: REVERSE SUBTRACT Ad*Fd - As*Fs 3'b011: MIN min(As*Fs, Ad*Fd) 3'b100: MAX max(As*Fs, Ad*Fd) 3'b101: LOGIC OP As op Ad .... reserved Bit[7:4] Source alpha Blending Factor AFs. 4'b0000 0 4'b0001 1 4'b0010 As 4'b0011 1 - As 4'b0100 Ad 4'b0101 1 – Ad 4'b0110 Ac 4'b0111 1 – Ac .... reserved Bit[3:0] Destination alpha Blending Factor AFd, when bit[10:8] != LOGIC OP. 4'b0000 0 4'b0001 1 4'b0010 As 4'b0011 1 - As 4'b0100 Ad 4'b0101 1 - Ad 4'b0110 Ac 4'b0111 1 - Ac .... reserved Bit[3:0] logic operations, when bit[10:8] == LOGIC OP. 4'b0000: CLEAR 0 4'b0001: COPY s 4'b0010: NOOP d 4'b0011: SET 1 4'b0100: COPY_INVERT ~s 4'b0101: INVERT ~d 4'b0110: AND_REVERSE s & ~d 4'b0111: OR_REVERSE s | ~d 4'b1000: AND s & d 4'b1001: OR s | d 4'b1010: NAND ~(s & d) 4'b1011: NOR ~(s | d) 4'b1100: XOR s ^ d 4'b1101: EQUIV ~(s ^ d) 4'b1110: AND_INVERTED ~s & d 4'b1111: OR_INVERTED ~s | d

GE2D_ALU_CONST_COLOR 0x8CD

Bit(s)	R/W	Default	Description
31-0	R/W	0x00808000	RGBA or YCbCrA.

GE2D_SRC1_KEY 0x8CE

Bit(s)	R/W	Default	Description
31-0	R/W	0	SRC1 Key.

GE2D_SRC1_KEY_MASK 0x8CF

Bit(s)	R/W	Default	Description
31-0	R/W	0	SRC1 Key Mask.

GE2D_SRC2_KEY 0x8D0

Bit(s)	R/W	Default	Description
31-0	R/W	0	SRC2 Key.

GE2D_SRC2_KEY_MASK 0x8D1

Bit(s)	R/W	Default	Description
31-0	R/W	0	Destination Bit Mask.

GE2D_DP_ONOFF_CTRL 0x8D3

Bit(s)	R/W	Default	Description
31	R/W	0	DP onoff mode. 0: on_counter means how many pixels will output before ge2d turns off; 1: on_counter means how many clocks will ge2d turn on before ge2d turns off.
30-16	R/W	0	DP on counter.
15	R/W	0	0: vd_format doesnt have onoff mode, 1: vd format has onoff mode.
14-0	R/W	0	DP off counter.

GE2D_SCALE_COEF_IDX 0x8D4

Because there are many coefficients used in the vertical filter and horizontal filters, indirect access the coefficients of vertical filter and horizontal filter is used. For vertical filter, there are 33x4 coefficients. For horizontal filter, there are 33x4 coefficients

Bit(s)	R/W	Default	Description
31-16	R	0	Unused.
15	R/W	0	index increment, if bit9 == 1 then (0: index increase 1, 1: index increase 2) else (index increase 2).
14	R/W	0	1: read coef through cbus enable, just for debug purpose in case when we wanna check the coef in ram in correct or not.
13-10	R	0	Unused.
9	R/W	0	if true, use 9bit resolution coef, other use 8bit resolution coef.
8	R/W	0	type of index, 0: vertical coef; 1: horizontal coef.
7	R	0	Unused.
6-0	R/W	0	coef index.

GE2D_SCALE_COEF 0x8D5

Bit(s)	R/W	Default	Description
31-0	R/W	0	coefficients for vertical filter and horizontal filter.

GE2D_SRC_OUTSIDE_ALPHA 0x8D6

Bit(s)	R/W	Default	Description
31-25	R	0	Unused.

24	R/W	0	src2 alpha fill mode: together with GE2D_GEN_CTRL0[14](fill_mode), define what alpha values are used.
			0: repeat innner alpha, 1: fill src2 outside alpha.
			
			for the area outside the clipping window. As below:
			fill_mode=0, alpha_fill_mode=0 : use inner alpha, (or default_alpha if src data have no alpha values);
			fill_mode=0, alpha_fill_mode=1 : use outside_alpha;
			fill_mode=1, alpha_fill_mode=0 : use default_alpha;
			fill_mode=1, alpha_fill_mode=1 : use outside_alpha.

23-16	R/W	16	src2 outside alpha.
15-9	R	0	Unused.
8	R/W	1	src1 alpha fill mode, refer to src2 alpha fill mode above.
7-0	R/W	0	src1 outside alpha.

GE2D_ANTIFLICK_CTRL0 0x8D8

Bit(s)	R/W	Default	Description
31	R/W	0	antiflick enable
24	R/W	0	1: alpha value for the first line use repeated alpha, 0: use bit 23:16 as the first line alpha
23-16	R/W	0	register value for the first line alpha when bit 24 is 1.
8	R/W	0	1: alpha value for the last line use repeated alpha, 0: use bit 7:0 as the last line alpha
7-0	R/W	0	register value for the last line alpha when bit 8 is 1.

GE2D_ANTIFLICK_CTRL1 0x8D9

Bit(s)	R/W	Default	Description
25	R/W	0	rgb_sel, 1: antiflick RGBA, 0: antiflick YCbCrA
24	R/W	0	cbcr_en, 1: also filter cbcr in case of antiflicking YCbCrA, 0: no filter on cbcr in case of antiflicking YCbCrA
23-16	R/W	0	R mult coef for converting RGB to Y
15- 8	R/W	0	G mult coef for converting RGB to Y

Bit(s)	R/W	Default	Description
7-0	R/W	0	B mult coef for converting RGB to Y

Y = (R * y_r + G * y_g + B * y_b) / 256

GE2D_ANTIFLICK_COLOR_FILT0 0x8DA

Bit(s)	R/W	Default	Description
31-24	R/W	0	Y threhold1, when 0<Y<=th1, use filter0.
23-16	R/W	0	color antiflick filter0 n3
15- 8	R/W	0	color antiflick filter0 n2
7-0	R/W	0	color antiflick filter0 n1

Y = (line_up * n1 + line_center * n2 + line_dn * n3) / 128

GE2D_ANTIFLICK_COLOR_FILT1 0x8DB

Bit(s)	R/W	Default	Description
31-24	R/W	0	Y threhold2, when th1<Y<=th2, use filter1.
23-16	R/W	0	color antiflick filter1 n3
15- 8	R/W	0	color antiflick filter1 n2
7-0	R/W	0	color antiflick filter1 n1

GE2D_ANTIFLICK_COLOR_FILT2 0x8DC

Bit(s)	R/W	Default	Description
31-24	R/W	0	Y threhold3, when th2<Y<=th3, use filter2 ; Y>th3, use filter3.
23-16	R/W	0	color antiflick filter2 n3
15- 8	R/W	0	color antiflick filter2 n2
7-0	R/W	0	color antiflick filter2 n1

GE2D_ANTIFLICK_COLOR_FILT3 0x8DD

Bit(s)	R/W	Default	Description
23-16	R/W	0	color antiflick filter3 n3
15- 8	R/W	0	color antiflick filter3 n2
7-0	R/W	0	color antiflick filter3 n1

GE2D_ANTIFLICK_ALPHA_FILT0 0x8DE

Bit(s)	R/W	Default	Description
31-24	R/W	0	Alpha threhold1, when 0< Alpha<=th1, use filter0.
23-16	R/W	0	Alpha antiflick filter0 n3

15- 8	R/W	0	Alpha antiflick filter0 n2
7-0	R/W	0	Alpha antiflick filter0 n1

Alpha = (line_up * n1 + line_center * n2 + line_dn * n3) / 128

GE2D_ANTIFLICK_ALPHA_FILT1 0x8DF

Bit(s)	R/W	Default	Description
31-24	R/W	0	Alpha threhold2, when th1< Alpha<=th2, use filter1.
23-16	R/W	0	Alpha antiflick filter1 n3
15- 8	R/W	0	Alpha antiflick filter1 n2
7-0	R/W	0	Alpha antiflick filter1 n1

GE2D_ANTIFLICK_ALPHA_FILT2 0x8E0

Bit(s)	R/W	Default	Description
31-24	R/W	0	Alpha threhold3, when th2< Alpha<=th3, use filter2; Alpha>th3, use filter3.
23-16	R/W	0	Alpha antiflick filter2 n3
15- 8	R/W	0	Alpha antiflick filter2 n2
7-0	R/W	0	Alpha antiflick filter2 n1

GE2D_ANTIFLICK_ALPHA_FILT3 0x8E1

Bit(s)	R/W	Default	Description
23-16	R/W	0	Alpha antiflick filter3 n3
15- 8	R/W	0	Alpha antiflick filter3 n2
7-0	R/W	0	Alpha antiflick filter3 n1

GE2D_SRC1_RANGE_MAP_Y_CTRL 0x8E3

Bit(s)	R/W	Default	Description
30-22	R/W	0	din_offset (signed data)
21-14	R/W	0	map_coef (unsigned data)
13- 10	R/W	0	map_sr (unsigned data)
9-1	R/W	0	dout_offset (signed data)
0	R/W	0	enable

dout = clipto_0_255(((din + din_offset) * map_coef + ((1 << (map_sr - 1))) >> map_sr + dout_offset)

GE2D_SRC1_RANGE_MAP_CB_CTRL 0x8E4

Bit(s)	R/W	Default	Description
30-22	R/W	0	din_offset (signed data)

21-14	R/W	0	map_coef (unsigned data)
13- 10	R/W	0	map_sr (unsigned data)
9-1	R/W	0	dout_offset (signed data)
0	R/W	0	enable

dout = clipto_0_255(((din + din_offset) * map_coef + ((1 << (map_sr - 1))) >> map_sr + dout_offset)

GE2D_SRC1_RANGE_MAP_CR_CTRL 0x8E5

Bit(s)	R/W	Default	Description
30-22	R/W	0	din_offset (signed data)
21-14	R/W	0	map_coef (unsigned data)
13- 10	R/W	0	map_sr (unsigned data)
9-1	R/W	0	dout_offset (signed data)
0	R/W	0	enable

dout = clipto_0_255(((din + din_offset) * map_coef + ((1 << (map_sr - 1))) >> map_sr + dout_offset)

GE2D_ARB_BURST_NUM 0x8E6

Bit(s)	R/W	Default	Description
21-16	R/W	0x3f	Src1 prearbitor burst number
13-8	R/W	0x3f	Src2 prearbitor burst number
5-0	R/W	0x3f	dst prearbitor burst number

GE2D_TID_TOKEN 0x8E7

Bit(s)	R/W	Default	Description
21-16	R/W	0x3f	Src1 ID. High 4bit are thread ID, low 2bits are the token
13-8	R/W	0x3f	Src2 ID. High 4bit are thread ID, low 2bits are the token
5-0	R/W	0x3f	dst ID. High 4bit are thread ID, low 2bits are the token

GE2D_GEN_CTRL3 0x8E8

Bit(s)	R/W	Default	Description
31-28	R/W	0	DST2_BYTEMASK_VAL: 1-bit mask for each byte (8-bit). Applicable only if both dst_bitmask_en=1 and dst_bytemask_only=1.
27-26	R/W	0	DST2_PIC_STRUCT: Define how destination 2 write the picture to DDR memory. 0: Write all lines (whole frame); 1: Reserved; 2: Write even lines only (top); 3: Write odd lines only (bottom).

Bit(s)	R/W	Default	Description
25-24	R/W	0	DST2_8B_MODE_SEL: Destination 8-bit mode component selection. 0: Select Y or R; 1: Select Cb or G; 2: Select Cr or B; 3: Select Alpha.
23	R	0	Unused
22-19	R/W	0	DST2_COLOR_MAP: Applicable to 16-bit, 24-bit and 32-bit pixel, defines the bit-field allocation of the pixel data. For whether to truncate or round full 8-bit to output, refer to GE2D_GEN_CTRL3.DST2_COLOR_ROUND_MODE. For 16-bit mode (DST2_FORMAT=1): 0 = Unused; 1 = 6:5:5 format. Bit[15:10] is Y[7:2] or R[7:2], bit[9:5] is Cb[7:3] or G[7:3], bit[4:0] is Cr[7:3] or B[7:3]; 2 = 8:4:4 format. Bit[15:8] is Y or R, bit[7:4] is Cb[7:4] or G[7:4], bit[3:0] is Cr[7:4] or B[7:4]]; 3 = 6:4:4:2 format. Bit[15:10] is Y[7:2] or R[7:2], bit[9:6] is Cb[7:4] or G[7:4], bit[5:2] is Cr[7:4] or B[7:4], bit[1:0] is Alpha[7:6]; 4 = 4:4:4:4 format. Bit[15:12] is Y[7:4] or R[7:4], bit[11:8] is Cb[7:4] or G[7:4], bit[7:4] is Cr[7:4] or B[7:4], bit[3:0] is Alpha[7:4]; 5 = 5:6:5 format. Bit[15:11] is Y[7:3] or R[7:3], bit[10:5] is Cb[7:2] or G[7:2], bit[4:0] is Cr[7:3] or B[7:3]; 6 = 4:4:4:4 format. Bit[15:12] is Alpha[7:4], bit[11:8] is Y[7:4] or R[7:4], bit[7:4] is Cb[7:4] or G[7:4], bit[3:0] is Cr[7:4] or B[7:4]; 7 = 1:5:5:5 format. Bit[15] is Alpha[7], bit[14:10] is Y[7:3] or R[7:3], bit[9:5] is Cb[7:3] or G[7:3], bit[4:0] is Cr[7:3] or B[7:3]; 8 = 4:6:4:2 format. Bit[15:12] is Y[7:4] or R[7:4], bit[11:6] is Cb[7:2] or G[7:2], bit[5:2] is Cr[7:4] or B[7:4], bit[1:0] is Alpha[1:0]; 9 = CbCr format. Bit[15:8] is Cb, bit[7:0] is Cr; 10 = CrCb format. Bit[15:8] is Cr, bit[7:0] is Cb. For 24-bit mode (DST2_FORMAT=2): 0 = RGB 8:8:8 mode. Bit[23:16] is Y or R, bit[15:8] is Cb or G, bit[7:0] is Cr or B; 1 = RGBA 5:6:5:8 mode. Bit[23:19] is Y[7:3] or R[7:3], bit[18:13] is Cb[7:2] or G[7:2], bit[12:8] is Cr[7:3] or B[7:3], bit[7:0] is Alpha; 2 = ARGB 8:5:6:5 mode. Bit[23:16] is Alpha, bit[15:11] is Y[7:3] or R[7:3], bit[10:5] is Cb[7:2] or G[7:2], bit[4:0] is Cr[7:3] or B[7:3]; 3 = RGBA 6:6:6:6 mode. Bit[23:18] is Y[7:2] or R[7:2], bit[17:12] is Cb[7:2] or G[7:2], bit[11:6] is Cr[7:2] or B[7:2], bit[5:0] is Alpha[7:2]; 4 = ARGB 6:6:6:6 mode. Bit[23:18] is Alpha[7:2];, bit[17:12] is Y[7:2] or R[7:2], bit[11:6] is Cb[7:2] or G[7:2], bit[5:0] is Cr[7:2] or B[7:2]; 5 = BGR 8:8:8 mode. Bit[23:16] is Cr or B, bit[15:8] is Cb or G, bit[7:0] is Y or R. For 32-bit mode (DST2_FORMAT=3):

Bit(s)	R/W	Default	Description
			0 = RGBA 8:8:8:8 format. Bit[31:24] is Y or R, bit[23:16] is Cb or G; bit[15:8] is Cr or B; bit[7:0] is Alpha; 1 = ARGB 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Y or R; bit[15:8] is Cb or G; bit[7:0] is Cr or B; 2 = ABGR 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Cr or B; bit[15:8] is Cb or G; bit[7:0] is Y or R; 3 = BGRA 8:8:8:8 format. Bit[31:24] is Cr or B, bit[23:16] is Cb or G; bit[15:8] is Y or R; bit[7:0] is Alpha.
18	R	0	Unused
17-16	R/W	0	DST2_FORMAT: define DST2 output pixel byte-width. 0: Output pixel is 1-byte (8-bit) color component; 1: Output pixel is 2-byte (16-bit), refer to GE2D_GEN_CTRL3.DST2_COLOR_MAP for further pixel color mapping; 2: Output pixel is 3-byte (24-bit), refer to GE2D_GEN_CTRL3.DST2_COLOR_MAP for further pixel color mapping; 3: Output pixel is 4-byte (32-bit), refer to GE2D_GEN_CTRL3.DST2_COLOR_MAP for further pixel color mapping.
15	R	0	Unused
14	R/W	0	DST2_COLOR_ROUND_MODE. 1 = Truncate the full bit color components to required output bit width; 0 = Round (+ 0.5) the full bit color components to required output bit width.
13-12	R/W	0	DST2_X_DISCARD_MODE: Define how DST2 discard X direction data before writing to DDR. Note: x is post reverse/rotation. 0: No discard; 1: Reserved; 2: discard even x; 3: discard odd x.
11-10	R/W	0	DST2_Y_DISCARD_MODE: Define how DST2 discard Y direction data before writing to DDR. Note: y is post reverse/rotation. 0: No discard; 1: Reserved; 2: discard even y; 3: discard odd y.
9	R	0	Unused
8	R/W	0	DST2_ENABLE: 0: Disable destination 2; 1: Enable destination 2.
7-6	R	0	Unused

Bit(s)	R/W	Default	Description
5-4	R/W	0	DST1_X_DISCARD_MODE: Define how DST1 discard X direction data before writing to DDR. Note: x is post reverse/rotation. 0: No discard; 1: Reserved; 2: discard even x; 3: discard odd x.
3-2	R/W	0	DST1_Y_DISCARD_MODE: Define how DST1 discard Y direction data before writing to DDR. Note: y is post reverse/rotation. 0: No discard; 1: Reserved; 2: discard even y; 3: discard odd y.
1	R	0	Unused
0	R/W	1	DST1_ENABLE: 0: Disable destination 1; 1: Enable destination 1.

GE2D_STATUS2 0x8E9

Bit(s)	R/W	Default	Description
13-12	R	0	ge2d_dst1.ctrl_status
11	R	0	ge2d_dst1.map_srdy
10	R	0	ge2d_dst1.map_d1_srdy
9	R	0	ge2d_dst1.s_v
8	R	0	ge2d_dst1.ofifo_dout_srdy
7-1	R	0	ge2d_dst1.ofifo_cnt
0	R	0	ge2d_dst1.dst_busy

GE2D_GEN_CTRL4 0x8EA

Bit(s)	R/W	Default	Description
0	R/W	0	dis_dp_hang_bugfix.

GE2D_DST1_BADDR_CTRL 0x8f1

Bit(s)	R/W	Default	Description
31-0	R/W	0	DST1 base address in 64bits.

GE2D_DST1_STRIDE_CTRL 0x8f2

Bit(s)	R/W	Default	Description
19-0	R/W	0	DST1 stride size in 64bits.

GE2D_SRC1_BADDR_CTRL 0x8f3

Bit(s)	R/W	Default	Description
31-0	R/W	0	SRC1 base address in 64bits.

GE2D_SRC1_STRIDE_CTRL 0x8f4

Bit(s)	R/W	Default	Description
19-0	R/W	0	SRC1 stride size in 64bits.

GE2D_SRC2_BADDR_CTRL 0x8f5

Bit(s)	R/W	Default	Description
31-0	R/W	0	SRC2 base address in 64bits.

GE2D_SRC2_STRIDE_CTRL 0x8f6

Bit(s)	R/W	Default	Description
19-0	R/W	0	SRC2 stride size in 64bits.

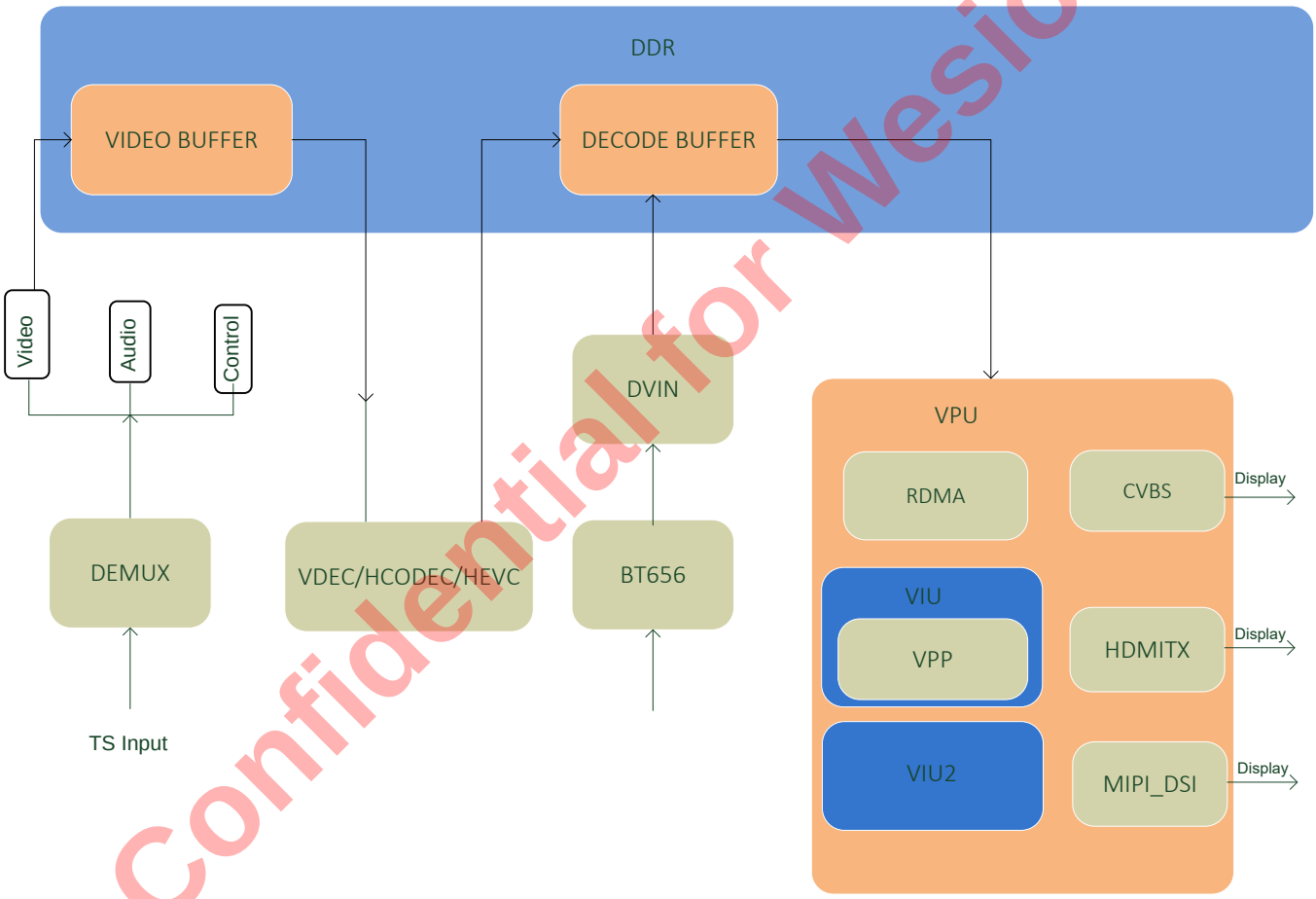
10. Video Path

The data path of the video path module is shown in Figure 10-1 below:

This section describe S922X video path from the following aspects:

- Video Input
- Video Output
  - RDMA
  - VPP(VIU/VPP)
  - CVBS
  - HDMITX
  - MIPI-DSI

Figure 10-1 Data Path of Video Path



10.1 Video Input

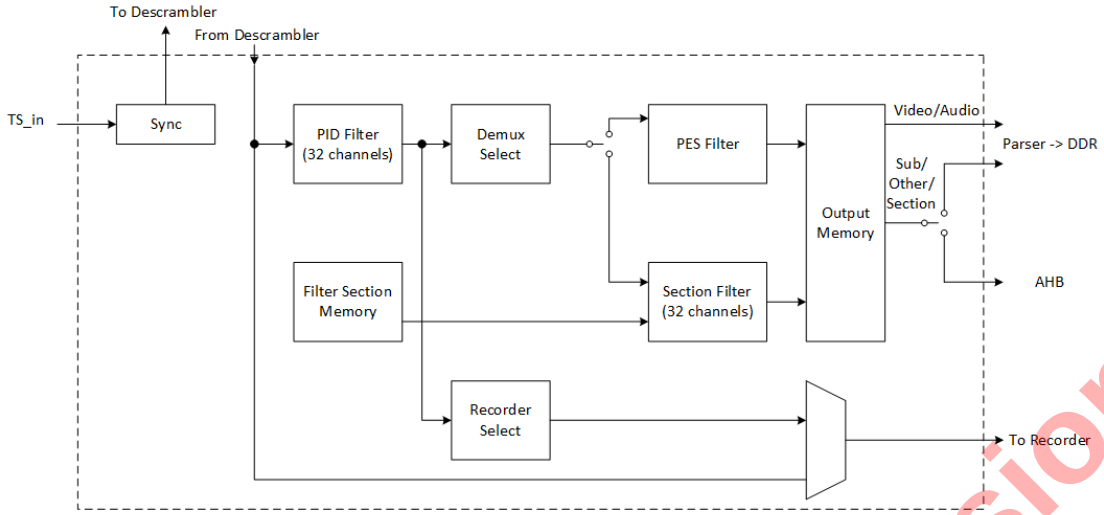
10.1.1 Overview

This part describes the video input module of S922X,include DEMUX.

S922X uses DEMUX to handle input video signals, and store the demuxed signal in DDR, which can be read directrly.

Demux submodule of S922X is designed for connecting to external digital TV tuner/demodular and de-mux the input signal. It decomposes the input signal from TS/DDR into video signal, audio signal, and

clock signal, the video signal will be first written in to DDR, then decoded by the video decoder(VDEC). Below shows the diagram of Demux submodule.



S922X intergrades 3 demux controllers, with the following features:

- Supports PID filter up to 32 channels
- Supports session filter up to 32 channels
- Supports up to 96 channels filter when use 3 demux as one group
- Supports AES/DES/DVB-CSA 1.0 crypto
- Supports inputs from both external TS inputs (serial or parallel, depending on pinmux) and DDR memory (file playback path)
- Supports input loopback to TS-out pins
- Supports separate selection for playback and recorder

10.1.2 Register Description

Demux Common Register

Final address = 0xFFD06000+ offset * 4

STB_TOP_CONFIG 0xf0

Bit(s)	R/W	Default	Description
30:28	RW	0	ciplus_o_sel
27:26	RW	0	ciplus_i_sel
25	RW	0	use FAIL from TS2
24	RW	0	use FAIL from TS1
23	RW	0	use FAIL from TS0
22	RW	0	invert fec_error for S2P1
21	RW	0	invert fec_data for S2P1
20	RW	0	invert fec_sync for S2P1
19	RW	0	invert fec_valid for S2P1

Bit(s)	R/W	Default	Description
18	RW	0	invert fec_clk for S2P1
17:16	RW	0	fec_s_sel for S2P1 00 - select TS0, 01 -- select TS1, 10 -- select TS2, 11 - reserved
15	RW	0	enable_des_pl_clk
14:13	RW	0	reserved
12:10	RW	0	ts_out_select, 0-TS0, 1-TS1, 2-TS2, 3,4-Reserved, 5-S2P1, 6-S2P0, 7-File
9:8	RW	0	des_i_sel 00 -- select_fec_0, 01 -- select_fec_1, 10 -- select_fec_2, 11 - reserved
7	RW	0	enable_des_pl
6	RW	0	invert fec_error for S2P0
5	RW	0	invert fec_data for S2P0
4	RW	0	invert fec_sync for S2P0
3	RW	0	invert fec_valid for S2P0
2	RW	0	invert fec_clk for S2P0
1:0	RW	0	fec_s_sel for S2P0 00 - select TS0, 01 -- select TS1, 10 -- select TS2, 11 - reserved

TS_TOP_CONFIG 0xf1

Bit(s)	R/W	Default	Description
31:28	RW	7	s2p1_clk_div
27:24	RW	7	s2p0_clk_div
23	RW	0	s2p1_disable
22	RW	0	s2p0_disable
21	RW	0	Reserved
20	RW	0	TS_OUT_error_INVERT
19	RW	0	TS_OUT_data_INVERT
18	RW	0	TS_OUT_sync_INVERT
17	RW	0	TS_OUT_valid_INVERT
16	RW	0	TS_OUT_clk_INVERT
15:8	RW	187	TS_package_length_sub_1 (default : 187)
7:0	RW	0x47	fec_sync_byte (default : 0x47)

TS_FILE_CONFIG 0xf2

Bit(s)	R/W	Default	Description
25:24	RW	3	transport_scrambling_control_odd_2 // should be 3
23:16	RW	0	file_m2ts_skip_bytes
15:8	RW	0	des_out_dly
7:6	RW	3	transport_scrambling_control_odd // should be 3
5	RW	0	ts_hiu_enable
4:0	RW	4	fec_clk_div

TS_PL_PID_INDEX 0xf3

Bit(s)	R/W	Default	Description
19:14	R	0	des_2 ts pl state -- Read Only
13:8	R	0	des ts pl state -- Read Only
3:0	RW	0	PID index to 8 PID to get key-set auto increse after TS_PL_PID_DATA read/write

TS_PL_PID_DATA 0xf4

Bit(s)	R/W	Default	Description
29	RW	0	PID #INDEX +1 match disble
28:16	RW	0	PID #INDEX+1
13	RW	0	PID #INDEX match disble
12:0	RW	0	PID #INDEX

COMM_DESC_KEY0 0xf5

Bit(s)	R/W	Default	Description
31:0	RW	0	Common descrambler key (key Bits[63:32])

COMM_DESC_KEY1 0xf6

Bit(s)	R/W	Default	Description
31:0	RW	0	Common descrambler key (key Bits[31:0])

COMM_DESC_KEY_RW 0xf7

Bit(s)	R/W	Default	Description
7	RW	0	Key endian;
6	RW	0	Write key ladder cw [127:64] to key;

Bit(s)	R/W	Default	Description
5	RW	0	Write key ladder cw [63:0] to key;
4	RW	0	0: write to descramble 1; 1: write to descramble 2;
3:0	RW	0	The address of key

CIPLUS_KEY0 0xf8

Bit(s)	R/W	Default	Description
31:0	RW	0	CI+ Register defines Bits[31:0] of the key

CIPLUS_KEY1 0xf9

Bit(s)	R/W	Default	Description
31:0	RW	0	CI+ Register defines Bits[63:32] of the key

CIPLUS_KEY2 0xfa

Bit(s)	R/W	Default	Description
31:0	RW	0	CI+ Register defines Bits[95:64] of the key

CIPLUS_KEY3 0xfb

Bit(s)	R/W	Default	Description
31:0	RW	0	CI+ Register defines Bits[127:96] of the key

CIPLUS_KEY_WR 0xfc

Bit(s)	R/W	Default	Description
5	RW	0	write AES IV B value
4	RW	0	write AES IV A value
3	RW	0	write AES B key
2	RW	0	write AES A key
1	RW	0	write DES B key
0	RW	0	write DES A key

CIPLUS_CONFIG 0xfd

Bit(s)	R/W	Default	Description
15:8	RW	0	TS out delay. This controls the rate at which the CIplus module drives TS out
3	RW	0	General enable for the ciplus module
2	RW	0	AES CBC disable (default should be 0 to enable AES CBC)

Bit(s)	R/W	Default	Description
1	RW	0	AES Enable
0	RW	0	DES Eanble

CIPLUS_ENDIAN 0xfe

Bit(s)	R/W	Default	Description
31:28	RW	0	AES IV endian
27:24	RW	0	AES message out endian
23:20	RW	0	AES message in endian
19:16	RW	0	AES key endian
15:11	RW	0	unused
10:8	RW	0	DES message out endian
6:4	RW	0	DES message in endian
2:0	RW	0	DES key endian

COMM_DESC_2_CTL 0xff

Bit(s)	R/W	Default	Description
15:8	RW	0	des_out_dly_2
7	RW	0	reserved
6	RW	0	enable_des_pl_clk_2
5	RW	0	enable_des_pl_2
4:2	RW	0	use_des_2 Bit[2] -- demux0, Bit[3] -- demux1, Bit[4] -- demux2
1:0	RW	0	des_i_sel_2 00 -- select_fec_0, 01 -- select_fec_1, 10 -- select_fec_2, 11 - reserved

demux core register

demux core 0 Final address = 0xc1105800 + offset * 4

demux core 1 Final address = 0xc1105940 + offset * 4

demux core 2 Final address = 0xc1105a80 + offset * 4

STB_VERSION_O 0x00

Bit(s)	R/W	Default	Description
31:0	R	0x30003	The version of stb

STB_TEST_REG_O 0x01

Bit(s)	R/W	Default	Description

31:0	RW	0xfe015aa5	Test register.

FEC_INPUT_CONTROL_O 0x02

Bit(s)	R/W	Default	Description
15	RW	0	fec_core_select 1 - select descramble output
14:12	RW	0	fec_select 0-TS0, 1-TS1, 2-TS2, 3,4-Reserved, 5-S2P1, 6-S2P0, 7-File
11	RW	0	FEC_CLK
10	RW	0	SOP
9	RW	0	D_VALID
8	RW	0	D_FAIL
7:0	RW	0	D_DATA 7:0

FEC_INPUT_DATA_O 0x03

Bit(s)	R/W	Default	Description
11	R	0	FEC_CLK
20	R	0	SOP
9	R	0	VALID
8	R	0	FAIL
7:0	R	0	FEC DATAIN

DEMUX_CONTROL_O 0x04

Bit(s)	R/W	Default	Description
31	RW	0	enable_free_clk_fec_data_valid
30	RW	0	enable_free_clk_stb_reg
29	RW	0	always_use_pes_package_length
28	RW	0	disable_pre_incomplete_section_fix
27	RW	0	pointer_field_multi_pre_en
26	RW	0	ignore_pre_incomplete_section
25	RW	0	video2_enable
24:22	RW	0	video2_type
21	RW	0	do_not_trust_pes_package_length
20 (bit4)	RW	0	Bypass use recoder path
19 (bit3)	RW	0	clear_PID_continuity_counter_valid

Bit(s)	R/W	Default	Description
18 (bit2)	RW	0	Disable Splicing
17 (bit1)	RW	0	Insert PES_STRONG_SYNC in Audio PES
16 (bit0)	RW	0	Insert PES_STRONG_SYNC in Video PES
15	RW	0	do not trust section length
14	RW	0	om cmd push even zero
13	RW	0	set_buff_ready_even_not_busy
12	RW	0	SUB, OTHER PES interrupt at beginning of PES
11	RW	0	discard_av_package -- for ts_recorder use only
10	RW	0	ts_recorder_select 0:after PID filter 1:before PID filter
9	RW	0	ts_recorder_enable
8	RW	0	(table_id == 0xff) means section_end
7	RW	0	do not send uncomplete section
6	RW	0	do not discard duplicate package
5	RW	0	search SOP when trasport_error_indicator
4	RW	0	stb demux enable
3	RW	0	do not reset state machine on SOP
2	RW	0	search SOP when error happened (when ignore_fail_n_sop, will have this case)
1	RW	0	do not use SOP input (check FEC sync byte instead)
0	RW	0	ignore fec_error bit when non sop (check error on SOP only)

FEC_SYNC_BYTE_O 0x05

Bit(s)	R/W	Default	Description
15:8	RW	187	demux package length - 1 (default : 187)
7:0	RW	0	default is 0x47

FM_WR_DATA_O 0x06

Bit(s)	R/W	Default	Description
31:16	RW	0	filter memory write data hi[31:16]
15:0	RW	0	filter memory write data low [15:0]

FM_WR_ADDR_O 0x07

Bit(s)	R/W	Default	Description
31:24	RW	0	advanced setting hi
23:16	RW	0	advanced setting low
15	R	0	filter memory write data request
7:0	R	0	filter memory write addr

MAX_FM_COMP_ADDR_O 0x08

Bit(s)	R/W	Default	Description
13:8	R	0	demux state -- read only
7:4	RW	0	maxnum section filter compare address
3:0	RW	0	maxnum PID filter compare address

TS_HEAD_0_O 0x09

Bit(s)	R/W	Default	Description
15	RW	0	transport_error_indicator
14	RW	0	payload_unit_start_indicator
13	RW	0	transport_priority
12:0	RW	0	PID

TS_HEAD_1_O 0x0a

Bit(s)	R/W	Default	Description
7:6	R	0	transport_scrambling_control
5:4	R	0	adaptation_field_control
3:0	R	0	continuity_counter

OM_CMD_STATUS_O 0x0b

Bit(s)	R/W	Default	Description
15:12	R	0	om_cmd_count (read only)
11:9	R	0	overflow_count // bit 11:9 -- om_cmd_wr_ptr (read only)
8:6	R	0	om_overwrite_count // bit 8:6 -- om_cmd_rd_ptr (read only)
5:3	R	0	type_stb_om_w_rd (read only)
2	R	0	unit_start_stb_om_w_rd (read only)
1	R	0	om_cmd_overflow (read only)

Bit(s)	R/W	Default	Description
0	R	0	om_cmd_pending (read)

OM_CMD_DATA_O 0x0c

Bit(s)	R/W	Default	Description
15:9	R	0	count_stb_om_w_rd (read only)
8:0	R	0	start_stb_om_wa_rd (read only)

OM_CMD_DATA2_O 0x0d

Bit(s)	R/W	Default	Description
11:0	R	0	offset for section data

SEC_BUFF_01_START_O 0x0e

Bit(s)	R/W	Default	Description
31:16	RW	0	base address for section buffer group 0 (*0x400 to get real address)
15:0	RW	0	base address for section buffer group 1 (*0x400 to get real address)

SEC_BUFF_23_START_O 0x0f

Bit(s)	R/W	Default	Description
31:16	RW	0	base address for section buffer group 2 (*0x400 to get real address)
15:0	RW	0	base address for section buffer group 3 (*0x400 to get real address)

SEC_BUFF_SIZE_O 0x10

Bit(s)	R/W	Default	Description
3:0	RW	0	section buffer size for group 0 (bitused, for example, 10 means 1K)
7:4	RW	0	section buffer size for group 1
11:8	RW	0	section buffer size for group 2
15:12	RW	0	section buffer size for group 3

SEC_BUFF_BUSY_O 0x11

Bit(s)	R/W	Default	Description
31:0	R	0	Section buffer busy status for buff 31:0 (Read Only)

SEC_BUFF_READY_O 0x12

Bit(s)	R/W	Default	Description
31:0	RW	0	section buffer write status for buff 31:0 -- Read clear buffer status (buff READY and BUSY) – write

SEC_BUFF_NUMBER_O 0x13

Bit(s)	R/W	Default	Description
4:0	RW	0	SEC_BUFFER_INDEX RW
12:8	RW	0	SEC_BUFFER_NUMBER for the INDEX buffer Read_Only
14	RW	0	output_section_buffer_valid
15	RW	0	section_reset_busy (Read Only)

ASSIGN_PID_NUMBER_O 0x14

Bit(s)	R/W	Default	Description
9:5	RW	0	BYPASS PID number
4:0	RW	0	PCR PID number

VIDEO_STREAM_ID_O 0x15

Bit(s)	R/W	Default	Description
31:16	RW	0	for video2
15:0	RW	0	stream_id filter Bit(s) enable

AUDIO_STREAM_ID_O 0x16

Bit(s)	R/W	Default	Description
15:0	RW	0	For audio

SUB_STREAM_ID_O 0x17

Bit(s)	R/W	Default	Description
15:0	RW	0	For sub

OTHER_STREAM_ID_O 0x18

Bit(s)	R/W	Default	Description
15:0	RW	0	For other

PCR90K_CTL_O 0x19

Bit(s)	R/W	Default	Description
12	RW	0	PCR_EN
11:0	RW	0	PCR90K_DIV

PCR_DEMUX_O 0x1a

Bit(s)	R/W	Default	Description
31:0	RW	0	PCR

VIDEO_PTS_DEMUX_O 0x1b

Bit(s)	R/W	Default	Description
31:0	RW	0	VPTS

VIDEO_DTS_DEMUX_O 0x1c

Bit(s)	R/W	Default	Description
31:0	RW	0	VDTS

AUDIO_PTS_DEMUX_O 0x1d

Bit(s)	R/W	Default	Description
31:0	RW	0	APTS

SUB_PTS_DEMUX_O 0x1e

Bit(s)	R/W	Default	Description
31:0	RW	0	SPTS

STB_PTS_DTS_STATUS_O 0x1f

Bit(s)	R/W	Default	Description
15	R	0	SUB_PTS[32]
14	R	0	AUDIO_PTS[32]
13	R	0	VIDEO_DTS[32]
12	R	0	VIDEO_PTS[32]
3	R	0	sub_pts_ready
2	R	0	audio_pts_ready
1	R	0	video_dts_ready
0	R	0	video_pts_ready

STB_DEBUG_INDEX_O 0x20

Bit(s)	R/W	Default	Description
3	RW	0	pes_ctr_byte[7:0], pes_flag_byte[7:0]
2	RW	0	pes_package_bytes_left[15:0]
1	RW	0	stream_id[7:0], pes_header_bytes_left[7:0]
0	RW	0	adaptation_field_length[7:0], adaption_field_byte_1[7:0]

STB_DEBUG_DATAOUT_O 0x21

Bit(s)	R/W	Default	Description
15:0	R	0	Debug data out[15:0]

STB_MOM_CTL_O 0x22

Bit(s)	R/W	Default	Description
31	RW	0	no_match_record_en
30:16	RW	0	reserved
15:9	RW	0	MAX OM DMA COUNT (default: 0x40)
8:0	RW	0	LAST ADDR OF OM ADDR (default: 127)

STB_INT_STATUS_O 0x23

Bit(s)	R/W	Default	Description
12	R	0	INPUT_TIME_OUT
11	R	0	PCR_ready
10	R	0	audio_splicing_point
9	R	0	video_splicing_point
8	R	0	other_PES_int
7	R	0	sub_PES_int
6	R	0	discontinuity
5	R	0	duplicated_pack_found
4	R	0	New PDTS ready
3	R	0	om_cmd_buffer ready for access
2	R	0	section buffer ready
1	R	0	transport_error_indicator
0	R	0	TS ERROR PIN

DEMUX_ENDIAN_O 0x24

Bit(s)	R/W	Default	Description
23:21	RW	0	demux om write endian control for OTHER_PES_PACKET
20:18	RW	0	demux om write endian control for SCR_ONLY_PACKET
17:15	RW	0	demux om write endian control for SUB_PACKET
14:12	RW	0	demux om write endian control for AUDIO_PACKET

11:9	RW	0	demux om write endian control for VIDEO_PACKET
8:6	RW	0	demux om write endian control for else
5:3	RW	0	demux om write endian control for bypass
2:0	RW	0	demux om write endian control for section

TS_HIU_CTL_O 0x25

Bit(s)	R/W	Default	Description
15:8	RW	0	last_burst_threshold
7	RW	0	use hi_bsf interface
6:2	RW	0	fec_clk_div
1	RW	0	ts_source_sel
0	RW	0	Hiu TS generate enable

SEC_BUFF_BASE_O 0x26

Bit(s)	R/W	Default	Description
15:0	RW	0	base address for section buffer start (*0x10000 to get real base)

DEMUX_MEM_REQ_EN_O 0x27

Bit(s)	R/W	Default	Description
11	RW	0	mask bit for OTHER_PES_AHB_DMA_EN
10	RW	0	mask bit for SUB_AHB_DMA_EN
9	RW	0	mask bit for BYPASS_AHB_DMA_EN
8	RW	0	mask bit for SECTION_AHB_DMA_EN
7	RW	0	mask bit for recoder stream
6:0	RW	0	mask bit for each type

VIDEO_PDTS_WR_PTR_O 0x28

Bit(s)	R/W	Default	Description
31:0	RW	0	vb_wr_ptr for video PDTS

AUDIO_PDTS_WR_PTR_O 0x29

Bit(s)	R/W	Default	Description
31:0	RW	0	ab_wr_ptr for video PDTS

SUB_WR_PTR_O 0x2a

Bit(s)	R/W	Default	Description
20:0	RW	0	SB_WRITE_PTR (sb_wr_ptr << 3 == byte write position)

SB_START_O 0x2b

Bit(s)	R/W	Default	Description
19:0	RW	0	SB_START (sb_start << 12 == byte address);

SB_LAST_ADDR_O 0x2c

Bit(s)	R/W	Default	Description
20:0	RW	0	SB_SIZE (sb_size << 3 == byte size, 16M maximun)

SB_PES_WR_PTR_O 0x2d

Bit(s)	R/W	Default	Description
31:0	RW	0	sb_wr_ptr for sub PES

OTHER_WR_PTR_O 0x2e

Bit(s)	R/W	Default	Description
31:21	RW	0	ob_wr_ptr for other PES
20:0	RW	0	OB_WRITE_PTR (ob_wr_ptr << 3 == byte write position)

OB_START_O 0x2f

Bit(s)	R/W	Default	Description
19:0	RW	0	OB_START (ob_start << 12 == byte address);

OB_LAST_ADDR_O 0x30

Bit(s)	R/W	Default	Description
20:0	RW	0	OB_SIZE (ob_size << 3 == byte size, 16M maximun)

OB_PES_WR_PTR_O 0x31

Bit(s)	R/W	Default	Description
31:0	RW	0	ob_wr_ptr for sub PES

STB_INT_MASK_O 0x32

Bit(s)	R/W	Default	Description
9	RW	0	splicing_point
8	RW	0	other_PES_int

Bit(s)	R/W	Default	Description
7	RW	0	sub_PES_int
6	RW	0	discontinuity
5	RW	0	duplicated_pack_found
4	RW	0	New PDTS ready
3	RW	0	om_cmd_buffer ready for access
2	RW	0	section buffer ready
1	RW	0	transport_error_indicator
0	RW	0	TS ERROR PIN

VIDEO_SPLICING_CTL_O 0x33

Bit(s)	R/W	Default	Description
15	RW	0	splicing VIDEO PID change enable
14:10	RW	0	VIDEO PID FILTER ADDRESS
9	RW	0	PES splicing active (Read Only)
8	RW	0	splicing active (Read Only)
7:0	RW	0	splicing countdown (Read Only)

AUDIO_SPLICING_CTL_O 0x34

Bit(s)	R/W	Default	Description
15	RW	0	splicing AUDIO PID change enable
14:10	RW	0	AUDIO PID FILTER ADDRESS
9	RW	0	PES splicing active (Read Only)
8	RW	0	splicing active (Read Only)
7:0	RW	0	splicing countdown (Read Only)

TS_PACKAGE_BYTE_COUNT_O 0x35

Bit(s)	R/W	Default	Description
23:16	RW	0	M2TS_SKIP_BYTES
15:8	RW	0	LAST TS PACKAGE BYTE COUNT (Read Only)
7:0	RW	0	PACKAGE BYTE COUNT (Read Only)

PES_STRONG_SYNC_O 0x36

Bit(s)	R/W	Default	Description

15:0	RW	0	2 bytes strong sync add to PES

OM_DATA_RD_ADDR_O 0x37

Bit(s)	R/W	Default	Description
15	RW	0	stb_om_ren
14:11	RW	0	reserved
10:0	RW	0	OM_DATA_RD_ADDR

OM_DATA_RD_O 0x38

Bit(s)	R/W	Default	Description
15:0	RW	0	OM_DATA_RD

SECTION_AUTO_STOP_3_O 0x39

Bit(s)	R/W	Default	Description
31	RW	0	Auto stop count 31 Wr_en
30:28	RW	0	Auto stop count 31
27	RW	0	Auto stop count 30 Wr_en
26:24	RW	0	Auto stop count 30
23	RW	0	Auto stop count 29 Wr_en
22:20	RW	0	Auto stop count 29
19	RW	0	Auto stop count 28 Wr_en
18:16	RW	0	Auto stop count 28
15	RW	0	Auto stop count 27 Wr_en
14:12	RW	0	Auto stop count 27
11	RW	0	Auto stop count 26 Wr_en
10:8	RW	0	Auto stop count 26
7	RW	0	Auto stop count 25 Wr_en
6:4	RW	0	Auto stop count 25
3	RW	0	Auto stop count 24 Wr_en
2:0	RW	0	Auto stop count 24

SECTION_AUTO_STOP_2_O 0x3a

Bit(s)	R/W	Default	Description

31	RW	0	Auto stop count 23 Wr_en
30:28	RW	0	Auto stop count 23
27	RW	0	Auto stop count 22 Wr_en
26:24	RW	0	Auto stop count 22
23	RW	0	Auto stop count 21 Wr_en
22:20	RW	0	Auto stop count 21
19	RW	0	Auto stop count 20 Wr_en
18:16	RW	0	Auto stop count 20
15	RW	0	Auto stop count 19 Wr_en
14:12	RW	0	Auto stop count 19
11	RW	0	Auto stop count 18 Wr_en
10:8	RW	0	Auto stop count 18
7	RW	0	Auto stop count 17 Wr_en
6:4	RW	0	Auto stop count 17
3	RW	0	Auto stop count 16 Wr_en
2:0	RW	0	Auto stop count 16

SECTION_AUTO_STOP_1_O 0x3b

Bit(s)	R/W	Default	Description
31	RW	0	Auto stop count 15 Wr_en
30:28	RW	0	Auto stop count 15
27	RW	0	Auto stop count 14 Wr_en
26:24	RW	0	Auto stop count 14
23	RW	0	Auto stop count 13 Wr_en
22:20	RW	0	Auto stop count 13
19	RW	0	Auto stop count 12 Wr_en
18:16	RW	0	Auto stop count 12
15	RW	0	Auto stop count 11 Wr_en
14:12	RW	0	Auto stop count 11
11	RW	0	Auto stop count 10 Wr_en
10:8	RW	0	Auto stop count 10

Bit(s)	R/W	Default	Description
7	RW	0	Auto stop count 9 Wr_en
6:4	RW	0	Auto stop count 9
3	RW	0	Auto stop count 8 Wr_en
2:0	RW	0	Auto stop count 8

SECTION_AUTO_STOP_0_O 0x3c

Bit(s)	R/W	Default	Description
31	RW	0	Auto stop count 7 Wr_en
30:28	RW	0	Auto stop count 7
27	RW	0	Auto stop count 6 Wr_en
26:24	RW	0	Auto stop count 6
23	RW	0	Auto stop count 5 Wr_en
22:20	RW	0	Auto stop count 5
19	RW	0	Auto stop count 4 Wr_en
18:16	RW	0	Auto stop count 4
15	RW	0	Auto stop count 3 Wr_en
14:12	RW	0	Auto stop count 3
11	RW	0	Auto stop count 2 Wr_en
10:8	RW	0	Auto stop count 2
7	RW	0	Auto stop count 1 Wr_en
6:4	RW	0	Auto stop count 1
3	RW	0	Auto stop count 0 Wr_en
2:0	RW	0	Auto stop count 0

DEMUX_CHANNEL_RESET_O 0x3d

Bit(s)	R/W	Default	Description
31:0	R	0	Bit 31:0 reset channel status - Each Bit reset each channel

DEMUX_SCRAMBLING_STATE_O 0x3e

Bit(s)	R/W	Default	Description
31:0	R	0	Scrambling state of each channel

DEMUX_CHANNEL_ACTIVITY_O 0x3f

Bit(s)	R/W	Default	Description
31:0	R	0	Channel activity of each channel

DEMUX_STAMP_CTL_O 0x40

Bit(s)	R/W	Default	Description
4	RW	0	video_stamp_use_dts
3	RW	0	audio_stamp_sync_1_en
2	RW	0	audio_stamp_insert_en
1	RW	0	video_stamp_sync_1_en
0	RW	0	video_stamp_insert_en

DEMUX_VIDEO_STAMP_SYNC_0_O 0x41

Bit(s)	R/W	Default	Description
31:0	RW	0	Video stamp sync [63:32]

DEMUX_VIDEO_STAMP_SYNC_1_O 0x42

Bit(s)	R/W	Default	Description
31:0	RW	0	Video stamp sync [31:0]

DEMUX_AUDIO_STAMP_SYNC_0_O 0x43

Bit(s)	R/W	Default	Description
31:0	RW	0	Aideo stamp sync [63:32]

DEMUX_AUDIO_STAMP_SYNC_1_O 0x44

Bit(s)	R/W	Default	Description
31:0	RW	0	Aideo stamp sync [31:0]

DEMUX_SECTION_RESET_O 0x45

Bit(s)	R/W	Default	Description
31:0	R	0	Write : Bit[4:0] secter filter number for reset Read : select according to output_section_buffer_valid: per bit per section buffer valid status or section_buffer_ignore

DEMUX_INPUT_TIMEOUT_C_O 0x46

Bit(s)	R/W	Default	Description

31:0	RW	0	channel_reset_timeout_disable

DEMUX_INPUT_TIMEOUT_O 0x47

Bit(s)	R/W	Default	Description
31	RW	0	no_match_reset_timeout_disable
30:0	RW	0	input_time_out_int_cnt (0 -- means disable) Wr-setting, Rd-count

DEMUX_PACKET_COUNT_O 0x48

Bit(s)	R/W	Default	Description
31:0	RW	0	channel_packet_count_disable

DEMUX_PACKET_COUNT_C_O 0x49

Bit(s)	R/W	Default	Description
31	RW	0	no_match_packet_count_disable
30:0	RW	0	input_packet_count

DEMUX_CHAN_RECORD_EN_O 0x4a

Bit(s)	R/W	Default	Description
31:0	RW	0xffffffff	channel_record_enable

DEMUX_CHAN_PROCESS_EN_O 0x4b

Bit(s)	R/W	Default	Description
31:0	RW	0xffffffff	channel_process_enable

DEMUX_SMALL_SEC_CTL_O 0x4c

Bit(s)	R/W	Default	Description
31:24	RW	0	small_sec_size ((n+1) * 256 Bytes)
23:16	RW	0	small_sec_rd_ptr
15:8	RW	0	small_sec_wr_ptr
7:2	RW	0	reserved
1	RW	0	small_sec_wr_ptr_wr_enable
0	RW	0	small_section_enable

10.2 Video Output

10.2.1 Overview

This section describes S922X's VPU sub-module, including RDMA sub-module, VIU sub-module, HDMITX sub-module, CVBS sub-module and MIPI_DSI sub-module.

10.2.2 VPU

VPU is display process unit, the main function is to receive data from decoder/ddr/hdmirx etc, then process the source data in order to get the high-quality video picture, and finally send out the video to the screen by HDMITX/CVBS/MIPI etc.

10.2.3 Register Description

10.2.3.1 VPU Registers

VPU_VDIN_PRE_ARB_CTRL 0x2714

Bit(s)	R/W	Default	Description
31	R/W	0	VDIN_PREARB_SOFT_RESET: 1=SW reset vdin_mmc_pre_arb.
30-18	R	0	Reserved.
17	R/W	1	VDIN_ACG_EN: 1=Enable auto-clock gating of vdin_mmc_pre_arb.
16	R/W	0	VDIN_DISABLE_CLK: 1=Disalbe the clock to vdin_mmc_pre_arb.
15-0	R/W	0xffff	VDIN_REQ_EN: vdin_mmc_pre_arb request enable.

VPU_VDISP_PRE_ARB_CTRL 0x2715

Bit(s)	R/W	Default	Description
31	R/W	0	VDISP_PREARB_SOFT_RESET: 1=SW reset vdisp_mmc_pre_arb.
30-18	R	0	Reserved.
17	R/W	1	VDISP_ACG_EN: 1=Enable auto-clock gating of vdisp_mmc_pre_arb.
16	R/W	0	VDISP_DISABLE_CLK: 1=Disalbe the clock to vdisp_mmc_pre_arb.
15-0	R/W	0xffff	VDISP_REQ_EN: vdisp_mmc_pre_arb request enable.

VPU_VPUARB2_PRE_ARB_CTRL 0x2716

Bit(s)	R/W	Default	Description
31	R/W	0	VPUARB2_PREARB_SOFT_RESET: 1=SW reset vpuarb2_mmc_pre_arb.
30-18	R	0	Reserved.
17	R/W	1	VPUARB2_ACG_EN: 1=Enable auto-clock gating of vpuarb2_mmc_pre_arb.
16	R/W	0	VPUARB2_DISABLE_CLK: 1=Disalbe the clock to vpuarb2_mmc_pre_arb.
15-0	R/W	0xffff	VPUARB2_REQ_EN: vpuarb2_mmc_pre_arb request enable.

VPU_VIU_VENC_MUX_CTRL 0x271a

Bit(s)	R/W	Default	Description
17-16	W	0	RASP DPI CLOCK SEL : 00 : cph1 01 : cph2 10/11 : cph3
11-8	R/W	0	VIU_VDIN_SEL_DATA: Select which data to VDI6 path, must clear it first before switching the data. 4'b0000: Disable VIU to VDI6 path 4'b0001: Select ENCI data to VDI6 4'b0010: Select ENCP data to VDI6 4'b0100: Select ENCT data to VDI6 4'b1000: Select ENCL data to VDI6
7-4	R/W	0	VIU_VDIN_SEL_CLK: Select which clock to VDI6 path, must clear it first before switching the clock. 4'b0000: Disable VIU to VDI6 clock 4'b0001: Select ENCI clock to VDI6 4'b0010: Select ENCP clock to VDI6 4'b0100: Select ENCT clock to VDI6 4'b1000: Select ENCL clock to VDI6
3-2	R/W	0	VIU2_SEL_VENC: Select which one of the encl/P/T/L that Viu2 connects to. 0: ENCL 1: ENCI 2: ENCP 3: ENCT
1-0	R/W	0	VIU1_SEL_VENC: Select which one of the encl/P/T/L that Viu1 connects to. 0: ENCL 1: ENCI 2: ENCP 3: ENCT

VPU_HDMI_SETTING 0x271b

Bit(s)	R/W	Default	Description
15-12	R/W	0	RD_RATE: Read rate to the async FIFO between VENC and HDMI. 0: One read every rd_clk 1: One read every 2 rd_clk 2: One read every 3 rd_clk … 15: One read every 16 rd_clk
11-8	R/W	0	WR_RATE: Write rate to the async FIFO between VENC and HDMI.

Bit(s)	R/W	Default	Description
			0: One write every wr_clk 1: One write every 2 wr_clk 2: One write every 3 wr_clk … 15: One write every 16 wr_clk
7-5	R/W	0	DATA_COMP_MAP: Input data is CrYCr(BRG), map the output data to desired format: 0: output CrYCb (BRG) 1: output YCbCr (RGB) 2: output YCrCb (RBG) 3: output CbCrY (GBR) 4: output CbYCr (GRB) 5: output CrCbY (BGR) 6,7: Reserved
4	R/W	0	INV_DVI_CLK: If true, invert the polarity of clock output to external DVI interface. (NOT internal HDMI).
3	R/W	0	INV_VSYNC: If true, invert the polarity of VSYNC input from VENC
2	R/W	0	INV_HSYNC: If true, invert the polarity of HSYNC input from VENC
1-0	R/W	0	SRC_SEL: Select which HDMI source from between ENCI and ENCP. 2'b00: Disable HDMI source 2'b01: Select ENCI data to HDMI 2'b10: Select ENCP data to HDMI

ENCI_INFO_READ 0x271c

Bit(s)	R/W	Default	Description
31-29	R	0	Current ENCI field status.
28-25	R	0	Reserved
24-16	R	0	Current ENCI line counter status.
15-11	R	0	Reserved
10-0	R	0	Current ENCI pixel counter status.

ENCP_INFO_READ 0x271d

Bit(s)	R/W	Default	Description
31-29	R	0	Current ENCP field status.
28-16	R	0	Current ENCP line counter status.
15-13	R	0	Reserved
12-0	R	0	Current ENCP pixel counter status.

ENCT_INFO_READ 0x271e

Bit(s)	R/W	Default	Description
31-29	R	0	Current ENCT field status.
28-16	R	0	Current ENCT line counter status.
15-13	R	0	Reserved
12-0	R	0	Current ENCT pixel counter status.

ENCL_INFO_READ 0x271f

Bit(s)	R/W	Default	Description
31-29	R	0	Current ENCL field status.
28-16	R	0	Current ENCL line counter status.
15-13	R	0	Reserved
12-0	R	0	Current ENCL pixel counter status.

VPU_SW_RESET 0x2720

Bit(s)	R/W	Default	Description
3	R/W	0	vpuarb2_mmc_arb_rst_n
2	R/W	0	vdisp_mmc_arb_rst_n
1	R/W	0	vdin_mmc_arb_rst_n
0	R/W	0	viu_rst_n

VPU_CLK_GATE 0x2723

Bit(s)	R/W	Default	Description
30-18	R/W	0x0	Reserved
17	R/W	0x1	Clk_b control register no latch
16	R/W	0x1	Clk_vib enable
15	R/W	0x1	Gvapbclk enable
14	R/W	0x1	Clk mpeg vlock enable
13	R/W	0x1	Reserved
12	R/W	0x1	Venc_dac_process_clk enable
11	R/W	0x1	Venc_i_top enable
10	R/W	0x1	Venci_int enable

Bit(s)	R/W	Default	Description
9:8	R/W	0x11	Clk_vib latch sync source select 00: di2ldim_go_field, 01: post_frame_rst;10: pre_frame_rst, 11:viu_go_field
7	R/W	0x1	Reserved
6	R/W	0x1	Vpu_misc_clk enable
5	R/W	0x1	Venc_l_top enable
4	R/W	0x1	Venc_l_int enable
3	R/W	0x1	Venc_p_int enable
2	R/W	0x1	Reserved
1	R/W	0x1	Vi_top clock enable
0	R/W	0x1	Venc_p_top enable

VPU_HDMI_DATA_OVR 0x2727

Bit(s)	R/W	Default	Description
31	R/W	0	DATA_OVR_EN: Control if override HDMI input data with DATA_OVR[29:0], for display e.g. black or blue screen. 0: No override 1: Enable override
30	R	0	Reserved
29-0	R/W	0	DATA_OVR: programmable pixel data value for override.

VPU_VPU_PWM_V0 0x2730

Bit(s)	R/W	Default	Description
31	R/W	0	reg_vpu_pwm_inv, 1: invert the pwm signal, active low
30-29	R/W	0	reg_vpu_pwm_src_sel, 00: encl, enct, encp
28-16	R/W	0	reg_vpu_pwm_v_end0
15-14	R/W	0	reg_vpu_pwm_setting_latch_mode
13	R/W	1	reg_vpu_pwm_vs_inv
12-0	R/W	0	reg_vpu_pwm_v_start0

VPU_VPU_PWM_V1 0x2731

Bit(s)	R/W	Default	Description
28-16	R/W	0	reg_vpu_pwm_v_end1
12-0	R/W	0	reg_vpu_pwm_v_start1

VPU_VPU_PWM_V2 0x2732

Bit(s)	R/W	Default	Description
28-16	R/W	0	reg_vpu_pwm_v_end2
12-0	R/W	0	reg_vpu_pwm_v_start2

VPU_VPU_PWM_V3 0x2733

Bit(s)	R/W	Default	Description
28-16	R/W	0	reg_vpu_pwm_v_end3
12-0	R/W	0	reg_vpu_pwm_v_start3

VPU_VPU_PWM_H0 0x2734

Bit(s)	R/W	Default	Description
28-16	R/W	0	reg_vpu_pwm_h_end0
12-0	R/W	0	reg_vpu_pwm_h_start0

VPU_VPU_PWM_H1 0x2735

Bit(s)	R/W	Default	Description
28-16	R/W	0	reg_vpu_pwm_h_end1
12-0	R/W	0	reg_vpu_pwm_h_start1

VPU_VPU_PWM_H2 0x2736

Bit(s)	R/W	Default	Description
28-16	R/W	0	reg_vpu_pwm_h_end2
12-0	R/W	0	reg_vpu_pwm_h_start2

VPU_VPU_PWM_H3 0x2737

Bit(s)	R/W	Default	Description
28-16	R/W	0	reg_vpu_pwm_h_end3
12-0	R/W	0	reg_vpu_pwm_h_start3

VPU_VPU_3D_SYNC1 0x2738

Bit(s)	R/W	Default	Description
31	R/W	0	reg_3dsync_enable, 1: enable 3d sync output
30	R/W	0	3dsync setting vsync latch
29	R/W	0	3dsync go high field polarity: 1, go high while field[0]=1
28-16	R/W	0	reg_3dsync_v_end0

15	R/W	0	3dsync out inv
14	R/W	0	3dsync vbo out inv
13	R/W	1	Vbo 3d en, to v by one, 3d enable
12-0	R/W	0	reg_3dsync_v_start0

VPU_VPU_3D_SYNC2 0x2739

Bit(s)	R/W	Default	Description
31	R/W	0	Reg_3dsync_field_bit_sel: 1. Keep 3dsync not changed for two fields, i.e. L-L-R-R-L-L-R-R (11001100) 0. Change 3dsync every field i.e. L-R-L-R (1010)
28-16	R/W	0	reg_3dsync_h_end
12-0	R/W	0	reg_3dsync_h_start

VPU_HDMI_FMT_CTRL 0x2743

Bit(s)	R/W	Default	Description
12 bit to 10 bit dither control register. 10 bit to 8 bit see VPU_HDMI_DITH_CNTL			
21-19	R/W	0	rame count offset for B
18-16	R/W	0	frame count offset for G
15	R/W	0	hcnt hold when de valid
14	R/W	0	RGB frame count seperate
13	R/W	0	dith4x4 : frame random enable
12	R/W	0	dith4x4 enable
11	R/W	0	tunnel enable for DOLBY
10	R/W	0	rounding enable
9-6	R/W	0	Cntl_hdmi_dith10 :
5	R/W	0	Cntl_hdmi_dith_md:
4	R/W	0	Cntl_hdmi_dith_en: dither 10-b to 8-b enable
3-2	R/W	0	Cntl_chroma_dnsmp: Chroma down sample mode when convert to 422 or 420. 0 = use pixel 0; 1 = use pixel 1; 2 = use average;
1-0	R/W	0	Cntl_hdmi_vid_fmt: Control whether to convert ENCP's 444 data to 422 or 420 0 = No convertion; 1 = Convert to 422; 2 = Convert to 420;

PU_VDIN_ASYNC_HOLD_CTRL 0x2744

Bit(s)	R/W	Default	Description
31-24	R/W	‘h18	Wr_hold_num
23-16	R/W	‘h10	Wr_rel_num
15-8	R/W	‘h18	Rd_hold_num
7-0	R/W	‘h10	Rd_rel_num

VPU_VDISP_ASYNC_HOLD_CTRL 0x2745

Bit(s)	R/W	Default	Description
31-24	R/W	‘h18	Wr_hold_num
23-16	R/W	‘h10	Wr_rel_num
15-8	R/W	‘h18	Rd_hold_num
7-0	R/W	‘h10	Rd_rel_num

VPU_VPUARB2_ASYNC_HOLD_CTRL 0x2746

Bit(s)	R/W	Default	Description
31-24	R/W	‘h18	Wr_hold_num
23-16	R/W	‘h10	Wr_rel_num
15-8	R/W	‘h18	Rd_hold_num
7-0	R/W	‘h10	Rd_rel_num

VPU_ARB_URG_CTRL 0x2747

Bit(s)	R/W	Default	Description
11	R/W	0	Rdma_ddr_reg_busy to vpuarb2_urg_ctrl
10	R/W	0	Rdma_ddr_reg_busy to vdisp_urg_ctrl
9	R/W	0	Rdma_ddr_reg_busy to vdin_urg_ctrl
8	R/W	0	Vdin1_lff_urg_ctrl to vpuarb2_urg_ctrl
7	R/W	0	Vdin0_lff_urg_ctrl to vpuarb2_urg_ctrl
6	R/W	0	Vpp_off_urg_ctrl to vpuarb2_urg_ctrl
5	R/W	0	Vdin1_lff_urg_ctrl to vdisp_urg_ctrl
4	R/W	0	Vdin0_lff_urg_ctrl to vdisp_urg_ctrl
3	R/W	0	Vpp_off_urg_ctrl to vdisp_urg_ctrl
2	R/W	0	Vdin1_lff_urg_ctrl to vdin_urg_ctrl

1	R/W	0	Vdin0_lff_urg_ctrl to vdin_urg_ctrl
0	R/W	0	Vpp_off_urg_ctrl to vdin_urg_ctrl

VPU_VENCL_DITH_EN 0x2749

R/W	Default	Description
R/W	0	dith_en //dith bettwen vpp and encl_int
R/W	0	dith hsize

VPU_422TO444_RST 0x274a

Bit(s)	R/W	Default	Description
3	R/W	0	Change output of viu_12bit422to10bit444_vd1 to 10bits 1:cut high 2 bits 0:cut low 2 bits
2	R/W	0	Change output of viu_12bit422to10bit444_encp to 10bits 1:cut high 2 bits 0:cut low 2 bits
1	R/W	0	Soft rst enable of viu_12bit422to10bit444_vd1 module active high
0	R/W	0	Soft rst enable of viu_12bit422to10bit444_encp module active high

VPU_422TO444_CTRL0 0x274b

Bit(s)	R/W	Default	Description
31	R/W	0	bypass_mode : active high if this bit set high ,viu_12bit422to10bit444_encp work in bypass mode, dout = din
30	/	/	reversed
29	R/W	0	clip10_mode: active high if this bit set high ,viu_12bit422to10bit444_encp work clip10_mode ,output high 10bits value of din and clip data to 10bit,then expand to 12bits according clip_lend.
28	R/W	0	Clip8_mode: active high if this bit set high ,viu_12bit422to10bit444_encp work clip8_mode ,output high 8bits value of din and clip data to 8bit,then expand to 12bits according clip_lend.
27	R/W	0	clip_lend :active high make output data to 12bits in clip10_mode/clip8_mode/scramble 1;expand bits in high bits 0: ;expand bits low bits
26	R/W	0	scramble_mode :active high viu_12bit422to10bit444_encp module get luma and chroma from 422 format data erery pixel ,then reorginize data to 8bits according to reg_tunnel value(this model can change data from 422 to 444??).

Bit(s)	R/W	Default	Description
			Luma = din[35:24] Chroma =odd_pixel?din[23:12]:din[11:0]
25	R/W	0	go_field_en 1:rst odd_pixel to 0 when go_filed come
24	R/W	0	go_line_en 1:rst rst odd_pixel to 0 when go_line come
23	R/W	0	oft_rst_en 1:soft rst rst odd_pixel to 0
22	R/W	0	de_sel : input data de singal 1:chose encp require singal as de 0:close this module
17:15	R/W	0	reg_tunnel_sel_b1 : select high 4 bits for B from {luma,chroma} ,only work in scramble mode: 0:bit[3:0] 1:bit[7:3] 2:bit[11:8] 3:bit[15:12] 4:bit[19:16] 5:bit[23:20] Default:bit[23:20]
14:12	R/W	0	reg_tunnel_sel_g1 : select high 4 bits for G from {luma,chroma} ,only work in scramble mode: 0:bit[3:0] 1:bit[7:3] 2:bit[11:8] 3:bit[15:12] 4:bit[19:16] 5:bit[23:20] Default:bit[23:20]
11:9	R/W	0	reg_tunnel_sel_r1 : select high 4 bits for R from {luma,chroma} ,only work in scramble mode: 0:bit[3:0] 1:bit[7:3] 2:bit[11:8] 3:bit[15:12] 4:bit[19:16] 5:bit[23:20] Default:bit[23:20]
8:6	R/W	0	reg_tunnel_sel_B0 : select low 4 bits for B from {luma,chroma} ,only work in scramble mode: 0:bit[3:0] 1:bit[7:3] 2:bit[11:8]

Bit(s)	R/W	Default	Description
			3:bit[15:12] 4:bit[19:16] 5:bit[23:20] Default:bit[23:20]
5:3	R/W	0	reg_tunnel_sel_g0 : select low 4 bits for G from {luma,chroma} ,only work in scramble mode: 0:bit[3:0] 1:bit[7:3] 2:bit[11:8] 3:bit[15:12] 4:bit[19:16] 5:bit[23:20] Default:bit[23:20]
2:0	R/W	0	reg_tunnel_sel_r0 : select low 4 bits for R from {luma,chroma} ,only work in scramble mode: 0:bit[3:0] 1:bit[7:3] 2:bit[11:8] 3:bit[15:12] 4:bit[19:16] 5:bit[23:20] Default:bit[23:20]

VPU_422TO444_CTRL1 0x274c

Same as VPU_422TO444_CTRL0,viu_12bit422to10bit444_vd1

VPU_VIU2VDIN_HDN_CTRL 0x2780

Bit(s)	R/W	Default	Description
20	R/W	0	software reset
19-18	R/W	0	reg_viu2vdin_dn_ratio: down-scale ratio: 0->no scale; 1-> 1/2; 2->1/4; 3->reserved
17-16	R/W	0	reg_viu2vdin_flt_mode: filter mode; 0->no filter; 1->[0 2 2 0]/4; 2->[1 1 1 1]/4; 3->[1 3 3 1]/8
15-14	R/W	0	reserved
13-0	R/W	0	reg_viu2vdin_hsize: source horizontal size

VPU_RDARB_MODE_L1C1 0x2790

Bit(s)	R/W	Default	Description
21:16	R/W	0	rdarb_sel : uns, default = 0 , rdarb_sel [0]==0 slave dc0 connect master port0 rdarb_sel[0]==1 slave dc0 connect master port1 rdarb_sel [1]==0 slave dc1 connect master port0 rdarb_sel[1]==1 slave dc1 connect master port1 rdarb_sel [2]==0 slave dc2 connect master port0 rdarb_sel[2]==1 slave dc2 connect master port1 rdarb_sel [3]==0 slave dc3 connect master port0 rdarb_sel[3]==1 slave

Bit(s)	R/W	Default	Description
			dc3 connect master port1 rdarb_sel [4]==0 slave dc4 connect master port0 rdarb_sel[4]==1 slave dc4 connect master port1 rdarb_sel [5]==0 slave dc5 connect master port0 rdarb_sel[5]==1 slave dc5 connect master port1
9:8	R/W	0	rdarb_arb_mode : uns, default = 0 , rdarb_arb_mode [0] master port0 arb way, rdarb_arb_mode [1] master port1 arb way,
3:0	R/W	0	rdarb_gate_clk_ctrl : uns, default = 0 , rdarb_gate_clk_ctrl [1:0] master port0 clk gate control rdarb_gate_clk_ctrl [3:2] master port1 clk gate control

VPU_RDARB_REQEN_SLV_L1C1 0x2791

Bit(s)	R/W	Default	Description
11:0	R/W	0xfff	rdarb_dc_req_en : unsigned , default = 12'hfff rdarb_dc_req_en [0]: the slv0 req to mst port0 enable, rdarb_dc_req_en [1]: the slv1 req to mst port0 enable, rdarb_dc_req_en [2]: the slv2 req to mst port0 enable, rdarb_dc_req_en [3]: the slv3 req to mst port0 enable, rdarb_dc_req_en [4]: the slv4 req to mst port0 enable, rdarb_dc_req_en [5]: the slv5 req to mst port0 enable, rdarb_dc_req_en [6]: the slv0 req to mst port1 enable, rdarb_dc_req_en [7]: the slv1 req to mst port1 enable, rdarb_dc_req_en [8]: the slv2 req to mst port1 enable, rdarb_dc_req_en [9]: the slv3 req to mst port1 enable, rdarb_dc_req_en [10]: the slv4 req to mst port1 enable, rdarb_dc_req_en [11]: the slv5 req to mst port1 enable,

VPU_RDARB_WEIGH0_SLV_L1C1 0x2792

Bit(s)	R/W	Default	Description
29:0	R/W	0	rddc_weigh_sxn : unsigned , default = 0 rddc_weigh_sxn [0*6+:6]: the slv0 req weigh number rddc_weigh_sxn [1*6+:6]: the slv1 req weigh number rddc_weigh_sxn [2*6+:6]: the slv2 req weigh number rddc_weigh_sxn [3*6+:6]: the slv3 req weigh number rddc_weigh_sxn [4*6+:6]: the slv4 req weigh number

VPU_RDARB_WEIGH1_SLV_L1C1 0x2793

Bit(s)	R/W	Default	Description
5:0	R/W	0	rddc_weigh_sxn : unsigned , default = 0 rddc_weigh_sxn [5*6+:6]: the slv5 req weigh number

VPU_WRARB_MODE_L1C1 0x2794

Bit(s)	R/W	Default	Description

21:16	R/W	0	wrarb_sel : uns, default = 0 , wrarb_sel [0]==0 slave dc0 connect master port0 wrarb_sel[0]==1 slave dc0 connect master port1 wrarb_sel [1]==0 slave dc1 connect master port0 wrarb_sel[1]==1 slave dc1 connect master port1 wrarb_sel [2]==0 slave dc2 connect master port0 wrarb_sel[2]==1 slave dc2 connect master port1 wrarb_sel [3]==0 slave dc3 connect master port0 wrarb_sel[3]==1 slave dc3 connect master port1 wrarb_sel [4]==0 slave dc4 connect master port0 wrarb_sel[4]==1 slave dc4 connect master port1 wrarb_sel [5]==0 slave dc5 connect master port0 wrarb_sel[5]==1 slave dc5 connect master port1
9:8	R/W	0	wrarb_arb_mode : uns, default = 0 , wrarb_arb_mode [0] master port0 arb way, wrarb_arb_mode [1] master port1 arb way,
3:0	R/W	0	wrarb_gate_clk_ctrl : uns, default = 0 , wrarb_gate_clk_ctrl [1:0] master port0 clk gate control wrarb_gate_clk_ctrl [3:2] master port1 clk gate control

VPU_WRARB_REQEN_SLV_L1C1 0x2795

Bit(s)	R/W	Default	Description
11:0	R/W	0	wrarb_dc_req_en : unsigned , default = 0 wrarb_dc_req_en [0]: the slv0 req to mst port0 enable, wrarb_dc_req_en [1]: the slv1 req to mst port0 enable, wrarb_dc_req_en [2]: the slv2 req to mst port0 enable, wrarb_dc_req_en [3]: the slv3 req to mst port0 enable, wrarb_dc_req_en [4]: the slv4 req to mst port0 enable, wrarb_dc_req_en [5]: the slv5 req to mst port0 enable, wrarb_dc_req_en [0]: the slv0 req to mst port1 enable, wrarb_dc_req_en [1]: the slv1 req to mst port1 enable, wrarb_dc_req_en [2]: the slv2 req to mst port1 enable, wrarb_dc_req_en [3]: the slv3 req to mst port1 enable, wrarb_dc_req_en [4]: the slv4 req to mst port1 enable, wrarb_dc_req_en [5]: the slv5 req to mst port1 enable,

VPU_WRARB_WEIGH0_SLV_L1C1 0x2796

Bit(s)	R/W	Default	Description
29:0	R/W	0	wrdc_weigh_sxn : unsigned , default = 0 wrdc_weigh_sxn [0*6+:6]: the slv0 req weigh number wrdc_weigh_sxn [1*6+:6]: the slv1 req weigh number wrdc_weigh_sxn [2*6+:6]: the slv2 req weigh number wrdc_weigh_sxn [3*6+:6]: the slv3 req weigh number wrdc_weigh_sxn [4*6+:6]: the slv4 req weigh number

VPU_WRARB_WEIGH1_SLV_L1C1 0x2797

Bit(s)	R/W	Default	Description
5:0	R/W	0	wrdc_weigh_sxn : unsigned , default = 0 wrdc_weigh_sxn [5*6+:6]: the slv5 req weigh number

VPU_RDWR_ARB_STATUS_L1C1 0x2798

Bit(s)	R/W	Default	Description
3:2	R/W	0	wrarb_arb_busy : unsigned , default = 0
1:0	R/W	0	rdarb_arb_busy : unsigned , default = 0

VPU_RDARB_MODE_L1C2 0x2799

Bit(s)	R/W	Default	Description
20:16	R/W	0	rdarb_sel : uns, default = 0 , rdarb_sel [0]==0 slave dc0 connect master port0 rdarb_sel[0]==1 slave dc0 connect master port1 rdarb_sel [1]==0 slave dc1 connect master port0 rdarb_sel[1]==1 slave dc1 connect master port1 rdarb_sel [2]==0 slave dc2 connect master port0 rdarb_sel[2]==1 slave dc2 connect master port1 rdarb_sel [3]==0 slave dc3 connect master port0 rdarb_sel[3]==1 slave dc3 connect master port1 rdarb_sel [4]==0 slave dc4 connect master port0 rdarb_sel[4]==1 slave dc4 connect master port1
9:8	R/W	0	rdarb_arb_mode : uns, default = 0 , rdarb_arb_mode [0] master port0 arb way, rdarb_arb_mode [1] master port1 arb way,
3:0	R/W	0	rdarb_gate_clk_ctrl : uns, default = 0 , rdarb_gate_clk_ctrl [1:0] master port0 clk gate control rdarb_gate_clk_ctrl [3:2] master port0 clk gate control

VPU_RDARB_REQEN_SLV_L1C2 0x279a

Bit(s)	R/W	Default	Description
9:0	R/W	0	rdarb_dc_req_en : unsigned , default = 0 rdarb_dc_req_en [0]: the slv0 req to mst port0 enable, rdarb_dc_req_en [1]: the slv1 req to mst port0 enable, rdarb_dc_req_en [2]: the slv2 req to mst port0 enable, rdarb_dc_req_en [3]: the slv3 req to mst port0 enable, rdarb_dc_req_en [4]: the slv4 req to mst port0 enable, rdarb_dc_req_en [5]: the slv0 req to mst port1 enable, rdarb_dc_req_en [6]: the slv1 req to mst port1 enable, rdarb_dc_req_en [7]: the slv2 req to mst port1 enable, rdarb_dc_req_en [8]: the slv3 req to mst port1 enable, rdarb_dc_req_en [9]: the slv4 req to mst port1 enable,

VPU_RDARB_WEIGH0_SLV_L1C2 0x279b

Bit(s)	R/W	Default	Description
29:0	R/W	0	rddc_weigh_sxn : unsigned , default = 0 rddc_weigh_sxn [0*6+:6]: the slv0 req weigh number rddc_weigh_sxn [1*6+:6]: the slv1 req weigh number rddc_weigh_sxn [2*6+:6]: the slv2 req weigh number rddc_weigh_sxn [3*6+:6]: the slv3 req weigh number rddc_weigh_sxn [4*6+:6]: the slv4 req weigh number

VPU_RDWR_ARB_STATUS_L1C2 0x279c

Bit(s)	R/W	Default	Description
1:0	R/W	0	rdarb_arb_busy : unsigned , default = 0

VPU_RDARB_MODE_L2C1 0x279d

Bit(s)	R/W	Default	Description
27:16	R/W	0	rdarb_sel : uns, default = 0 , rdarb_sel [0]==0 slave dc0 connect master port0 rdarb_sel[0]==1 slave dc0 connect master port1 rdarb_sel [1]==0 slave dc1 connect master port0 rdarb_sel[1]==1 slave dc1 connect master port1 rdarb_sel [2]==0 slave dc2 connect master port0 rdarb_sel[2]==1 slave dc2 connect master port1 rdarb_sel [3]==0 slave dc3 connect master port0 rdarb_sel[3]==1 slave dc3 connect master port1 rdarb_sel [4]==0 slave dc4 connect master port0 rdarb_sel[4]==1 slave dc4 connect master port1 rdarb_sel [5]==0 slave dc5 connect master port0 rdarb_sel[5]==1 slave dc5 connect master port1
10:8	R/W	0	rdarb_arb_mode : uns, default = 0 , rdarb_arb_mode [0] master port0 arb way, rdarb_arb_mode [1] master port1 arb way,
5:0	R/W	0	rdarb_gate_clk_ctrl : uns, default = 0 , rdarb_gate_clk_ctrl [1:0] master port0 clk gate control rdarb_gate_clk_ctrl [3:2] master port1 clk gate control rdarb_gate_clk_ctrl [5:4] master port2 clk gate control

VPU_RDARB_REQEN_SLV_L2C1 0x279e

Bit(s)	R/W	Default	Description
17:0	R/W	0	rdarb_dc_req_en : unsigned , default = 0 rdarb_dc_req_en [0]: the slv0 req to mst port0 enable, rdarb_dc_req_en [1]: the slv1 req to mst port0 enable, rdarb_dc_req_en [2]: the slv2 req to mst port0 enable, rdarb_dc_req_en [3]: the slv3 req to mst port0 enable, rdarb_dc_req_en [4]: the slv4 req to mst port0 enable, rdarb_dc_req_en [5]: the slv5 req to mst port0 enable, rdarb_dc_req_en [0]: the slv0 req to mst port1 enable, rdarb_dc_req_en [1]: the slv1 req to mst port1 enable, rdarb_dc_req_en [2]: the slv2 req to mst port1 enable, rdarb_dc_req_en [3]: the slv3 req to mst port1 enable, rdarb_dc_req_en [4]: the slv4 req to mst port1 enable, rdarb_dc_req_en [5]: the slv5 req to mst port1 enable,

VPU_RDARB_WEIGH0_SLV_L2C1 0x279f

Bit(s)	R/W	Default	Description
29:0	R/W	0	rddc_weigh_sxn : unsigned , default = 0 rddc_weigh_sxn [0*6+:6]: the slv0 req weigh number rddc_weigh_sxn [1*6+:6]: the slv1 req weigh number rddc_weigh_sxn [2*6+:6]: the slv2 req weigh number rddc_weigh_sxn [3*6+:6]: the slv3 req weigh number rddc_weigh_sxn [4*6+:6]: the slv4 req weigh number

VPU_RDARB_WEIGH1_SLV_L2C1 0x27a0

Bit(s)	R/W	Default	Description
5:0	R/W	0	rddc_weigh_sxn : unsigned , default = 0 rddc_weigh_sxn [5*6+:6]: the slv5 req weigh number

VPU_RDWR_ARB_STATUS_L2C1 0x27a1

Bit(s)	R/W	Default	Description
3:2	R/W	0	wrarb_arb_busy : unsigned , default = 0
1:0	R/W	0	rdarb_arb_busy : unsigned , default = 0

VPU_WRARB_MODE_L2C1 0x27a2

Bit(s)	R/W	Default	Description
19:16	R/W	0	wrarb_sel : uns, default = 0 , wrarb_sel [0]==0 slave dc0 connect master port0wrarb_sel[0]==1 slave dc0 connect master port1wrarb_sel [1]==0 slave dc1 connect master port0wrarb_sel[1]==1 slave dc1 connect master port1wrarb_sel [2]==0 slave dc2 connect master port0wrarb_sel[2]==1 slave dc2 connect master port1wrarb_sel [3]==0 slave dc3 connect master port0wrarb_sel[3]==1 slave dc3 connect master port1
9:8	R/W	0	wrarb_arb_mode : uns, default = 0 , wrarb_arb_mode [0] master port0 arb way,wrarb_arb_mode [1] master port1 arb way,
3:0	R/W	0	wrarb_gate_clk_ctrl : uns, default = 0 , wrarb_gate_clk_ctrl [1:0] master port0 clk gate controlwrarb_gate_clk_ctrl [3:2] master port0 clk gate control

VPU_WRARB_REQEN_SLV_L2C1 0x27a3

Bit(s)	R/W	Default	Description
7:0	R/W	0	wrarb_dc_req_en : unsigned , default = 0 wrarb_dc_req_en [0]: the slv0 req to mst port0 enable,wrarb_dc_req_en [1]: the slv1 req to mst port0 enable,wrarb_dc_req_en [2]: the slv2 req to mst port0 enable,wrarb_dc_req_en [3]: the slv3 req to mst port0 enable,wrarb_dc_req_en [0]: the slv0 req to mst port1 enable,wrarb_dc_req_en [1]: the slv1 req to mst port1 enable,wrarb_dc_req_en [2]: the slv2 req to mst port1 enable,wrarb_dc_req_en [3]: the slv3 req to mst port1 enable,

VPU_WRARB_WEIGH0_SLV_L2C1 0x27a4

Bit(s)	R/W	Default	Description
23:0	R/W	0	wrdc_weigh_sxn : unsigned , default = 0 wrdc_weigh_sxn [0*6+:6]: the slv0 req weigh numberwrdc_weigh_sxn [1*6+:6]: the slv1 req weigh numberwrdc_weigh_sxn [2*6+:6]: the slv2 req weigh numberwrdc_weigh_sxn [3*6+:6]: the slv3 req weigh number

VPU_ASYNC_RD_MODE0 0x27a5

Bit(s)	R/W	Default	Description
18	R/W	0	req_en : unsigned , default = 0 async enable
17:16	R/W	0	clk_gate_ctrl : unsigned , default = 0 async clock gate control
15:12	R/W	4	auto_arugt_weight : unsigned , default = 4

10:9	R/W	0	arugt_sel : unsigned , default = 0 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 00 : use the input arguent
8	R/W	0	arguent_cfg : unsigned , default = 0 register arguent control bit
7:4	R/W	4	rd_hold_num : unsigned , default = 4 hold the read command threshold
3:0	R/W	0	rd_rel_num : unsigned , default = 0 release the read command threshold

VPU_ASYNC_RD_MODE1 0x27a6

Bit(s)	R/W	Default	Description
18	R/W	0	req_en : unsigned , default = 0 async enable
17:16	R/W	0	clk_gate_ctrl : unsigned , default = 0 async clock gate control
15:12	R/W	4	auto_arugt_weight : unsigned , default = 4
10:9	R/W	0	arugt_sel : unsigned , default = 0 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 00 : use the input arguent
8	R/W	0	arguent_cfg : unsigned , default = 0 register arguent control bit
7:4	R/W	4	rd_hold_num : unsigned , default = 4 hold the read command threshold
3:0	R/W	0	rd_rel_num : unsigned , default = 0 release the read command threshold

VPU_ASYNC_RD_MODE2 0x27a7

Bit(s)	R/W	Default	Description
18	R/W	0	req_en : unsigned , default = 0 async enable
17:16	R/W	0	clk_gate_ctrl : unsigned , default = 0 async clock gate control
15:12	R/W	4	auto_arugt_weight : unsigned , default = 4
10:9	R/W	0	arugt_sel : unsigned , default = 0 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 00 : use the input arguent
8	R/W	0	arguent_cfg : unsigned , default = 0 register arguent control bit
7:4	R/W	4	rd_hold_num : unsigned , default = 4 hold the read command threshold
3:0	R/W	0	rd_rel_num : unsigned , default = 0 release the read command threshold

VPU_ASYNC_RD_MODE3 0x27a8

Bit(s)	R/W	Default	Description
18	R/W	0	req_en : unsigned , default = 0 async enable
17:16	R/W	0	clk_gate_ctrl : unsigned , default = 0 async clock gate control
15:12	R/W	4	auto_arugt_weight : unsigned , default = 4
10:9	R/W	0	arugt_sel : unsigned , default = 0 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 00 : use the input arguent

8	R/W	0	arguent_cfg : unsigned , default = 0 register arguent control bit
7:4	R/W	4	rd_hold_num : unsigned , default = 4 hold the read command threshold
3:0	R/W	0	rd_rel_num : unsigned , default = 0 release the read command threshold

VPU_ASYNC_RD_MODE4 0x27a9

Bit(s)	R/W	Default	Description
18	R/W	0	req_en : unsigned , default = 0 async enable
17:16	R/W	0	clk_gate_ctrl : unsigned , default = 0 async clock gate control
15:12	R/W	4	auto_arugt_weight : unsigned , default = 4
10:9	R/W	0	arugt_sel : unsigned , default = 0 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 00 : use the input arguent
8	R/W	0	arguent_cfg : unsigned , default = 0 register arguent control bit
7:4	R/W	4	rd_hold_num : unsigned , default = 4 hold the read command threshold
3:0	R/W	0	rd_rel_num : unsigned , default = 0 release the read command threshold

VPU_ASYNC_WR_MODE0 0x27aa

Bit(s)	R/W	Default	Description
18	R/W	0	req_en : unsigned , default = 0 async enable
17:16	R/W	0	clk_gate_ctrl : unsigned , default = 0 async clock gate control
15:12	R/W	4	auto_arugt_weight : unsigned , default = 4
10:9	R/W	0	arugt_sel : unsigned , default = 0 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 00 : use the input arguent
8	R/W	0	arguent_cfg : unsigned , default = 0 register arguent control bit
7:4	R/W	4	wr_hold_num : unsigned , default = 4 hold the read command threshold
3:0	R/W	0	wr_rel_num : unsigned , default = 0 release the write command threshold

VPU_ASYNC_WR_MODE1 0x27ab

Bit(s)	R/W	Default	Description
18	R/W	0	req_en : unsigned , default = 0 async enable
17:16	R/W	0	clk_gate_ctrl : unsigned , default = 0 async clock gate control
15:12	R/W	4	auto_arugt_weight : unsigned , default = 4
10:9	R/W	0	arugt_sel : unsigned , default = 0 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 00 : use the input arguent
8	R/W	0	arguent_cfg : unsigned , default = 0 register arguent control bit

7:4	R/W	4	wr_hold_num : unsigned , default = 4 hold the read command threshold
3:0	R/W	0	wr_rel_num : unsigned , default = 0 release the write command threshold

VPU_ASYNC_WR_MODE2 0x27ac

Bit(s)	R/W	Default	Description
18	R/W	0	req_en : unsigned , default = 0 async enable
17:16	R/W	0	clk_gate_ctrl : unsigned , default = 0 async clock gate control
15:12	R/W	4	auto_arugt_weight : unsigned , default = 4
10:9	R/W	0	arugt_sel : unsigned , default = 0 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 00 : use the input arguent
8	R/W	0	arguent_cfg : unsigned , default = 0 register arguent control bit
7:4	R/W	4	wr_hold_num : unsigned , default = 4 hold the read command threshold
3:0	R/W	0	wr_rel_num : unsigned , default = 0 release the write command threshold

VPU_ASYNC_STAT 0x27ad

Bit(s)	R/W	Default	Description
18	R/W	0x0	axiwr2_chan_idle : unsigned , RO, axi write channel2 idle state
17	R/W	0x0	axiwr1_chan_idle : unsigned , RO, axi write channel1 idle state
16	R/W	0x0	axiwr0_chan_idle : unsigned , RO, axi write channel0 idle state
4	R/W	0x0	axird4_chan_idle : unsigned , RO, axi read channel4 idle state
3	R/W	0x0	axird3_chan_idle : unsigned , RO, axi read channel3 idle state
2	R/W	0x0	axird2_chan_idle : unsigned , RO, axi read channel2 idle state
1	R/W	0x0	axird1_chan_idle : unsigned , RO, axi read channel1 idle state
0	R/W	0x0	axird0_chan_idle : unsigned , RO, axi read channel0 idle state

VPU_HDMI_DITH_01_04 0x27f0

Bit(s)	R/W	Default	Description
31-0	R/W	0x8214_1428	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_01_15 0x27f1

Bit(s)	R/W	Default	Description
31-0	R/W	0x4128_2841	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_01_26 0x27f2

Bit(s)	R/W	Default	Description
31-0	R/W	0x2841_4182	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_01_37 0x27f3

Bit(s)	R/W	Default	Description
31-0	R/W	0x1482_8214	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_10_04 0x27f4

Bit(s)	R/W	Default	Description
31-0	R/W	0x9669_9696	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_10_15 0x27f5

Bit(s)	R/W	Default	Description
31-0	R/W	0x3c3c_6969	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_10_26 0x27f6

Bit(s)	R/W	Default	Description
31-0	R/W	0x6996_9696	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_10_37 0x27f7

Bit(s)	R/W	Default	Description
31-0	R/W	0xc3c3_6969	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_11_04 0x27f8

Bit(s)	R/W	Default	Description
31-0	R/W	0x7deb_ebd7	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_11_15 0x27f9

Bit(s)	R/W	Default	Description

31-0	R/W	0xbed7_d7be	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_11_26 0x27fa

Bit(s)	R/W	Default	Description
31-0	R/W	0xd7be_be7d	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_11_37 0x27fb

Bit(s)	R/W	Default	Description
31-0	R/W	0xeb7d_7deb	dith lut VPU_HDMI_DITH_CNTL[0] : 1 : lut for 10b to 8b 0 : lut for 12b to 10b

VPU_HDMI_DITH_CNTL 0x27fc

Bit(s)	R/W	Default	Description
10b to 8b dither control register. 12b to 10b see VPU_HDMI_FMT_CTRL			
21-19	R/W	0	rame count offset for B
18-16	R/W	0	frame count offset for G
15	R/W	0	hcnt hold when de valid
14	R/W	0	RGB frame count seperate
13	R/W	0	dith4x4 : frame random enable
12	R/W	0	dith4x4 enable
11	R/W	0	tunnel enable for DOLBY
10	R/W	0	rounding enable
9-6	R/W	0	Cntl_hdmi_dith10 :
5	R/W	0	Cntl_hdmi_dith_md:
4	R/W	0	Cntl_hdmi_dith_en: dither 10-b to 8-b enable
3	R/W	0	hsync invert
2	R/W	0	vsync invert
0	R/W	0	dither lut sel : 1 : sel 10b to 8b 0: sel 12b to 10b

VPU_VENCL_DITH_CTRL 0x27e0

Bit(s)	R/W	Default	Description
19-17	R/W	0	dither 2x2 : frame number sel

Bit(s)	R/W	Default	Description
16	R/W	0	dither 2x2 : frame number random
15-14	R/W	0	Reserved
13-11	R/W	7	G frame number offset
10-8	R/W	3	B frame number offset
7	R/W	0	Reserved
6	R/W	1	dither 4x4 : G/B frame number = B frame number + offset g/b
5	R/W	0	dither 4x4 : frame number random
4	R/W	1	dither 4x4 : enable
3	R/W	0	Reserved
2	R/W	0	dither md
1	R/W	0	rounding enable
0	R/W	1	dither enable

VPU_VENCL_DITH_LUT_1 0x27e1

Bit(s)	R/W	Default	Description
31-0	R/W	0x8214_1428	dith lut

VPU_VENCL_DITH_LUT_2 0x27e2

Bit(s)	R/W	Default	Description
31-0	R/W	0x4128_2841	dith lut

VPU_VENCL_DITH_LUT_3 0x27e3

Bit(s)	R/W	Default	Description
31-0	R/W	0x2841_4182	dith lut

VPU_VENCL_DITH_LUT_4 0x27e4

Bit(s)	R/W	Default	Description
31-0	R/W	0x1482_8214	dith lut

VPU_VENCL_DITH_LUT_5 0x27e5

Bit(s)	R/W	Default	Description
31-0	R/W	0x9669_9696	dith lut

VPU_VENCL_DITH_LUT_6 0x27e6

Bit(s)	R/W	Default	Description
31-0	R/W	0x3c3c_6969	dith lut

VPU_VENCL_DITH_LUT_7 0x27e7

Bit(s)	R/W	Default	Description
31-0	R/W	0x6996_9696	dith lut

VPU_VENCL_DITH_LUT_8 0x27e8

Bit(s)	R/W	Default	Description
31-0	R/W	0xc3c3_6969	dith lut

VPU_VENCL_DITH_LUT_9 0x27e9

Bit(s)	R/W	Default	Description
31-0	R/W	0x7deb_ebd7	dith lut

VPU_VENCL_DITH_LUT_10 0x27ea

Bit(s)	R/W	Default	Description
31-0	R/W	0xbed7_d7be	dith lut

VPU_VENCL_DITH_LUT_11 0x27eb

Bit(s)	R/W	Default	Description
31-0	R/W	0xd7be_be7d	dith lut

VPU_VENCL_DITH_LUT_12 0x27ec

Bit(s)	R/W	Default	Description
31-0	R/W	0xeb7d_7deb	dith lut

10.2.3.2 VPU Video Lock Registers

VPU_VLOCK_CTRL 0x3000

Bit(s)	R/W	Default	Description
31	R/W	0x0	Vid_lock_en: 1: enable video lock module
30	R/W	0x0	Reg_adj_enc: enable video lock to adjust encoder
29	R/W	0x0	Adj_pll: enable video lock to adjust PLL
28	R/W	0x0	Mpeg_vs: set this to 1, then 0, this is software controlled mpeg vsync
27-26	R/W	0x0	Output goes to which module: 0: encl, 1: encp, 2:enci

Bit(s)	R/W	Default	Description
25-20	R/W	0x0	Output vsync width extend: make sure the vsync width is extended big enough for vpu_vid_lock_clk to sample
19	R/W	0x0	m frac right shift 1 : right shift 2 bit 0 : no shift
18-16	R/W	0x0	Input Vsync source select: 0: unuse, 1: fromhdmi rx , 2:from tv-decoder, 3: from dvin, 4: from dvin, 5: from 2nd bt656
15	R/W	0x0	Output vsync invert: 1, invert
14	R/W	0x0	Input vsync invert: 1, invert
13-8	R/W	0x0	Input vsync width extend: make sure the vsync width is extended big enough for vpu_vid_lock_clk to sample
7	R/W	0x0	Force loop1 err enable: 1. Force error of loop1
6	R/W	0x0	Force loop0 err enable: 1. Force error of loop0
5	R/W	0x0	Overwrite accum0 enable
4	R/W	0x0	Loop0 adjust capture enable
3	R/W	0x0	Loop0 adjust pll enable
2	R/W	0x0	Overwrite accum1 enable
1	R/W	0x0	Loop1 adjust capture enable
0	R/W	0x0	Loop0 adjust pll enable

VPU_VLOCK_MISC_CTRL 0x3001

Bit(s)	R/W	Default	Description
26-24	R/W	0x0	Adj_capt_pxgrouprs, make sure the pixel number in one line of encoder is multiples of 2^pxgrouprs
23-16	R/W	0x0	Ifrm_cnt_mod: (output vsync freq)/(input vsync_freq * ifrm_cnt_mod) must be integer
15-8	R/W	0x0	Output vsync frequency
7-0	R/W	0x0	Input vsync frequency

VPU_VLOCK_LOOP0_ACCUM_LMT 0x3002

Bit(s)	R/W	Default	Description
26-0	R/W	0x0	LOOP0 accumulator limit

VPU_VLOCK_LOOP0_CTRL0 0x3003

Bit(s)	R/W	Default	Description
31-24	R/W	0x0	Loop0 errclip rate

23-20	R/W	0x0	Loop0_adj_pll_rs, right shift of loop0 adjust pll portion
19-12	R/W	0x0	Loop0_adj_pll_gain, u1.7
11-8	R/W	0x0	Loop0_adj_capt_rs, right shift of loop0 adjust capture portion
7-0	R/W	0x0	Loop0_adj_capt_gain

VPU_VLOCK_LOOP0_CTRL1 0x3004

Bit(s)	R/W	Default	Description
23-20	R/W	0x0	Loop1_adj_pll_rs
19-12	R/W	0x0	Loop1_adj_pll_gain
11-8	R/W	0x0	Loop1_adj_capt_rs
7-0	R/W	0x0	Loop1_adj_capt_gain

VPU_VLOCK_LOOP1_IMISSYNC_MAX 0x3005

Bit(s)	R/W	Default	Description
27-0	R/W	0x0	Loop1 imissync max, input signal is missed after input vsync counter is larger than this max threshold

VPU_VLOCK_LOOP1_IMISSYNC_MIN 0x3006

Bit(s)	R/W	Default	Description
27-0	R/W	0x0	Loop1 imissync min, input signal is missed after input vsync counter is less than this max threshold

VPU_VLOCK_OVERWRITE_ACCUM0 0x3007

Bit(s)	R/W	Default	Description
27-0	R/W	0x0	Overwrite value of accum0

VPU_VLOCK_OVERWRITE_ACCUM1 0x3008

Bit(s)	R/W	Default	Description
27-0	R/W	0x0	Overwrite value of accum1

VPU_VLOCK_OUTPUT0_CAPT_LMT 0x3009

Bit(s)	R/W	Default	Description
26-0	R/W	0x0	Output0 capture limit

VPU_VLOCK_OUTPUT0_PLL_LMT 0x300a

Bit(s)	R/W	Default	Description
26-0	R/W	0x0	Output0 pll limit

VPU_VLOCK_OUTPUT1_CAPT_LMT 0x300b

Bit(s)	R/W	Default	Description
26-0	R/W	0x0	Output1 capture limit

VPU_VLOCK_OUTPUT1_PLL_LMT 0x300c

Bit(s)	R/W	Default	Description
26-0	R/W	0x0	Output1 pll limit

VPU_VLOCK_LOOP1_PHSDIF_TARGET 0x300d

Bit(s)	R/W	Default	Description
27-0	R/W	0x0	Loop1 phase difference target, (input vsync - output vsync) phase distance target

VPU_VLOCK_RO_LOOP0_ACCUM 0x300e

Bit(s)	R/W	Default	Description
27-0	R	0x0	Read only, loop0 accum result

VPU_VLOCK_RO_LOOP1_ACCUM 0x300f

Bit(s)	R/W	Default	Description
27-0	R	0x0	Read only, loop1 accum result

VPU_VLOCK_OROW_OCOL_MAX 0x3010

Bit(s)	R/W	Default	Description
29-16	R/W	0x0	Ocol_max
13-0	R/W	0x0	Orow_max

VPU_VLOCK_RO_VS_I_D 0x3011

Bit(s)	R/W	Default	Description
27-0	R	0x0	Read only, input vsync counter

VPU_VLOCK_RO_VS_O_D 0x3012

Bit(s)	R/W	Default	Description
27-0	R	0x0	Read only, output vsync counter

VPU_VLOCK_RO_LINE_PIX_ADJ 0x3013

Bit(s)	R/W	Default	Description
29-16	R	0x0	Read only, encoder line adjust number
13-0	R	0x0	Read only, encoder pix adjust number

VPU_VLOCK_RO_OUTPUT_00_01 0x3014

Bit(s)	R/W	Default	Description
31-16	R	0x0	Read only, accum0 output 00
15-0	R	0x0	Read only, accum0 output 01

VPU_VLOCK_RO_OUTPUT_10_11 0x3015

Bit(s)	R/W	Default	Description
31-16	R	0x0	Read only, accum1 output 10
15-0	R	0x0	Read only, accum1 output 11

VPU_VLOCK_MX4096 0x3016

Bit(s)	R/W	Default	Description
20-0	R/W	0x0	Mx4096

VPU_VLOCK_STBDET_WIN0_WIN1 0x3017

Bit(s)	R/W	Default	Description
15-8	R/W	0x0	Verr_stbdet_win1
7-0	R/W	0x0	Verr_stbdet_win0

VPU_VLOCK_STBDET_CLP 0x3018

Bit(s)	R/W	Default	Description
15-8	R	0x0	Read only, ro_verr_clp_win1, verr_clp number in win0
7-0	R	0x0	Read only, ro_verr_clp_win0, verr clp number in win1

VPU_VLOCK_STBDET_ABS_WIN0 0x3019

Bit(s)	R/W	Default	Description
23-0	R	0x0	Read only, ro_verr_abs_win0

VPU_VLOCK_STBDET_ABS_WIN1 0x301a

Bit(s)	R/W	Default	Description
23-0	R	0x0	Read only, ro_verr_abs_win1

VPU_VLOCK_STBDET_SGN_WIN0 0x301b

Bit(s)	R/W	Default	Description
23-0	R	0x0	Read only, ro_verr_sgn_win0

VPU_VLOCK_STBDET_SGN_WIN1 0x301c

Bit(s)	R/W	Default	Description
23-0	R	0x0	Read only, ro_verr_sgn_win1

VPU_VLOCK_ADJ_EN_SYNC_CTRL 0x301d

Bit(s)	R/W	Default	Description
31-24	R/W	0x0	PLL adjust enable signal sync ctrl, adj_en_for_pll_end, end counter of adj_en_pll signal fall to 0
23-16	R/W	0x0	PLL adjust enable signal sync ctrl, adj_en_for_pll_start, start counter of adj_en_pll signal go to 1, start must be larger than end
15-8	R/W	0x0	Adj_en_sync_latch_cnt, this is a delay to latch the adj_en signal
7-0	R/W	0x0	Adj_en_ext_cnt, extend the adj_en signal from vid_lock clock domain to pll sample domain, make sure it's wide enough

VPU_VLOCK_GCLK_EN 0x301e

Bit(s)	R/W	Default	Description
2	R/W	0x0	Ref clock enable
1	R/W	0x0	Vsout clk enable
0	R/W	0x0	Vsin clk enable

VPU_VLOCK_LOOP1_ACCUM_LMT 0x301f

Bit(s)	R/W	Default	Description
26-0	R/W	0x0	LOOP1 accumulator limit

VPU_VLOCK_RO_M_INT_FRAC 0x3020

Bit(s)	R/W	Default	Description
29-16	R	0x0	Read only, m_int to PLL
13-0	R	0x0	Read only, m_frac to PLL

10.2.3.3 VIU Top-Level Registers

VIU_SW_RESET 0x1A01

Bit(s)	R/W	Default	Description
31	R/W	0	Osd1 afbcd reset
30	R/W	0	hist_spl reset
29	R/W	0	Ldim stts reset
8	R/W	0	Vd2 Dos afbcd reset

7	R/W	0	vpp_reset
6	R/W	0	di_dsr1to2_reset
5	R/W	0	vd2_fmt_reset
4	R/W	0	vd2_reset
3	R/W	0	vd1_fmt_reset
2	R/W	0	vd1_reset
1	R/W	0	osd2_reset
0	R/W	0	osd1_reset

VIU_SW_RESET0 0x1A02

Bit(s)	R/W	Default	Description
2	R/W	0	Vd1 Dos afbcd reset

VIU_MISC_CTRL0 0x1A06

Bit(s)	R/W	Default	Description
29	R/W	0	afbc go field/line _sel: 1->pre frame rst/pre_go_line; 0-> enci/p/l frame rst/go_line
28	R/W	0	vd1 go field/line _sel: 1->post frame rst/post_go_line; 0-> enci/p/l frame rst/go_line
27-22	R/W	0	afbc gate clk ctrl
20	R/W	0	afbc vd1 set: 1-> afbc to vd1(afbc) ; 0->vd1 mif(ddr) to vd1(afbc)
18	R/W	0	di_mad_en: di post to vpp enable
17	R/W	0	mif0_to_vpp_en: enable vd1(afbc) to vpp
16	R/W	0	di_mif0_en: vd1(afbc) to di post(if0) enable
8	R/W	0	vsync_int , enable field to vsync
6:5	R/W	0	2'b0:close data_path 2'b01: data path to tffbff_dscale 2'b10: data path to NR,2'b11: data path to tffbff_dscale & NR
4	R/W	0	0:data come from di_mif0 1: vpp input as di post din
3:2	R/W	0	Bit3:Deint_wr_mif luma fix_disable. Bit2:Deint_wr_mif chroma fix disable.
1:0	R/W	0	Bit1:Nr_wr_mif luma_fix_disable Bit0:Nr_wr_mif chroma_fix_disable

VIU_MISC_CTRL1 0x1A07

Bit(s)	R/W	Default	Description
27-22	R/W	0	afbc gate clk ctrl

15:14	R/W	0	Dvalin afbcd clock gate control
12	R/W	0	Osd1 axi bus select 1 : select Dvalin afbcd 0 : normal osd1
11:8	R/W	0	di_mad_en: di post to vpp enable
7-2	R/W	0	Afbcd2 Clock gate control
1	R/W	0	1 : connect dos afbcd2 to vpp vd1, 0 : connect mif to vpp vd1
0	R/W	0	1 : Dos afbcd2 output to di ; 0 : dos afbcd2 output to vpp

VIUB_SW_RESET 0x2001

Bit(s)	R/W	Default	Description
31	R/W	0	mcvecwr_mif_rst_n
30	R/W	0	reserved
29	R/W	0	reserved
28	R/W	0	di_cont_rd_mif_rst_n
27	R/W	0	di_cont_wr_mif_rst_n
26	R/W	0	reserved
25	R/W	0	reserved
24	R/W	0	vdin1_wr_rst_n
23	R/W	0	vdin0_wr_rst_n
22	R/W	0	nrin_mux_rst_n
21	R/W	0	vdin1_rst_n
20	R/W	0	vdin0_rst_n
19	R/W	0	di_mad_rst_n
18	R/W	0	di_mtn_rd_mif_rst_n
17	R/W	0	di_mtn_wr_mif_rst_n
16	R/W	0	di_chan2_mif_rst_n
15	R/W	0	dein_wr_mif_rst_n
14	R/W	0	di_nr_wr_mif_rst_n
13	R/W	0	di_mem_fmt_rst_n
12	R/W	0	di_mem_rst_n
11	R/W	0	di_inp_fmt_rst_n
10	R/W	0	di_inp_rst_n

Bit(s)	R/W	Default	Description
9	R/W	0	di_if1_fmt_rst_n
8	R/W	0	di_if1_rst_n
7-0	R/W	0	RESERVED

VIUB_SW_RESET0 0x2002

Bit(s)	R/W	Default	Description
3	R/W	0	di_axi_arb_rst_n
2	R/W	0	mcinford_mif_rst_n
1	R/W	0	mcinfowr_mif_rst_n
0	R/W	0	mcvecrd_mif_rst_n

VIUB_MISC_CTRL0 0x2006

Bit(s)	R/W	Default	Description
17	R/W	0	input2pre enable: 1->di inp data from vdin0 0->di inp data from inp_afbc(see bit16)
16	R/W	0	AFBC_INP_SEL: 1->di inp_afbc data from afbc 0->di inp_afbc data from inp mif(ddr)
7-6	R/W	0	Fix_disable: vdin1_wr_mif
5-4	R/W	0	Fix_disable: vdin0_wr_mif
3-2	R/W	0	Fix_disable:dein_wr_mif
1-0	R/W	0	Fix_disable: di_nr_wr_mif

VIUB_GCLK_CTRL0 0x2007

Bit(s)	R/W	Default	Description
31:16	R/W	0	Reserved
15	R/W	0	Di_gate_all,for old di
14	R/W	1	Di_no_clk_gate,for old di
13	R/W	0	reserved
12	R/W	0	Di_post clock enable ,from div clock
11	R/W	0	Mcdi clock enable,from div clock
10	R/W	0	Div clock enable,di slow clock including di&mcdi
9	R/W	0	Mad post clock enable,from mad clock
8	R/W	0	Mad pre clock enable,from mad clock

7:1	R/W		reserved
0	R/W	1	Def=1 di_top_wrap clk enable

VIUB_GCLK_CTRL1 0x2008

Bit(s)	R/W	Default	Description
31:28	R/W	0	Reserved
27:26	R/W	0	Mcdi pre mif clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
25:24	R/W	0	Mtn mif clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
23:22	R/W	0	Nr wrmif clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
21:20	R/W	0	Chan rdmif clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
19:18	R/W	0	Mem rdmif clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
17:16	R/W	0	Inp rdmif clock gate clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
15:12	R/W	0	reserved
11:10	R/W	0	Mcdi post clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
9-8	R/W	0	Mtnrd post mif clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
7-6	R/W	0	De wrmif clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
5-4	R/W	0	If2 rdmif clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
3-2	R/W	0	If1 rdmif clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
1-0	R/W	0	Mif-sub-arb clock gate 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock

VIUB_GCLK_CTRL2 0x2009

Bit(s)	R/W	Default	Description
31-14	R/W	0	reserved
13-12	R/W	0	mcdi clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
11:10	R/W	0	Nr&dnr blend clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
9-8	R/W	0	Dnr clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
7-6	R/W	0	nning clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
5-4	R/W	0	Mtn det clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
3-2	R/W	0	pd clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
1-0	R/W	0	Nr clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock

VIUB_GCLK_CTRL3 0x200a

Bit(s)	R/W	Default	Description
31-6	R/W	0	Reserved
5-4	R/W	0	Di blend clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
3-2	R/W	0	Ei clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock
1-0	R/W	0	Ei_0 clock gate ctrl 2'b00: gate clock ,2'b01: close clock,2'b1x: always clock

VIU_MISC_CTRL1 0x3107

Bit(s)	R/W	Default	Description
15:14	R/W	0x0	Dvalin_afbcd_gclk_ctrl : Dvalin_afbcd clock gate control[5:4]
12	R/W	0x0	osd1_afbcd_axi_mux : 0 : use the osd mif as input; 1 : use afbcd as input
11:8	R/W	0x0	Dvalin_afbcd_gclk_ctrl : Dvalin_afbcd clock gate control[3:0]
7:2	R/W	0x0	vd2_afbcd_gclk_ctrl : vd2_afbcd clock gate control
1	R/W	0x0	vpp_vd2_din_sel : 0: vpp vd2 sel the mif input; 1: vpp vd2 sel the dos afbcd
0	R/W	0x0	vd2_afbcd_out_sel : 0: vd2_afbcd output to vpp; 1 : vd2_afbcd output to di inp

10.2.3.4 DI_AXI_ARB Registers

DI_RDARB_MODE_L1C1 0x2050

Bit(s)	R/W	Default	Description
21:16	R/W	0	rdarb_sel : uns, default = 0 ,rdarb_sel [0]==0 slave dc0 connect master port0rdarb_sel[0]==1 slave dc0 connect master port1rdarb_sel [1]==0 slave dc1 connect master port0rdarb_sel[1]==1 slave dc1 connect master port1rdarb_sel [2]==0 slave dc2 connect master port0rdarb_sel[2]==1 slave dc2 connect master port1rdarb_sel [3]==0 slave dc3 connect master port0rdarb_sel[3]==1 slave dc3 connect master port1rdarb_sel [4]==0 slave dc4 connect master port0rdarb_sel[4]==1 slave dc4 connect master port1rdarb_sel [5]==0 slave dc5 connect master port0rdarb_sel[5]==1 slave dc5 connect master port1
9:8	R/W	0	rdarb_arb_mode : uns, default = 0 ,rdarb_arb_mode [0] master port0 arb way,rdarb_arb_mode [1] master port1 arb way,
3:0	R/W	0	rdarb_gate_clk_ctrl : uns, default = 0 ,rdarb_gate_clk_ctrl [1:0] master port0 clk gate controlrdarb_gate_clk_ctrl [3:2] master port1 clk gate control

DI_RDARB_REQEN_SLV_L1C1 0x2051

Bit(s)	R/W	Default	Description
11:0	R/W	0xfff	rdarb_dc_req_en : unsigned , default = 12'hfffrdarb_dc_req_en [0]: the slv0 req to mst port0 enable,rdarb_dc_req_en [1]: the slv1 req to mst port0 enable,rdarb_dc_req_en [2]: the slv2 req to mst port0 enable,rdarb_dc_req_en [3]: the slv3 req to mst port0 enable,rdarb_dc_req_en [4]: the slv4 req to mst port0 enable,rdarb_dc_req_en [5]: the slv5 req to mst port0 enable,rdarb_dc_req_en [6]: the slv0 req to mst port1 enable,rdarb_dc_req_en [7]: the slv1 req to mst port1 enable,rdarb_dc_req_en [8]: the slv2 req to mst port1 enable,rdarb_dc_req_en [9]:

Bit(s)	R/W	Default	Description
			the slv3 req to mst port1 enable,rdarb_dc_req_en [10]: the slv4 req to mst port1 enable,rdarb_dc_req_en [11]: the slv5 req to mst port1 enable,

DI_RDARB_WEIGH0_SLV_L1C1 0x2052

Bit(s)	R/W	Default	Description
29:0	R/W	0	rddc_weigh_sxn : unsigned , default = 0 rddc_weigh_sxn [0*6+:6]: the slv0 req weigh number rddc_weigh_sxn [1*6+:6]: the slv1 req weigh number rddc_weigh_sxn [2*6+:6]: the slv2 req weigh number rddc_weigh_sxn [3*6+:6]: the slv3 req weigh number rddc_weigh_sxn [4*6+:6]: the slv4 req weigh number

DI_RDARB_WEIGH1_SLV_L1C1 0x2053

Bit(s)	R/W	Default	Description
5:0	R/W	0	rddc_weigh_sxn : unsigned , default = 0 rddc_weigh_sxn [5*6+:6]: the slv5 req weigh number

DI_WRARB_MODE_L1C1 0x2054

Bit(s)	R/W	Default	Description
21:16	R/W	0	wrarb_sel : uns, default = 0 ,wrarb_sel [0]==0 slave dc0 connect master port0wrarb_sel[0]==1 slave dc0 connect master port1wrarb_sel [1]==0 slave dc1 connect master port0wrarb_sel[1]==1 slave dc1 connect master port1wrarb_sel [2]==0 slave dc2 connect master port0wrarb_sel[2]==1 slave dc2 connect master port1wrarb_sel [3]==0 slave dc3 connect master port0wrarb_sel[3]==1 slave dc3 connect master port1wrarb_sel [4]==0 slave dc4 connect master port0wrarb_sel[4]==1 slave dc4 connect master port1wrarb_sel [5]==0 slave dc5 connect master port0wrarb_sel[5]==1 slave dc5 connect master port1
9:8	R/W	0	wrarb_arb_mode : uns, default = 0 ,wrarb_arb_mode [0] master port0 arb way,wrarb_arb_mode [1] master port1 arb way,
3:0	R/W	0	wrarb_gate_clk_ctrl : uns, default = 0 ,wrarb_gate_clk_ctrl [1:0] master port0 clk gate controlwrarb_gate_clk_ctrl [3:2] master port1 clk gate control

DI_WRARB_REQEN_SLV_L1C1 0x2055

Bit(s)	R/W	Default	Description
11:0	R/W	0	wrarb_dc_req_en : unsigned , default = 0wrarb_dc_req_en [0]: the slv0 req to mst port0 enable,wrarb_dc_req_en [1]: the slv1 req to mst port0 enable,wrarb_dc_req_en [2]: the slv2 req to mst port0 enable,wrarb_dc_req_en [3]: the slv3 req to mst port0 enable,wrarb_dc_req_en [4]: the slv4 req to mst port0 enable,wrarb_dc_req_en [5]: the slv5 req to mst port0 enable,wrarb_dc_req_en [0]: the slv0 req to mst port1 enable,wrarb_dc_req_en [1]: the slv1 req to mst port1 enable,wrarb_dc_req_en [2]: the slv2 req to mst port1 enable,wrarb_dc_req_en [3]: the slv3 req to mst port1 enable,wrarb_dc_req_en [4]: the slv4 req to mst port1 enable,wrarb_dc_req_en [5]: the slv5 req to mst port1 enable,

DI_WRARB_WEIGH0_SLV_L1C1 0x2056

Bit(s)	R/W	Default	Description

29:0	R/W	0	wrdc_weigh_sxn : unsigned , default = 0wrdc_weigh_sxn [0*6+:6]: the slv0 req weigh numberwrdc_weigh_sxn [1*6+:6]: the slv1 req weigh numberwrdc_weigh_sxn [2*6+:6]: the slv2 req weigh numberwrdc_weigh_sxn [3*6+:6]: the slv3 req weigh numberwrdc_weigh_sxn [4*6+:6]: the slv4 req weigh number

DI_WRARB_WEIGH1_SLV_L1C1 0x2057

Bit(s)	R/W	Default	Description
5:0	R/W	0	wrdc_weigh_sxn : unsigned , default = 0wrdc_weigh_sxn [5*6+:6]: the slv5 req weigh number

DI_RDWR_ARB_STATUS_L1C1 0x2058

Bit(s)	R/W	Default	Description
3:2	R/W	0	wrarb_arb_busy : unsigned , default = 0
1:0	R/W	0	rdarb_arb_busy : unsigned , default = 0

DI_ARB_DBG_CTRL_L1C1 0x2059

Bit(s)	R/W	Default	Description
31:0	R/W	8	det_cmd_ctrl : unsigned , default = 8

DI_ARB_DBG_STAT_L1C1 0x205a

Bit(s)	R/W	Default	Description
31:0	R/W	0	det_dbg_stat : unsigned , default = 0

DI_RDARB_UGT_L1C1 0x205b

Bit(s)	R/W	Default	Description
15:0	R/W	0x0	rdarb_ugt_basic : unsigned , default = {8{2'b1}};

DI_RDARB_LIMT0_L1C1 0x205c

Bit(s)	R/W	Default	Description
31:0	R/W	0x0	rdarb_req_limt_num : unsigned , default = {2{16'h3f3f}};

DI_WRARB_UGT_L1C1 0x205d

Bit(s)	R/W	Default	Description
11:0	R/W	0	wrarb_ugt_basic : unsigned , default = 0

DI_SUB_RDARB_MODE 0x37c0

Bit(s)	R/W	Default	Description
23:16	R/W	0	rdarb_sel : uns, default = 0 ,rdarb_sel [0]==0 slave dc0 connect master port0rdarb_sel[0]==1 slave dc0 connect master port1rdarb_sel [1]==0 slave dc1 connect master port0rdarb_sel[1]==1 slave dc1 connect master port1rdarb_sel [2]==0 slave dc2 connect master port0rdarb_sel[2]==1 slave dc2 connect master

Bit(s)	R/W	Default	Description
			port1rdarb_sel [3]==0 slave dc3 connect master port0rdarb_sel[3]==1 slave dc3 connect master port1rdarb_sel [4]==0 slave dc4 connect master port0rdarb_sel[4]==1 slave dc4 connect master port1rdarb_sel [5]==0 slave dc5 connect master port0rdarb_sel[5]==1 slave dc5 connect master port1rdarb_sel [6]==0 slave dc5 connect master port0rdarb_sel[6]==1 slave dc6 connect master port1rdarb_sel [7]==0 slave dc5 connect master port0rdarb_sel[7]==1 slave dc7 connect master port1
9:8	R/W	0	rdarb_arb_mode : uns, default = 0 ,rdarb_arb_mode [0] master port0 arb way,rdarb_arb_mode [1] master port1 arb way,
3:0	R/W	0	rdarb_gate_clk_ctrl : uns, default = 0 ,rdarb_gate_clk_ctrl [1:0] master port0 clk gate controlrdarb_gate_clk_ctrl [3:2] master port1 clk gate control

DI_SUB_RDARB_REQEN_SLV 0x37c1

Bit(s)	R/W	Default	Description
15:0	R/W	0xffff	rdarb_dc_req_en : uns, default = 16'hffff , slv0~slv7 enable to mst.

DI_SUB_RDARB_WEIGH0_SLV 0x37c2

Bit(s)	R/W	Default	Description
29:0	R/W	0	rddc_weigh_sxn : unsigned , default = 0 rddc_weigh_sxn [0*6+:6]: the slv0 req weigh number rddc_weigh_sxn [1*6+:6]: the slv1 req weigh number rddc_weigh_sxn [2*6+:6]: the slv2 req weigh number rddc_weigh_sxn [3*6+:6]: the slv3 req weigh number rddc_weigh_sxn [4*6+:6]: the slv4 req weigh number

DI_SUB_RDARB_WEIGH1_SLV 0x37c3

Bit(s)	R/W	Default	Description
17:0	R/W	0	rddc_weigh_sxn : unsigned , default = 0 rddc_weigh_sxn [0*6+:6]: the slv6 req weigh number rddc_weigh_sxn [1*6+:6]: the slv7 req weigh number rddc_weigh_sxn [2*6+:6]: the slv8 req weigh number

DI_SUB_RDARB_UGT 0x37c4

Bit(s)	R/W	Default	Description
15:0	R/W	0x0	rdarb_ugt_basic : unsigned , default = {8'{2'h1}}rdarb_ugt_basic [0*1+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [1*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [2*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [3*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [4*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [5*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [6*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [7*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguen

DI_SUB_RDARB_LIMT0 0x37c5

Bit(s)	R/W	Default	Description

31:0	R/W	0x0	rdarb_req_limt_num : unsigned , default = {2{16'h3f3f}}

DI_SUB_WRARB_MODE 0x37c6

Bit(s)	R/W	Default	Description
21:16	R/W	0	wrarb_sel : uns, default = 0 ,wrarb_sel [0]==0 slave dc0 connect master port0wrarb_sel[0]==1 slave dc0 connect master port1wrarb_sel [1]==0 slave dc1 connect master port0wrarb_sel[1]==1 slave dc1 connect master port1wrarb_sel [2]==0 slave dc2 connect master port0wrarb_sel[2]==1 slave dc2 connect master port1wrarb_sel [3]==0 slave dc3 connect master port0wrarb_sel[3]==1 slave dc3 connect master port1wrarb_sel [4]==0 slave dc4 connect master port0wrarb_sel[4]==1 slave dc3 connect master port1wrarb_sel [5]==0 slave dc5 connect master port0wrarb_sel[5]==1 slave dc3 connect master port1
8	R/W	0	wrarb_arb_mode : uns, default = 0 ,wrarb_arb_mode[0] master port0 arb way,
1:0	R/W	0	wrarb_gate_clk_ctrl : uns, default = 0 ,wrarb_gate_clk_ctrl[1:0] master port0 clk gate control

DI_SUB_WRARB_REQEN_SLV 0x37c7

Bit(s)	R/W	Default	Description
5:0	R/W	0	wrarb_dc_req_en : unsigned , default = 0wrarb_dc_req_en [0]: the slv0 req to mst port0 enable,wrarb_dc_req_en [1]: the slv1 req to mst port0 enable,wrarb_dc_req_en [2]: the slv2 req to mst port0 enable,wrarb_dc_req_en [0]: the slv0 req to mst port1 enable,wrarb_dc_req_en [1]: the slv1 req to mst port1 enable,wrarb_dc_req_en [2]: the slv2 req to mst port1 enable,

DI_SUB_WRARB_WEIGH0_SLV 0x37c8

Bit(s)	R/W	Default	Description
29:0	R/W	0	wrdc_weigh_sxn : unsigned , default = 0wrdc_weigh_sxn [0*6+:6]: the slv0 req weigh numberwrdc_weigh_sxn [1*6+:6]: the slv1 req weigh numberwrdc_weigh_sxn [2*6+:6]: the slv2 req weigh numberwrdc_weigh_sxn [3*6+:6]: the slv3 req weigh numberwrdc_weigh_sxn [4*6+:6]: the slv4 req weigh number

DI_SUB_WRARB_WEIGH1_SLV 0x37c9

Bit(s)	R/W	Default	Description
5:0	R/W	0	wrdc_weigh_sxn : unsigned , default = 0 the slv5 req weigh number

DI_SUB_WRARB_UGT 0x37ca

Bit(s)	R/W	Default	Description
11:0	R/W	0x0	rdarb_ugt_basic : unsigned , default = {8'{2'h1}}rdarb_ugt_basic [0*1+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [1*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [2*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [3*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [4*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguenrdarb_ugt_basic [5*2+:2]: 00 : use auto fifo arugt generate the output arugt. 01 : use the register bit control 10 : use the input arguen

DI_SUB_RDWR_ARB_STATUS 0x37cb

Bit(s)	R/W	Default	Description
2	R.O	0	ro_wrarb_arb_busy : unsigned , default = 0
1	R/W	0x0	reserve :
0	R.O	0	ro_rdarb_arb_busy : unsigned , default = 0

DI_SUB_ARB_DBG_CTRL 0x37cc

Bit(s)	R/W	Default	Description
31:0	R.O	0	ro_det_cmd_ctrl : unsigned , default = 0

DI_SUB_ARB_DBG_STAT 0x37cd

Bit(s)	R/W	Default	Description
31:0	R.O	0	ro_det_dbg_stat : unsigned , default = 0

CONTRD_CTRL1 0x37d0

Bit(s)	R/W	Default	Description
25:24	R/W	0	reg_sync_sel : unsigned , default = 0
23:16	R/W	0	reg_canvas_id : unsigned , default = 0
14:12	R/W	1	reg_cmd_intr_len : unsigned , default = 1
11:10	R/W	1	reg_cmd_req_size : unsigned , default = 1
9:8	R/W	2	reg_burst_len : unsigned , default = 2
7	R/W	0	reg_swap_64bit : unsigned , default = 0
6	R/W	0	reg_little_endian : unsigned , default = 0
5	R/W	0	reg_y_rev : unsigned , default = 0
4	R/W	0	reg_x_rev : unsigned , default = 0
2:0	R/W	1	reg_pack_mode : unsigned , default = 1

CONTRD_CTRL2 0x37d1

Bit(s)	R/W	Default	Description
31:30	R/W	0	reg_sw_rst : unsigned , default = 0
23:18	R/W	0	reg_gclk_ctrl : unsigned , default = 0
16:0	R/W	0	reg_urgent_ctrl : unsigned , default = 0

CONTRD_SCOPE_X 0x37d2

Bit(s)	R/W	Default	Description

28:16	R/W	0	reg_x_end : unsigned , default = 0
12:0	R/W	0	reg_x_start : unsigned , default = 0

CONTRD_SCOPE_Y 0x37d3

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_y_end : unsigned , default = 0
12:0	R/W	0	reg_y_start : unsigned , default = 0

CONTRD_RO_STAT 0x37d4

Bit(s)	R/W	Default	Description
15:0	R.O	0	ro_reg_status : unsigned , default = 0

CONT2RD_CTRL1 0x37d5

Bit(s)	R/W	Default	Description
25:24	R/W	0	reg_sync_sel : unsigned , default = 0
23:16	R/W	0	reg_canvas_id : unsigned , default = 0
14:12	R/W	1	reg_cmd_intr_len : unsigned , default = 1
11:10	R/W	1	reg_cmd_req_size : unsigned , default = 1
9:8	R/W	2	reg_burst_len : unsigned , default = 2
7	R/W	0	reg_swap_64bit : unsigned , default = 0
6	R/W	0	reg_little_endian : unsigned , default = 0
5	R/W	0	reg_y_rev : unsigned , default = 0
4	R/W	0	reg_x_rev : unsigned , default = 0
2:0	R/W	1	reg_pack_mode : unsigned , default = 1

CONT2RD_CTRL2 0x37d6

Bit(s)	R/W	Default	Description
31:30	R/W	0	reg_sw_rst : unsigned , default = 0
23:18	R/W	0	reg_gclk_ctrl : unsigned , default = 0
16:0	R/W	0	reg_urgent_ctrl : unsigned , default = 0

CONT2RD_SCOPE_X 0x37d7

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_x_end : unsigned , default = 0
12:0	R/W	0	reg_x_start : unsigned , default = 0

CONT2RD_SCOPE_Y 0x37d8

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_y_end : unsigned , default = 0
12:0	R/W	0	reg_y_start : unsigned , default = 0

CONT2RD_RO_STAT 0x37d9

Bit(s)	R/W	Default	Description
15:0	R.O	0	ro_reg_status : unsigned , default = 0

MTNRD_CTRL1 0x37da

Bit(s)	R/W	Default	Description
25:24	R/W	0	reg_sync_sel : unsigned , default = 0
23:16	R/W	0	reg_canvas_id : unsigned , default = 0
14:12	R/W	1	reg_cmd_intr_len : unsigned , default = 1
11:10	R/W	1	reg_cmd_req_size : unsigned , default = 1
9:8	R/W	2	reg_burst_len : unsigned , default = 2
7	R/W	0	reg_swap_64bit : unsigned , default = 0
6	R/W	0	reg_little_endian : unsigned , default = 0
5	R/W	0	reg_y_rev : unsigned , default = 0
4	R/W	0	reg_x_rev : unsigned , default = 0
2:0	R/W	1	reg_pack_mode : unsigned , default = 1

MTNRD_CTRL2 0x37db

Bit(s)	R/W	Default	Description
31:30	R/W	0	reg_sw_rst : unsigned , default = 0
23:18	R/W	0	reg_gclk_ctrl : unsigned , default = 0
16:0	R/W	0	reg_urgent_ctrl : unsigned , default = 0

MTNRD_SCOPE_X 0x37dc

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_x_end : unsigned , default = 0
12:0	R/W	0	reg_x_start : unsigned , default = 0

MTNRD_SCOPE_Y 0x37dd

Bit(s)	R/W	Default	Description

28:16	R/W	0	reg_y_end : unsigned , default = 0
12:0	R/W	0	reg_y_start : unsigned , default = 0

MTNRD_RO_STAT 0x37de

Bit(s)	R/W	Default	Description
15:0	R.O	0	ro_reg_status : unsigned , default = 0

MCVECRD_CTRL1 0x37df

Bit(s)	R/W	Default	Description
25:24	R/W	0	reg_sync_sel : unsigned , default = 0
23:16	R/W	0	reg_canvas_id : unsigned , default = 0
14:12	R/W	1	reg_cmd_intr_len : unsigned , default = 1
11:10	R/W	1	reg_cmd_req_size : unsigned , default = 1
9:8	R/W	2	reg_burst_len : unsigned , default = 2
7	R/W	0	reg_swap_64bit : unsigned , default = 0
6	R/W	0	reg_little_endian : unsigned , default = 0
5	R/W	0	reg_y_rev : unsigned , default = 0
4	R/W	0	reg_x_rev : unsigned , default = 0
2:0	R/W	1	reg_pack_mode : unsigned , default = 1

MCVECRD_CTRL2 0x37e0

Bit(s)	R/W	Default	Description
31:30	R/W	0	reg_sw_rst : unsigned , default = 0
23:18	R/W	0	reg_gclk_ctrl : unsigned , default = 0
16:0	R/W	0	reg_urgent_ctrl : unsigned , default = 0

MCVECRD_SCOPE_X 0x37e1

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_x_end : unsigned , default = 0
12:0	R/W	0	reg_x_start : unsigned , default = 0

MCVECRD_SCOPE_Y 0x37e2

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_y_end : unsigned , default = 0
12:0	R/W	0	reg_y_start : unsigned , default = 0

MCVECRD_RO_STAT 0x37e3

Bit(s)	R/W	Default	Description
15:0	R.O	0	ro_reg_status : unsigned , default = 0

MCINFRD_CTRL1 0x37e4

Bit(s)	R/W	Default	Description
25:24	R/W	0	reg_sync_sel : unsigned , default = 0
23:16	R/W	0	reg_canvas_id : unsigned , default = 0
14:12	R/W	1	reg_cmd_intr_len : unsigned , default = 1
11:10	R/W	1	reg_cmd_req_size : unsigned , default = 1
9:8	R/W	2	reg_burst_len : unsigned , default = 2
7	R/W	0	reg_swap_64bit : unsigned , default = 0
6	R/W	0	reg_little_endian : unsigned , default = 0
5	R/W	0	reg_y_rev : unsigned , default = 0
4	R/W	0	reg_x_rev : unsigned , default = 0
2:0	R/W	1	reg_pack_mode : unsigned , default = 1

MCINFRD_CTRL2 0x37e5

Bit(s)	R/W	Default	Description
31:30	R/W	0	reg_sw_rst : unsigned , default = 0
23:18	R/W	0	reg_gclk_ctrl : unsigned , default = 0
16:0	R/W	0	reg_urgent_ctrl : unsigned , default = 0

MCINFRD_SCOPE_X 0x37e6

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_x_end : unsigned , default = 0
12:0	R/W	0	reg_x_start : unsigned , default = 0

MCINFRD_SCOPE_Y 0x37e7

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_y_end : unsigned , default = 0
12:0	R/W	0	reg_y_start : unsigned , default = 0

MCINFRD_RO_STAT 0x37e8

Bit(s)	R/W	Default	Description

15:0	R.O	0	ro_reg_status : unsigned , default = 0

CONTWR_X 0x37e9

Bit(s)	R/W	Default	Description
31:30	R/W	2	burst_len : unsigned , default = 2
29	R/W	0	rev_x : unsigned , default = 0
28:16	R/W	0	start_x : unsigned , default = 0
12:0	R/W	2	end_x : unsigned , default = 2cf

CONTWR_Y 0x37ea

Bit(s)	R/W	Default	Description
31:30	R/W	0	canvas_id : unsigned , default = 0
29	R/W	0	rev_y : unsigned , default = 0
28:16	R/W	0	start_y : unsigned , default = 0
12:0	R/W	0	end_y : unsigned , default = 0x1df

CONTWR_CTRL 0x37eb

Bit(s)	R/W	Default	Description
31:16	R/W	0	urgent_ctrl : unsigned , default = 0
15	R/W	0	force_wvalid : unsigned , default = 0
14	R/W	0	canvas_syncen : unsigned , default = 0
13	R/W	1	canvas_wr : unsigned , default = 1
12	R/W	0	req_en : unsigned , default = 0
10	R/W	0	clr_wrrsp : unsigned , default = 0
8	R/W	0	urgent : unsigned , default = 0
7:0	R/W	0	canvas_index : unsigned , default = 0

CONTWR_CAN_SIZE 0x37ec

Bit(s)	R/W	Default	Description
30:29	R/W	0	reg_rst : unsigned , default = 0
28:16	R/W	0	hsizem1 : unsigned , default = 0x2cf
14	R/W	0	reg_reset : unsigned , default = 0
13	R/W	0	little_endian : unsigned , default = 0
12:0	R/W	0	vsizem1 : unsigned , default = 0x1df

MTNWR_X 0x37ed

Bit(s)	R/W	Default	Description
31:30	R/W	2	burst_len : unsigned , default = 2
29	R/W	0	rev_x : unsigned , default = 0
28:16	R/W	0	start_x : unsigned , default = 0
12:0	R/W	2	end_x : unsigned , default = 2cf

MTNWR_Y 0x37ee

Bit(s)	R/W	Default	Description
31:30	R/W	0	canvas_id : unsigned , default = 0
29	R/W	0	rev_y : unsigned , default = 0
28:16	R/W	0	start_y : unsigned , default = 0
12:0	R/W	0	end_y : unsigned , default = 0x1df

MTNWR_CTRL 0x37ef

Bit(s)	R/W	Default	Description
31:16	R/W	0	urgent_ctrl : unsigned , default = 0
15	R/W	0	force_wvalid : unsigned , default = 0
14	R/W	0	canvas_syncen : unsigned , default = 0
13	R/W	1	canvas_wr : unsigned , default = 1
12	R/W	0	req_en : unsigned , default = 0
10	R/W	0	clr_wrrsp : unsigned , default = 0
8	R/W	0	urgent : unsigned , default = 0
7:0	R/W	0	canvas_index : unsigned , default = 0

MTNWR_CAN_SIZE 0x37f0

Bit(s)	R/W	Default	Description
30:29	R/W	0	reg_rst : unsigned , default = 0
28:16	R/W	0	hsizem1 : unsigned , default = 0x2cf
14	R/W	0	reg_reset : unsigned , default = 0
13	R/W	0	little_endian : unsigned , default = 0
12:0	R/W	0	vsizem1 : unsigned , default = 0x1df

MCVECWR_X 0x37f1

Bit(s)	R/W	Default	Description
31:30	R/W	2	burst_len : unsigned , default = 2
29	R/W	0	rev_x : unsigned , default = 0
28:16	R/W	0	start_x : unsigned , default = 0
12:0	R/W	2	end_x : unsigned , default = 2cf

MCVECWR_Y 0x37f2

Bit(s)	R/W	Default	Description
31:30	R/W	0	canvas_id : unsigned , default = 0
29	R/W	0	rev_y : unsigned , default = 0
28:16	R/W	0	start_y : unsigned , default = 0
12:0	R/W	0	end_y : unsigned , default = 0x1df

MCVECWR_CTRL 0x37f3

Bit(s)	R/W	Default	Description
31:16	R/W	0	urgent_ctrl : unsigned , default = 0
15	R/W	0	force_wvalid : unsigned , default = 0
14	R/W	0	canvas_syncen : unsigned , default = 0
13	R/W	1	canvas_wr : unsigned , default = 1
12	R/W	0	req_en : unsigned , default = 0
10	R/W	0	clr_wrrsp : unsigned , default = 0
8	R/W	0	urgent : unsigned , default = 0
7:0	R/W	0	canvas_index : unsigned , default = 0

MCVECWR_CAN_SIZE 0x37f4

Bit(s)	R/W	Default	Description
30:29	R/W	0	reg_rst : unsigned , default = 0
28:16	R/W	0	hsizem1 : unsigned , default = 0x2cf
14	R/W	0	reg_reset : unsigned , default = 0
13	R/W	0	little_endian : unsigned , default = 0

12:0	R/W	0	vsizem1 : unsigned , default = 0x1df

MCINFWR_X 0x37f5

Bit(s)	R/W	Default	Description
31:30	R/W	2	burst_len : unsigned , default = 2
29	R/W	0	rev_x : unsigned , default = 0
28:16	R/W	0	start_x : unsigned , default = 0
12:0	R/W	2	end_x : unsigned , default = 2cf

MCINFWR_Y 0x37f6

Bit(s)	R/W	Default	Description
31:30	R/W	0	canvas_id : unsigned , default = 0
29	R/W	0	rev_y : unsigned , default = 0
28:16	R/W	0	start_y : unsigned , default = 0
12:0	R/W	0	end_y : unsigned , default = 0x1df

MCINFWR_CTRL 0x37f7

Bit(s)	R/W	Default	Description
31:16	R/W	0	urgent_ctrl : unsigned , default = 0
15	R/W	0	force_wvalid : unsigned , default = 0
14	R/W	0	canvas_syncen : unsigned , default = 0
13	R/W	1	canvas_wr : unsigned , default = 1
12	R/W	0	req_en : unsigned , default = 0
10	R/W	0	clr_wrrsp : unsigned , default = 0
8	R/W	0	urgent : unsigned , default = 0
7:0	R/W	0	canvas_index : unsigned , default = 0

MCINFWR_CAN_SIZE 0x37f8

Bit(s)	R/W	Default	Description
30:29	R/W	0	reg_rst : unsigned , default = 0
28:16	R/W	0	hsizem1 : unsigned , default = 0x2cf
14	R/W	0	reg_reset : unsigned , default = 0
13	R/W	0	little_endian : unsigned , default = 0
12:0	R/W	0	vsizem1 : unsigned , default = 0x1df

10.2.3.5 DI_RMEM_IF0 Registers

DI_IF0_GEN_REG 0x2030

Bit(s)	R/W	Default	Description
31	R/W	0	ENABLE_FREE_CLK. 0: Gated clock for power saving 1: Free-running clock to drive logic
30	R/W	0	SW_RESET: Write 1 to this bit to generate a pulse to reset everything except registers.
29	R/W	0	RESET_ON_GO_FIELD: Define whether to reset state machines on go_field pulse. 0: No reset on go_field 1: go_field reset everything except registers
28	R/W	0	URGENT_CHROMA: Set urgent level for chroma fifo request from DDR. 0: Non urgent 1: Urgent
27	R/W	0	URGENT_LUMA: Set urgent level for luma fifo request from DDR. 0: Non urgent 1: Urgent
26	R/W	0	Chroma_end_at_last line: For chroma line, similar to luma_end_at_last_line, as below. Not used if data are stored together in one canvas.
25	R/W	0	Luma_end_at_last_line: Control whether continue outputting luma line past last line. 0: Repeat the last line or dummy pixels, after past the last line 1: Stop outputting data, once past the last line.
24-19	R/W	4	Hold_lines: After go_field, the number of lines to hold before the module is enabled.
18	R/W	0	LAST_LINE: This bit controls whether we simply repeat the last line or we push dummy pixels. '1' tells the state-machines to repeat the last line using the dummy pixels defined in the register below. '0' indicates that the state-machine should re-read the last line of real data.
17	R	0	Busy status of the state-machines. '1' = busy, '0' = idle
16	R/W	0	DEMUX_MODE: 0 = 4:2:2, 1 = RGB (24-bit). This value is used to control the demuxing logic when the picture is stored together. When a picture is stored together, the data is read into a single FIFO (the Y FIFO) and must be demultiplexed into the "drain" outputs. In the case of 4:2:2 the data is assumed to be stored in memory in 16-bit chunks: <YCb><YCr><YCb><YCr>,… the Y, Cb and Cr 8-bit values are pulled from the single Y-FIFO and sent out in pairs. This value is only valid when the picture is stored together. If the picture is separated into different canvases, then this bit field is ignored.

Bit(s)	R/W	Default	Description
15-14	R/W	0	BYTES_PER_PIXEL: This value is used to determine how many bytes are associated with each pixel. 0: This value should be used if the image is stored separately (e.g. RGB or Y, Cb, Cr). 1: This value should be used if the data is 4:2:2 data stored together. In this case each pixel , YCb or YCr, is 16-bits (two bytes). 2: This value should be used if the RGB (24-bit) data is stored together. 3: reserved for future use (alpha RGB).
13-12	R/W	0	DDR_BURST_SIZE_CR: This value is used to control the DDR burst request size for the Cr FIFO. 0: Maximum burst = 24 64-bit values 1: Maximum burst = 32 64-bit values 2: Maximum burst = 48 64-bit values 3: Maximum burst = 64 64-bit values
11-10	R/W	0	DDR_BURST_SIZE_CB: This value is used to control the DDR burst request size for the Cb FIFO. 0: Maximum burst = 24 64-bit values 1: Maximum burst = 32 64-bit values 2: Maximum burst = 48 64-bit values 3: Maximum burst = 64 64-bit values
9-8	R/W	0	DDR_BURST_SIZE_Y: This value is used to control the DDR burst request size for the Y FIFO. 0: Maximum burst = 24 64-bit values 1: Maximum burst = 32 64-bit values 2: Maximum burst = 48 64-bit values 3: Maximum burst = 64 64-bit values
7	R/W	0	MANUAL_START_FRAME: non-latching bit that can be used to simulate the go_field signal for simulation.
6	R/W	0	CHRO_RPT_LASTL_CTRL: This bit controls whether to allow VPP's chroma-repeat request. 0: Chroma-repeat pulses from VPP are ignored 1: Chroma-repeat pulses from VPP are used.
5	R/W	0	Unused
4	R/W	0	LITTLE_ENDIAN: This bit defines the endianess of the memory data . 0: Pixel data are stored big-endian in memory 1: Pixel data are stored little-endian in memory
3	R/W	0	Chroma_hz_avg: For chroma line output control, similar to luma_hz_avg, as below. Not used if data are stored together in one canvas.
2	R/W	0	Luma_hz_avg: Enable output half amount of data per line to save bandwidth. 0: Output every pixel per line 1: Output half line, each data averaged between every 2 pixels Note: For 4:2:2 mode data stored together in one canvas, only do averaging over luma data.

Bit(s)	R/W	Default	Description
1	R/W	0	SEPARATE_EN: Set this bit to 1 if the image is in separate canvas locations.
0	R/W	0	ENABLE: This bit is set to 1 to enable the FIFOs and other logic. This bit can be set to 0 to cleanup and put the logic into an IDLE state.

DI_IF0_CANVAS0 – Picture 0 0x2031

Bit(s)	R/W	Default	Description
31-24	R/W	0	unused
23-16	R/W	0	CANVAS0_ADDR2: Canvas table address for picture 0 for component 2 (Cr FIFO). This value is ignored when the picture is stored together
15-8	R/W	0	CANVAS0_ADDR1: Canvas table address for picture 0 for component 1 (Cb FIFO). This value is ignored when the picture is stored together
7-0	R/W	0	CANVAS0_ADDR0: Canvas table address for picture 0 for component 0 (Y FIFO).

DI_IF0_LUMA_X0 – Picture 0 0x2032

Bit(s)	R/W	Default	Description
31	R/W	0	Unused
30-16	R/W	0	LUMA_X_END0: Picture 0, luma X end value
15	R/W	0	Unused
14-0	R/W	0	LUMA_X_START0: Picture 0, luma X start value

DI_IF0_LUMA_Y0 – Picture 0 0x2033

Bit(s)	R/W	Default	Description
31-29	R/W	0	Unused
28-16	R/W	0	LUMA_Y_END0: Picture 0, luma Y end value
15-13	R/W	0	Unused
12-0	R/W	0	LUMA_Y_START0: Picture 0, luma Y start value

DI_IF0_CHROMA_X0 – Picture 0 0x2034

Bit(s)	R/W	Default	Description
31	R/W	0	Unused
30-16	R/W	0	CHROMA_X_END0: Picture 0, chroma X end value. This value is only used when the picture is not stored tegether.
15	R/W	0	Unused
14-0	R/W	0	CHROMA _X_START0: Picture 0, chroma X start value. This value is only used when the picture is not stored tegether.

DI_IF0_ CHROMA_Y0 – Picture 0 0x2035

Bit(s)	R/W	Default	Description
31-29	R/W	0	Unused
28-16	R/W	0	CHROMA _Y_END0: Picture 0, chroma Y end value. This value is only used when the picture is not stored together.
15-13	R/W	0	Unused
12-0	R/W	0	CHROMA _Y_START0: Picture 0, chroma Y start value. This value is only used when the picture is not stored together.

DI_IF0_REPEAT_LOOP – Pictures 0 and 1 0x2036

Bit(s)	R/W	Default	Description
31-24	R/W	0	CHROMA_RPT_LOOP1: Repeat loop for Picture 1. Bits[6:4] = start loop pointer, bits [2:0] = end loop pointer. Bits [7] and [3] are ignored.
23-16	R/W	0	LUMA_RPT_LOOP1: Repeat loop for Picture 1. Bits[6:4] = start loop pointer, bits [2:0] = end loop pointer. Bits [7] and [3] are ignored.
15-8	R/W	0	CHROMA_RPT_LOOP0: Repeat loop for Picture 0. Bits[6:4] = start loop pointer, bits [2:0] = end loop pointer. Bits [7] and [3] are ignored.
7-0	R/W	0	LUMA_RPT_LOOP0: Repeat loop for Picture 0. Bits[6:4] = start loop pointer, bits [2:0] = end loop pointer. Bits [7] and [3] are ignored.

DI_IF0_LUMA0_RPT_PAT – Picture 0 LUMA repeat pattern 0x2037

Bit(s)	R/W	Default	Description
31-0	R/W	0	Luma repeat/skip pattern for picture 0

Bits	Pattern Index	Pattern description
31-28	7	Repeat/skip pattern: Bit[3] = 0 indicates repeat. Bit[3] = 1 indicates either skip, or output this line and then skip. How to interpret this bit depends on the value of the previous pattern’ s Bit[3]. If previous Bit[3]=0, then skip; If previous Bit[3]=1, then output this line and then skip. Bits[2:0] indicate the skip / repeat count. Blow is an example of consecutive patterns, the start line is line 0: {0010} Repeat this line (line 0) two more times for a total of three line reads. Proceed to next line (line 1). {0000} Don’ t repeat this line (line 1). This line will be read just once. Proceed to next line (line 2). {1000} Skip one line (line 2) to get to the next line (line 3). The skip implies that the next line (line 3) should be read at least once.

		{1011} Read this line (line 3) once, and then skip the next four lines to get to the next line (line 8). The skip implies that the next line (line 8) should be read at least once. {0100} Repeat this line (line 8) four more times for a total of five line read. Proceed to next line (line 9). {1001} Skip two lines to get to the next line (line 11). The skip implies that the next line (line 11) should be read at least once.
27-24	6	See pattern definition above.
23-20	5	See pattern definition above.
19-16	4	See pattern definition above.
15-12	3	See pattern definition above.
11-8	2	See pattern definition above.
7-4	1	See pattern definition above.
3-0	0	See pattern definition above.

DI_IF0_CHROMA0_RPT_PAT – Picture 0 CHROMA repeat pattern 0x2038

Bit(s)	R/W	Default	Description
31-0	R/W	0	Chroma repeat/skip pattern for picture 0. See picture 0 luma pattern for description. This value is only used when the picture is not stored together.

DI_IF0_DUMMY_PIXEL 0x2039

Bit(s)	R/W	Default	Description
31-24	R/W	0	Y or R dummy pixel value
23-16	R/W	0	Cb or G dummy pixel value
15-8	R/W	0	Cr or B dummy pixel value
7-0	R/W	0	unused

DI_IF0_LUMA_FIFO_SIZE 0x203A

Bit(s)	R/W	Default	Description
8-0	W/R	128	fifo size

DI_IF0_RANGE_MAP_Y 0x203B

31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_IF0_RANGE_MAP_CB 0x203C

31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_IF0_RANGE_MAP_CR 0x203D

31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

Output data range conversion function:

Y[n] = clip(Round(((Y[n]+DIN_OFFSET) * RANGE_MAP_COEF) / (1<<RANGE_MAP_SR)) +DOUT_OFFSET);

To perform VC-1 range reduction, set the following:

DIN_OFFSET = 0x180 = -128;

RANGE_MAP_COEF = RANGE_MAPY + 9

RANGE_MAP_SR = 3

DOUT_OFFSET = 0x080 = 128

To get the equivalent function:

Y[n] = clip((((Y[n]-128) * (RANGE_MAPY + 9) + 4) >> 3) +128);

Bit(s)	R/W	Default	Description
31-23	R/W	0	DIN_OFFSET
22-15	R/W	0	RANGE_MAP_COEF
14	R/W	0	unused
13-10	R/W	0	RANGE_MAP_SR
9-1	R/W	0	DOUT_OFFSET
0	R/W	0	RANGE_MAP_EN

DI_IF0_GEN_REG2 0x203E

Bit(s)	R/W	Default	Description

31-2	R/W	0	unused
1-0	R/W	0	COLOR_MAP: Define color map for NV12 or NV21 mode. Only applicable when VD1_IF0_GEN_REG.SEPARATE_EN = 1. 0: NOT NV12 or NV21; 1: NV12 (CbCr); 2: NV21 (CrCb).

DI_IF0_FMT_CTRL 0x203F

Bit(s)	R/W	Default	Description
31	R/W	0	gate_clk_en. 0=No clock gating, free-running; 1=Enable clock gating for power saving.
30	R/W	0	soft_rst. If true, reset formatters.
29	R/W	0	unused
28	R/W	0	if true, horizontal formatter use repeating to generete pixel, otherwise use bilinear interpolation
27-24	R/W	0	horizontal formatter initial phase
23	R/W	0	horizontal formatter repeat pixel 0 enable
22-21	R/W	0	horizontal Y/C ratio, 00: 1:1, 01: 2:1, 10: 4:1
20	R/W	0	horizontal formatter enable
19	R/W	0	if true, always use phase0 while vertical formater, meaning always repeat data, no interpolation
18	R/W	0	if true, disable vertical formatter chroma repeat last line
17	R/W	0	veritcal formatter dont need repeat line on phase0, 1: enable, 0: disable
16	R/W	0	veritcal formatter repeat line 0 enable
15-12	R/W	0	vertical formatter skip line num at the beginning
11-8	R/W	0	vertical formatter initial phase
7-1	R/W	0	vertical formatter phase step (3.4)
0	R/W	0	vertical formatter enable

DI_IF0_FMT_W 0x2040

Bit(s)	R/W	Default	Description
27-16	R/W	0	horizontal formatter width
11-0	R/W	0	vertical formatter width

10.2.3.6 VIUB Registers (slow clock)

DI_IF2_GEN_REG 0x2010

Same as DI_IF1_GEN_REG

DI_IF2_CANVAS0 0x2011

Same as DI_IF1_CANVAS0

DI_IF2_LUMA_X0 0x2012

Same as DI_IF1_LUMA_X0

DI_IF2_LUMA_Y0 0x2013

Same as DI_IF1_LUMA_Y0

DI_IF2_CHROMA_X0 0x2014

Same as DI_IF1_CHROMA_X0

DI_IF2_CHROMA_Y0 0x2015

Same as DI_IF1_CHROMA_Y0

DI_IF2_RPT_LOOP 0x2016

Same as DI_IF1_RPT_LOOP

DI_IF2_LUMA0_RPT_PAT 0x2017

Same as DI_IF1_LUMA0_RPT_PAT

DI_IF2_CHROMA0_RPT_PAT 0x2018

Same as DI_IF1_CHROMA0_RPT_PAT

DI_IF2_DUMMY_PIXEL 0x2019

Same as DI_IF1_DUMMY_PIXEL

DI_IF2_LUMA_FIFO_SIZE 0x201a

Same as DI_IF1_LUMA_FIFO_SIZE

DI_IF2_RANGE_MAP_Y 0x201b

Same as DI_IF1_RANGE_MAP_Y

DI_IF2_RANGE_MAP_CB 0x201c

Same as DI_IF1_RANGE_MAP_CB

DI_IF2_RANGE_MAP_CR 0x201d

Same as DI_IF1_RANGE_MAP_CR

DI_IF2_GEN_REG2 0x201e

Same as DI_IF1_GEN_REG2

DI_IF2_FMT_CTRL 0x201f

Same as DI_IF1_FMT_CTRL

DI_IF2_FMT_W 0x2020

Same as DI_IF1_FMT_W

DI_IF2_URGENT_CTRL 0x2021

Same as DI_IF1_URGENT_CTRL

DI_IF1_GEN_REG3 0x20a7

Bit(s)	R/W	Default	Description
11-10	R/W	0	cntl_dbg_mode

9-8	R/W	0	cntl_bits_mode : 0->8bit 1->10bit 422 2->10bit 444
6-4	R/W	3	cntl_blk_len
2-1	R/W	1	cntl_burst_len
0	R/W	1	cntl_64bit_rev

DI_IF2_GEN_REG3 0x2022

Bit(s)	R/W	Default	Description
11-10	R/W	0	cntl_dbg_mode
9-8	R/W	0	cntl_bits_mode : 0->8bit 1->10bit 422 2->10bit 444
6-4	R/W	3	cntl_blk_len
2-1	R/W	1	cntl_burst_len
0	R/W	1	cntl_64bit_rev

DI_INP_GEN_REG3 0x20a8

Bit(s)	R/W	Default	Description
11-10	R/W	0	cntl_dbg_mode
9-8	R/W	0	cntl_bits_mode : 0->8bit 1->10bit 422 2->10bit 444
6-4	R/W	3	cntl_blk_len
2-1	R/W	1	cntl_burst_len
0	R/W	1	cntl_64bit_rev

DI_MEM_GEN_REG3 0x20a9

Bit(s)	R/W	Default	Description
11-10	R/W	0	cntl_dbg_mode
9-8	R/W	0	cntl_bits_mode : 0->8bit 1->10bit 422 2->10bit 444
6-4	R/W	3	cntl_blk_len
2-1	R/W	1	cntl_burst_len
0	R/W	1	cntl_64bit_rev

DI_CHAN2_GEN_REG3 0x20aa

Bit(s)	R/W	Default	Description
11-10	R/W	0	cntl_dbg_mode
9-8	R/W	0	cntl_bits_mode : 0->8bit 1->10bit 422 2->10bit 444
6-4	R/W	3	cntl_blk_len

2-1	R/W	1	cntl_burst_len
0	R/W	1	cntl_64bit_rev

10.2.3.7 De-Interlace Registers

DI_IF1_GEN_REG 0x17E8

Bit(s)	R/W	Default	Description
31	W/R	0	enable free clk
30	W/R	0	sw reset : pulse bit
29	W/R	0	reset on go field
28	W/R	0	urgent chroma
27	W/R	0	urgent luma
26	W/R	0	chroma end at last line : 0 = read last line or push dummy after last line; 1 = stop read after last line
25	W/R	0	luma end at last line : 0 = read last line or push dummy after last line; 1 = stop read after last line
24-19	W/R	4	hold line[5:0], see GEN_REG2[6]
18	W/R	1	last line mode: 0 = read last line; 1 = push fixed value
16	W/R	0	demux mode: 0 = 4:2:2 demux; 1 = RGB demuxing from a single FIFO
15-14	W/R	0	bytes per pixel : 0= 1byte per pixel; 1 = 2 bytes per pixel; 2 = 3bytes per pixel
13-12	W/R	0	burst size cr: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
11-10	W/R	0	burst size cb: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
9-8	W/R	0	burst size y: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
7	W/R	0	start frame manual : pulse bit
6	W/R	0	chroma repeat last1
5	W/R	0	Reserved
4	W/R	0	little endian: 0=Pixels are big-endian in memory; 1=Pixel are little-endian in memory
3	W/R	0	chroma hz avg: 0= output pixel by pixel per line; 1= output half line ,average between every 2 pixels
2	W/R	0	luma_hz_avg: 0= output pixel by pixel per line; 1= output half line ,average between every 2 pixels
1	W/R	0	separate_en: Set to 1 to use 3 separate FIFO’s
0	W/R	0	enable

DI_IF1_GEN_REG2 0x1790

Bit(s)	R/W	Default	Description
25-14	W/R	0	shift pat cr
17-16	W/R	0	shift pat cb
9-8	W/R	0	shift pat y
6	W/R	0	hold_lines[6]
3	W/R	0	y_rev: X read direction: 0=default ,normal read; 1=reverse read
2	W/R	0	x_rev: Y read direction: 0=default ,normal read; 1=reverse read
1-0	W/R	0	color map: 0=default color map as defined by “bytes per pixel”; 1=NV12(CbCr); 2=NV21(CrCb)

DI_IF1_CANVAS0 0x17E9

Bit(s)	R/W	Default	Description
31	W/R	0	canvas addr syncen
23-16	W/R	0	canvas addr2
15-8	W/R	0	canvas addr1
7-0	W/R	0	canvas addr0

DI_IF1_LUMA_X0 0x17EA

Bit(s)	R/W	Default	Description
30-16	W/R	0	luma_x_end
14-0	W/R	0	luma_x_start

DI_IF1_LUMA_Y0 0x17EB

Bit(s)	R/W	Default	Description
28-16	W/R	0	luma_y_end
12-0	W/R	0	luma_y_start

DI_IF1_CHROMA_X0 0x17EC

Bit(s)	R/W	Default	Description
30-16	W/R	0	chroma_x_end
14-0	W/R	0	chroma_x_start

DI_IF1_CHROMA_Y0 0x17ED

Bit(s)	R/W	Default	Description
28-16	W/R	0	chroma_y_end
12-0	W/R	0	chroma_y_start

DI_IF1_RPT_LOOP 0x17EE

Bit(s)	R/W	Default	Description
15-8	W/R	0	chroma repeat loop
7-0	W/R	0	luma repeat loop

DI_IF1_LUMA0_RPT_PAT 0x17EF

Bit(s)	R/W	Default	Description
31-0	W/R	0	luma repeat pattern

DI_IF1_CHROMA0_RPT_PAT 0x17F0

Bit(s)	R/W	Default	Description
7-0	W/R	0	chroma repeat loop

DI_IF1_DUMMY_PIXEL 0x17F1

Bit(s)	R/W	Default	Description
31-0	W/R	0x808000	dummy pixel

DI_IF1_LUMA_FIFO_SIZE 0x17F2

Bit(s)	R/W	Default	Description
8-0	W/R	128	fifo size

DI_IF1_RANGE_MAP_Y 0x17FC

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_IF1_RANGE_MAP_CB 0x17FD

Bit(s)	R/W	Default	Description

31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_IF1_RANGE_MAP_CR 0x17FE

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_IF1_URGENT_CTRL 0x17A3

Bit(s)	R/W	Default	Description
13-16	W/R	0	urgent_ctrl_luma: bit 15: auto urgent_en bit 14: urgent_wr bit 7-4: up_threshold bit 3-0: down_threshold
15-0	W/R	0	urgent_ctrl_chroma: bit 15: auto urgent_en bit 14: urgent_wr bit 7-4: up_threshold bit 3-0: down_threshold

DI_IF1_FMT_CTRL 0x17F3

Bit(s)	R/W	Default	Description
31	R/W	0	gate_clk_en. 0=No clock gating, free-running; 1=Enable clock gating for power saving.
30	R/W	0	soft_rst. If true, reset formatters.
29	R/W	0	unused
28	R/W	0	if true, horizontal formatter use repeating to generete pixel, otherwise use bilinear interpolation
27-24	R/W	0	horizontal formatter initial phase
23	R/W	0	horizontal formatter repeat pixel 0 enable

Bit(s)	R/W	Default	Description
22-21	R/W	0	horizontal Y/C ratio, 00: 1:1, 01: 2:1, 10: 4:1
20	R/W	0	horizontal formatter enable
19	R/W	0	if true, always use phase0 while vertical formater, meaning always repeat data, no interpolation
18	R/W	0	if true, disable vertical formatter chroma repeat last line
17	R/W	0	veritcal formatter dont need repeat line on phase0, 1: enable, 0: disable
16	R/W	0	veritcal formatter repeat line 0 enable
15-12	R/W	0	vertical formatter skip line num at the beginning
11-8	R/W	0	vertical formatter initial phase
7-1	R/W	0	vertical formatter phase step (3.4)
0	R/W	0	vertical formatter enable

DI_IF1_FMT_W 0x17F4

Bit(s)	R/W	Default	Description
27-16	R/W	0	horizontal formatter width
12-0	R/W	0	vertical formatter width

DI_INP_GEN_REG 0x17CE

Bit(s)	R/W	Default	Description
31	W/R	0	enable free clk
30	W/R	0	sw reset : pulse bit
29	W/R	0	reset on go field
28	W/R	0	urgent chroma
27	W/R	0	urgent luma
26	W/R	0	chroma end at last line : 0 = read last line or push dummy after last line; 1 = stop read after last line
25	W/R	0	luma end at last line : 0 = read last line or push dummy after last line; 1 = stop read after last line
24-19	W/R	4	hold line[5:0], see GEN_REG2[6]
18	W/R	1	last line mode: 0 = read last line; 1 = push fixed value
16	W/R	0	demux mode: 0 = 4:2:2 demux; 1 = RGB demuxing from a single FIFO
15-14	W/R	0	bytes per pixel : 0= 1byte per pixel; 1 = 2 bytes per pixel; 2 = 3bytes per pixel

Bit(s)	R/W	Default	Description
13-12	W/R	0	burst size cr: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
11-10	W/R	0	burst size cb: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
9-8	W/R	0	burst size y: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
7	W/R	0	start frame manual : pulse bit
6	W/R	0	chroma repeat last1
5	W/R	0	Reserved
4	W/R	0	little endian: 0=Pixels are big-endian in memory; 1=Pixel are little-endian in memory
3	W/R	0	chroma hz avg: 0= output pixel by pixel per line; 1= output half line ,average between every 2 pixels
2	W/R	0	luma_hz_avg: 0= output pixel by pixel per line; 1= output half line ,average between every 2 pixels
1	W/R	0	separate_en: Set to 1 to use 3 separate FIFO's
0	W/R	0	enable

DI_INP_GEN_REG2 0x1791

Bit(s)	R/W	Default	Description
25-14	W/R	0	shift pat cr
17-16	W/R	0	shift pat cb
9-8	W/R	0	shift pat y
6	W/R	0	hold_lines[6]
3	W/R	0	y_rev: X read direction: 0=default ,normal read; 1=reverse read
2	W/R	0	x_rev: Y read direction: 0=default ,normal read; 1=reverse read
1-0	W/R	0	color map: 0=default color map as defined by "bytes per pixel"; 1=NV12(CbCr); 2=NV21(CrCb)

DI_INP_CANVAS0 0x17CF

Bit(s)	R/W	Default	Description
31	W/R	0	canvas addr syncen
23-16	W/R	0	canvas addr2
15-8	W/R	0	canvas addr1
7-0	W/R	0	canvas addr0

DI_INP_LUMA_X0 0x17D0

Bit(s)	R/W	Default	Description
30-16	W/R	0	luma_x_end
14-0	W/R	0	luma_x_start

DI_INP_LUMA_Y0 0x17D1

Bit(s)	R/W	Default	Description
28-16	W/R	0	luma_y_end
12-0	W/R	0	luma_y_start

DI_INP_CHROMA_X0 0x17D2

Bit(s)	R/W	Default	Description
30-16	W/R	0	chroma_x_end
14-0	W/R	0	chroma_x_start

DI_INP_CHROMA_Y0 0x17D3

Bit(s)	R/W	Default	Description
28-16	W/R	0	chroma_y_end
12-0	W/R	0	chroma_y_start

DI_INP_RPT_LOOP 0x17D4

Bit(s)	R/W	Default	Description
15-8	W/R	0	chroma repeat loop
7-0	W/R	0	luma repeat loop

DI_INP_LUMA0_RPT_PAT 0x17D5

Bit(s)	R/W	Default	Description
31-0	W/R	0	luma repeat pattern

DI_INP_CHROMA0_RPT_PAT 0x17D6

Bit(s)	R/W	Default	Description
7-0	W/R	0	chroma repeat loop

DI_INP_DUMMY_PIXEL 0x17D7

Bit(s)	R/W	Default	Description
31-0	W/R	0x808000	dummy pixel

DI_INP_LUMA_FIFO_SIZE 0x17D8

Bit(s)	R/W	Default	Description
8-0	W/R	128	fifo size

DI_INP_RANGE_MAP_Y 0x17BA

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_INP_RANGE_MAP_CB 0x17BB

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_INP_RANGE_MAP_CR 0x17BC

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_INP_URGENT_CTRL 0x17A4

Bit(s)	R/W	Default	Description
13-16	W/R	0	urgent_ctrl_luma: bit 15: auto urgent_en bit 14: urgent_wr bit 7-4: up_threshold bit 3-0: down_threshold

15-0	W/R	0	urgent_ctrl_chroma: bit 15: auto urgent_en bit 14: urgent_wr bit 7-4: up_threshold bit 3-0: down_threshold

DI_INP_FMT_CTRL 0x17D9

Bit(s)	R/W	Default	Description
31	R/W	0	gate_clk_en. 0=No clock gating, free-running; 1=Enable clock gating for power saving.
30	R/W	0	soft_rst. If true, reset formatters.
29	R/W	0	unused
28	R/W	0	if true, horizontal formatter use repeating to generete pixel, otherwise use bilinear interpolation
27-24	R/W	0	horizontal formatter initial phase
23	R/W	0	horizontal formatter repeat pixel 0 enable
22-21	R/W	0	horizontal Y/C ratio, 00: 1:1, 01: 2:1, 10: 4:1
20	R/W	0	horizontal formatter enable
19	R/W	0	if true, always use phase0 while vertical formater, meaning always repeat data, no interpolation
18	R/W	0	if true, disable vertical formatter chroma repeat last line
17	R/W	0	veritcal formatter dont need repeat line on phase0, 1: enable, 0: disable
16	R/W	0	veritcal formatter repeat line 0 enable
15-12	R/W	0	vertical formatter skip line num at the beginning
11-8	R/W	0	vertical formatter initial phase
7-1	R/W	0	vertical formatter phase step (3.4)
0	R/W	0	vertical formatter enable

DI_INP_FMT_W 0x17DA

Bit(s)	R/W	Default	Description
27-16	R/W	0	horizontal formatter width
12-0	R/W	0	vertical formatter width

DI_MEM_GEN_REG 0x17DB

Bit(s)	R/W	Default	Description
31	W/R	0	enable free clk

Bit(s)	R/W	Default	Description
30	W/R	0	sw reset : pulse bit
29	W/R	0	reset on go field
28	W/R	0	urgent chroma
27	W/R	0	urgent luma
26	W/R	0	chroma end at last line : 0 = read last line or push dummy after last line; 1 = stop read after last line
25	W/R	0	luma end at last line : 0 = read last line or push dummy after last line; 1 = stop read after last line
24-19	W/R	4	hold line[5:0], see GEN_REG2[6]
18	W/R	1	last line mode: 0 = read last line; 1 = push fixed value
16	W/R	0	demux mode: 0 = 4:2:2 demux; 1 = RGB demuxing from a single FIFO
15-14	W/R	0	bytes per pixel : 0= 1byte per pixel; 1 = 2 bytes per pixel; 2 = 3bytes per pixel
13-12	W/R	0	burst size cr: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
11-10	W/R	0	burst size cb: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
9-8	W/R	0	burst size y: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
7	W/R	0	start frame manual : pulse bit
6	W/R	0	chroma repeat last1
5	W/R	0	Reserved
4	W/R	0	little endian: 0=Pixels are big-endian in memory; 1=Pixel are little-endian in memory
3	W/R	0	chroma hz avg: 0= output pixel by pixel per line; 1= output half line ,average between every 2 pixels
2	W/R	0	luma_hz_avg: 0= output pixel by pixel per line; 1= output half line ,average between every 2 pixels
1	W/R	0	separate_en: Set to 1 to use 3 separate FIFO’s
0	W/R	0	enable

DI_MEM_GEN_REG2 0x1792

Bit(s)	R/W	Default	Description
25-14	W/R	0	shift pat cr
17-16	W/R	0	shift pat cb
9-8	W/R	0	shift pat y
6	W/R	0	hold_lines[6]

3	W/R	0	y_rev: X read direction: 0=default ,normal read; 1=reverse read
2	W/R	0	x_rev: Y read direction: 0=default ,normal read; 1=reverse read
1-0	W/R	0	color map: 0=default color map as defined by “bytes per pixel”; 1=NV12(CbCr); 2=NV21(CrCb)

DI_MEM_CANVAS0 0x17DC

Bit(s)	R/W	Default	Description
31	W/R	0	canvas addr syncen
23-16	W/R	0	canvas addr2
15-8	W/R	0	canvas addr1
7-0	W/R	0	canvas addr0

DI_MEM_LUMA_X0 0x17DD

Bit(s)	R/W	Default	Description
30-16	W/R	0	luma_x_end
14-0	W/R	0	luma_x_start

DI_MEM_LUMA_Y0 0x17DE

Bit(s)	R/W	Default	Description
28-16	W/R	0	luma_y_end
12-0	W/R	0	luma_y_start

DI_MEM_CHROMA_X0 0x17DF

Bit(s)	R/W	Default	Description
30-16	W/R	0	chroma_x_end
14-0	W/R	0	chroma_x_start

DI_MEM_CHROMA_Y0 0x17E0

Bit(s)	R/W	Default	Description
28-16	W/R	0	chroma_y_end
12-0	W/R	0	chroma_y_start

DI_MEM_RPT_LOOP 0x17E1

Bit(s)	R/W	Default	Description
15-8	W/R	0	chroma repeat loop

7-0	W/R	0	luma repeat loop

DI_MEM_LUMA0_RPT_PAT 0x17E2

Bit(s)	R/W	Default	Description
31-0	W/R	0	luma repeat pattern

DI_MEM_CHROMA0_RPT_PAT 0x17E3

Bit(s)	R/W	Default	Description
7-0	W/R	0	chroma repeat loop

DI_MEM_DUMMY_PIXEL 0x17E4

Bit(s)	R/W	Default	Description
31-0	W/R	0x808000	dummy pixel

DI_MEM_LUMA_FIFO_SIZE 0x17E5

Bit(s)	R/W	Default	Description
8-0	W/R	128	fifo size

DI_MEM_RANGE_MAP_Y 0x17BD

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_MEM_RANGE_MAP_CB 0x17BE

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_MEM_RANGE_MAP_CR 0x17BF

Bit(s)	R/W	Default	Description

31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_MEM_URGENT_CTRL 0x17A5

Bit(s)	R/W	Default	Description
13-16	W/R	0	urgent_ctrl_luma: bit 15: auto urgent_en bit 14: urgent_wr bit 7-4: up_threshold bit 3-0: down_threshold
15-0	W/R	0	urgent_ctrl_chroma: bit 15: auto urgent_en bit 14: urgent_wr bit 7-4: up_threshold bit 3-0: down_threshold

DI_MEM_FMT_CTRL 0x17E6

Bit(s)	R/W	Default	Description
31	R/W	0	gate_clk_en. 0=No clock gating, free-running; 1=Enable clock gating for power saving.
30	R/W	0	soft_rst. If true, reset formatters.
29	R/W	0	unused
28	R/W	0	if true, horizontal formatter use repeating to generete pixel, otherwise use bilinear interpolation
27-24	R/W	0	horizontal formatter initial phase
23	R/W	0	horizontal formatter repeat pixel 0 enable
22-21	R/W	0	horizontal Y/C ratio, 00: 1:1, 01: 2:1, 10: 4:1
20	R/W	0	horizontal formatter enable
19	R/W	0	if true, always use phase0 while vertical formater, meaning always repeat data, no interpolation
18	R/W	0	if true, disable vertical formatter chroma repeat last line
17	R/W	0	veritcal formatter dont need repeat line on phase0, 1: enable, 0: disable
16	R/W	0	veritcal formatter repeat line 0 enable

15-12	R/W	0	vertical formatter skip line num at the beginning
11-8	R/W	0	vertical formatter initial phase
7-1	R/W	0	vertical formatter phase step (3.4)
0	R/W	0	vertical formatter enable

DI_MEM_FMT_W 0x17E7

Bit(s)	R/W	Default	Description
27-16	R/W	0	horizontal formatter width
12-0	R/W	0	vertical formatter width

DI_CHAN2_GEN_REG 0x17F5

Bit(s)	R/W	Default	Description
31	W/R	0	enable free clk
30	W/R	0	sw reset : pulse bit
29	W/R	0	reset on go field
28	W/R	0	urgent chroma
27	W/R	0	urgent luma
26	W/R	0	chroma end at last line : 0 = read last line or push dummy after last line; 1 = stop read after last line
25	W/R	0	luma end at last line : 0 = read last line or push dummy after last line; 1 = stop read after last line
24-19	W/R	4	hold line[5:0], see GEN_REG2[6]
18	W/R	1	last line mode: 0 = read last line; 1 = push fixed value
16	W/R	0	demux mode: 0 = 4:2:2 demux; 1 = RGB demuxing from a single FIFO
15-14	W/R	0	bytes per pixel : 0= 1byte per pixel; 1 = 2 bytes per pixel; 2 = 3bytes per pixel
13-12	W/R	0	burst size cr: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
11-10	W/R	0	burst size cb: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
9-8	W/R	0	burst size y: 0 = 24x64; 1 = 32x64; 2 = 48x64; 3 = 64x64
7	W/R	0	start frame manual : pulse bit
6	W/R	0	chroma repeat last1
5	W/R	0	Reserved
4	W/R	0	little endian: 0=Pixels are big-endian in memory; 1=Pixel are little-endian in memory

Bit(s)	R/W	Default	Description
3	W/R	0	chroma hz avg: 0= output pixel by pixel per line; 1= output half line ,average between every 2 pixels
2	W/R	0	luma_hz_avg: 0= output pixel by pixel per line; 1= output half line ,average between every 2 pixels
1	W/R	0	separate_en: Set to 1 to use 3 separate FIFO's
0	W/R	0	enable

DI_CHAN2_GEN_REG2 0x17B7

Bit(s)	R/W	Default	Description
25-14	W/R	0	shift pat cr
17-16	W/R	0	shift pat cb
9-8	W/R	0	shift pat y
6	W/R	0	hold_lines[6]
3	W/R	0	y_rev: X read direction: 0=default ,normal read; 1=reverse read
2	W/R	0	x_rev: Y read direction: 0=default ,normal read; 1=reverse read
1-0	W/R	0	color map: 0=default color map as defined by "bytes per pixel"; 1=NV12(CbCr); 2=NV21(CrCb)

DI_CHAN2_CANVAS0 0x17F6

Bit(s)	R/W	Default	Description
31	W/R	0	canvas addr syncen
23-16	W/R	0	canvas addr2
15-8	W/R	0	canvas addr1
7-0	W/R	0	canvas addr0

DI_CHAN2_LUMA_X0 0x17F7

Bit(s)	R/W	Default	Description
30-16	W/R	0	luma_x_end
14-0	W/R	0	luma_x_start

DI_CHAN2_LUMA_Y0 0x17F8

Bit(s)	R/W	Default	Description
28-16	W/R	0	luma_y_end
12-0	W/R	0	luma_y_start

DI_CHAN2_CHROMA_X0 0x17F9

Bit(s)	R/W	Default	Description
30-16	W/R	0	chroma_x_end
14-0	W/R	0	chroma_x_start

DI_CHAN2_CHROMA_Y0 0x17FA

Bit(s)	R/W	Default	Description
28-16	W/R	0	chroma_y_end
12-0	W/R	0	chroma_y_start

DI_CHAN2_RPT_LOOP 0x17FB

Bit(s)	R/W	Default	Description
15-8	W/R	0	chroma repeat loop
7-0	W/R	0	luma repeat loop

DI_CHAN2_LUMA0_RPT_PAT 0x17B0

Bit(s)	R/W	Default	Description
31-0	W/R	0	luma repeat pattern

DI_CHAN2_CHROMA0_RPT_PAT 0x17B1

Bit(s)	R/W	Default	Description
7-0	W/R	0	chroma repeat loop

DI_CHAN2_DUMMY_PIXEL 0x17B2

Bit(s)	R/W	Default	Description
31-0	W/R	0x808000	dummy pixel

DI_CHAN2_LUMA_FIFO_SIZE 0x17B3

Bit(s)	R/W	Default	Description
8-0	W/R	128	fifo size

DI_CHAN2_RANGE_MAP_Y 0x17B4

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef

Bit(s)	R/W	Default	Description
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_CHAN2_RANGE_MAP_CB 0x17B5

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_CHAN2_RANGE_MAP_CR 0x17B6

Bit(s)	R/W	Default	Description
31-23	W/R	0	din offset
22-15	W/R	0	range map coef
13-10	W/R	0	range map din offset mult
9-1	W/R	0	dout offset
0	W/R	0	range map enable

DI_CHAN2_URGENT_CTRL 0x17A6

Bit(s)	R/W	Default	Description
13-16	W/R	0	urgent_ctrl_luma: bit 15: auto urgent_en bit 14: urgent_wr bit 7-4: up_threshold bit 3-0: down_threshold
15-0	W/R	0	urgent_ctrl_chroma: bit 15: auto urgent_en bit 14: urgent_wr bit 7-4: up_threshold bit 3-0: down_threshold

DI_CHAN2_FMT_CTRL 0x17B8

Bit(s)	R/W	Default	Description
31	R/W	0	gate_clk_en. 0=No clock gating, free-running; 1=Enable clock gating for power saving.
30	R/W	0	soft_rst. If true, reset formatters.
29	R/W	0	unused
28	R/W	0	if true, horizontal formatter use repeating to generete pixel, otherwise use bilinear interpolation
27-24	R/W	0	horizontal formatter initial phase
23	R/W	0	horizontal formatter repeat pixel 0 enable
22-21	R/W	0	horizontal Y/C ratio, 00: 1:1, 01: 2:1, 10: 4:1
20	R/W	0	horizontal formatter enable
19	R/W	0	if true, always use phase0 while vertical formater, meaning always repeat data, no interpolation
18	R/W	0	if true, disable vertical formatter chroma repeat last line
17	R/W	0	veritcal formatter dont need repeat line on phase0, 1: enable, 0: disable
16	R/W	0	veritcal formatter repeat line 0 enable
15-12	R/W	0	vertical formatter skip line num at the beginning
11-8	R/W	0	vertical formatter initial phase
7-1	R/W	0	vertical formatter phase step (3.4)
0	R/W	0	vertical formatter enable

DI_CHAN2_FMT_W 0x17B9

Bit(s)	R/W	Default	Description
27-16	R/W	0	horizontal formatter width
12-0	R/W	0	vertical formatter width

DI_NRWR_CTRL 0x17C2

Bit(s)	R/W	Default	Description
31	R/W	0	Pending_ddr_wrrsp_nrwr
30	R/W	0	Nrwr_reg_swap
29-26	R/W	0	Nrwr_burst_lim
25	R/W	0	Nrwr_canvas_syncen
24	R/W	0	Nrwr_no_clk_gate

Bit(s)	R/W	Default	Description
23-22	R/W	0	Nrwr_rgb_mode , 0: 4:2:2 to one canvas; 1: 4:4:4 to one canvas; 2: Y to luma canvas, CbCr to chroma canvas, for NV12/21; 3: Reserved.
21-20	R/W	0	Nrwr_hconv_mode
19-18	R/W	0	Nrwr_vconv_mode
17	R/W	0	Nrwr_swap_cbcr
16	R/W	0	Nrwr_urgent
15-8	R/W	0	Nrwr_canvas_index_chroma
7-0	R/W	0	Nrwr_canvas_index_luma

DI_NRWR_X 0x17C0

Bit(s)	R/W	Default	Description
31	R/W	0	Nrwr_little_endian
30	R/W	0	Nrwr_rev_x
29-16	R/W	0	Nrwr_start_x
15-14	R/W	0	nrwr_words_lim[3:2]
13-0			Nrwr_end_x

DI_NRWR_Y 0x17C1

Bit(s)	R/W	Default	Description
31-30	R/W	1	Nrwr_words_lim[1:0]
29	R/W	0	Nrwr_rev_y
28-16	R/W	0	Nrwr_start_y
15	R/W	0	Nrwr_ext_en
14	R/W	1	Nrwr bit10 mode
12-0	R/W	0	Nrwr_end_y

DI_DIWR_CTRL 0x17C8

Bit(s)	R/W	Default	Description
31	R/W	0	Pending_ddr_wrrsp_Diwr
30	R/W	0	Diwr_reg_swap
29-26	R/W	0	Diwr_burst_lim
25	R/W	0	Diwr_canvas_syncen

Bit(s)	R/W	Default	Description
24	R/W	0	Diwr_no_clk_gate
23-22	R/W	0	Diwr_rgb_mode , 0: 4:2:2 to one canvas; 1: 4:4:4 to one canvas; 2: Y to luma canvas, CbCr to chroma canvas, for NV12/21; 3: Reserved.
21-20	R/W	0	Diwr_hconv_mode
19-18	R/W	0	Diwr_vconv_mode
17	R/W	0	Diwr_swap_cbcr
16	R/W	0	Diwr_urgent
15-8	R/W	0	Diwr_canvas_index_chroma
7-0	R/W	0	Diwr_canvas_index_luma

DI_DIWR_X 0x17C6

Bit(s)	R/W	Default	Description
31	R/W	0	Diwr_little_endian
30	R/W	0	Diwr_rev_x
29-16	R/W	0	Diwr_start_x
15-14	R/W	0	Diwr_words_lim[3:2]
13-0			Diwr_end_x

DI_DIWR_Y 0x17C7

Bit(s)	R/W	Default	Description
31-30	R/W	1	Diwr_words_lim[1:0]
29	R/W	0	Diwr_rev_y
28-16	R/W	0	Diwr_start_y
15	R/W	0	Diwr_ext_en
14	R/W	1	Diwr bit10 mode
12-0	R/W	0	Diwr_end_y

DI_PRE_GL_THD 0x20ac

Bit(s)	R/W	Default	Description
21:16	W/R	10	DI PRE hold line number
15:0	W/R	1920	H total pixel number for pre count

DI_POST_GL_CTRL 0x20ad

Bit(s)	R/W	Default	Description
31	W/R	0	post count enable
30	W/R	0	post count reset
29:16	W/R	0x20	total line number for post count
13:0	W/R	0xc	the line number of post frame reset

DI_POST_GL_THD 0x20ae

Bit(s)	R/W	Default	Description
21:16	W/R	10	DI POST hold line number
15:0	W/R	1920	H total pixel number for post count

DI_PRE_CTRL 0x1700

Bit(s)	R/W	Default	Description
31	W/R	0	cbus_pre_frame_rst
30	W/R	0	cbus_pre_soft_rst
29	W/R	0	pre_field_num
27:26	W/R	0	mode_444c422
25	W/R	0	di_cont_read_en
24:23	W/R	0	mode_422c444
22	W/R	0	mtn_after_nr
21	W/R	0	pre field num for nr
20	W/R	0	pre field num for pulldown
19	W/R	0	pre field num for mcdi
18	W/R	0	pd_mtn_swap
17	W/R	0	reg_me_autoen
16	W/R	0	reg_me_en
15	W/R	0	nr_wr_by
14	W/R	0	use_vdin_go_line
13	W/R	0	di_prevdin_en
12	W/R	0	di_pre_viu_link
11	W/R	0	di_chan3_enable

Bit(s)	R/W	Default	Description
10	W/R	0	di_mcinfo_rd_mif en
9	W/R	0	di_buf2_en
8	W/R	0	di_chan2_en
7	W/R	0	prenr_hist_en
6	W/R	0	chan2_hist_en
5	W/R	0	hist_check_en
4	W/R	0	check_after_nr
3	W/R	0	check222p_en
2	W/R	0	check322p_en
1	W/R	0	mtn_en
0	W/R	0	nr_en

DI_POST_CTRL 0x1701

Bit(s)	R/W	Default	Description
31	W/R	0	cbus_post_frame_rst
30	W/R	0	cbus_post_soft_rst
29	W/R	0	post_field_num
21:16	W/R	0	post_hold_fifo_lines
13	W/R	0	prepost_link
12	W/R	0	di_post_viu_link
11	W/R	0	di_post_repeat
10	W/R	0	di_post_drop_1st
9	W/R	0	mif0_to_vpp_en
8	W/R	0	di_vpp_out_en
7	W/R	0	di_wr_bk_en
6	W/R	0	di_mux_en
5	W/R	0	di_blend_en
4	W/R	0	di_mtnp_read_en
3	W/R	0	di_mcvec_read_en
2	W/R	0	di_ei_en

Bit(s)	R/W	Default	Description
1	W/R	0	di_buf1_en
0	W/R	0	di_buf0_en

DI_POST_SIZE 0x1702

Bit(s)	R/W	Default	Description
28:16	W/R	0	vsize1post
12:0	W/R	0	hsize1post

DI_PRE_SIZE 0x1703

Bit(s)	R/W	Default	Description
28:16	W/R	0	vsize1pre
12:0	W/R	0	hsize1pre

DI_EI_CTRL0 0x1704

Bit(s)	R/W	Default	Description
23:16	W/R	0	ei0_filter[2:+] abs_diff_left>filter && ...right>filter && ...top>filter && ...bot>filter -> filter
15:8	W/R	0	ei0_threshold[2:+]
3	W/R	0	ei0_vertical
2	W/R	0	ei0_bpscf2
1	W/R	0	ei0_bpsfar1

DI_EI_CTRL1 0x1705

Bit(s)	R/W	Default	Description
31:24	W/R	0	ei0_diff
23:16	W/R	0	ei0_angle45
15:8	W/R	0	ei0_peak
7:0	W/R	0	ei0_cross

DI_EI_CTRL2 0x1706

Bit(s)	R/W	Default	Description
31:24	W/R	0	ei0_close2
23:16	W/R	0	ei0_close1
15:8	W/R	0	ei0_far2
7:0	W/R	0	ei0_far1

DI_NR_CTRL0 0x1707

Bit(s)	R/W	Default	Description
26	W/R	0	nr_cue_en
25	W/R	0	nr2_en

DI_NR_CTRL1 0x1708

Bit(s)	R/W	Default	Description
31:30	W/R	0	mot_p1txtcore_mode
29:24	W/R	0	mot_p1txtcore_clmt
21:16	W/R	0	mot_p1txtcore_ylmt
15:8	W/R	0	mot_p1txtcore_crate
7:0	W/R	0	mot_p1txtcore_yrate

DI_NR_CTRL2 0x1709

Bit(s)	R/W	Default	Description
29:24	W/R	0	mot_curtxtcore_clmt
21:16	W/R	0	mot_curtxtcore_ylmt
15:8	W/R	0	mot_curtxtcore_crate
7:0	W/R	0	mot_curtxtcore_yrate

DI_CANVAS_URGENT0 0x170a

Bit(s)	R/W	Default	Description
26	R/W	0	di write mif bvalid_sel: 1. Bvalid_singnal from bus, 0: bytes_wr handshakes
25	R/W	0	di write mif burst last sel: 1. All kind of burst last signal include ext_data_last. 0. Used the normal burst last signal
24:16	W/R	0	Di write mif urgent ctrl
9	R/W	0	nr write mif bvalid_sel: 1. Bvalid_singnal from bus, 0: bytes_wr handshakes
8	R/W	0	nr write mif burst last sel: 1. All kind of burst last signal include ext_data_last. 0. Used the normal burst last signal
7:0	W/R	0	Nr write mif urgent ctrl

DI_CANVAS_URGENT1 0x170b

Bit(s)	R/W	Default	Description
31:16	W/R	0	Cont_wr_urgent_ctrl
15:0	W/R	0	mtn_wr_urgent_ctrl

DI_MTN_CTRL1 0x170c

Bit(s)	R/W	Default	Description
11:8	W/R	0	mtn_paramtnthd
7:0	W/R	0	mtn_parafltthd

DI_BLEND_CTRL 0x170d

Bit(s)	R/W	Default	Description
31	W/R	0	blend_1_en
30	W/R	0	blend_mtn_lpf
28	W/R	0	post_mb_en
21:20	W/R	0	blend_top_mode 00: mtn, 01: weave mode, 10: bob mode, 11 : blend mode
19	W/R	0	blend_reg3_enable
18	W/R	0	blend_reg2_enable
17	W/R	0	blend_reg1_enable
16	W/R	0	blend_reg0_enable
15:14	W/R	0	blend_reg3_mode
13:12	W/R	0	blend_reg2_mode
11:10	W/R	0	blend_reg1_mode
9:8	W/R	0	blend_reg0_mode

DI_CANVAS_URGENT2 0x170e

Bit(s)	R/W	Default	Description
31:16	W/R	0	Mtn_rd_urgent_ctrl
15:0	W/R	0	cont_rd_urgent_ctrl

DI_ARB_CTRL 0x170f

Bit(s)	R/W	Default	Description
31:26	W/R	0x20	Di_arb_thd1
25:20	W/R	0x20	Di_arb_thd0
19	W/R	0	Di_arb_tid_mode
18	W/R	0	Di_arb_arb_mode
17	W/R	0	Di_arb_acg_en
16	W/R	0	Di_arb_disable_clk

15:0	W/R	0	Di_arb_req_en

DI_BLEND_REG0_X 0x1710

Bit(s)	R/W	Default	Description
28:16	W/R	0	blend_reg0_startx
12:0	W/R	0	blend_reg0_endx

DI_BLEND_REG0_Y 0x1711

Bit(s)	R/W	Default	Description
28:16	W/R	0	blend_reg0_starty
12:0	W/R	0	blend_reg0_endy

DI_BLEND_REG1_X 0x1712

Bit(s)	R/W	Default	Description
28:16	W/R	0	blend_reg1_startx
12:0	W/R	0	blend_reg1_endx

DI_BLEND_REG1_Y 0x1713

Bit(s)	R/W	Default	Description
28:16	W/R	0	blend_reg1_starty
12:0	W/R	0	blend_reg1_endy

DI_BLEND_REG2_X 0x1714

Bit(s)	R/W	Default	Description
28:16	W/R	0	blend_reg2_startx
12:0	W/R	0	blend_reg2_endx

DI_BLEND_REG2_Y 0x1715

Bit(s)	R/W	Default	Description
28:16	W/R	0	blend_reg2_starty
12:0	W/R	0	blend_reg2_endy

DI_BLEND_REG3_X 0x1716

Bit(s)	R/W	Default	Description
28:16	W/R	0	blend_reg3_startx
12:0	W/R	0	blend_reg3_endx

DI_BLEND_REG3_Y 0x1717

Bit(s)	R/W	Default	Description
28:16	W/R	0	blend_reg3_starty
12:0	W/R	0	blend_reg3_endy

DI_EI_CTRL4 0x171a

Bit(s)	R/W	Default	Description
29	W/R	0	reg_ei_caldrt_amblike2_biasvertical
28:24	W/R	21	reg_ei_caldrt_addxla2list_drtmax
23	W/R	0	N/A
22:20	W/R	1	reg_ei_caldrt_addxla2list_signm0th
19	W/R	1	reg_ei_caldrt_addxla2list_mode
18:16	W/R	3	reg_ei_signm_sad_cor_rate
15:12	W/R	3	reg_ei_signm_sadi_cor_rate
11:6	W/R	2	reg_ei_signm_sadi_cor_ofst
5:0	W/R	4	reg_ei_signm_sad_ofst

DI_EI_CTRL5 0x171b

Bit(s)	R/W	Default	Description
30:28	W/R	5	reg_ei_caldrt_cnflcctchk_frcverthrd
27	W/R	0	N/A
26:24	W/R	2	reg_ei_caldrt_cnflctchk_mg
23:22	W/R	1	reg_ei_caldrt_cnflctchk_ws
21	W/R	1	reg_ei_caldrt_cnflctchk_en
20	W/R	1	reg_ei_caldrt_verfrc_final_en
19	W/R	0	reg_ei_caldrt_verfrc_retimflt_en
18:16	W/R	3	reg_ei_caldrt_verftc_eithratemth
15	W/R	0	reg_ei_caldrt_verfrc_retiming_en
14:12	W/R	2	reg_ei_caldrt_verfrc_bothratemth
11:9	W/R	0	reg_ei_caldrt_ver_thrd
8:4	W/R	4	reg_ei_caldrt_addxla2list_drtmin
3:0	W/R	15	reg_ei_caldrt_addxla2list_drtlimit

DI_EI_CTRL6 0x171c

Bit(s)	R/W	Default	Description
31:24	W/R	80	reg_ei_caldrt_abext_sad12thhig
23:16	W/R	35	reg_ei_caldrt_abext_sad00thlow
15:8	W/R	28	reg_ei_caldrt_abext_sad12thlow
6:4	W/R	1	reg_ei_caldrt_abext_ratemth
2:0	W/R	5	reg_ei_caldrt_abext_drtthrd

DI_EI_CTRL7 0x171d

Bit(s)	R/W	Default	Description
29	W/R	1	reg_ei_caldrt_xlanopeak_codien
28:24	W/R	15	reg_ei_caldrt_xlanopeak_drtmax
23	W/R	1	reg_ei_caldrt_xlanopeak_en
22:20	W/R	3	reg_ei_caldrt_abext_monotrnd_alpha
19:18	W/R	1	reg_ei_caldrt_abext_mononum12_thrd
17:16	W/R	1	reg_ei_caldrt_abext_mononum00_thrd
15:12	W/R	6	reg_ei_caldrt_abext_sad00rate
11:8	W/R	6	reg_ei_caldrt_abext_sad12rate
7:0	W/R	80	reg_ei_caldrt_abext_sad00thhig

DI_EI_CTRL8 0x171e

Bit(s)	R/W	Default	Description
30:28	W/R	2	reg_ei_assign_headtail_magin
26:24	W/R	3	reg_ei_retime_lastcurpncnfltchk_mode
22:21	W/R	0	reg_ei_retime_lastcurpncnfltchk_drtth
20	W/R	0	reg_ei_caldrt_histchk_cnfid
19:16	W/R	0	reg_ei_caldrt_histchk_thrd
15	W/R	0	reg_ei_caldrt_histchk_abext
14	W/R	0	reg_ei_caldrt_histchk_npen
13:11	W/R	3	reg_ei_caldrt_amblike2_drtmg
10:8	W/R	1	reg_ei_caldrt_amblike2_valmg
7:4	W/R	10	reg_ei_caldrt_amblike2_alpha

Bit(s)	R/W	Default	Description
3:0	W/R	4	reg_ei_caldrt_amblike2_drtth

DI_EI_CTRL9 0x171f

Bit(s)	R/W	Default	Description
31:28	W/R	7	reg_ei_caldrt_hcnfcheck_frcvert_xla_th3
27	W/R	1	reg_ei_caldrt_hcnfcheck_frcvert_xla_en
26:24	W/R	4	reg_ei_caldrt_conf_drtth
23:20	W/R	11	reg_ei_caldrt_conf_absdrtth
19:18	W/R	2	reg_ei_caldrt_abcheck_mode1
17:16	W/R	1	reg_ei_caldrt_abcheck_mode0
15:12	W/R	11	reg_ei_caldrt_abcheck_drtth1
11:8	W/R	11	reg_ei_caldrt_abcheck_drtth0
6:4	W/R	3	reg_ei_caldrt_abpnchk1_th
1	W/R	1	reg_ei_caldrt_abpnchk1_en
0	W/R	1	reg_ei_caldrt_abpnchk0_en

DI_EI_CTRL10 0x1793

Bit(s)	R/W	Default	Description
31:28	W/R	0	reg_ei_caldrt_hstrrgchk_drtth
27:24	W/R	8	reg_ei_caldrt_hstrrgchk_frcverthrd
23:20	W/R	4	reg_ei_caldrt_hstrrgchk_mg
19	W/R	0	reg_ei_caldrt_hstrrgchk_1sidnul
18	W/R	0	reg_ei_caldrt_hstrrgchk_excpcnf
17:16	W/R	2	reg_ei_caldrt_hstrrgchk_ws
15	W/R	1	reg_ei_caldrt_hstrrgchk_en
14:13	W/R	2	reg_ei_caldrt_hpncheck_mode
12	W/R	0	reg_ei_caldrt_hpncheck_mute
11:9	W/R	3	reg_ei_caldrt_hcnfcheck_mg2
8:6	W/R	2	reg_ei_caldrt_hcnfcheck_mg1
5:4	W/R	2	reg_ei_caldrt_hcnfcheck_mode
3:0	W/R	9	reg_ei_caldrt_hcnfcheck_frcvert_xla_th5

DI_EI_CTRL11 0x179e

Bit(s)	R/W	Default	Description
30:29	W/R	2	reg_ei_amb_detect_mode
28:24	W/R	8	reg_ei_amb_detect_winth
23:21	W/R	3	reg_ei_amb_decide_rppth
20:19	W/R	1	reg_ei_retime_lastmappncnfltchk_drtth
18:16	W/R	2	reg_ei_retime_lastmappncnfltchk_mode
15:14	W/R	2	reg_ei_retime_lastmapvertfrcchk_mode
13:12	W/R	3	reg_ei_retime_lastvertfrcchk_mode
11:8	W/R	0	reg_ei_retime_lastpnchk_drtth
6	W/R	1	reg_ei_retime_lastpnchk_en
5:4	W/R	3	reg_ei_retime_mode
3	W/R	1	reg_ei_retime_last_en
2	W/R		reg_ei_retime_ab_en
1	W/R	1	reg_ei_caldrt_hstrvertfrcchk_en
0	W/R	0	reg_ei_caldrt_hstrrgchk_mode

DI_EI_CTRL12 0x179f

Bit(s)	R/W	Default	Description
31:28	W/R	13	reg_ei_drtdelay2_lmt
27:26	W/R	2	reg_ei_drtdelay2_notver_lrwin
25:24	W/R	3	reg_ei_drtdelay_mode
23	W/R	0	reg_ei_drtdelay2_mode
22:20	W/R	0	reg_ei_assign_xla_signm0th
19	W/R	1	reg_ei_assign_pkbiasvert_en
18	W/R	1	reg_ei_assign_xla_en
17:16	W/R	0	reg_ei_assign_xla_mode
15:12	W/R	2	reg_ei_assign_nlfilter_magin
11:8	W/R	5	reg_ei_localsearch_maxrange
7:4	W/R	0	reg_ei_xla_drtth
3:0	W/R	3	reg_ei_flatmsad_thrd

DI_EI_CTRL13 0x17a8

Bit(s)	R/W	Default	Description
27:24	W/R	15	reg_ei_int_drt2x_chrdrt_limit
23:20	W/R	0	reg_ei_int_drt16x_core
19:16	W/R	2	reg_ei_int_drtdelay2_notver_cancv
15:8	W/R	20	reg_ei_int_drtdelay2_notver_sadth
7:0	W/R	20	reg_ei_int_drtdelay2_vlddrt_sadth

DI_EI_DRT_CTRL 0x1778

Bit(s)	R/W	Default	Description
31	W/R	0	reg_rectg_en: Low angle enable.
30	W/R	0	reg_recbld_en: New and old drt blend enable.
29:28	W/R	2	reg_rectg_ws : window sideto calculate the reference direction: 0:1x1; 1:1x3; 2:1x5; 3:1x7
27			reserved
26:24	W/R	2	reg_abq_margin : top and bottom curve trend quantilization margin of noise for direction assignments.
23			reserved
22:20	W/R	3	reg_trend_mg : the Margin of the top/bot trend.
19:16	W/R	1	reg_int_d16xc1 : Coring to drtf.
15:14			reserved
13:8	W/R	40	reg_int_chlmt1: Limit to drtf(16x) for chroma angle
7			reserved
6:4	W/R	5	reg_nscheck_thrd:check whether the pixels id noise or not.
3			reserved
2:0	W/R	7	reg_horsl_ws: window side to check the existent number of low angle drt, if the number<the value, drt=raw drt.

DI_EI_DRT_PIXTH 0x1779

Bit(s)	R/W	Default	Description
31:24	W/R	22	reg_min_pix: the threshold of min pix of photos, <threshold the pix do not participate in the monotonic trend calculation.
23:16	W/R	203	reg_max_pix:the threshold of max pix of photos, >threshold the pix do not participate in the monotonic trend calculation.
15:8	W/R	50	reg_dmaxmin_thrdma: the max pixel and min pixel difference is larger than the value the trend existent.

7:0	W/R	30	reg_dmaxmin_thrdmi: the max pixel and min pixel difference is smaller than the value the trend non-existent.

DI_EI_DRT_CORRPIXTH 0x177a

Bit(s)	R/W	Default	Description
31:24	W/R	40	reg_newcorrpix_maxthrd:the new low angle drt sad threshold.
23:16	W/R	60	reg_corrpix_diffthrd: the top and bottom pixel difference is larger than the value, the case may be ultra-low angle.
15:8	W/R	10	reg_corrpix_minthrd: the difference of top and bottom pixel is smaller than the value, the drt may be raw drt.
7:0	W/R	20	reg_corrpix_maxthrd: the difference of top and bottom pixel is larger than the value, the drt may be raw drt.

DI_EI_DRT_RECTG_WAVE 0x177b

Bit(s)	R/W	Default	Description
31:29			reserved
28:24	W/R	0	reg_max_pixwave: the wave of the max pix threshold, prevent min pix close to reg_max_pix caused the number between max and min pix zeros.
23:21			reserved
20:16	W/R	15	reg_pix_wave: the wave of the max and min pix, the max pixel smaller than the value or the min pixel larger than the value, may be the ultra-low angle case.
15:14			reserved
13:8	W/R	40	reg_maxdrt_thrd: the threshold of the low angle max drt.
7:0	W/R	20	reg_wave_thrd:in bilateral cases tow pixel difference is smaller than the value, the trend between the tow pixel not change.

DI_EI_DRT_PIX_DIFFTH 0x177c

Bit(s)	R/W	Default	Description
31:24	W/R	32	reg_newraw_thrd: the old drt and new drt transition threshold.
23:16	W/R	10	reg_tb_max_thrd: the threshold of top and bottom max or min pixel.
15:8	W/R	20	reg_diffpix_thrd: Max-Min<the value,the trend is non-existent.
7:6			reserved
5:0	W/R	5	reg_bilt_trendnumt: in bilateral cases the difference between the top and bottom pixel number of the monotonic trend smaller than the value is low angle.

DI_EI_DRT_UNBITREND_TH 0x177d

Bit(s)	R/W	Default	Description
31:29			reserved

28:24	W/R	10	reg_trend_numb:in bilateral cases the pixel number of the monotonic trend larger than the value is low angle.
23:21			reserved
20:16	W/R	4	reg_bilt_trendnum:in bilateral cases the pixel number of the trend larger than the value is low angle.
15:13			reserved
12:8	W/R	7	reg_unil_trendnumt: in unilateral cases the difference between the top and bottom pixel number of the monotonic trend smaller than the value is low angle.
7:5			reserved
4:0	W/R	10	reg_trend_num: in unilateral cases the pixel number of the trend larger than the value is ultra-low angle.

DI_EI_XWIN0 0x1798

Bit(s)	R/W	Default	Description
27:16	W/R	0	ei_xend0
11:0	W/R	0	ei_xstart0

DI_EI_XWIN1 0x1799

Bit(s)	R/W	Default	Description
27:16	W/R	0	ei_xend01
11:0	W/R	0	ei_xstart1

DI_MC_REG0_X 0x1720

Bit(s)	R/W	Default	Description
27:16	W/R	0	mc_reg0_start_x
11:0	W/R	0	mc_reg0_end_x

DI_MC_REG0_Y 0x1721

Bit(s)	R/W	Default	Description
27:16	W/R	0	mc_reg0_start_y
11:0	W/R	0	mc_reg0_end_y

DI_MC_REG1_X 0x1722

Bit(s)	R/W	Default	Description
27:16	W/R	0	mc_reg1_start_x
11:0	W/R	0	mc_reg1_end_x

DI_MC_REG1_Y 0x1723

Bit(s)	R/W	Default	Description
27:16	W/R	0	mc_reg1_start_y
11:0	W/R	0	mc_reg1_end_y

DI_MC_REG2_X 0x1724

Bit(s)	R/W	Default	Description
27:16	W/R	0	mc_reg2_start_x
11:0	W/R	0	mc_reg2_end_x

DI_MC_REG2_Y 0x1725

Bit(s)	R/W	Default	Description
27:16	W/R	0	mc_reg2_start_y
11:0	W/R	0	mc_reg2_end_y

DI_MC_REG3_X 0x1726

Bit(s)	R/W	Default	Description
27:16	W/R		mc_reg3_start_x
11:0	W/R		mc_reg3_end_x

DI_MC_REG3_Y 0x1727

Bit(s)	R/W	Default	Description
27:16	W/R	0	mc_reg3_start_y
11:0	W/R	0	mc_reg3_end_y

DI_MC_REG4_X 0x1728

Bit(s)	R/W	Default	Description
27:16	W/R	0	mc_reg4_start_x
11:0	W/R	0	mc_reg4_end_x

DI_MC_REG4_Y 0x1729

Bit(s)	R/W	Default	Description
27:16	W/R	0	mc_reg4_start_y
11:0	W/R	0	mc_reg4_end_y

DI_MC_32LVL0 0x172a

Bit(s)	R/W	Default	Description

31:24	W/R	0	mc_reg2_32lvl
23:16	W/R	0	mc_reg1_32lvl
15:8	W/R	0	mc_reg0_32lvl
7:0	W/R	0	field_32lvl

DI_MC_32LVL1 0x172b

Bit(s)	R/W	Default	Description
15:8	W/R	0	mc_reg3_32lvl
7:0	W/R	0	mc_reg4_32lvl

DI_MC_22LVL0 0x172c

Bit(s)	R/W	Default	Description
31:16	W/R	0	mc_reg0_22lvl
15:0	W/R	0	field_22lvl

DI_MC_22LVL1 0x172d

Bit(s)	R/W	Default	Description
31:16	W/R	0	mc_reg2_22lvl
15:0	W/R	0	mc_reg1_22lvl

DI_MC_22LVL2 0x172e

Bit(s)	R/W	Default	Description
31:16	W/R	0	mc_reg4_22lvl
15:0	W/R	0	mc_reg3_22lvl

DI_MC_CTRL 0x172f

Bit(s)	R/W	Default	Description
4	W/R	0	mc_reg4_en
3	W/R	0	mc_reg3_en
2	W/R	0	mc_reg2_en
1	W/R	0	mc_reg1_en
0	W/R	0-	mc_reg0_en

DI_INTR_CTRL 0x1730

Bit(s)	R/W	Default	Description
31	W/R	0	Deint_irq_mode ==0: predi_int/postdi_int created, interrupt flag is on Deint_irq_mode==1 when any interrupt is generated in DI modules and no Mask action, interrupt flag is on
30:25	W/R	0	reserved
24	w/R	0	Det3d_int_mask
23	w/R	0	Mcinfowr_int_mask
22	w/R	0	Mcvecwr_int_mask
21	w/R	0	Medi_int_mask
20	w/R	0	Contwr_int_mask
19	w/R	0	Hist_int_mask
18	w/R	0	Diwr_int_mask
17	w/R	0	Mtn_wr_int_mask
16	w/R	0	Nrwr_int_mask
15:9	W/R	0	reserved
8	R	0	Det3d_done
7	R	0	Mcinfowr_done (not valid in GX)
6	R	0	Mcvecwr_done (not valid in GX)
5	R	0	Medi_done(not valid in GX)
4	R	0	Contwr_done
3	R	0	Hist_done
2	R	0	diwr_done
1	R	0	Mtnwr_done
0	R	0	Nrwr_done

DI_INFO_ADDR 0x1731

Addr_0

Bit(s)	R/W	Default	Description
31:0	R	0	Field_32p , sum of difference between n-2 and n

Addr_1

Bit(s)	R/W	Default	Description

31:24	R	0	Field_32max, maximum difference between n-2 and n
23:0	R	0	Field_32num, numbers of pixels difference > threshold

Addr_2

Bit(s)	R/W	Default	Description
31:0	R	0	Field_22p, sum of difference between temporal and vertical difference

Addr_3

Bit(s)	R/W	Default	Description
15:0	R	0	Field_22max , maximum difference between temporal and verticaldifference

Addr_4

Bit(s)	R/W	Default	Description
23:0	R	0	Field_22num, pixel sum which difference > threshold

Addr_5

Bit(s)	R/W	Default	Description
31:0	R	0	Luma sum

Addr_6

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 32, sum in area 0

Addr_7

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 32, sum in area 1

Addr_8

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 32, sum in area 2

Addr_9

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 32, sum in area 3

Addr_10

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 32, sum in area 4

Addr_11

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 22, sum in area 0

Addr_12

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 22, sum in area 1

Addr_13

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 22, sum in area 2

Addr_14

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 22, sum in area 3

Addr_15

Bit(s)	R/W	Default	Description
31:0	R	0	Difference of 22, sum in area 4

Addr_16

Bit(s)	R/W	Default	Description
31:0	R	0	luma, sum in area 0

Addr_17

Bit(s)	R/W	Default	Description
31:0	R	0	luma, sum in area 1

Addr_18

Bit(s)	R/W	Default	Description
31:0	R	0	luma, sum in area 2

Addr_19

Bit(s)	R/W	Default	Description
31:0	R	0	luma, sum in area 3

Addr_20

Bit(s)	R/W	Default	Description

31:0	R	0	luma, sum in area 4

Addr_21

Bit(s)	R/W	Default	Description
31:24	R	0	Field_32max, maximum difference between n-2 and n in area0
23:0	R	0	Field_32num, numbers of pixels difference > threshold in area0

Addr_22

Bit(s)	R/W	Default	Description
31:24	R	0	Field_32max, maximum difference between n-2 and n in area1
23:0	R	0	Field_32num, numbers of pixels difference > threshold in area1

Addr_23

Bit(s)	R/W	Default	Description
31:24	R	0	Field_32max, maximum difference between n-2 and n in area2
23:0	R	0	Field_32num, numbers of pixels difference > threshold in area2

Addr_24

Bit(s)	R/W	Default	Description
31:24	R	0	Field_32max, maximum difference between n-2 and n in area3
23:0	R	0	Field_32num, numbers of pixels difference > threshold in area3

Addr_25

Bit(s)	R/W	Default	Description
31:24	R	0	Field_32max, maximum difference between n-2 and n in area4
23:0	R	0	Field_32num, numbers of pixels difference > threshold in area4

Addr_26

Bit(s)	R/W	Default	Description
31:20	R	0	Field_22max/16, in area 0
19:0	R	0	Field_22 num/16, in area 0

Addr_27

Bit(s)	R/W	Default	Description
31:20	R	0	Field_22max/16, in area 1
19:0	R	0	Field_22 num/16, in area 1

Addr_28

Bit(s)	R/W	Default	Description
31:20	R	0	Field_22max/16, in area 2
19:0	R	0	Field_22 num/16, in area 2

Addr_29

Bit(s)	R/W	Default	Description
31:20	R	0	Field_22max/16, in area 3
19:0	R	0	Field_22 num/16, in area 3

Addr_30

Bit(s)	R/W	Default	Description
31:20	R	0	Field_22max/16, in area 4
19:0	R	0	Field_22 num/16, in area 4

DI_PRE_HOLD 0x1733

Bit(s)	R/W	Default	Description
31	R/W	0	cntl_pre_hold_enable
27:16	R/W	0	cntl_pre_hold_count
11:0	R/W	0	cntl_pre_pass_count

DI_MTN_1_CTRL1 0x1740

Bit(s)	R/W	Default	Description
31	W/R	0	reg_mtn_1_en
30	W/R	0	reg_mtn_init
29	W/R	0	reg_di2nr_txt_en
28	W/R	0	reg_di2nr_txt_mode
27:24	W/R	0	reg_mtn_def
23:16	W/R	32	reg_DI_cmb_adp_YCrate
15: 8	W/R	32	reg_DI_cmb_adp_2Crate
7: 0	W/R	21	reg_DI_cmb_adp_2Yrate

DI_MTN_1_CTRL2 0x1741

Bit(s)	R/W	Default	Description
31:24	W/R	26	reg_DI_m1b_core_Ykinter

23:16	W/R	26	reg_DI_m1b_core_Ckinter
15:8	W/R	58	reg_DI_m1b_core_Ykintra
7:0	W/R	98	reg_DI_m1b_core_Ckintra

DI_MTN_1_CTRL3 0x1742

Bit(s)	R/W	Default	Description
31:24	W/R	21	reg_DI_m1b_thr_2Yrate
23:16	W/R	32	reg_DI_m1b_thr_2Crate
15: 8	W/R	10	reg_DI_m1b_core_mxcmbY
7: 0	W/R	10	reg_DI_m1b_core_mxcmbC

DI_MTN_1_CTRL4 0x1743

Bit(s)	R/W	Default	Description
31:24	W/R	1	reg_DI_m1b_coreY
23:16	W/R	0	reg_DI_m1b_coreC
15: 8	W/R	8	reg_DI_m1b_thrd_min
7: 0	W/R	128	reg_DI_m1b_thrd_max

DI_MTN_1_CTRL5 0x1744

Bit(s)	R/W	Default	Description
31:27	W/R	7	reg_DI_m1b_pp_extnd_num
27:24	W/R	4	reg_DI_m1b_pp_errod_num
21:20	W/R	0	Re_di2nr_txt_mode 0:average of top/bot 1:max; 2: |a+c-2b|;
15: 8	W/R	13	reg_DI_mot_core_Ykinter
7: 0	W/R	13	reg_DI_mot_core_Ckinter

DI_MTN_1_CTRL6 0x17a9

Bit(s)	R/W	Default	Description
31:24	W/R	13	reg_DI_mot_core_Ykintra
23:16	W/R	90	reg_DI_mot_core_Ckintra
15: 8	W/R	21	reg_DI_mot_cor_2Yrate
7: 0	W/R	32	reg_DI_mot_cor_2Crate

DI_MTN_1_CTRL7 0x17aa

Bit(s)	R/W	Default	Description

31:24	W/R	10	reg_DI_mot_core_mxcmbY
23:16	W/R	10	reg_DI_mot_core_mxcmbC
15: 8	W/R	2	reg_DI_mot_coreY
7: 0	W/R	1	reg_DI_mot_coreC

DI_MTN_1_CTRL8 0x17ab

Bit(s)	R/W	Default	Description
31:24	W/R	26	reg_DI_fmot_core_Ykinter
23:16	W/R	26	reg_DI_fmot_core_Ckinter
15: 8	W/R	38	reg_DI_fmot_core_Ykintra
7: 0	W/R	98	reg_DI_fmot_core_Ckintra

DI_MTN_1_CTRL9 0x17ac

Bit(s)	R/W	Default	Description
31:24	W/R	13	reg_DI_fmot_cor_2Yrate
23:16	W/R	32	reg_DI_fmot_cor_2Crate
15: 8	W/R	3	reg_DI_fmot_coreY
7: 0	W/R	2	reg_DI_fmot_coreC

DI_MTN_1_CTRL10 0x17ad

Bit(s)	R/W	Default	Description
27:24	W/R	2	reg_DI_m1b_suremot_num_fld0
19:16	W/R	2	reg_DI_m1b_surestl_num_fld0
11: 8	W/R	6	reg_DI_m1b_suremot_num_fld1
3: 0	W/R	6	reg_DI_m1b_surestl_num_fld1

DI_MTN_1_CTRL11 0x17ae

Bit(s)	R/W	Default	Description
27:24	W/R	5	reg_DI_m1b_suremot_evn_th
20:16	W/R	8	reg_DI_m1b_suremot_odd_th
11: 8	W/R	3	reg_DI_m1b_surestl_evn_th
6	W/R	0	reg_DI_m1b_suremot_fast_en
5	W/R	0	reg_DI_m1b_surestl_fast_en
4: 0	W/R	4	reg_DI_m1b_surestl_odd_th

DI_MTN_1_CTRL12 0x17af

Bit(s)	R/W	Default	Description
31:24	W/R	64	reg_DI_mot_norm_gain
17:16	W/R	2	reg_DI_mot_alpha_lpf
15: 8	W/R	10	reg_DI_m1b_surestl_thrd
4: 0	W/R	4	reg_DI_mot_surestl_gain

10.2.3.8 NR2 Registers

DET3D_MOTN_CFG 0x1734

Bit(s)	R/W	Default	Description
16	R/W	0	reg_det3d_intr_en : Det3d interrupt enable
9:8	R/W	0	reg_Det3D_Motion_Mode : U2 Different mode for Motion Calculation of Luma and Chroma: 0 : MotY, 1: (2*MotY + (MotU + MotV))/4; 2: Max(MotY, MotU,MotV); 3:Max(MotY, (MotU+MotV)/2)
7:4	R/W	0	reg_Det3D_Motion_Core_Rate : U4 K Rate to Edge (HV) details for coring of Motion Calculations, normalized to 32
3:0	R/W	0	reg_Det3D_Motion_Core_Thrd : U4 2X: static coring value for Motion Detection.

DET3D_CB_CFG 0x1735

Bit(s)	R/W	Default	Description
7:4	R/W	0	reg_Det3D_ChessBd_HV_ofst : U4, Noise immune offset for Horizotnal or vertical combing detection.
3:0	R/W	0	reg_Det3D_ChessBd_NHV_ofst : U4, Noise immune offset for NON-Horizotnal or vertical combing detection.

DET3D_SPLT_CFG 0x1736

Bit(s)	R/W	Default	Description
7:4	R/W	0x0	reg_Det3D_SplitValid_ratio : U4, Ratio between max_value and the avg_value of the edge mapping for split line valid detection. The smaller of this value, the easier of the split line detected.
3:0	R/W	0x0	reg_Det3D_AvgIdx_ratio : U4, Ratio to the avg_value of the edge mapping for split line position estimation. The smaller of this value, the more samples will be added to the estimation.

DET3D_HV_MUTE 0x1737

Bit(s)	R/W	Default	Description
23:20	R/W	0x0	reg_Det3D_Edge_Ver_Mute : U4 X2: Horizontal pixels to be mute from H/V Edge calculation Top and Bottom border part.
19:16	R/W	0x0	reg_Det3D_Edge_Hor_Mute : U4 X2: Horizontal pixels to be mute from H/V Edge calculation Left and right border part.

15:12	R/W	0x0	reg_Det3D_ChessBd_Ver_Mute : U4 X2: Horizontal pixels to be mute from ChessBoard statistics calculation in middle part
11:8	R/W	0x0	reg_Det3D_ChessBd_Hor_Mute : U4 X2: Horizontal pixels to be mute from ChessBoard statistics calculation in middle part
7:4	R/W	0x0	reg_Det3D_STA8X8_Ver_Mute : U4 1X: Vertical pixels to be mute from 8x8 statistics calculation in each block.
3:0	R/W	0x0	reg_Det3D_STA8X8_Hor_Mute : U4 1X: Horizontal pixels to be mute from 8x8 statistics calculation in each block.

DET3D_MAT_STA_P1M1 0x1738

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_Det3D_STA8X8_P1_K0_R8 : U8 SAD to SAI ratio to decide P1, normalized to 256 (0.8)
23:16	R/W	0x0	reg_Det3D_STA8X8_P1_K1_R7 : U8 SAD to ENG ratio to decide P1, normalized to 128 (0.5)
15:8	R/W	0x0	reg_Det3D_STA8X8_M1_K0_R6 : U8 SAD to SAI ratio to decide M1, normalized to 64 (1.1)
7:0	R/W	0x0	reg_Det3D_STA8X8_M1_K1_R6 : U8 SAD to ENG ratio to decide M1, normalized to 64 (0.8)

DET3D_MAT_STA_P1TH 0x1739

Bit(s)	R/W	Default	Description
23:16	R/W	0x0	reg_Det3D_STAYUV_P1_TH_L4 : U8 SAD to ENG Thrd offset to decide P1, X16 (100)
15:8	R/W	0x0	reg_Det3D_STAEDG_P1_TH_L4 : U8 SAD to ENG Thrd offset to decide P1, X16 (80)
7:0	R/W	0x0	reg_Det3D_STAMOT_P1_TH_L4 : U8 SAD to ENG Thrd offset to decide P1, X16 (48)

DET3D_MAT_STA_M1TH 0x173a

Bit(s)	R/W	Default	Description
23:16	R/W	0x0	reg_Det3D_STAYUV_M1_TH_L4 : U8 SAD to ENG Thrd offset to decide M1, X16 (100)
15:8	R/W	0x0	reg_Det3D_STAEDG_M1_TH_L4 : U8 SAD to ENG Thrd offset to decide M1, X16 (80)
7:0	R/W	0x0	reg_Det3D_STAMOT_M1_TH_L4 : U8 SAD to ENG Thrd offset to decide M1, X16 (64)

DET3D_MAT_STA_RSFT 0x173b

Bit(s)	R/W	Default	Description

5:4	R/W	0x0	reg_Det3D_STAYUV_RSHFT : U2 YUV statistics SAD and SAI calculation result right shift bits to accommodate the 12bits clipping: 0 : mainly for images <=720x480: 1: mainly for images <=1366x768: 2: mainly for images <=1920X1080: 2; 3: other higher resolutions
3:2	R/W	0x0	reg_Det3D_STAEDG_RSHFT : U2 Horizontal and Vertical Edge Statistics SAD and SAI calculation result right shift bits to accommodate the 12bits clipping: 0 : mainly for images <=720x480: 1: mainly for images <=1366x768: 2: mainly for images <=1920X1080: 2; 3: other higher resolutions
1:0	R/W	0x0	reg_Det3D_STAMOT_RSHFT : U2 Motion SAD and SAI calculation result right shift bits to accommodate the 12bits clipping: 0 : mainly for images <=720x480: 1: mainly for images <=1366x768: 2: mainly for images <=1920X1080: 2; 3: other higher resolutions

DET3D_MAT_SYMTC_TH 0x173c

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_Det3D_STALUM_symtc_Th : U8 threshold to decide if the Luma statistics is TB or LR symmetric.
23:16	R/W	0x0	reg_Det3D_STACHR_symtc_Th : U8 threshold to decide if the Chroma (UV) statistics is TB or LR symmetric.
15:8	R/W	0x0	reg_Det3D_STAEDG_symtc_Th : U8 threshold to decide if the Horizontal and Vertical Edge statistics is TB or LR symmetric.
7:0	R/W	0x0	reg_Det3D_STAMOT_symtc_Th : U8 threshold to decide if the Motion statistics is TB or LR symmetric.

DET3D_RO_DET_CB_HOR 0x173d

Bit(s)	R/W	Default	Description
31:16	R.O	0x0	RO_Det3D_ChessBd_NHor_value : U16 X64: number of Pixels of Horizontally Surely NOT matching Chessboard pattern.
15:0	R.O	0x0	RO_Det3D_ChessBd_Hor_value : U16 X64: number of Pixels of Horizontally Surely matching Chessboard pattern.

DET3D_RO_DET_CB_VER 0x173e

Bit(s)	R/W	Default	Description
31:16	R.O	0x0	RO_Det3D_ChessBd_NVer_value : U16 X64: number of Pixels of Vertically Surely NOT matching Chessboard pattern.
15:0	R.O	0x0	RO_Det3D_ChessBd_Ver_value : U16 X64: number of Pixels of Vertically Surely matching Chessboard pattern.

DET3D_RO_SPLT_HT 0x173f

Bit(s)	R/W	Default	Description
24	R.O	0x0	RO_Det3D_Split_HT_valid : U1 horizontal LR split border detected valid signal for top half picture
20:16	R.O	0x0	RO_Det3D_Split_HT_pxnum : U5 number of pixels included for the LR split position estimation for top half picture

9:0	R.O	0x0	RO_Det3D_Split_HT_idxX4 : S10 X4: horizontal pixel shifts of LR split position to the (ColMax/2) for top half picture

NR2 REG DEFINE BEGIN////

NR2_MET_NM_CTRL 0x1745

Bit(s)	R/W	Default	Description
28	R/W	0x0	reg_NM_reset : Reset to the status of the Loop filter.
27:24	R/W	0x0	reg_NM_calc_length : Length mode of the Noise measurement sample number for statistics. 0 : 256 samples; 1: 512 samples; 2: 1024 samples; ¡-X: 2^(8+x) samples
23:20	R/W	0x0	reg_NM_inc_step : Loop filter input gain increase step.
19:16	R/W	0x0	reg_NM_dec_step : Loop filter input gain decrease step.
15:8	R/W	0x0	reg_NM_YHPmot_thrd : Luma channel HP portion motion for condition of pixels included in Luma Noise measurement.
7:0	R/W	0x0	reg_NM_CHPmot_thrd : Chroma channel HP portion motion for condition of pixels included in Chroma Noise measurement.

NR2_MET_NM_YCTRL 0x1746

Bit(s)	R/W	Default	Description
31:28	R/W	0x0	reg_NM_YPLL_target : Target rate of NM_Ynoise_thrd to mean of the Luma Noise
27:24	R/W	0x0	reg_NM_YLPmot_thrd : Luma channel LP portion motion for condition of pixels included in Luma Noise measurement.
23:16	R/W	0x0	reg_NM_YHPmot_thrd_min : Minimum threshold for Luma channel HP portion motion to decide whether the pixel will be included in Luma noise measurement.
15:8	R/W	0x0	reg_NM_YHPmot_thrd_max : Maximum threshold for Luma channel HP portion motion to decide whether the pixel will be included in Luma noise measurement.
7:0	R/W	0x0	reg_NM_Ylock_rate : Rate to decide whether the Luma noise measurement is lock or not.

NR2_MET_NM_CCTRL 0x1747

Bit(s)	R/W	Default	Description
31:28	R/W	0x0	reg_NM_CPLL_target : Target rate of NM_Cnoise_thrd to mean of the Chroma Noise
27:24	R/W	0x0	reg_NM_CLPmot_thrd : Chroma channel LP portion motion for condition of pixels included in Chroma Noise measurement.
23:16	R/W	0x0	reg_NM_CHPmot_thrd_min : Minimum threshold for Chroma channel HP portion motion to decide whether the pixel will be included in Chroma noise measurement.
15:8	R/W	0x0	reg_NM_CHPmot_thrd_max : Maximum threshold for Chroma channel HP portion motion to decide whether the pixel will be included in Chroma noise measurement.
7:0	R/W	0x0	reg_NM_Clock_rate : Rate to decide whether the Chroma noise measurement is lock or not;

NR2_MET_NM_TNR 0x1748

Bit(s)	R/W	Default	Description
25	R.O	0x0	ro_NM_TNR_Ylock : Read-only register to tell ifLuma channel noise measurement is locked or not.
24	R.O	0x0	ro_NM_TNR_Clock : Read-only register to tell if Chroma channel noise measurement is locked or not.
23:12	R.O	0x0	ro_NM_TNR_Ylevel : Read-only register to give Luma channel noise level. It was 16x of pixel difference in 8 bits of YHPmot.
11:0	R.O	0x0	ro_NM_TNR_Clevel : Read-only register to give Chroma channel noise level. It was 16x of pixel difference in 8 bits of CHPmot.

NR2_MET_NMFRM_TNR_YLEV 0x1749

Bit(s)	R/W	Default	Description
28:0	R.O	0x0	ro_NMFrm_TNR_Ylevel : Frame based Read-only register to give Luma channel noise level within one frame/field.

NR2_MET_NMFRM_TNR_YCNT 0x174a

Bit(s)	R/W	Default	Description
23:0	R.O	0x0	ro_NMFrm_TNR_Ycount : Number ofLuma channel pixels included in Frame/Field based noise level measurement.

NR2_MET_NMFRM_TNR_CLEV 0x174b

Bit(s)	R/W	Default	Description
28:0	R.O	0x0	ro_NMFrm_TNR_Clevel : Frame based Read-only register to give Chroma channel noise level within one frame/field.

NR2_MET_NMFRM_TNR_CCNT 0x174c

Bit(s)	R/W	Default	Description
23:0	R.O	0x0	ro_NMFrm_TNR_Ccount : Number of Chroma channel pixels included in Frame/Field based noise level measurement.

NR2_3DEN_MODE 0x174d

Bit(s)	R/W	Default	Description
6:4	R/W	0x0	Blend_3dnr_en_r :
2:0	R/W	0x0	Blend_3dnr_en_l :

NR2_IIR_CTRL 0x174e

Bit(s)	R/W	Default	Description
15:14	R/W	0x0	reg_LP_IIR_8bit_mode : LP IIR membitwidth mode:0: 10bits will be store in memory;1: 9bits will be store in memory; 2 : 8bits will be store in memory;3: 7bits will be store in memory;

Bit(s)	R/W	Default	Description
13:12	R/W	0x0	reg_LP_IIR_mute_mode : Mode for the LP IIR mute,
11:8	R/W	0x0	reg_LP_IIR_mute_thrd : Threshold of LP IIR mute to avoid ghost:
7:6	R/W	0x0	reg_HP_IIR_8bit_mode : IIR membitwidth mode:0: 10bits will be store in memory;1: 9bits will be store in memory; 2 : 8bits will be store in memory;3: 7bits will be store in memory;
5:4	R/W	0x0	reg_HP_IIR_mute_mode : Mode for theLP IIR mute
3:0	R/W	0x0	reg_HP_IIR_mute_thrd : Threshold of HP IIR mute to avoid ghost

NR2_SNR_SAD_CFG 0x1751

Bit(s)	R/W	Default	Description
12	R/W	0x1	reg_MATNR_SNR_SAD_CenRPL : U1, Enable signal for Current pixel position SAD to be replaced by SAD_min.0: do not replace Current pixel position SAD by SAD_min;1: do replacements
11:8	R/W	0x3	reg_MATNR_SNR_SAD_coring : Coring value of the intra-frame SAD. sum = (sum - reg_MATNR_SNR_SAD_coring);sum = (sum<0) ? 0: (sum>255)? 255: sum;
6:5	R/W	0x1	reg_MATNR_SNR_SAD_WinMod : Unsigned, Intra-frame SAD matching window mode:0: 1x1; 1: [1 1 1] 2: [1 2 1]; 3: [1 2 2 2 1];
4:0	R/W	0x1	Sad_coef_num : Sad coeffient

NR2_MATNR_SNR_OS 0x1752

Bit(s)	R/W	Default	Description
7:4	R/W	0x8	reg_MATNR_SNR_COS : SNR Filter overshoot control margin for UV channel (X2 to u10 scale)
3:0	R/W	0xd	reg_MATNR_SNR_YOS : SNR Filter overshoot control margin for luma channel (X2 to u10 scale)

NR2_MATNR_SNR_NRM_CFG 0x1753

Bit(s)	R/W	Default	Description
23:16	R/W	0x40	reg_MATNR_SNR_NRM_ofst : Edge based SNR boosting normalization offset to SAD_max ;
15:8	R/W	0xff	reg_MATNR_SNR_NRM_max : Edge based SNR boosting normalization Max value
7:0	R/W	0x0	reg_MATNR_SNR_NRM_min : Edge based SNR boosting normalization Min value

NR2_MATNR_SNR_NRM_GAIN 0x1754

Bit(s)	R/W	Default	Description
15:8	R/W	0x0	reg_MATNR_SNR_NRM_Cgain : Edge based SNR boosting normalization Gain for Chrm channel (norm 32 as 1)
7:0	R/W	0x20	reg_MATNR_SNR_NRM_Ygain : Edge based SNR boosting normalization Gain for Luma channel (norm 32 as 1)

NR2_MATNR_SNR_LPF_CFG 0x1755

Bit(s)	R/W	Default	Description
23:16	R/W	0xc	reg_MATNR_SNRLPF_SADmaxTH : U8, Threshold to SADmax to use TNRLPF to replace SNRLPF. i.e.if (SAD_max<reg_MATNR_SNRLPF_SADmaxTH) SNRLPF_yuv[k] = TNRLPF_yuv[k];
13:11	R/W	0x2	reg_MATNR_SNRLPF_Cmode : LPF based SNR filtering mode on CHRM channel: 0 : gradient LPF [1 1]/2, 1: gradient LPF [2 1 1]/4; 2: gradient LPF [3 3 2]/8; 3: gradient LPF [5 4 4 3]/16; 4 : TNRLPF; 5 : CurLPF3x3_yuv[]; 6: CurLPF3o3_yuv[] 7: CurLPF3x5_yuv[]
10: 8	R/W	0x2	reg_MATNR_SNRLPF_Ymode : LPF based SNR filtering mode on LUMA channel: 0 : gradient LPF //Bit [1 1]/2, 1: gradient LPF [2 1 1]/4; 2: gradient LPF [3 3 2]/8;3: gradient LPF [5 4 4 3]/16; 4 : TNRLPF; 5 : CurLPF3x3_yuv[]; 6: CurLPF3o3_yuv[] 7: CurLPF3x5_yuv[]
7:4	R/W	0x6	reg_MATNR_SNRLPF_SADmin3TH : Offset threshold to SAD_min to Discard SAD_min3 corresponding pixel in LPF SNR filtering. (X8 to u8 scale)
3:0	R/W	0x4	reg_MATNR_SNRLPF_SADmin2TH : Offset threshold to SAD_min to Discard SAD_min2 corresponding pixel in LPF SNR filtering. (X8 to u8 scale)

NR2_MATNR_SNR_USF_GAIN 0x1756

Bit(s)	R/W	Default	Description
15:8	R/W	0x0	reg_MATNR_SNR_USF_Cgain : Un-sharp (HP) compensate back Chrm portion gain, (norm 64 as 1)
7:0	R/W	0x0	reg_MATNR_SNR_USF_Ygain : Un-sharp (HP) compensate back Luma portion gain, (norm 64 as 1)

NR2_MATNR_SNR_EDGE2B 0x1757

Bit(s)	R/W	Default	Description
15:8	R/W	0x80	reg_MATNR_SNR_Edge2Beta_ofst : U8, Offset for Beta based on Edge.
7:0	R/W	0x10	reg_MATNR_SNR_Edge2Beta_gain : U8. Gain to SAD_min for Beta based on Edge. (norm 16 as 1)

NR2_MATNR_BETA_EGAIN 0x1758

Bit(s)	R/W	Default	Description
15:8	R/W	0x20	reg_MATNR_CBeta_Egain : U8, Gain to Edge based Beta for Chrm channel. (normalized to 32 as 1)
7:0	R/W	0x20	reg_MATNR_YBeta_Egain : U8, Gain to Edge based Beta for Luma channel. (normalized to 32 as 1)

NR2_MATNR_BETA_BRT 0x1759

Bit(s)	R/W	Default	Description

31:28	R/W	0x0	reg_MATNR_beta_BRT_limt_hi : U4, Beta adjustment based on Brightness high side Limit. (X16 to u8 scale)
27:24	R/W	0x0	reg_MATNR_beta_BRT_slop_hi : U4, Beta adjustment based on Brightness high side slope. Normalized to 16 as 1
23:16	R/W	0xa0	reg_MATNR_beta_BRT_thrd_hi : U8, Beta adjustment based on Brightness high threshold.(u8 scale)
15:12	R/W	0x6	reg_MATNR_beta_BRT_limt_lo : U4, Beta adjustment based on Brightness low side Limit. (X16 to u8 scale)
11:8	R/W	0x6	reg_MATNR_beta_BRT_slop_lo : U4, Beta adjustment based on Brightness low side slope. Normalized to 16 as 1
7:0	R/W	0x64	reg_MATNR_beta_BRT_thrd_lo : U8, Beta adjustment based on Brightness low threshold.(u8 scale)

NR2_MATNR_XBETA_CFG 0x175a

Bit(s)	R/W	Default	Description
19:18	R/W	0x0	reg_MATNR_CBeta_use_mode : U2, Beta options (mux) from beta_motion and beta_edge for Chrm channel;
17:16	R/W	0x0	reg_MATNR_YBeta_use_mode : U2, Beta options (mux) from beta_motion and beta_edge for Luma channel;
15: 8	R/W	0x0	reg_MATNR_CBeta_Ofst : U8, Offset to Beta for Chrm channel.(after beta_edge and beta_motion mux)
7: 0	R/W	0x0	reg_MATNR_YBeta_Ofst : U8, Offset to Beta for Luma channel.(after beta_edge and beta_motion mux)

NR2_MATNR_YBETA_SCL 0x175b

Bit(s)	R/W	Default	Description
31:24	R/W	0x3c	reg_MATNR_YBeta_scale_min : U8, Final step Beta scale low limit for Luma channel;
23:16	R/W	0xff	reg_MATNR_YBeta_scale_max : U8, Final step Beta scale high limit for Luma channe;
15: 8	R/W	0x20	reg_MATNR_YBeta_scale_gain : U8, Final step Beta scale Gain for Luma channel (normalized 32 to 1);
7 : 0	R/W	0x0	reg_MATNR_YBeta_scale_ofst : S8, Final step Beta scale offset for Luma channel ;

NR2_MATNR_CBETA_SCL 0x175c

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_MATNR_CBeta_scale_min : Final step Beta scale low limit for Chrm channel.Similar to Y
23:16	R/W	0xff	reg_MATNR_CBeta_scale_max : U8, Final step Beta scale high limit for Chrm channel.Similar to Y
15: 8	R/W	0x20	reg_MATNR_CBeta_scale_gain : U8, Final step Beta scale Gain for Chrm channel Similar to Y

Bit(s)	R/W	Default	Description
7: 0	R/W	0x0	reg_MATNR_CBeta_scale_ofst : S8, Final step Beta scale offset for Chrm channel Similar to Y

NR2_SNR_MASK 0x175d

Bit(s)	R/W	Default	Description
20:0	R/W	0x0	SAD_MSK : Valid signal in the 3x7 SAD surface

NR2_SAD2NORM_LUT0 0x175e

Bit(s)	R/W	Default	Description
31:24	R/W	0x72	reg_MATNR_SAD2Norm_LUT_3 : SAD convert normal LUT node 3
23:16	R/W	0x92	reg_MATNR_SAD2Norm_LUT_2 : SAD convert normal LUT node 2
15: 8	R/W	0xab	reg_MATNR_SAD2Norm_LUT_1 : SAD convert normal LUT node 1
7: 0	R/W	0xcd	reg_MATNR_SAD2Norm_LUT_0 : SAD convert normal LUT node 0

NR2_SAD2NORM_LUT1 0x175f

Bit(s)	R/W	Default	Description
31:24	R/W	0x1c	reg_MATNR_SAD2Norm_LUT_7 : SAD convert normal LUT node 7
23:16	R/W	0x23	reg_MATNR_SAD2Norm_LUT_6 : SAD convert normal LUT node 6
15: 8	R/W	0x31	reg_MATNR_SAD2Norm_LUT_5 : SAD convert normal LUT node 5
7: 0	R/W	0x4f	reg_MATNR_SAD2Norm_LUT_4 : SAD convert normal LUT node 4

NR2_SAD2NORM_LUT2 0x1760

Bit(s)	R/W	Default	Description
31:24	R/W	0xf	reg_MATNR_SAD2Norm_LUT_11 : SAD convert normal LUT node 11
23:16	R/W	0x11	reg_MATNR_SAD2Norm_LUT_10 : SAD convert normal LUT node 10
15: 8	R/W	0x13	reg_MATNR_SAD2Norm_LUT_9 : SAD convert normal LUT node 9
7: 0	R/W	0x17	reg_MATNR_SAD2Norm_LUT_8 : SAD convert normal LUT node 8

NR2_SAD2NORM_LUT3 0x1761

Bit(s)	R/W	Default	Description
31:24	R/W	0x8	reg_MATNR_SAD2Norm_LUT_15 : SAD convert normal LUT node 15
23:16	R/W	0x9	reg_MATNR_SAD2Norm_LUT_14 : SAD convert normal LUT node 14
15:8	R/W	0xa	reg_MATNR_SAD2Norm_LUT_13 : SAD convert normal LUT node 13
7:0	R/W	0xc	reg_MATNR_SAD2Norm_LUT_12 : SAD convert normal LUT node 12

NR2_EDGE2BETA_LUT0 0x1762

Bit(s)	R/W	Default	Description
31:24	R/W	0x80	reg_MATNR_Edge2Beta_LUT_3 : Edge convert beta LUT node 3
23:16	R/W	0xa0	reg_MATNR_Edge2Beta_LUT_2 : Edge convert beta LUT node 2
15: 8	R/W	0xe0	reg_MATNR_Edge2Beta_LUT_1 : Edge convert beta LUT node 1
7: 0	R/W	0xff	reg_MATNR_Edge2Beta_LUT_0 : Edge convert beta LUT node 0

NR2_EDGE2BETA_LUT1 0x1763

Bit(s)	R/W	Default	Description
31:24	R/W	0x4	reg_MATNR_Edge2Beta_LUT_7 : Edge convert beta LUT node 7
23:16	R/W	0x10	reg_MATNR_Edge2Beta_LUT_6 : Edge convert beta LUT node 6
15: 8	R/W	0x20	reg_MATNR_Edge2Beta_LUT_5 : Edge convert beta LUT node 5
7: 0	R/W	0x50	reg_MATNR_Edge2Beta_LUT_4 : Edge convert beta LUT node 4

NR2_EDGE2BETA_LUT2 0x1a64

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_MATNR_Edge2Beta_LUT_11 : Edge convert beta LUT node 11
23:16	R/W	0x0	reg_MATNR_Edge2Beta_LUT_10 : Edge convert beta LUT node 10
15: 8	R/W	0x0	reg_MATNR_Edge2Beta_LUT_9 : Edge convert beta LUT node 9
7: 0	R/W	0x2	reg_MATNR_Edge2Beta_LUT_8 : Edge convert beta LUT node 8

NR2_EDGE2BETA_LUT3 0x1765

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_MATNR_Edge2Beta_LUT_15 : Edge convert beta LUT node 15
23:16	R/W	0x0	reg_MATNR_Edge2Beta_LUT_14 : Edge convert beta LUT node 14
15: 8	R/W	0x0	reg_MATNR_Edge2Beta_LUT_13 : Edge convert beta LUT node 13
7: 0	R/W	0x0	reg_MATNR_Edge2Beta_LUT_12 : Edge convert beta LUT node 12

NR2_MOTION2BETA_LUT0 0x1766

Bit(s)	R/W	Default	Description
31:24	R/W	0x20	reg_MATNR_Mot2Beta_LUT_3 : Motion convert beta LUT node 3
23:16	R/W	0x10	reg_MATNR_Mot2Beta_LUT_2 : Motion convert beta LUT node 2
15: 8	R/W	0x4	reg_MATNR_Mot2Beta_LUT_1 : Motion convert beta LUT node 1
7: 0	R/W	0x0	reg_MATNR_Mot2Beta_LUT_0 : Motion convert beta LUT node 0

NR2_MOTION2BETA_LUT1 0x1767

Bit(s)	R/W	Default	Description
31:24	R/W	0xc4	reg_MATNR_Mot2Beta_LUT_7 : Motion convert beta LUT node 7
23:16	R/W	0x80	reg_MATNR_Mot2Beta_LUT_6 : Motion convert beta LUT node 6
15: 8	R/W	0x40	reg_MATNR_Mot2Beta_LUT_5 : Motion convert beta LUT node 5
7: 0	R/W	0x30	reg_MATNR_Mot2Beta_LUT_4 : Motion convert beta LUT node 4

NR2_MOTION2BETA_LUT2 0x1768

Bit(s)	R/W	Default	Description
31:24	R/W	0xff	reg_MATNR_Mot2Beta_LUT_11 : Motion convert beta LUT node 11
23:16	R/W	0xff	reg_MATNR_Mot2Beta_LUT_10 : Motion convert beta LUT node 10
15: 8	R/W	0xf0	reg_MATNR_Mot2Beta_LUT_9 : Motion convert beta LUT node 9
7: 0	R/W	0xe0	reg_MATNR_Mot2Beta_LUT_8 : Motion convert beta LUT node 8

NR2_MOTION2BETA_LUT3 0x1769

Bit(s)	R/W	Default	Description
31:24	R/W	0xff	reg_MATNR_Mot2Beta_LUT_15 : Motion convert beta LUT node 15
23:16	R/W	0xff	reg_MATNR_Mot2Beta_LUT_14 : Motion convert beta LUT node 14
15: 8	R/W	0xff	reg_MATNR_Mot2Beta_LUT_13 : Motion convert beta LUT node 13
7: 0	R/W	0xff	reg_MATNR_Mot2Beta_LUT_12 : Motion convert beta LUT node 12

NR2_MATNR_MTN_CRTL 0x176a

Bit(s)	R/W	Default	Description
25:24	R/W	0x0	reg_MATNR_Vmtn_use_mode : Motion_yuvV channel motion selection mode:0: Vmot;1:Ymot/2 + (Umot+Vmot)/4; 2:Ymot/2 + max(Umot,Vmot)/2; 3: max(Ymot,Umot, Vmot)
21:20	R/W	0x0	reg_MATNR_Umtn_use_mode : Motion_yuvU channel motion selection mode:0:Umot;1:Ymot/2 + (Umot+Vmot)/4; 2:Ymot/2 + max(Umot,Vmot)/2; 3: max(Ymot,Umot, Vmot)
17:16	R/W	0x0	reg_MATNR_Ymtn_use_mode : Motion_yuvLuma channel motion selection mode:0: Ymot, 1: Ymot/2 + (Umot+Vmot)/4; 2: Ymot/2 + max(Umot,Vmot)/2; 3: max(Ymot,Umot, Vmot)
13:12	R/W	0x1	reg_MATNR_mtn_txt_mode : Texture detection mode for adaptive coring of HP motion
9: 8	R/W	0x1	reg_MATNR_mtn_cor_mode : Coring selection mode based on texture detection;
6: 4	R/W	0x8	reg_MATNR_mtn_hpf_mode : video mode of current and previous frame/field for MotHPF_yuv[k] calculation:

2: 0	R/W	0x6	reg_MATNR_mtn_lpf_mode : LPF video mode of current and previous frame/field for MotLPF_yuv[k] calculation:

NR2_MATNR_MTN_CRTL2 0x176b

Bit(s)	R/W	Default	Description
18:16	R/W	0x6	reg_MATNR_iir_BS_Ymode : IIR TNR filter Band split filter mode for Luma LPF result generation (Cur and Prev);
15: 8	R/W	0x40	reg_MATNR_mtnb_alpLP_Cgain : Scale of motion_brthp_uv to motion_brtlp_uv, normalized to 32 as 1
7: 0	R/W	0x40	reg_MATNR_mtnb_alpLP_Ygain : Scale of motion_brthp_y to motion_brtlp_y, normalized to 32 as 1

NR2_MATNR_MTN_COR 0x176c

Bit(s)	R/W	Default	Description
15:12	R/W	0x3	reg_MATNR_mtn_cor_Cofst : Coring Offset for Chroma Motion.
11: 8	R/W	0x3	reg_MATNR_mtn_cor_Cgain : Gain to texture based coring for Chroma Motion. Normalized to 16 as 1
7: 4	R/W	0x3	reg_MATNR_mtn_cor_Yofst : Coring Offset for Luma Motion.
3: 0	R/W	0x3	reg_MATNR_mtn_cor_Ygain : Gain to texture based coring for Luma Motion. Normalized to 16 as 1

NR2_MATNR_MTN_GAIN 0x176d

Bit(s)	R/W	Default	Description
31:24	R/W	0x40	reg_MATNR_mtn_hp_Cgain : Gain to MotHPF_yuv[k] Chrm channel for motion calculation, normalized to 64 as 1
23:16	R/W	0x40	reg_MATNR_mtn_hp_Ygain : Gain to MotHPF_yuv[k] Luma channel for motion calculation, normalized to 64 as 1
15: 8	R/W	0x40	reg_MATNR_mtn_lp_Cgain : Gain to MotLPF_yuv[k] Chrm channel for motion calculation, normalized to 32 as 1
7: 0	R/W	0x40	reg_MATNR_mtn_lp_Ygain : Gain to MotLPF_yuv[k] Luma channel for motion calculation, normalized to 32 as 1

NR2_MATNR_DEGHOST 0x176e

Bit(s)	R/W	Default	Description
30:28	R/W	0	reg_matnr_deghost_mode : // unsigned , default = 0 0:old_deghost; 1:soft_denoise & strong_deghost; 2:strong_denoise & soft_deghost; 3:strong_denoise & strong_deghost
24:20	R/W	4	reg_matnr_deghost_ygain : // unsigned , default = 4
16:12	R/W	4	reg_matnr_deghost_cgain : // unsigned , default = 4

8	R/W	1	reg_matnr_deghost_en : // unsigned , default = 1 0: disable; 1: enable Enable signal for DeGhost function:0: disable; 1: enable
7: 4	R/W	3	reg_matnr_deghost_cos : // unsigned , default = 3 DeGhost Overshoot margin for UV channel, (X2 to u10 scale)
3: 0	R/W	3	reg_matnr_deghost_yos : // unsigned , default = 3 DeGhost Overshoot margin for Luma channel, (X2 to u10 scale)

NR2_MATNR_ALPHALP_LUT0 0x176f

Bit(s)	R/W	Default	Description
31:24	R/W	0x40	reg_MATNR_AlphaLP_LUT_3 : Matnr low-pass filter alpha LUT node 3
23:16	R/W	0x80	reg_MATNR_AlphaLP_LUT_2 : Matnr low-pass filter alpha LUT node 2
15: 8	R/W	0x80	reg_MATNR_AlphaLP_LUT_1 : Matnr low-pass filter alpha LUT node 1
7: 0	R/W	0x80	reg_MATNR_AlphaLP_LUT_0 : Matnr low-pass filter alpha LUT node 0

NR2_MATNR_ALPHALP_LUT1 0x1770

Bit(s)	R/W	Default	Description
31:24	R/W	0xff	reg_MATNR_AlphaLP_LUT_7 : Matnr low-pass filter alpha LUT node 7
23:16	R/W	0x80	reg_MATNR_AlphaLP_LUT_6 : Matnr low-pass filter alpha LUT node 6
15: 8	R/W	0x50	reg_MATNR_AlphaLP_LUT_5 : Matnr low-pass filter alpha LUT node 5
7: 0	R/W	0x40	reg_MATNR_AlphaLP_LUT_4 : Matnr low-pass filter alpha LUT node 4

NR2_MATNR_ALPHALP_LUT2 0x1771

Bit(s)	R/W	Default	Description
31:24	R/W	0xff	reg_MATNR_AlphaLP_LUT_11 : Matnr low-pass filter alpha LUT node 11
23:16	R/W	0xff	reg_MATNR_AlphaLP_LUT_10 : Matnr low-pass filter alpha LUT node 10
15: 8	R/W	0xff	reg_MATNR_AlphaLP_LUT_9 : Matnr low-pass filter alpha LUT node 9
7: 0	R/W	0xff	reg_MATNR_AlphaLP_LUT_8 : Matnr low-pass filter alpha LUT node 8

NR2_MATNR_ALPHALP_LUT3 0x1772

Bit(s)	R/W	Default	Description
31:24	R/W	0xff	reg_MATNR_AlphaLP_LUT_15 : Matnr low-pass filter alpha LUT node 15
23:16	R/W	0xff	reg_MATNR_AlphaLP_LUT_14 : Matnr low-pass filter alpha LUT node 14
15: 8	R/W	0xff	reg_MATNR_AlphaLP_LUT_13 : Matnr low-pass filter alpha LUT node 13
7: 0	R/W	0xff	reg_MATNR_AlphaLP_LUT_12 : Matnr low-pass filter alpha LUT node 12

NR2_MATNR_ALPHAHP_LUT0 0x1773

Bit(s)	R/W	Default	Description
31:24	R/W	0x40	reg_MATNR_AlphaHP_LUT_3 : Matnr high-pass filter alpha LUT node 3
23:16	R/W	0x80	reg_MATNR_AlphaHP_LUT_2 : Matnr high-pass filter alpha LUT node 2
15: 8	R/W	0x80	reg_MATNR_AlphaHP_LUT_1 : Matnr high-pass filter alpha LUT node 1
7: 0	R/W	0x80	reg_MATNR_AlphaHP_LUT_0 : Matnr high-pass filter alpha LUT node 0

NR2_MATNR_ALPHAHP_LUT1 0x1774

Bit(s)	R/W	Default	Description
31:24	R/W	0xff	reg_MATNR_AlphaHP_LUT_7 : Matnr high-pass filter alpha LUT node 7
23:16	R/W	0x80	reg_MATNR_AlphaHP_LUT_6 : Matnr high-pass filter alpha LUT node 6
15: 8	R/W	0x50	reg_MATNR_AlphaHP_LUT_5 : Matnr high-pass filter alpha LUT node 5
7: 0	R/W	0x40	reg_MATNR_AlphaHP_LUT_4 : Matnr high-pass filter alpha LUT node 4

NR2_MATNR_ALPHAHP_LUT2 0x1775

Bit(s)	R/W	Default	Description
31:24	R/W	0xff	reg_MATNR_AlphaHP_LUT_11 : Matnr high-pass filter alpha LUT node 11
23:16	R/W	0xff	reg_MATNR_AlphaHP_LUT_10 : Matnr high-pass filter alpha LUT node 10
15: 8	R/W	0xff	reg_MATNR_AlphaHP_LUT_9 : Matnr high-pass filter alpha LUT node 9
7: 0	R/W	0xff	reg_MATNR_AlphaHP_LUT_8 : Matnr high-pass filter alpha LUT node 8

NR2_MATNR_ALPHAHP_LUT3 0x1776

Bit(s)	R/W	Default	Description
31:24	R/W	0xff	reg_MATNR_AlphaHP_LUT_15 : Matnr high-pass filter alpha LUT node 15
23:16	R/W	0xff	reg_MATNR_AlphaHP_LUT_14 : Matnr high-pass filter alpha LUT node 14
15: 8	R/W	0xff	reg_MATNR_AlphaHP_LUT_13 : Matnr high-pass filter alpha LUT node 13
7: 0	R/W	0xff	reg_MATNR_AlphaHP_LUT_12 : Matnr high-pass filter alpha LUT node 12

NR2_MATNR_MTNB_BRT 0x1777

Bit(s)	R/W	Default	Description
31:28	R/W	0x0	reg_MATNR_mtnb_BRT_limt_hi : Motion adjustment based on Brightness high side Limit. (X16 to u8 scale)
27:24	R/W	0x0	reg_MATNR_mtnb_BRT_slop_hi : Motion adjustment based on Brightness high side slope. Normalized to 16 as 1

23:16	R/W	0xa0	reg_MATNR_mtnb_BRT_thrd_hi : Motion adjustment based on Brightness high threshold.(u8 scale)
15:12	R/W	0x6	reg_MATNR_mtnb_BRT_limt_lo : Motion adjustment based on Brightness low side Limit. (X16 to u8 scale)
11: 8	R/W	0x6	reg_MATNR_mtnb_BRT_slop_lo : Motion adjustment based on Brightness low side slope. Normalized to 16 as 1
7: 0	R/W	0x64	reg_MATNR_mtnb_BRT_thrd_lo : Motion adjustment based on Brightness low threshold.(u8 scale)

DET 3D REG DEFINE BEGIN //// 8 'h80~0x8f

DET3D_RO_SPLT_HB 0x1780

Bit(s)	R/W	Default	Description
24	R.O	0x0	RO_Det3D_Split_HB_valid : U1 horizontal LR split border detected valid signal for top half picture
20:16	R.O	0x0	RO_Det3D_Split_HB_pxnum : U5 number of pixels included for the LR split position estimation for top half picture
9: 0	R.O	0x0	RO_Det3D_Split_HB_idxX4 : S10 X4: horizontal pixel shifts of LR split position to the (ColMax/2) for top half picture

DET3D_RO_SPLT_VL 0x1781

Bit(s)	R/W	Default	Description
24	R.O	0x0	RO_Det3D_Split_VL_valid : U1 horizontal LR split border detected valid signal for top half picture
20:16	R.O	0x0	RO_Det3D_Split_VL_pxnum : U5 number of pixels included for the LR split position estimation for top half picture
9: 0	R.O	0x0	RO_Det3D_Split_VL_idxX4 : S10 X4: horizontal pixel shifts of LR split position to the (ColMax/2) for top half picture

DET3D_RO_SPLT_VR 0x1782

Bit(s)	R/W	Default	Description
24	R.O	0x0	RO_Det3D_Split_VR_valid : U1 horizontal LR split border detected valid signal for top half picture
20:16	R.O	0x0	RO_Det3D_Split_VR_pxnum : U5 number of pixels included for the LR split position estimation for top half picture
9: 0	R.O	0x0	RO_Det3D_Split_VR_idxX4 : S10 X4: horizontal pixel shifts of LR split position to the (ColMax/2) for top half picture

DET3D_RO_MAT_LUMA_LR 0x1783

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_Luma_LR_score : S2*8 LUMA statistics left right decision score for each band (8bands vertically), it can be -1/0/1:-1: most likely not LR symmetric 0: not sure 1: most likely LR symmetric

7:0	R.O	0x0	RO_Luma_LR_symtc : U1*8 Luma statistics left right pure symmetric for each band (8bands vertically), it can be 0/1: 0: not sure 1: most likely LR is pure symmetric
4:0	R.O	0x0	RO_Luma_LR_sum : S5 Total score of 8x8 Luma statistics for LR like decision, the larger this score, the more confidence that this is a LR 3D video. It is sum of RO_Luma_LR_score[0~7]

DET3D_RO_MAT_LUMA_TB 0x1784

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_Luma_TB_score : S2*8 LUMA statistics Top/Bottom decision score for each band (8bands Horizontally),
7:0	R.O	0x0	RO_Luma_TB_symtc : Luma statistics Top/Bottompure symmetric for each band (8bands Horizontally),
4:0	R.O	0x0	RO_Luma_TB_sum : Total score of 8x8 Luma statistics for TB like decision,

DET3D_RO_MAT_CHRU_LR 0x1785

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_ChrU_LR_score : S2*8 LUMA statistics left right decision score for each band (8bands vertically),
7:0	R.O	0x0	RO_ChrU_LR_symtc : CHRU statistics left right pure symmetric for each band (8bands vertically),
4:0	R.O	0x0	RO_ChrU_LR_sum : Total score of 8x8 ChrU statistics for LR like decision,

DET3D_RO_MAT_CHRU_TB 0x1786

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_ChrU_TB_score : S2*8 CHRU statistics Top/Bottom decision score for each band (8bands Horizontally)
7:0	R.O	0x0	RO_ChrU_TB_symtc : CHRU statistics Top/Bottompure symmetric for each band (8bands Horizontally)
4:0	R.O	0x0	RO_ChrU_TB_sum : Total score of 8x8 ChrU statistics for TB like decision

DET3D_RO_MAT_CHRV_LR 0x1787

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_ChrV_LR_score : S2*8 CHRUstatistics left right decision score for each band (8bands vertically)
7:0	R.O	0x0	RO_ChrV_LR_symtc : CHRV statistics left right pure symmetric for each band (8bands vertically)
4:0	R.O	0x0	RO_ChrV_LR_sum : Total score of 8x8 ChrV statistics for LR like decision

DET3D_RO_MAT_CHRV_TB 0x1788

Bit(s)	R/W	Default	Description

15:0	R.O	0x0	RO_ChrV_TB_score : CHRV statistics Top/Bottom decision score for each band (8bands Horizontally)
7:0	R.O	0x0	RO_ChrV_TB_symtc : CHRV statistics Top/Bottompure symmetric for each band (8bands Horizontally)
4:0	R.O	0x0	RO_ChrV_TB_sum : Total score of 8x8 ChrV statistics for TB like decision

DET3D_RO_MAT_HEDG_LR 0x1789

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_Hedg_LR_score : Horizontal Edge statistics left right decision score for each band (8bands vertically)
7:0	R.O	0x0	RO_Hedg_LR_symtc : Horizontal Edge statistics left right pure symmetric for each band (8bands vertically)
4:0	R.O	0x0	RO_Hedg_LR_sum : Total score of 8x8 Hedg statistics for LR like decision

DET3D_RO_MAT_HEDG_TB 0x178a

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_Hedg_TB_score : Horizontal Edge statistics Top/Bottom decision score for each band (8bands Horizontally)
7:0	R.O	0x0	RO_Hedg_TB_symtc : Horizontal Edge statistics Top/Bottompure symmetric for each band (8bands Horizontally)
4:0	R.O	0x0	RO_Hedg_TB_sum : Total score of 8x8 Hedg statistics for TB like decision

DET3D_RO_MAT_VEDG_LR 0x178b

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_Vedg_LR_score : Vertical Edge statistics left right decision score for each band (8bands vertically)
7:0	R.O	0x0	RO_Vedg_LR_symtc : Vertical Edge statistics left right pure symmetric for each band (8bands vertically)
4:0	R.O	0x0	RO_Vedg_LR_sum : Total score of 8x8 Vedg statistics for LR like decision

DET3D_RO_MAT_VEDG_TB 0x178c

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_Vedg_TB_score : Vertical Edge statistics Top/Bottom decision score for each band (8bands Horizontally)
7:0	R.O	0x0	RO_Vedg_TB_symtc : Vertical Edge statistics Top/Bottompure symmetric for each band (8bands Horizontally)
4:0	R.O	0x0	RO_Vedg_TB_sum : Total score of 8x8 Vedg statistics for TB like decision

DET3D_RO_MAT_MOTN_LR 0x178d

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_Motn_LR_score : Motion statistics left right decision score for each band (8bands vertically)
7:0	R.O	0x0	RO_Motn_LR_symtc : Motion statistics left right pure symmetric for each band (8bands vertically)
4:0	R.O	0x0	RO_Motn_LR_sum : Total score of 8x8 Motion statistics for LR like decision

DET3D_RO_MAT_MOTN_TB 0x178e

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_Motn_TB_score : Motion statistics Top/Bottom decision score for each band (8bands Horizontally)
7:0	R.O	0x0	RO_Motn_TB_symtc : Motion statistics Top/Bottompure symmetric for each band (8bands Horizontally)
4:0	R.O	0x0	RO_Motn_TB_sum : Total score of 8x8 Motion statistics for TB like decision

DET3D_RO_FRM_MOTN 0x178f

Bit(s)	R/W	Default	Description
15:0	R.O	0x0	RO_Det3D_Frame_Motion : U16 frame based motion value sum for still image decision in FW. mat ram read enter addr

DET3D_RAMRD_ADDR_PORT 0x179a

DET3D_RAMRD_DATA_PORT 0x179b

NR2_CFR_PARA_CFG0 0x179c

Bit(s)	R/W	Default	Description
8	R/W	0x0	reg_CFR_CurDif_luma_mode : Current Field Top/Bot line Luma difference calculation mode
7:6	R/W	0x0	reg_MACFR_frm_phase : U2 This will be a field based phase register that need to be set by FW phase to phase: this will be calculated based on dbdr_phase of the specific line of this frame. u1 : dbdr_phase=1, center line is DB in current line; dbdr_phase=2, center line is Dr in current line;
5:4	R/W	0x0	reg_CFR_CurDif_tran_mode : U2 Current Field Top/Bot line Luma/Chroma transition level calculation mode,
3:2	R/W	0x0	reg_CFR_alpha_mode : U2 Alpha selection mode for CFR block from curAlp and motAlp i.e. 0: motAlp; 1: (motAlp+curAlp)/2; 2: min(motAlp,curAlp); 3: max(motAlp,curAlp);
1:0	R/W	0x0	reg_CFR_Motion_Luma_mode : U2 LumaMotion Calculation mode for MA-CFR. 0: top/bot Lumma motion; 1: middle Luma Motion 2: top/bot + middle motion; 3: max(top/tot motion, middle motion)

NR2_CFR_PARA_CFG1 0x179d

Bit(s)	R/W	Default	Description
23:16	R/W	0x0	reg_CFR_alpha_gain : gain to map muxed curAlp and motAlp to alpha that will be used for final blending.
15: 8	R/W	0x0	reg_CFR_Motion_ofst : Offset to Motion to calculate the motAlp, e,g:motAlp= reg_CFR_Motion_ofst- Motion;This register can be seen as the level of motion that we consider it at moving.
7: 0	R/W	0x0	reg_CFR_CurDif_gain : gain to CurDif to map to alpha, normalized to 32;

NR3_MODE 0x2ff0

Bit(s)	R/W	Default	Description
5	R/W	0x0	reg_3dnr_nr3_vtxt_mode : ;
4	R/W	0x0	reg_3dnr_nr3_cbyy_ignor_coop : ; // u1: ignore coop condition for cbyy motion decision
3	R/W	0x0	reg_3dnr_nr3_ybyc_ignor_cnoop : ; // u1: ignore cnoop condition for ybyc motion decision
2:0	R/W	0x3	reg_3dnr_nr3_suremot_txt_mode : ; // u3: 0: cur, 1:p2; 2: (cur+p2)/2; 3/up: min(cur,p2)

NR3_COOP_PARA 0x2ff1

Bit(s)	R/W	Default	Description
21:20	R/W	0x2	reg_3dnr_nr3_coop_mode : ; // u2 0 original pixel 1: [1 2 1]/4 lpf; 2: [1 2 2 2 1]/8; 3: 3x3 lpf
19:16	R/W	0x8	reg_3dnr_nr3_coop_ratio : ; // u4 cur and p2 color oop decision ratio: (avg1<(MAX(sat0,sat2)*ratio/8 + ofst));
15:8	R/W	0x0	reg_3dnr_nr3_coop_ofset : ; // s8 cur and p2 color oop decision ofst: (avg1<(MAX(sat0,sat2)*ratio/8 + ofst));
7:0	R/W	0x0	reg_3dnr_nr3_coop_sat_thrd : ; // u8 cur and p2 color oop decision min(sat0,sat1) threshold;

NR3_CNOOP_GAIN 0x2ff2

Bit(s)	R/W	Default	Description
23:20	R/W	0x8	reg_3dnr_nr3_cnoop_ratio0 : ; // u4 cur and p2 color noop decision ratio0: (avg1<(MAX(sat0,sat2)*ratio0/8 + ofst0));
19:16	R/W	0x8	reg_3dnr_nr3_cnoop_ratio1 : ; // u4 cur and p2 color noop decision ratio1: (dif1<(MIN(sat0,sat2)*ratio1/8 + ofst1));
15:8	R/W	0x19	reg_3dnr_nr3_cnoop_ofset0 : ; // s8 cur and p2 color noop decision ofset0: (avg1<(MAX(sat0,sat2)*ratio0/8 + ofst0));
7:0	R/W	0x0	reg_3dnr_nr3_cnoop_ofset1 : ; // s8 cur and p2 color noop decision ofset1: (dif1<(MIN(sat0,sat2)*ratio1/8 + ofst1));

NR3_YMOT_PARA 0x2ff3

Bit(s)	R/W	Default	Description
19	R/W	0x1	reg_3dnr_nr3_ymot_only_en : ; // u1: enable signal for ignor chroma motion: (ytxt &coop)
18	R/W	0x1	reg_3dnr_nr3_ymot_only_cmtmode : ; // u1: 0: cmot=ymot; 1: cmot = MIN(ymot, cmot)
17:16	R/W	0x0	reg_3dnr_nr3_ymot_only_txtmode : ; // u2: 0, min(txt0,txt2); 1, max(txt0,txt2);2, (txt0+txt2)/2; 3: sat(txt0, txt2)
15:8	R/W	0xa	reg_3dnr_nr3_ymot_only_txtthrd : ; // u8: threshold to luma texture to decide use ymot only
7:0	R/W	0x1e	reg_3dnr_nr3_ymot_only_motthrd : ; // u8: threshold to luma motion to decide use ymot only

NR3_CMOT_PARA 0x2ff4

Bit(s)	R/W	Default	Description
19	R/W	0x1	reg_3dnr_nr3_cmot_only_en : ; // u1: enable signal for ignor luma motion: (ctxt &cnoop)
18	R/W	0x0	reg_3dnr_nr3_cmot_only_ymtmode : ; // u1: 0: ymot=cmot+ymot/4; 1: ymot = MIN(ymot, cmot)
17:16	R/W	0x0	reg_3dnr_nr3_cmot_only_txtmode : ; // u2: 0, min(txt0,txt2); 1, max(txt0,txt2);2, (txt0+txt2)/2; 3: sat(txt0, txt2)
15:8	R/W	0x14	reg_3dnr_nr3_cmot_only_txtthrd : ; // u8: threshold to chroma texture to decide use cmot only
7:0	R/W	0xf	reg_3dnr_nr3_cmot_only_motthrd : ; // u8: threshold to chroma motion to decide use cmot only

NR3_SUREMOT_YGAIN 0x2ff5

Bit(s)	R/W	Default	Description
31:24	R/W	0x10	reg_3dnr_nr3_suremot_dec_yrate : ; // u8: (norm 16)lpfMot>(dec_rate*txt +ofst) then force lpfMot*frc_gain+frc_ofset
23:16	R/W	0xc	reg_3dnr_nr3_suremot_dec_yofst : ; // u8: lpfMot>(dec_rate*txt +ofst) then force lpfMot*frc_gain+frc_ofset
15:8	R/W	0x40	reg_3dnr_nr3_suremot_frc_ygain : ; // u8: (norm 8)lpfMot>(dec_rate*txt +ofst) then force lpfMot*frc_gain+frc_ofset
7:0	R/W	0x14	reg_3dnr_nr3_suremot_frc_yofst : ; // u8: lpfMot>(dec_rate*txt +ofst) then force lpfMot*frc_gain+frc_ofset

NR3_SUREMOT_CGAIN 0x2ff6

Bit(s)	R/W	Default	Description
31:24	R/W	0x22	reg_3dnr_nr3_suremot_dec_crate : ; // u8: (norm 16)lpfMot>(dec_rate*txt +ofst) then force lpfMot*frc_gain+frc_ofset
23:16	R/W	0x26	reg_3dnr_nr3_suremot_dec_cofst : ; // u8: lpfMot>(dec_rate*txt +ofst) then force lpfMot*frc_gain+frc_ofset

Bit(s)	R/W	Default	Description
15:8	R/W	0x40	reg_3dnr_nr3_suremot_frc_cgain : ; // u8: (norm 8)lpfMot>(dec_rate*txt +ofst) then force lpfMot*frc_gain+frc_ofset
7:0	R/W	0x14	reg_3dnr_nr3_suremot_frc_cofst : ; // u8: lpfMot>(dec_rate*txt +ofst) then force lpfMot*frc_gain+frc_ofset

10.2.3.9 LBUF Registers

LBUF_TOP_CTRL 0x2fff

Bit(s)	R/W	Default	Description
25:20	R/W	6'd0	gate clk control of line buf . LBUF_TOP_CTRL[25:24] is the clk control of current line linebuffer. LBUF_TOP_CTRL[23:22] is the clk control of previous one line.linebuffer, LBUF_TOP_CTRL[21:20] is the clk control of previous two line linebuffer.
17	R/W	1'b1	lbuf_fmt444_mode; format of data store in linebuf ,high mean store 444 data into linebuf
16	R/W	1'b1	lbuf_line5_mode;Store 5 line or 3 lines in linebuf ,high means 5 lines
12:0	R/W	13'd342	pre_lbuf_size: size of linebuf

10.2.3.10 DI_SCALE & HDR Registers

DI_SCO_FIFO_CTRL 0x374e

Bit(s)	R/W	Default	Description
28:0	R/W	0x0	sco_fifo_ctrl

DI_SC_TOP_CTRL 0x374f

Bit(s)	R/W	Default	Description
31	R.O	0	prog_interlace : // unsigned , default = 0x0
30	R/W	0	path_sel : // unsigned , default = 0x0
29	R/W	0	go_field_sel : // unsigned , default = 0x0
28	R/W	0	sw_resets : // unsigned , default = 0x0
27	R/W	0	pps_dummy_data_mode : // unsigned , default = 0x0
26	R/W	0	field_inv : // unsigned , default = 0x0
25	R/W	0	reg_field : // unsigned , default = 0x0
5:4	R/W	0	hdr_gclk_ctrl : // unsigned , default = 0x0
1	R/W	0	reg_gclk_ctrl : // unsigned , default = 0x0

DI_SC_DUMMY_DATA 0x3750

Bit(s)	R/W	Default	Description
29:20	R/W	0	VD1_SC_Y : // unsigned , default = 0x10,dummy data used in the VD1 scaler,according VPP_DOLBY_CTRL[17] 1:set 8bit value 2:set 10bit value
19:10	R/W	0	VD1_SC_CB : // unsigned , default = 0x80,dummy data used in the VD1 scaler,according VPP_DOLBY_CTRL[17] 1:set 8bit value 2:set 10bit value
9 :0	R/W	0	VD1_SC_CR : // unsigned , default = 0x80,dummy data used in the VD1 scaler,according VPP_DOLBY_CTRL[17] 1:set 8bit value 2:set 10bit value

DI_SC_LINE_IN_LENGTH 0x3751

Bit(s)	R/W	Default	Description
13:0	R/W	14	line_in_length : // unsigned , default = 14'd1920,VD1 scaler input hsize

DI_SC_PIC_IN_HEIGHT 0x3752

Bit(s)	R/W	Default	Description
12:0	R/W	0x1fff	line_in_height : // unsigned , default = 13'h1fff,VD1 scaler input vsize

DI_SC_COEF_IDX 0x3753

Bit(s)	R/W	Default	Description
15	R/W	0	index_inc : // unsigned , default = 0x0 ,index increment, if bit9 = 1 then (0: index increase 1, 1: index increase 2) else (index increase 2)
14	R/W	0	rd_cbus_coef_en : // unsigned , default = 0x0 ,1: read coef through cbus enable, just for debug purpose in case when we wanna check the coef in ram in correct or not
13	R/W	0	vf_sep_coef_en : // unsigned , default = 0x0 ,if true, vertical separated coef enable
9	R/W	0	high_reso_en : // unsigned , default = 0x0 ,if true, use 9bit resolution coef, other use 8bit resolution coef
8:7	R/W	0	type_index : // unsigned , default = 0x0 ,type of index, 00: vertical coef, 01: vertical chroma coef: 10: horizontal coef, 11: resevered
6:0	R/W	0	coef_index : // unsigned , default = 0x0 ,coef index

DI_SC_COEF 0x3754

Bit(s)	R/W	Default	Description
31:24	R/W	0	coef0 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter
23:16	R/W	0	coef1 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter
15:8	R/W	0	coef2 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter
7 :0	R/W	0	coef3 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter

DI_VSC_REGION12_STARTP 0x3755

Bit(s)	R/W	Default	Description
28:16	R/W	0	region1_startp : //unsigned , default = 0 ,region1 startp
12:0	R/W	0	region2_startp : //unsigned , default = 0 ,region2 startp

DI_VSC_REGION34_STARTP 0x3756

Bit(s)	R/W	Default	Description
28:16	R/W	0	region3_startp : //unsigned , default = 0x0438,region3 startp
12:0	R/W	0	region4_startp : //unsigned , default = 0x0438,region4 startp

DI_VSC_REGION4_ENDP 0x3757

Bit(s)	R/W	Default	Description
12:0	R/W	13	region4_endp : //unsigned , default = 13'd1079 ,region4 endp

DI_VSC_START_PHASE_STEP 0x3758

Bit(s)	R/W	Default	Description
27:24	R/W	1	integer_part : //unsigned , default = 1,vertical start phase step, (source/dest)*(2^24),integer part of step
23:0	R/W	0	fraction_part : //unsigned , default = 0,vertical start phase step, (source/dest)*(2^24),fraction part of step

DI_VSC_REGION0_PHASE_SLOPE 0x3759

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,vertical scaler region0 phase slope,region0 phase slope

DI_VSC_REGION1_PHASE_SLOPE 0x375a

Bit(s)	R/W	Default	Description
24:0	R/W	0	region1_phase_slope : //signed , default = 0,region1 phase slope

DI_VSC_REGION3_PHASE_SLOPE 0x375b

Bit(s)	R/W	Default	Description
24:0	R/W	0	region3_phase_slope : //signed , default = 0,region3 phase slope

DI_VSC_REGION4_PHASE_SLOPE 0x375c

Bit(s)	R/W	Default	Description
24:0	R/W	0	region4_phase_slope : //signed , default = 0,region4 phase slope

DI_VSC_PHASE_CTRL 0x375d

Bit(s)	R/W	Default	Description
18:17	R/W	0	vsc_double_line_mode : //unsigned , default = 0, double line mode, input/output line width of vscaler becomes 2X, so only 2 line buffer in this case, use for 3D line by line interleave scaling bit1 true, double the input width and half input height, bit0 true, change line buffer 2 lines instead of 4 lines
16	R.O	0	prog_interlace : //unsigned , default = 0,0: progressive output, 1: interlace output
15	R/W	0	vsc_bot_l0_out_en : //unsigned , default = 0,vertical scaler output line0 in advance or not for bottom field
14:13	R/W	1	vsc_bot_rpt_l0_num : //unsigned , default = 1,vertical scaler initial repeat line0 number for bottom field
11:8	R/W	4	vsc_bot_ini_rcv_num : //unsigned , default = 4,vertical scaler initial receiving number for bottom field
7	R/W	0	vsc_top_l0_out_en : //unsigned , default = 0,vertical scaler output line0 in advance or not for top field
6:5	R/W	1	vsc_top_rpt_l0_num : //unsigned , default = 1,vertical scaler initial repeat line0 number for top field
3:0	R/W	4	vsc_top_ini_rcv_num : //unsigned , default = 4,vertical scaler initial receiving number for top field

DI_VSC_INI_PHASE 0x375e

Bit(s)	R/W	Default	Description
31:16	R/W	0	vertical scaler field initial phase for bottom field
15:0	R/W	0	vertical scaler field initial phase for top field

DI_HSC_REGION12_STARTP 0x3760

Bit(s)	R/W	Default	Description
28:16	R/W	0	region1_startp : //unsigned , default = 0,region1 startp
12:0	R/W	0	region2_startp : //unsigned , default = 0,region2 startp

DI_HSC_REGION34_STARTP 0x3761

Bit(s)	R/W	Default	Description
28:16	R/W	0	region3 : startp //unsigned , default = 0x780,region3 startp
12:0	R/W	0	region4 : startp //unsigned , default = 0x780,region4 startp

DI_HSC_REGION4_ENDP 0x3762

Bit(s)	R/W	Default	Description
12:0	R/W	13	region4 : startp //unsigned , default = 13'd1919,region4 startp

DI_HSC_START_PHASE_STEP 0x3763

Bit(s)	R/W	Default	Description
27:24	R/W	1	integer_part : //unsigned , default = 1,integer part of step
23:0	R/W	0	fraction_part : //unsigned , default = 0,fraction part of step

DI_HSC_REGION0_PHASE_SLOPE 0x3764

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region0 phase slope

DI_HSC_REGION1_PHASE_SLOPE 0x3765

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region1 phase slope

DI_HSC_REGION3_PHASE_SLOPE 0x3766

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region3 phase slope

DI_HSC_REGION4_PHASE_SLOPE 0x3767

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region4 phase slope

DI_HSC_PHASE_CTRL 0x3768

Bit(s)	R/W	Default	Description
22:21	R/W	1	hsc_rpt_p0_num0 : //unsigned , default = 1 ,horizontal scaler initial repeat pixel0 number0
19:16	R/W	4	hsc_ini_rcv_num0 : //unsigned , default = 4 ,horizontal scaler initial receiving number0
15:0	R/W	0	hsc_ini_phase0 : //unsigned , default = 0 ,horizontal scaler top field initial phase0

DI_SC_MISC 0x3769

Bit(s)	R/W	Default	Description
22	R/W	0	hsc_len_div2_en : //unsigned , default = 0 ,if true, divide VSC line length 2 as the HSC input length, othwise VSC length length is the same as the VSC line length just for special usage, more flexibilityi
21	R/W	0	lbuf_mode : //unsigned , default = 0 ,if true, prevsc uses lin buffer, otherwise prevsc does not use line buffer, it should be same as prevsc_en
20	R/W	0	prehsc_en : //unsigned , default = 0 ,prehsc_en
19	R/W	0	prevsc_en : //unsigned , default = 0 ,prevsc_en
18	R/W	0	vsc_en : //unsigned , default = 0 ,vsc_en
17	R/W	0	hsc_en : //unsigned , default = 0 ,hsc_en
16	R/W	0	sc_top_en : //unsigned , default = 0 ,scale_top_en
15	R/W	0	sc_vd_en : //unsigned , default = 0 ,video1 scale out enable
12	R/W	1	hsc_nonlinear_4region_en : //unsigned , default = 1 ,if true, region0,region4 are nonlinear regions, otherwise they are not scaling regions, for horizontal scaler
10:8	R/W	0	hsc_bank_length : //unsigned , default = 0 ,horizontal scaler bank length
5	R/W	4	vsc_phase_field_mode : //unsigned , default = 4 ,vertical scaler phase field mode, if true, disable the opposite parity line output, more bandwith needed if output 1080i
4	R/W	0	vsc_nonlinear_4region_en : //unsigned , default = 0 ,if true, region0,region4 are nonlinear regions, otherwise they are not scaling regions, for vertical scaler
2:0	R/W	4	vsc_bank_length : //unsigned , default = 4 ,vertical scaler bank length

DI_HSC_PHASE_CTRL1 0x376a

Bit(s)	R/W	Default	Description
22:21	R/W	1	hsc_rpt_p0_num0 : //unsigned , default = 1 ,horizontal scaler initial repeat pixel0 number0
19:16	R/W	4	hsc_ini_rcv_num0 : //unsigned , default = 4 ,horizontal scaler initial receiving number0
15:0	R/W	0	hsc_ini_phase0 : //unsigned , default = 0 ,horizontal scaler top field initial phase0

VPP_SC_MISC 0x1D19

Bit(s)	R/W	Default	Description
22	R/W	0	hsc_len_div2_en : //unsigned , default = 0 ,if true, divide VSC line length 2 as the HSC input length, othwise VSC length length is the same as the VSC line length just for special usage, more flexibilityi
21	R/W	0	lbuf_mode : //unsigned , default = 0 ,if true, prevsc uses lin buffer, otherwise prevsc does not use line buffer, it should be same as prevsc_en
20	R/W	0	prehsc_en : //unsigned , default = 0 ,prehsc_en

Bit(s)	R/W	Default	Description
19	R/W	0	prevsc_en : //unsigned , default = 0 ,prevsc_en
18	R/W	0	vsc_en : //unsigned , default = 0 ,vsc_en
17	R/W	0	hsc_en : //unsigned , default = 0 ,hsc_en
16	R/W	0	sc_top_en : //unsigned , default = 0 ,scale_top_en
15	R/W	0	sc_vd_en : //unsigned , default = 0 ,video1 scale out enable
12	R/W	1	hsc_nonlinear_4region_en : //unsigned , default = 1 ,if true, region0,region4 are nonlinear regions, otherwise they are not scaling regions, for horizontal scaler
10:8	R/W	0	hsc_bank_length : //unsigned , default = 0 ,horizontal scaler bank length
5	R/W	4	vsc_phase_field_mode : //unsigned , default = 4 ,vertical scaler phase field mode, if true, disable the opposite parity line output, more bandwith needed if output 1080i
4	R/W	0	vsc_nonlinear_4region_en : //unsigned , default = 0 ,if true, region0,region4 are nonlinear regions, otherwise they are not scaling regions, for vertical scaler
2:0	R/W	4	vsc_bank_length : //unsigned , default = 4 ,vertical scaler bank length

DI_HSC_INI_PAT_CTRL 0x376b

Bit(s)	R/W	Default	Description
31:24	R/W	0	prehsc_pattern : //unsigned , default = 0, prehsc pattern, each patten 1 bit, from lsb -> msb
22:20	R/W	0	prehsc_pat_star : //unsigned , default = 0, prehsc pattern start
18:16	R/W	0	prehsc_pat_end : //unsigned , default = 0, prehsc pattern end
15:8	R/W	0	hsc_pattern : //unsigned , default = 0, hsc pattern, each patten 1 bit, from lsb -> msb
6:4	R/W	0	hsc_pat_start : //unsigned , default = 0, hsc pattern start
2:0	R/W	0	hsc_pat_end : //unsigned , default = 0, hsc pattern end

DI_SC_GCLK_CTRL 0x376c

Bit(s)	R/W	Default	Description
15:0	R/W	0	vpp_sc_gclk_ctrl : //unsigned , default = 0,

DI_SC_HOLD_LINE 0x376d

Bit(s)	R/W	Default	Description
31:0	R/W	0	sc_hold_line : //unsigned , default = 0,

DI_HDR_IN_HSIZE 0x376e

Bit(s)	R/W	Default	Description
12:0	R/W	0	hdr input h size

DI_HDR_IN_VSIZE 0x376f

Bit(s)	R/W	Default	Description
12:0	R/W	0	hdr input v size

DI_HDR2_CTRL 0x3770

Bit(s)	R/W	Default	Description
20:18	R/W	0	reg_din_swap : // unsigned , default = 0
17	R/W	0	reg_out_fmt : // unsigned , default = 0
16	R/W	0	reg_only_mat : // unsigned , default = 0
13	R/W	0	reg_VDIN0_HDR2_top_en : // unsigned , default = 0
12	R/W	1	reg_cgain_mode : // unsigned , default = 1
7: 6	R/W	1	reg_gmut_mode : // unsigned , default = 1
5	R/W	0	reg_in_shift : // unsigned , default = 0
4	R/W	1	reg_in_fmt : // unsigned , default = 1
3	R/W	1	reg_eo_enable : // unsigned , default = 1
2	R/W	1	reg_oe_enable : // unsigned , default = 1
1	R/W	1	reg_ogain_enable : // unsigned , default = 1
0	R/W	1	reg_cgain_enable : // unsigned , default = 1

DI_HDR2_CLK_GATE 0x3771

Bit(s)	R/W	Default	Description
31:30	R/W	0	clk_tm : gate clock ctrl (main clock) // unsigned , default = 0
29:28	R/W	0	output : matrix clock gate ctrl // unsigned , default = 0
25:24	R/W	0	input : matrix clock gate ctrl // unsigned , default = 0
23:22	R/W	0	hdr : top cbus clock gate ctrl // unsigned , default = 0
21:20	R/W	0	eotf : cbus clock gate ctrl // unsigned , default = 0
19:18	R/W	0	oetf : cbus clock gate ctrl // unsigned , default = 0
17:16	R/W	0	gamma : mult cbus clock gate ctrl // unsigned , default = 0
15:14	R/W	0	adaptive : cbus scaler clock gate ctrl // unsigned , default = 0

Bit(s)	R/W	Default	Description
13:12	R/W	0	cgain : cbus clock gate ctrl // unsigned , default = 0
11:10	R/W	0	eotf : clock gate ctrl // unsigned , default = 0
9:8	R/W	0	oetf : clock gate ctrl // unsigned , default = 0
7:6	R/W	0	gamma : mult clock gate ctrl // unsigned , default = 0
5:4	R/W	0	adaptive : scaler clock gate ctrl // unsigned , default = 0
3:2	R/W	0	uv : gain clock gate ctrl // unsigned , default = 0
1:0	R/W	0	cgain : clock gate ctrl // unsigned , default = 0

DI_HDR2_MATRIXI_COEF00_01 0x3772

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0
12:0	R/W	0	coef01 : // signed , default = 0

DI_HDR2_MATRIXI_COEF02_10 0x3773

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

DI_HDR2_MATRIXI_COEF11_12 0x3774

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

DI_HDR2_MATRIXI_COEF20_21 0x3775

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

DI_HDR2_MATRIXI_COEF22 0x3776

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

DI_HDR2_MATRIXI_COEF30_31 0x3777

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

DI_HDR2_MATRIXI_COEF32_40 0x3778

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

DI_HDR2_MATRIXI_COEF41_42 0x3779

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

DI_HDR2_MATRIXI_OFFSET0_1 0x377A

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

DI_HDR2_MATRIXI_OFFSET2 0x377B

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

DI_HDR2_MATRIXI_PRE_OFFSET0_1 0x377C

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

DI_HDR2_MATRIXI_PRE_OFFSET2 0x377D

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

DI_HDR2_MATRIXO_COEF00_01 0x377E

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0
12:0	R/W	0	coef01 : // signed , default = 0

DI_HDR2_MATRIXO_COEF02_10 0x377F

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

DI_HDR2_MATRIXO_COEF11_12 0x3780

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

DI_HDR2_MATRIXO_COEF20_21 0x3781

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

DI_HDR2_MATRIXO_COEF22 0x3782

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

DI_HDR2_MATRIXO_COEF30_31 0x3783

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

DI_HDR2_MATRIXO_COEF32_40 0x3784

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

DI_HDR2_MATRIXO_COEF41_42 0x3785

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

DI_HDR2_MATRIXO_OFFSET0_1 0x3786

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

DI_HDR2_MATRIXO_OFFSET2 0x3787

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

DI_HDR2_MATRIXO_PRE_OFFSET0_1 0x3788

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

DI_HDR2_MATRIXO_PRE_OFFSET2 0x3789

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

DI_HDR2_MATRIXI_CLIP 0x378A

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

DI_HDR2_MATRIXO_CLIP 0x378B

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

DI_HDR2_CGAIN_OFFT 0x378C

Bit(s)	R/W	Default	Description
26:16	R/W	0	reg_cgain_oft2 : // signed , default = 0
10:0	R/W	0	reg_cgain_oft1 : // signed , default = 0

DI_EOTF_LUT_ADDR_PORT 0x378E

Bit(s)	R/W	Default	Description
7:0	R/W	0	eotf_lut_addr : // unsigned , default = 0

DI_EOTF_LUT_DATA_PORT 0x378F

Bit(s)	R/W	Default	Description
19:0	R/W	0	eotf_lut_data : // unsigned , default = 0

DI_OETF_LUT_ADDR_PORT 0x3790

Bit(s)	R/W	Default	Description
7:0	R/W	0	oetf_lut_addr : // unsigned , default = 0

DI_OETF_LUT_DATA_PORT 0x3791

Bit(s)	R/W	Default	Description
11:0	R/W	0	oetf_lut_data : // unsigned , default = 0

DI_CGAIN_LUT_ADDR_PORT 0x3792

Bit(s)	R/W	Default	Description
7:0	R/W	0	cgain_lut_addr : // unsigned , default = 0

DI_CGAIN_LUT_DATA_PORT 0x3793

Bit(s)	R/W	Default	Description
11:0	R/W	0	cgain_lut_data : // unsigned , default = 0

DI_HDR2_CGAIN_COEF0 0x3794

Bit(s)	R/W	Default	Description
27:16	R/W	0	reg_cgain_coef1 : // unsigned , default = 0
11:0	R/W	0	reg_cgain_coef0 : //unsigned , default = 0

DI_HDR2_CGAIN_COEF1 0x3795

Bit(s)	R/W	Default	Description
11:0	R/W	0	reg_cgain_coef2 : // unsigned , default = 0

DI_OGAIN_LUT_ADDR_PORT 0x3796

Bit(s)	R/W	Default	Description
7:0	R/W	0	ogain_lut_addr : // unsigned , default = 0

10.2.3.11 DI_SCALE & HDR Registers

DI_SCO_FIFO_CTRL 0x374e

DI_OGAIN_LUT_DATA_PORT 0x3797

DI_HDR2_ADPS_CTRL 0x3798

DI_HDR2_ADPS_ALPHA0 0x3799

Bit(s)	R/W	Default	Description
29:16	R/W	0x1000	reg_adpscl_alpha1 : // unsigned , default = 0x1000
13:0	R/W	0x1000	reg_adpscl_alpha0 : // unsigned , default = 0x1000

DI_HDR2_ADPS_ALPHA1 0x379A

Bit(s)	R/W	Default	Description
27:24	R/W	0xc	reg_adpscl_shift0 : // unsigned , default = 0xc
23:20	R/W	0xc	reg_adpscl_shift1 : // unsigned , default = 0xc
19:16	R/W	0xc	reg_adpscl_shift2 : // unsigned , default = 0xc
13:0	R/W	0x1000	reg_adpscl_alpha2 : // unsigned , default = 0x1000

DI_HDR2_ADPS_BETA0 0x379B

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta0_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta0 : // unsigned , default = 0xfc000

DI_HDR2_ADPS_BETA1 0x379C

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta1_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta1 : // unsigned , default = 0xfc000

DI_HDR2_ADPS_BETA2 0x379D

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta2_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta2 : // unsigned , default = 0xfc000

DI_HDR2_ADPS_COEF0 0x379E

Bit(s)	R/W	Default	Description
27:16	R/W	460	reg_adpscl_ys_coef1 : // unsigned , default = 460
11:0	R/W	1188	reg_adpscl_ys_coef0 : // unsigned , default = 1188

DI_HDR2_ADPS_COEF1 0x379F

Bit(s)	R/W	Default	Description
11:0	R/W	104	reg_adpscl_ys_coef2 : // unsigned , default = 104

DI_HDR2_GMUT_CTRL 0x37A0

Bit(s)	R/W	Default	Description
3:0	R/W	14	reg_gmut_shift : // unsigned , default = 14

DI_HDR2_GMUT_COEF0 0x37A1

Bit(s)	R/W	Default	Description
31:16	R/W	674	reg_gmut_coef01 : // unsigned , default = 674
15:0	R/W	1285	reg_gmut_coef00 : // unsigned , default = 1285

DI_HDR2_GMUT_COEF1 0x37A2

Bit(s)	R/W	Default	Description
31:16	R/W	142	reg_gmut_coef10 : // unsigned , default = 142
15:0	R/W	89	reg_gmut_coef02 : // unsigned , default = 89

DI_HDR2_GMUT_COEF2 0x37A3

Bit(s)	R/W	Default	Description
31:16	R/W	23	reg_gmut_coef12 : // unsigned , default = 23
15:0	R/W	1883	reg_gmut_coef11 : // unsigned , default = 1883

DI_HDR2_GMUT_COEF3 0x37A4

Bit(s)	R/W	Default	Description
31:16	R/W	180	reg_gmut_coef21 : // unsigned , default = 180
15:0	R/W	34	reg_gmut_coef20 : // unsigned , default = 34

DI_HDR2_GMUT_COEF4 0x37A5

Bit(s)	R/W	Default	Description
15:0	R/W	1834	reg_gmut_coef22 : // unsigned , default = 1834

DI_HDR2_PIPE_CTRL1 0x37A6

Bit(s)	R/W	Default	Description
31:24	R/W	4	vblank_num_oetf : // unsigned , default = 4
23:16	R/W	4	hblank_num_oetf : // unsigned , default = 4
15:8	R/W	10	vblank_num_eotf : // unsigned , default = 10

7:0	R/W	10	hblank_num_eotf : // unsigned , default = 10

DI_HDR2_PIPE_CTRL2 0x37A7

Bit(s)	R/W	Default	Description
31:24	R/W	10	vblank_num_cgain : // unsigned , default = 10
23:16	R/W	10	hblank_num_cgain : // unsigned , default = 10
15:8	R/W	11	vblank_num_gmut : // unsigned , default = 11
7:0	R/W	11	hblank_num_gmut : // unsigned , default = 11

DI_HDR2_PIPE_CTRL3 0x37A8

Bit(s)	R/W	Default	Description
31:24	R/W	22	vblank_num_adps : // unsigned , default = 22
23:16	R/W	2	hblank_num_adps : // unsigned , default = 2
15:8	R/W	4	vblank_num_uv : // unsigned , default = 4
7:0	R/W	4	hblank_num_uv : // unsigned , default = 4

DI_HDR2_PROC_WIN1 0x37A9

Bit(s)	R/W	Default	Description
28:16	R/W	0	proc_win_h_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_h_st : // unsigned , default = 0

DI_HDR2_PROC_WIN2 0x37AA

Bit(s)	R/W	Default	Description
31	R/W	0	proc_win_gmut_en : // unsigned , default = 0
30	R/W	0	proc_win_adps_en : // unsigned , default = 0
29	R/W	0	proc_win_cgain_en : // unsigned , default = 0
28:16	R/W	0	proc_win_v_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_v_st : // unsigned , default = 0

DI_HDR2_MATRIXI_EN_CTRL 0x37AB

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

DI_HDR2_MATRIXO_EN_CTRL 0x37AC

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

10.2.3.12 NR_SCALE registers

NRDSWR_X 0x37f9

Bit(s)	R/W	Default	Description
31:30	R/W	2	burst_len : unsigned , default = 2
29	R/W	0	rev_x : unsigned , default = 0
28:16	R/W	0	start_x : unsigned , default = 0
12:0	R/W	2	end_x : unsigned , default = 2cf

NRDSWR_Y 0x37fa

Bit(s)	R/W	Default	Description
31:30	R/W	0	canvas_id : unsigned , default = 0
29	R/W	0	rev_y : unsigned , default = 0
28:16	R/W	0	start_y : unsigned , default = 0
12:0	R/W	0	end_y : unsigned , default = 0x1df

NRDSWR_CTRL 0x37fb

Bit(s)	R/W	Default	Description
31:16	R/W	0	urgent_ctrl : unsigned , default = 0
15	R/W	0	force_wvalid : unsigned , default = 0
14	R/W	0	canvas_syncen : unsigned , default = 0
13	R/W	1	canvas_wr : unsigned , default = 1
12	R/W	0	req_en : unsigned , default = 0
10	R/W	0	clr_wrrsp : unsigned , default = 0
8	R/W	0	urgent : unsigned , default = 0
7:0	R/W	0	canvas_index : unsigned , default = 0

NRDSWR_CAN_SIZE 0x37fc

Bit(s)	R/W	Default	Description
30:29	R/W	0	reg_rst : unsigned , default = 0
28:16	R/W	0	hsizem1 : unsigned , default = 0x2cf
14	R/W	0	reg_reset : unsigned , default = 0
13	R/W	0	little_endian : unsigned , default = 0
12:0	R/W	0	vsizem1 : unsigned , default = 0x1df

*NR*_DS_BUF_SIZE 0x3740

Bit(s)	R/W	Default	Description
31:24	R.O	96	dsbuf_rowmax : // unsigned , default = 96
23:16	R/W	128	dsbuf_colmax : // unsigned , default = 128
15: 8	R.O	128	dsbuf_orow : // unsigned , default = 128
7: 0	R/W	128	dsbuf_ocol : // unsigned , default = 128

NR_DS_CTRL 0x3741

Bit(s)	R/W	Default	Description
29:24	R/W	8	reg_h_step : // unsigned , default = 8 rand lut0
21:16	R/W	8	reg_v_step : // unsigned , default = 8 rand lut0
14:12	R/W	4	reg_haa_sel : // unsigned , default = 4
10: 8	R/W	4	reg_vaa_sel : // unsigned , default = 4
6: 4	R/W	1	reg_use_hphase : // unsigned , default = 1
0	R/W	0	reg_yuv_bldmode : // unsigned , default = 0

NR_DS_OFFSET 0x3742

Bit(s)	R/W	Default	Description
25:16	R/W	0	reg_h_ofst : // signed , default = 0
9: 0	R/W	0	reg_v_ofst : // signed , default = 0

NR_DS_BLD_COEF 0x3743

Bit(s)	R/W	Default	Description
23:16	R/W	128	reg_yuv_bldcoef2 : // unsigned , default = 128
15: 8	R/W	64	reg_yuv_bldcoef1 : // unsigned , default = 64
7: 0	R/W	64	reg_yuv_bldcoef0 : // unsigned , default = 64

10.2.3.13 MCDI registers

MCDI_HV_SIZEIN 0x2f00

Bit(s)	R/W	Default	Description
31-29	R/W		reserved
28-16	R/W	1024	reg_mcdi_hsize image horizontal size (number of cols) default=1024
15-13	R/W		reserved
12-0	R/W	1024	reg_mcdi_vsize image vertical size (number of rows) default=1024

MCDI_HV_BLKSIZEIN 0x2f01

Bit(s)	R/W	Default	Description
31	R/W	0	reg_mcdi_vrev default = 0
30	R/W	0	reg_mcdi_hrev default = 0
29-28	R/W	0	reserved
27-16	R/W	1024	reg_mcdi_blkhsize image horizontal blk size (number of cols) default=1024
15-13	R/W	0	reserved
11-0	R/W	1024	reg_mcdi_blkvsize image vertical blk size (number of rows) default=1024

MCDI_BLKTOTAL 0x2f02

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved
23-0	R/W	0	reg_mcdi_blktotal

MCDI_MOTINEN 0x2f03

Bit(s)	R/W	Default	Description
31-2	R/W	0	reserved
1	R/W	1	reg_mcdi_motionrefen. enable motion refinement of MA, default = 1
0	R/W	1	reg_mcdi_motionparadoxen. enable motion paradox detection, default = 1

MCDI_CTRL_MODE 0x2f04

Bit(s)	R/W	Default	Description
31-28	R/W		reserved
27-26	R/W	2	reg_mcdi_lmvlocken 0:disable, 1: use max Lmv, 2: use no-zero Lmv, lmv lock enable mode, default = 2

Bit(s)	R/W	Default	Description
25	R/W	1	reg_mcdi_reldetrptchken 0-unable; 1: enableenable repeat pattern check (not repeat mv detection) in rel det part, default = 1
24	R/W	1	reg_mcdi_reldetgmvpd22chken 0-unable; 1: enable enable pull-down 22 mode check in gmv lock mode for rel det, default = 1
23	R/W	1	reg_mcdi_pd22chken 0-unable; 1: enable enable pull-down 22 mode check (lock) function, default = 1
22	R/W	1	reg_mcdi_reldetlpfen 0-unable; 1: enable enable det value lpf, default = 1
21	R/W	1	reg_mcdi_reldetlmvpd22chken 0-unable; 1: enable enable pull-down 22 mode check in lmv lock mode for rel det, default = 1
20	R/W	1	reg_mcdi_reldetlmvdifchken 0-unable; 1: enable enable lmv dif check in lmv lock mode for rel det, default = 1
19	R/W	1	reg_mcdi_reldetgmvdifchken 0-unable; 1: enable enable lmv dif check in lmv lock mode for rel det, default = 1
18	R/W	1	reg_mcdi_reldetpd22chken 0-unable; 1: enable enable pull-down 22 mode check for rel det refinement, default = 1
17	R/W	1	reg_mcdi_reldetfrqchken 0-unable; 1: enable enable mv frequency check in rel det, default = 1
16	R/W		reg_mcdi_qmeen 0-unable; 1: enable enable quarter motion estimation, defautl = 1
15	R/W	1	reg_mcdi_refrptmven 0-unable; 1: enable use repeat mv in refinement, default = 1
14	R/W	1	reg_mcdi_refgmven 0-unable; 1: enable use gmv in refinement, default = 1
13	R/W	1	reg_mcdi_reflmven 0-unable; 1: enable use lmvs in refinement, default = 1
12	R/W	1	reg_mcdi_refnmven 0-unable; 1: enable use neighoring mvs in refinement, default = 1
11	R/W		reserved
10	R/W	1	reg_mcdi_referrfrqchken 0-unable; 1: enable enable mv frquency check while finding min err in ref, default = 1
9	R/W	1	reg_mcdi_refen 0-unable; 1: enable enable mv refinement, default = 1
8	R/W	1	reg_mcdi_horlineen 0-unable; 1: enable enable horizontal lines detection by sad map, default = 1
7	R/W	1	reg_mcdi_highvertfrqdeten 0-unable; 1: enable enable high vertical frequency pattern detection, default = 1
6	R/W	1	reg_mcdi_gmvlocken 0-unable; 1: enable enable gmv lock mode, default = 1
5	R/W	1	reg_mcdi_rptmven 0-unable; 1: enable enable repeat pattern detection, default = 1
4	R/W	1	reg_mcdi_gmven 0-unable; 1: enable enable global motion estimation, default = 1
3	R/W	1	reg_mcdi_lmven0-unable; 1: enable enable line mv estimation for hme, default = 1
2	R/W	1	reg_mcdi_chkedgeen0-unable; 1: enable enable check edge function, default = 1

Bit(s)	R/W	Default	Description
1	R/W	1	reg_mcdi_txtdeten 0-unable; 1: enable enable texture detection, default = 1

MCDI_UNI_MVDST 0x2f05

Bit(s)	R/W	Default	Description
31-20	R/W		reserved
19-17	R/W	1	reg_mcdi_unimvdstabsseg0 segment0 for uni-mv abs, default = 1
16-12	R/W	15	reg_mcdi_unimvdstabsseg1 segment1 for uni-mv abs, default = 15
11-8	R/W	2	reg_mcdi_unimvdstabsdifgain0 2/2, gain0 of uni-mv abs dif for segment0, normalized 2 to '1', default = 2
7-5	R/W	2	reg_mcdi_unimvdstabsdifgain1 2/2, gain1 of uni-mv abs dif for segment1, normalized 2 to '1', default = 2
4-2	R/W	2	reg_mcdi_unimvdstabsdifgain2 2/2, gain2 of uni-mv abs dif beyond segment1, normalized 2 to '1', default = 2
1-0	R/W	0	reg_mcdi_unimvdstsgnshft shift for neighboring distance of uni-mv, default = 0

MCDI_BI_MVDST 0x2f06

Bit(s)	R/W	Default	Description
31-20	R/W	0	reserved
19-17	R/W	1	reg_mcdi_bimvdstabsseg0 segment0 for bi-mv abs, default = 1
16-12	R/W	9	reg_mcdi_bimvdstabsseg1 segment1 for bi-mv abs, default = 9
11-8	R/W	6	reg_mcdi_bimvdstabsdifgain0 6/2, gain0 of bi-mv abs dif for segment0, normalized 2 to '1', default = 6
7-5	R/W	3	reg_mcdi_bimvdstabsdifgain1 3/2, gain1 of bi-mvabs dif for segment1, normalized 2 to '1', default = 3
4-2	R/W	2	reg_mcdi_bimvdstabsdifgain2 2/2, gain2 of bi-mvabs dif beyond segment1, normalized 2 to '1', default = 2
1-0	R/W	0	reg_mcdi_bimvdstsgnshft shift for neighboring distance of bi-mv, default = 0

MCDI_SAD_GAIN 0x2f07

Bit(s)	R/W	Default	Description
31-19	R/W	0	reserved
18-17	R/W	3	reg_mcdi_unisadcorepxlgain uni-sad core pixels gain, default = 3
16	R/W	0	reg_mcdi_unisadcorepxlnormen enable uni-sad core pixels normalization, default = 0
15-11	R/W		reserved

10-9	R/W	3	reg_mcdi_bisadcorepxlgain bi-sad core pixels gain, default = 3
8	R/W	1	reg_mcdi_bisadcorepxlnormen enable bi-sad core pixels normalization, default = 1
7-3	R/W		reserved
2-1	R/W	3	reg_mcdi_biqsadcorepxlgain bi-qsad core pixels gain, default = 3
0	R/W	1	reg_mcdi_biqsadcorepxlnormen enable bi-qsad core pixels normalization, default = 1

MCDI_TXT_THD 0x2f08

Bit(s)	R/W	Default	Description
31-24	R/W		reserved
23-16	R/W	24	reg_mcdi_txtminmaxdifthd, min max dif threshold (>=) for texture detection, default = 24
15-8	R/W	9	reg_mcdi_txtmeandifthd, mean dif threshold (<) for texture detection, default = 9
7-3	R/W	0	reserved
2-0	R/W	2	reg_mcdi_txtdetthd, texture detecting threshold, 0~4, default = 2

MCDI_FLT_MODESEL 0x2f09

Bit(s)	R/W	Default	Description
31	R/W	0	reserved
30-28	R/W	1	reg_mcdi_flthorlineselmode mode for horizontal line detecting flat calculation, default = 1, same as below
27	R/W	0	reserved
26-24	R/W	4	reg_mcdi_fltgmvselmode mode for gmv flat calculation, default = 4, same as below
23	R/W	0	reserved
22-20	R/W	2	reg_mcdi_fltsadselmode mode for sad flat calculation, default = 2, same as below
19	R/W	0	reserved
18-16	R/W	3	reg_mcdi_fltbadwselmode mode for badw flat calculation, default = 3, same as below
15	R/W	0	reserved
14-12	R/W	4	reg_mcdi_fltrptmvselmode mode for repeat mv flat calculation, default = 4, same as below
11	R/W	0	reserved
10-8	R/W	4	reg_mcdi_fltbadrelselmode mode for bad rel flat calculation, default = 4, same as below
7	R/W	0	reserved
6-4	R/W	2	reg_mcdi_fltcolcfdselmode mode for col cfd flat calculation, default = 2, same as below
3	R/W	0	reserved

Bit(s)	R/W	Default	Description
2-0	R/W	2	reg_mcdi_fltpd22chkselmode mode for pd22 check flat calculation, default = 2, 0:cur dif h, 1: cur dif v, 2: pre dif h, 3: pre dif v, 4: cur flt, 5: pre flt, 6: cur+pre, 7: max all(cur,pre)

MCDI_CHK_EDGE_THD 0x2f0a

Bit(s)	R/W	Default	Description
23-28	R/W	0	reserved.
27-24	R/W	1	reg_mcdi_chkedgedifsadthd. thd (<=) for sad dif check, 0~8, default = 1
23-16	R/W	0	reserved.
15-12	R/W	15	reg_mcdi_chkedgemaxedgethd. max drt of edge, default = 15
11-8	R/W	2	reg_mcdi_chkedgeminedgethd. min drt of edge, default = 2
7	R/W	0	reserved.
6-0	R/W	14	reg_mcdi_chkedgevdifthd. thd for vertical dif in check edge, default = 14

MCDI_CHK_EDGE_GAIN_OFFST 0x2f0b

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved.
23-20	R/W	4	reg_mcdi_chkedgedifthd1. thd1 for edge dif check (<=), default = 4
19-16	R/W	15	reg_mcdi_chkedgedifthd0. thd0 for edge dif check (>=), default = 15
-15	R/W	0	reserved.
14-10	R/W	24	reg_mcdi_chkedgechklen. total check length for edge check, 1~24 (>0), default = 24
9-8	R/W	1	reg_mcdi_chkedgeedgesel. final edge select mode, 0: original start edge, 1: lpf start edge, 2: orignal start+end edge, 3: lpf start+end edge, default = 1
7-3	R/W	4	reg_mcdi_chkedgesaddstgain. distance gain for sad calc while getting edges, default = 4
2	R/W	0	reg_mcdi_chkedgechkmode. edge used in check mode, 0- original edge, 1: lpf edge, defautl = 1
1	R/W	0	reg_mcdi_chkedgestartedge. edge mode for start edge, 0- original edge, 1: lpf edge, defautl = 0
0	R/W	0	reg_mcdi_chkedgeedgelpf. edge lpf mode, 0-[0,2,4,2,0], 1:[1,2,2,2,1], default = 0

MCDI_LMV_RT 0x2f0c

Bit(s)	R/W	Default	Description
31-15	R/W	0	reserved
14-12	R/W	0	reg_mcdi_lmvvalidmode valid mode for lmv calc., 100b: use char det, 010b: use flt, 001b: use hori flg

11-10	R/W	1	reg_mcdi_lmvgainmvmode four modes of mv selection for lmv weight calucluation, default = 1
// // lst(x-1	R/W	0	x,x+1); 1- cur(x-4,x-3), lst(x,x+1); 2: cur(x-5,x-4,x-3), lst(x-1,x,x+1,x+2,x+3); 3: cur(x-6,x-5,x-4,x-3), lst(x-1,x,x+1,x+2);
9	R/W	0	reg_mcdi_lmvinitmode initial lmvs at first row of input field, 0- intial value = 0; 1: inital = 32 (invalid), default = 0
8	R/W	0	reserved
7-4	R/W	5	reg_mcdi_lmvrt0 ratio of max mv, default = 5
3-0	R/W	5	reg_mcdi_lmvrt1 ratio of second max mv, default = 5

MCDI_LMV_GAINTHD 0x2f0d

Bit(s)	R/W	Default	Description
31-24	R/W	96	reg_mcdi_lmvvxmaxgain max gain of lmv weight, default = 96
23	R/W	0	reserved
22-20	R/W	1	reg_mcdi_lmvdifthd0 dif threshold 0 (<) for small lmv, default = 1
19-17	R/W	2	reg_mcdi_lmvdifthd1 dif threshold 1 (<) for median lmv, default = 2
16-14	R/W	3	reg_mcdi_lmvdifthd2 dif threshold 2 (<) for large lmv, default = 3
13-8	R/W	20	reg_mcdi_lmvnumlmt least/limit number of (total number - max0), default = 20
7-0	R/W	9	reg_mcdi_lmvfltthd flt cnt thd (<) for lmv, default = 9

MCDI_RPTMV_THD0 0x2f0e

Bit(s)	R/W	Default	Description
31-25	R/W	64	reg_mcdi_rptmvslpthd2 slope thd (>=) between i and i+3/i-3 (i+4/i-4), default = 64
24-20	R/W	4	reg_mcdi_rptmvslpthd1 slope thd (>=) between i and i+2/i-2, default = 4
19-10	R/W	300	reg_mcdi_rptmvampthd2 amplitude thd (>=) between max and min, when count cycles, default = 300
9-0	R/W	400	reg_mcdi_rptmvampthd1 amplitude thd (>=) between average of max and min, default = 400

MCDI_RPTMV_THD1 0x2f0f

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-25	R/W	2	reg_mcdi_rptmvcyccntthd thd (>=) of total cycles count, default = 2
24-21	R/W	3	reg_mcdi_rptmvcycdifthd dif thd (<) of cycles length, default = 3
20-18	R/W	1	reg_mcdi_rptmvcycvldthd thd (>) of valid cycles number, default = 1

17-15	R/W	2	reg_mcdi_rptmvhalfcycminthd min length thd (>=) of half cycle, default = 2
14-11	R/W	5	reg_mcdi_rptmvhalfcycdifthd neighboring half cycle length dif thd (<), default = 5
10-8	R/W	2	reg_mcdi_rptmvminmaxcntthd least number of valid max and min, default = 2
7-5	R/W	2	reg_mcdi_rptmvcycminthd min length thd (>=) of cycles, default = 2
4-0	R/W	17	reg_mcdi_rptmvcycmaxthd max length thd (<) of cycles, default = 17

MCDI_RPTMV_THD2 0x2f10

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved
23-16	R/W	8	reg_mcdi_rptmvhdifthd0 higher hdif thd (>=) (vertical edge) for rpt detection, default = 8
15-8	R/W	4	reg_mcdi_rptmvhdifthd1 hdif thd (>=) (slope edge) for rpt detection, default = 4
7-0	R/W	1	reg_mcdi_rptmvvdifthd vdif thd (>=) (slope edge) for rpt detection, default = 1

MCDI_RPTMV_SAD 0x2f11

Bit(s)	R/W	Default	Description
31-26	R/W	0	reserved
25-16	R/W	336	reg_mcdi_rptmvsaddifthdgain 7x3x(16/16), gain for sad dif thd in rpt mv detection, 0~672, normalized 16 as '1', default = 336
15-10	R/W	0	reserved
9-0	R/W	16	reg_mcdi_rptmvsaddifthdoffst offset for sad dif thd in rpt mv detection, -512~511, default = 16

MCDI_RPTMV_FLG 0x2f12

Bit(s)	R/W	Default	Description
31-18	R/W	0	reserved
17-16	R/W	2	reg_mcdi_rptmvmode select mode of mvs for repeat motion estimation, 0: hmv, 1: qmv/2, 2 or 3: qmv/4, default = 2
15-8	R/W	64	reg_mcdi_rptmvflgcntthd thd (>=) of min count number for rptmv of whole field, for rptmv estimation, default = 64
7-5	R/W	0	reserved
4-0	R/W	0	reg_mcdi_rptmvflgcntrt 4/32, ratio for repeat mv flag count, normalized 32 as '1', set 31 to 32,

MCDI_RPTMV_GAIN 0x2f13

Bit(s)	R/W	Default	Description
31-24	R/W	96	reg_mcdi_rptmvlftgain up repeat mv gain for hme, default = 96

23-16	R/W	32	reg_mcdi_rptmvuplftgain up left repeat mv gain for hme, default = 32
15-8	R/W	64	reg_mcdi_rptmvupgain up repeat mv gain for hme, default = 64
7-0	R/W	32	reg_mcdi_rptmvuprightgain up right repeat mv gain for hme, default = 32

MCDI_GMV_RT 0x2f14

Bit(s)	R/W	Default	Description
31	R/W	0	reserved
30-24	R/W	32	reg_mcdi_gmvmtnrt0 ratio 0 for motion senario, set 127 to 128, normalized 128 as '1', default =32
23	R/W	0	reserved
22-16	R/W	56	reg_mcdi_gmvmtnrt1 ratio 1 for motion senario, set 127 to 128, normalized 128 as '1', default = 56
15	R/W	0	reserved
14-8	R/W	56	reg_mcdi_gmvstlrt0 ratio 0 for still senario, set 127 to 128, normalized 128 as '1', default = 56
7	R/W	0	reserved
6-0	R/W	80	reg_mcdi_gmvstlrt1 ratio 1 for still senario, set 127 to 128, normalized 128 as '1', default = 80

MCDI_GMV_GAIN 0x2f15

Bit(s)	R/W	Default	Description
31-25	R/W	100	reg_mcdi_gmvzeromvlockrt0 ratio 0 for locking zero mv, set 127 to 128, normalized 128 as '1', default = 100
24-18	R/W	112	reg_mcdi_gmvzeromvlockrt1 ratio 1 for locking zero mv, set 127 to 128, normalized 128 as '1', default = 112
17-16	R/W	3	reg_mcdi_gmvvalidmode valid mode for gmv calc., 10b: use flt, 01b: use hori flg, default = 3
15-8	R/W	0	reg_mcdi_gmvvxgain gmv's vx gain when gmv locked for hme, default = 0
7-0	R/W	3	reg_mcdi_gmvfltthd flat thd (<) for gmv calc. default = 3

MCDI_HOR_SADOFST 0x2f16

Bit(s)	R/W	Default	Description
31-25	R/W	0	reserved
24-16	R/W	21	reg_mcdi_horsaddifthdgain 21*1/8, gain/divisor for sad dif threshold in hor line detection, normalized 8 as '1', default = 21
15-8	R/W	0	reg_mcdi_horsaddifthdoffst offset for sad dif threshold in hor line detection, -128~127, default = 0
7-0	R/W	24	reg_mcdi_horvdifthd threshold (>=) of vertical dif of next block for horizontal line detection, default = 24

MCDI_REF_MV_NUM 0x2f17

Bit(s)	R/W	Default	Description
31-2	R/W	0	reserved
1-0	R/W	0	reg_mcdi_refmcmode. motion compensated mode used in refinement, 0: pre, 1: next, 2: (pre+next)/2, default = 0

MCDI_REF_BADW_THD_GAIN 0x2f18

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-24	R/W	6	reg_mcdi_refbadwcnt2gain. gain for badwv count num==3, default = 6
23-20	R/W	3	reg_mcdi_refbadwcnt1gain. gain for badwv count num==2, default = 3
19-16	R/W	1	reg_mcdi_refbadwcnt0gain. gain for badwv count num==1, default = 1
15-12	R/W	4	reg_mcdi_refbadwthd3. threshold 3 for detect badweave with largest average luma, default = 4
11-8	R/W	3	reg_mcdi_refbadwthd2. threshold 2 for detect badweave with third smallest average luma, default = 3
7-4	R/W	2	reg_mcdi_refbadwthd1. threshold 1 for detect badweave with second smallest average luma, default = 2
3-0	R/W	1	reg_mcdi_refbadwthd0. threshold 0 for detect badweave with smallest average luma, default = 1

MCDI_REF_BADW_SUM_GAIN 0x2f19

Bit(s)	R/W	Default	Description
31-13	R/W	0	reserved
12-8	R/W	8	reg_mcdi_refbadwsumgain0. sum gain for r channel, 0~16, default = 8
7-5	R/W	0	reserved
4	R/W	0	reg_mcdi_refbadwcalcmode. mode for badw calculation, 0-sum, 1:max, default = 0
3-0	R/W	0	reserved

MCDI_REF_BS_THD_GAIN 0x2f1a

Bit(s)	R/W	Default	Description
31-28	R/W	2	reg_mcdi_refbsudgain1. up & down block stregth gain1, normalized to 8 as '1', default = 2
27-24	R/W	4	reg_mcdi_refbsudgain0. up & down block stregth gain0, normalized to 8 as '1', default = 4
23-19	R/W	0	reserved
18-16	R/W	0	reg_mcdi_refbslftgain. left block strength gain, default = 0
15-13	R/W	0	reserved

12-8	R/W	16	reg_mcdi_refbsthd1. threshold 1 for detect block stregth in refinment, default = 16
7-5	R/W	0	reserved
4-0	R/W	8	reg_mcdi_refbsthd0. threshold 0 for detect block stregth in refinment, default = 8

MCDI_REF_ERR_GAIN0 0x2f1b

Bit(s)	R/W	Default	Description
31	R/W	0	reserved
30-24	R/W	48	reg_mcdi_referrnbrdstgain. neighoring mv distances gain for err calc. in ref, normalized to 8 as '1', default = 48
23-20	R/W	0	reserved
19-16	R/W	4	reg_mcdi_referrbsgain. bs gain for err calc. in ref, normalized to 8 as '1', default = 4
15	R/W	0	reserved
14-8	R/W	64	reg_mcdi_referrbadwgain. badw gain for err calc. in ref, normalized to 8 as '1', default = 64
7-4	R/W	0	reserved
3-0	R/W	4	reg_mcdi_referrsadgain. sad gain for err calc. in ref, normalized to 8 as '1', default = 4

MCDI_REF_ERR_GAIN1 0x2f1c

Bit(s)	R/W	Default	Description
31-20	R/W	0	reserved
19-16	R/W	4	reg_mcdi_referrchkedgegain. check edge gain for err calc. in ref, normalized to 8 as '1', default = 4
15-12	R/W	0	reserved
11-8	R/W	0	reg_mcdi_referrlmvgain. (locked) lmv gain for err calc. in ref, normalized to 8 as '1', default = 0
7-4	R/W	0	reserved
3-0	R/W	0	reg_mcdi_referrgmvgain. (locked) gmv gain for err calc. in ref, normalized to 8 as '1', default = 0

MCDI_REF_ERR_FRQ_CHK 0x2f1d

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-24	R/W	10	reg_mcdi_referrfrqgain. gain for mv frquency, normalized to 4 as '1', default = 10
23-21	R/W	0	reserved
20-16	R/W	31	reg_mcdi_referrfrqmax. max gain for mv frquency check, default = 31
15	R/W	0	reserved

14-12	R/W	3	reg_mcdi_ref_errfrqmvdifthd2. mv dif threshold 2 (<) for mv frquency check, default = 3
11	R/W	0	reserved
10-8	R/W	2	reg_mcdi_ref_errfrqmvdifthd1. mv dif threshold 1 (<) for mv frquency check, default = 2
7	R/W	0	reserved
6-4	R/W	1	reg_mcdi_ref_errfrqmvdifthd0. mv dif threshold 0 (<) for mv frquency check, default = 1
3-0	R/W		reserved

MCDI_QME_LPF_MSK 0x2f1e

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-24	R/W	7	reg_mcdi_qmechkedgelpfmsk0. lpf mask0 for chk edge in qme, 0~8, msk1 = (8-msk0), normalized to 8 as '1', default = 7
23-20	R/W	0	reserved
19-16	R/W	7	reg_mcdi_qmebslpfmsk0. lpf mask0 for bs in qme, 0~8, msk1 = (8-msk0), normalized to 8 as '1', default = 7
15-12	R/W	0	reserved
11-8	R/W	7	reg_mcdi_qmebadwlpfmsk0. lpf mask0 for badw in qme, 0~8, msk1 = (8-msk0), normalized to 8 as '1', default = 7
7-4	R/W	0	reserved
3-0	R/W	7	reg_mcdi_qmesadlpfmsk0. lpf mask0 for sad in qme, 0~8, msk1 = (8-msk0), normalized to 8 as '1', default = 7

MCDI_REL_DIF_THD_02 0x2f1f

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved.
23-16	R/W	9	reg_mcdi_reldifthd2. thd (<) for (hdif+vdif), default = 9
15-8	R/W	5	reg_mcdi_reldifthd1. thd (<) for (vdif), default = 5
7-0	R/W	48	reg_mcdi_reldifthd0. thd (>=) for (hdif-vdif), default = 48

MCDI_REL_DIF_THD_34 0x2f20

Bit(s)	R/W	Default	Description
31-16	R/W	0	reserved.
15-8	R/W	255	reg_mcdi_reldifthd4. thd (<) for (hdif), default = 255
7-0	R/W	48	reg_mcdi_reldifthd3. thd (>=) for (vdif-hdif), default = 48

MCDI_REL_BADW_GAIN_OFFST_01 0x2f21

Bit(s)	R/W	Default	Description
31-24	R/W	0	reg_mcdi_relbadwoffst1. offset for badw adj, for flat block, -128~127, default = 0
23-16	R/W	128	reg_mcdi_relbadwgain1. gain for badw adj, for flat block, default = 128
15-8	R/W	0	reg_mcdi_relbadwoffst0. offset for badw adj, for vertical block, -128~127, default = 0
7-0	R/W	160	reg_mcdi_relbadwgain0. gain for badw adj, for vertical block, default = 160

MCDI_REL_BADW_GAIN_OFFST_23 0x2f22

Bit(s)	R/W	Default	Description
31-24	R/W	0	reg_mcdi_relbadwoffst3. offset for badw adj, for other block, -128~127, default = 0
23-16	R/W	48	reg_mcdi_relbadwgain3. gain for badw adj, for other block, default = 48
15-8	R/W	0	reg_mcdi_relbadwoffst2. offset for badw adj, for horizontal block, -128~127, default = 0
7-0	R/W	48	reg_mcdi_relbadwgain2. gain for badw adj, for horizontal block, default = 48

MCDI_REL_BADW_THD_GAIN_OFFST 0x2f23

Bit(s)	R/W	Default	Description
31-23	R/W	0	reserved.
22-16	R/W	0	reg_mcdi_relbadwoffst. offset for badw thd adj, -64~63, default = 0
15-8	R/W	0	reserved.
7-0	R/W	16	reg_mcdi_relbadwthdgain. gain0 for badw thd adj, normalized to 16 as '1', default = 16

MCDI_REL_BADW_THD_MIN_MAX 0x2f24

Bit(s)	R/W	Default	Description
31-18	R/W	0	reserved.
17-8	R/W	256	reg_mcdi_relbadwthdmax. max for badw thd adj, default = 256
7-0	R/W	16	reg_mcdi_relbadwthdmin. min for badw thd adj, default = 16

MCDI_REL_SAD_GAIN_OFFST_01 0x2f25

Bit(s)	R/W	Default	Description
31-24	R/W	0	reg_mcdi_relsadoffst1. offset for sad adj, for flat block, -128~127, default = 0
23-20	R/W	0	reserved.
19-16	R/W	8	reg_mcdi_relsadgain1. gain for sad adj, for flat block, normalized to 8 as '1', default = 8
15-8	R/W	0	reg_mcdi_relsadoffst0. offset for sad adj, for vertical block, -128~127, default = 0
7-4	R/W	0	reserved.

3-0	R/W	6	reg_mcdi_relsadgain0. gain for sad adj, for vertical block, normalized to 8 as '1', default = 6

MCDI_REL_SAD_GAIN_OFFST_23 0x2f26

Bit(s)	R/W	Default	Description
31-24	R/W	0	reg_mcdi_relsadoffst3. offset for sad adj, for other block, -128~127, default = 0
23-20	R/W	0	reserved.
19-16	R/W	8	reg_mcdi_relsadgain3. gain for sad adj, for other block, normalized to 8 as '1', default = 8
15-8	R/W	0	reg_mcdi_relsadoffst2. offset for sad adj, for horizontal block, -128~127, default = 0
7-4	R/W	0	reserved.
3-0	R/W	12	reg_mcdi_relsadgain2. gain for sad adj, for horizontal block, normalized to 8 as '1', default = 12

MCDI_REL_SAD_THD_GAIN_OFFST 0x2f27

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved.
23-16	R/W	0	reg_mcdi_relsadoffst. offset for sad thd adj, -128~127, default = 0
15-10	R/W	0	reserved.
9-0	R/W	42	reg_mcdi_relsadthdgain. gain for sad thd adj, 21*2/16, normalized to 16 as '1', default = 42

MCDI_REL_SAD_THD_MIN_MAX 0x2f28

Bit(s)	R/W	Default	Description
31-27	R/W	0	reserved.
26-16	R/W	672	reg_mcdi_relsadthdmax. max for sad thd adj, 21*32, default = 672
15-9	R/W	0	reserved.
8-0	R/W	42	reg_mcdi_relsadthdmin. min for sad thd adj, 21*2, default = 42

MCDI_REL_DET_GAIN_00 0x2f29

Bit(s)	R/W	Default	Description
31-21	R/W	0	reserved.
20-16	R/W	8	reg_mcdi_reldetbsgain0. gain0 (gmv locked) for bs, for det. calc. normalized to 16 as '1', default = 8
15-14	R/W	0	reserved.
13-8	R/W	12	reg_mcdi_reldetbadwgain0. gain0 (gmv locked) for badw, for det. calc. normalized to 16 as '1', default = 12
7-5	R/W	0	reserved.

Bit(s)	R/W	Default	Description
4-0	R/W	8	reg_mcdi_reldetsadgain0. gain0 (gmv locked) for qsad, for det. calc. normalized to 16 as '1', default = 8

MCDI_REL_DET_GAIN_01 0x2f2a

Bit(s)	R/W	Default	Description
31-14	R/W	0	reserved.
12-8	R/W	2	reg_mcdi_reldetchkedgegain0. gain0 (gmv locked) for chk_edge, for det. calc. normalized to 16 as '1', default = 2
7	R/W	0	reserved.
6-0	R/W	24	reg_mcdi_reldetnbrdstgain0. gain0 (gmv locked) for neighoring dist, for det. calc. normalized to 16 as '1', default = 24

MCDI_REL_DET_GAIN_10 0x2f2b

Bit(s)	R/W	Default	Description
31-21	R/W	0	reserved.
20-16	R/W	0	reg_mcdi_reldetbsgain1. gain1 (lmv locked) for bs, for det. calc. normalized to 16 as '1', default = 0
15-14	R/W	0	reserved.
13-8	R/W	8	reg_mcdi_reldetbadwgain1. gain1 (lmv locked) for badw, for det. calc. normalized to 16 as '1', default = 8
7-5	R/W	0	reserved.
4-0	R/W	8	reg_mcdi_reldetsadgain1. gain1 (lmv locked) for qsad, for det. calc. normalized to 16 as '1', default = 8

MCDI_REL_DET_GAIN_11 0x2f2c

Bit(s)	R/W	Default	Description
31-14	R/W	0	reserved.
12-8	R/W	0	reg_mcdi_reldetchkedgegain1. gain1 (lmv locked) for chk_edge, for det. calc. normalized to 16 as '1', default = 0
7	R/W	0	reserved.
6-0	R/W	24	reg_mcdi_reldetnbrdstgain1. gain1 (lmv locked) for neighoring dist, for det. calc. normalized to 16 as '1', default = 24

MCDI_REL_DET_GAIN_20 0x2f2d

Bit(s)	R/W	Default	Description
31-21	R/W	0	reserved.

20-16	R/W	12	reg_mcdi_reldetbsgain2. gain2 (no locked) for bs, for det. calc. normalized to 16 as '1', default = 12
15-14	R/W	0	reserved.
13-8	R/W	32	reg_mcdi_reldetbadwgain2. gain2 (no locked) for badw, for det. calc. normalized to 16 as '1', default = 32
7-5	R/W	0	reserved.
4-0	R/W	16	reg_mcdi_reldetsadgain2. gain2 (no locked) for qsad, for det. calc. normalized to 16 as '1', default = 16

M CDI_REL_DET_GAIN_21 0x2f2e

Bit(s)	R/W	Default	Description
31-26	R/W	0	reserved
25-16	R/W	0	reg_mcdi_reldetoffst. offset for rel calculation, for det. calc. -512~511, default = 0
15-14	R/W	0	reserved.
12-8	R/W	10	reg_mcdi_reldetchkedgegain2. gain2 (no locked) for chk_edge, for det. calc. normalized to 16 as '1', default = 10
7	R/W	0	reserved.
6-0	R/W	32	reg_mcdi_reldetnbrdstgain2. gain2 (no locked) for neighoring dist, for det. calc. normalized to 16 as '1', default = 32

MCDI_REL_DET_GMV_DIF_CHK 0x2f2f

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved.
23-16	R/W	0	reg_mcdi_reldetgmvfltthd. flat thd (>=) for gmv lock decision, default = 0
15	R/W	0	reserved.
14-12	R/W	3	reg_mcdi_reldetgmvdifthd. dif thd (>=) for current mv different from gmv for gmv dif check, actually used in Lmv lock check, default = 3
11	R/W	0	reserved.
10-8	R/W	1	reg_mcdi_reldetgmvdifmin. min mv dif for gmv dif check, default = 1, note: dif between reg_mcdi_rel_det_gmv_dif_max and reg_mcdi_rel_det_gmv_dif_min should be; 0,1,3,7, not work for others
7-4	R/W	4	reg_mcdi_reldetgmvdifmax. max mv dif for gmv dif check, default = 4
3-1	R/W	0	reserved
0	R/W	0	reg_mcdi_reldetgmvdifmvmode. mv mode used for gmv dif check, 0- use refmv, 1: use qmv, default = 0

MCDI_REL_DET_LMV_DIF_CHK 0x2f30

Bit(s)	R/W	Default	Description

31-24	R/W	0	reserved.
23-16	R/W	12	reg_mcdi_reldetlmvfltthd. flat thd (>=) for lmv lock decision, default = 12
15-14	R/W	0	reserved.
13-12	R/W	1	reg_mcdi_reldetlmvlockchkmode. lmv lock check mode, 0:cur Lmv, 1: cur & (last | next), 2: last & cur & next Lmv, default = 1
11	R/W	0	reserved.
10-8	R/W	1	reg_mcdi_reldetlmvdifmin. min mv dif for lmv dif check, default = 1, note: dif between reg_mcdi_rel_det_lmv_dif_max and reg_mcdi_rel_det_lmv_dif_min should be; 0,1,3,7, not work for others
7-4	R/W	4	reg_mcdi_reldetlmvdifmax. max mv dif for lmv dif check, default = 4
3-1	R/W	0	reserved
0	R/W	0	reg_mcdi_reldetlmvdifmvmode. mv mode used for lmv dif check, 0- use refmv, 1: use qmv, default = 0

MCDI_REL_DET_FRQ_CHK 0x2f31

Bit(s)	R/W	Default	Description
31-12	R/W	0	reserved.
11-8	R/W	10	reg_mcdi_reldetfrqgain. gain for frequency check, normalized to 4 as '1', default = 10
7-5	R/W	0	reserved
4-0	R/W	31	reg_mcdi_reldetfrqmax. max value for frequency check, default = 31

MCDI_REL_DET_PD22_CHK 0x2f32

Bit(s)	R/W	Default	Description
31-18	R/W	0	reserved.
30-21	R/W	512	reg_mcdi_reldetpd22chkoffst1. offset for pd22 check happened, default = 512
20-16	R/W	12	reg_mcdi_reldetpd22chkgain1. gain for pd22 check happened, normalized to 8 as '1', default = 12
14-5	R/W	512	reg_mcdi_reldetpd22chkoffst0. offset for pd22 check happened, default = 512
4-0	R/W	12	reg_mcdi_reldetpd22chkgain0. gain for pd22 check happened, normalized to 8 as '1', default = 12

MCDI_REL_DET_RPT_CHK_ROW 0x2f33

Bit(s)	R/W	Default	Description
31-27	R/W	0	reserved
26-16	R/W	2047	reg_mcdi_reldetrptchkendrow. end row (<) number for repeat check, default = 2047
15-11	R/W	0	reserved

10-0	R/W	0	reg_mcdi_reldetrptchkstartrow. start row (>=) number for repeat check, default = 0

MCDI_REL_DET_RPT_CHK_GAIN_QMV 0x2f34

Bit(s)	R/W	Default	Description
31-30	R/W	0	reserved
29-24	R/W	15	reg_mcdi_reldetrptchkqmvmax. max thd (<) of abs qmv for repeat check, default = 15, note that quarter mv's range is -63~63
23-22	R/W	0	reserved
21-16	R/W	10	reg_mcdi_reldetrptchkqmvmin. min thd (>=) of abs qmv for repeat check, default = 10, note that quarter mv's range is -63~63
15	R/W	0	reserved/
14-4	R/W	512	reg_mcdi_reldetrptchkoffst. offset for repeat check, default = 512
3-0	R/W	4	reg_mcdi_reldetrptchkgain. gain for repeat check, normalized to 8 as '1', default = 4

MCDI_REL_DET_RPT_CHK_THD_0 0x2f35

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved
23-16	R/W	255	reg_mcdi_reldetrptchkzerosadthd. zero sad thd (<) for repeat check, default = 255
15-14	R/W	0	reserved.
13-8	R/W	16	reg_mcdi_reldetrptchkzerobadwthd. zero badw thd (>=) for repeat check, default = 16
7-4	R/W	0	reserved
3-0	R/W	5	reg_mcdi_reldetrptchkfrqdifthd. frequency dif thd (<) for repeat check, 0~10, default = 5

MCDI_REL_DET_RPT_CHK_THD_1 0x2f36

Bit(s)	R/W	Default	Description
31-16	R/W	0	reserved
15-8	R/W	16	reg_mcdi_reldetrptchkvdifthd. vertical dif thd (<) for repeat check, default = 16
7-0	R/W	16	reg_mcdi_reldetrptchkhdifthd. horizontal dif thd (>=) for repeat check, default = 16

MCDI_REL_DET_LPF_DIF_THD 0x2f37

Bit(s)	R/W	Default	Description
31-24	R/W	9	reg_mcdi_reldetlpfdifthd3. hdif thd (<) for lpf selection of horizontal block, default = 9
23-16	R/W	48	reg_mcdi_reldetlpfdifthd2. vdif-hdif thd (>=) for lpf selection of horizontal block, default = 48
15-8	R/W	9	reg_mcdi_reldetlpfdifthd1. vdif thd (<) for lpf selection of vertical block, default = 9

7-0	R/W	48	reg_mcdi_reldetlpfdifthd0. hdif-vdif thd (>=) for lpf selection of vertical block, default = 48

MCDI_REL_DET_LPF_MSK_00_03 0x2f38

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-24	R/W	1	reg_mcdi_reldetlpfmsk03. det lpf mask03 for gmv/lmv locked mode, 0~16, default = 1
23-21	R/W	0	reserved
20-16	R/W	1	reg_mcdi_reldetlpfmsk02. det lpf mask02 for gmv/lmv locked mode, 0~16, default = 1
15-13	R/W	0	reserved
12-8	R/W	5	reg_mcdi_reldetlpfmsk01. det lpf mask01 for gmv/lmv locked mode, 0~16, default = 5
7-5	R/W	0	reserved
4-0	R/W	8	reg_mcdi_reldetlpfmsk00. det lpf mask00 for gmv/lmv locked mode, 0~16, default = 8

MCDI_REL_DET_LPF_MSK_04_12 0x2f39

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-24	R/W	0	reg_mcdi_reldetlpfmsk12. det lpf mask12 for vertical blocks, 0~16, default = 0
23-21	R/W	0	reserved
20-16	R/W	0	reg_mcdi_reldetlpfmsk11. det lpf mask11 for vertical blocks, 0~16, default = 0
15-13	R/W	0	reserved
12-8	R/W	16	reg_mcdi_reldetlpfmsk10. det lpf mask10 for vertical blocks, 0~16, default = 16
7-5	R/W	0	reserved
4-0	R/W	1	reg_mcdi_reldetlpfmsk04. det lpf mask04 for gmv/lmv locked mode, 0~16, default = 1

MCDI_REL_DET_LPF_MSK_13_21 0x2f3a

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-24	R/W	6	reg_mcdi_reldetlpfmsk21. det lpf mask21 for horizontal blocks, 0~16, default = 6
23-21	R/W	0	reserved
20-16	R/W	8	reg_mcdi_reldetlpfmsk20. det lpf mask20 for horizontal blocks, 0~16, default = 8

15-13	R/W	0	reserved
12-8	R/W	0	reg_mcdi_reldetlpfmsk14. det lpf mask14 for vertical blocks, 0~16, default = 0
7-5	R/W	0	reserved
4-0	R/W	0	reg_mcdi_reldetlpfmsk13. det lpf mask13 for vertical blocks, 0~16, default = 0

MCDI_REL_DET_LPF_MSK_22_30 0x2f3b

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-24	R/W	16	reg_mcdi_reldetlpfmsk30. det lpf mask30 for other blocks, 0~16, default = 16
23-21	R/W	0	reserved
20-16	R/W	1	reg_mcdi_reldetlpfmsk24. det lpf mask24 for horizontal blocks, 0~16, default = 1
15-13	R/W	0	reserved
12-8	R/W	0	reg_mcdi_reldetlpfmsk23. det lpf mask23 for horizontal blocks, 0~16, default = 0
7-5	R/W	0	reserved
4-0	R/W	1	reg_mcdi_reldetlpfmsk22. det lpf mask22 for horizontal blocks, 0~16, default = 1

MCDI_REL_DET_LPF_MSK_31_34 0x2f3c

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-24	R/W	0	reg_mcdi_reldetlpfmsk34. det lpf mask34 for other blocks, 0~16, default = 0
23-21	R/W	0	reserved
20-16	R/W	0	reg_mcdi_reldetlpfmsk33. det lpf mask33 for other blocks, 0~16, default = 0
15-13	R/W	0	reserved
12-8	R/W	0	reg_mcdi_reldetlpfmsk32. det lpf mask32 for other blocks, 0~16, default = 0
7-5	R/W	0	reserved
4-0	R/W	0	reg_mcdi_reldetlpfmsk31. det lpf mask31 for other blocks, 0~16, default = 0

MCDI_REL_DET_MIN 0x2f3d

Bit(s)	R/W	Default	Description
31-7	R/W	0	reserved
6-0	R/W	16	reg_mcdi_reldetmin. min of detected value, default = 16

MCDI_REL_DET_LUT_0_3 0x2f3e

Bit(s)	R/W	Default	Description
31-24	R/W	8	reg_mcdi_reldetmaplut3. default = 8
23-16	R/W	4	reg_mcdi_reldetmaplut2. default = 4
15-8	R/W	2	reg_mcdi_reldetmaplut1. default = 2
7-0	R/W	0	reg_mcdi_reldetmaplut0. default = 0

MCDI_REL_DET_LUT_4_7 0x2f3f

Bit(s)	R/W	Default	Description
31-24	R/W	64	reg_mcdi_reldetmaplut7. default = 64
23-16	R/W	48	reg_mcdi_reldetmaplut6. default = 48
15-8	R/W	32	reg_mcdi_reldetmaplut5. default = 32
7-0	R/W	16	reg_mcdi_reldetmaplut4. default = 16

MCDI_REL_DET_LUT_8_11 0x2f40

Bit(s)	R/W	Default	Description
31-24	R/W	160	reg_mcdi_reldetmaplut11. default = 160
23-16	R/W	128	reg_mcdi_reldetmaplut10. default = 128
15-8	R/W	96	reg_mcdi_reldetmaplut9. default = 96
7-0	R/W	80	reg_mcdi_reldetmaplut8. default = 80

MCDI_REL_DET_LUT_12_15 0x2f41

Bit(s)	R/W	Default	Description
31-24	R/W	255	reg_mcdi_reldetmaplut15. default = 255
23-16	R/W	240	reg_mcdi_reldetmaplut14. default = 240
15-8	R/W	224	reg_mcdi_reldetmaplut13. default = 224
7-0	R/W	192	reg_mcdi_reldetmaplut12. default = 192

MCDI_REL_DET_COL_CFD_THD 0x2f42

Bit(s)	R/W	Default	Description
31-24	R/W	5	reg_mcdi_reldetcolcfdfltthd. thd for flat smaller than (<) of column cofidence, default = 5
23-16	R/W	160	reg_mcdi_reldetcolcfdthd1. thd for rel larger than (>=) in rel calc. mode col confidence without gmv locking, default = 160
15-8	R/W	100	reg_mcdi_reldetcolcfdthd0. thd for rel larger than (>=) in rel calc. mode col confidence when gmv locked, default = 100

7-2	R/W	16	reg_mcdi_reldetcolcfdbadwthd. thd for badw larger than (>=) in qbadw calc. mode of column cofidence, default = 16
1	R/W	0	reserved
0	R/W	0	reg_mcdi_reldetcolcfdcalcmode. calc. mode for column cofidence, 0- use rel, 1: use qbadw, default = 0

MCDI_REL_DET_COL_CFD_AVG_LUMA 0x2f43

Bit(s)	R/W	Default	Description
31-24	R/W	235	reg_mcdi_reldetcolcfdavgmin1. avg luma min1 (>=) for column cofidence, valid between 16~235, default = 235
23-16	R/W	235	reg_mcdi_reldetcolcfdavgmax1. avg luma max1 (<) for column cofidence, valid between 16~235, default = 235
15-8	R/W	16	reg_mcdi_reldetcolcfdavgmin0. avg luma min0 (>=) for column cofidence, valid between 16~235, default = 16
7-0	R/W	21	reg_mcdi_reldetcolcfdavgmax0. avg luma max0 (<) for column cofidence, valid between 16~235, default = 21

MCDI_REL_DET_BAD_THD_0 0x2f44

Bit(s)	R/W	Default	Description
31-16	R/W	0	reserved
15-8	R/W	120	reg_mcdi_reldetbadsadthd. thd (>=) for bad sad, default = 120 (480/4)
7-6	R/W	0	reserved
5-0	R/W	12	reg_mcdi_reldetbadbadwthd. thd (>=) for bad badw, 0~42, default = 12

MCDI_REL_DET_BAD_THD_1 0x2f45

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved
23-16	R/W	4	reg_mcdi_reldetbadrelfltthd. thd (>=) of flat for bad rel detection, default = 4
15-8	R/W	160	reg_mcdi_reldetbadrelthd1. thd (>=) for bad rel without gmv/lmv locked, default = 160
7-0	R/W	120	reg_mcdi_reldetbadrelthd0. thd (>=) for bad rel with gmv/lmv locked, default = 120

MCDI_PD22_CHK_THD 0x2f46

Bit(s)	R/W	Default	Description
31-25	R/W	0	reserved
24-16	R/W	64	reg_mcdi_pd22chksaddifthd. sad dif thd (>=) for (pd22chksad - qsad) for pd22 check, default = 64
15-14	R/W	0	reserved

13-8	R/W	2	reg_mcdi_pd22chkqmvthd. thd (>=) of abs qmv for pd22 check, default = 2
7-0	R/W	4	reg_mcdi_pd22chkfltthd. thd (>=) of flat for pd22 check, default = 4

MCDI_PD22_CHK_GAIN_OFFST_0 0x2f47

Bit(s)	R/W	Default	Description
31-24	R/W	0	reg_mcdi_pd22chkedgeoffst0. offset0 of pd22chkedge from right film22 phase, -128~127, default = 0
23-21	R/W	0	reserved
20-16	R/W	16	reg_mcdi_pd22chkedgegain0. gain0 of pd22chkedge from right film22 phase, normalized to 16 as '1', default = 16
15-12	R/W	0	reserved
11-8	R/W	0	reg_mcdi_pd22chkbadwoffst0. offset0 of pd22chkbadw from right film22 phase, -8~7, default = 0
7-5	R/W	0	reserved
4-0	R/W	8	reg_mcdi_pd22chkbadwgain0. gain0 of pd22chkbadw from right film22 phase, normalized to 16 as '1', default = 8

MCDI_PD22_CHK_GAIN_OFFST_1 0x2f48

Bit(s)	R/W	Default	Description
31-24	R/W	0	reg_mcdi_pd22chkedgeoffst1. offset1 of pd22chkedge from right film22 phase, -128~127, default = 0
23-21	R/W	0	reserved
20-16	R/W	16	reg_mcdi_pd22chkedgegain1. gain1 of pd22chkedge from right film22 phase, normalized to 16 as '1', default = 16
15-12	R/W	0	reserved
11-8	R/W	0	reg_mcdi_pd22chkbadwoffst1. offset1 of pd22chkbadw from right film22 phase, -8~7, default = 0
7-5	R/W	0	reserved
4-0	R/W	12	reg_mcdi_pd22chkbadwgain1. gain1 of pd22chkbadw from right film22 phase, normalized to 16 as '1', default = 12

MCDI_LMV_LOCK_CNT_THD_GAIN 0x2f49

Bit(s)	R/W	Default	Description
31-20	R/W	0	reserved
19-16	R/W	6	reg_mcdi_lmvlockcntmax. max lmv lock count number, default = 6
15-12	R/W	0	reg_mcdi_lmvlockcntoffst. offset for lmv lock count, -8~7, default = 0
11-8	R/W	8	reg_mcdi_lmvlockcntgain. gain for lmv lock count, normalized 8 as '1', 15 is set to 16, default = 8

Bit(s)	R/W	Default	Description
7-5	R/W	0	reserved
4-0	R/W	4	reg_mcdi_lmvlockcntthd. lmv count thd (>=) before be locked, 1~31, default = 4

MCDI_LMV_LOCK_ABS_DIF_THD 0x2f4a

Bit(s)	R/W	Default	Description
31-27	R/W	0	reserved
26-24	R/W	1	reg_mcdi_lmvlockdifthd2. lmv dif thd for third part, before locked, default = 1
23	R/W	0	reserved
22-20	R/W	1	reg_mcdi_lmvlockdifthd1. lmv dif thd for second part, before locked, default = 1
19	R/W	0	reserved
18-16	R/W	1	reg_mcdi_lmvlockdifthd0. lmv dif thd for first part, before locked, default = 1
15-13	R/W	0	reserved
12-8	R/W	24	reg_mcdi_lmvlockabsmax. max abs (<) of lmv to be locked, default = 24
7-5	R/W	0	reserved
4-0	R/W	1	reg_mcdi_lmvlockabsmin. min abs (>=) of lmv to be locked, default = 1

MCDI_LMV_LOCK_ROW 0x2f4b

Bit(s)	R/W	Default	Description
31-27	R/W	0	reserved
26-16	R/W	2047	reg_mcdi_lmvlockendrow. end row (<) for lmv lock, default = 2047
15-11	R/W	0	reserved
10-0	R/W	0	reg_mcdi_lmvlockstartrow. start row (>=) for lmv lock, default = 0

MCDI_LMV_LOCK_RT_MODE 0x2f4c

Bit(s)	R/W	Default	Description
31-27	R/W	0	reserved
26-24	R/W	2	reg_mcdi_lmvlockextmode. extend lines for lmv lock check, check how many lines for lmv locking, default = 2
23-16	R/W	32	reg_mcdi_lmvlockfltcntrt. ratio of flt cnt for lock check, normalized 256 as '1', 255 is set to 256, default = 32
15-8	R/W	48	reg_mcdi_lmvlocklmvcntrt1. ratio when use non-zero lmv for lock check, normalized 256 as '1', 255 is set to 256, default = 48
7-0	R/W	106	reg_mcdi_lmvlocklmvcntrt0. ratio when use max lmv for lock check, normalized 256 as '1', 255 is set to 256, default = 106

MCDI_GMV_LOCK_CNT_THD_GAIN 0x2f4d

Bit(s)	R/W	Default	Description
31-20	R/W	0	reserved
19-16	R/W	6	reg_mcdi_gmvlockcntmax. max gmv lock count number, default = 6
15-12	R/W	0	reg_mcdi_gmvlockcntoffst. offset for gmv lock count, -8~7, default = 0
11-8	R/W	8	reg_mcdi_gmvlockcntgain. gain for gmv lock count, normalized 8 as '1', 15 is set to 16, default = 8
7-5	R/W	0	reserved
4-0	R/W	4	reg_mcdi_gmvlockcntthd. gmv count thd (>=) before be locked, 1~31, default = 4

MCDI_GMV_LOCK_ABS_DIF_THD 0x2f4e

Bit(s)	R/W	Default	Description
31-27	R/W	0	reserved
26-24	R/W	3	reg_mcdi_gmvlockdifthd2. gmv dif thd for third part, before locked, default = 3
23	R/W	0	reserved
22-20	R/W	2	reg_mcdi_gmvlockdifthd1. gmv dif thd for second part, before locked, default = 2
19	R/W	0	reserved
18-16	R/W	1	reg_mcdi_gmvlockdifthd0. gmv dif thd for first part, before locked, default = 1
15-13	R/W	0	reserved
12-8	R/W	15	reg_mcdi_gmvlockabsmax. max abs of gmv to be locked, default = 15
7-5	R/W	0	reserved
4-0	R/W	1	reg_mcdi_gmvlockabsmin. min abs of gmv to be locked, default = 1

MCDI_HIGH_VERT_FRQ_DIF_THD 0x2f4f

Bit(s)	R/W	Default	Description
31-0	R/W	103680	reg_mcdi_highvertfrqfldavgdifthd. high_vert_frq field average luma dif thd (>=), 3*Blk_Width*Blk_Height, set by software, default = 103680

MCDI_HIGH_VERT_FRQ_DIF_DIF_THD 0x2f50

Bit(s)	R/W	Default	Description
31-0	R/W	103680	reg_mcdi_highvertfrqfldavgdifdifthd. high_vert_frq field average luma dif's dif thd (<), 3*Blk_Width*Blk_Height, set by software, default = 103680

MCDI_HIGH_VERT_FRQ_RT_GAIN 0x2f51

Bit(s)	R/W	Default	Description

31-20	R/W	0	reserved
19-16	R/W	4	reg_mcdi_highvertfrqcntthd. high_vert_frq count thd (>=) before locked, 1~31, default = 4
15-8	R/W	24	reg_mcdi_highvertfrqbadsadrt. ratio for high_vert_frq bad sad count, normalized 256 as '1', 255 is set to 256, default = 24
7-0	R/W	130	reg_mcdi_highvertfrqbadbadwrt. ratio for high_vert_frq badw count, normalized 256 as '1', 255 is set to 256, default = 130

MCDI_MOTION_PARADOX_THD 0x2f52

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-24	R/W	4	reg_mcdi_motionparadoxcntthd. motion paradox count thd (>=) before locked, 1~31, default = 4
23-22	R/W	0	reserved
21-16	R/W	32	reg_mcdi_motionparadoxgmvthd. abs gmv thd (<) of motion paradox, 0~32, note that 32 means invalid gmv, be careful, default = 32
15-0	R/W	0	reserved

MCDI_MOTION_PARADOX_RT 0x2f53

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved
23-16	R/W	24	reg_mcdi_motionparadoxbadsadrt. ratio for field bad sad count of motion paradox, normalized 256 as '1', 255 is set to 256, default = 24
15-8	R/W	120	reg_mcdi_motionparadoxbadrelrt. ratio for field bad reliabilty count of motion paradox, normalized 256 as '1', 255 is set to 256, default = 120
7-0	R/W	218	reg_mcdi_motionparadoxmtnrt. ratio for field motion count of motion paradox, normalized 256 as '1', 255 is set to 256, default = 218

MCDI_MOTION_REF_THD 0x2f54

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved
23-20	R/W	15	reg_mcdi_motionrefoffst. motion ref additive offset, default = 15
19-16	R/W	8	reg_mcdi_motionrefgain. motion ref gain, normalized 8 as '1', default = 8
15-13	R/W	0	reserved
12-8	R/W	1	reg_mcdi_motionrefrptmvthd. abs thd (>=) of rpt mv (0~31, 32 means invalid) for motion ref, default = 1

7-2	R/W	2	reg_mcdi_motionrefqmvthd. min thd (>=) of abs qmv for motion ref, note that quarter mv's range is -63~63, default = 2
1-0	R/W	1	reg_mcdi_motionreflpfmode. Mv and (8 x repeat flg) 's lpf mode of motion refinement, 0: no lpf, 1: [1 2 1], 2: [1 2 2 2 1], default = 1

MCDI_REL_COL_REF_RT 0x2f55

Bit(s)	R/W	Default	Description
31-8	R/W	0	reserved
7-0	R/W	135	reg_mcdi_relcolrefrt. ratio for column cofidence level against column number, for refinement, default = 135

MCDI_PD22_CHK_THD_RT 0x2f56

Bit(s)	R/W	Default	Description
31-27	R/W	0	reserved
26-16	R/W	1	reg_mcdi_pd22chkfltcntrt. ratio for flat count of field pulldown 22 check, normalized 2048 as '1', 2047 is set to 2048, default = 1
15-8	R/W	100	reg_mcdi_pd22chkcntrt. ratio of pulldown 22 check count, normalized 256 as '1', 255 is set to 256, default = 100
7-5	R/W	0	reserved
4-0	R/W	4	reg_mcdi_pd22chkcntthd. thd (>=) for pd22 count before locked, 1~31, default = 4

MCDI_CHAR_DET_DIF_THD 0x2f57

Bit(s)	R/W	Default	Description
31-24	R/W	0	reserved
23-16	R/W	64	reg_mcdi_chardetminmaxdifthd. thd (>=) for dif between min and max value, default = 64
15-8	R/W	17	reg_mcdi_chardetmaxdifthd. thd (<) for dif between max value, default = 17
7-0	R/W	17	reg_mcdi_chardetmindifthd. thd (<) for dif between min value, default = 17

MCDI_CHAR_DET_CNT_THD 0x2f58

Bit(s)	R/W	Default	Description
31-21	R/W	0	reserved
20-16	R/W	18	reg_mcdi_chardettotcntthd. thd (>=) for total count, 0~21, default = 18
15-13	R/W	0	reserved
12-8	R/W	1	reg_mcdi_chardetmaxcntthd. thd (>=) for max count, 0~21, default = 1
7-5	R/W	0	reserved
4-0	R/W	1	reg_mcdi_chardetmincntthd. thd (>=) for min count, 0~21, default = 1

MCDI_PD_22_CHK_WND0_X 0x2f59

Bit(s)	R/W	Default	Description
28-16	R/W	719	reg_mcdi_pd22chkwnd0_x1
12-0	R/W	0	reg_mcdi_pd22chkwnd0_x0

MCDI_PD_22_CHK_WND0_Y 0x2f5a

Bit(s)	R/W	Default	Description
28-16	R/W	39	reg_mcdi_pd22chkwnd0_y1
12-0	R/W	0	reg_mcdi_pd22chkwnd0_y0

MCDI_PD_22_CHK_WND1_X 0x2f5b

Bit(s)	R/W	Default	Description
28-16	R/W	719	reg_mcdi_pd22chkwnd1_x1
12-0	R/W	0	reg_mcdi_pd22chkwnd1_x0

MCDI_PD_22_CHK_WND1_Y 0x2f5c

Bit(s)	R/W	Default	Description
28-16	R/W	199	reg_mcdi_pd22chkwnd1_y1
12-0	R/W	40	reg_mcdi_pd22chkwnd1_y0

MCDI_PD_22_CHK_FRC_LMV 0x2f5d

Bit(s)	R/W	Default	Description
10	R/W	1	reg_mcdi_pd22chklmvchk2
9	R/W	0	reg_mcdi_pd22chklmvchk1
8	R/W	0	reg_mcdi_pd22chklmvchk0
6	R/W	0	reg_mcdi_pd22chkfrcpd2
5	R/W	0	reg_mcdi_pd22chkfrcpd1
4	R/W	0	reg_mcdi_pd22chkfrcpd0
2	R/W	1	reg_mcdi_pd22chkfrcvof2
1	R/W	0	reg_mcdi_pd22chkfrcvof1
0	R/W	0	reg_mcdi_pd22chkfrcvof0

MCDI_PD_22_CHK_FRC_LMV 0x2f5e

Bit(s)	R/W	Default	Description
26	R/W	0	reg_mcdi_pd22chkflg2

25	R/W	0	reg_mcdi_pd22chkflg1
24	R/W	0	reg_mcdi_pd22chkflg
23-16	R/W	1	reg_mcdi_pd22chkcnt2
15-8	R/W	0	reg_mcdi_pd22chkcnt1
7-0	R/W	0	reg_mcdi_pd22chkcnt

MCDI_FIELD_MV 0x2f60

Bit(s)	R/W	Default	Description
31-24	R/W	0	reg_mcdi_pd22chkcnt
23-16	R/W	0	reg_mcdi_fieldgmvcnt
15	R/W	0	reg_mcdi_pd22chkflg
14	R/W	0	reg_mcdi_fieldgmvlock
13-8	R/W	0	reg_mcdi_fieldrptmv. last field rpt mv
7-6	R/W	0	reserved
5-0	R/W	0	reg_mcdi_fieldgmv. last field gmv

MCDI_FIELD_HVF_PRDX_CNT 0x2f61

Bit(s)	R/W	Default	Description
31-24	R/W	0	reg_mcdi_motionparadoxcnt.
23-17	R/W	0	reserved
16	R/W	0	reg_mcdi_motionparadoxflg.
15-8	R/W	0	reg_mcdi_highvertfrqcnt.
7-4	R/W	0	reserved
3-2	R/W	0	reg_mcdi_highvertfrqphase.
1	R/W	0	reserved
0	R/W	0	reg_mcdi_highvertfrqflg.

MCDI_FIELD_LUMA_AVG_SUM_0 0x2f62

Bit(s)	R/W	Default	Description
31-0	R/W	0	reg_mcdi_fld_luma_avg_sum0.

MCDI_FIELD_LUMA_AVG_SUM_1 0x2f63

Bit(s)	R/W	Default	Description
31-0	R/W	0	reg_mcdi_fld_luma_avg_sum1.

MCDI_YCBCR_BLEND_CRTL 0x2f64

Bit(s)	R/W	Default	Description
31-16	R/W	0	reserved
15-8	R/W	0	reg_mcdi_ycbcrblendgain. ycbcr blending gain for cbcr in ycbcr. default = 0
7-2	R/W	0	reserved.
1-0	R/W	2	reg_mcdi_ycbcrblendmode. 0:y+cmb(cb,cr), 1:med(r,g,b), 2:max(r,g,b), default = 2

MCDI_MC_CRTL 0x2f70

Bit(s)	R/W	Default	Description
31-20	R/W	0	reserved
19	R/W	0	reg buf1 enable (if1)
18	R/W	0	reg buf2 enable (if2)
17	R/W	0	reg mv invert
16	R/W	0	mcvec force 0
15	R/W	0	buf2 always en
14-12	R/W	0	reg_mcdi_mcvec_offset: 0: disable 1: 1 pixel offset of mcvec 2: 2 pixel offset of mcvec 3: 3 pixel offset of mcvec 4: 4 pixel offset of mcvec
11	R/w	0	reg_di_weave_both_side
10	R/W	0	reg_mcdi_mc_uv_en: mc for uv if needed, else use ma of uv
9-8	R/W	1	reg_mcdi_mcpreflg. flag to use previous field for MC, 0-forward field, 1: previous field,2-use forward & previous. default = 1
7	R/W	1	reg_mcdi_mcrelrefbycolcfden. enable rel refinement by column cofidence in mc blending, default = 1
6-5	R/W	0	reg_mcdi_mclpfen. enable mc pixles/rel lpf, 0:disable, 1: lpf rel, 2: lpf mc pxls, 3: lpf both rel and mc pxls, default = 0
4-2	R/W	0	reg_mcdi_mcdebugmode. enable mc debug mode, 0:disable, 1: split left/right, 2: split top/bottom, 3: debug mv, 4: debug rel, default = 0
1-0	R/W	1	reg_mcdi_mcen. mcdi enable mode, 0:disable, 1: blend with ma, 2: full mc, default = 1

MCDI_MC_LPF_MSK_0 0x2f71

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved

28-21	R	0	debug info
20-16	R/W	0	reg_mcdi_mclpfmsk02. mc lpf coef. 2 for pixel 0 of current block, normalized 16 as '1', default = 0
15-13	R/W	0	reserved
12-8	R/W	9	reg_mcdi_mclpfmsk01. mc lpf coef. 1 for pixel 0 of current block, normalized 16 as '1', default = 9
7-5	R/W	0	reserved
4-0	R/W	7	reg_mcdi_mclpfmsk00. mc lpf coef. 0 for pixel 0 of current block, normalized 16 as '1', default = 7

MCDI_MC_LPF_MSK_1 0x2f72

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-21	R	0	debug info
20-16	R/W	0	reg_mcdi_mclpfmsk12. mc lpf coef. 2 for pixel 1 of current block, 0~16, normalized 16 as '1', default = 0
15-13	R/W	0	reserved
12-8	R/W	11	reg_mcdi_mclpfmsk11. mc lpf coef. 1 for pixel 1 of current block, 0~16, normalized 16 as '1', default = 11
7-5	R/W	0	reserved
4-0	R/W	5	reg_mcdi_mclpfmsk10. mc lpf coef. 0 for pixel 1 of current block, 0~16, normalized 16 as '1', default = 5

MCDI_MC_LPF_MSK_2 0x2f73

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-21	R	0	debug info
20-16	R/W	1	reg_mcdi_mclpfmsk22. mc lpf coef. 2 for pixel 2 of current block, 0~16, normalized 16 as '1', default = 1
15-13	R/W	0	reserved
12-8	R/W	14	reg_mcdi_mclpfmsk21. mc lpf coef. 1 for pixel 2 of current block, 0~16, normalized 16 as '1', default = 14
7-5	R/W	0	reserved
4-0	R/W	1	reg_mcdi_mclpfmsk20. mc lpf coef. 0 for pixel 2 of current block, 0~16, normalized 16 as '1', default = 1

MCDI_MC_LPF_MSK_3 0x2f74

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-21	R	0	debug info
20-16	R/W	5	reg_mcdi_mclpfmsk32. mc lpf coef. 2 for pixel 3 of current block, 0~16, normalized 16 as '1', default = 5
15-13	R/W	0	reserved
12-8	R/W	11	reg_mcdi_mclpfmsk31. mc lpf coef. 1 for pixel 3 of current block, 0~16, normalized 16 as '1', default = 11
7-5	R/W	0	reserved
4-0	R/W	0	reg_mcdi_mclpfmsk30. mc lpf coef. 0 for pixel 3 of current block, 0~16, normalized 16 as '1', default = 0

MCDI_MC_LPF_MSK_4 0x2f75

Bit(s)	R/W	Default	Description
31-21	R/W	0	reserved
20-16	R/W	7	reg_mcdi_mclpfmsk42. mc lpf coef. 2 for pixel 4 of current block, 0~16, normalized 16 as '1', default = 7
15-13	R/W	0	reserved
12-8	R/W	9	reg_mcdi_mclpfmsk41. mc lpf coef. 1 for pixel 4 of current block, 0~16, normalized 16 as '1', default = 9
7-5	R/W	0	reserved
4-0	R/W	0	reg_mcdi_mclpfmsk40. mc lpf coef. 0 for pixel 4 of current block, 0~16, normalized 16 as '1', default = 0

MCDI_MC_REL_GAIN_OFFST_0 0x2f76

Bit(s)	R/W	Default	Description
31-26	R/W	0	reserved
25	R/W	0	reg_mcdi_mcmotionparadoxflg. flag of motion paradox, initial with 0 and read from software, default = 0
24	R/W	0	reg_mcdi_mchighvertfrqflg. flag of high vert frq, initial with 0 and read from software, default = 0
23-16	R/W	128	reg_mcdi_mcmotionparadoxoffst. offset (r+ offset) for rel (MC blending coef.) refinement if motion paradox detected before MC blending before MC blending, default = 128
15-12	R/W	0	reserved
11-8	R/W	8	reg_mcdi_mcmotionparadoxgain. gain for rel (MC blending coef.) refinement if motion paradox detected before MC blending, normalized 8 as '1', set 15 to 16, default = 8

7-4	R/W	15	reg_mcdi_mchighvertfrqoffst. minus offset (alpha - offset) for motion (MA blending coef.) refinement if high vertical frequency detected before MA blending, default = 15
3-0	R/W	8	reg_mcdi_mchighvertfrqgain. gain for motion (MA blending coef.) refinement if high vertical frequency detected before MA blending, normalized 8 as '1', set 15 to 16, default = 8

MCDI_MC_REL_GAIN_OFFST_1 0x2f77

Bit(s)	R/W	Default	Description
31-24	R/W	255	reg_mcdi_mcoutofboundrayoffst. offset (rel + offset) for rel (MC blending coef.) refinement if MC pointed out of boundray before MC blending before MC blending, default = 255
23-20	R/W	0	reserved
19-16	R/W	8	reg_mcdi_mcoutofboundraygain. gain for rel (MC blending coef.) refinement if MC pointed out of boundray before MC blending, normalized 8 as '1', set 15 to 16, default = 8
15-8	R/W	255	reg_mcdi_mcrelrefbycolcfdoffst. offset (rel + offset) for rel (MC blending coef.) refinement if motion paradox detected before MC blending before MC blending, default = 255
7-4	R/W	0	reserved.
3-0	R/W	8	reg_mcdi_mcrelrefbycolcfdgain. gain for rel (MC blending coef.) refinement if column cofidence failed before MC blending, normalized 8 as '1', set 15 to 16, default = 8

MCDI_MC_COL_CFD_0 0x2f78

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_0. column cofidence value 0 read from software. initial = 0

MCDI_MC_COL_CFD_1 0x2f79

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_1. column cofidence value 1 read from software. initial = 0

MCDI_MC_COL_CFD_2 0x2f7a

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_2. column cofidence value 2 read from software. initial = 0

MCDI_MC_COL_CFD_3 0x2f7b

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_3. column cofidence value 3 read from software. initial = 0

MCDI_MC_COL_CFD_4 0x2f7c

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 4 read from software. initial = 0

MCDI_MC_COL_CFD_5 0x2f7d

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 5 read from software. initial = 0

MCDI_MC_COL_CFD_6 0x2f7e

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 6 read from software. initial = 0

MCDI_MC_COL_CFD_7 0x2f7f

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 7 read from software. initial = 0

MCDI_MC_COL_CFD_8 0x2f80

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 8 read from software. initial = 0

MCDI_MC_COL_CFD_9 0x2f81

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 9 read from software. initial = 0

MCDI_MC_COL_CFD_10 0x2f82

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 10 read from software. initial = 0

MCDI_MC_COL_CFD_11 0x2f83

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 11 read from software. initial = 0

MCDI_MC_COL_CFD_12 0x2f84

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 12 read from software. initial = 0

MCDI_MC_COL_CFD_13 0x2f85

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 13 read from software. initial = 0

MCDI_MC_COL_CFD_14 0x2f86

Bit(s)	R/W	Default	Description
31	W/R	0	canvas addr syncen
23-16	W/R	0	canvas addr2
15-8	W/R	0	canvas addr1
7-0	W/R	0	canvas addr0

MCDI_MC_COL_CFD_15 0x2f87

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 15 read from software. initial = 0

MCDI_MC_COL_CFD_16 0x2f88

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 16 read from software. initial = 0

MCDI_MC_COL_CFD_17 0x2f89

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 17 read from software. initial = 0

MCDI_MC_COL_CFD_18 0x2f8a

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 18 read from software. initial = 0

MCDI_MC_COL_CFD_19 0x2f8b

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 19 read from software. initial = 0

MCDI_MC_COL_CFD_20 0x2f8c

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 20 read from software. initial = 0

MCDI_MC_COL_CFD_21 0x2f8d

Bit(s)	R/W	Default	Description
31-0	R/W		mcdi_mc_col_cfd_4. column cofidence value 21 read from software. initial = 0

MCDI_MC_COL_CFD_22 0x2f8e

Bit(s)	R/W	Default	Description

31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 22 read from software. initial = 0

MCDI_MC_COL_CFD_23 0x2f8f

Bit(s)	R/W	Default	Description
31-0	R/W	0	mcdi_mc_col_cfd_4. column cofidence value 23 read from software. initial = 0

MCDI_MC_COL_CFD_24 0x2f90

Bit(s)	R/W	Default	Description
31-0	R/W		mcdi_mc_col_cfd_4. column cofidence value 24 read from software. initial = 0

MCDI_MC_COL_CFD_25 0x2f91

Bit(s)	R/W	Default	Description
31-0	R/W		mcdi_mc_col_cfd_4. column cofidence value 25 read from software. initial = 0

MCDI_RO_FLD_LUMA_AVG_SUM 0x2fa0

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_fldlumaavgsum. block's luma avg sum of current filed (block based). initial = 0

MCDI_RO_GMV_VLD_CNT 0x2fa1

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_gmvvldcnt. valid gmv's count of pre one filed (block based). initial = 0

MCDI_RO_RPT_FLG_CNT 0x2fa2

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_rptflgcnt. repeat mv's count of pre one filed (block based). initial = 0

MCDI_RO_FLD_BAD_SAD_CNT 0x2fa3

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_fldbadsadcnt. bad sad count of whole pre one field (block based). initial = 0

MCDI_RO_FLD_BAD_BADW_CNT 0x2fa4

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_fldbadbadwcnt. bad badw count of whole pre one field (block based). initial = 0

MCDI_RO_FLD_BAD_REL_CNT 0x2fa5

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_fldbadrelcnt. bad rel count of whole pre one field (block based). initial = 0

MCDI_RO_FLD_MTN_CNT 0x2fa6

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_fldmtncnt. motion count of whole pre one field (pixel based). initial = 0

MCDI_RO_FLD_VLD_CNT 0x2fa7

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_fldvldcnt. valid motion count of whole pre one field (pixel based). initial = 0

MCDI_RO_FLD_PD_22_PRE_CNT 0x2fa8

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_fldpd22precnt. prevoius pd22 check count of whole pre one field (block based). initial = 0

MCDI_RO_FLD_PD_22_FOR_CNT 0x2fa9

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_fldpd22forcnt. forward pd22 check count of whole pre one field (block based). initial = 0

MCDI_RO_FLD_PD_22_FLT_CNT 0x2faa

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_fldpd22fltcnt. flat count (for pd22 check) of whole pre one field (block based). initial = 0

MCDI_RO_HIGH_VERT_FRQ_FLG 0x2fab

Bit(s)	R/W	Default	Description
31-16	R	0	reserved.
15-8	R	0	ro_mcdi_highvertfrqcnt. high vertical frequency count till prevoius one field. initial = 0
7-3	R	0	reserved.
2-1	R	0	ro_mcdi_highvertfrqphase. high vertical frequency phase of prevoius one field. initial = 2
0	R	0	ro_mcdi_highvertfrqflg. high vertical frequency flag of prevoius one field. initial = 0

MCDI_RO_GMV_LOCK_FLG 0x2fac

Bit(s)	R/W	Default	Description
31-16	R	0	reserved.
15-8	R	0	ro_mcdi_gmvlckcnt. global mv lock count till prevoius one field. initial = 0
7-2	R	0	ro_mcdi_gmv. global mv of prevoius one field. -31~31, initial = 32 (invalid value)
1	R	0	ro_mcdi_zerogmvlckflg. zero global mv lock flag of prevoius one field. initial = 0

0	R	0	ro_mcdi_gmvlckflg. global mv lock flag of prevoius one field. initial = 0

MCDI_RO_RPT_MV 0x2fad

Bit(s)	R/W	Default	Description
5-0	R	0	ro_mcdi_rptmv. repeate mv of prevoius one field. -31~31, initial = 32 (invalid value)

MCDI_RO_MOTION_PARADOX_FLG 0x2fae

Bit(s)	R/W	Default	Description
31-16	R	0	reserved.
15-8	R	0	ro_mcdi_motionparadoxcnt. motion paradox count till prevoius one field. initial = 0
7-1	R	0	reserved.
0	R	0	ro_mcdi_motionparadoxflg. motion paradox flag of prevoius one field. initial = 0

MCDI_RO_PD_22_FLG 0x2faf

Bit(s)	R/W	Default	Description
31-16	R	0	reserved.
26	R	0	ro_mcdi_pd22flg2. pull down 22 flag of prevoius one field. initial = 0
25	R	0	ro_mcdi_pd22flg1. pull down 22 flag of prevoius one field. initial = 0
24	R	0	ro_mcdi_pd22flg0. pull down 22 flag of prevoius one field. initial = 0
23-16	R	0	ro_mcdi_pd22cnt2. pull down 22 count till prevoius one field. initial = 0
15-8	R	0	ro_mcdi_pd22cnt1. pull down 22 count till prevoius one field. initial = 0
7-0	R	0	ro_mcdi_pd22cnt0. pull down 22 count till prevoius one field. initial = 0

MCDI_RO_COL_CFD_0 0x2fb0

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_0. column cofidence value 0. initial = 0

MCDI_RO_COL_CFD_1 0x2fb1

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_1. column cofidence value 1. initial = 0

MCDI_RO_COL_CFD_2 0x2fb2

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_2. column cofidence value 2. initial = 0

MCDI_RO_COL_CFD_3 0x2fb3

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_3. column cofidence value 3. initial = 0

MCDI_RO_COL_CFD_4 0x2fb4

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_4. column cofidence value 4. initial = 0

MCDI_RO_COL_CFD_5 0x2fb5

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_5. column cofidence value 5. initial = 0

MCDI_RO_COL_CFD_6 0x2fb6

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_6. column cofidence value 6. initial = 0

MCDI_RO_COL_CFD_7 0x2fb7

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_7. column cofidence value 7. initial = 0

MCDI_RO_COL_CFD_8 0x2fb8

Bit(s)	R/W	Default	Description
31-0	R		ro_mcdi_col_cfd_8. column cofidence value 8. initial = 0

MCDI_RO_COL_CFD_9 0x2fb9

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_9. column cofidence value 9. initial = 0

MCDI_RO_COL_CFD_10 0x2fba

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_10. column cofidence value 10. initial = 0

MCDI_RO_COL_CFD_11 0x2fbb

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_11. column cofidence value 11. initial = 0

MCDI_RO_COL_CFD_12 0x2fbc

Bit(s)	R/W	Default	Description

31-0	R	0	ro_mcdi_col_cfd_12. column cofidence value 12. initial = 0

MCDI_RO_COL_CFD_13 0x2fbd

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_13. column cofidence value 13. initial = 0

MCDI_RO_COL_CFD_14 0x2fbe

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_14. column cofidence value 14. initial = 0

MCDI_RO_COL_CFD_15 0x2fbf

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_15. column cofidence value 15. initial = 0

MCDI_RO_COL_CFD_16 0x2fc0

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_16. column cofidence value 16. initial = 0

MCDI_RO_COL_CFD_17 0x2fc1

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_17. column cofidence value 17. initial = 0

MCDI_RO_COL_CFD_18 0x2fc2

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_18. column cofidence value 18. initial = 0

MCDI_RO_COL_CFD_19 0x2fc3

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_19. column cofidence value 19. initial = 0

MCDI_RO_COL_CFD_20 0x2fc4

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_20. column cofidence value 20. initial = 0

MCDI_RO_COL_CFD_21 0x2fc5

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_21. column cofidence value 21. initial = 0

MCDI_RO_COL_CFD_22 0x2fc6

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_22. column cofidence value 22. initial = 0

MCDI_RO_COL_CFD_23 0x2fc7

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_23. column cofidence value 23. initial = 0

MCDI_RO_COL_CFD_24 0x2fc8

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_24. column cofidence value 24. initial = 0

MCDI_RO_COL_CFD_25 0x2fc9

Bit(s)	R/W	Default	Description
31-0	R	0	ro_mcdi_col_cfd_25. column cofidence value 25. initial = 0

MCDI_RO_FLD_PD_22_PRE_CNT1 0x2fca

Bit(s)	R/W	Default	Description
31-0	R	0	previous pd22 check count of whole pre one field(block based). initial = 0

MCDI_RO_FLD_PD_22_POR_CNT1 0x2fcb

Bit(s)	R/W	Default	Description
31-0	R	0	forward pd22 check count of whole pre one field(block based). initial = 0

MCDI_RO_FLD_PD_22_FLT_CNT1 0x2fcc

Bit(s)	R/W	Default	Description
31-0	R	0	flat count(for pd22 check) of whole pre one field(block based). initial = 0

MCDI_RO_FLD_PD_22_PRE_CNT2 0x2fcd

Bit(s)	R/W	Default	Description
31-0	R	0	previous pd22 check count of whole pre one field(block based). initial = 0

MCDI_RO_FLD_PD_22_POR_CNT2 0x2fce

Bit(s)	R/W	Default	Description
31-0	R	0	forward pd22 check count of whole pre one field(block based). initial = 0

MCDI_RO_FLD_PD_22_FLT_CNT2 0x2fcf

Bit(s)	R/W	Default	Description

31-0	R	0	flat count(for pd22 check) of whole pre one field(block based). initial = 0

10.2.3.14 PULLDOWN

DIPD_COMB_CTRL0 0x2fd0

Bit(s)	R/W	Default	Description
31-24	W	0	Cmb_v_dif_min
23-16	W	0	Cmb_v_dif_max
15-8	W	0	Cmb_crg_min
7-0	W	0	Cmb_crg_max

DIPD_COMB_CTRL1 0x2fd1

Bit(s)	R/W	Default	Description
31	W	0	Pd_check_en
29-24	W	0	Cmb_wv_min3
21-16	W	0	Cmb_wv_min2
13-8	W	0	Cmb_wv_min1
5-0	W	0	Cmb_wv_min0

DIPD_COMB_CTRL2 0x2fd2

Bit(s)	R/W	Default	Description
31-28	W	0	Cmb_wnd_cnt1
25-20	W	0	Ccnt_cmmin1
19-16	W	0	Ccnt_mtmin
13-8	W	0	Ccnt_cmmin
5-0	W	0	Cmb_wv_min4

DIPD_COMB_CTRL3 0x2fd3

Bit(s)	R/W	Default	Description
31	W	0	Cmb32spcl
17-12	W	0	Cmb_wnd_mthd
11-4	W	0	Cmb_abs_nocmb
3-0	W	0	Cnt_minlen

		0	

DIPD_COMB_CTRL4 0x2fd4

Bit(s)	R/W	Default	Description
30	W	0	Flm_stamtn_en
29-28	W	0	In_horflt
27-20	W	0	Alpha
19-16	W	0	Rhtran_ctmtd
15-8	W	0	Htran_mnth1
7-0	W	0	Htran_mnth0

DIPD_COMB_CTRL5 0x2fd5

Bit(s)	R/W	Default	Description
31-24	W	0	Fld_mindif
23-16	W	0	Frm_mindif
13-8	W	0	Flm_smp_mtn_cnt
7-0	W	0	Flm_smp_mtn_thd

DIPD_RO_COMB_0 0x2fd6

Bit(s)	R/W	Default	Description
31-0	R	0	frmdif

DIPD_RO_COMB_1 0x2fd7

Bit(s)	R/W	Default	Description
31-0	R	0	Frmdif0

DIPD_RO_COMB_2 0x2fd8

Bit(s)	R/W	Default	Description
31-0	R	0	Frmdif1

DIPD_RO_COMB_3 0x2fd9

Bit(s)	R/W	Default	Description
31-0	R	0	Frmdif2

DIPD_RO_COMB_4 0x2fda

Bit(s)	R/W	Default	Description

31-0	R	0	Frmdif3

DIPD_RO_COMB_5 0x2fdb

Bit(s)	R/W	Default	Description
31-0	R	0	Frmdif4

DIPD_RO_COMB_6 0x2fdc

Bit(s)	R/W	Default	Description
31-0	R	0	flddif

DIPD_RO_COMB_7 0x2fdd

Bit(s)	R/W	Default	Description
31-0	R	0	Flddif0

DIPD_RO_COMB_8 0x2fde

Bit(s)	R/W	Default	Description
31-0	R	0	Flddif1

DIPD_RO_COMB_9 0x2fdf

Bit(s)	R/W	Default	Description
31-0	R	0	Flddif2

DIPD_RO_COMB_10 0x2fe0

Bit(s)	R/W	Default	Description
31-0	R	0	Flddif3

DIPD_RO_COMB_11 0x2fe1

Bit(s)	R/W	Default	Description
31-0	R	0	Flddif4

DIPD_RO_COMB_12 0x2fe2

Bit(s)	R/W	Default	Description
31-0	R	0	Ro_rt0

DIPD_RO_COMB_13 0x2fe3

Bit(s)	R/W	Default	Description
31-0	R	0	Ro_rt1

DIPD_RO_COMB_14 0x2fe4

Bit(s)	R/W	Default	Description
31-0	R	0	Ro_rt2

DIPD_RO_COMB_15 0x2fe5

Bit(s)	R/W	Default	Description
31-0	R	0	Ro_rt3

DIPD_RO_COMB_16 0x2fe6

Bit(s)	R/W	Default	Description
31-0	R	0	Ro_rt4

DIPD_RO_COMB_17 0x2fe7

Bit(s)	R/W	Default	Description
31-0	R	0	Ro_rt5

DIPD_RO_COMB_18 0x2fe8

Bit(s)	R/W	Default	Description
31-0	R	0	Ro_rt6

DIPD_RO_COMB_19 0x2fe9

Bit(s)	R/W	Default	Description
31-0	R	0	Ro_rt7

DIPD_RO_COMB_20 0x2fea

Bit(s)	R/W	Default	Description
31-0	R	0	Ro_rt8

DIPD_COMB_CTRL5 0x2fd5

Bit(s)	R/W	Default	Description
31-24	W	0	Fld_mindif
23-16	W	0	Frm_mindif
13-8	W	0	Flm_smp_mtn_cnt
7-0	W	0	Flm_smp_mtn_thd

DIPD_COMB_CTRL6 0x2feb

Bit(s)	R/W	Default	Description
21	Rw	0	Reg_edit_sel
20	Rw	1	Reg_horflt_en
16-12	Rw	7	Reg_combseglen
9-4	Rw	6	Reg_trancombrat
3-0	Rw	4	Reg_combsegmin

10.2.3.15 DNR Registers

DNR_CTRL 0x2d00

Bit(s)	R/W	Default	Description
31:17	R/W	0	reserved
16	R/W	0	reg_dnr_en
15	R/W	0	reg_dnr_db_vdbstep , vdb step, 0: 4, 1: 8 . unsigned , default = 1
14	R/W	0	reg_dnr_db_vdbprten , vdb protectoin enable . unsigned , default = 1
13	R/W	0	reg_dnr_gbs_difen , enable dif (between LR and LL/RR) condition for gbs stat.. unsigned , default = 0
12	R/W	0	reg_dnr_luma_en , enable ycbcr2luma module . unsigned , default = 1
11:10	R/W	0	reg_dnr_db_mod , deblocking mode, 0: disable, 1: horizontal deblocking, 2: vertical deblocking, 3: horizontal & vertical deblocking. unsigned , default = 3
9	R/W	0	reg_dnr_db_chrmen , enable chroma deblocking . unsigned , default = 1
8	R/W	0	reg_dnr_hvdif_mod , 0: calc. difs by original Y, 1: by new luma. unsigned , default = 1
7	R/W	0	reserved
6: 4	R/W	0	reg_dnr_demo_lften , b0: Y b1:U b2:V . unsigned , default = 7
3	R/W	0	reserved
2: 0	R/W	0	reg_dnr_demo_rgten , b0: Y b1:U b2:V . unsigned , default = 7

DNR_HVSIZE 0x2d01

Bit(s)	R/W	Default	Description
31:29	R/W	0	reserved
28:16	R/W	0	reg_dnr_hsize , hsize . unsigned , default = 0
15:13	R/W	0	reserved
12: 0	R/W	0	reg_dnr_vsize , vsize . unsigned , default = 0

DNR_DBLK_BLANK_NUM 0x2d02

Bit(s)	R/W	Default	Description
31:16	R/W	0	reserved
15: 8	R/W	0	reg_dblk_hblank_num , deblock hor blank num . unsigned , default = 16
7: 0	R/W	0	reg_dblk_vblank_num , deblock ver blank num . unsigned , default = 45

DNR_BLK_OFFST 0x2d03

Bit(s)	R/W	Default	Description
31: 7	R/W	0	reserved
6: 4	R/W	0	reg_dnr_hbofst , horizontal block offset may provide by software calc.. unsigned , default = 0
3	R/W	0	reserved
2: 0	R/W	0	reg_dnr_vbofst , vertical block offset may provide by software calc.. unsigned , default = 0

DNR_GBS 0x2d04

Bit(s)	R/W	Default	Description
31: 2	R/W	0	reserved
1: 0	R/W	0	reg_dnr_gbs , global block strength may update by software calc.. unsigned , default = 0

DNR_HBOFFST_STAT 0x2d05

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_hbof_difthd , dif threshold (>=) between LR and LL/RR. unsigned , default = 2
23:16	R/W	0	reg_dnr_hbof_edgethd , edge threshold (<=) for LR . unsigned , default = 32
15: 8	R/W	0	reg_dnr_hbof_flatthd , flat threshold (>=) for LR . unsigned , default = 0
7	R/W	0	reserved
6: 4	R/W	0	reg_dnr_hbof_delta , delta for weighted bin accumulator. unsigned , default = 1
3	R/W	0	reserved
2: 0	R/W	0	reg_dnr_hbof_statmod , statistic mode for horizontal block offset, 0: count flags for 8-bin, 1: count LRs for 8-bin, 2: count difs for 8-bin, 3: count weighted flags for 8-bin, 4: count flags for first 32-bin, 5: count LRs for first 32-bin, 6 or 7: count difs for first 32-bin. unsigned , default = 2

DNR_VBOFFST_STAT 0x2d06

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_vbof_difthd , dif threshold (>=) between Up and Dw. unsigned , default = 1
23:16	R/W	0	reg_dnr_vbof_edgethd , edge threshold (<=) for Up/Dw. unsigned , default = 16
15: 8	R/W	0	reg_dnr_vbof_flatthd , flat threshold (>=) for Up/Dw. unsigned , default = 0
7	R/W	0	reserved
6: 4	R/W	0	reg_dnr_vbof_delta , delta for weighted bin accumulator. unsigned , default = 1
3	R/W	0	reserved
2: 0	R/W	0	reg_dnr_vbof_statmod , statistic mode for vertical block offset, 0: count flags for 8-bin, 1: count Ups for 8-bin, 2: count difs for 8-bin, 3: count weighted flags for 8-bin, 4: count flags for first 32-bin, 5: count Ups for first 32-bin, 6 or 7: count difs for first 32-bin. unsigned , default = 2

DNR_GBS_STAT 0x2d07

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_gbs_edgethd , edge threshold (<=) for LR . unsigned , default = 32
23:16	R/W	0	reg_dnr_gbs_flatthd , flat threshold (>=) for LR . unsigned , default = 0
15: 8	R/W	0	reg_dnr_gbs_varthd , variation threshold (<=) for Lvar/Rvar. unsigned , default = 16
7: 0	R/W	0	reg_dnr_gbs_difthd , dif threshold (>=) between LR and LL/RR. unsigned , default = 2

DNR_STAT_X_START_END 0x2d08

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:16	R/W	0	reg_dnr_stat_xst . unsigned , default = 24
15:14	R/W	0	reserved
13: 0	R/W	0	reg_dnr_stat_xed . unsigned , default = HSIZE - 25

DNR_STAT_Y_START_END 0x2d09

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:16	R/W	0	reg_dnr_stat_yst . unsigned , default = 24
15:14	R/W	0	reserved
13: 0	R/W	0	reg_dnr_stat_yed . unsigned , default = VSIZE - 25

DNR_LUMA 0x2d0a

Bit(s)	R/W	Default	Description
31:27	R/W	0	reserved
26:24	R/W	0	reg_dnr_luma_sqrtshft , left shift for fast squart of chroma, [0, 4]. unsigned , default = 2
23:21	R/W	0	reserved
20:16	R/W	0	reg_dnr_luma_sqrtoffst , offset for fast squart of chroma. signed , default = 0
15	R/W	0	reserved
14:12	R/W	0	reg_dnr_luma_wcmod , theta related to warm/cool segment line, 0: 0, 1: 45, 2: 90, 3: 135, 4: 180, 5: 225, 6: 270, 7: 315. . unsigned , default = 3
11: 8	R/W	0	reg_dnr_luma_cshft , shift for calc. delta part, 0~8, . unsigned , default = 8
7: 6	R/W	0	reserved
5: 0	R/W	0	reg_dnr_luma_cgain , final gain for delta part, 32 normalized to "1". unsigned , default = 4

DNR_DB_YEDGE_THD 0x2d0b

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_db_yedgethd0 , edge threshold0 for luma . unsigned , default = 12
23:16	R/W	0	reg_dnr_db_yedgethd1 , edge threshold1 for luma . unsigned , default = 15
15: 8	R/W	0	reg_dnr_db_yedgethd2 , edge threshold2 for luma . unsigned , default = 18
7: 0	R/W	0	reg_dnr_db_yedgethd3 , edge threshold3 for luma . unsigned , default = 25

DNR_DB_CEDGE_THD 0x2d0c

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_db_cedgethd0 , edge threshold0 for chroma . unsigned , default = 12
23:16	R/W	0	reg_dnr_db_cedgethd1 , edge threshold1 for chroma . unsigned , default = 15
15: 8	R/W	0	reg_dnr_db_cedgethd2 , edge threshold2 for chroma . unsigned , default = 18
7: 0	R/W	0	reg_dnr_db_cedgethd3 , edge threshold3 for chroma . unsigned , default = 25

DNR_DB_HGAP 0x2d0d

Bit(s)	R/W	Default	Description
31:24	R/W	0	reserved
23:16	R/W	0	reg_dnr_db_hgapthd , horizontal gap thd (<=) for very sure blockiness . unsigned , default = 8
15: 8	R/W	0	reg_dnr_db_hgapdifthd , dif thd between hgap and lft/rgt hdifs. unsigned , default = 1
7: 1	R/W	0	reserved
0	R/W	0	reg_dnr_db_hgapmod , horizontal gap calc. mode, 0: just use current col x, 1: find max between (x-1, x, x+1) . unsigned , default = 0

DNR_DB_HBS 0x2d0e

Bit(s)	R/W	Default	Description
31: 6	R/W	0	reserved
5: 4	R/W	0	reg_dnr_db_hbsup , horizontal bs up value . unsigned , default = 1
3: 2	R/W	0	reg_dnr_db_hbsmax , max value of hbs for global control. unsigned , default = 3
1: 0	R/W	0	reg_dnr_db_hgbsthd , gbs thd (>=) for hbs calc. . unsigned , default = 1

DNR_DB_HACT 0x2d0f

Bit(s)	R/W	Default	Description
31:16	R/W	0	reserved
15: 8	R/W	0	reg_dnr_db_hactthd0 , thd0 of hact, for block classification. unsigned , default = 10
7: 0	R/W	0	reg_dnr_db_hactthd1 , thd1 of hact, for block classification. unsigned , default = 32

DNR_DB_YHDELTA_GAIN 0x2d10

Bit(s)	R/W	Default	Description
31:27	R/W	0	reserved
26:24	R/W	0	reg_dnr_db_yhdeltagain1 , (p1-q1) gain for Y's delta calc. when bs=1, normalized 8 as "1" . unsigned , default = 2
23	R/W	0	reserved
22:20	R/W	0	reg_dnr_db_yhdeltagain2 , (p1-q1) gain for Y's delta calc. when bs=2, normalized 8 as "1" . unsigned , default = 0
19	R/W	0	reserved
18:16	R/W	0	reg_dnr_db_yhdeltagain3 , (p1-q1) gain for Y's delta calc. when bs=3, normalized 8 as "1" . unsigned , default = 0
15	R/W	0	reserved

Bit(s)	R/W	Default	Description
14: 8	R/W	0	reg_dnr_db_yhdeltaadjoffst , offset for adjust Y's hdelta (-64, 63). signed , default = 0
7: 6	R/W	0	reserved
5: 0	R/W	0	reg_dnr_db_yhdeltaadjgain , gain for adjust Y's hdelta, normalized 32 as "1" . unsigned , default = 32

DNR_DB_YHDELTA2_GAIN 0x2d11

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:24	R/W	0	reg_dnr_db_yhdelta2gain2 , gain for bs=2's adjust Y's hdelta2, normalized 64 as "1" . unsigned , default = 8
23:21	R/W	0	reserved
20:16	R/W	0	reg_dnr_db_yhdelta2offst2 , offset for bs=2's adjust Y's hdelta2 (-16, 15). signed , default = 0
15:14	R/W	0	reserved
13: 8	R/W	0	reg_dnr_db_yhdelta2gain3 , gain for bs=3's adjust Y's hdelta2, normalized 64 as "1" . unsigned , default = 4
7: 5	R/W	0	reserved
4: 0	R/W	0	reg_dnr_db_yhdelta2offst3 , offset for bs=3's adjust Y's hdelta2 (-16, 15). signed , default = 0

DNR_DB_CHDELTA_GAIN 0x2d12

Bit(s)	R/W	Default	Description
31:27	R/W	0	reserved
26:24	R/W	0	reg_dnr_db_chdeltagain1 , (p1-q1) gain for UV's delta calc. when bs=1, normalized 8 as "1". unsigned , default = 2
23	R/W	0	reserved
22:20	R/W	0	reg_dnr_db_chdeltagain2 , (p1-q1) gain for UV's delta calc. when bs=2, normalized 8 as "1". unsigned , default = 0
19	R/W	0	reserved
18:16	R/W	0	reg_dnr_db_chdeltagain3 , (p1-q1) gain for UV's delta calc. when bs=3, normalized 8 as "1". unsigned , default = 0
15	R/W	0	reserved
14: 8	R/W	0	reg_dnr_db_chdeltaadjoffst , offset for adjust UV's hdelta (-64, 63). signed , default = 0
7: 6	R/W	0	reserved

Bit(s)	R/W	Default	Description
5: 0	R/W	0	reg_dnr_db_chdeltaadjgain , gain for adjust UV's hdelta, normalized 32 as "1". unsigned , default = 32

DNR_DB_CHDELTA2_GAIN 0x2d13

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:24	R/W	0	reg_dnr_db_chdelta2gain2 , gain for bs=2's adjust UV's hdelta2, normalized 64 as "1" . unsigned , default = 8
23:21	R/W	0	reserved
20:16	R/W	0	reg_dnr_db_chdelta2offst2 , offset for bs=2's adjust UV's hdelta2 (-16, 15). signed , default = 0
15:14	R/W	0	reserved
13: 8	R/W	0	reg_dnr_db_chdelta2gain3 , gain for bs=2's adjust UV's hdelta2, normalized 64 as "1" . unsigned , default = 4
7: 5	R/W	0	reserved
4: 0	R/W	0	reg_dnr_db_chdelta2offst3 , offset for bs=2's adjust UV's hdelta2 (-16, 15). signed , default = 0

DNR_DB_YC_VEDGE_THD 0x2d14

Bit(s)	R/W	Default	Description
31:16	R/W	0	reserved
15: 8	R/W	0	reg_dnr_db_yvedgethd , special Y's edge thd for vdb. unsigned , default = 12
7: 0	R/W	0	reg_dnr_db_cvedgethd , special UV's edge thd for vdb. unsigned , default = 12

DNR_DB_VBS_MISC 0x2d15

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_db_vgapthd , vertical gap thd (<=) for very sure blockiness . unsigned , default = 8
23:16	R/W	0	reg_dnr_db_vactthd , thd of vact, for block classification . unsigned , default = 10
15: 8	R/W	0	reg_dnr_db_vgapdifthd , dif thd between vgap and vact. unsigned , default = 4
7: 4	R/W	0	reserved
3: 2	R/W	0	reg_dnr_db_vbsmax , max value of vbs for global control. unsigned , default = 2
1: 0	R/W	0	reg_dnr_db_vgbsthd , gbs thd (>=) for vbs calc. . unsigned , default = 1

DNR_DB_YVDELTA_GAIN 0x2d16

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:24	R/W	0	reg_dnr_db_yvdeltaadjgain , gain for adjust Y's vdelta, normalized 32 as "1". unsigned , default = 32
23	R/W	0	reserved
22:16	R/W	0	reg_dnr_db_yvdeltaadjoffst , offset for adjust Y's vdelta (-64, 63). signed , default = 0
15:14	R/W	0	reserved
13: 8	R/W	0	reg_dnr_db_yvdelta2gain , gain for adjust Y's vdelta2, normalized 64 as "1". unsigned , default = 8
7: 5	R/W	0	reserved
4: 0	R/W	0	reg_dnr_db_yvdelta2offst , offset for adjust Y's vdelta2 (-16, 15). signed , default = 0

DNR_DB_CVDELTA_GAIN 0x2d17

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:24	R/W	0	reg_dnr_db_cvdeltaadjgain , gain for adjust UV's vdelta, normalized 32 as "1". unsigned , default = 32
23	R/W	0	reserved
22:16	R/W	0	reg_dnr_db_cvdeltaadjoffst , offset for adjust UV's vdelta (-64, 63). signed , default = 0
15:14	R/W	0	reserved
13: 8	R/W	0	reg_dnr_db_cvdelta2gain , gain for adjust UV's vdelta2, normalized 64 as "1". unsigned , default = 8
7: 5	R/W	0	reserved
4: 0	R/W	0	reg_dnr_db_cvdelta2offst , offset for adjust UV's vdelta2 (-16, 15). signed , default = 0

DNR_RO_GBS_STAT_LR 0x2d18

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_lr . unsigned , default = 0

DNR_RO_GBS_STAT_LL 0x2d19

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_ll . unsigned , default = 0

DNR_RO_GBS_STAT_RR 0x2d1a

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_rr . unsigned , default = 0

DNR_RO_GBS_STAT_DIF 0x2d1b

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_dif . unsigned , default = 0

DNR_RO_GBS_STAT_CNT 0x2d1c

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_cnt . unsigned , default = 0

DNR_RO_HBOF_STAT_CNT_0 0x2d1d

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_hbof_stat_cnt0 . unsigned , default = 0

DNR_RO_HBOF_STAT_CNT_31 0x2d3c

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_hbof_stat_cnt31 . unsigned , default = 0

DNR_RO_VBOF_STAT_CNT_0 0x2d3d

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_vbof_stat_cnt0 . unsigned , default = 0

DNR_RO_VBOF_STAT_CNT_31 0x2d5c

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_vbof_stat_cnt31 . unsigned , default = 0

DNR_CTRL 0x2d00

Bit(s)	R/W	Default	Description
31:17	R/W	0	reserved
16	R/W	0	reg_dnr_en
15	R/W	0	reg_dnr_db_vdbstep , vdb step, 0: 4, 1: 8 . unsigned , default = 1
14	R/W	0	reg_dnr_db_vdbprten , vdb protectoin enable . unsigned , default = 1
13	R/W	0	reg_dnr_gbs_difen , enable dif (between LR and LL/RR) condition for gbs stat.. unsigned , default = 0
12	R/W	0	reg_dnr_luma_en , enable ycbcr2luma module . unsigned , default = 1
11:10	R/W	0	reg_dnr_db_mod , deblocking mode, 0: disable, 1: horizontal deblocking, 2: vertical deblocking, 3: horizontal & vertical deblocking. unsigned , default = 3
9	R/W	0	reg_dnr_db_chrmen , enable chroma deblocking . unsigned , default = 1
8	R/W	0	reg_dnr_hvdif_mod , 0: calc. difs by original Y, 1: by new luma. unsigned , default = 1
7	R/W	0	reserved
6: 4	R/W	0	reg_dnr_demo_lften , b0: Y b1:U b2:V . unsigned , default = 7
3	R/W	0	reserved
2: 0	R/W	0	reg_dnr_demo_rgten , b0: Y b1:U b2:V . unsigned , default = 7

DNR_HVSIZE 0x2d01

Bit(s)	R/W	Default	Description
31:29	R/W	0	reserved
28:16	R/W	0	reg_dnr_hsize , hsize . unsigned , default = 0
15:13	R/W	0	reserved
12: 0	R/W	0	reg_dnr_vsize , vsize . unsigned , default = 0

DNR_DBLK_BLANK_NUM 0x2d02

Bit(s)	R/W	Default	Description
31:16	R/W	0	reserved
15: 8	R/W	0	reg_dblk_hblank_num , deblock hor blank num . unsigned , default = 16
7: 0	R/W	0	reg_dblk_vblank_num , deblock ver blank num . unsigned , default = 45

DNR_BLK_OFFST 0x2d03

Bit(s)	R/W	Default	Description
31: 7	R/W	0	reserved
6: 4	R/W	0	reg_dnr_hbofst , horizontal block offset may provide by software calc.. unsigned , default = 0
3	R/W	0	reserved
2: 0	R/W	0	reg_dnr_vbofst , vertical block offset may provide by software calc.. unsigned , default = 0

DNR_GBS 0x2d04

Bit(s)	R/W	Default	Description
31: 2	R/W	0	reserved
1: 0	R/W	0	reg_dnr_gbs , global block strength may update by software calc.. unsigned , default = 0

DNR_HBOFFST_STAT 0x2d05

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_hbof_difthd , dif threshold (>=) between LR and LL/RR. unsigned , default = 2
23:16	R/W	0	reg_dnr_hbof_edgethd , edge threshold (<=) for LR . unsigned , default = 32
15: 8	R/W	0	reg_dnr_hbof_flatthd , flat threshold (>=) for LR . unsigned , default = 0
7	R/W	0	reserved
6: 4	R/W	0	reg_dnr_hbof_delta , delta for weighted bin accumulator. unsigned , default = 1
3	R/W	0	reserved
2: 0	R/W	0	reg_dnr_hbof_statmod , statistic mode for horizontal block offset, 0: count flags for 8-bin, 1: count LRs for 8-bin, 2: count difs for 8-bin, 3: count weighted flags for 8-bin, 4: count flags for first 32-bin, 5: count LRs for first 32-bin, 6 or 7: count difs for first 32-bin. unsigned , default = 2

DNR_VBOFFST_STAT 0x2d06

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_vbof_difthd , dif threshold (>=) between Up and Dw. unsigned , default = 1
23:16	R/W	0	reg_dnr_vbof_edgethd , edge threshold (<=) for Up/Dw. unsigned , default = 16
15: 8	R/W	0	reg_dnr_vbof_flatthd , flat threshold (>=) for Up/Dw. unsigned , default = 0
7	R/W	0	reserved
6: 4	R/W	0	reg_dnr_vbof_delta , delta for weighted bin accumulator. unsigned , default = 1

3	R/W	0	reserved
2: 0	R/W	0	reg_dnr_vbof_statmod , statistic mode for vertical block offset, 0: count flags for 8-bin, 1: count Ups for 8-bin, 2: count difs for 8-bin, 3: count weighted flags for 8-bin, 4: count flags for first 32-bin, 5: count Ups for first 32-bin, 6 or 7: count difs for first 32-bin. unsigned , default = 2

DNR_GBS_STAT 0x2d07

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_gbs_edgethd , edge threshold (<=) for LR . unsigned , default = 32
23:16	R/W	0	reg_dnr_gbs_flatthd , flat threshold (>=) for LR . unsigned , default = 0
15: 8	R/W	0	reg_dnr_gbs_varthd , variation threshold (<=) for Lvar/Rvar. unsigned , default = 16
7: 0	R/W	0	reg_dnr_gbs_difthd , dif threshold (>=) between LR and LL/RR. unsigned , default = 2

DNR_STAT_X_START_END 0x2d08

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:16	R/W	0	reg_dnr_stat_xst . unsigned , default = 24
15:14	R/W	0	reserved
13: 0	R/W	0	reg_dnr_stat_xed . unsigned , default = HSIZE - 25

DNR_STAT_Y_START_END 0x2d09

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:16	R/W	0	reg_dnr_stat_yst . unsigned , default = 24
15:14	R/W	0	reserved
13: 0	R/W	0	reg_dnr_stat_yed . unsigned , default = VSIZE - 25

DNR_LUMA 0x2d0a

Bit(s)	R/W	Default	Description
31:27	R/W	0	reserved
26:24	R/W	0	reg_dnr_luma_sqrtshft , left shift for fast squart of chroma, [0, 4]. unsigned , default = 2
23:21	R/W	0	reserved
20:16	R/W	0	reg_dnr_luma_sqrtoffst , offset for fast squart of chroma. signed , default = 0
15	R/W	0	reserved

14:12	R/W	0	reg_dnr_luma_wcmod , theta related to warm/cool segment line, 0: 0, 1: 45, 2: 90, 3: 135, 4: 180, 5: 225, 6: 270, 7: 315. . unsigned , default = 3
11: 8	R/W	0	reg_dnr_luma_cshft , shift for calc. delta part, 0~8, . unsigned , default = 8
7: 6	R/W	0	reserved
5: 0	R/W	0	reg_dnr_luma_cgain , final gain for delta part, 32 normalized to "1". unsigned , default = 4

DNR_DB_YEDGE_THD 0x2d0b

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_db_yedgethd0 , edge threshold0 for luma . unsigned , default = 12
23:16	R/W	0	reg_dnr_db_yedgethd1 , edge threshold1 for luma . unsigned , default = 15
15: 8	R/W	0	reg_dnr_db_yedgethd2 , edge threshold2 for luma . unsigned , default = 18
7: 0	R/W	0	reg_dnr_db_yedgethd3 , edge threshold3 for luma . unsigned , default = 25

DNR_DB_CEDGE_THD 0x2d0c

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_db_cedgethd0 , edge threshold0 for chroma . unsigned , default = 12
23:16	R/W	0	reg_dnr_db_cedgethd1 , edge threshold1 for chroma . unsigned , default = 15
15: 8	R/W	0	reg_dnr_db_cedgethd2 , edge threshold2 for chroma . unsigned , default = 18
7: 0	R/W	0	reg_dnr_db_cedgethd3 , edge threshold3 for chroma . unsigned , default = 25

DNR_DB_HGAP 0x2d0d

Bit(s)	R/W	Default	Description
31:24	R/W	0	reserved
23:16	R/W	0	reg_dnr_db_hgapthd , horizontal gap thd (<=) for very sure blockiness . unsigned , default = 8
15: 8	R/W	0	reg_dnr_db_hgapdifthd , dif thd between hgap and lft/rgt hdifs. unsigned , default = 1
7: 1	R/W	0	reserved
0	R/W	0	reg_dnr_db_hgapmod , horizontal gap calc. mode, 0: just use current col x, 1: find max between (x-1, x, x+1) . unsigned , default = 0

DNR_DB_HBS 0x2d0e

Bit(s)	R/W	Default	Description
31: 6	R/W	0	reserved
5: 4	R/W	0	reg_dnr_db_hbsup , horizontal bs up value . unsigned , default = 1
3: 2	R/W	0	reg_dnr_db_hbsmax , max value of hbs for global control. unsigned , default = 3
1: 0	R/W	0	reg_dnr_db_hgbsthd , gbs thd (>=) for hbs calc. . unsigned , default = 1

DNR_DB_HACT 0x2d0f

Bit(s)	R/W	Default	Description
31:16	R/W	0	reserved
15: 8	R/W	0	reg_dnr_db_hactthd0 , thd0 of hact, for block classification. unsigned , default = 10
7: 0	R/W	0	reg_dnr_db_hactthd1 , thd1 of hact, for block classification. unsigned , default = 32

DNR_DB_YHDELTA_GAIN 0x2d10

Bit(s)	R/W	Default	Description
31:27	R/W	0	reserved
26:24	R/W	0	reg_dnr_db_yhdeltagain1 , (p1-q1) gain for Y's delta calc. when bs=1, normalized 8 as "1" . unsigned , default = 2
23	R/W	0	reserved
22:20	R/W	0	reg_dnr_db_yhdeltagain2 , (p1-q1) gain for Y's delta calc. when bs=2, normalized 8 as "1" . unsigned , default = 0
19	R/W	0	reserved
18:16	R/W	0	reg_dnr_db_yhdeltagain3 , (p1-q1) gain for Y's delta calc. when bs=3, normalized 8 as "1" . unsigned , default = 0
15	R/W	0	reserved
14: 8	R/W	0	reg_dnr_db_yhdeltaadjoffst , offset for adjust Y's hdelta (-64, 63). signed , default = 0
7: 6	R/W	0	reserved
5: 0	R/W	0	reg_dnr_db_yhdeltaadjgain , gain for adjust Y's hdelta, normalized 32 as "1" . unsigned , default = 32

DNR_DB_YHDELTA2_GAIN 0x2d11

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:24	R/W	0	reg_dnr_db_yhdelta2gain2 , gain for bs=2's adjust Y's hdelta2, normalized 64 as "1" . unsigned , default = 8
23:21	R/W	0	reserved
20:16	R/W	0	reg_dnr_db_yhdelta2offst2 , offset for bs=2's adjust Y's hdelta2 (-16, 15). signed , default = 0
15:14	R/W	0	reserved
13: 8	R/W	0	reg_dnr_db_yhdelta2gain3 , gain for bs=3's adjust Y's hdelta2, normalized 64 as "1" . unsigned , default = 4
7: 5	R/W	0	reserved
4: 0	R/W	0	reg_dnr_db_yhdelta2offst3 , offset for bs=3's adjust Y's hdelta2 (-16, 15). signed , default = 0

DNR_DB_CHDELTA_GAIN 0x2d12

Bit(s)	R/W	Default	Description
31:27	R/W	0	reserved
26:24	R/W	0	reg_dnr_db_chdeltagain1 , (p1-q1) gain for UV's delta calc. when bs=1, normalized 8 as "1". unsigned , default = 2
23	R/W	0	reserved
22:20	R/W	0	reg_dnr_db_chdeltagain2 , (p1-q1) gain for UV's delta calc. when bs=2, normalized 8 as "1". unsigned , default = 0
19	R/W	0	reserved
18:16	R/W	0	reg_dnr_db_chdeltagain3 , (p1-q1) gain for UV's delta calc. when bs=3, normalized 8 as "1". unsigned , default = 0
15	R/W	0	reserved
14: 8	R/W	0	reg_dnr_db_chdeltaadjoffst , offset for adjust UV's hdelta (-64, 63). signed , default = 0
7: 6	R/W	0	reserved
5: 0	R/W	0	reg_dnr_db_chdeltaadjgain , gain for adjust UV's hdelta, normalized 32 as "1". unsigned , default = 32

DNR_DB_CHDELTA2_GAIN 0x2d13

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved

29:24	R/W	0	reg_dnr_db_chdelta2gain2 , gain for bs=2's adjust UV's hdelta2, normalized 64 as "1" . unsigned , default = 8
23:21	R/W	0	reserved
20:16	R/W	0	reg_dnr_db_chdelta2offst2 , offset for bs=2's adjust UV's hdelta2 (-16, 15). signed , default = 0
15:14	R/W	0	reserved
13: 8	R/W	0	reg_dnr_db_chdelta2gain3 , gain for bs=2's adjust UV's hdelta2, normalized 64 as "1" . unsigned , default = 4
7: 5	R/W	0	reserved
4: 0	R/W	0	reg_dnr_db_chdelta2offst3 , offset for bs=2's adjust UV's hdelta2 (-16, 15). signed , default = 0

DNR_DB_YC_VEDGE_THD 0x2d14

Bit(s)	R/W	Default	Description
31:16	R/W	0	reserved
15: 8	R/W	0	reg_dnr_db_yvedgethd , special Y's edge thd for vdb. unsigned , default = 12
7: 0	R/W	0	reg_dnr_db_cvedgethd , special UV's edge thd for vdb. unsigned , default = 12

DNR_DB_VBS_MISC 0x2d15

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_dnr_db_vgapthd , vertical gap thd (<=) for very sure blockiness . unsigned , default = 8
23:16	R/W	0	reg_dnr_db_vactthd , thd of vact, for block classification . unsigned , default = 10
15: 8	R/W	0	reg_dnr_db_vgapdifthd , dif thd between vgap and vact. unsigned , default = 4
7: 4	R/W	0	reserved
3: 2	R/W	0	reg_dnr_db_vbsmax , max value of vbs for global control. unsigned , default = 2
1: 0	R/W	0	reg_dnr_db_vgbsthd , gbs thd (>=) for vbs calc. . unsigned , default = 1

DNR_DB_YVDELTA_GAIN 0x2d16

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:24	R/W	0	reg_dnr_db_yvdeltaadjgain , gain for adjust Y's vdelta, normalized 32 as "1". unsigned , default = 32
23	R/W	0	reserved
22:16	R/W	0	reg_dnr_db_yvdeltaadjoffst , offset for adjust Y's vdelta (-64, 63). signed , default = 0
15:14	R/W	0	reserved

13: 8	R/W	0	reg_dnr_db_yvdelta2gain , gain for adjust Y's vdelta2, normalized 64 as "1". unsigned , default = 8
7: 5	R/W	0	reserved
4: 0	R/W	0	reg_dnr_db_yvdelta2offst , offset for adjust Y's vdelta2 (-16, 15). signed , default = 0

DNR_DB_CVDELTA_GAIN 0x2d17

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29:24	R/W	0	reg_dnr_db_cvdeltaadjgain , gain for adjust UV's vdelta, normalized 32 as "1". unsigned , default = 32
23	R/W	0	reserved
22:16	R/W	0	reg_dnr_db_cvdeltaadjoffst , offset for adjust UV's vdelta (-64, 63). signed , default = 0
15:14	R/W	0	reserved
13: 8	R/W	0	reg_dnr_db_cvdelta2gain , gain for adjust UV's vdelta2, normalized 64 as "1". unsigned , default = 8
7: 5	R/W	0	reserved
4: 0	R/W	0	reg_dnr_db_cvdelta2offst , offset for adjust UV's vdelta2 (-16, 15). signed , default = 0

DNR_RO_GBS_STAT_LR 0x2d18

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_lr . unsigned , default = 0

DNR_RO_GBS_STAT_LL 0x2d19

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_ll . unsigned , default = 0

DNR_RO_GBS_STAT_RR 0x2d1a

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_rr . unsigned , default = 0

DNR_RO_GBS_STAT_DIF 0x2d1b

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_dif . unsigned , default = 0

DNR_RO_GBS_STAT_CNT 0x2d1c

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_gbs_stat_cnt . unsigned , default = 0

DNR_RO_HBOF_STAT_CNT_0 0x2d1d

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_hbof_stat_cnt0 . unsigned , default = 0

DNR_RO_HBOF_STAT_CNT_31 0x2d3c

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_hbof_stat_cnt31 . unsigned , default = 0

DNR_RO_VBOF_STAT_CNT_0 0x2d3d

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_vbof_stat_cnt0 . unsigned , default = 0

DNR_RO_VBOF_STAT_CNT_31 0x2d5c

Bit(s)	R/W	Default	Description
31: 0	R	0	ro_vbof_stat_cnt31 . unsigned , default = 0

DNR_DM_CTRL 0x2d60

Bit(s)	R/W	Default	Description
31:13	R/W	0	reserved
12	R/W	1	reg_dnr_fedgeflg_en , 1 to enable edge flag calculation for each frame
11	R/W	1	reg_dnr_fedgeflg_cl , 1 to clear the edge flag to 0 for each frame
10	R/W	0	reg_dnr_fedgeflg_df , user defined edge flag when reg_dnr_fedgeflg_en = 0
9	R/W	0	reg_dnr_dm_en , 1 to enable de-mosquito unit
8	R/W	1	reg_dnr_dm_chrmen , 1 to enable chrome processing for de-mosquito
7: 6	R/W	3	reg_dnr_dm_level , de-mosquito level
5: 4	R/W	1	reg_dnr_dm_leveldw0 , level down when gbs is small
3: 2	R/W	1	reg_dnr_dm_leveldw1 , level down for flat blocks
1: 0	R/W	0	reg_dnr_dm_gbsthd , small/large threshold for gbs

DNR_DM_NR_BLND 0x2d61

Bit(s)	R/W	Default	Description
31:25	R/W	0	reserved
24	R/W	0	reg_dnr_dm_defalpen , 1 to enable user defined alpha for DM/NR blend
23:16	R/W	0	reg_dnr_dm_defalp , user defined alpha for DM/NR blend
15:14	R/W	0	reserved
13: 8	R/W	32	reg_dnr_dm_alpgain , gain for DM/NR alpha, normalized 32 as 1
7: 0	R/W	0	reg_dnr_dm_alpoffset , offset for DM/NR alpha

DNR_DM_RNG_THD 0x2d62

Bit(s)	R/W	Default	Description
31:24	R/W	0	reserved
23:16	R/W	2	reg_dnr_dm_rgnminthd
15: 8	R/W	64	reg_dnr_dm_rgnmaxthd
7: 0	R/W	4	reg_dnr_dm_rgndifthd

DNR_DM_RNG_GAIN_OFST 0x2d63

Bit(s)	R/W	Default	Description
31:14	R/W	0	reserved
13: 8	R/W	16	reg_dnr_dm_rgngain , normalized 16 as 1
7:6	R/W	0	reserved
5: 0	R/W	0	reg_dnr_dm_rgnofst

DNR_DM_DIR_MISC 0x2d64

Bit(s)	R/W	Default	Description
31:30	R/W	0	reserved
29	R/W	1	reg_dnr_dm_diralpen
28:24	R/W	0	reg_dnr_dm_diralpgain
23:22	R/W	0	reserved
21:16	R/W	0	reg_dnr_dm_diralpofst
15:13	R/W	0	reserved
12: 8	R/W	0	reg_dnr_dm_diralpmin
7:5	R/W	0	reserved

4: 0	R/W	31	reg_dnr_dm_diralpmax

DNR_DM_COR_DIF 0x2d65

Bit(s)	R/W	Default	Description
31:4	R/W	0	reserved
3:1	R/W	3	reg_dnr_dm_cordifshft
0	R/W	1	reg_dnr_dm_cordifmod , 0: use max dir dif as cordif, 1: use max3x3-min3x3 as cordif

DNR_DM_FLT_THD 0x2d66

Bit(s)	R/W	Default	Description
31:24	R/W	4	reg_dnr_dm_fltthd00 , block flat threshold0 for block average difference when gbs is small
23:16	R/W	6	reg_dnr_dm_fltthd01 , block flat threshold1 for block average difference when gbs is small
15: 8	R/W	9	reg_dnr_dm_fltthd10 , block flat threshold0 for block average difference when gbs is larger
7: 0	R/W	12	reg_dnr_dm_fltthd11 , block flat threshold1 for block average difference when gbs is larger

DNR_DM_VAR_THD 0x2d67

Bit(s)	R/W	Default	Description
31:24	R/W	2	reg_dnr_dm_varthd00 , block variance threshold0 (>=) when gbs is small
23:16	R/W	15	reg_dnr_dm_varthd01 , block variance threshold1 (<=) when gbs is small
15: 8	R/W	3	reg_dnr_dm_varthd10 , block variance threshold0 (>=) when gbs is larger
7: 0	R/W	24	reg_dnr_dm_varthd11 , block variance threshold1 (<=) when gbs is larger

DNR_DM_EDGE_DIF_THD 0x2d68

Bit(s)	R/W	Default	Description
31:24	R/W	32	reg_dnr_dm_edgethd0 , block edge threshold (<=) when gbs is small
23:16	R/W	48	reg_dnr_dm_edgethd1 , block edge threshold (<=) when gbs is larger
15: 8	R/W	48	reg_dnr_dm_difthd0 , block dif threshold (<=) when gbs is small
7: 0	R/W	64	reg_dnr_dm_difthd1 , block dif threshold (<=) when gbs is larger

DNR_DM_AVG_THD 0x2d69

Bit(s)	R/W	Default	Description
31:16	R/W		reserved
15: 8	R/W	160	reg_dnr_dm_avgthd0 , block average threshold (>=) when gbs is small
7: 0	R/W	128	reg_dnr_dm_avgthd1 , block average threshold (<=) when gbs is larger

DNR_DM_AVG_VAR_DIF_THD 0x2d6a

Bit(s)	R/W	Default	Description
31:16	R/W		reserved
15: 8	R/W	12	reg_dnr_dm_avgdifthd , block average dif threshold(<) between cur and up block for flat block
7: 0	R/W	1	reg_dnr_dm_vardifthd , block variance dif threshold(>=) between cur and up block

DNR_DM_EDGE_DIF_THD2 0x2d6b

Bit(s)	R/W	Default	Description
31:24	R/W		reserved
23:16	R/W	24	reg_dnr_dm_varthd2 , block variance threshold (>=) for edge block detect
15: 8	R/W	40	reg_dnr_dm_edgethd2 , block edge threshold (>=)
7: 0	R/W	80	reg_dnr_dm_difthd2 , block dif threshold (>=)

DNR_DM_DIF_FLT_MISC 0x2d6c

Bit(s)	R/W	Default	Description
31:28	R/W	0	reg_dnr_dm_ldifoob, pre-defined large dif when pixel out of block
27:24	R/W	0	reg_dnr_dm_bdifoob, pre-defined block dif when pixel out of block
23:16	R/W	200	reg_dnr_dm_fltalp, pre-defined alpha for dm and nr blending when block is flat with mos
15:12	R/W		reserved
11:8	R/W	12	reg_dnr_dm_fltminbdif, pre-defined min block dif for dm filter when block is flat with mos
7	R/W		reserved
6:2	R/W	16	reg_dnr_dm_difnormgain, gain for pixel dif normalization for dm filter.
1	R/W	1	reg_dnr_dm_difnormen , enable pixel dif normalization for dm filter
0	R/W	0	reg_dnr_dm_difupden , enable block dif update using max of left,cur,right difs

DNR_DM_SDIF_LUT0_2 0x2d6d

Bit(s)	R/W	Default	Description
31:21	R/W		reserved
20:16	R/W	16	reg_dnr_dm_sdiflut0, normally 0-16
15:13	R/W		reserved
12:8	R/W	14	reg_dnr_dm_sdiflut1
7:5	R/W		reserved
4:0	R/W	13	reg_dnr_dm_sdiflut2

DNR_DM_SDIF_LUT3_5 0x2d6e

Bit(s)	R/W	Default	Description
31:21	R/W		reserved
20:16	R/W	10	reg_dnr_dm_sdiflut3
15:13	R/W		reserved
12:8	R/W	7	reg_dnr_dm_sdiflut4
7:5	R/W		reserved
4:0	R/W	5	reg_dnr_dm_sdiflut5

DNR_DM_SDIF_LUT6_8 0x2d6f

Bit(s)	R/W	Default	Description
31:21	R/W		reserved
20:16	R/W	3	reg_dnr_dm_sdiflut6
15:13	R/W		reserved
12:8	R/W	1	reg_dnr_dm_sdiflut7
7:5	R/W		reserved
4:0	R/W	0	reg_dnr_dm_sdiflut8

DNR_DM_LDIF_LUT0_2 0x2d70

Bit(s)	R/W	Default	Description
31:21	R/W		reserved
20:16	R/W	0	reg_dnr_dm_ldiflut0
15:13	R/W		reserved
12:8	R/W	4	reg_dnr_dm_ldiflut1
7:5	R/W		reserved
4:0	R/W	12	reg_dnr_dm_ldiflut2

DNR_DM_LDIF_LUT3_5 0x2d71

Bit(s)	R/W	Default	Description
31:21	R/W		reserved
20:16	R/W	14	reg_dnr_dm_ldiflut3
15:13	R/W		reserved
12:8	R/W	15	reg_dnr_dm_ldiflut4
7:5	R/W		reserved
4:0	R/W	16	reg_dnr_dm_ldiflut5

DNR_DM_LDIF_LUT6_8 0x2d72

Bit(s)	R/W	Default	Description
31:21	R/W		reserved
20:16	R/W	16	reg_dnr_dm_ldiflut6
15:13	R/W		reserved
12:8	R/W	16	reg_dnr_dm_ldiflut7
7:5	R/W		reserved
4:0	R/W	16	reg_dnr_dm_ldiflut8

DNR_DM_DIF2NORM_LUT0_2 0x2d73

Bit(s)	R/W	Default	Description
31:21	R/W		reserved
20:16	R/W	16	reg_dnr_dm_dif2normlut0
15:13	R/W		reserved
12:8	R/W	5	reg_dnr_dm_dif2normlut1
7:5	R/W		reserved
4:0	R/W	3	reg_dnr_dm_dif2normlut2

DNR_DM_DIF2NORM_LUT3_5 0x2d74

Bit(s)	R/W	Default	Description
31:21	R/W		reserved
20:16	R/W	2	reg_dnr_dm_dif2normlut3
15:13	R/W		reserved
12:8	R/W	2	reg_dnr_dm_dif2normlut4
7:5	R/W		reserved
4:0	R/W	1	reg_dnr_dm_dif2normlut5

DNR_DM_DIF2NORM_LUT6_8 0x2d75

Bit(s)	R/W	Default	Description
31:21	R/W		reserved
20:16	R/W	1	reg_dnr_dm_dif2normlut6
15:13	R/W		reserved
12:8	R/W	1	reg_dnr_dm_dif2normlut7
7:5	R/W		reserved
4:0	R/W	1	reg_dnr_dm_dif2normlut8

DNR_DM_GMS_THD 0x2d76

Bit(s)	R/W	Default	Description
31:16	R/W		reserved
15:8	R/W	0	reg_gms_stat_thd0
7:0	R/W	128	reg_gms_stat_thd1

DNR_RO_DM_GMS_STST_CNT 0x2d77

Bit(s)	R/W	Default	Description
31:0	RO	0	ro_dm_gms_stat_cnt

DNR_RO_DM_GMS_STST_MS 0x2d78

Bit(s)	R/W	Default	Description
31:0	RO	0	ro_dm_gms_stat_ms

DECOMB_DET_VERT_CON0 0x2d80

Bit(s)	R/W	Default	Description
31:24	R/W	60	reg_di_dcmb_det_vcon_thd0 : default = 60 // u8
23:16	R/W	80	reg_di_dcmb_det_vcon_thd1 : default = 80 // u8
15: 8	R/W	63	reg_di_dcmb_det_valp_lmt0 : default = 63 // u8
7: 0	R/W	4	reg_di_dcmb_det_valp_lmt1 : default = 4 // u8

DECOMB_DET_VERT_CON1 0x2d81

Bit(s)	R/W	Default	Description
23:16	R/W	0	reg_di_dcmb_det_valp_lmt2 : default = 0 // u8
15: 8	R/W	32	reg_di_dcmb_det_vrate0 : default = 32 // u8
7: 0	R/W	4	reg_di_dcmb_det_vrate1 : default = 4 // u8

DECOMB_DET_EDGE_CON0 0x2d82

Bit(s)	R/W	Default	Description
31:24	R/W	60	reg_di_dcmb_det_econ_thd0 : default = 60 // u8
23:16	R/W	80	reg_di_dcmb_det_econ_thd1 : default = 80 // u8
15: 8	R/W	63	reg_di_dcmb_det_ealp_lmt0 : default = 63 // u8
7: 0	R/W	4	reg_di_dcmb_det_ealp_lmt1 : default = 4 // u8

DECOMB_DET_EDGE_CON1 0x2d83

Bit(s)	R/W	Default	Description
23:16	R/W	0	reg_di_dcmb_det_ealp_lmt2 : default = 0 // u8
15: 8	R/W	32	reg_di_dcmb_det_erate0 : default = 32 // u8
7: 0	R/W	4	reg_di_dcmb_det_erate1 : default = 4 // u8

DECOMB_PARA 0x2d84

Bit(s)	R/W	Default	Description

29:28	R/W	1	reg_di_dcmb_cmb_lpf : default = 1 // u2, 0:no lpf, 1:[1 2 1], 2,3: [1 2 2 2 1]
27:26	R/W	0	reg_di_dcmb_vedge_chk : default = 0 // u2, vertical edge check, 0: no check, 1: vrt!=0, 2: vrt==3
25:24	R/W	0	reg_di_dcmb_nedge_chk : default = 0 // u2, no idea edge check, 0, no check, 1, check
23:20	R/W	0	reg_di_dcmb_edge_min : default = 0 // u4, min edge for edge cmb
19:16	R/W	15	reg_di_dcmb_edge_max : default = 15 // u4, min edge for edge cmb
15:8	R/W	255	reg_di_dcmb_bld_alp : default = 255 // u8, user defined alpha for di & decmb blend
7:0	R/W	40	reg_di_dcmb_bld_alp_beta : default = 40 // u8, beta for mtn & cmb blend, for bld alpha calc.

DECOMB_BLND_CON0 0x2d85

Bit(s)	R/W	Default	Description
31:24	R/W	100	reg_di_dcmb_bld_con_thd0 : default = 100 // u8
23:16	R/W	120	reg_di_dcmb_bld_con_thd1 : default = 120 // u8
15: 8	R/W	0	reg_di_dcmb_bld_alp_lmt0 : default = 0 // u8
7: 0	R/W	128	reg_di_dcmb_bld_alp_lmt1 : default = 128 // u8

DECOMB_BLND_CON1 0x2d86

Bit(s)	R/W	Default	Description
23:16	R/W	255	reg_di_dcmb_bld_alp_lmt2 : default = 255 // u8
15: 8	R/W	32	reg_di_dcmb_bld_rate0 : default = 32 // u8
7: 0	R/W	32	reg_di_dcmb_bld_rate1 : default = 32 // u8

DECOMB_YC_THRD 0x2d87

Bit(s)	R/W	Default	Description
15: 8	R/W	2	reg_di_dcmb_ythd : default = 2 // u8, default = 2
7: 0	R/W	2	reg_di_dcmb_cthd : default = 2 // u8, default = 2

DECOMB_MTN_GAIN_OFST 0x2d88

Bit(s)	R/W	Default	Description
21:16	R/W	16	reg_di_dcmb_mtn_alp_gain : default = 16 // u6, 16 is normalized to '1'
8:0	R/W	0	reg_di_dcmb_mtn_alp_ofst : default = 0 // s9, [-256, 255]

DECOMB_CMB_SEL_GAIN_OFST 0x2d89

Bit(s)	R/W	Default	Description
21:16	R/W	48	reg_di_dcmb_cmb_sel_gain : default = 48 // u6, 16 is normalized to '1'

8:0	R/W	0	reg_di_dcmb_cmb_sel_ofst : default = 0 // s9, [-256, 255]

DECOMB_WIND00 0x2d8a

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_di_dcmb_wnd00 : default = 0 // u13, x0 for window 0, software control
12:0	R/W	719	reg_di_dcmb_wnd01 : default = 719 // u13, x1 for window 0, HSIZE-1, software control

DECOMB_WIND01 0x2d8b

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_di_dcmb_wnd02 : default = 0 // u13, y0 for window 0, software control
12:0	R/W	39	reg_di_dcmb_wnd03 : default = 39 // u13, y1 for window 0, software control

DECOMB_WIND10 0x2d8c

Bit(s)	R/W	Default	Description
28:16	R/W	0	reg_di_dcmb_wnd10 : default = 0 // u13, x0 for window 1, software control
12:0	R/W	719	reg_di_dcmb_wnd11 : default = 719 // u13, x1 for window 1, HSIZE-1, software control

DECOMB_WIND11 0x2d8d

Bit(s)	R/W	Default	Description
28:16	R/W	40	reg_di_dcmb_wnd12 : default = 40 // u13, y0 for window 1, software control
12:0	R/W	239	reg_di_dcmb_wnd13 : default = 239 // u13, y1 for window 1, VSIZE-1-40, software control

DECOMB_MODE 0x2d8e

Bit(s)	R/W	Default	Description
15	R/W	1	reg_di_dcmb_is_cmb_bef : default = 1 // u1, 1: decide is_cmb before cmbing refine, 0: decide is_cmb after cmbing refine
14	R/W	1	reg_di_dcmb_en0 : default = 1 // u1, enable decmobing for wind0
13	R/W	1	reg_di_dcmb_en1 : default = 1 // u1, enable decmobing for wind1
12	R/W	1	reg_di_dcmb_en2 : default = 1 // u1, enable decmobing for wind2
11:10	R/W	2	reg_di_dcmb_lpf_mod0 : default = 2 // u2, get combing free pixels of wind0 by: 0, vertical lpf, 1, edge lpf, 2,3, ei data
9:8	R/W	2	reg_di_dcmb_lpf_mod1 : default = 2 // u2, get combing free pixels of wind1 by: 0, vertical lpf, 1, edge lpf, 2,3, ei data
7:6	R/W	0	reg_di_dcmb_lpf_mod2 : default = 0 // u2, get combing free pixels of wind2 by: 0, vertical lpf, 1, edge lpf, 2,3, ei data
5	R/W	1	reg_di_dcmb_cmb_sel0 : default = 1 // u1, wind0 decmb based on: 0, vert cmb, 1, edge cmb

4	R/W	1	reg_di_dcmb_cmb_sel1 : default = 1 // u1, wind1 decmb based on: 0, vert cmb, 1, edge cmb
3	R/W	0	reg_di_dcmb_cmb_sel2 : default = 0 // u1, wind2 decmb based on: 0, vert cmb, 1, edge cmb
2	R/W	1	reg_di_dcmb_alp_mod0 : default = 1 // u1, wind0 decmb alpha based on: 0, user-defined, 1, motion adaptive
1	R/W	1	reg_di_dcmb_alp_mod1 : default = 1 // u1, wind1 decmb alpha based on: 0, user-defined, 1, motion adaptive
0	R/W	1	reg_di_dcmb_alp_mod2 : default = 1 // u1, wind2 decmb alpha based on: 0, user-defined, 1, motion adaptive

DECOMB_FRM_SIZE 0x2d8f

Bit(s)	R/W	Default	Description
28:16	R/W	1920	hsize_in : default = 1920 // u13, pic horz size in unit: pixel
12:0	R/W	1080	vsize_in : default = 1080 // u13, pic vert size in unit: pixel

DECOMB_HV_BLANK 0x2d90

Bit(s)	R/W	Default	Description
15:8	R/W	20	hblank_num : default = 20 // u8, hor blank time
7:0	R/W	50	vblank_num : default = 50 // u8, ver blank time

NR2_POLAR3_MODE 0x2d98

Bit(s)	R/W	Default	Description
19:18	R/W	3	reg_polar3_f02lpf_mod_0 : default = 3 //u2x2: low pass filter mode for field 0 and field2 before polar3 detection; 0 for no lpf, 1: [1 2 1]/4 vert lpf; 2: [1 2 1; 2 4 2; 1 2 1]/16 2d lpf, p1 no hlpf; 2: [1 2 1; 2 4 2; 1 2 1]/16 2d lpf, p1 [1 2 1]/4 hlpf
17:16	R/W	3	reg_polar3_f02lpf_mod_1 : default = 3 //u2x2: low pass filter mode for field 0 and field2 before polar3 detection; 0 for no lpf, 1: [1 2 1]/4 vert lpf; 2: [1 2 1; 2 4 2; 1 2 1]/16 2d lpf, p1 no hlpf; 2: [1 2 1; 2 4 2; 1 2 1]/16 2d lpf, p1 [1 2 1]/4 hlpf
15:8	R/W	5	reg_polar3_dif02_thrd_0 : default = 5 //u8x2: threshold of dif for polar3 detection except for 32 detection, only do polar3 detection on obvious motion, [0] for luma, 1[1] for chroma
7:0	R/W	5	reg_polar3_dif02_thrd_1 : default = 5 //u8x2: threshold of dif for polar3 detection except for 32 detection, only do polar3 detection on obvious motion, [0] for luma, 1[1] for chroma

NR2_POLAR3_THRD 0x2d99

Bit(s)	R/W	Default	Description
31:24	R/W	30	reg_polar3_txtf02_thrd_0 : default = 30 //u8x2: threshold to vertical f0f2 texture, if texture larger than this threshold, will not do the polar3 decision.
23:16	R/W	30	reg_polar3_txtf02_thrd_1 : default = 30 //u8x2: threshold to vertical f0f2 texture, if texture larger than this threshold, will not do the polar3 decision.

15:8	R/W	20	reg_polar3_txtf1_thrd_0 : default = 20 //u8x2: threshold to vertical f0f2 texture, if texture larger than this threshold, will not do the polar3 decision.
7:0	R/W	20	reg_polar3_txtf1_thrd_1 : default = 20 //u8x2: threshold to vertical f0f2 texture, if texture larger than this threshold, will not do the polar3 decision.

NR2_POLAR3_PARA0 0x2d9a

Bit(s)	R/W	Default	Description
31:28	R/W	6	reg_polar3_rate0_0 : default = 6 //u4x2: delt = rate*dif02/32, e.g. f2<f0, if f1 within((f0+f2)/2 - delt), ((f0+f2)/2 + delt), then polar3_smoothmv++;
27:24	R/W	6	reg_polar3_rate0_1 : default = 6 //u4x2: delt = rate*dif02/32, e.g. f2<f0, if f1 within((f0+f2)/2 - delt), ((f0+f2)/2 + delt), then polar3_smoothmv++;
23:20	R/W	8	reg_polar3_rate1_0 : default = 8 //u4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<((f0+f2)/2 - delt), then polar3_m1++; if f1>((f0+f2)/2 + delt), then polar3_p1++;
19:16	R/W	8	reg_polar3_rate1_1 : default = 8 //u4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<((f0+f2)/2 - delt), then polar3_m1++; if f1>((f0+f2)/2 + delt), then polar3_p1++;
15:12	R/W	2	reg_polar3_rate2_0 : default = 2 //u4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofset2), then polar3_m2++; if f1>((f0 + delt+ ofset2), then polar3_p2++;
11:8	R/W	2	reg_polar3_rate2_1 : default = 2 //u4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofset2), then polar3_m2++; if f1>((f0 + delt+ ofset2), then polar3_p2++;
7:4	R/W	1	reg_polar3_ofst1_0 : default = 1 //s4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofset2), then polar3_m2++; if f1>((f0 + delt+ ofset2), then polar3_p2++;
3:0	R/W	1	reg_polar3_ofst1_1 : default = 1 //s4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofset2), then polar3_m2++; if f1>((f0 + delt+ ofset2), then polar3_p2++;

NR2_POLAR3_PARA1 0x2d9b

Bit(s)	R/W	Default	Description
31:24	R/W	48	reg_polar3_rate3_0 : default = 48 //u8x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofset3) or f1>((f0 + delt+ofst3), then polar3_32++;
23:16	R/W	48	reg_polar3_rate3_1 : default = 48 //u8x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofset3) or f1>((f0 + delt+ofst3), then polar3_32++;
15:12	R/W	2	reg_polar3_ofst3_0 : default = 2 //s4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofst3) or f1>((f0 + delt+ofst3), then polar3_32++;
11:8	R/W	2	reg_polar3_ofst3_1 : default = 2 //s4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofst3) or f1>((f0 + delt+ofst3), then polar3_32++;
7:4	R/W	2	reg_polar3_ofst2_0 : default = 2 //s4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofset2), then polar3_m2++; if f1>((f0 + delt+ ofset2), then polar3_p2++;
3:0	R/W	2	reg_polar3_ofst2_1 : default = 2 //s4x2: delt = rate*dif02/32, e.g. f2<f0, if f1<(f2 - delt-ofset2), then polar3_m2++; if f1>((f0 + delt+ ofset2), then polar3_p2++;

NR2_POLAR3_CTRL 0x2d9c

Bit(s)	R/W	Default	Description

16	R.O	0	reg_polar3_ro_reset : default = 0 //u1: reset signal of the polar3 read only registers
15:8	R/W	10	reg_polar3_h_mute : default = 10 //u8: horizontally pixels to mute for left right sides for polar3 detection;
7:0	R/W	10	reg_polar3_v_mute : default = 10 //u8: horizontally pixels to mute for top and bottom sides for polar3 detection;

NR2_RO_POLAR3_NUMOFPIX 0x2d9d

Bit(s)	R/W	Default	Description
23:0	R.O	0	ro_polar3_numofpix : default = 0 //u24, number of pixels detected as polar3

NR2_RO_POLAR3_SMOOTHMV 0x2d9e

Bit(s)	R/W	Default	Description
23:0	R.O	0	ro_polar3_smoothmv : default = 0 //u24, number of pixels with smooth mv, F(t) is close between avg of f(t-1) and f(t+1);

NR2_RO_POLAR3_M1 0x2d9f

Bit(s)	R/W	Default	Description
23:0	R.O	0	ro_polar3_m1 : default = 0 //u24, number of pixels with F(t) is close to f(t-1) instead of f(t+1), but in between [f(t-1), f(t+1)];

NR2_RO_POLAR3_P1 0x2da0

Bit(s)	R/W	Default	Description
23:0	R.O	0	ro_polar3_p1 : default = 0 //u24, number of pixels with F(t) is close to f(t+1) instead of f(t-1), but in between [f(t-1), f(t+1)];

NR2_RO_POLAR3_M2 0x2da1

Bit(s)	R/W	Default	Description
23:0	R.O	0	ro_polar3_m2 : default = 0 //u24, number of pixels with F(t) is close to f(t-1) instead of f(t+1), but out side of (f(t-1), f(t+1));

NR2_RO_POLAR3_P2 0x2da2

Bit(s)	R/W	Default	Description
23:0	R.O	0	ro_polar3_p2 : default = 0 //u24, number of pixels with F(t) is close to f(t+1) instead of f(t-1), but out side of (f(t-1), f(t+1));

NR2_RO_POLAR3_32 0x2da3

Bit(s)	R/W	Default	Description
23:0	R.O	0	ro_polar3_32 : default = 0 //u24, number of pixels with F(t) far from [f(t-1),f(t+1)] and f(t-1) is close to f(t+1);

NR4_DRT_CTRL 0x2da4

Bit(s)	R/W	Default	Description
31:24	R/W	16	reg_nr4_ydrt_3line_ssd_gain : // unsigned , default = 16 gain to max ssd normalized 16 as '1'
23:16	R/W	16	reg_nr4_ydrt_5line_ssd_gain : // unsigned , default = 16 gain to max ssd normalized 16 as '1'
14:13	R/W	1	reg_nr4_drt_yhsad_mode : // unsigned , default = 1 mode for luma horiztonal sad calc., 0: no vertical lpf, 1: vertical [1 2 1], 2 or 3: vertical [1 2 2 2 1] if 5 lines
12:11	R/W	1	reg_nr4_drt_chsad_mode : // unsigned , default = 1 mode for chroma horiztonal sad calc., 0: no vertical lpf, 1: vertical [1 2 1], 2 or 3: vertical [1 2 2 2 1] if 5 lines
10	R/W	1	reg_nr4_drt_yhsad_hlpf : // unsigned , default = 1 hlpf for luma hsad of drt calculation, 0: no lpf, 1: with [1 2 1] hlpf
9	R/W	1	reg_nr4_drt_yvsad_hlpf : // unsigned , default = 1 hlpf for luma vsad of drt calculation, 0: no lpf, 1: with [1 2 1] hlpf
8	R/W	1	reg_nr4_drt_ydsad_hlpf : // unsigned , default = 1 hlpf for luma dsad of drt calculation, 0: no lpf, 1: with [1 2 1] hlpf
7	R/W	1	reg_nr4_drt_chsad_hlpf : // unsigned , default = 1 hlpf for chrome hsad of drt calculation, 0: no lpf, 1: with [1 2 1] hlpf
6	R/W	1	reg_nr4_drt_cvsad_hlpf : // unsigned , default = 1 hlpf for chroma vsad of drt calculation, 0: no lpf, 1: with [1 2 1] hlpf
5	R/W	1	reg_nr4_drt_cdsad_hlpf : // unsigned , default = 1 hlpf for chroma dsad of drt calculation, 0: no lpf, 1: with [1 2 1] hlpf
4	R/W	1	reg_nr4_ydrt_dif_mode : // unsigned , default = 1 0:y_dif, 1: y_dif + (u_dif + v_dif)/2
3: 2	R/W	2	reg_nr4_cdrt_dif_mode : // unsigned , default = 2 0:(u_dif + v_dif), 1: y_dif/4 + (u_dif + v_dif)*3/4, 2:y_dif/2 + (u_dif + v_dif)/2, 3: y_dif (not recommended)
1:0	R/W		reserved

NR4_DRT_YSAD_GAIN 0x2da5

Bit(s)	R/W	Default	Description
31:24	R/W	16	reg_nr4_ysad_hrz_gain : // unsigned , default = 16 gain for horizontal sad, 16 normalized to "1"
23:16	R/W	20	reg_nr4_ysad_diag_gain : // unsigned , default = 20 gain for diagonal sad, 16 normalized to "1"
15: 8	R/W	16	reg_nr4_ysad_vrt_gain : // unsigned , default = 16 gain for vertical sad, 16 normalized to "1"
5: 0	R/W	6	reg_nr4_drt_ysad_core_rate : // unsigned , default = 6 rate of coring for sad(theta) - sad(theta+pi/2)*rate/64

NR4_DRT_CSAD_GAIN 0x2da6

Bit(s)	R/W	Default	Description

31:24	R/W	16	reg_nr4_csad_hrz_gain : // unsigned , default = 16 gain for horizontal sad, 16 normalized to "1"
23:16	R/W	20	reg_nr4_csad_diag_gain : // unsigned , default = 20 gain for diagonal sad, 16 normalized to "1"
15: 8	R/W	16	reg_nr4_csad_vrt_gain : // unsigned , default = 16 gain for vertical sad, 16 normalized to "1"
5: 0	R/W	6	reg_nr4_drt_csad_core_rate : // unsigned , default = 6 rate of coring for sad(theta) - sad(theta+pi/2)*rate/64

NR4_DRT_SAD_ALP_CORE 0x2da7

Bit(s)	R/W	Default	Description
23:20	R/W	0	reg_nr4_ydrt_alp_core_rate : // unsigned , default = 0 luma ratio to min_err, alpha = (min_err - (max_err - min_err)*rate + ofst)/max_err * 64; dft = 0/32
19:16	R/W	0	reg_nr4_cdrt_alp_core_rate : // unsigned , default = 0 chroma ratio to min_err, alpha = (min_err - (max_err - min_err)*rate + ofst)/max_err * 64; dft = 0/32
13: 8	R/W	10	reg_nr4_ydrt_alp_core_ofst : // unsigned , default = 10 luma offset to min_err, alpha = (min_err - (max_err - min_err)*rate + ofst)/max_err * 64; dft = 10
5: 0	R/W	10	reg_nr4_cdrt_alp_core_ofst : // unsigned , default = 10 chroma offset to min_err, alpha = (min_err - (max_err - min_err)*rate + ofst)/max_err * 64; dft = 10

NR4_DRT_ALP_MINMAX 0x2da8

Bit(s)	R/W	Default	Description
29:24	R/W	0	reg_nr4_ydrt_alp_min : // unsigned , default = 0 luma min value of alpha, dft = 0
21:16	R/W	63	reg_nr4_ydrt_alp_max : // unsigned , default = 63 luma max value of alpha, dft = 63
13: 8	R/W	0	reg_nr4_cdrt_alp_min : // unsigned , default = 0 chroma min value of alpha, dft = 0
5: 0	R/W	63	reg_nr4_cdrt_alp_max : // unsigned , default = 63 chroma max value of alpha, dft = 63

NR4_SNR_CTRL_REG 0x2da9

Bit(s)	R/W	Default	Description
12	R/W	1	reg_nr4_bet2_sel : // unsigned , default = 1
11: 9	R/W	0	reg_nr4_snr2_sel_mode : // unsigned , default = 0 0: no filter, 1: adpgau, adp_drt_lpf blend; 2: adpgau, drt4_lpf blend; 3: adp_drt_lpf method, 4: drt4_lpf method, 5: adp_drt_ //original image blend, 6: drt4_lpf, original image blend, 7: adpgau method; dft=1
8	R/W	1	reg_nr4_snr2_gaulpf_mode : // unsigned , default = 1 0: 3*5 or 5*5 gaussian lpf; 1: 3*3 (window size) gaussian lpf; dft=1
7: 6	R/W	3	reg_nr4_snr2_alpha0_sad_mode : // unsigned , default = 3 0: max_sad*max_ssd; 1: max_sad*max_sad; 2: adp_max_sad*max_ssd; 3: adp_max_sad*adp_max_sad dft=3
5: 4	R/W	2	reg_nr4_snr2_alpha1_sad_mode : // unsigned , default = 2 0: max_sad; 1: cross_max_sad; 2 or 3: adp_sad dft=2

1: 0	R/W	3	reg_nr4_snr2_adp_drtlpf_mode : // unsigned , default = 3 0: adp_drtlpf [2 1 1]/4, 1: adp_drtlpf [4 2 1 1]/8; 2: adp_drtlpf [2 2 2 1 1]/8; 3: adp_drtlpf [7 7 7 6 5]/32; dft=3;

NR4_SNR_ALPHA0_MAX_MIN 0x2daa

Bit(s)	R/W	Default	Description
29:23	R/W	127	reg_nr4_snr2_alp0_ymin : // unsigned , default = 127 normalized to 128 as '1'
22:16	R/W	127	reg_nr4_snr2_alp0_ymax : // unsigned , default = 127 normalized to 128 as '1'
13: 7	R/W	127	reg_nr4_snr2_alp0_cmin : // unsigned , default = 127 normalized to 128 as '1'
6: 0	R/W	127	reg_nr4_snr2_alp0_cmax : // unsigned , default = 127 normalized to 128 as '1'

NR4_ALP0C_ERR2CURV_LIMIT0 0x2dab

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_nr4_snr2_alp0_minerr_cpar0 : // unsigned , default = 0 threshold0 of curve to map mierr to alp0 for chroma channel, this will be set value of flat region mierr that no need blur.
23:16	R/W	25	reg_nr4_snr2_alp0_minerr_cpar1 : // unsigned , default = 25 threshold1 of curve to map mierr to alp0 for chroma channel,this will be set value of texture region mierr that can not blur.
15: 8	R/W	40	reg_nr4_snr2_alp0_minerr_cpar5 : // unsigned , default = 40 rate0 (for mierr<th0) of curve to map mierr to alp0 for chroma channel. the larger of the value, the deep of the slope. 0~255.
7: 0	R/W	40	reg_nr4_snr2_alp0_minerr_cpar6 : // unsigned , default = 40 rate1 (for mierr>th1) of curve to map mierr to alp0 for chroma channel. the larger of the value, the deep of the slope. 0~255.

NR4_ALP0C_ERR2CURV_LIMIT1 0x2dac

Bit(s)	R/W	Default	Description
23:16	R/W	127	reg_nr4_snr2_alp0_minerr_cpar2 : // unsigned , default = 127 level limit(for mierr<th0) of curve to map mierr to alp0 for chroma channel, that we can do for flat region. 0~255.
15: 8	R/W	0	reg_nr4_snr2_alp0_minerr_cpar3 : // unsigned , default = 0 level limit(for th0<mierr<th1) of curve to map mierr to alp0 for chroma channel, that we can do for misc region. 0~255.
7: 0	R/W	127	reg_nr4_snr2_alp0_minerr_cpar4 : // unsigned , default = 127 level limit(for mierr>th1) of curve to map mierr to alp0 for chroma channel, that we can do for texture region. 0~255.

NR4_ALP0Y_ERR2CURV_LIMIT0 0x2dad

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_nr4_snr2_alp0_minerr_ypar0 : // unsigned , default = 0 threshold0 of curve to map mierr to alp0 for luma channel, this will be set value of flat region mierr that no need blur. 0~255.
23:16	R/W	25	reg_nr4_snr2_alp0_minerr_ypar1 : // unsigned , default = 25 threshold1 of curve to map mierr to alp0 for luma channel,this will be set value of texture region mierr that can not blur.

15: 8	R/W	40	reg_nr4_snr2_alp0_minerr_ypar5 : // unsigned , default = 40 rate0 (for mierr<th0) of curve to map mierr to alp0 for luma channel. the larger of the value, the deep of the slope. 0~255.
7: 0	R/W	40	reg_nr4_snr2_alp0_minerr_ypar6 : // unsigned , default = 40 rate1 (for mierr>th1) of curve to map mierr to alp0 for luma channel. the larger of the value, the deep of the slope. 0~255.

NR4_ALP0Y_ERR2CURV_LIMIT1 0x2dae

Bit(s)	R/W	Default	Description
23:16	R/W	127	reg_nr4_snr2_alp0_minerr_ypar2 : // unsigned , default = 127 level limit(for mierr<th0) of curve to map mierr to alp0 for luma channel, set to alp0 that we can do for flat region. 0~255.
15: 8	R/W	0	reg_nr4_snr2_alp0_minerr_ypar3 : // unsigned , default = 0 level limit(for th0<mierr<th1) of curve to map mierr to alp0 for luma channel, alp0 that we can do for misc region. 0~255.
7: 0	R/W	127	reg_nr4_snr2_alp0_minerr_ypar4 : // unsigned , default = 127 level limit(for mierr>th1) of curve to map mierr to alp0 for luma channel, alp0 that we can do for texture region. 0~255.

NR4_SNR_ALPA1_RATE_AND_OFST 0x2daf

Bit(s)	R/W	Default	Description
23:18	R/W	0	reg_nr4_snr2_alp1_ycore_rate : // unsigned , default = 0 normalized 64 as "1"
17:12	R/W	0	reg_nr4_snr2_alp1_ccore_rate : // unsigned , default = 0 normalized 64 as "1"
11: 6	R/W	3	reg_nr4_snr2_alp1_ycore_ofst : // signed , default = 3 normalized 64 as "1"
5: 0	R/W	3	reg_nr4_snr2_alp1_ccore_ofst : // signed , default = 3 normalized 64 as "1"

NR4_SNR_ALPHA1_MAX_MIN 0x2db0

Bit(s)	R/W	Default	Description
23:18	R/W	0	reg_nr4_snr2_alp1_ymin : // unsigned , default = 0 normalized to 64 as '1'
17:12	R/W	63	reg_nr4_snr2_alp1_ymax : // unsigned , default = 63 normalized to 64 as '1'
11: 6	R/W	0	reg_nr4_snr2_alp1_cmin : // unsigned , default = 0 normalized to 64 as '1'
5: 0	R/W	63	reg_nr4_snr2_alp1_cmax : // unsigned , default = 63 normalized to 64 as '1'

NR4_ALP1C_ERR2CURV_LIMIT0 0x2db1

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_nr4_snr2_alp1_minerr_cpar0 : // unsigned , default = 0 annel, this will be set value of flat region mierr that no need directional NR. 0~255.
23:16	R/W	24	reg_nr4_snr2_alp1_minerr_cpar1 : // unsigned , default = 24 hannel,this will be set value of texture region mierr that can not do directional NR. 0~255.
15: 8	R/W	0	reg_nr4_snr2_alp1_minerr_cpar5 : // unsigned , default = 0 a/chroma channel. the larger of the value, the deep of the slope.

7: 0	R/W	20	reg_nr4_snr2_alp1_minerr_cpar6 : // unsigned , default = 20 a/chroma channel. the larger of the value, the deep of the slope. 0~255

NR4_ALP1C_ERR2CURV_LIMIT1 0x2db2

Bit(s)	R/W	Default	Description
23:16	R/W	0	reg_nr4_snr2_alp1_minerr_cpar2 : // unsigned , default = 0 will be set to alp1 that we can do for flat region. 0~255.
15: 8	R/W	16	reg_nr4_snr2_alp1_minerr_cpar3 : // unsigned , default = 16 this will be set to alp1 that we can do for misc region. 0~255.
7: 0	R/W	63	reg_nr4_snr2_alp1_minerr_cpar4 : // unsigned , default = 63 will be set to alp1 that we can do for texture region. 0~255.255 before

NR4_ALP1Y_ERR2CURV_LIMIT0 0x2db3

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_nr4_snr2_alp1_minerr_ypar0 : // unsigned , default = 0 thra/chroma channel, this will be set value of flat region mierr that no need directional NR. 0~255.
23:16	R/W	24	reg_nr4_snr2_alp1_minerr_ypar1 : // unsigned , default = 24 thra/chroma channel,this will be set value of texture region mierr that can not do directional NR. 0~255.
15: 8	R/W	0	reg_nr4_snr2_alp1_minerr_ypar5 : // unsigned , default = 0 ratlp1 for luma/chroma channel. the larger of the value, the deep of the slope.
7: 0	R/W	20	reg_nr4_snr2_alp1_minerr_ypar6 : // unsigned , default = 20 ratlp1 for luma/chroma channel. the larger of the value, the deep of the slope. 0~255

NR4_ALP1Y_ERR2CURV_LIMIT1 0x2db4

Bit(s)	R/W	Default	Description
23:16	R/W	0	reg_nr4_snr2_alp1_minerr_ypar2 : // unsigned , default = 0 lev to alp1 for luma/chroma channel, this will be set to alp1 that we can do for flat region. 0~255.
15: 8	R/W	16	reg_nr4_snr2_alp1_minerr_ypar3 : // unsigned , default = 16 levierr to alp1 for luma/chroma channel, this will be set to alp1 that we can do for misc region. 0~255.
7: 0	R/W	63	reg_nr4_snr2_alp1_minerr_ypar4 : // unsigned , default = 63 lev to alp1 for luma/chroma channel, this will be set to alp1 that we can do for texture region. 0~255.255 before

NR4_MTN_CTRL 0x2db5

Bit(s)	R/W	Default	Description
1	R/W	1	reg_nr4_mtn_ref_en : // unsigned , default = 1 enable motion refinement, dft = 1
0	R/W	0	reg_nr4_mtn_ref_bet_sel : // unsigned , default = 0 beta selection mode for motion refinement, 0: beta1, 1: beta2, dft = 0

NR4_MTN_REF_PAR0 0x2db6

Bit(s)	R/W	Default	Description
31:24	R/W	24	reg_nr4_mtn_ref_par0 : // unsigned , default = 24 par0 for beta to gain, dft =

Bit(s)	R/W	Default	Description
23:16	R/W	60	reg_nr4_mtn_ref_par1 : // unsigned , default = 60 par1 for beta to gain, dft =
15: 8	R/W	4	reg_nr4_mtn_ref_par2 : // unsigned , default = 4 par2 for beta to gain, dft =
7: 0	R/W	32	reg_nr4_mtn_ref_par3 : // unsigned , default = 32 par3 for beta to gain, dft =

NR4_MTN_REF_PAR1 0x2db7

Bit(s)	R/W	Default	Description
23:16	R/W	128	reg_nr4_mtn_ref_par4 : // unsigned , default = 128 par4 for beta to gain, dft =
15: 8	R/W	40	reg_nr4_mtn_ref_par5 : // unsigned , default = 40 par5 for beta to gain, dft =
7: 0	R/W	20	reg_nr4_mtn_ref_par6 : // unsigned , default = 20 par6 for beta to gain, dft =

NR4_MCNR_LUMA_ENH_CTRL 0x2db8

Bit(s)	R/W	Default	Description
3	R/W	1	reg_nr4_luma_plus_en : // unsigned , default = 1 enable luma enhancement, dft = 1
2	R/W	1	reg_nr4_luma_plus_wt_mode : // unsigned , default = 1 luma weight calc mode, 0:sqrt(1+x^2), 1: 1+abs(x), dft = 0
1: 0	R/W	1	reg_nr4_luma_plus_orient_mode : // unsigned , default = 1 0: only use previous orient for pre and cur luma plus, 1: 0: only use current orient for pre and cur luma plus

NR4_MCNR_LUMA_STAT_LIMTX 0x2db9

Bit(s)	R/W	Default	Description
29:16	R/W	8	reg_nr4_luma_plus_xst : // unsigned , default = 8 start for luma plus statistic, dft = 8
13: 0	R/W	711	reg_nr4_luma_plus_xed : // unsigned , default = 711 end for luma plus statistic, dft = HSIZE-8-1;

NR4_MCNR_LUMA_STAT_LIMTY 0x2dba

Bit(s)	R/W	Default	Description
29:16	R/W	8	reg_nr4_luma_plus_yst : // unsigned , default = 8 start for luma plus statistic, dft = 8
13: 0	R/W	231	reg_nr4_luma_plus_yed : // unsigned , default = 231 end for luma plus statistic, dft = VSIZE-8-1

NR4_MCNR_LUMA_DIF_CALC 0x2dbb

Bit(s)	R/W	Default	Description
29:24	R/W	8	reg_nr4_luma_plus_ugain : // unsigned , default = 8 U's gain for luma enhancement, 16 normalized as '1'
21:16	R/W	8	reg_nr4_luma_plus_vgain : // unsigned , default = 8 V's gain for luma enhancement, 16 normalized as '1'

Bit(s)	R/W	Default	Description
15: 8	R/W	2	reg_nr4_luma_plus_ycor_thd : // unsigned , default = 2 Y coring threshold for difference calc., dft = 0
7: 0	R/W	0	reg_nr4_luma_plus_ccor_thd : // unsigned , default = 0 C coring threshold for difference calc., dft = 0

NR4_MCNR_LUMAPRE_CAL_PRAM 0x2dbc

Bit(s)	R/W	Default	Description
25:24	R/W	0	reg_nr4_pre_u_orient : // signed , default = 0 orientation of previous U, initial to 0, and will be updated by software
17:16	R/W	0	reg_nr4_pre_v_orient : // signed , default = 0 orientation of previous V, initial to 0, and will be updated by software
15: 8	R/W	0	reg_nr4_pre_u_mean : // unsigned , default = 0 mean of previous U, initial to 0, and will be updated by software
7: 0	R/W	0	reg_nr4_pre_v_mean : // unsigned , default = 0 mean of previousV, initial to 0, and will be updated by software

NR4_MCNR_LUMACUR_CAL_PRAM 0x2dbd

Bit(s)	R/W	Default	Description
25:24	R/W	0	reg_nr4_cur_u_orient : // signed , default = 0 orientation of current U, initial to 0, and will be updated by software
17:16	R/W	0	reg_nr4_cur_v_orient : // signed , default = 0 orientation of current V, initial to 0, and will be updated by software
15: 8	R/W	0	reg_nr4_cur_u_mean : // unsigned , default = 0 mean of current U, initial to 0, and will be updated by software
7: 0	R/W	0	reg_nr4_cur_v_mean : // unsigned , default = 0 mean of current, initial to 0, and will be updated by software

NR4_MCNR_MV_CTRL_REG 0x2dbe

Bit(s)	R/W	Default	Description
13:12	R/W	2	reg_nr4_sad_bitw : // unsigned , default = 2 sad bit width (8 + x) before clip to u8, dft = 1
11: 4	R/W	64	reg_nr4_glb_gain : // unsigned , default = 64 global gain calc. by software, 64 is normalized as '1'
3: 0	R/W	8	reg_nr4_mv_err_rsft : // unsigned , default = 8 right shift for mv err calc., dft = 9

NR4_MCNR_MV_GAIN0 0x2dbf

Bit(s)	R/W	Default	Description
31:28	R/W	1	reg_nr4_lftmvx_gain : // unsigned , default = 1 left mvx gain for err calc., dft = 1
27:24	R/W	1	reg_nr4_lftmvy_gain : // unsigned , default = 1 left mvy gain for err calc., dft = 1
23:20	R/W	5	reg_nr4_zmvx_gain : // unsigned , default = 5 zero mvx gain for err calc., dft = 2

19:16	R/W	5	reg_nr4_zmvy_gain : // unsigned , default = 5 zero mvy gain for err calc., dft = 4
15:12	R/W	2	reg_nr4_lmvx0_gain : // unsigned , default = 2 line mvx0 gain for err calc., dft = 1
11: 8	R/W	2	reg_nr4_lmvx1_gain : // unsigned , default = 2 line mvx1 gain for err calc., dft = 1
7: 4	R/W	2	reg_nr4_lmvy0_gain : // unsigned , default = 2 line mvy0 gain for err calc., dft = 1
3: 0	R/W	2	reg_nr4_lmvy1_gain : // unsigned , default = 2 line mvy1 gain for err calc., dft = 1

NR4_MCNR_LMV_PARM 0x2dc0

Bit(s)	R/W	Default	Description
31:28	R/W	3	reg_nr4_lmv_rt0 : // unsigned , default = 3 ratio of max lmv
27:24	R/W	3	reg_nr4_lmv_rt1 : // unsigned , default = 3 ratio of second max lmv
21:16	R/W	16	reg_nr4_lmv_num_lmt0 : // unsigned , default = 16 lmv0 least/limit number of (total number - zero_bin)
13: 8	R/W	8	reg_nr4_lmv_num_lmt1 : // unsigned , default = 8 lmv1 least/limit number of (total number - zero_bin - max0)
1: 0	R/W	1	reg_nr4_max_sad_rng : // unsigned , default = 1 search range of max2 sad in small region, dft = 1

NR4_MCNR_ALP0_REG 0x2dc1

Bit(s)	R/W	Default	Description
25	R/W	1	reg_nr4_alp0_fail_chk : // unsigned , default = 1 enable check for alp0 fail status
24	R/W	1	reg_nr4_bet0_coef_ref_en : // unsigned , default = 1 bet1 refinement by coef_blt
23:16	R/W	255	reg_nr4_alp0_posad_gain : // unsigned , default = 255 the sad (norm) gain for pixel pointed by MV;
9: 8	R/W	0	reg_nr4_alp0_norm_mode : // unsigned , default = 0 alp0 select sad norm mode, 0: disable, 1: enable dc norm, 2: enable ac norm, 3: enable both (dc/ac) norm, dft = 3
5: 0	R/W	16	reg_nr4_alp0_norm_gain : // unsigned , default = 16 alp0 gain for sad norm, '32' as '1', dft = 1

NR4_MCNR_ALP1_AND_BET0_REG 0x2dc2

Bit(s)	R/W	Default	Description
25:24	R/W	3	reg_nr4_alp1_norm_mode : // unsigned , default = 3 alp1 select sad norm mode, 0: disable, 1: enable dc norm, 2: enable ac norm, 3: enable both (dc/ac) norm, dft = 3
21:16	R/W	3	reg_nr4_alp1_norm_gain : // unsigned , default = 3 alp1 gain for sad norm, '32' as '1', dft = 1
9: 8	R/W	3	reg_nr4_bet0_norm_mode : // unsigned , default = 3 bet0 select sad norm mode, 0: disable, 1: enable dc norm, 2: enable ac norm, 3: enable both (dc/ac) norm, dft = 3
5: 0	R/W	8	reg_nr4_bet0_norm_gain : // unsigned , default = 8 bet0 gain for sad norm, '32' as '1', dft = 1

NR4_MCNR_BET1_AND_BET2_REG 0x2dc3

Bit(s)	R/W	Default	Description
25:24	R/W	3	reg_nr4_bet1_norm_mode : // unsigned , default = 3 bet1 select sad norm mode, 0: disable, 1: enable dc norm, 2: enable ac norm, 3: enable both (dc/ac) norm, dft = 3
21:16	R/W	8	reg_nr4_bet1_norm_gain : // unsigned , default = 8 bet1 gain for sad norm, '32' as '1', dft = 1
9: 8	R/W	0	reg_nr4_bet2_norm_mode : // unsigned , default = 0 bet2 select sad norm mode, 0: disable, 1: enable dc norm, 2: enable ac norm, 3: enable both (dc/ac) norm, dft = 3
5: 0	R/W	16	reg_nr4_bet2_norm_gain : // unsigned , default = 16 bet2 gain for sad norm, '32' as '1', dft = 1

NR4_MCNR_AC_DC_CRTL 0x2dc4

Bit(s)	R/W	Default	Description
11	R/W	1	reg_nr4_dc_mode : // unsigned , default = 1 mode for dc selection,0: Y_lpf, 1: Y_lpf + (U_Lpf+V_lpf)/2,
10	R/W	1	reg_nr4_ac_mode : // unsigned , default = 1 mode for ac selection, 0: Y_abs_dif, 1: Y_abs_dif + (U_abs_dif + V_abs_dif)/2
9	R/W	0	reg_nr4_dc_sel : // unsigned , default = 0 selection mode for dc value, 0: 3x5, 1: 5x5, dft = 1
8	R/W	0	reg_nr4_ac_sel : // unsigned , default = 0 selection mode for ac value, 0: 3x5, 1: 5x5, dft = 1
6: 4	R/W	2	reg_nr4_dc_shft : // unsigned , default = 2 right shift for dc value, dft = 2
2: 0	R/W	0	reg_nr4_ac_shft : // unsigned , default = 0 right shift for ac value, dft = 2

NR4_MCNR_CM_CTRL0 0x2dc5

Bit(s)	R/W	Default	Description
28	R/W	0	reg_nr4_cm_skin_prc_bet0 : // unsigned , default = 0 enable skin tone processing for mcnr bet0 calc., dft = 1
27:26	R/W	1	reg_nr4_cm_chrm_sel : // unsigned , default = 1 chrome selection for color match, 0: 1x1, 1: 3X3LPF, 2: 3x5LPF, 3: 5x5LPF for 5lines, 3x5LPF for 3lines, dft = 3
25:24	R/W	1	reg_nr4_cm_luma_sel : // unsigned , default = 1 luma selection for color match, 0: 1x1, 1: 3X3LPF, 2: 3x5LPF, 3: 5x5LPF for 5lines, 3x5LPF for 3lines, dft = 3
23:21	R/W	3	reg_nr4_cm_skin_rshft_bet0 : // unsigned , default = 3 right shift for bet0's skin color gains, dft = 3
20	R/W	1	reg_nr4_cm_var_sel : // unsigned , default = 1 variation selection for color match, 0: 3x5, 1: 5x5 for 5lines, 3x5 for 3lines, dft = 1
19	R/W	1	reg_nr4_cm_green_prc_bet0 : // unsigned , default = 1 enable green processing for mcnr bet0 calc., dft = 1
18:16	R/W	4	reg_nr4_cm_green_rshft_bet0 : // unsigned , default = 4 right shift for bet0's green color gains, dft = 4

Bit(s)	R/W	Default	Description
15:14	R/W	2	reg_nr4_preflt_mod : // unsigned , default = 2 pre filter mode in mcnr, 0: mv pointed pixel, 1: bilater filter
13:12	R/W	1	reg_nr4_alp1_mode : // unsigned , default = 1 mode for alpha1's sad selection, 0: max sad, 1: three min sads, 2: min sad, 3: co sad
9: 8	R/W	0	reg_nr4_bet0_mode : // unsigned , default = 0 mode for bet0's sad selection, 0: max sad, 1: three min sads, 2: min sad, 3: co sad, else: (co sad) - (min sad)
5: 4	R/W	2	reg_nr4_bet1_mode : // unsigned , default = 2 mode for bet1's sad selection, 0: max sad, 1: three min sads, 2: min sad, 3: co sad, else: (co sad) - (min sad)
1: 0	R/W	1	reg_nr4_bet2_mode : // unsigned , default = 1 mode for bet2's sad selection, 0: max sad, 1: three min sads, 2: min sad, 3: co sad, else: (co sad) - (min sad)

NR4_MCNR_CM_PRAM 0x2dc6

Bit(s)	R/W	Default	Description
29	R/W	1	reg_nr4_cm_blue_prc_alp0 : // unsigned , default = 1 enable blue processing for mcnr alpha0 calc., dft = 1
28	R/W	1	reg_nr4_cm_blue_prc_alp1 : // unsigned , default = 1 enable blue processing for mcnr alpha1 calc., dft = 1
27	R/W	1	reg_nr4_cm_skin_prc_alp0 : // unsigned , default = 1 enable skin tone processing for mcnr alpha0 calc., dft = 1
26	R/W	1	reg_nr4_cm_green_prc_alp0 : // unsigned , default = 1 enable green processing for mcnr alpha0 clac., dft = 1
25	R/W	1	reg_nr4_cm_skin_prc_alp1 : // unsigned , default = 1 enable skin tone processing for mcnr alpha0 calc., dft = 1
24	R/W	1	reg_nr4_cm_green_prc_alp1 : // unsigned , default = 1 enable green processing for mcnr alpha1 clac., dft = 1
23:20	R/W	13	reg_nr4_cm_blue_hue_st : // unsigned , default = 13 hue start of blue, dft =
19:16	R/W	15	reg_nr4_cm_blue_hue_ed : // unsigned , default = 15 hue end of blue, dft =
15:12	R/W	7	reg_nr4_cm_green_hue_st : // unsigned , default = 7 hue start of green, dft =
11: 8	R/W	10	reg_nr4_cm_green_hue_ed : // unsigned , default = 10 hue end of green, dft =
7: 4	R/W	5	reg_nr4_cm_skin_hue_st : // unsigned , default = 5 hue start of skin, dft =
3: 0	R/W	6	reg_nr4_cm_skin_hue_ed : // unsigned , default = 6 hue end of skin, dft =

NR4_MCNR_CM_RSHFT_ALP0 0x2dc7

Bit(s)	R/W	Default	Description
27:25	R/W	5	reg_nr4_cm_blue_rshft_bet0 : // unsigned , default = 5 right shift for bet0's blue color gains, dft = 5
24	R/W	1	reg_nr4_cm_blue_prc_bet0 : // unsigned , default = 1 enable blue processing for mcnr bet0 calc., dft = 1

22:20	R/W	5	reg_nr4_cm_blue_rshft_alp0 : // unsigned , default = 5 right shift for alpha0/1's blue color gains, dft = 5
18:16	R/W	5	reg_nr4_cm_blue_rshft_alp1 : // unsigned , default = 5 right shift for alpha0/1's blue color gains, dft = 5
14:12	R/W	4	reg_nr4_cm_green_rshft_alp0 : // unsigned , default = 4 right shift for alpha0/1's green color gains, dft = 4
10: 8	R/W	4	reg_nr4_cm_green_rshft_alp1 : // unsigned , default = 4 right shift for alpha0/1's green color gains, dft = 4
6: 4	R/W	3	reg_nr4_cm_skin_rshft_alp0 : // unsigned , default = 3 right shift for alpha0/1's skin color gains, dft = 3
2: 0	R/W	3	reg_nr4_cm_skin_rshft_alp1 : // unsigned , default = 3 right shift for alpha0/1's skin color gains, dft = 3

NR4_MCNR_BLUE_CENT 0x2dc8

Bit(s)	R/W	Default	Description
23:16	R/W	157	reg_nr4_cm_blue_centx : // unsigned , default = 157 x coordinate of center of blue, dft =
7: 0	R/W	110	reg_nr4_cm_blue_centy : // unsigned , default = 110 y coordinate of center of blue, dft =

NR4_MCNR_BLUE_GAIN_PAR0 0x2dc9

Bit(s)	R/W	Default	Description
31:24	R/W	32	reg_nr4_cm_blue_gain_par0 : // unsigned , default = 32 par0 for blue gain, dft =
23:16	R/W	255	reg_nr4_cm_blue_gain_par1 : // unsigned , default = 255 par1 for blue gain, dft =
15: 8	R/W	4	reg_nr4_cm_blue_gain_par2 : // unsigned , default = 4 par2 for blue gain, dft =
7: 0	R/W	32	reg_nr4_cm_blue_gain_par3 : // unsigned , default = 32 par3 for blue gain, dft =

NR4_MCNR_BLUE_GAIN_PAR1 0x2dca

Bit(s)	R/W	Default	Description
23:16	R/W	32	reg_nr4_cm_blue_gain_par4 : // unsigned , default = 32 par4 for blue gain, dft =
15: 8	R/W	32	reg_nr4_cm_blue_gain_par5 : // unsigned , default = 32 par5 for blue gain, dft =
7: 0	R/W	0	reg_nr4_cm_blue_gain_par6 : // unsigned , default = 0 par6 for blue gain, dft =

NR4_MCNR_CM_BLUE_CLIP0 0x2dcb

Bit(s)	R/W	Default	Description
23:16	R/W	40	reg_nr4_cm_blue_luma_min : // unsigned , default = 40 luma min for blue color matching, dft =
7: 0	R/W	180	reg_nr4_cm_blue_luma_max : // unsigned , default = 180 luma max for blue color matching, dft =

NR4_MCNR_CM_BLUE_CLIP1 0x2dcc

Bit(s)	R/W	Default	Description
31:24	R/W	5	reg_nr4_cm_blue_sat_min : // unsigned , default = 5 saturation min for blue color matching, dft =
23:16	R/W	255	reg_nr4_cm_blue_sat_max : // unsigned , default = 255 saturation max for blue color matching, dft =
15: 8	R/W	0	reg_nr4_cm_blue_var_min : // unsigned , default = 0 variation min for blue color matching, dft =
7: 0	R/W	12	reg_nr4_cm_blue_var_max : // unsigned , default = 12 variation max for blue color matching, dft =

NR4_MCNR_GREEN_CENT 0x2dcd

Bit(s)	R/W	Default	Description
23:16	R/W	114	reg_nr4_cm_green_centx : // unsigned , default = 114 x coordinate of center of green, dft =
7: 0	R/W	126	reg_nr4_cm_green_centy : // unsigned , default = 126 y coordinate of center of green, dft =

NR4_MCNR_GREEN_GAIN_PAR0 0x2dce

Bit(s)	R/W	Default	Description
31:24	R/W	16	reg_nr4_cm_green_gain_par0 : // unsigned , default = 16 par0 for green gain, dft =
23:16	R/W	255	reg_nr4_cm_green_gain_par1 : // unsigned , default = 255 par1 for green gain, dft =
15: 8	R/W	255	reg_nr4_cm_green_gain_par2 : // unsigned , default = 255 par2 for green gain, dft =
7: 0	R/W	16	reg_nr4_cm_green_gain_par3 : // unsigned , default = 16 par3 for green gain, dft =

NR4_MCNR_GREEN_GAIN_PAR1 0x2dcf

Bit(s)	R/W	Default	Description
23:16	R/W	16	reg_nr4_cm_green_gain_par4 : // unsigned , default = 16 par4 for green gain, dft =
15: 8	R/W	128	reg_nr4_cm_green_gain_par5 : // unsigned , default = 128 par5 for green gain, dft =
7: 0	R/W	0	reg_nr4_cm_green_gain_par6 : // unsigned , default = 0 par6 for green gain, dft =

NR4_MCNR_GREEN_CLIP0 0x2dd0

Bit(s)	R/W	Default	Description
23:16	R/W	40	reg_nr4_cm_green_luma_min : // unsigned , default = 40 luma min for green color matching, dft =
7: 0	R/W	160	reg_nr4_cm_green_luma_max : // unsigned , default = 160 luma max for green color matching, dft =

NR4_MCNR_GREEN_CLIP2 0x2dd1

Bit(s)	R/W	Default	Description
31:24	R/W	4	reg_nr4_cm_green_sat_min : // unsigned , default = 4 saturation min for green color matching, dft =
23:16	R/W	255	reg_nr4_cm_green_sat_max : // unsigned , default = 255 saturation max for green color matching, dft =
15: 8	R/W	0	reg_nr4_cm_green_var_min : // unsigned , default = 0 variation min for green color matching, dft =
7: 0	R/W	12	reg_nr4_cm_green_var_max : // unsigned , default = 12 variation max for green color matching, dft =

NR4_MCNR_SKIN_CENT 0x2dd2

Bit(s)	R/W	Default	Description
23:16	R/W	112	reg_nr4_cm_skin_centx : // unsigned , default = 112 x coordinate of center of skin tone, dft =
7: 0	R/W	149	reg_nr4_cm_skin_centy : // unsigned , default = 149 y coordinate of center of skin tone, dft =

NR4_MCNR_SKIN_GAIN_PAR0 0x2dd3

Bit(s)	R/W	Default	Description
31:24	R/W	20	reg_nr4_cm_skin_gain_par0 : // unsigned , default = 20 par0 for skin gain, dft =
23:16	R/W	255	reg_nr4_cm_skin_gain_par1 : // unsigned , default = 255 par1 for skin gain, dft =
15: 8	R/W	255	reg_nr4_cm_skin_gain_par2 : // unsigned , default = 255 par2 for skin gain, dft =
7: 0	R/W	8	reg_nr4_cm_skin_gain_par3 : // unsigned , default = 8 par3 for skin gain, dft =

NR4_MCNR_SKIN_GAIN_PAR1 0x2dd4

Bit(s)	R/W	Default	Description
23:16	R/W	8	reg_nr4_cm_skin_gain_par4 : // unsigned , default = 8 par4 for skin gain, dft =
15: 8	R/W	128	reg_nr4_cm_skin_gain_par5 : // unsigned , default = 128 par5 for skin gain, dft =
7: 0	R/W	0	reg_nr4_cm_skin_gain_par6 : // unsigned , default = 0 par6 for skin gain, dft =

NR4_MCNR_SKIN_CLIP0 0x2dd5

Bit(s)	R/W	Default	Description
23:16	R/W	40	reg_nr4_cm_skin_luma_min : // unsigned , default = 40 luma min for skin color matching, dft =
7: 0	R/W	180	reg_nr4_cm_skin_luma_max : // unsigned , default = 180 luma max for skin color matching, dft =

NR4_MCNR_SKIN_CLIP1 0x2dd6

Bit(s)	R/W	Default	Description
31:24	R/W	5	reg_nr4_cm_skin_sat_min : // unsigned , default = 5 saturation min for skin color matching, dft =
23:16	R/W	255	reg_nr4_cm_skin_sat_max : // unsigned , default = 255 saturation max for skin color matching, dft =
15: 8	R/W	0	reg_nr4_cm_skin_var_min : // unsigned , default = 0 variation min for skin color matching, dft =
7: 0	R/W	12	reg_nr4_cm_skin_var_max : // unsigned , default = 12 variation max for skin color matching, dft =

NR4_MCNR_ALP1_GLB_CTRL 0x2dd7

Bit(s)	R/W	Default	Description
31	R/W	0	reg_nr4_alp1_glb_gain_en : // unsigned , default = 0 alp1 adjust by global gain, dft = 1
30:28	R/W	6	reg_nr4_alp1_glb_gain_lsft : // unsigned , default = 6 alp1 left shift before combine with global gain
27	R/W	1	reg_nr4_bet0_glb_gain_en : // unsigned , default = 1 bet0 adjust by global gain, dft = 1
26:24	R/W	6	reg_nr4_bet0_glb_gain_lsft : // unsigned , default = 6 bet1 left shift before combine with global gain
23	R/W	0	reg_nr4_bet1_glb_gain_en : // unsigned , default = 0 bet1 adjust by global gain, dft = 0
22:20	R/W	6	reg_nr4_bet1_glb_gain_lsft : // unsigned , default = 6 bet1 left shift before combine with global gain
19	R/W	1	reg_nr4_bet2_glb_gain_en : // unsigned , default = 1 bet2 adjust by global gain, dft = 1
18:16	R/W	6	reg_nr4_bet2_glb_gain_lsft : // unsigned , default = 6 bet2 left shift before combine with global gain
15	R/W	1	reg_nr4_alp1_ac_en : // unsigned , default = 1 alp1 adjust by ac, dft = 1
14:12	R/W	5	reg_nr4_alp1_ac_lsft : // unsigned , default = 5 alp1 left shift before combine with ac
11	R/W	0	reg_nr4_bet0_ac_en : // unsigned , default = 0 bet0 adjust by ac, dft = 1
10: 8	R/W	5	reg_nr4_bet0_ac_lsft : // unsigned , default = 5 bet0 left shift before combine with ac
7	R/W	0	reg_nr4_bet1_ac_en : // unsigned , default = 0 bet1 adjust by ac, dft = 1
6: 4	R/W	5	reg_nr4_bet1_ac_lsft : // unsigned , default = 5 bet1 left shift before combine with ac
3	R/W	0	reg_nr4_bet2_ac_en : // unsigned , default = 0 bet2 adjust by ac, dft = 1
2: 0	R/W	5	reg_nr4_bet2_ac_lsft : // unsigned , default = 5 bet2 left shift before combine with ac

NR4_MCNR_DC2NORM_LUT0 0x2dd8

Bit(s)	R/W	Default	Description

28:24	R/W	16	reg_nr4_dc2norm_lut0 : // unsigned , default = 16 normal 0~16, dc to norm for alpha adjust, dft =
20:16	R/W	16	reg_nr4_dc2norm_lut1 : // unsigned , default = 16 normal 0~16, dc to norm for alpha adjust, dft =
12: 8	R/W	16	reg_nr4_dc2norm_lut2 : // unsigned , default = 16 normal 0~16, dc to norm for alpha adjust, dft =
4: 0	R/W	16	reg_nr4_dc2norm_lut3 : // unsigned , default = 16 normal 0~16, dc to norm for alpha adjust, dft =

NR4_MCNR_DC2NORM_LUT1 0x2dd9

Bit(s)	R/W	Default	Description
28:24	R/W	16	reg_nr4_dc2norm_lut4 : // unsigned , default = 16 normal 0~16, dc to norm for alpha adjust, dft =
20:16	R/W	16	reg_nr4_dc2norm_lut5 : // unsigned , default = 16 normal 0~16, dc to norm for alpha adjust, dft =
12: 8	R/W	16	reg_nr4_dc2norm_lut6 : // unsigned , default = 16 normal 0~16, dc to norm for alpha adjust, dft =
4: 0	R/W	12	reg_nr4_dc2norm_lut7 : // unsigned , default = 12 normal 0~16, dc to norm for alpha adjust, dft =

NR4_MCNR_DC2NORM_LUT2 0x2dda

Bit(s)	R/W	Default	Description
4: 0	R/W	8	reg_nr4_dc2norm_lut8 : // unsigned , default = 8 normal 0~16, dc to norm for alpha adjust, dft =

NR4_MCNR_AC2NORM_LUT0 0x2ddb

Bit(s)	R/W	Default	Description
28:24	R/W	2	reg_nr4_ac2norm_lut0 : // unsigned , default = 2 normal 0~16, ac to norm for alpha adjust, dft =
20:16	R/W	16	reg_nr4_ac2norm_lut1 : // unsigned , default = 16 normal 0~16, ac to norm for alpha adjust, dft =
12: 8	R/W	16	reg_nr4_ac2norm_lut2 : // unsigned , default = 16 normal 0~16, ac to norm for alpha adjust, dft =
4: 0	R/W	12	reg_nr4_ac2norm_lut3 : // unsigned , default = 12 normal 0~16, ac to norm for alpha adjust, dft =

NR4_MCNR_AC2NORM_LUT1 0x2ddc

Bit(s)	R/W	Default	Description
28:24	R/W	4	reg_nr4_ac2norm_lut4 : // unsigned , default = 4 normal 0~16, ac to norm for alpha adjust, dft =

20:16	R/W	2	reg_nr4_ac2norm_lut5 : // unsigned , default = 2 normal 0~16, ac to norm for alpha adjust, dft =
12: 8	R/W	1	reg_nr4_ac2norm_lut6 : // unsigned , default = 1 normal 0~16, ac to norm for alpha adjust, dft =
4: 0	R/W	1	reg_nr4_ac2norm_lut7 : // unsigned , default = 1 normal 0~16, ac to norm for alpha adjust, dft =

NR4_MCNR_AC2NORM_LUT2 0x2ddd

Bit(s)	R/W	Default	Description
4: 0	R/W	1	reg_nr4_ac2norm_lut8 : // unsigned , default = 1 normal 0~16, ac to norm for alpha adjust, dft =

NR4_MCNR_SAD2ALP0_LUT0 0x2dde

Bit(s)	R/W	Default	Description
31:24	R/W	255	reg_nr4_sad2alp0_lut0 : // unsigned , default = 255 sad to alpha0 for temporal pixel value, dft = 255
23:16	R/W	252	reg_nr4_sad2alp0_lut1 : // unsigned , default = 252 sad to alpha0 for temporal pixel value, dft = 252
15: 8	R/W	249	reg_nr4_sad2alp0_lut2 : // unsigned , default = 249 sad to alpha0 for temporal pixel value, dft = 249
7: 0	R/W	235	reg_nr4_sad2alp0_lut3 : // unsigned , default = 235 sad to alpha0 for temporal pixel value, dft = 70

NR4_MCNR_SAD2ALP0_LUT1 0x2ddf

Bit(s)	R/W	Default	Description
31:24	R/W	185	reg_nr4_sad2alp0_lut4 : // unsigned , default = 185 sad to alpha0 for temporal pixel value, dft = 12
23:16	R/W	70	reg_nr4_sad2alp0_lut5 : // unsigned , default = 70 sad to alpha0 for temporal pixel value, dft = 1
15: 8	R/W	14	reg_nr4_sad2alp0_lut6 : // unsigned , default = 14 sad to alpha0 for temporal pixel value, dft = 0
7: 0	R/W	1	reg_nr4_sad2alp0_lut7 : // unsigned , default = 1 sad to alpha0 for temporal pixel value, dft = 0

NR4_MCNR_SAD2ALP0_LUT2 0x2de0

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_nr4_sad2alp0_lut8 : // unsigned , default = 0 sad to alpha0 for temporal pixel value, dft = 0
23:16	R/W	0	reg_nr4_sad2alp0_lut9 : // unsigned , default = 0 sad to alpha0 for temporal pixel value, dft = 0
15: 8	R/W	0	reg_nr4_sad2alp0_lut10 : // unsigned , default = 0 sad to alpha0 for temporal pixel value, dft = 0

7: 0	R/W	0	reg_nr4_sad2alp0_lut11 : // unsigned , default = 0 sad to alpha0 for temporal pixel value, dft = 0

NR4_MCNR_SAD2ALP0_LUT3 0x2de1

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_nr4_sad2alp0_lut12 : // unsigned , default = 0 sad to alpha0 for temporal pixel value, dft = 0
23:16	R/W	0	reg_nr4_sad2alp0_lut13 : // unsigned , default = 0 sad to alpha0 for temporal pixel value, dft = 0
15: 8	R/W	0	reg_nr4_sad2alp0_lut14 : // unsigned , default = 0 sad to alpha0 for temporal pixel value, dft = 0
7: 0	R/W	0	reg_nr4_sad2alp0_lut15 : // unsigned , default = 0 sad to alpha0 for temporal pixel value, dft = 0

NR4_MCNR_SAD2ALP1_LUT0 0x2de2

Bit(s)	R/W	Default	Description
31:24	R/W	192	reg_nr4_sad2alp1_lut0 : // unsigned , default = 192 sad to alpha1 for temporal blending, dft = 128
23:16	R/W	160	reg_nr4_sad2alp1_lut1 : // unsigned , default = 160 sad to alpha1 for temporal blending, dft = 128
15: 8	R/W	128	reg_nr4_sad2alp1_lut2 : // unsigned , default = 128 sad to alpha1 for temporal blending, dft = 128
7: 0	R/W	96	reg_nr4_sad2alp1_lut3 : // unsigned , default = 96 sad to alpha1 for temporal blending, dft = 64

NR4_MCNR_SAD2ALP1_LUT1 0x2de3

Bit(s)	R/W	Default	Description
31:24	R/W	64	reg_nr4_sad2alp1_lut4 : // unsigned , default = 64 sad to alpha1 for temporal blending, dft = 64
23:16	R/W	32	reg_nr4_sad2alp1_lut5 : // unsigned , default = 32 sad to alpha1 for temporal blending, dft = 128
15: 8	R/W	16	reg_nr4_sad2alp1_lut6 : // unsigned , default = 16 sad to alpha1 for temporal blending, dft = 255
7: 0	R/W	8	reg_nr4_sad2alp1_lut7 : // unsigned , default = 8 sad to alpha1 for temporal blending, dft = 255

NR4_MCNR_SAD2ALP1_LUT2 0x2de4

Bit(s)	R/W	Default	Description
31:24	R/W	4	reg_nr4_sad2alp1_lut8 : // unsigned , default = 4 sad to alpha1 for temporal blending, dft = 255
23:16	R/W	0	reg_nr4_sad2alp1_lut9 : // unsigned , default = 0 sad to alpha1 for temporal blending, dft = 255

Bit(s)	R/W	Default	Description
15: 8	R/W	16	reg_nr4_sad2alp1_lut10 : // unsigned , default = 16 sad to alpha1 for temporal blending, dft = 255
7: 0	R/W	64	reg_nr4_sad2alp1_lut11 : // unsigned , default = 64 sad to alpha1 for temporal blending, dft = 255

NR4_MCNR_SAD2ALP1_LUT3 0x2de5

Bit(s)	R/W	Default	Description
31:24	R/W	96	reg_nr4_sad2alp1_lut12 : // unsigned , default = 96 sad to alpha1 for temporal blending, dft = 255
23:16	R/W	224	reg_nr4_sad2alp1_lut13 : // unsigned , default = 224 sad to alpha1 for temporal blending, dft = 255
15: 8	R/W	255	reg_nr4_sad2alp1_lut14 : // unsigned , default = 255 sad to alpha1 for temporal blending, dft = 255
7: 0	R/W	255	reg_nr4_sad2alp1_lut15 : // unsigned , default = 255 sad to alpha1 for temporal blending, dft = 255

NR4_MCNR_SAD2BET0_LUT0 0x2de6

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_nr4_sad2bet0_lut0 : // unsigned , default = 0 sad to beta0 for tnr and mcnr blending, dft = 0
23:16	R/W	2	reg_nr4_sad2bet0_lut1 : // unsigned , default = 2 sad to beta0 for tnr and mcnr blending, dft = 2
15: 8	R/W	4	reg_nr4_sad2bet0_lut2 : // unsigned , default = 4 sad to beta0 for tnr and mcnr blending, dft = 4
7: 0	R/W	8	reg_nr4_sad2bet0_lut3 : // unsigned , default = 8 sad to beta0 for tnr and mcnr blending, dft = 8

NR4_MCNR_SAD2BET0_LUT1 0x2de7

Bit(s)	R/W	Default	Description
31:24	R/W	16	reg_nr4_sad2bet0_lut4 : // unsigned , default = 16 sad to beta0 for tnr and mcnr blending, dft = 16
23:16	R/W	32	reg_nr4_sad2bet0_lut5 : // unsigned , default = 32 sad to beta0 for tnr and mcnr blending, dft = 32
15: 8	R/W	48	reg_nr4_sad2bet0_lut6 : // unsigned , default = 48 sad to beta0 for tnr and mcnr blending, dft = 48
7: 0	R/W	64	reg_nr4_sad2bet0_lut7 : // unsigned , default = 64 sad to beta0 for tnr and mcnr blending, dft = 64

NR4_MCNR_SAD2BET0_LUT2 0x2de8

Bit(s)	R/W	Default	Description

31:24	R/W	80	reg_nr4_sad2bet0_lut8 : // unsigned , default = 80 sad to beta0 for tnr and mcnr blending, dft = 80
23:16	R/W	96	reg_nr4_sad2bet0_lut9 : // unsigned , default = 96 sad to beta0 for tnr and mcnr blending, dft = 96
15: 8	R/W	112	reg_nr4_sad2bet0_lut10 : // unsigned , default = 112 sad to beta0 for tnr and mcnr blending, dft = 112
7: 0	R/W	128	reg_nr4_sad2bet0_lut11 : // unsigned , default = 128 sad to beta0 for tnr and mcnr blending, dft = 128

NR4_MCNR_SAD2BET0_LUT3 0x2de9

Bit(s)	R/W	Default	Description
31:24	R/W	196	reg_nr4_sad2bet0_lut12 : // unsigned , default = 196 sad to beta0 for tnr and mcnr blending, dft = 160
23:16	R/W	224	reg_nr4_sad2bet0_lut13 : // unsigned , default = 224 sad to beta0 for tnr and mcnr blending, dft = 192
15: 8	R/W	255	reg_nr4_sad2bet0_lut14 : // unsigned , default = 255 sad to beta0 for tnr and mcnr blending, dft = 224
7: 0	R/W	255	reg_nr4_sad2bet0_lut15 : // unsigned , default = 255 sad to beta0 for tnr and mcnr blending, dft = 255

NR4_MCNR_SAD2BET1_LUT0 0x2dea

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_nr4_sad2bet1_lut0 : // unsigned , default = 0 sad to beta1 for deghost blending, dft = 0
23:16	R/W	2	reg_nr4_sad2bet1_lut1 : // unsigned , default = 2 sad to beta1 for deghost blending, dft = 2
15: 8	R/W	4	reg_nr4_sad2bet1_lut2 : // unsigned , default = 4 sad to beta1 for deghost blending, dft = 4
7: 0	R/W	8	reg_nr4_sad2bet1_lut3 : // unsigned , default = 8 sad to beta1 for deghost blending, dft = 8

NR4_MCNR_SAD2BET1_LUT1 0x2deb

Bit(s)	R/W	Default	Description
31:24	R/W	16	reg_nr4_sad2bet1_lut4 : // unsigned , default = 16 sad to beta1 for deghost blending, dft = 16
23:16	R/W	32	reg_nr4_sad2bet1_lut5 : // unsigned , default = 32 sad to beta1 for deghost blending, dft = 32
15: 8	R/W	48	reg_nr4_sad2bet1_lut6 : // unsigned , default = 48 sad to beta1 for deghost blending, dft = 48
7: 0	R/W	64	reg_nr4_sad2bet1_lut7 : // unsigned , default = 64 sad to beta1 for deghost blending, dft = 64

NR4_MCNR_SAD2BET1_LUT2 0x2dec

Bit(s)	R/W	Default	Description
31:24	R/W	80	reg_nr4_sad2bet1_lut8 : // unsigned , default = 80 sad to beta1 for deghost blending, dft = 80
23:16	R/W	96	reg_nr4_sad2bet1_lut9 : // unsigned , default = 96 sad to beta1 for deghost blending, dft = 96
15: 8	R/W	112	reg_nr4_sad2bet1_lut10 : // unsigned , default = 112 sad to beta1 for deghost blending, dft = 112
7: 0	R/W	128	reg_nr4_sad2bet1_lut11 : // unsigned , default = 128 sad to beta1 for deghost blending, dft = 128

NR4_MCNR_SAD2BET1_LUT3 0x2ded

Bit(s)	R/W	Default	Description
31:24	R/W	160	reg_nr4_sad2bet1_lut12 : // unsigned , default = 160 sad to beta1 for deghost blending, dft = 160
23:16	R/W	192	reg_nr4_sad2bet1_lut13 : // unsigned , default = 192 sad to beta1 for deghost blending, dft = 192
15: 8	R/W	224	reg_nr4_sad2bet1_lut14 : // unsigned , default = 224 sad to beta1 for deghost blending, dft = 224
7: 0	R/W	255	reg_nr4_sad2bet1_lut15 : // unsigned , default = 255 sad to beta1 for deghost blending, dft = 255

NR4_MCNR_SAD2BET2_LUT0 0x2dee

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg_nr4_sad2bet2_lut0 : // unsigned , default = 0 sad to beta2 for snr and mcnr blending, dft = 0
23:16	R/W	1	reg_nr4_sad2bet2_lut1 : // unsigned , default = 1 sad to beta2 for snr and mcnr blending, dft = 2
15: 8	R/W	2	reg_nr4_sad2bet2_lut2 : // unsigned , default = 2 sad to beta2 for snr and mcnr blending, dft = 4
7: 0	R/W	4	reg_nr4_sad2bet2_lut3 : // unsigned , default = 4 sad to beta2 for snr and mcnr blending, dft = 8

NR4_MCNR_SAD2BET2_LUT1 0x2def

Bit(s)	R/W	Default	Description
31:24	R/W	8	reg_nr4_sad2bet2_lut4 : // unsigned , default = 8 sad to beta2 for snr and mcnr blending, dft = 16
23:16	R/W	16	reg_nr4_sad2bet2_lut5 : // unsigned , default = 16 sad to beta2 for snr and mcnr blending, dft = 32
15: 8	R/W	32	reg_nr4_sad2bet2_lut6 : // unsigned , default = 32 sad to beta2 for snr and mcnr blending, dft = 48

7: 0	R/W	48	reg_nr4_sad2bet2_lut7 : // unsigned , default = 48 sad to beta2 for snr and mcnr blending, dft = 64

NR4_MCNR_SAD2BET2_LUT2 0x2df0

Bit(s)	R/W	Default	Description
31:24	R/W	64	reg_nr4_sad2bet2_lut8 : // unsigned , default = 64 sad to beta2 for snr and mcnr blending, dft = 80
23:16	R/W	80	reg_nr4_sad2bet2_lut9 : // unsigned , default = 80 sad to beta2 for snr and mcnr blending, dft = 96
15: 8	R/W	96	reg_nr4_sad2bet2_lut10 : // unsigned , default = 96 sad to beta2 for snr and mcnr blending, dft = 112
7: 0	R/W	112	reg_nr4_sad2bet2_lut11 : // unsigned , default = 112 sad to beta2 for snr and mcnr blending, dft = 128

NR4_MCNR_SAD2BET2_LUT3 0x2df1

Bit(s)	R/W	Default	Description
31:24	R/W	128	reg_nr4_sad2bet2_lut12 : // unsigned , default = 128 sad to beta2 for snr and mcnr blending, dft = 160
23:16	R/W	160	reg_nr4_sad2bet2_lut13 : // unsigned , default = 160 sad to beta2 for snr and mcnr blending, dft = 192
15: 8	R/W	224	reg_nr4_sad2bet2_lut14 : // unsigned , default = 224 sad to beta2 for snr and mcnr blending, dft = 224
7: 0	R/W	255	reg_nr4_sad2bet2_lut15 : // unsigned , default = 255 sad to beta2 for snr and mcnr blending, dft = 255

NR4_MCNR_RO_U_SUM 0x2df2

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_u_sum : // unsigned , default = 0 sum of U of current field/frame

NR4_MCNR_RO_V_SUM 0x2df3

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_v_sum : // unsigned , default = 0 sum of V of current field/frame

NR4_MCNR_RO_GRDU_SUM 0x2df4

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_grdu_sum : // unsigned , default = 0 sum of gradient U of current field/frame

NR4_MCNR_RO_GRDV_SUM 0x2df5

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_grdv_sum : // unsigned , default = 0 sum of gradient V of current field/frame

NR4_TOP_CTRL 0x2dff

Bit(s)	R/W	Default	Description
31:20	R/W	0	reg_gclk_ctrl : // unsigned , default = 0
18	R/W	1	reg_nr4_mcnr_en : // unsigned , default = 1 ncnr enable or bypass, dft = 1
17	R/W	1	reg_nr2_en : // unsigned , default = 1 nr2 enable, dft = 1
16	R/W	1	reg_nr4_en : // unsigned , default = 1 nr4 enable, dft = 1
15	R/W	1	reg_nr2_proc_en : // unsigned , default = 1
14	R/W	1	reg_det3d_en : // unsigned , default = 1
13	R/W	1	di_polar_en : // unsigned , default = 1 do does not have in C
12	R/W	0	reg_cfr_enable : // unsigned , default = 0 0-disable; 1:enable
11: 9	R/W	7	reg_3dnr_enable_l : // unsigned , default = 7 b0: Y b1:U b2:V
8: 6	R/W	7	reg_3dnr_enable_r : // unsigned , default = 7 b0: Y b1:U b2:V
5	R/W	1	reg_nr4_lnbuf_ctrl : // unsigned , default = 1 line buf ctrl for nr4: 0, 3lines, 1, 5lines, dft = 1
4	R/W	0	reg_nr4_snr2_en : // unsigned , default = 0 snr2 enable, 0: use old snr, 1: use new snr2, dft = 1
3	R/W	1	reg_nr4_scene_change_en : // unsigned , default = 1 enable scene change proc. dft = 1
2	R/W	1	nr2_sw_en : // unsigned , default = 1 do does not have in C
0	R/W	0	reg_nr4_scene_change_flg : // unsigned , default = 0 flags for scene change, dft = 0

NR4_MCNR_SAD_GAIN 0x3700

Bit(s)	R/W	Default	Description
24	R/W	0	reg_nr4_bld12vs3_usemaxsad : // unsigned , default = 0 use minsad/maxsad instead of minsad/avgsad to decision if it was texture or flat region, 1: use minsad/maxsad
23:16	R/W	64	reg_nr4_bld12vs3_rate_gain : // unsigned , default = 64 gain to minsad/maxsad or minsad/avgsad before LUT, 64 normalized as "1"
15: 8	R/W	32	reg_nr4_bld1vs2_rate_gain : // unsigned , default = 32 gain to minsad/maxsad or minsad/avgsad before the LUT, 64 normalized as"1"
7: 0	R/W	64	reg_nr4_coefblt_gain : // unsigned , default = 64 gain to final coefblt, normalized 64 as "1"

NR4_MCNR_LPF_CTRL 0x3701

Bit(s)	R/W	Default	Description
30:22	R/W	0	reg_nr4_preflt_alpofst : // signed , default = 0 pre filter alpha ofst
21:16	R/W	16	reg_nr4_preflt_alpgain : // unsigned , default = 16 pre filter alpha gain

15:14	R/W	3	reg_nr4_preflt_alpsel : // unsigned , default = 3 pre filter alpha selection for adaptive blending, 0: mv pointed sad, 1: weighted mv pointed sad, 2or3: coefblt
13: 8	R/W	8	reg_nr4_avgsad_gain : // unsigned , default = 8 gain for avg sad before luts
6	R/W	1	reg_nr4_maxsad_mod : // unsigned , default = 1 max sad select mode, 0: mx2_sad, 1: max sad
5	R/W	1	reg_nr4_minsad_mod : // unsigned , default = 1 min sad select mode, 0: sad with min err, 1: min sad
4	R/W	1	reg_nr4_minmaxsad_lpf : // unsigned , default = 1 mode of lpf for minmaxsad, 0: no LPF, 1: [1 2 1]/4
3	R/W	1	reg_nr4_avgsad_lpf : // unsigned , default = 1 mode of lpf for avgsad, 0: no LPF, 1: [1 2 1]/4
2	R/W	1	reg_nr4_minavgsad_ratio_lpf : // unsigned , default = 1 mode of lpf for minsad/avgsad and zmvsad/avgsad, 0: no LPF, 1: [1 2 1]/4
1	R/W	1	reg_nr4_bldvs_lut_lpf : // unsigned , default = 1 mode of lpf for bld12vs3 and bld1vs2 LUT results, 0: no LPF, 1: [1 2 1]/4
0	R/W	1	reg_nr4_final_coef_lpf : // unsigned , default = 1 mode of lpf for final coef_blt_blend123, 0: no LPF, 1: [1 2 1]/4

NR4_MCNR_BLD_VS3LUT0 0x3702

Bit(s)	R/W	Default	Description
29:24	R/W	0	reg_nr4_bld12vs3_lut0 : // unsigned , default = 0
21:16	R/W	8	reg_nr4_bld12vs3_lut1 : // unsigned , default = 8
13: 8	R/W	10	reg_nr4_bld12vs3_lut2 : // unsigned , default = 10
5: 0	R/W	11	reg_nr4_bld12vs3_lut3 : // unsigned , default = 11

NR4_MCNR_BLD_VS3LUT1 0x3703

Bit(s)	R/W	Default	Description
29:24	R/W	12	reg_nr4_bld12vs3_lut4 : // unsigned , default = 12
21:16	R/W	14	reg_nr4_bld12vs3_lut5 : // unsigned , default = 14
13: 8	R/W	16	reg_nr4_bld12vs3_lut6 : // unsigned , default = 16
5: 0	R/W	24	reg_nr4_bld12vs3_lut7 : // unsigned , default = 24

NR4_MCNR_BLD_VS3LUT2 0x3704

Bit(s)	R/W	Default	Description
29:24	R/W	50	reg_nr4_bld12vs3_lut8 : // unsigned , default = 50
21:16	R/W	58	reg_nr4_bld12vs3_lut9 : // unsigned , default = 58
13: 8	R/W	63	reg_nr4_bld12vs3_lut10 : // unsigned , default = 63

5: 0	R/W	63	reg_nr4_bld12vs3_lut11 : // unsigned , default = 63

NR4_MCNR_BLD_VS2LUT0 0x3705

Bit(s)	R/W	Default	Description
29:24	R/W	63	reg_nr4_bld1vs2_lut0 : // unsigned , default = 63
21:16	R/W	32	reg_nr4_bld1vs2_lut1 : // unsigned , default = 32
13: 8	R/W	16	reg_nr4_bld1vs2_lut2 : // unsigned , default = 16
5: 0	R/W	8	reg_nr4_bld1vs2_lut3 : // unsigned , default = 8

NR4_MCNR_BLD_VS2LUT1 0x3706

Bit(s)	R/W	Default	Description
29:24	R/W	4	reg_nr4_bld1vs2_lut4 : // unsigned , default = 4
21:16	R/W	2	reg_nr4_bld1vs2_lut5 : // unsigned , default = 2
13: 8	R/W	1	reg_nr4_bld1vs2_lut6 : // unsigned , default = 1
5: 0	R/W	0	reg_nr4_bld1vs2_lut7 : // unsigned , default = 0

NR4_COEFBLT_LUT10 0x3707

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_nr4_coefblt_lut10 : // signed , default = -128
23:16	R/W	0x0	reg_nr4_coefblt_lut11 : // signed , default = -128
15: 8	R/W	0x0	reg_nr4_coefblt_lut12 : // signed , default = -126
7: 0	R/W	0x0	reg_nr4_coefblt_lut13 : // signed , default = -124

NR4_COEFBLT_LUT11 0x3708

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_nr4_coefblt_lut14 : // signed , default = -120
23:16	R/W	0x0	reg_nr4_coefblt_lut15 : // signed , default = -110
15: 8	R/W	0x0	reg_nr4_coefblt_lut16 : // signed , default = -100
7: 0	R/W	0x0	reg_nr4_coefblt_lut17 : // signed , default = -90

NR4_COEFBLT_LUT12 0x3709

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_nr4_coefblt_lut18 : // signed , default = -56
23:16	R/W	0x0	reg_nr4_coefblt_lut19 : // signed , default = -32
15: 8	R/W	0x0	reg_nr4_coefblt_lut110 : // signed , default = -64

7: 0	R/W	0x0	reg_nr4_coefblt_lut111 : // signed , default = -128

NR4_COEFBLT_LUT20 0x370a

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_nr4_coefblt_lut20 : // signed , default = -128
23:16	R/W	0x0	reg_nr4_coefblt_lut21 : // signed , default = -120
15: 8	R/W	0x0	reg_nr4_coefblt_lut22 : // signed , default = -112
7: 0	R/W	0x0	reg_nr4_coefblt_lut23 : // signed , default = -104

NR4_COEFBLT_LUT21 0x370b

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_nr4_coefblt_lut24 : // signed , default = -96
23:16	R/W	0x0	reg_nr4_coefblt_lut25 : // signed , default = -88
15: 8	R/W	0x0	reg_nr4_coefblt_lut26 : // signed , default = -76
7: 0	R/W	0x0	reg_nr4_coefblt_lut27 : // signed , default = -64

NR4_COEFBLT_LUT22 0x370c

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_nr4_coefblt_lut28 : // signed , default = -48
23:16	R/W	0x0	reg_nr4_coefblt_lut29 : // signed , default = -32
15: 8	R/W	0x0	reg_nr4_coefblt_lut210 : // signed , default = -64
7: 0	R/W	0x0	reg_nr4_coefblt_lut211 : // signed , default = -108

NR4_COEFBLT_LUT30 0x370d

Bit(s)	R/W	Default	Description
31:24	R/W	8	reg_nr4_coefblt_lut30 : // signed , default = 8
23:16	R/W	16	reg_nr4_coefblt_lut31 : // signed , default = 16
15: 8	R/W	24	reg_nr4_coefblt_lut32 : // signed , default = 24
7: 0	R/W	30	reg_nr4_coefblt_lut33 : // signed , default = 30

NR4_COEFBLT_LUT31 0x370e

Bit(s)	R/W	Default	Description
31:24	R/W	36	reg_nr4_coefblt_lut34 : // signed , default = 36
23:16	R/W	48	reg_nr4_coefblt_lut35 : // signed , default = 48

15: 8	R/W	70	reg_nr4_coefblt_lut36 : // signed , default = 70
7: 0	R/W	96	reg_nr4_coefblt_lut37 : // signed , default = 96

NR4_COEFBLT_LUT32 0x370f

Bit(s)	R/W	Default	Description
31:24	R/W	120	reg_nr4_coefblt_lut38 : // signed , default = 120
23:16	R/W	64	reg_nr4_coefblt_lut39 : // signed , default = 64
15: 8	R/W	16	reg_nr4_coefblt_lut310 : // signed , default = 16
7: 0	R/W	0x0	reg_nr4_coefblt_lut311 : // signed , default = -8

NR4_COEFBLT_CONV 0x3710

Bit(s)	R/W	Default	Description
23:16	R/W	0	reg_nr4_coefblt_convmin : // unsigned , default = 0 minimum of coef. bilateral conversion
15: 8	R/W	255	reg_nr4_coefblt_convmax : // unsigned , default = 255 maximum of coef. bilateral conversion
7: 0	R/W	128	reg_nr4_coefblt_convmid : // unsigned , default = 128 value at midpoint of coef. bilateral conversion

NR4_DBGWIN_YX0 0x3711

Bit(s)	R/W	Default	Description
29:16	R/W	100	reg_nr4_dgbwin_yx0 : // unsigned , default = 100 ystart for debug window
13: 0	R/W	160	reg_nr4_dgbwin_yx1 : // unsigned , default = 160 yend for debug window

NR4_DBGWIN_YX1 0x3712

Bit(s)	R/W	Default	Description
29:16	R/W	200	reg_nr4_dgbwin_yx2 : // unsigned , default = 200 xstart for debug window
13: 0	R/W	300	reg_nr4_dgbwin_yx3 : // unsigned , default = 300 xend for debug window

NR4_NM_X_CFG 0x3713

Bit(s)	R/W	Default	Description
29:16	R/W	8	reg_nr4_nm_xst : // unsigned , default = 8 start for noise meter statistic, dft = 8
13: 0	R/W	711	reg_nr4_nm_xed : // unsigned , default = 711 end for noise meter statistic, dft = HSIZE-8-1;

NR4_NM_Y_CFG 0x3714

Bit(s)	R/W	Default	Description
29:16	R/W	8	reg_nr4_nm_yst : // unsigned , default = 8 start for noise meter statistic, dft = 8;

13: 0	R/W	231	reg_nr4_nm_yed : // unsigned , default = 231 end for noise meter statistic, dft = VSIZE-8-1;

NR4_NM_SAD_THD 0x3715

Bit(s)	R/W	Default	Description
7: 0	R/W	255	reg_nr4_nm_sad_thd : // unsigned , default = 255 threshold for (flat region) sad count, dft = 4

NR4_MCNR_BANDSPLIT_PRAM 0x3716

Bit(s)	R/W	Default	Description
4	R/W	1	reg_nr4_mc_use_bandsplit : // unsigned , default = 1 separate lp and us for mc IIR filter, 0: no BS used; 1: use BS
3	R/W	1	reg_nr4_mc_apply_on_lp : // unsigned , default = 1 use mcnr only on lowpass portion;
2	R/W	1	reg_nr4_mc_apply_on_us : // unsigned , default = 1 use mcnr only on lp complimentary portion;
1: 0	R/W	1	reg_nr4_mc_zmvbs_use_adplpf : // unsigned , default = 1 use adapptive LPF for the zmv pointing data for MCNR, for abs(mvx)<th

NR4_MCNR_ALP1_SGN_COR 0x3717

Bit(s)	R/W	Default	Description
31:24	R/W	10	reg_nr4_mc_aph1_sgn_coring0 : // unsigned , default = 10 coring to cur-pre before do sgn decision
23:16	R/W	7	reg_nr4_mc_aph1_sgn_coring1 : // unsigned , default = 7 coring to cur-pre before do sgn decision
15: 8	R/W	90	reg_nr4_mc_aph1_sgn_core_max0 : // unsigned , default = 90 maximum of coring, default = 30/15
7: 0	R/W	15	reg_nr4_mc_aph1_sgn_core_max1 : // unsigned , default = 15 maximum of coring, default = 30/15

NR4_MCNR_ALP1_SGN_PRAM 0x3718

Bit(s)	R/W	Default	Description
10	R/W	1	reg_nr4_mc_alp1_sgn_half : // unsigned , default = 1 half block sgn sum mode enable, 0: only use 3x5 whole block sum of sgns; 1: use max(sgn_3x5, sqrt(sgn_left+sgn_righ))
9	R/W	1	reg_nr4_mc_alp1_sgn_frczmv : // unsigned , default = 1 force zmv to calculate the sign_sum;
8	R/W	1	reg_nr4_mc_alp1_sgnmvx_mode : // unsigned , default = 1 blend mode of sgnlut and mvxlut blend mode: 0: sgnlut+ mvxlut; 1: max(sgnlut, mvxlut), default =1
7: 4	R/W	4	reg_nr4_mc_aph1_sgn_crate0 : // unsigned , default = 4 rate to var, norm to 16 as 1, default = 2
3: 0	R/W	2	reg_nr4_mc_aph1_sgn_crate1 : // unsigned , default = 2 rate to var, norm to 16 as 1, default = 2

NR4_MCNR_ALP1_MVX_LUT1 0x3719

Bit(s)	R/W	Default	Description
31:28	R/W	14	reg_nr4_mc_alp1_mvx_luty3 : // unsigned , default = 14 alp1 of luma vas mvx(0~7), and alp1 vs mvy(0,1)
27:24	R/W	14	reg_nr4_mc_alp1_mvx_lutc3 : // unsigned , default = 14 alp1 of chrm vas mvx(0~7), and alp1 vs mvy(0,1)
23:20	R/W	12	reg_nr4_mc_alp1_mvx_luty2 : // unsigned , default = 12 alp1 of luma vas mvx(0~7), and alp1 vs mvy(0,1)
19:16	R/W	12	reg_nr4_mc_alp1_mvx_lutc2 : // unsigned , default = 12 alp1 of chrm vas mvx(0~7), and alp1 vs mvy(0,1)
15:12	R/W	5	reg_nr4_mc_alp1_mvx_luty1 : // unsigned , default = 5 alp1 of luma vas mvx(0~7), and alp1 vs mvy(0,1)
11: 8	R/W	5	reg_nr4_mc_alp1_mvx_lutc1 : // unsigned , default = 5 alp1 of chrm vas mvx(0~7), and alp1 vs mvy(0,1)
7: 4	R/W	3	reg_nr4_mc_alp1_mvx_luty0 : // unsigned , default = 3 alp1 of luma vas mvx(0~7), and alp1 vs mvy(0,1)
3: 0	R/W	3	reg_nr4_mc_alp1_mvx_lutc0 : // unsigned , default = 3 alp1 of chrm vas mvx(0~7), and alp1 vs mvy(0,1)

NR4_MCNR_ALP1_MVX_LUT2 0x371a

Bit(s)	R/W	Default	Description
31:28	R/W	15	reg_nr4_mc_alp1_mvx_luty7 : // unsigned , default = 15 alp1 of luma vas mvx(0~7), and alp1 vs mvy(0,1)
27:24	R/W	15	reg_nr4_mc_alp1_mvx_lutc7 : // unsigned , default = 15 alp1 of chrm vas mvx(0~7), and alp1 vs mvy(0,1)
23:20	R/W	15	reg_nr4_mc_alp1_mvx_luty6 : // unsigned , default = 15 alp1 of luma vas mvx(0~7), and alp1 vs mvy(0,1)
19:16	R/W	15	reg_nr4_mc_alp1_mvx_lutc6 : // unsigned , default = 15 alp1 of chrm vas mvx(0~7), and alp1 vs mvy(0,1)
15:12	R/W	15	reg_nr4_mc_alp1_mvx_luty5 : // unsigned , default = 15 alp1 of luma vas mvx(0~7), and alp1 vs mvy(0,1)
11: 8	R/W	15	reg_nr4_mc_alp1_mvx_lutc5 : // unsigned , default = 15 alp1 of chrm vas mvx(0~7), and alp1 vs mvy(0,1)
7: 4	R/W	15	reg_nr4_mc_alp1_mvx_luty4 : // unsigned , default = 15 alp1 of luma vas mvx(0~7), and alp1 vs mvy(0,1)
3: 0	R/W	15	reg_nr4_mc_alp1_mvx_lutc4 : // unsigned , default = 15 alp1 of chrm vas mvx(0~7), and alp1 vs mvy(0,1)

NR4_MCNR_ALP1_MVX_LUT3 0x371b

Bit(s)	R/W	Default	Description

7: 4	R/W	6	reg_nr4_mc_alp1_mvx_luty8 : // unsigned , default = 6 alp1 of luma vas mvx(0~7), and alp1 vs mvy(0,1)
3: 0	R/W	6	reg_nr4_mc_alp1_mvx_lutc8 : // unsigned , default = 6 alp1 of chrm vas mvx(0~7), and alp1 vs mvy(0,1)

NR4_MCNR_ALP1_LP_PRAM 0x371c

Bit(s)	R/W	Default	Description
17:16	R/W	1	reg_nr4_mc_alp1_lp_sel : // unsigned , default = 1 mode for alp1_lp for lp portion IIR, 0: apha1, 1:dc_dif vs ac analysis; 2: gain/ofst of alp1; 3: max of 1/ 2 results
15: 8	R/W	64	reg_nr4_mc_alp1_lp_gain : // unsigned , default = 64 gain to alp1 to get the alp1_lp = alp1*gain/32 + ofset, default =64;
7: 0	R/W	0	reg_nr4_mc_alp1_lp_ofst : // signed , default = 0 offset to alp1 to get the alp1_lp = alp1*gain/32 + ofset, default =10;

NR4_MCNR_ALP1_SGN_LUT1 0x371d

Bit(s)	R/W	Default	Description
31:28	R/W	3	reg_nr4_mc_alp1_sgn_lut0 : // unsigned , default = 3 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
27:24	R/W	3	reg_nr4_mc_alp1_sgn_lut1 : // unsigned , default = 3 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
23:20	R/W	3	reg_nr4_mc_alp1_sgn_lut2 : // unsigned , default = 3 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
19:16	R/W	4	reg_nr4_mc_alp1_sgn_lut3 : // unsigned , default = 4 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
15:12	R/W	5	reg_nr4_mc_alp1_sgn_lut4 : // unsigned , default = 5 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
11: 8	R/W	6	reg_nr4_mc_alp1_sgn_lut5 : // unsigned , default = 6 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
7: 4	R/W	7	reg_nr4_mc_alp1_sgn_lut6 : // unsigned , default = 7 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
3: 0	R/W	8	reg_nr4_mc_alp1_sgn_lut7 : // unsigned , default = 8 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move

NR4_MCNR_ALP1_SGN_LUT2 0x371e

Bit(s)	R/W	Default	Description
31:28	R/W	9	reg_nr4_mc_alp1_sgn_lut8 : // unsigned , default = 9 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
27:24	R/W	10	reg_nr4_mc_alp1_sgn_lut9 : // unsigned , default = 10 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move

23:20	R/W	11	reg_nr4_mc_alp1_sgn_lut10 : // unsigned , default = 11 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
19:16	R/W	12	reg_nr4_mc_alp1_sgn_lut11 : // unsigned , default = 12 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
15:12	R/W	13	reg_nr4_mc_alp1_sgn_lut12 : // unsigned , default = 13 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
11: 8	R/W	14	reg_nr4_mc_alp1_sgn_lut13 : // unsigned , default = 14 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
7: 4	R/W	15	reg_nr4_mc_alp1_sgn_lut14 : // unsigned , default = 15 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move
3: 0	R/W	15	reg_nr4_mc_alp1_sgn_lut15 : // unsigned , default = 15 alp1 vs x=abs|sgn(cur-pre)|, if x is small, less possibility of flat region move

NR4_RO_NM_SAD_SUM 0x371f

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_nm_sad_sum : // unsigned , default = 0 sum of sad, for scene change detectcion, in noise meter

NR4_RO_NM_SAD_CNT 0x3720

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_nm_sad_cnt : // unsigned , default = 0 cnt of sad, for scene change detectcion, in noise meter

NR4_RO_NM_VAR_SUM 0x3721

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_nm_var_sum : // unsigned , default = 0 sum of var, for noise level detection, in noise meter

NR4_RO_NM_VAR_SCNT 0x3722

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_nm_var_cnt : // unsigned , default = 0 cnt of var, for noise level detection, in noise meter

NR4_RO_NM_VAR_MIN_MAX 0x3723

Bit(s)	R/W	Default	Description
21:12	R.O	1023	ro_nr4_nm_min_var : // unsigned , default = 1023 min of var, for noise level detection, in noise meter
9: 0	R.O	0	ro_nr4_nm_max_var : // unsigned , default = 0 max of var, for noise level detection, in noise meter

NR4_RO_NR4_DBGPIX_NUM 0x3724

Bit(s)	R/W	Default	Description

27: 0	R.O	0	ro_nr4_dbgpix_num : // unsigned , default = 0 number of pixels statistic invoved (removed?)

NR4_RO_NR4_BLDVS2_SUM 0x3725

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_bld1vs2_sum : // unsigned , default = 0 sum of blend_1vs2 with the debug window

NR4_BLDVS3_SUM 0x3726

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_bld12vs3_sum : // unsigned , default = 0 sum of blend_12vs3 with the debug window

NR4_COEF12_SUM 0x3727

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_coef12_sum : // signed , default = 0 sum of coef_blt_blend12 with the debug window, under 8 bits precision

NR4_COEF123_SUM 0x3728

Bit(s)	R/W	Default	Description
31: 0	R.O	0	ro_nr4_coef123_sum : // signed , default = 0 sum of coef_final with the debug window, under 8 bits precision

NR_DB_FLT_CTRL 0x3738

Bit(s)	R/W	Default	Description
26	R/W	0	reg_nrdeband_reset1 : // unsigned , default = 0 0 : no reset seed 1: reload chroma seed
25	R/W	0	reg_nrdeband_reset0 : // unsigned , default = 0 0 : no reset seed 1: reload luma seed
24	R/W	0	reg_nrdeband_rgb : // unsigned , default = 0 0 : yuv 1: RGB
23	R/W	1	reg_nrdeband_en11 : // unsigned , default = 1 debanding registers of side lines, [0] for luma, same for below
22	R/W	1	reg_nrdeband_en10 : // unsigned , default = 1 debanding registers of side lines, [1] for chroma, same for below
21	R/W	1	reg_nrdeband_siderand : // unsigned , default = 1 options to use side two lines use the rand, instead of use for the YUV three component of middle line, 0: seed[3]/bandrand[3] for middle line yuv; 1: seed[3]/bandrand[3] for nearby three lines Y;
20	R/W	0	reg_nrdeband_randmode : // unsigned , default = 0 mode of rand noise adding, 0: same noise strength for all difs; else: strenght of noise will not exceed the difs, MIN((pPKReg->reg_nrdeband_bandrand[m]), noise[m])
19:17	R/W	6	reg_nrdeband_bandrand2 : // unsigned , default = 6
15:13	R/W	6	reg_nrdeband_bandrand1 : // unsigned , default = 6

Bit(s)	R/W	Default	Description
11: 9	R/W	6	reg_nrdeband_bandrand0 : // unsigned , default = 6
7	R/W	1	reg_nrdeband_hpxor1 : // unsigned , default = 1 debanding random hp portion xor, [0] for luma
6	R/W	1	reg_nrdeband_hpxor0 : // unsigned , default = 1 debanding random hp portion xor, [1] for chroma
5	R/W	1	reg_nrdeband_en1 : // unsigned , default = 1 debanding registers, for luma
4	R/W	1	reg_nrdeband_en0 : // unsigned , default = 1 debanding registers, for chroma
3: 2	R/W	2	reg_nrdeband_lpf_mode1 : // unsigned , default = 2 lpf mode, 0: 3x3, 1:3x5; 2: 5x5; 3:5x7
1: 0	R/W	2	reg_nrdeband_lpf_mode0 : // unsigned , default = 2 lpf mode, 0: 3x3, 1:3x5; 2: 5x5; 3:5x7

NR_DB_FLT_YC_THRD 0x3739

Bit(s)	R/W	Default	Description
31:28	R/W	9	reg_nrdeband_luma_th3 : // unsigned , default = 9 threshold to |Y-Ylpf|, if < th[0] use lpf
27:24	R/W	7	reg_nrdeband_luma_th2 : // unsigned , default = 7 elseif <th[1] use (lpf*3 + y)/4
23:20	R/W	6	reg_nrdeband_luma_th1 : // unsigned , default = 6 elseif <th[1] use (lpf*3 + y)/4elseif <th[2] (lpf*1 + y)/2
19:16	R/W	5	reg_nrdeband_luma_th0 : // unsigned , default = 5 elseif <th[1] use (lpf*3 + y)/4elseif elseif <th[3] (lpf*1 + 3*y)/4; else
15:12	R/W	9	reg_nrdeband_chrm_th3 : // unsigned , default = 9 threshold to |Y-Ylpf|, if < th[0] use lpf
11: 8	R/W	7	reg_nrdeband_chrm_th2 : // unsigned , default = 7 elseif <th[1] use (lpf*3 + y)/4
7: 4	R/W	6	reg_nrdeband_chrm_th1 : // unsigned , default = 6 elseif <th[1] use (lpf*3 + y)/4elseif <th[2] (lpf*1 + y)/2
3: 0	R/W	5	reg_nrdeband_chrm_th0 : // unsigned , default = 5 elseif <th[1] use (lpf*3 + y)/4elseif elseif

NR_DB_FLT_RANDLUT 0x373a

Bit(s)	R/W	Default	Description
23:21	R/W	1	reg_nrdeband_randslut7 : // unsigned , default = 1 lut0
20:18	R/W	1	reg_nrdeband_randslut6 : // unsigned , default = 1 lut0
17:15	R/W	1	reg_nrdeband_randslut5 : // unsigned , default = 1 lut0
14:12	R/W	1	reg_nrdeband_randslut4 : // unsigned , default = 1 lut0
11: 9	R/W	1	reg_nrdeband_randslut3 : // unsigned , default = 1 lut0

8: 6	R/W	1	reg_nrdeband_randslut2 : // unsigned , default = 1 lut0
5: 3	R/W	1	reg_nrdeband_randslut1 : // unsigned , default = 1 lut0
2: 0	R/W	1	reg_nrdeband_randslut0 : // unsigned , default = 1 lut0

NR_DB_FLT_PXI_THRD 0x373b

Bit(s)	R/W	Default	Description
25:16	R/W	0	reg_nrdeband_yc_th1 : // unsigned , default = 0 to luma/|u/v| for using the denoise
9: 0	R/W	0	reg_nrdeband_yc_th0 : // unsigned , default = 0 to luma/|u/v| for using the denoise

NR_DB_FLT_SEED_Y 0x373c

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438240	reg_nrdeband_seed0 : // unsigned , default = 1621438240 noise adding seed for Y. seed[0]= 0x60a52f20; as default

NR_DB_FLT_SEED_U 0x373d

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438247	reg_nrdeband_seed1 : // unsigned , default = 1621438247 noise adding seed for U. seed[0]= 0x60a52f27; as default

NR_DB_FLT_SEED_V 0x373e

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438242	reg_nrdeband_seed2 : // unsigned , default = 1621438242 noise adding seed for V. seed[0]= 0x60a52f22; as default

NR_DB_FLT_SEED3 0x373f

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438242	reg_nrdeband_seed3 : // unsigned , default = 1621438242 noise adding seed for V. seed[0]= 0x60a52f22; as default

10.2.3.16 VIU Top-Level Registers

VIU_SW_RESET 0x1a01

Bit(s)	R/W	Default	Description
29	R/W	0	di_dsr1to2_reset : // unsigned , default = 0
28	R/W	0	vd2_lbuf_reset : // unsigned , default = 0
27	R/W	0	afbc_dec1_reset : // unsigned , default = 0

Bit(s)	R/W	Default	Description
26	R/W	0	vd2_reset : // unsigned , default = 0
25	R/W	0	vd1_lbuf_reset : // unsigned , default = 0
24	R/W	0	afbc_dec0_reset : // unsigned , default = 0
23	R/W	0	vd1_reset : // unsigned , default = 0
21	R/W	0	osd1_afbcd_reset : // unsigned , default = 0
20	R/W	0	afbc_arb_reg_reset : // unsigned , default = 0
19	R/W	0	afbc_arb_reset : // unsigned , default = 0
17	R/W	0	osd4_reset : // unsigned , default = 0
16	R/W	0	osd3_reset : // unsigned , default = 0
15	R/W	0	osd2_reset : // unsigned , default = 0
14	R/W	0	osd1_reset : // unsigned , default = 0
12	R/W	0	vpp_axi_reset : // unsigned , default = 0
8	R/W	0	osd24bld_reset : // unsigned , default = 0
7	R/W	0	osd13bld_reset : // unsigned , default = 0
6	R/W	0	prime_reset : // unsigned , default = 0
5	R/W	0	hist_spl_reset : // unsigned , default = 0
4	R/W	0	ldim_stts_reset : // unsigned , default = 0
3	R/W	0	dolby1b_reset : // unsigned , default = 0
2	R/W	0	dolby1a_reset : // unsigned , default = 0
1	R/W	0	dolby0_reset : // unsigned , default = 0
0	R/W	0	vpp_reset : // unsigned , default = 0

VIU_SW_RESET0 0x1a02

Bit(s)	R/W	Default	Description
2	R/W	0	software : reset for mcinford_mif // unsigned , default = 0
1	R/W	0	software : reset for mcinfowr_mif // unsigned , default = 0
0	R/W	0	software : reset for mcvecrd_mif // unsigned , default = 0

VIU_SECURE_REG 0x1a04

Bit(s)	R/W	Default	Description
19:18	R/W	0	prebld_en : // unsigned , default = 0

17:16	R/W	0	postbld_en : // unsigned , default = 0
9	R/W	0	matrxi : probe // unsigned , default = 0
8	R/W	0	dolby_core3 : // unsigned , default = 0
7	R/W	0	dolby_graphic : // unsigned , default = 0
6	R/W	0	dolby_video : // unsigned , default = 0
5	R/W	0	primel : // unsigned , default = 0

DOLBY_INT_STAT 0x1a05

Bit(s)	R/W	Default	Description
3:0	R/W	0	dolby_int_state : // unsigned , default = 0

VIU_MISC_CTRL0 0x1a06

Bit(s)	R/W	Default	Description
8	R/W	0	vsync_int_ctrl : default = 0
0	R/W	0	scan_reg : default = 0

VIU_MISC_CTRL1 0x1a07

Bit(s)	R/W	Default	Description
31:0	R/W	0	viu_misc_ctrl1 : // unsigned , default = 0

VIU_SECURE_REG1 0x1a08

Bit(s)	R/W	Default	Description
31:0	R/W	0	viu_secure_reg1 : // unsigned , default = 0

VIU_SECURE_REG2 0x1a09

Bit(s)	R/W	Default	Description
31:0	R/W	0	viu_secure_reg2 : // unsigned , default = 0

VD1_AFBCD0_MISC_CTRL 0x1a0a

Bit(s)	R/W	Default	Description
21:20	R/W	0	vd1_go_field_sel : // unsigned , default = 0 0: go_file 1: go_field_post 2: go_filed_pre
16	R/W	0	vd1_lbuf_ram_sel : // unsigned , default = 0 0: NO share ram 1: afbc0 share ext ram
15:14	R/W	0	vd1_in_mux_sel : // unsigned , default = 0 0: vd1 to dobly 2: vd1 to primel
13:12	R/W	0	vd1_axi_sel : // unsigned , default = 0 0: afbc0 mif to axi 1: vd1 mif to axi

11	R/W	0	afbc0_osd3_mux_vd1 : // unsigned , default = 0 0: afbc0 to vd1 1: osd3 to vd1
10	R/W	0	afbc_vd1_sel : // unsigned , default = 0 0: vd1_mif to vpp 1: afbc0_mif
9	R/W	0	afbc0_mux_vpp_mad : // unsigned , default = 0 0: afbc0 to vpp 1: afbc0 to di
8	R/W	0	di_mif0_en : // unsigned , default = 0 0: select mif to vpp 1: mif to di
7:0	R/W	0	afbc0_gclk_ctrl : // unsigned , default = 0

VD2_AFBCD1_MISC_CTRL 0x1a0b

Bit(s)	R/W	Default	Description
31:24	R/W	0	vd2_osd_mux_alpha : // unsigned , default = 0 config vd2_osd alpha value.
21:20	R/W	0	vd1_go_field_sel : // unsigned , default = 0 0: go_file 1: go_field_post 2: go_filed_pre
16	R/W	0	vd2_lbuf_ram_sel : // unsigned , default = 0 0: NO share ram 1: afbc1 share ext ram
15:14	R/W	0	vd2_in_mux_sel : // unsigned , default = 0 0: afbc2_vd2 to vd2 2: osd4 to vd2
13:12	R/W	0	vd2_axi_sel : // unsigned , default = 0 0: afbc1 mif to axi 1: vd2 mif to axi
11	R/W	0	afbc1_osd4_mux_vd1 : // unsigned , default = 0 0: afbc1 to vd1 1: osd4 to vd2
10	R/W	0	afbc_vd2_sel : // unsigned , default = 0 0: vd2_mif to vpp 1: afbc1_mif
9	R/W	0	afbc1_mux_vpp_mad : // unsigned , default = 0 0: afbc1 to vpp 1: afbc1 to di
8	R/W	0	afbc_2mad_out_sel : // unsigned , default = 0 0: select mif to vpp 1: mif to di
7:0	R/W	0	afbc0_gclk_ctrl : // unsigned , default = 0

DOLBY_PATH_CTRL 0x1a0c

Bit(s)	R/W	Default	Description
7	R/W	0	vpp_osd2_ext_mod : // unsigned , default = 0 // 0: datax4 1: datax1
6	R/W	0	vpp_osd1_ext_mod : // unsigned , default = 0 // 0: datax4 1: datax1
5	R/W	0	dolby1_vd2_ext_mod : // unsigned , default = 0 // 0: datax4 1: datax1
4	R/W	0	dolby0_vd1_ext_mod : // unsigned , default = 0 // 0: datax4 1: datax1
3:0	R/W	0	dolby_bypass_en : // unsigned , default = 0

WR_BACK_MISC_CTRL 0x1a0d

Bit(s)	R/W	Default	Description
1	R/W	0	wrbak_chan1_hsyn_en : // unsigned , default = 0 vd1 wrbak hsync enable
0	R/W	0	wrbak_chan0_hsyn_en : // unsigned , default = 0 vd0 wrbak hsync enable

OSD_PATH_MISC_CTRL 0x1a0e

Bit(s)	R/W	Default	Description

7:4	R/W	0	osd_axi_sel : // unsigned , default = 0
3	R/W	0	osd4_in_mux_sel : // unsigned , default = 0 0: osd4 to vpp 1: vd2_osd4 to vpp
2	R/W	0	osd4_mux_vpp_vd2 : // unsigned , default = 0 0: osd4 to vpp 1: osd4 to vd2
1	R/W	0	osd3_in_mux_sel : // unsigned , default = 0 0: osd3 to vpp 1: vd2_osd3 to vpp
0	R/W	0	osd3_mux_vpp_vd1 : // unsigned , default = 0 0: osd3 to vpp 1: osd3 to vd1

MALI_AFBCD_TOP_CTRL 0x1a0f

Bit(s)	R/W	Default	Description
31	R.O	0	ro_mali_dec_active : // unsigned , default = 0
19:12	R/W	0	gclk_ctrl_osd1_afbcd : // unsigned , default = 0 mail afbc gate clock
8	R/W	1	axim_data_128b : // unsigned , default = 1 0: axi bus width=64 1: 128bit
6:4	R/W	7	axim_max_len : // unsigned , default = 7
2:0	R/W	5	axim_outstanding_trans : // unsigned , default =5

VIU_GCLK_CTRL 0x1a4f

Bit(s)	R/W	Default	Description
7:0	R/W	0	viu_gclk_ctrl : // unsigned , default =0

10.2.3.17 VD1 Path vd_rmem_if0 Registers

VD1_IF0_GEN_REG 0x3200

Bit(s)	R/W	Default	Description
31	R/W	0	ENABLE_FREE_CLK. 0: Gated clock for power saving 1: Free-running clock to drive logic
30	R/W	0	SW_RESET: Write 1 to this bit to generate a pulse to reset everything except registers.
29	R/W	0	RESET_ON_GO_FIELD: Define whether to reset state machines on go_field pulse. 0: No reset on go_field 1: go_field reset everything except registers
28	R/W	0	URGENT_CHROMA: Set urgent level for chroma fifo request from DDR. 0: Non urgent 1: Urgent
27	R/W	0	URGENT_LUMA: Set urgent level for luma fifo request from DDR. 0: Non urgent 1: Urgent

Bit(s)	R/W	Default	Description
26	R/W	0	Chroma_end_at_last line: For chroma line, similar to luma_end_at_last_line, as below. Not used if data are stored together in one canvas.
25	R/W	0	Luma_end_at_last_line: Control whether continue outputting luma line past last line. 0: Repeat the last line or dummy pixels, after past the last line 1: Stop outputting data, once past the last line.
24-19	R/W	4	Hold_lines: After go_field, the number of lines to hold before the module is enabled.
18	R/W	0	LAST_LINE: This bit controls whether we simply repeat the last line or we push dummy pixels. '1' tells the state-machines to repeat the last line using the dummy pixels defined in the register below. '0' indicates that the state-machine should re-read the last line of real data.
17	R	0	Busy status of the state-machines. '1' = busy, '0' = idle
16	R/W	0	DEMUX_MODE: 0 = 4:2:2, 1 = RGB (24-bit). This value is used to control the demuxing logic when the picture is stored together. When a picture is stored together, the data is read into a single FIFO (the Y FIFO) and must be demultiplexed into the "drain" outputs. In the case of 4:2:2 the data is assumed to be stored in memory in 16-bit chunks: <YCb><YCr><YCb><YCr>,… the Y, Cb and Cr 8-bit values are pulled from the single Y-FIFO and sent out in pairs. This value is only valid when the picture is stored together. If the picture is separated into different canvases, then this bit field is ignored.
15-14	R/W	0	BYTES_PER_PIXEL: This value is used to determine how many bytes are associated with each pixel. 0: This value should be used if the image is stored separately (e.g. RGB or Y, Cb, Cr). 1: This value should be used if the data is 4:2:2 data stored together. In this case each pixel , YCb or YCr, is 16-bits (two bytes). 2: This value should be used if the RGB (24-bit) data is stored together. 3: reserved for future use (alpha RGB).
13-12	R/W	0	DDR_BURST_SIZE_CR: This value is used to control the DDR burst request size for the Cr FIFO. 0: Maximum burst = 24 64-bit values 1: Maximum burst = 32 64-bit values 2: Maximum burst = 48 64-bit values 3: Maximum burst = 64 64-bit values
11-10	R/W	0	DDR_BURST_SIZE_CB: This value is used to control the DDR burst request size for the Cb FIFO. 0: Maximum burst = 24 64-bit values 1: Maximum burst = 32 64-bit values 2: Maximum burst = 48 64-bit values 3: Maximum burst = 64 64-bit values

Bit(s)	R/W	Default	Description
9-8	R/W	0	DDR_BURST_SIZE_Y: This value is used to control the DDR burst request size for the Y FIFO. 0: Maximum burst = 24 64-bit values 1: Maximum burst = 32 64-bit values 2: Maximum burst = 48 64-bit values 3: Maximum burst = 64 64-bit values
7	R/W	0	MANUAL_START_FRAME: non-latching bit that can be used to simulate the go_field signal for simulation.
6	R/W	0	CHRO_RPT_LASTL_CTRL: This bit controls whether to allow VPP's chroma-repeat request. 0: Chroma-repeat pulses from VPP are ignored 1: Chroma-repeat pulses from VPP are used.
5	R/W	0	Unused
4	R/W	0	LITTLE_ENDIAN: This bit defines the endianess of the memory data . 0: Pixel data are stored big-endian in memory 1: Pixel data are stored little-endian in memory
3	R/W	0	Chroma_hz_avg: For chroma line output control, similar to luma_hz_avg, as below. Not used if data are stored together in one canvas.
2	R/W	0	Luma_hz_avg: Enable output half amount of data per line to save bandwidth. 0: Output every pixel per line 1: Output half line, each data averaged between every 2 pixels Note: For 4:2:2 mode data stored together in one canvas, only do averaging over luma data.
1	R/W	0	SEPARATE_EN: Set this bit to 1 if the image is in separate canvas locations.
0	R/W	0	ENABLE: This bit is set to 1 to enable the FIFOs and other logic. This bit can be set to 0 to cleanup and put the logic into an IDLE state.

VD1_IF0_CANVAS0 – Picture 0 0x3201

Bit(s)	R/W	Default	Description
31-24	R/W	0	unused
23-16	R/W	0	CANVAS0_ADDR2: Canvas table address for picture 0 for component 2 (Cr FIFO). This value is ignored when the picture is stored together
15-8	R/W	0	CANVAS0_ADDR1: Canvas table address for picture 0 for component 1 (Cb FIFO). This value is ignored when the picture is stored together
7-0	R/W	0	CANVAS0_ADDR0: Canvas table address for picture 0 for component 0 (Y FIFO).

VD1_IF0_CANVAS1 – Picture 1 0x3202

Bit(s)	R/W	Default	Description
31-24	R/W	0	unused

23-16	R/W	0	CANVAS1_ADDR2: Canvas table address for picture 1 for component 2 (Cr FIFO). This value is ignored when the picture is stored together
15-8	R/W	0	CANVAS1_ADDR1: Canvas table address for picture 1 for component 1 (Cb FIFO). This value is ignored when the picture is stored together
7-0	R/W	0	CANVAS1_ADDR0: Canvas table address for picture 1 for component 0 (Y FIFO).

VD1_IF0_LUMA_X0 – Picture 0 0x3203

Bit(s)	R/W	Default	Description
31	R/W	0	Unused
30-16	R/W	0	LUMA_X_END0: Picture 0, luma X end value
15	R/W	0	Unused
14-0	R/W	0	LUMA_X_START0: Picture 0, luma X start value

VD1_IF0_LUMA_Y0 – Picture 0 0x3204

Bit(s)	R/W	Default	Description
31-29	R/W	0	Unused
28-16	R/W	0	LUMA_Y_END0: Picture 0, luma Y end value
15-13	R/W	0	Unused
12-0	R/W	0	LUMA_Y_START0: Picture 0, luma Y start value

VD1_IF0_CHROMA_X0 – Picture 0 0x3205

Bit(s)	R/W	Default	Description
31	R/W	0	Unused
30-16	R/W	0	CHROMA_X_END0: Picture 0, chroma X end value. This value is only used when the picture is not stored tegether.
15	R/W	0	Unused
14-0	R/W	0	CHROMA _X_START0: Picture 0, chroma X start value. This value is only used when the picture is not stored tegether.

VD1_IF0_CHROMA_Y0 – Picture 0 0x3206

Bit(s)	R/W	Default	Description
31-29	R/W	0	Unused
28-16	R/W	0	CHROMA _Y_END0: Picture 0, chroma Y end value. This value is only used when the picture is not stored tegether.
15-13	R/W	0	Unused
12-0	R/W	0	CHROMA _Y_START0: Picture 0, chroma Y start value. This value is only used when the picture is not stored tegether.

VD1_IF0_LUMA_X1 – Picture 1 0x3207

Bit(s)	R/W	Default	Description
31	R/W	0	Unused
30-16	R/W	0	LUMA_X_END1: Picture 1, luma X end value
15	R/W	0	Unused
14-0	R/W	0	LUMA_X_START1: Picture 1, luma X start value

VD1_IF0_LUMA_Y1 – Picture 1 0x3208

Bit(s)	R/W	Default	Description
31-29	R/W	0	Unused
28-16	R/W	0	LUMA_Y_END1: Picture 1, luma Y end value
15-13	R/W	0	Unused
12-0	R/W	0	LUMA_Y_START1: Picture 1, luma Y start value

VD1_IF0_CHROMA_X1 – Picture 1 0x3209

Bit(s)	R/W	Default	Description
31	R/W	0	Unused
30-16	R/W	0	CHROMA_X_END1: Picture 1, chroma X end value. This value is only used when the picture is not stored together.
15	R/W	0	Unused
14-0	R/W	0	CHROMA _X_START1: Picture 1, chroma X start value. This value is only used when the picture is not stored together.

VD1_IF0_ CHROMA_Y1 – Picture 1 0x320A

Bit(s)	R/W	Default	Description
31-29	R/W	0	Unused
28-16	R/W	0	CHROMA _Y_END1: Picture 1, chroma Y end value. This value is only used when the picture is not stored together.
15-13	R/W	0	Unused
12-0	R/W	0	CHROMA _Y_START1: Picture 1, chroma Y start value. This value is only used when the picture is not stored together.

VD1_IF0_REPEAT_LOOP – Pictures 0 and 1 0x320B

Bit(s)	R/W	Default	Description

31-24	R/W	0	CHROMA_RPT_LOOP1: Repeat loop for Picture 1. Bits[6:4] = start loop pointer, bits [2:0] = end loop pointer. Bits [7] and [3] are ignored.
23-16	R/W	0	LUMA_RPT_LOOP1: Repeat loop for Picture 1. Bits[6:4] = start loop pointer, bits [2:0] = end loop pointer. Bits [7] and [3] are ignored.
15-8	R/W	0	CHROMA_RPT_LOOP0: Repeat loop for Picture 0. Bits[6:4] = start loop pointer, bits [2:0] = end loop pointer. Bits [7] and [3] are ignored.
7-0	R/W	0	LUMA_RPT_LOOP0: Repeat loop for Picture 0. Bits[6:4] = start loop pointer, bits [2:0] = end loop pointer. Bits [7] and [3] are ignored.

VD1_IF0_LUMA0_RPT_PAT – Picture 0 LUMA repeat pattern 0x320C

Bit(s)	R/W	Default	Description
31-0	R/W	0	Luma repeat/skip pattern for picture 0

Bits	Pattern Index	Pattern description
31-28	7	Repeat/skip pattern: Bit[3] = 0 indicates repeat. Bit[3] = 1 indicates either skip, or output this line and then skip. How to interpret this bit depends on the value of the previous pattern's Bit[3]. If previous Bit[3]=0, then skip; If previous Bit[3]=1, then output this line and then skip. Bits[2:0] indicate the skip / repeat count. Blow is an example of consecutive patterns, the start line is line 0: {0010} Repeat this line (line 0) two more times for a total of three line reads. Proceed to next line (line 1). {0000} Don't repeat this line (line 1). This line will be read just once. Proceed to next line (line 2). {1000} Skip one line (line 2) to get to the next line (line 3). The skip implies that the next line (line 3) should be read at least once. {1011} Read this line (line 3) once, and then skip the next four lines to get to the next line (line 8). The skip implies that the next line (line 8) should be read at least once. {0100} Repeat this line (line 8) four more times for a total of five line read. Proceed to next line (line 9). {1001} Skip two lines to get to the next line (line 11). The skip implies that the next line (line 11) should be read at least once.
27-24	6	See pattern definition above.
23-20	5	See pattern definition above.
19-16	4	See pattern definition above.
15-12	3	See pattern definition above.
11-8	2	See pattern definition above.
7-4	1	See pattern definition above.
3-0	0	See pattern definition above.

VD1_IF0_CHROMA0_RPT_PAT – Picture 0 CHROMA repeat pattern 0x320D

Bit(s)	R/W	Default	Description
31-0	R/W	0	Chroma repeat/skip pattern for picture 0. See picture 0 luma pattern for description. This value is only used when the picture is not stored together.

VD1_IF0_LUMA1_RPT_PAT – Picture 1 LUMA repeat pattern 0x320E

Bit(s)	R/W	Default	Description
31-0	R/W	0	Luma repeat/skip pattern for picture 1. See picture 0 luma pattern for description.

VD1_IF0_CHROMA1_RPT_PAT – Picture 1 CHROMA repeat pattern 0x320F

Bit(s)	R/W	Default	Description
31-0	R/W	0	Chroma repeat/skip pattern for picture 1. See picture 0 luma pattern for description. This value is only used when the picture is not stored together.

VD1_IF0_LUMA_PSEL – Picture 0 and 1's LUMA 0x3210

Bit(s)	R/W	Default	Description
31-28	R/W	0	unused
27-26	R/W	0	Luma_psel_mode: controls whether it's single-picture or two-picture mode. {00} Only picture 0 is used. Ignore settings defined in Luma_psel_last_line, Luma_psel_pattern and Luma_psel_loop. {01} Only picture 1 is used. Ignore settings defined in Luma_psel_last_line, Luma_psel_pattern and Luma_psel_loop. {1x} Two-picture mode.
25-24	R/W	0	Luma_psel_last_line: select which picture's last line to output, during repeat last line mode. Bit[0]=0, when picture 0 past the last line, use picture 0's last line during repeat last line mode; Bit[0]=1, when picture 0 past the last line, use picture 1's last line during repeat last line mode; Bit[1]=0, when picture 1 past the last line, use picture 0's last line during repeat last line mode; Bit[1]=1, when picture 1 past the last line, use picture 1's last line during repeat last line mode.
23-8	R/W	0	Luma_psel_pattern. If the value of the bit pointed by the loop pointer is 0, output picture 0's luma line, if the bit value is 1, output picture 1's luma line.
7-4	R/W	0	Luma_psel_loop start pointer.
3-0	R/W	0	Luma_psel_loop end pointer.

VD1_IF0_CHROMA_PSEL – Picture 0 and 1's CHROMA 0x3211

Bit(s)	R/W	Default	Description
31-28	R/W	0	unused

27-26	R/W	0	Chroma_psel_mode: see luma_psel_mode. This value is only used when the picture is not stored together.
25-24	R/W	0	Chroma_psel_last_line: See luma_psel_last_line. This value is only used when the picture is not stored together.
23-8	R/W	0	Chroma_psel_pattern. If the value of the bit pointed by the loop pointer is 0, output picture 0's chroma line, if the bit value is 1, output picture 1's chroma line. This value is only used when the picture is not stored together.
7-4	R/W	0	Chroma_psel_loop start pointer. This value is only used when the picture is not stored together.
3-0	R/W	0	Chroma_psel_loop end pointer. This value is only used when the picture is not stored together.

VD1_IF0_DUMMY_PIXEL 0x3212

Bit(s)	R/W	Default	Description
31-24	R/W	0x00	Y or R dummy pixel value ,8bit
23-16	R/W	0x80	Cb or G dummy pixel value ,8bit
15-8	R/W	0x80	Cr or B dummy pixel value ,8bit
7-0	R/W	0	unused

VD1_RANGE_MAP_Y 0x321A

VD1_RANGE_MAP_CB 0x321B

VD1_RANGE_MAP_CR 0x321C

Output data range conversion function:

Y[n] = clip(Round(((Y[n]+DIN_OFFSET) * RANGE_MAP_COEF) / (1<<RANGE_MAP_SR)) +DOUT_OFFSET);

To perform VC-1 range reduction, set the following:

DIN_OFFSET = 0x180 = -128;

RANGE_MAP_COEF = RANGE_MAPY + 9

RANGE_MAP_SR = 3

DOUT_OFFSET = 0x080 = 128

To get the equivalent function:

Y[n] = clip((((Y[n]-128) * (RANGE_MAPY + 9) + 4) >> 3) +128);

Bit(s)	R/W	Default	Description
31-23	R/W	0	DIN_OFFSET
22-15	R/W	0	RANGE_MAP_COEF
14	R/W	0	unused
13-10	R/W	0	RANGE_MAP_SR
9-1	R/W	0	DOUT_OFFSET

0	R/W	0	RANGE_MAP_EN

VD1_IF0_GEN_REG2 0x321D

Bit(s)	R/W	Default	Description
31-2	R/W	0	unused
1-0	R/W	0	COLOR_MAP: Define color map for NV12 or NV21 mode. Only applicable when VD1_IF0_GEN_REG.SEPARATE_EN = 1. 0: NOT NV12 or NV21; 1: NV12 (CbCr); 2: NV21 (CrCb).

VD1_IF0_GEN_REG3 0x3216

Bit(s)	R/W	Default	Description
11-10	R/W	0	cntl_dbg_mode
9-8	R/W	0	cntl_bits_mode : 0->8bit 1->10bit 422 2->10bit 444
6-4	R/W	3	cntl_blk_len
2-1	R/W	1	cntl_burst_len
0	R/W	1	cntl_64bit_rev

VIU_VD1_FMT_CTRL 0x3218

Bit(s)	R/W	Default	Description
31	R/W	0	gate_clk_en. 0=No clock gating, free-running; 1=Enable clock gating for power saving.
30	R/W	0	soft_rst. If true, reset formatters.
29	R/W	0	unused
28	R/W	0	if true, horizontal formatter use repeating to generete pixel, otherwise use bilinear interpolation
27-24	R/W	0	horizontal formatter initial phase
23	R/W	0	horizontal formatter repeat pixel 0 enable
22-21	R/W	0	horizontal Y/C ratio, 00: 1:1, 01: 2:1, 10: 4:1
20	R/W	0	horizontal formatter enable
19	R/W	0	if true, always use phase0 while vertical formater, meaning always repeat data, no interpolation
18	R/W	0	if true, disable vertical formatter chroma repeat last line
17	R/W	0	veritcal formatter dont need repeat line on phase0, 1: enable, 0: disable
16	R/W	0	veritcal formatter repeat line 0 enable

Bit(s)	R/W	Default	Description
15-12	R/W	0	vertical formatter skip line num at the beginning
11-8	R/W	0	vertical formatter initial phase
7-1	R/W	0	vertical formatter phase step (3.4)
0	R/W	0	vertical formatter enable

VIU_VD1_FMT_W 0x3219

Bit(s)	R/W	Default	Description
27-16	R/W	0	horizontal formatter width
11-0	R/W	0	vertical formatter width

10.2.3.18 VD2 Path vd_rmem_if0 Registers

VD2_IF0_GEN_REG 0x3220

Same as VD1_IF0_GEN_REG

VD2_IF0_CANVAS0 0x3221

Same as VD1_IF0_CANVAS0

VD2_IF0_CANVAS1 0x3222

Same as VD1_IF0_CANVAS1

VD2_IF0_LUMA_X0 0x3223

Same as VD1_IF0_LUMA_X0

VD2_IF0_LUMA_Y0 0x3224

Same as VD1_IF0_LUMA_Y0

VD2_IF0_CHROMA_X0 0x3225

Same as VD1_IF0_CHROMA_X0

VD2_IF0_CHROMA_Y0 0x3226

Same as VD1_IF0_CHROMA_Y0

VD2_IF0_LUMA_X1 0x3227

Same as VD1_IF0_LUMA_X1

VD2_IF0_LUMA_Y1 0x3228

Same as VD1_IF0_LUMA_Y1

VD2_IF0_CHROMA_X1 0x3229

Same as VD1_IF0_CHROMA_X1

VD2_IF0_CHROMA_Y1 0x322A

Same as VD1_IF0_CHROMA_Y1

VD2_IF0_RPT_LOOP 0x322B

Same as VD1_IF0_RPT_LOOP

VD2_IF0_LUMA0_RPT_PAT 0x322C

Same as VD1_IF0_LUMA0_RPT_PAT

VD2_IF0_CHROMA0_RPT_PAT 0x322D

Same as VD1_IF0_CHROMA0_RPT_PAT

VD2_IF0_LUMA1_RPT_PAT 0x322E

Same as VD1_IF0_LUMA1_RPT_PAT

VD2_IF0_CHROMA1_RPT_PAT 0x322F

Same as VD1_IF0_CHROMA1_RPT_PAT

VD2_IF0_LUMA_PSEL 0x3230

Same as VD1_IF0_LUMA_PSEL

VD2_IF0_CHROMA_PSEL 0x3231

Same as VD1_IF0_CHROMA_PSEL

VD2_IF0_DUMMY_PIXEL 0x3232

Same as VD1_IF0_DUMMY_PIXEL

VD2_RANGE_MAP_Y 0x323A

Same as VD1_RANGE_MAP_Y

VD2_RANGE_MAP_CB 0x323B

Same as VD1_RANGE_MAP_CB

VD2_RANGE_MAP_CR 0x323C

Same as VD1_RANGE_MAP_CR

VD2_IF0_GEN_REG2 0x323D

Same as VD1_IF0_GEN_REG2

VD2_IF0_GEN_REG3 0x3236

Same as VD1_IF0_GEN_REG3

VIU_VD2_FMT_CTRL 0x3238

Bit(s)	R/W	Default	Description
31	R/W	0	gate_clk_en. 0=No clock gating, free-running; 1=Enable clock gating for power saving.
30	R/W	0	soft_rst. If true, reset formatters.
28	R/W	0	if true, horizontal formatter use repeating to generete pixel, otherwise use bilinear interpolation
27-24	R/W	0	horizontal formatter initial phase
23	R/W	0	horizontal formatter repeat pixel 0 enable
22-21	R/W	0	horizontal Y/C ratio, 00: 1:1, 01: 2:1, 10: 4:1
20	R/W	0	horizontal formatter enable
19	R/W	0	if true, always use phase0 while vertical formater, meaning always repeat data, no interpolation

18	R/W	0	if true, disable vertical formatter chroma repeat last line
17	R/W	0	veritcal formatter dont need repeat line on phase0, 1: enable, 0: disable
16	R/W	0	veritcal formatter repeat line 0 enable
15-12	R/W	0	vertical formatter skip line num at the beginning
11-8	R/W	0	vertical formatter initial phase
7-1	R/W	0	vertical formatter phase step (3.4)
0	R/W	0	vertical formatter enable

VIU_VD2_FMT_W 0x3239

Bit(s)	R/W	Default	Description
27-16	R/W	0	horizontal formatter width
11-0	R/W	0	vertical formatter width

10.2.3.19 Osd_blend Registers

VIU_OSD_BLEND_CTRL 0x37b0

Bit(s)	R/W	Default	Description
31:29	R/W	0x0	hold_lines : //unsigned , default = 3'h0, hold_lines(line) after go_field ,module active
28:27	R/W	0x3	blend2_premult_en : //unsigned , default = 2'h3, blend2 input premult label 1:premult input 0:unpremult input
26	R/W	0x1	din0_byp_blend : //unsigned , default = 1'h1, blend_din0 bypass to dout0 1:bypass 0:blend_din0 input to blend0
25	R/W	0x1	din2_osd_sel : //unsigned , default = 1'h1, blend1_dout bypass to blend2 1:blend1_dout to blend2 0:blend1_dout to dout1
24	R/W	0x1	din3_osd_sel : //unsigned , default = 1'h1, blend1_din3 bypass to dout1 1:bypass 0:blend_din3 input to blend1
23:20	R/W	0x5	blend_din_en : //unsigned , default = 4'h5, blend enable bits ,four bits for four input
19:16	R/W	0x0	din_premult_en : //unsigned , default = 4'h0, input premult label bits,four bits for four input
15:0	R/W	0x2341	din_reoder_sel : //unsigned , default = 16'h2341,osd_blend input reorder exp :din_reoder_sel[3:0] = 1 ,blend_din0 select osd1 din_reoder_sel [3:0] = 2 ,blend_din0 select osd2 din_reoder_sel [3:0] = 3 ,blend_din0 select osd3 din_reoder_sel [3:0] = else,blend_din0 no src din_reoder_sel [7:4] fot blend_din1

VIU_OSD_BLEND_CTRL1 0x37c0

Bit(s)	R/W	Default	Description
17:16	R/W	0	reg_alp1_mapping_mode : //unsigned , default = 0 , osd_blend dout1 alpha dividor mode 8bit alpha:set 0 9bit alpha:set 3

14:13	R/W	0	reg_div1_gclk_en : //unsigned , default = 0 , osd_blend dout1 alpha dividor gclk_en
12	R/W	0	reg_div1_alpha_en : //unsigned , default = 0 , osd_blend dout1 alpha dividor gclk_en enable
9:8	R/W	0	osdbld_gclk_ctrl : //unsigned , default = 0 , osdbld_gclk_ctrl
5:4	R/W	0	reg_alp_mapping_mode : //unsigned , default = 0 , osd_blend dout0 alpha dividor mode 8bit alpha:set 0 9bit alpha:set 3
2:1	R/W	0	reg_div_gclk_en : //unsigned , default = 0 , osd_blend dout0 alpha dividor gclk_en
0	R/W	0	reg_div_alpha_en : //unsigned , default = 0 , osd_blend dout0 alpha dividor gclk_en enable

VIU_OSD_BLEND_DIN0_SCOPE_H 0x37b1

Bit(s)	R/W	Default	Description
28:16	R/W	0x2d0	bld_din0_h_end : ///unsigned , default = 13'h2d0,blend_din0 h_end
12:0	R/W	0x0	bld_din0_h_start : ///unsigned , default = 13'h0 ,blend_din0 h_start

VIU_OSD_BLEND_DIN0_SCOPE_V 0x37b2

Bit(s)	R/W	Default	Description
28:16	R/W	0x1e0	bld_din0_v_end : ///unsigned , default = 13'h1e0,blend_din0 v_end
12:0	R/W	0x0	bld_din0_v_start : ///unsigned , default = 13'h0,blend_din0 v_start

VIU_OSD_BLEND_DIN1_SCOPE_H 0x37b3

Bit(s)	R/W	Default	Description
28:16	R/W	0x2d0	bld_din1_h_end : ///unsigned , default = 13'h2d0
12:0	R/W	0x0	bld_din1_h_start : ///unsigned , default = 13'h0

VIU_OSD_BLEND_DIN1_SCOPE_V 0x37b4

Bit(s)	R/W	Default	Description
28:16	R/W	0x1e0	bld_din1_v_end : ///unsigned , default = 13'h1e0
12:0	R/W	0x0	bld_din1_v_start : ///unsigned , default = 13'h0

VIU_OSD_BLEND_DIN2_SCOPE_H 0x37b5

Bit(s)	R/W	Default	Description
28:16	R/W	0x2d0	bld_din2_h_end : ///unsigned , default = 13'h2d0
12:0	R/W	0x0	bld_din2_h_start : ///unsigned , default = 13'h0

VIU_OSD_BLEND_DIN2_SCOPE_V 0x37b6

Bit(s)	R/W	Default	Description
28:16	R/W	0x1e0	bld_din2_v_end : ///unsigned , default = 13'h1e0
12:0	R/W	0x0	bld_din2_v_start : ///unsigned , default = 13'h0

VIU_OSD_BLEND_DIN3_SCOPE_H 0x37b7

Bit(s)	R/W	Default	Description
28:16	R/W	0x2d0	bld_din3_h_end : ///unsigned , default = 13'h2d0
12:0	R/W	0x0	bld_din3_h_start : ///unsigned , default = 13'h0

VIU_OSD_BLEND_DIN3_SCOPE_V 0x37b8

Bit(s)	R/W	Default	Description
28:16	R/W	0x1e0	bld_din3_v_end : ///unsigned , default = 13'h1e0
12:0	R/W	0x0	bld_din3_v_start : ///unsigned , default = 13'h0

VIU_OSD_BLEND_DUMMY_DATA0 0x37b9

Bit(s)	R/W	Default	Description
23:16	R/W	0x00	blend0_dummy_data_y : //unsigned , default = 0x00
15:8	R/W	0x80	blend0_dummy_data_cb : //unsigned , default = 0x80
7:0	R/W	0x80	blend0_dummy_data_cr : //unsigned , default = 0x80

VIU_OSD_BLEND_DUMMY_ALPHA 0x37ba

Bit(s)	R/W	Default	Description
28:20	R/W	0x0	blend0_dummy_alpha : //unsigned , default = 9'h0
19:11	R/W	0x0	blend1_dummy_alpha : //unsigned , default = 9'h0
8:0	R/W	0x0	blend2_dummy_alpha : //unsigned , default = 9'h0

VIU_OSD_BLEND_BLEND0_SIZE 0x37bb

Bit(s)	R/W	Default	Description
28:16	R/W	0x1e0	blend0_vsize : //unsigned , default = 13'h1e0,blend0_vsize
12:0	R/W	0x2d0	blend0_hsize : //unsigned , default = 13'h2d0,blend0_hsize

VIU_OSD_BLEND_BLEND1_SIZE 0x37bc

Bit(s)	R/W	Default	Description
28:16	R/W	0x1e0	blend1_vsize : //unsigned , default = 13'h1e0
12:0	R/W	0x2d0	blend1_hsize : ///unsigned , default = 13'h2d0

10.2.3.20 Pre/Post blend Registers

VPP_PRE_BLEND_CTRL 0x3960

Bit(s)	R/W	Default	Description
27:20	R/W	0x4	hold_lines : //unsigned , default = 0x4,blend work after hold_lines line after go_field
1:0	R/W	0x1	gclk_ctrl : //unsigned , default = 16'h1,gating ctrl

VPP_PRE_BLEND_BLEND_DUMMY_DATA 0x3961

Bit(s)	R/W	Default	Description
23:16	R/W	0x0	blend0_dummy_data_y : //unsigned , default = 0x0,blend dummy data
15:8	R/W	0x80	blend0_dummy_data_cb : //unsigned , default = 0x80,blend dummy data
7:0	R/W	0x80	blend0_dummy_data_cr : //unsigned , default = 0x80 ,blend dummy data

VPP_PRE_BLEND_DUMMY_ALPHA 0x3962

Bit(s)	R/W	Default	Description
28:20	R/W	0x0	blend0_dummy_alpha : //unsigned , default = 9'h0,blend dummy alpha
19:11	R/W	0x0	blend1_dummy_alpha : //unsigned , default = 9'h0,blend dummy alpha
8:0	R/W	0x0	blend2_dummy_alpha : //unsigned , default = 9'h0,blend dummy alpha

VPP_PRE_BLEND2_RO_CURRENT_XY 0x3963

Bit(s)	R/W	Default	Description
28:16	R.O	0x0	ro_current_x : //unsigned , default = 32'h0,blend out x point
12:0	R.O	0x0	ro_current_y : //unsigned , default = 32'h0,blend out x point

VPP_POST_PRE_BLEND_CTRL 0x3967

Bit(s)	R/W	Default	Description
27:20	R/W	0x4	hold_lines : //unsigned , default = 0x4,blend work after hold_lines line after go_field
1:0	R/W	0x1	gclk_ctrl : //unsigned , default = 16'h1,gating ctrl

VPP_POST_BLEND_BLEND_DUMMY_DATA 0x3968

Bit(s)	R/W	Default	Description
23:16	R/W	0x0	blend0_dummy_data_y : //unsigned , default = 0x0,blend dummy data
15:8	R/W	0x80	blend0_dummy_data_cb : //unsigned , default = 0x80,blend dummy data
7:0	R/W	0x80	blend0_dummy_data_cr : //unsigned , default = 0x80 ,blend dummy data

VPP_POST-BLEND_DUMMY_ALPHA 0x3969

Bit(s)	R/W	Default	Description
28:20	R/W	0x0	blend0_dummy_alpha : //unsigned , default = 9'h0,blend dummy alpha
19:11	R/W	0x0	blend1_dummy_alpha : //unsigned , default = 9'h0,blend dummy alpha
8:0	R/W	0x0	blend2_dummy_alpha : //unsigned , default = 9'h0,blend dummy alpha

VPP_POST_BLEND2_RO_CURRENT_XY 0x396a

Bit(s)	R/W	Default	Description
28:16	R.O	0x0	ro_current_x : //unsigned , default = 32'h0,blend out x point
12:0	R.O	0x0	ro_current_y : //unsigned , default = 32'h0,blend out x point

10.2.3.21 VPU AFBC Registers

AFBC_ENABLE 0x1ae0

Bit(s)	R/W	Default	Description
8	R/W	0	dec_enable : unsigned , default = 0
0	R/W	0	frm_start : unsigned , default = 0

AFBC_MODE 0x1ae1

Bit(s)	R/W	Default	Description
31	R/W	0x0	soft_reset : the use as go_field
28	R/W	0x0	Blk_mem_mode : Default = 0, body space save mode when blk_mem_mode ==1
27:26	R/W	0	rev_mode : uns, default = 0 , reverse mode
25:24	R/W	3	mif_urgent : uns, default = 3 , info mif and data mif urgent
22:16	R/W	0x0	hold_line_num :
15:14	R/W	1	burst_len : uns, default = 1, 0: burst1 1:burst2 2:burst4
13:8	R/W	0	compbits_yuv : uns, default = 0 , bit 1:0,: y component bitwidth : 00-8bit 01-9bit 10-10bit bit 3:2,: u component bitwidth : 00-8bit 01-9bit 10-10bit bit 5:4,: v component bitwidth : 00-8bit 01-9bit 10-10bit
7:6	R/W	0	vert_skip_y : uns, default = 0 , luma vert skip mode : 00-y0y1, 01-y0, 10-y1, 11-(y0+y1)/2
5:4	R/W	0	horz_skip_y : uns, default = 0 , luma horz skip mode : 00-y0y1, 01-y0, 10-y1, 11-(y0+y1)/2
3:2	R/W	0	vert_skip_uv : uns, default = 0 , chroma vert skip mode : 00-y0y1, 01-y0, 10-y1, 11-(y0+y1)/2

1:0	R/W	0	horz_skip_uv : uns, default = 0 , chroma horz skip mode : 00-y0y1, 01-y0, 10-y1, 11-(y0+y1)/2

AFBC_SIZE_IN 0x1ae2

Bit(s)	R/W	Default	Description
28:16	R/W	1920	hsize_in : uns, default = 1920 , pic horz size in unit: pixel
12:0	R/W	1080	vsize_in : uns, default = 1080 , pic vert size in unit: pixel

AFBC_DEC_DEF_COLOR 0x1ae3

Bit(s)	R/W	Default	Description
29:20	R/W	0	def_color_y : uns, default = 0, afbc dec y default setting value
19:10	R/W	0	def_color_u : uns, default = 0, afbc dec u default setting value
9: 0	R/W	0	def_color_v : uns, default = 0, afbc dec v default setting value

AFBC_CONV_CTRL 0x1ae4

Bit(s)	R/W	Default	Description
11: 0	R/W	256	conv_lbuf_len : uns, default = 256, unit=16 pixel need to set = 2^n

AFBC_LBUF_DEPTH 0x1ae5

Bit(s)	R/W	Default	Description
27:16	R/W	128	dec_lbuf_depth : uns, default = 128; // unit= 8 pixel
11:0	R/W	128	mif_lbuf_depth : uns, default = 128;

AFBC_HEAD_BADDR 0x1ae6

Bit(s)	R/W	Default	Description
31:0	R/W	0x0	mif_info_baddr : uns, default = 32'h0;

AFBC_BODY_BADDR 0x1ae7

Bit(s)	R/W	Default	Description
31:0	R/W	0x0001_0000	mif_data_baddr : uns, default = 32'h0001_0000;

AFBC_SIZE_OUT 0x1ae8

Bit(s)	R/W	Default	Description
31:29	R/W	0	reserved
28:16	R/W	1920	hszie_out: uns, default = 1920 ; // unit: 1 pixel
15:13	R/W	0	reserved

Bit(s)	R/W	Default	Description
12:0	R/W	1080	Vsize_out : uns, default = 1080 ; // unit: 1 pixel

AFBC_OUT_YSCOPE 0x1ae9

Bit(s)	R/W	Default	Description
28:16	R/W	0	out_vert_bgn : uns, default = 0 ; // unit: 1 pixel
12:0	R/W	1079	out_vert_end : uns, default = 1079 ; // unit: 1 pixel

AFBC_STAT 0x1aea

Bit(s)	R/W	Default	Description
0	RO	0x0	frm_end_stat : uns, frame end status

AFBC_VD_CFMT_CTRL 0x1aeb

Bit(s)	R/W	Default	Description
31	R/W	0x0	it : true, disable clock, otherwise enable clock
30	R/W	0x0	soft : rst bit
28	R/W	0x0	if : true, horizontal formatter use repeating to generete pixel, otherwise use bilinear interpolation
27:24	R/W	0x0	horizontal : formatter initial phase
23	R/W	0x0	horizontal : formatter repeat pixel 0 enable
22:21	R/W	0x0	horizontal : Y/C ratio, 00: 1:1, 01: 2:1, 10: 4:1
20	R/W	0x0	horizontal : formatter enable
19	R/W	0x0	if : true, always use phase0 while vertical formater, meaning always repeat data, no interpolation
18	R/W	0x0	if : true, disable vertical formatter chroma repeat last line
17	R/W	0x0	veritcal : formatter dont need repeat line on phase0, 1: enable, 0: disable
16	R/W	0x0	veritcal : formatter repeat line 0 enable
15:12	R/W	0x0	vertical : formatter skip line num at the beginning
11:8	R/W	0x0	vertical : formatter initial phase
7:1	R/W	0x0	vertical : formatter phase step (3.4)
0	R/W	0x0	vertical : formatter enable

AFBC_VD_CFMT_W 0x1aec

Bit(s)	R/W	Default	Description

27:16	R/W	0x0	horizontal : formatter width
11:0	R/W	0x0	vertical : formatter width

AFBC_MIF_HOR_SCOPE 0x1aed

Bit(s)	R/W	Default	Description
25:16	R/W	0	mif_blk_bgn_h : uns, default = 0 ; // unit: 32 pixel/block hor
9: 0	R/W	59	mif_blk_end_h : uns, default = 59 ; // unit: 32 pixel/block hor

AFBC_MIF_VER_SCOPE 0x1aee

Bit(s)	R/W	Default	Description
27:16	R/W	0	mif_blk_bgn_v : uns, default = 0 ; // unit: 32 pixel/block ver
11: 0	R/W	269	mif_blk_end_v : uns, default = 269; // unit: 32 pixel/block ver

AFBC_PIXEL_HOR_SCOPE 0x1aef

Bit(s)	R/W	Default	Description
28:16	R/W	0	dec_pixel_bgn_h : uns, default = 0 ; // unit: pixel
12: 0	R/W	1919	dec_pixel_end_h : uns, default = 1919 ; // unit: pixel

AFBC_PIXEL_VER_SCOPE 0x1af0

Bit(s)	R/W	Default	Description
28:16	R/W	0	dec_pixel_bgn_v : uns, default = 0 ; // unit: pixel
12: 0	R/W	1079	dec_pixel_end_v : uns, default = 1079 ; // unit: pixel

AFBC_VD_CFMT_H 0x1af1

Bit(s)	R/W	Default	Description
12:0	R/W	0x0	vertical : formatter height

VD2_AFBC_ENABLE 0x3180

Bit(s)	R/W	Default	Description
8	R/W	0	dec_enable : unsigned , default = 0
0	R/W	0	frm_start : unsigned , default = 0

VD2_AFBC_MODE 0x3181

Bit(s)	R/W	Default	Description
31	R/W	0x0	soft_reset : the use as go_field
28	R/W	0x0	Blk_mem_mode : Default = 0, body space save mode when blk_mem_mode ==1

27:26	R/W	0	rev_mode : uns, default = 0 , reverse mode
25:24	R/W	3	mif_urgent : uns, default = 3 , info mif and data mif urgent
22:16	R/W	0x0	hold_line_num :
15:14	R/W	1	burst_len : uns, default = 1, 0: burst1 1:burst2 2:burst4
13:8	R/W	0	compbits_yuv : uns, default = 0 , bit 1:0,: y component bitwidth : 00-8bit 01-9bit 10-10bit bit 3:2,: u component bitwidth : 00-8bit 01-9bit 10-10bit bit 5:4,: v component bitwidth : 00-8bit 01-9bit 10-10bit
7:6	R/W	0	vert_skip_y : uns, default = 0 , luma vert skip mode : 00-y0y1, 01-y0, 10-y1, 11-(y0+y1)/2
5:4	R/W	0	horz_skip_y : uns, default = 0 , luma horz skip mode : 00-y0y1, 01-y0, 10-y1, 11-(y0+y1)/2
3:2	R/W	0	vert_skip_uv : uns, default = 0 , chroma vert skip mode : 00-y0y1, 01-y0, 10-y1, 11-(y0+y1)/2
1:0	R/W	0	horz_skip_uv : uns, default = 0 , chroma horz skip mode : 00-y0y1, 01-y0, 10-y1, 11-(y0+y1)/2

VD2_AFBC_SIZE_IN 0x3182

Bit(s)	R/W	Default	Description
28:16	R/W	1920	hsize_in : uns, default = 1920 , pic horz size in unit: pixel
12:0	R/W	1080	vsize_in : uns, default = 1080 , pic vert size in unit: pixel

VD2_AFBC_DEC_DEF_COLOR 0x3183

Bit(s)	R/W	Default	Description
29:20	R/W	0	def_color_y : uns, default = 0, afbc dec y default setting value
19:10	R/W	0	def_color_u : uns, default = 0, afbc dec u default setting value
9: 0	R/W	0	def_color_v : uns, default = 0, afbc dec v default setting value

VD2_AFBC_CONV_CTRL 0x3184

Bit(s)	R/W	Default	Description
11: 0	R/W	256	conv_lbuf_len : uns, default = 256, unit=16 pixel need to set = 2^n

VD2_AFBC_LBUF_DEPTH 0x3185

Bit(s)	R/W	Default	Description
27:16	R/W	128	dec_lbuf_depth : uns, default = 128; // unit= 8 pixel

11:0	R/W	128	mif_lbuf_depth : uns, default = 128;

VD2_AFBC_HEAD_BADDR 0x3186

Bit(s)	R/W	Default	Description
31:0	R/W	0x0	mif_info_baddr : uns, default = 32'h0;

VD2_AFBC_BODY_BADDR 0x3187

Bit(s)	R/W	Default	Description
31:0	R/W	0x0001_0000	mif_data_baddr : uns, default = 32'h0001_0000;

VD2_AFBC_OUT_XSCOPE 0x3188

Bit(s)	R/W	Default	Description
28:16	R/W	0	out_horz_bgn : uns, default = 0 ; // unit: 1 pixel
12:0	R/W	1919	out_horz_end : uns, default = 1919 ; // unit: 1 pixel

VD2_AFBC_OUT_YSCOPE 0x3189

Bit(s)	R/W	Default	Description
28:16	R/W	0	out_vert_bgn : uns, default = 0 ; // unit: 1 pixel
12:0	R/W	1079	out_vert_end : uns, default = 1079 ; // unit: 1 pixel

VD2_AFBC_STAT 0x318A

Bit(s)	R/W	Default	Description
0	RO	0x0	frm_end_stat : uns, frame end status

VD2_AFBC_VD_CFMT_CTRL 0x318b

Bit(s)	R/W	Default	Description
31	R/W	0x0	it : true, disable clock, otherwise enable clock
30	R/W	0x0	soft : rst bit
28	R/W	0x0	if : true, horizontal formatter use repeating to generete pixel, otherwise use bilinear interpolation
27:24	R/W	0x0	horizontal : formatter initial phase
23	R/W	0x0	horizontal : formatter repeat pixel 0 enable
22:21	R/W	0x0	horizontal : Y/C ratio, 00: 1:1, 01: 2:1, 10: 4:1
20	R/W	0x0	horizontal : formatter enable
19	R/W	0x0	if : true, always use phase0 while vertical formater, meaning always repeat data, no interpolation
18	R/W	0x0	if : true, disable vertical formatter chroma repeat last line

Bit(s)	R/W	Default	Description
17	R/W	0x0	veritcal : formatter dont need repeat line on phase0, 1: enable, 0: disable
16	R/W	0x0	veritcal : formatter repeat line 0 enable
15:12	R/W	0x0	vertical : formatter skip line num at the beginning
11:8	R/W	0x0	vertical : formatter initial phase
7:1	R/W	0x0	vertical : formatter phase step (3.4)
0	R/W	0x0	vertical : formatter enable

VD2_AFBC_VD_CFMT_W 0x318c

Bit(s)	R/W	Default	Description
27:16	R/W	0x0	horizontal : formatter width
11:0	R/W	0x0	vertical : formatter width

VD2_AFBC_MIF_HOR_SCOPE 0x318d

Bit(s)	R/W	Default	Description
25:16	R/W	0	mif_blk_bgn_h : uns, default = 0 ; // unit: 32 pixel/block hor
9: 0	R/W	59	mif_blk_end_h : uns, default = 59 ; // unit: 32 pixel/block hor

VD2_AFBC_MIF_VER_SCOPE 0x318e

Bit(s)	R/W	Default	Description
27:16	R/W	0	mif_blk_bgn_v : uns, default = 0 ; // unit: 32 pixel/block ver
11: 0	R/W	269	mif_blk_end_v : uns, default = 269; // unit: 32 pixel/block ver

VD2_AFBC_PIXEL_HOR_SCOPE 0x318f

Bit(s)	R/W	Default	Description
28:16	R/W	0	dec_pixel_bgn_h : uns, default = 0 ; // unit: pixel
12: 0	R/W	1919	dec_pixel_end_h : uns, default = 1919 ; // unit: pixel

VD2_AFBC_PIXEL_VER_SCOPE 0x3190

Bit(s)	R/W	Default	Description
28:16	R/W	0	dec_pixel_bgn_v : uns, default = 0 ; // unit: pixel
12: 0	R/W	1079	dec_pixel_end_v : uns, default = 1079 ; // unit: pixel

VD2_AFBC_VD_CFMT_H 0x3191

Bit(s)	R/W	Default	Description

12:0	R/W	0x0	vertical : formatter height

10.2.3.22 VPP registers

VPP_DUMMY_DATA 0x1d00

Bit(s)	R/W	Default	Description
29:20	R/W	0	VD1_SC_Y : // unsigned , default = 0x10,dummy data used in the VD1 scaler,according VPP_DOLBY_CTRL[17] 1:set 8bit value 2:set 10bit value
19:10	R/W	0	VD1_SC_CB : // unsigned , default = 0x80,dummy data used in the VD1 scaler,according VPP_DOLBY_CTRL[17] 1:set 8bit value 2:set 10bit value
9 :0	R/W	0	VD1_SC_CR : // unsigned , default = 0x80,dummy data used in the VD1 scaler,according VPP_DOLBY_CTRL[17] 1:set 8bit value 2:set 10bit value

VPP_LINE_IN_LENGTH 0x1d01

Bit(s)	R/W	Default	Description
13:0	R/W	14	line_in_length : // unsigned , default = 14'd1920,VD1 scaler input hsize

VPP_PIC_IN_HEIGHT 0x1d02

Bit(s)	R/W	Default	Description
12:0	R/W	0x1fff	line_in_height : // unsigned , default = 13'h1fff,VD1 scaler input vsize

VPP_PREBLEND_VD1_H_START_END 0x1d1a

Bit(s)	R/W	Default	Description
28:16	R/W	0	start : //unsigned , default = 0x0000 ,preblend video1 horizontal start
12:0	R/W	0	end : //unsigned , default = 0x077f ,preblend video1 horizontal end

VPP_PREBLEND_VD1_V_START_END 0x1d1b

Bit(s)	R/W	Default	Description
28:16	R/W	0	start : //unsigned , default = 0x0000 ,preblend video1 vertical start
12:0	R/W	0	end : //unsigned , default = 0x0437 ,preblend video1 vertical end

VPP_POSTBLEND_VD1_H_START_END 0x1d1c

Bit(s)	R/W	Default	Description
28:16	R/W	0	start : //unsigned , default = 0x0000 ,postblend video1 horizontal start
12:0	R/W	0	end : //unsigned , default = 0x077f ,postblend video1 horizontal end

VPP_POSTBLEND_VD1_V_START_END 0x1d1d

Bit(s)	R/W	Default	Description
31:29	/	/	/

28:16	R/W	0	//unsigned , default = 0x0000 ,postblend video1 vertical start Bit
15:13	/	/	/
12:0		0x077f	//unsigned , default = 0x077f ,postblend video1 vertical end

VPP_BLEND_VD2_H_START_END 0x1d1e

Bit(s)	R/W	Default	Description
28:16	R/W	0	start : //unsigned , default = 0x0000 ,preblend/postblend video2 horizontal start
12:0	R/W	0	end : //unsigned , default = 0x077f ,preblend/postblend video2 horizontal end

VPP_BLEND_VD2_V_START_END 0x1d1f

Bit(s)	R/W	Default	Description
28:16	R/W	0	start : //unsigned , default = 0x0000 ,preblend/postblend video2 vertical start
12:0	R/W	0	end : //unsigned , default = 0x077f ,preblend/postblend video2 vertical end

VPP_PREBLEND_H_SIZE 0x1d20

Bit(s)	R/W	Default	Description
29:16	R/W	14	prebld_v_size : //unsigned , default = 14'd1080 ,preblend output vsize
13:0	R/W	14	prebld_h_size : //unsigned , default = 14'd1920 ,preblend output hsize

VPP_POSTBLEND_H_SIZE 0x1d21

Bit(s)	R/W	Default	Description
29:16	R/W	14	postbld_v_size : //unsigned , default = 14'd1080 ,postblend output vsize
13:0	R/W	14	postbld_h_size : //unsigned , default = 14'd1920 ,postblend output hsize

VPP hold lines VPP_HOLD_LINES 0x1d22

Bit(s)	R/W	Default	Description
15:8	R/W	4	prebld_hold_lines : //unsigned , default = 4,preblend hold lines
7:0	R/W	4	postbld_hold_lines : //unsigned , default = 4,postblend hold lines

VPP_MISC 0x1d26

Bit(s)	R/W	Default	Description
29	R/W	0	vpp_vks_en : //unsigned , default = 0 , vkstone enable
28	R/W	0	color_manage_en : //unsigned , default = 0 , color management enable
27	R/W	0	vd2_use_viu2_out_en : //unsigned , default = 0 , if true, vd2 use viu2 output as the input, otherwise use normal vd2 from memory
26:18	R/W	0	vd2 : alpha //unsigned , default = 0 , video 2 prebld/postbld alpha

7	R/W	1	postbld_en : //unsigned , default = 1 , postblend module enable
6	R/W	0	prebld_en : //unsigned , default = 0 , preblend module enable
3	R/W	0	sr4c1_path_sel : //unsigned , default = 0 , choose sr0 position 1:sr0 bettwen dnlp & CM 0:sr0 bettwen position after postblend
2	R/W	0	disable_rst_afifo : //unsigned , default = 0 , if true, disable resetting async fifo every vsync, otherwise every vsync the aync fifo will be reseted.
1	R/W	0	sr4c0_path_sel : //unsigned , default = 0 , choose sr0 position 1:sr0 bettwen prebld & vd1_scale 0:sr0 bettwen position after dnlp

VPP_OFIFO_SIZE 0x1d27

Bit(s)	R/W	Default	Description
31:20	R/W	0	ofifo_line_lenm1 : //unsigned , default = 0xfff , ofifo line length minus 1
19	R/W	0	vs_ctrl : //unsigned , default = 0 , if true invert input vs
18	R/W	0	hs_ctrl : //unsigned , default = 0 , if true invert input hs
17	R/W	0	force_top_bot_field_en : //unsigned , default = 0 , force top/bottom field, enable
16	R/W	0	fforce_top_bot_field : //unsigned , default = 0 , force top/bottom field, 0: top, 1: bottom
15	R/W	0	force_go_field : //unsigned , default = 0 , force one go_field, one pluse, write only
14	R/W	0	force_go_line : //unsigned , default = 0 , force one go_line, one pluse, write only
13:0	R/W	0	ofifo_size : //unsigned , default = 0x1000 , ofifo size (actually only bit 10:1 is valid), always even number

VPP_FIFO_STATUS 0x1d28

Bit(s)	R/W	Default	Description
29:25	R.O	0	ro_sco_ff_buf_count : //unsigned , default = 0,current scale out fifo counter
24:14	R.O	0	ro_afifo_count : //unsigned , default = 0,current enc afifo counter
13:1	R.O	0	ro_ofifo_buf_count : //unsigned , default = 0,current vpp line fifo ofifo counter

VPP_MATRIX_PROBE_COLOR 0x1d5c

Bit(s)	R/W	Default	Description
31:24	R.O	0	ro_probe_y_l : //unsigned , default = 0,low 8 bits of component 0
23:12	R.O	0	ro_probe_cr : //unsigned , default = 0,component 1
11:0	R.O	0	ro_probe_cb : //unsigned , default = 0,component 2

VPP_MATRIX_PROBE_COLOR1 0x1dd7

Bit(s)	R/W	Default	Description
31	R.O	0x0	ro_probe_pix_v : //it means this probe is valid in the last field/frame

3:0	R.O	0	ro_probe_y_h : //unsigned , default = 0,high 4 bits of component 0

VPP_MATRIX_HL_COLOR 0x1d5d

Bit(s)	R/W	Default	Description
23:16	R/W	0	hl_y : //unsigned , default = 0,component 0
15:8	R/W	0	hl_cb : //unsigned , default = 0,component 1
7:0	R/W	0	hl_cr : //unsigned , default = 0,component 2

VPP_MATRIX_PROBE_POS 0x1d5e

Bit(s)	R/W	Default	Description
28:16	R.O	0	probe_x : //unsigned , default = 0,probe x, postion
12:0	R.O	0	probe_y : //unsigned , default = 0,probe y, position

VPP_MATRIX_CTRL 0x1d5f

Bit(s)	R/W	Default	Description
16	R/W	0	highlight_en : //unsigned , default = 0, highlight enable
15	R.O	0	probe_post : //unsigned , default = 0, if true, probe pixel data after matrix, otherwise probe pixel data before matrix
10	R.O	0	probel_sel : //unsigned , default = 0, active when VIU_SECURE_REG[9] high, probel sel 0:vadj1 1:vadj2 2:osd2 3:postbld 4:osd1

VPP_GAINOFF_CTRL0 0x1d6a

Bit(s)	R/W	Default	Description
31	R/W	0	enable : //unsigned , default = 0, gainoff module enable
30	R/W	0	enable_sel : //unsigned , default = 0, gainoff module enable sync sel
26:16	R/W	0	gain0 : //unsigned , default = 0, gainoff module gain0
10:0	R/W	0	gain1 : //unsigned , default = 0, gainoff module gain1

VPP_GAINOFF_CTRL1 0x1d6b

Bit(s)	R/W	Default	Description
26:16	R/W	0	gain2 : //unsigned , default = 0, gainoff module gain2
12:0	R/W	0	offset0 : //signed , default = 0, gainoff module offset0

VPP_GAINOFF_CTRL2 0x1d6c

Bit(s)	R/W	Default	Description
28:16	R/W	0	offset1 : //signed , default = 0, gainoff module offset1

12:0	R/W	0	offset2 : //signed , default = 0, gainoff module offset2

VPP_GAINOFF_CTRL3 0x1d6d

Bit(s)	R/W	Default	Description
28:16	R/W	0	pre_offset0 : //signed , default = 0, gainoff module pre_offset0
12:0	R/W	0	pre_offset1 : //signed , default = 0, gainoff module pre_offset1

VPP_GAINOFF_CTRL4 0x1d6e

Bit(s)	R/W	Default	Description
12:0	R/W	0	pre_offset2 : //signed , default = 0, gainoff module pre_offset2

VPP_GAINOFF_GCLK_CTRL 0x1d6f

Bit(s)	R/W	Default	Description
1:0	R/W	0	gainoff_gclk_ctrl : //unsigned , default = 0,gainoff_gclk_ctrl

VPP_GCLK_CTRL0 0x1d72

Bit(s)	R/W	Default	Description
5:4	R/W	0	ofifo_clk1 : //unsigned , default = 0,gating clock of out linefifo in vpp
3:2	R/W	0	clk0 : //unsigned , default = 0,clk swtich of all vpp module
1	R/W	0	reg_gclk : //unsigned , default = 0,registers gate clk of vpp module

VPP_GCLK_CTRL1 0x1d73

Bit(s)	R/W	Default	Description
27:20	R/W	0	gclk_ctrl_wm : //unsigned , default = 0,gating clock of water_mark
18:15	R/W	0	dolby3_gclk_ctrl : //unsigned , default = 0,gating clock of dolby3
3:0	R/W	0	cm_gclk_ctrl : //unsigned , default = 0,gating clock of color manage

VPP_MISC1 0x1d76

Bit(s)	R/W	Default	Description
20:12	R/W	0	vd1_prebld_alpha : //unsigned , default = 0,VD1 alpha for preblend
8:0	R/W	0	vd1_postbld_alpha : //unsigned , default = 0,VD1 alpha for postblend

VPP_SRSCL_GCLK_CTRL 0x1d77

Bit(s)	R/W	Default	Description
15:8	R/W	0	gclk_ctrl_sr1 : //unsigned , default = 0 , gating clock of sr1
7:0	R/W	0	gclk_ctrl_sr0 : //unsigned , default = 0 , gating clock of sr0

VPP_BLACKEXT_CTRL 0x1d80

Bit(s)	R/W	Default	Description
31:24	R/W	0	blackext_start : //unsigned , default = 0 ,blackext_start
23:16	R/W	0	blackext_slope1 : //unsigned , default = 0 ,blackext_slope1
15:8	R/W	0	blackext_midpt : //unsigned , default = 0 ,blackext_midpt
7:0	R/W	0	blackext_slope2 : //unsigned , default = 0 ,blackext_slope2

VPP_DNLP_CTRL_00 0x1d81

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region03 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region02 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region01 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region00 output value

VPP_DNLP_CTRL_01 0x1d82

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region07 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region06 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region05 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region04 output value

VPP_DNLP_CTRL_02 0x1d83

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region11 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region10 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region09 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region08 output value

VPP_DNLP_CTRL_03 0x1d84

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region15 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region14 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region13 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region12 output value

VPP_DNLP_CTRL_04 0x1d85

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region19 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region18 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region17 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region16 output value

VPP_DNLP_CTRL_05 0x1d86

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region23 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region22 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region21 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region20 output value

VPP_DNLP_CTRL_06 0x1d87

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region27 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region26 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region25 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region24 output value

VPP_DNLP_CTRL_07 0x1d88

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region31 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region30 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region29 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region28 output value

VPP_DNLP_CTRL_08 0x1d89

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region35 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region34 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region33 output value

7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region32 output value

VPP_DNLP_CTRL_09 0x1d8a

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region39 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region38 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region37 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region36 output value

VPP_DNLP_CTRL_10 0x1d8b

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region43 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region42 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region41 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region40 output value

VPP_DNLP_CTRL_11 0x1d8c

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region47 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region46 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region45 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region44 output value

VPP_DNLP_CTRL_12 0x1d8d

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region51 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region50 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region49 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region48 output value

VPP_DNLP_CTRL_13 0x1d8e

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region55 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region54 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region53 output value

7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region52 output value

VPP_DNLP_CTRL_14 0x1d8f

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region59 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region58 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region57 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region56 output value

VPP_DNLP_CTRL_15 0x1d90

Bit(s)	R/W	Default	Description
31:24	R/W	0	data0 : //unsigned , default = 0 ,bottom of region63 output value
23:16	R/W	0	data1 : //unsigned , default = 0 ,bottom of region62 output value
15:8	R/W	0	data2 : //unsigned , default = 0 ,bottom of region61 output value
7:0	R/W	0	data3 : //unsigned , default = 0 ,bottom of region60 output value

VPP_SRSHARP0_CTRL 0x1d91

Bit(s)	R/W	Default	Description
28:16	R/W	0	srsharp_demo_split_sz : //unsigned , default = 0 ,srsharp demo top/bot left/right width
5:4	R/W	0	srsharp_demo_disp_post : //unsigned , default = 0 ,srsharp demo display postion
3	R/W	0	srsharp_demo_en : //unsigned , default = 0 ,srsharp demo enable
2	R/W	0	srsharp_c444to422_en : //unsigned , default = 0 ,srsharp format444 convert 422 enable
1	R/W	0	srsharp_buf_en : //unsigned , default = 0 ,srsharp buffer enable
0	R/W	0	srsharp_en : //unsigned , default = 0 ,srsharp enable

VPP_SRSHARP1_CTRL 0x1d92

Bit(s)	R/W	Default	Description
28:16	R/W	0	srsharp_demo_split_sz : //unsigned , default = 0 ,srsharp demo top/bot left/right width
5:4	R/W	0	srsharp_demo_disp_post : //unsigned , default = 0 ,srsharp demo display postion
3	R/W	0	srsharp_demo_en : //unsigned , default = 0 ,srsharp demo enable
2	R/W	0	srsharp_c444to422_en : //unsigned , default = 0 ,srsharp format444 convert 422 enable
1	R/W	0	srsharp_buf_en : //unsigned , default = 0 ,srsharp buffer enable
0	R/W	0	srsharp_en : //unsigned , default = 0 ,srsharp enable

VPP_DOLBY_CTRL 0x1d93

Bit(s)	R/W	Default	Description
17	R/W	0	pps_dummy_data_mode : //unsigned , default = 0 ,pps_dummy_data_mode 1:vd1_scale need setting 8 bits 0:vd1_scale need setting 10 bits
16	R/W	0	dolby3_path_sel : //unsigned , default = 0 ,1:dolby2->osd_mat->post_blend->dolby3->wm 0:dolby2->dolby3->osd_mat->post_blend->wm
10	R/W	0	vpp_clip_ext_mode2 : //unsigned , default = 0 ,Vpp out clip mode 1:10bit 0:12bit
9	R/W	0	vpp_clip_ext_mode1 : //unsigned , default = 0 ,Vpp Vd2 input clip mode 1:10bit 0:12bit
8	R/W	0	vpp_clip_ext_mode0 : //unsigned , default = 0 ,Vpp Vd1 input clip mode 1:10bit 0:12bit
3	R/W	0	vpp_dolby3_en : //unsigned , default = 0 ,dolby_core3 enable,active high
2	R/W	0	vpp_dpath_sel2 : //unsigned , default = 0 ,by_pass from WM to vks
1	R/W	0	vpp_dpath_sel1 : //unsigned , default = 0 ,by_pass from gainoff to vks
0	R/W	0	vpp_dpath_sel0 : //unsigned , default = 0 ,by_pass from preblend to blue_strech

VPP_SYNC_SEL0 0x1d96

Bit(s)	R/W	Default	Description
31:0	R/W	0	sync_sel_bits : //unsigned ,default = 0,sync_sel bits for VPP_DOLBY_CTRL

VPP_CCORING_CTRL 0x1da0

Bit(s)	R/W	Default	Description
27:16	R/W	0	reg_bypass_ccoring_ythd : // unsigned , default = 0 , bypass_ccoring_ythd
15:8	R/W	0	ccoring_th : // unsigned , default = 0 , Chroma coring threshold
3:0	R/W	0	ccoring_slope : // unsigned , default = 0 , Chroma coring slope

VPP_VE_ENABLE_CTRL 0x1da1

Bit(s)	R/W	Default	Description
29:28	R/W	0	dnlp_gclk_ctrl : // unsigned , default = 0 ,dnlp gclk ctrl
27:26	R/W	0	blackext_gclk_ctrl : // unsigned , default = 0 ,blackext gclk ctrl
25:24	R/W	0	ccoring_gclk_ctrl : // unsigned , default = 0 ,chroma coring gclk ctrl
20	R/W	0	demo_ccoring_enable : // unsigned , default = 0 ,demo chroma coring enable
19	R/W	0	demo_blackext_enable : // unsigned , default = 0 ,demo black enxtension enable
18	R/W	0	demo_dnlp_enable : // unsigned , default = 0 ,demo dynamic nonlinear luma processing enable

Bit(s)	R/W	Default	Description
15:14	R/W	0	demo_disp_position : // unsigned , default = 0 ,2'b00: demo adjust on top, 2'b01: demo adjust on bottom, 2'b10: demo adjust on left, 2'b11: demo adjust on right
4	R/W	0	ccoring_en : // unsigned , default = 0 , chroma coring enable
3	R/W	0	blackext_en : // unsigned , default = 0 , black enxtension enable
2	R/W	0	dnlp_en : // unsigned , default = 0 , dynamic nonlinear luma processing enable

VPP_VE_DEMO_LEFT_TOP_SCREEN_WIDTH 0x1da2

Bit(s)	R/W	Default	Description
12:0	R/W	0	ve_demo_left_top_screen_width : // unsigned , default = 0 demo left or top screen width

VPP_VE_DEMO_CENTER_BAR 0x1da3

Bit(s)	R/W	Default	Description
31	R/W	0	ve_demo_center_bar : // unsigned , default = 0 center bar enable
27:24	R/W	0	ve_demo_center_bar : // unsigned , default = 0 center bar width (*2)
23:16	R/W	0	ve_demo_center_bar : // unsigned , default = 0 center bar Cr (*4)
15:8	R/W	0	ve_demo_center_bar : // unsigned , default = 0 center bar Cb (*4)
7:0	R/W	0	ve_demo_center_bar : // unsigned , default = 0 center bar y (*4)

VPP_VE_H_V_SIZE 0x1da4

Bit(s)	R/W	Default	Description
28:16	R/W	780	ve_line_length : // unsigned , default = 780 ve_line_length
12:0	R/W	438	ve_pic_height : // unsigned , default = 438 ve_pic_height

VPP_OUT_H_V_SIZE 0x1da5

Bit(s)	R/W	Default	Description
28:16	R/W	780	vppout_line_length : / unsigned , default = 780 vd1_scale_out hsize
12:0	R/W	438	vppout_pic_height : // unsigned , default = 438 vd1_scale_out vsize

VPP_VDO_MEAS_CTRL 0x1da8

Bit(s)	R/W	Default	Description
10:0	R/W	0	vdo_meas_ctrl : // unsigned , default = 0 vdo_meas_ctrl

VPP_VDO_MEAS_VS_COUNT_HI *0x1da9*

Bit(s)	R/W	Default	Description
19 :16	RO	0	ro_ind_meas_count_n // unsigned , default = 0 ind_meas_count_n, every number of sync_span vsyncs, this counter add 1

15 :0,	RO	0	ro_counter_h // unsigned , default = 0 high bit portion of counter

VPP_VDO_MEAS_VS_COUNT_LO 0x1daa

Bit(s)	R/W	Default	Description
31 :0	RO	0	ro_counter_l // unsigned , default = 0,low bit portion of counter

VPP_INPUT_CTRL 0x1dab

Bit(s)	R/W	Default	Description
11:9	R/W	0	vd2_sel : // unsigned , default = 0, 001: select vd1_din, 010: select vd2_din, 011: select d2d3_l_din, 100: d2d3_r_din, otherwise no selection
8:6	R/W	0	vd1_l_sel : // unsigned , default = 0, 001: select vd1_din, 010: select vd2_din, 011: select d2d3_l_din, 100: d2d3_r_din, otherwise no selection,vd1_l_sel selected cannot be used as the source of vd1_r_sel or vd2_sel
5:3	R/W	0	vd1_r_sel : // unsigned , default = 0, 001: select vd1_din, 010: select vd2_din, 011: select d2d3_l_din, 100: d2d3_r_din, otherwise no selection,useful only vd1_interleave_mode is not 00. And the source vd1_r_sel used can not used for the vd2_sel any more bit 2:0 vd1_interleave_mode // unsigned , default = 0, 000: no interleave, 001: pixel interleaving, 010: line interleaving, 011: 2 pixel interleaving,100: 2 line interleaving

VPP_CTI_CTRL2 0x1dac

Bit(s)	R/W	Default	Description
25:24	R/W	0	cti_bpf_sel : // unsigned , default = 0
20:16	R/W	0	cti_blend_factor_gama : // unsigned , default = 0
12:8	R/W	0	cti_blend_factor_beta : // unsigned , default = 0
4:0	R/W	0	cti_blend_factor_alpha : // unsigned , default = 0

VPP_WRBAK_CTRL_SEC 0x1dad

Bit(s)	R/W	Default	Description
bit 31	R/W	0	vpp_wrbak_sel // unsigned , default = 0, 1: VPP_WRBAK_CTRL regs set to vpp_wrbak_data_ini,Cbus can't access 0: VPP_WRBAK_CTRL reg can be writen bit
30:0	R/W	0	vpp_wrbak_data_ini // unsigned , default = 0,

VD1_BLEND_SRC_CTRL_SEC 0x1dae

Bit(s)	R/W	Default	Description
31	R/W	0	vd1_blend_src_sel : // unsigned , default = 0, 1: VD1_BLEND_SRC_CTRL regs set to vd1_blend_src_data_ini,Cbus can't access 0: VD1_BLEND_SRC_CTRL reg can be writen
30:0	R/W	0	vd1_blend_src_data_ini : // unsigned , default = 0,

VD2_BLEND_SRC_CTRL_SEC 0x1daf

Bit(s)	R/W	Default	Description
31	R/W	0	vd2_blend_src_sel : // unsigned , default = 0, 1: VD2_BLEND_SRC_CTRL regs set to vd1_blend_src_data_ini,Cbus can't access 0: VD2_BLEND_SRC_CTRL reg can be writen
30:0	R/W	0	vd2_blend_src_data_ini : // unsigned , default = 0,

OSD1_BLEND_SRC_CTRL_SEC 0x1db0

Bit(s)	R/W	Default	Description
31	R/W	0	osd1_blend_src_sel : // unsigned , default = 0, 1: OSD1_BLEND_SRC_CTRL regs set to osd1_blend_src_data_ini,Cbus can't access 0: OSD1_BLEND_SRC_CTRL reg can be writen
30:0	R/W	0	osd1_blend_src_data_ini : // unsigned , default = 0,

OSD2_BLEND_SRC_CTRL_SEC 0x1db1

Bit(s)	R/W	Default	Description
31	R/W	0	osd2_blend_src_sel : // unsigned , default = 0, 1: OSD2_BLEND_SRC_CTRL regs set to osd2_blend_src_data_ini,Cbus can't access 0: OSD2_BLEND_SRC_CTRL reg can be writen
30:0	R/W	0	osd2_blend_src_data_ini : // unsigned , default = 0,

VPP_INT_LINE_NUM 0x1dce

Bit(s)	R/W	Default	Description
12:0	R/W	0x0	interrupt_line_num : //unsigned, default== 0x1fff,line number use to generate interrupt when line == this number

VPP_OFIFO_URG_CTRL 0x1dd8

Bit(s)	R/W	Default	Description
31	R/W	0x0	urgent_hold : //unsigned, default== 0, urgent fifo hold enable
28:12	R/W	0x0	urgent_fifo_th : //unsigned, default== 0, urgent fifo hold line threshold
15	R/W	0x0	urgent_ctrl_en : //unsigned, default== 0, urgent_ctrl_en
14	R/W	0x0	urgent_wr : //unsigned, default== 0, urgent_wr, if true for write buffer
13	R/W	0x0	out_inv_en : //unsigned, default== 0, out_inv_en
12	R/W	0x0	urgent_ini_value : //unsigned, default == 0, urgent_ini_value
11:6	R/W	0x0	up_th : //unsigned, default == 0, up_th up threshold

5:0	R/W	0x0	dn_th : //unsigned, default == 0, dn_th dn threshold

VPP_CLIP_MISC0 0x1dd9

Bit(s)	R/W	Default	Description
29:20	R/W	1023	r : // unsigned, default == 1023, final clip r channel top
19:10	R/W	1023	g : // unsigned, default == 1023, final clip g channel top
9: 0	R/W	1023	b : // unsigned, default == 1023, final clip b channel top

VPP_CLIP_MISC1 0x1dda

Bit(s)	R/W	Default	Description
29:20	R/W	0x0	r : // unsigned, default == 0, final clip r channel bottom
19:10	R/W	0x0	g : // unsigned, default == 0, final clip g channel bottom
9: 0	R/W	0x0	b : // unsigned, default == 0, final clip b channel bottom

VPP_VD1_CLIP_MISC0 0x1de1

Bit(s)	R/W	Default	Description
29:20	R/W	1023	r : //unsigned, default == 1023, vd1 clip r channel top
19:10	R/W	1023	g : //unsigned, default == 1023, vd1 clip g channel top
9: 0	R/W	1023	b : //unsigned, default == 1023, vd1 clip b channel top

VPP_VD1_CLIP_MISC1 0x1de2

Bit(s)	R/W	Default	Description
29:20	R/W	0x0	r : //unsigned, default = 0, vd1 clip r channel bottom
19:10	R/W	0x0	g : //unsigned, default = 0, vd1 clip g channel bottom
9: 0	R/W	0x0	b : //unsigned, default = 0, vd1 clip b channel bottom

VPP_VD2_CLIP_MISC0 0x1de3

Bit(s)	R/W	Default	Description
29:20	R/W	1023	r : //unsigned, default = 1023, vd2 clip r channel top
19:10	R/W	1023	g : //unsigned, default = 1023, vd2 clip g channel top
9: 0	R/W	1023	b : //unsigned, default = 1023, vd2 clip b channel top

VPP_VD2_CLIP_MISC1 0x1de4

Bit(s)	R/W	Default	Description
29:20	R/W	0x0	r : // unsigned, default = 0, vd2 clip r channel bottom

19:10	R/W	0x0	g : // unsigned, default = 0, vd2 clip g channel bottom
9: 0	R/W	0x0	b : // unsigned, default = 0, vd2 clip b channel bottom

VPP_VD2_HDR_IN_SIZE 0x1df0

Bit(s)	R/W	Default	Description
18:16	R/W	0	vd2_in_v_size : // unsigned, default = 0x2d0, VPP VD2 input vsize
12:0	R/W	0	vd2_in_h_size : // unsigned, default = 0x1e0, VPP VD2 input hsize

VPP_OSD1_IN_SIZE 0x1df1

Bit(s)	R/W	Default	Description
18:16	R/W	0	osd1_in_v_size : // unsigned, default = 0x2d0, VPP osd1 input vsize
12:0	R/W	0	osd1_in_h_size : // unsigned, default = 0x1e0, VPP osd1 input hsize

VPP_GCLK_CTRL2 0x1df2

Bit(s)	R/W	Default	Description
13:13	R/W	0	vks_gclk_ctrl : // unsigned, default = 0 ,vks gating clock

VD2_PPS_DUMMY_DATA 0x1df4

Bit(s)	R/W	Default	Description
23:16	R/W	0x0	Y : //unsigned, default = 0, vd2 scale dummy data
15:8	R/W	0x0	CB : //unsigned, default = 0, vd2 scale dummy data
7: 0	R/W	0x0	CR : //unsigned, default = 0, vd2 scale dummy data

VPP_OSD1_BLD_H_SCOPE 0x1df5

Bit(s)	R/W	Default	Description
28:16	R/W	0	blend_osd1_h_start : //unsigned, default = 0x0
12:0	R/W	0	blend_osd1_h_end : //unsigned, default = 0x2d0

VPP_OSD1_BLD_V_SCOPE 0x1df6

Bit(s)	R/W	Default	Description
28:16	R/W	0	blend_osd1_v_start : //unsigned, default = 0x0
12:0	R/W	0	blend_osd1_v_end : //unsigned, default = 0x1e0

VPP_OSD2_BLD_H_SCOPE 0x1df7

Bit(s)	R/W	Default	Description
28:16	R/W	0x0	blend_osd2_h_start : //unsigned, default = 0

12:0	R/W	0x0	blend_osd2_h_end : //unsigned, default = 0x2d0

VPP_OSD2_BLD_V_SCOPE 0x1df8

Bit(s)	R/W	Default	Description
28:16	R/W	0x0	blend_osd2_v_start : //unsigned, default = 0
12:0	R/W	0x0	blend_osd2_v_end : //unsigned, default = 0x1e0

VPP_WRBAK_CTRL 0x1df9

Bit(s)	R/W	Default	Description
23:16	R/W	0	wrbak_din_lnblank : //unsigned, default = 0,
11:8	R/W	0	wrbak_din_only_en : //unsigned, default = 0,
6:4	R/W	0	wrbak_chan1_sel : //unsigned, default = 0,1:vd1 2:vd2 3:osd1 4:osd2 5:posd_blend
2:0	R/W	0	wrbak_chan0_sel : //unsigned, default = 0,1:vd1 2:vd2 3:osd1 4:osd2 5:posd_blend

VPP_SLEEP_CTRL 0x1dfa

Bit(s)	R/W	Default	Description
31	R/W	1	sleep_always_en : //unsigned, default = 1
30	R/W	0	sleep_always_dis : //unsigned, default = 0
29:16	R/W	0	sleep_line_len : //unsigned, default = 0
15:14	R/W	0	sleep_mode : //unsigned, default = 0
13:0	R/W	0	sleep_beg_line : //unsigned, default = 0

VD1_BLEND_SRC_CTRL 0x1dfb

Bit(s)	R/W	Default	Description
16	R/W	0	vd1_postbld_premult : //unsigned, default = 0
11:8	R/W	1	vd1_postbld_src : //unsigned, default = 1 , 0:close 1:vd1 2:vd2 3:osd1 4:osd2
4	R/W	0	vd1_prebld_premult : //unsigned, default = 0
3:0	R/W	1	vd1_prebld_src : //unsigned, default = 1 , 0:close 1:vd1 2:vd2 3:osd1 4:osd2

VD2_BLEND_SRC_CTRL 0x1dfc

Bit(s)	R/W	Default	Description
20	R/W	0	vd2_blend_path_sel : //unsigned, default = 0
16	R/W	0	vd2_postbld_premult : //unsigned, default = 0
11:8	R/W	2	vd2_postbld_src : //unsigned, default = 2 , 0:close 1:vd1 2:vd2 3:osd1 4:osd2
4	R/W	0	vd2_prebld_premult : //unsigned, default = 0

Bit(s)	R/W	Default	Description
3:0	R/W	0	vd2_prebld_src : //unsigned, default = 0 ,0:close 1:vd1 2:vd2 3:osd1 4:osd2

OSD1_BLEND_SRC_CTRL 0x1dfd

Bit(s)	R/W	Default	Description
20	R/W	0	osd1_blend_path_sel : //unsigned, default = 0
16	R/W	0	osd1_postbld_premult : //unsigned, default = 0
11:8	R/W	3	osd1_postbld_src : //unsigned, default = 3 , 0:close 1:vd1 2:vd2 3:osd1 4:osd2
4	R/W	0	osd1_prebld_premult : //unsigned, default = 0
3:0	R/W	0	osd1_prebld_src : //unsigned, default = 0 ,0:close 1:vd1 2:vd2 3:osd1 4:osd2

OSD2_BLEND_SRC_CTRL 0x1dfe

Bit(s)	R/W	Default	Description
20	R/W	0	osd2_blend_path_sel : //unsigned, default = 0
16	R/W	0	osd2_postbld_premult : //unsigned, default = 0
11:8	R/W	4	osd2_postbld_src : //unsigned, default = 4 , 0:close 1:vd1 2:vd2 3:osd1 4:osd2
4	R/W	0	osd2_prebld_premult : //unsigned, default = 0
3:0	R/W	0	osd2_prebld_src : //unsigned, default = 0 , 0:close 1:vd1 2:vd2 3:osd1 4:osd2

10.2.3.23 CM registers

VPP_CHROMA_ADDR_PORT 0x1d70

Bit(s)	R/W	Default	Description
31-0	R/W	0	Color management address port

VPP_CHROMA_DATA_PORT 0x1d71

Bit(s)	R/W	Default	Description
31-0	R/W	0	Color management data port

Color management internal registers is indirectly accessed by the registers VPP_CHROMA_ADDR_PORT and VPP_CHROMA_DATA_PORT.

Color management registers

The example to access the Color management registers is like this:

Wr(VPP_CHROMA_ADDR_PORT);

Wr(VPP_CHROMA_DATA_PORT):

REG_CHROMA_CONTROL 0x30

Bit(s)	R/W	Default	Description

31	R/W	0	reg_chroma_en. enable color manage function 1'b1: enable 1'b0: bypass
6	R/W	0	sat_sel. uv_max or u^2+v^2 selected as sat for reference 1'b1: uv_max(default) 1'b0: u^2+v^2
5	R/W	0	uv_adj_en. final uv_adjust enable 1'b1: enable 1'b0: bypass
2	R/W	0	hue_en. rgb to hue enable 1'b1: enable(default) 1'b0: bypass
1-0	R/W	0	csc_sel. define input YUV with different color type 2'b00: 601(16-235) 2'b01: 709(16-235) 2'b10: 601(0-255) 2'b11: 709(0-255)

SAT_BYYB_NODE0 0x200

Bit(s)	R/W	Default	Description
31-24	R/W	0	The 4th node, the same as below
23-16	R/W	0	The 3th node, the same as below
15-8	R/W	0	The 2th node, the same as below
7-0	R/W	0	Signed, The 1th node about saturation gain offset along Y coordinate, the gain normalized to 128 as “1”.

SAT_BYYB_NODE1 0x201

Bit(s)	R/W	Default	Description
31-24	R/W	0	The 8th node, the same as below
23-16	R/W	0	The 7th node, the same as below
15-8	R/W	0	The 6th node, the same as below
7-0	R/W	0	Signed, The 5th node about saturation gain offset along Y coordinate, the gain normalized to 128 as “1”.

SAT_BYYB_NODE2 0x202

Bit(s)	R/W	Default	Description
31-8	R/W	0	reserved
7-0	R/W	0	Signed, The 9th node about saturation gain offset along Y coordinate, the gain normalized to 128 as “1”.

SAT_SRC_NODE 0x203

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved

27-16	R/W	0x800	unsign, Threshold of input saturation for second & third piece. i.e. it is boundary for reg_CM2_Adj_Sat_via_HS[1][:]and reg_CM2_Adj_Sat_via_HS[2][:]
15-12	R/W	0	reserved
11-0	R/W	0x400	unsign, Threshold of input saturation for first and second piece.i.e. it is boundary for reg_CM2_Adj_Sat_via_HS[0][:] and reg_CM2_Adj_Sat_via_HS[1][:]

CM_ENH_SFT_MODE 0x204

Bit(s)	R/W	Default	Description
31-9	R/W	0	reserved
8-6	R/W	0	Hue offset adjustments scale for Reg_CM2_Adj_Hue_via_H[:]& Reg_CM2_Adj_Hue_via_S[:]& Reg_CM2_Adj_Hue_via_Y[:]: 0: no scale up; 1: upscale by 2 - (-128,127)x2; 2: upscale by 4 - (-128,127)x4; 3: upscale by 8 - (-128,127)x8;
5-4	R/W	0	Luma offset adjustments scale for reg_CM2_Adj_Luma_via_Hue[i]: 0: no scale up; 1: upscale by2 - (-128,127)x2; 2: upscale by 4 - (-128,127)x4; 3: upscale by 8 - (-128,127)x8;
3-2	R/W	0	Saturation again adjustments scale for reg_CM2_Adj_Sat_via_Y[:][:] &Reg_CM2_Adj_SatGLBgain_via_Y[:]: 0: no scale up/down; 1: dnscale by 2 (-128,127)/2; 2: dnscale by 4 (-128,127)/4; 3: dnscale by 8 (-128,127)/8;
1-0	R/W	0	Saturation again adjustments scale for reg_CM2_Adj_Sat_via_HS[:][:]: 0: no scale up/down; 1: dnscale by 2 (-128,127)/2; 2: dnscale by 4 (-128,127)/4; 3: dnscale by 8 (-128,127)/8;

FRM_SIZE 0x205

Bit(s)	R/W	Default	Description
31-29	R/W	0	reserved
28-16	R/W	0x438	The frame height size

15-13	R/W	0	reserved
12-0	R/W	0x780	The frame width size

FITLER_CFG 0x206

Bit(s)	R/W	Default	Description
31-5	R/W	0	reserved
4	R/W	0	Horizontal Interleave filter (zero-padding) for 3D considerations: 0: using non-zero padding LPF 1: using zero-padding LPF
3-0	R/W	0	Apply CM on LP portion or original video pixels options: bits[1:0]: is for Luma path control; bits[3:2]: is for U/V path control; 0: no filter but still match the delay; 1: 5 taps LP filter 2: 9 taps LP filter 3: 13 taps LP filter

CM_GLOBAL_GAIN 0x207

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0x200	Global Saturation Gain for general color adjustments (0~4095 <=> 0~8), 512 normalized to “1”.
15-12	R/W	0	reserved
11-0	R/W	0	Global Hue offsets for general color adjustments (0~4095 <=> 0~360 degree)

CM_ENH_CTL 0x208

Bit(s)	R/W	Default	Description
31-7	R/W	0	reserved
6	R/W	0	Enable signal for CM2 Hue adjustments;
5	R/W	0	Enable signal for CM2 Saturation adjustments;
4	R/W	0	Enable signal for CM2 Luma adjustments;
3	R/W	0	reserved
2	R/W	0	cm2_filt_en :apply cm on lp portion enable
1	R/W	0	CM2 enable signal
0	R/W	0	CM1 enable signal

ROI_X_SCOPE 0x209

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0	Ending col index of the Region of Interest (ROI)
15-12	R/W	0	reserved
11-0	R/W	0	Start col index of the Region of Interest (ROI)

ROI_Y_SCOPE 0x20a

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0	Ending col index of the Region of Interest (ROI)
15-12	R/W	0	reserved
11-0	R/W	0	Start col index of the Region of Interest (ROI)

POI_XY_DIR 0x20b

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0	Row index of the pixel(position) of Interest (POI)
15-12	R/W	0	reserved
11-0	R/W	0	Col index of the pixel(position) of Interest (POI)

COI_Y_SCOPE 0x20c

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0	Higher bound of luma value for color of interest (COI), 8bits precision
15-12	R/W	0	reserved
11-0	R/W	0	Lower bound of luma value for color of interest (COI), 8bits precision

COI_H_SCOPE 0x20d

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0	Higher bound of Hue value for color of interest (COI),12 8bits precision
15-12	R/W	0	reserved
11-0	R/W	0	Lower bound of Hue value for color of interest (COI), 12 bits precision

COI_S_SCOPE 0x20e

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0	Higher bound of Sat value for color of interest (COI),12 8bits precision
15-12	R/W	0	reserved
11-0	R/W	0	Lower bound of Sat value for color of interest (COI), 12 bits precision

IFO_MODE 0x20f

Bit(s)	R/W	Default	Description
31-8	R/W	0	reserved
7-4	R/W	0	Mode control for COI replacement, bit[3:2] control COI pixels: 0: no replacement for COI pixels 1: disable CM2 enhance for COI pixels; 2: keep COI pixels Y but replace HS by [*HS]; 3: replace COI pixels to [*YHS] bit[1:0] controls non-COI pixels: 0: no replacement for non-COI pixels 1: disable CM2 enhance for non-COI pixels; 2: keep COI pixels Y but replace HS by [*HS]; 3: replace non- COI pixels to [*YHS]
3-0	R/W	0	Enhance mode control of pixels inside and outside Region of Interest (ROI) , bit [3:2] control ROI: 0: enable CM2 processing in ROI; 1: disable CM2 processing in ROI; 2: keep ROI pixels Y but replace HS by [*HS]; 3: ow ROI pixels to [*YHS] bit [1:0] control pixels other than ROI similarly. 0: enable CM2 processing in non-ROI; 1: disable CM2 processing in non-ROI; 2: keep ROI pixels Y but replace HS by [*HS]; 3: ow non-ROI pixels to [*YHS]

POI_RPL_MODE0x210

Bit(s)	R/W	Default	Description
31-4	R/W	0	reserved

3-0	R/W	0	Pixel of interest (POI) replacement mode: 0: no replacements; 1: one pixel of POI position replaced to [*YHS] 2: 3X3 pixels centering POI position replaced to [*YHS] 3: 5X5 pixels centering POI position replaced to [*YHS] … 15: 29X29 pixels centering POI position replaced to [*YHS]

DEMO_OWR_YHS 0x211

Bit(s)	R/W	Default	Description
31-24	R/W	0	Saturation value overwriting to ROI/POI/COI; 12bits precision, equal to saturation precision.
23-12	R/W	0	Hue value overwriting to ROI/POI/COI; 12 bits precision, equal to 1/4 hue precision. E.g. { Reg_CM2Demo_OWR_H, 2'h0}
11-0	R/W	0	Luma value overwriting to ROI/POI/COI; 8bits precision, equal to 1/4 luma precision, e.g. { Reg_CM2Demo_OWR_Y, 2'h0}

DEMO_POI_Y 0x212

Bit(s)	R/W	Default	Description
31-8	RO	0	Reserved
7-0	RO	0	Luma value for pixel of interest (POI), only get locked higher 8bits

DEMO_POI_H 0x213

Bit(s)	R/W	Default	Description
31-12	RO	0	Reserved
11-0	RO	0	Hue value for pixel of interest (POI), only get locked higher 12bits

DEMO_POI_S 0x214

Bit(s)	R/W	Default	Description
31-12	RO	0	Reserved
11-0	RO	0	Saturation value for pixel of interest (POI), only get locked higher 12bits

LUMA_ADJ_LIMT 0x215

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0	Slope to do the Luma adjust degrade speed based on Saturation. It was normalized to 16 as '1'.

15-12	R/W	0	reserved
11-0	R/W	0	Threshold to saturation to do Luma adjustment degrade. Only pixels' saturation lower than this threshold will degrade the Luma adjustment.

SAT_ADJ_LIMT 0x216

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0	Slope to do the Sat adjust degrade speed based on Saturation. It was normalized to 16 as '1'.
15-12	R/W	0	reserved
11-0	R/W	0	Threshold to saturation to do Sat adjustment degrade. Only pixels' saturation lower than this threshold will degrade the Luma adjustment.

HUE_ADJ_LIMT 0x217

Bit(s)	R/W	Default	Description
31-28	R/W	0	reserved
27-16	R/W	0	Slope to do the Hue adjust degrade speed based on Saturation. It was normalized to 16 as '1'.
15-12	R/W	0	reserved
11-0	R/W	0	Threshold to saturation to do Hue adjustment degrade. Only pixels' saturation lower than this threshold will degrade the Luma adjustment.

UVHS_OFST 0x218

Bit(s)	R/W	Default	Description
31-24	R/W	0	V offset after CM2, under s10 scale
23-16	R/W	0	U offset after CM2, under s10 scale
15-8	R/W	0	V offset before CM2, under s10 scale
7-0	R/W	0	U offset before CM2, under s10 scale.

HUE_CFG_PARA 0x219

Bit(s)	R/W	Default	Description
31-17	R/W	0	reserved
16	R/W	0	Options to protect HUE after CM2 adjustments. This will be added to avoid HUE distortion if Saturation is enhanced too much.
15-13	R/W	0	Hue adjustment via HS the Saturation division mode: 0: 1024/2048/3072, 4095; 1: 512, 1024, 1536, 2048; 2: 256, 512, 768, 1024;

Bit(s)	R/W	Default	Description
			3: 128, 256, 384, 512; 4: 512/1024/2048/4096; 5: 256/512/1024/2048; 6: 128/256/512/1024; 7: 64,128,256,512
12	R/W	0	Hue slice division mode: 0: 32 pieces, 360/32 degrees each slice; 1/up: first 20 slices with 360/64 degrees each slice, others 360/16 degrees each slices. Notes, this option provide options to get more precise Hue adjustments for FTC/Red and so on
11-0	R/W	0	Hue offset before CM2 adjustment, this will provide options to divide the Hue slices with a precise offset. But need to compensate back with the global Hue after CM2 adjusments

DEMO_SPLT_CFG 0x21a

Bit(s)	R/W	Default	Description
31-22	R/W	0	reserved
21-20	R/W	0	Demo split post
19-16	R/W	0	Demo split width
12-0	R/W	0	Demo split mode

DEMO_SPLT_YHS 0x21b

Bit(s)	R/W	Default	Description
31-24	R/W	0	Luma value
23-12	R/W	0	Hue value
11-0	R/W	0	Sat value

XVYCC_YSCP_REG 0x21c

Bit(s)	R/W	Default	Description
27:16	R/W	0x3ff	xvycc_y_max
11:0	R/W	0x0	xvycc_y_min

XVYCC_USCP_REG 0x21d

Bit(s)	R/W	Default	Description
27:16	R/W	0x3ff	xvycc_u_max
11:0	R/W	0x0	xvycc_u_min

XVYCC_VSCP_REG 0x21e

Bit(s)	R/W	Default	Description
27:16	R/W	0x3ff	xvycc_v_max
11:0	R/W	0x0	xvycc_v_min

the main adjust parameter is saved according to 32 hue node order, one by one. The parameter of each hue node is same. All the parameter for each hue occupy 5 register-addr-space. For the addr-offset aligned, we allocate 8 addr-space to each node parameter, for example,

the parameter of 1th node uses the address space : 0x100, 0x101, 0x102, 0x103, 0x104,

and 2th node uses the address space : 0x108, 0x109, 0x10a, 0x10b, 0x10c,

and 2th node uses the address space : 0x110, 0x111, 0x112, 0x113, 0x114,

… … … …

CM2_ENH_COEF0_H00 0x100

Bit(s)	R/W	Default	Description
31-24	R/W	0	Same as last
23-16	R/W	0	Same as last
15-8	R/W	0	Signed, Saturation gain offset for three pieces saturation on Hue section (totally 32 sections) node 0; the gain normalized to 128 as “1”.
7-0	R/W	0	Signed, Luma offsets for Hue section (totally 32 sections) nodes 0 , range (-128,127)

CM2_ENH_COEF1_H00 0x101

Bit(s)	R/W	Default	Description
31-24	R/W	0	Signed, Hue offset for each four pieces Luma region on each Hue section (totally 32 sections) nodes, this is for y=2/4, hue node 0
23-16	R/W	0	Signed, Hue offset for each four pieces Luma region on each Hue section (totally 32 sections) nodes, this is for y=1/4, hue node 0
15-8	R/W	0	Signed, Hue offset for each four pieces Luma region on each Hue section (totally 32 sections) nodes, this is for y=0, hue node 0
7-0	R/W	0	Signed, Hue offset on Hue section (totally 32 sections) node 0

CM2_ENH_COEF2_H00 0x102

Bit(s)	R/W	Default	Description
31-24	R/W	0	Signed, Hue offset for each four pieces Saturation region on each Hue section (totally 32 sections) nodes; This is for sat = 1/4, hue node 0
23-16	R/W	0	Signed, Hue offset for each four pieces Saturation region on each Hue section (totally 32 sections) nodes; This is for sat = 0, hue node 0
15-8	R/W	0	Signed, Hue offset for each four pieces Luma region on each Hue section (totally 32 sections) nodes, this is for y=4/4, hue node 0

Bit(s)	R/W	Default	Description
7-0	R/W	0	Signed, Hue offset for each four pieces Luma region on each Hue section (totally 32 sections) nodes, this is for y=3/4, hue node 0

CM2_ENH_COEF3_H00 0x103

Bit(s)	R/W	Default	Description
31-24	R/W	0	Saturation gain offsetfor four pieces Luma on each Hue section (totally 32 sections) nodes; the gain normalized to 128 as “1”. This is for sat = 0, hue node 0
23-16	R/W	0	Signed, Hue offset for each four pieces Saturation region on each Hue section (totally 32 sections) nodes; This is for sat = 4/4, hue node 0
15-8	R/W	0	Signed, Hue offset for each four pieces Saturation region on each Hue section (totally 32 sections) nodes; This is for sat = 3/4, hue node 0
7-0	R/W	0	Signed, Hue offset for each four pieces Saturation region on each Hue section (totally 32 sections) nodes; This is for sat = 2/4, hue node 0

CM2_ENH_COEF4_H00 0x104

Bit(s)	R/W	Default	Description
31-24	R/W	0	Saturation gain offsetfor four pieces Luma on each Hue section (totally 32 sections) nodes; the gain normalized to 128 as “1”. This is for sat = 4/4, hue node 0
23-16	R/W	0	Saturation gain offsetfor four pieces Luma on each Hue section (totally 32 sections) nodes; the gain normalized to 128 as “1”. This is for sat = 3/4, hue node 0
15-8	R/W	0	Saturation gain offsetfor four pieces Luma on each Hue section (totally 32 sections) nodes; the gain normalized to 128 as “1”. This is for sat = 2/4, hue node 0
7-0	R/W	0	Saturation gain offsetfor four pieces Luma on each Hue section (totally 32 sections) nodes; the gain normalized to 128 as “1”. This is for sat = 1/4, hue node 0

CM2_ENH_COEF0_H01 0x108

CM2_ENH_COEF1_H01 0x109

CM2_ENH_COEF2_H01 0x10a

CM2_ENH_COEF3_H01 0x10b

CM2_ENH_COEF4_H01 0x10c

CM2_ENH_COEF0_H02 0x110

CM2_ENH_COEF1_H02 0x111

CM2_ENH_COEF2_H02 0x112

CM2_ENH_COEF3_H02 0x113

CM2_ENH_COEF4_H02 0x114

CM2_ENH_COEF0_H31 0x1f8

CM2_ENH_COEF1_H31 0x1f9

CM2_ENH_COEF2_H31 0x1fa

CM2_ENH_COEF3_H31 0x1fb

CM2_ENH_COEF4_H31 0x1fc

10.2.3.24 VPP OSD1 SCALER

VPP_OSD_VSC_PHASE_STEP 0x1dc0

Bit(s)	R/W	Default	Description
27-0	R/W	0x01000000	4.24 format

VPP_OSD_VSC_INI_PHASE 0x1dc1

Bit(s)	R/W	Default	Description
31-16	R/W	0x0	botttom vertical scaler initial phase
15-0	R/W	0x0	top vertical scaler initial phase

VPP_OSD_VSC_CTRL0 0x1dc2

Bit(s)	R/W	Default	Description
24	R/W	0x0	osd vertical Scaler enable
23	R/W	0x0	osd_prog_interlace 0: current field is progressive, 1: current field is interlace
22-21	R/W	0x0	osd_vsc_double_line_mode, bit1, double input width and half input height, bit0, change line buffer becomes 2 lines
20	R/W	0x0	osd_vsc_phase0_always_en
19	R/W	0x0	osd_vsc_nearest_en
17-16	R/W	0x0	osd_vsc_bot_rpt_l0_num
14-11	R/W	0x0	osd_vsc_bot_ini_rcv_num

Bit(s)	R/W	Default	Description
9-8	R/W	0x0	osd_vsc_top_rpt_l0_num
6-3	R/W	0x0	osd_vsc_top_ini_rcv_num
2-0	R/W	0x0	osd_vsc_bank_length

VPP_OSD_HSC_PHASE_STEP 0x1dc3

Bit(s)	R/W	Default	Description
27-0	R/W	0x01000000	4.24 format

VPP_OSD_HSC_INI_PHASE 0x1dc4

Bit(s)	R/W	Default	Description
31-16	R/W	0x0	horizontal scaler initial phase1
15-0	R/W	0x0	horizontal scaler initial phase0

VPP_OSD_HSC_CTRL0 0x1dc5

Bit(s)	R/W	Default	Description
22	R/W	0x0	osd horizontal Scaler enable
21	R/W	0x0	osd_hsc_double_pix_mode
20	R/W	0x0	osd_hsc_phase0_always_en
19	R/W	0x0	osd_vsc_nearest_en
17-16	R/W	0x0	osd_hsc_rpt_p0_num1
14-11	R/W	0x0	osd_hsc_ini_rcv_num1
9-8	R/W	0x0	osd_hsc_rpt_p0_num0
6-3	R/W	0x0	osd_hsc_ini_rcv_num0
2-0	R/W	0x0	osd_hsc_bank_length

VPP_OSD_HSC_INI_PAT_CTRL 0x1dc6

Bit(s)	R/W	Default	Description
15-8	R/W	0x0	for 3D quincunx sub-sampling. pattern, each patten 1 bit, from lsb -> msb
6-4	R/W	0x0	pattern start
2-0	R/W	0x0	pattern end

VPP_OSD_SC_DUMMY_DATA 0x1dc7

Bit(s)	R/W	Default	Description
31-24	R/W	0x0	componet 0 ,data = (dst_data >> 4) ,so 8 bit in 12bit mode, 6bit in 10bit mode
23-16	R/W	0x0	component 1 ,data = (dst_data >> 4) ,so 8 bit in 12bit mode, 6bit in 10bit mode
15-8	R/W	0x0	component 2 ,data = (dst_data >> 4) ,so 8 bit in 12bit mode, 6bit in 10bit mode
7-0	R/W	0x0	component 3 , alpha

VPP_OSD_SC_CTRL0 0x1dc8

Bit(s)	R/W	Default	Description
27:16	R/W	0	osd_sc_gclk_ctrl : //unsigned,default = 0,osd_sc_gclk_ctrl
13	R/W	0	osd_sc_din_osd_alpha_mode : //unsigned,default = 0,osc_sc_din_osd2_alpha_mode, 1: (alpha >= 128) ? alpha -1: alpha, 0: (alpha >=1) ? alpha - 1: alpha.
12	R/W	0	osd_sc_dout_alpha_mode : //unsigned,default = 0,osc_sc_alpha_mode, 1: (alpha >= 128) ? alpha + 1: alpha, 0: (alpha >=1) ? alpha + 1: alpha.
11:4	R/W	0	osd_sc_alpha : //unsigned,default = 0,default alpha for vd1 or vd2 if they are selected as the source
3	R/W	0	osd_sc_path_en : //unsigned,default = 0,osd scaler path enable
2	R/W	0	osd_sc_en : //unsigned,default = 0,osd scaler enable

VPP_OSD_SCI_WH_M1 0x1dc9

Bit(s)	R/W	Default	Description
28-16	R/W	0x0	OSD scaler input width minus 1
12-0	R/W	0x0	OSD scaler input height minus 1

VPP_OSD_SCO_H_START_END 0x1dca

Bit(s)	R/W	Default	Description
27-16	R/W	0x0	OSD scaler output horizontal start
11-0	R/W	0x0	OSD scaler output horizontal end

VPP_OSD_SCO_V_START_END 0x1dcb

Bit(s)	R/W	Default	Description
27-16	R/W	0x0	OSD scaler output vertical start
11-0	R/W	0x0	OSD scaler output vertical end

VPP_OSD_SCALE_COEF_IDX 0x1dcc

Bit(s)	R/W	Default	Description

15	R/W	0x0	Because there are many coefficients used in the vertical filter and horizontal filters,
			//indirect access the coefficients of vertical filter and horizontal filter is used.
			//For vertical filter, there are 33x4 coefficients
			//For horizontal filter, there are 33x4 coefficients
			//Bit 15 index increment, if bit9 == 1 then (0: index increase 1, 1: index increase 2) else (index increase 2)

14	R/W	0x0	1: read coef through cbus enable, just for debug purpose in case when we wanna check the coef in ram in correct or not
9	R/W	0x0	if true, use 9bit resolution coef, other use 8bit resolution coef
8	R/W	0x0	type of index, 0: vertical coef, 1: horizontal coef
6-0	R/W	0x0	coef index

VPP_OSD_SCALE_COEF 0x1dcd

Bit(s)	R/W	Default	Description
31-0	R/W	0x0	

OSD_DB_FLT_CTRL 0x3140

Bit(s)	R/W	Default	Description
26	R/W	1	reg_nrdeband_reset1 : // unsigned , default = 0 0 : no reset seed 1: reload chroma seed
25	R/W	1	reg_nrdeband_reset0 : // unsigned , default = 0 0 : no reset seed 1: reload luma seed
24	R/W	0	reg_nrdeband_rgb : // unsigned , default = 0 0 : yuv 1: RGB
23	R/W	0	reg_nrdeband_en11 : // unsigned , default = 1 debanding registers of side lines, [0] for luma, same for below
22	R/W	0	reg_nrdeband_en10 : // unsigned , default = 1 debanding registers of side lines, [1] for chroma, same for below
21	R/W	1	reg_nrdeband_siderand : // unsigned , default = 1 options to use side two lines use the rand, instead of use for the YUV three component of middle line, 0: seed[3]/bandrand[3] for middle line yuv; 1: seed[3]/bandrand[3] for nearby three lines Y;
20	R/W	0	reg_nrdeband_randmode : // unsigned , default = 0 mode of rand noise adding, 0: same noise strength for all difs; else: strenght of noise will not exceed the difs, MIN((pPKReg->reg_nrdeband_bandrand[m]), noise[m])
19:17	R/W	6	reg_nrdeband_bandrand2 : // unsigned , default = 6
15:13	R/W	6	reg_nrdeband_bandrand1 : // unsigned , default = 6
11: 9	R/W	6	reg_nrdeband_bandrand0 : // unsigned , default = 6
7	R/W	1	reg_nrdeband_hpxor1 : // unsigned , default = 1 debanding random hp portion xor, [0] for luma
6	R/W	1	reg_nrdeband_hpxor0 : // unsigned , default = 1 debanding random hp portion xor, [1] for chroma

Bit(s)	R/W	Default	Description
5	R/W	0	reg_nrdeband_en1 : // unsigned , default = 1 debanding registers, for luma
4	R/W	0	reg_nrdeband_en0 : // unsigned , default = 1 debanding registers, for chroma
3: 2	R/W	2	reg_nrdeband_lpf_mode1 : // unsigned , default = 2 lpf mode, 0: 3x3, 1:3x5; 2: 5x5; 3:5x7
1: 0	R/W	2	reg_nrdeband_lpf_mode0 : // unsigned , default = 2 lpf mode, 0: 3x3, 1:3x5; 2: 5x5; 3:5x7

OSD_DB_FLT_CTRL1 0x3141

Bit(s)	R/W	Default	Description
31:18			reserved
17:16	R/W	2	Reg_osddeband_noise_rs
15:12	R/W	8	Reg_osddeband_randgain
11		reserved	
10:8	R/W	6	Reg_osddedband_bandrand5
7		reserved	
6: 4	R/W	6	Reg_osddedband_bandrand4
3		reserved	
2: 0	R/W	6	Reg_osddeband_bandrand3

OSD_DB_FLT_LUMA_THRD 0x3142

Bit(s)	R/W	Default	Description
31:30			
29:24	R/W	36	reg_nrdeband_luma_th3 : // unsigned , default = 7 elseif <th[1] use (lpf*3 + y)/4
23:22			
21:16	R/W	28	reg_nrdeband_luma_th2 : // unsigned , default = 5 elseif <th[1] use (lpf*3 + y)/4elseif elseif <th[3] (lpf*1 + 3*y)/4; else
15:14			
13: 8	R/W	24	reg_nrdeband_luma_th1 : // unsigned , default = 7 elseif <th[1] use (lpf*3 + y)/4
7: 6			
5: 0	R/W	20	reg_nrdeband_luma_th0 : // unsigned , default = 5 elseif <th[1] use (lpf*3 + y)/4elseif elseif

OSD_DB_FLT_LUMA_THRD 0x3143

Bit(s)	R/W	Default	Description

31:30			
29:24	R/W	36	reg_nrdeband_chrm_th3 : // unsigned , default = 7 elseif <th[1] use (lpf*3 + y)/4
23:22			
21:16	R/W	28	reg_nrdeband_chrm_th2 : // unsigned , default = 5 elseif <th[1] use (lpf*3 + y)/4elseif elseif <th[3] (lpf*1 + 3*y)/4; else
15:14			
13: 8	R/W	24	reg_nrdeband_chrm_th1 : // unsigned , default = 7 elseif <th[1] use (lpf*3 + y)/4
7: 6			
5: 0	R/W	20	reg_nrdeband_chrm_th0 : // unsigned , default = 5 elseif <th[1] use (lpf*3 + y)/4elseif elseif

OSD_DB_FLT_RANDLUT 0x3144

Bit(s)	R/W	Default	Description
23:21	R/W	1	reg_nrdeband_randslut7 : // unsigned , default = 1 lut0
20:18	R/W	1	reg_nrdeband_randslut6 : // unsigned , default = 1 lut0
17:15	R/W	1	reg_nrdeband_randslut5 : // unsigned , default = 1 lut0
14:12	R/W	1	reg_nrdeband_randslut4 : // unsigned , default = 1 lut0
11: 9	R/W	1	reg_nrdeband_randslut3 : // unsigned , default = 1 lut0
8: 6	R/W	1	reg_nrdeband_randslut2 : // unsigned , default = 1 lut0
5: 3	R/W	1	reg_nrdeband_randslut1 : // unsigned , default = 1 lut0
2: 0	R/W	1	reg_nrdeband_randslut0 : // unsigned , default = 1 lut0

OSD_DB_FLT_PXI_THRD 0x3145

Bit(s)	R/W	Default	Description
25:16	R/W	0	reg_nrdeband_yc_th1 : // unsigned , default = 0 to luma/|u/v| for using the denoise
9: 0	R/W	0	reg_nrdeband_yc_th0 : // unsigned , default = 0 to luma/|u/v| for using the denoise

OSD_DB_FLT_SEED_Y 0x3146

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438240	reg_nrdeband_seed0 : // unsigned , default = 1621438240 noise adding seed for Y. seed[0]= 0x60a52f20; as default

OSD_DB_FLT_SEED_U 0x3147

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438247	reg_nrdeband_seed1 : // unsigned , default = 1621438247 noise adding seed for U. seed[0]= 0x60a52f27; as default

OSD_DB_FLT_SEED_V 0x3148

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438242	reg_nrdeband_seed2 : // unsigned , default = 1621438242 noise adding seed for V. seed[0]= 0x60a52f22; as default

OSD_DB_FLT_SEED3 0x3149

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438242	reg_nrdeband_seed3 : // unsigned , default = 1621438242 noise adding seed for V. seed[0]= 0x60a52f22; as default

OSD_DB_FLT_SEED4 0x314a

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438242	reg_nrdeband_seed4 : // unsigned , default = 1621438242 noise adding seed for V. seed[0]= 0x60a52f22; as default

OSD_DB_FLT_SEED5 0x314b

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438242	reg_nrdeband_seed5 : // unsigned , default = 1621438242 noise adding seed for V. seed[0]= 0x60a52f22; as default

10.2.3.25 VPP OSD2 SCALER

OSD2_VSC_PHASE_STEP 0x3d00

Bit(s)	R/W	Default	Description
27-0	R/W	0x01000000	4.24 format

OSD2_VSC_INI_PHASE 0x3d01

Bit(s)	R/W	Default	Description
31-16	R/W	0x0	botttom vertical scaler initial phase
15-0	R/W	0x0	top vertical scaler initial phase

OSD2_VSC_CTRL0 0x3d02

Bit(s)	R/W	Default	Description
24	R/W	0x0	osd vertical Scaler enable
23	R/W	0x0	osd_prog_interlace 0: current field is progressive, 1: current field is interlace
22-21	R/W	0x0	osd_vsc_double_line_mode, bit1, double input width and half input height, bit0, change line buffer becomes 2 lines
20	R/W	0x0	osd_vsc_phase0_always_en

19	R/W	0x0	osd_vsc_nearest_en
17-16	R/W	0x0	osd_vsc_bot_rpt_l0_num
14-11	R/W	0x0	osd_vsc_bot_ini_rcv_num
9-8	R/W	0x0	osd_vsc_top_rpt_l0_num
6-3	R/W	0x0	osd_vsc_top_ini_rcv_num
2-0	R/W	0x0	osd_vsc_bank_length

OSD2_HSC_PHASE_STEP 0x3d03

Bit(s)	R/W	Default	Description
27-0	R/W	0x01000000	4.24 format

OSD2_HSC_INI_PHASE 0x3d04

Bit(s)	R/W	Default	Description
31-16	R/W	0x0	horizontal scaler initial phase1
15-0	R/W	0x0	horizontal scaler initial phase0

OSD2_HSC_CTRL0 0x3d05

Bit(s)	R/W	Default	Description
22	R/W	0x0	osd horizontal Scaler enable
21	R/W	0x0	osd_hsc_double_pix_mode
20	R/W	0x0	osd_hsc_phase0_always_en
19	R/W	0x0	osd_vsc_nearest_en
17-16	R/W	0x0	osd_hsc_rpt_p0_num1
14-11	R/W	0x0	osd_hsc_ini_rcv_num1
9-8	R/W	0x0	osd_hsc_rpt_p0_num0
6-3	R/W	0x0	osd_hsc_ini_rcv_num0
2-0	R/W	0x0	osd_hsc_bank_length

OSD2_HSC_INI_PAT_CTRL 0x3d06

Bit(s)	R/W	Default	Description
15-8	R/W	0x0	for 3D quincunx sub-sampling. pattern, each patten 1 bit, from lsb -> msb
6-4	R/W	0x0	pattern start
2-0	R/W	0x0	pattern end

OSD2_SC_DUMMY_DATA 0x3d07

Bit(s)	R/W	Default	Description
31-24	R/W	0x0	componet 0 ,data = (dst_data >> 4) ,so 8 bit in 12bit mode, 6bit in 10bit mode
23-16	R/W	0x0	component 1 ,data = (dst_data >> 4) ,so 8 bit in 12bit mode, 6bit in 10bit mode
15-8	R/W	0x0	component 2 ,data = (dst_data >> 4) ,so 8 bit in 12bit mode, 6bit in 10bit mode
7-0	R/W	0x0	component 3 , alpha

OSD2_SC_CTRL0 0x3d08

Bit(s)	R/W	Default	Description
27:16	R/W	0	osd_sc_gclk_ctrl : //unsigned,default = 0,osd_sc_gclk_ctrl
13	R/W	0	osd_sc_din_osd_alpha_mode : //unsigned,default = 0,osc_sc_din_osd2_alpha_mode, 1: (alpha >= 128) ? alpha -1: alpha, 0: (alpha >=1) ? alpha - 1: alpha.
12	R/W	0	osd_sc_dout_alpha_mode : //unsigned,default = 0,osc_sc_alpha_mode, 1: (alpha >= 128) ? alpha + 1: alpha, 0: (alpha >=1) ? alpha + 1: alpha.
11:4	R/W	0	osd_sc_alpha : //unsigned,default = 0,default alpha for vd1 or vd2 if they are selected as the source
3	R/W	0	osd_sc_path_en : //unsigned,default = 0,osd scaler path enable
2	R/W	0	osd_sc_en : //unsigned,default = 0,osd scaler enable

OSD2_SCI_WH_M1 0x3d09

Bit(s)	R/W	Default	Description
28-16	R/W	0x0	OSD scaler input width minus 1
12-0	R/W	0x0	OSD scaler input height minus 1

OSD2_SCO_H_START_END 0x3d0a

Bit(s)	R/W	Default	Description
27-16	R/W	0x0	OSD scaler output horizontal start
11-0	R/W	0x0	OSD scaler output horizontal end

OSD2_SCO_V_START_END 0x3d0b

Bit(s)	R/W	Default	Description
27-16	R/W	0x0	OSD scaler output vertical start
11-0	R/W	0x0	OSD scaler output vertical end

OSD2_SCALE_COEF_IDX 0x3d18

Bit(s)	R/W	Default	Description

15	R/W	0x0	Because there are many coefficients used in the vertical filter and horizontal filters,
			//indirect access the coefficients of vertical filter and horizontal filter is used.
			//For vertical filter, there are 33x4 coefficients
			//For horizontal filter, there are 33x4 coefficients
			//Bit 15 index increment, if bit9 == 1 then (0: index increase 1, 1: index increase 2) else (index increase 2)

14	R/W	0x0	1: read coef through cbus enable, just for debug purpose in case when we wanna check the coef in ram in correct or not
9	R/W	0x0	if true, use 9bit resolution coef, other use 8bit resolution coef
8	R/W	0x0	type of index, 0: vertical coef, 1: horizontal coef
6-0	R/W	0x0	coef index

OSD2_SCALE_COEF 0x3d19

Bit(s)	R/W	Default	Description
31-0	R/W	0x0	scaler coef

10.2.3.26 VPP OSD_BLD34 SCALER

OSD34_VSC_PHASE_STEP 0x3d20

Bit(s)	R/W	Default	Description
27-0	R/W	0x01000000	4.24 format

OSD34_VSC_INI_PHASE 0x3d21

Bit(s)	R/W	Default	Description
31-16	R/W	0x0	botttom vertical scaler initial phase
15-0	R/W	0x0	top vertical scaler initial phase

OSD34_VSC_CTRL0 0x3d22

Bit(s)	R/W	Default	Description
24	R/W	0x0	osd vertical Scaler enable
23	R/W	0x0	osd_prog_interlace 0: current field is progressive, 1: current field is interlace
22-21	R/W	0x0	osd_vsc_double_line_mode, bit1, double input width and half input height, bit0, change line buffer becomes 2 lines
20	R/W	0x0	osd_vsc_phase0_always_en
19	R/W	0x0	osd_vsc_nearest_en

Bit(s)	R/W	Default	Description
17-16	R/W	0x0	osd_vsc_bot_rpt_l0_num
14-11	R/W	0x0	osd_vsc_bot_ini_rcv_num
9-8	R/W	0x0	osd_vsc_top_rpt_l0_num
6-3	R/W	0x0	osd_vsc_top_ini_rcv_num
2-0	R/W	0x0	osd_vsc_bank_length

OSD34_HSC_PHASE_STEP 0x3d23

Bit(s)	R/W	Default	Description
27-0	R/W	0x01000000	4.24 format

OSD34_HSC_INI_PHASE 0x3d24

Bit(s)	R/W	Default	Description
31-16	R/W	0x0	horizontal scaler initial phase1
15-0	R/W	0x0	horizontal scaler initial phase0

OSD34_HSC_CTRL0 0x3d25

Bit(s)	R/W	Default	Description
22	R/W	0x0	osd horizontal Scaler enable
21	R/W	0x0	osd_hsc_double_pix_mode
20	R/W	0x0	osd_hsc_phase0_always_en
19	R/W	0x0	osd_vsc_nearest_en
17-16	R/W	0x0	osd_hsc_rpt_p0_num1
14-11	R/W	0x0	osd_hsc_ini_rcv_num1
9-8	R/W	0x0	osd_hsc_rpt_p0_num0
6-3	R/W	0x0	osd_hsc_ini_rcv_num0
2-0	R/W	0x0	osd_hsc_bank_length

OSD34_HSC_INI_PAT_CTRL 0x3d26

Bit(s)	R/W	Default	Description
15-8	R/W	0x0	for 3D quincunx sub-sampling. pattern, each patten 1 bit, from lsb -> msb
6-4	R/W	0x0	pattern start
2-0	R/W	0x0	pattern end

OSD34_SC_DUMMY_DATA 0x3d27

Bit(s)	R/W	Default	Description
31-24	R/W	0x0	componet 0 ,data = (dst_data >> 4) ,so 8 bit in 12bit mode, 6bit in 10bit mode
23-16	R/W	0x0	component 1 ,data = (dst_data >> 4) ,so 8 bit in 12bit mode, 6bit in 10bit mode
15-8	R/W	0x0	component 2 ,data = (dst_data >> 4) ,so 8 bit in 12bit mode, 6bit in 10bit mode
7-0	R/W	0x0	component 3 , alpha

OSD34_SC_CTRL0 0x3d28

Bit(s)	R/W	Default	Description
27:16	R/W	0	osd_sc_gclk_ctrl : //unsigned,default = 0,osd_sc_gclk_ctrl
13	R/W	0	osd_sc_din_osd_alpha_mode : //unsigned,default = 0,osc_sc_din_osd2_alpha_mode, 1: (alpha >= 128) ? alpha -1: alpha, 0: (alpha >=1) ? alpha - 1: alpha.
12	R/W	0	osd_sc_dout_alpha_mode : //unsigned,default = 0,osc_sc_alpha_mode, 1: (alpha >= 128) ? alpha + 1: alpha, 0: (alpha >=1) ? alpha + 1: alpha.
11:4	R/W	0	osd_sc_alpha : //unsigned,default = 0,default alpha for vd1 or vd2 if they are selected as the source
3	R/W	0	osd_sc_path_en : //unsigned,default = 0,osd scaler path enable
2	R/W	0	osd_sc_en : //unsigned,default = 0,osd scaler enable

OSD34_SCI_WH_M1 0x3d29

Bit(s)	R/W	Default	Description
28-16	R/W	0x0	OSD scaler input width minus 1
12-0	R/W	0x0	OSD scaler input height minus 1

OSD34_SCO_H_START_END 0x3d2a

Bit(s)	R/W	Default	Description
27-16	R/W	0x0	OSD scaler output horizontal start
11-0	R/W	0x0	OSD scaler output horizontal end

OSD34_SCO_V_START_END 0x3d2b

Bit(s)	R/W	Default	Description
27-16	R/W	0x0	OSD scaler output vertical start
11-0	R/W	0x0	OSD scaler output vertical end

OSD34_SCALE_COEF_IDX 0x3d1e

Bit(s)	R/W	Default	Description

15	R/W	0x0	Because there are many coefficients used in the vertical filter and horizontal filters,
			//indirect access the coefficients of vertical filter and horizontal filter is used.
			//For vertical filter, there are 33x4 coefficients
			//For horizontal filter, there are 33x4 coefficients
			//Bit 15 index increment, if bit9 == 1 then (0: index increase 1, 1: index increase 2) else (index increase 2)

14	R/W	0x0	1: read coef through cbus enable, just for debug purpose in case when we wanna check the coef in ram in correct or not
9	R/W	0x0	if true, use 9bit resolution coef, other use 8bit resolution coef
8	R/W	0x0	type of index, 0: vertical coef, 1: horizontal coef
6-0	R/W	0x0	coef index

OSD34_SCALE_COEF 0x3d1f

Bit(s)	R/W	Default	Description
31-0	R/W	0x0	scaler coef

10.2.3.27 VPP VD1 SCALER

Because there are many coefficients used in the vertical filter and horizontal filters, indirect access the coefficients of vertical filter and horizontal filter is used. For vertical filter, there are 33x4 coefficients For horizontal filter, there are 33x4 coefficients

VPP_SCALE_COEF_IDX 0x1d03

Bit(s)	R/W	Default	Description
15	R/W	0	index_inc : // unsigned , default = 0x0 ,index increment, if bit9 == 1 then (0: index increase 1, 1: index increase 2) else (index increase 2)
14	R/W	0	rd_cbus_coef_en : // unsigned , default = 0x0 ,1: read coef through cbus enable, just for debug purpose in case when we wanna check the coef in ram in correct or not
13	R/W	0	vf_sep_coef_en : // unsigned , default = 0x0 ,if true, vertical separated coef enable
9	R/W	0	high_reso_en : // unsigned , default = 0x0 ,if true, use 9bit resolution coef, other use 8bit resolution coef
8:7	R/W	0	type_index : // unsigned , default = 0x0 ,type of index, 00: vertical coef, 01: vertical chroma coef: 10: horizontal coef, 11: resevered
6:0	R/W	0	coef_index : // unsigned , default = 0x0 ,coef index

VPP_SCALE_COEF 0x1d04

Bit(s)	R/W	Default	Description
31:24	R/W	0	coef0 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter
23:16	R/W	0	coef1 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter

15:8	R/W	0	coef2 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter
7 :0	R/W	0	coef3 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter

VPP_VSC_REGION12_STARTP 0x1d05

Bit(s)	R/W	Default	Description
28:16	R/W	0	region1_startp : //unsigned , default = 0 ,region1 startp
12:0	R/W	0	region2_startp : //unsigned , default = 0 ,region2 startp

VPP_VSC_REGION34_STARTP 0x1d06

Bit(s)	R/W	Default	Description
28:16	R/W	0	region3_startp : //unsigned , default = 0x0438,region3 startp
12:0	R/W	0	region4_startp : //unsigned , default = 0x0438,region4 startp

VPP_VSC_REGION4_ENDP 0x1d07

Bit(s)	R/W	Default	Description
12:0	R/W	13	region4_endp : //unsigned , default = 13'd1079 ,region4 endp

VPP_VSC_START_PHASE_STEP 0x1d08

Bit(s)	R/W	Default	Description
27:24	R/W	1	integer_part : //unsigned , default = 1,vertical start phase step, (source/dest)*(2^24),integer part of step
23:0	R/W	0	fraction_part : //unsigned , default = 0,vertical start phase step, (source/dest)*(2^24),fraction part of step

VPP_VSC_REGION0_PHASE_SLOPE 0x1d09

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,vertical scaler region0 phase slope,region0 phase slope

VPP_VSC_REGION1_PHASE_SLOPE 0x1d0a

Bit(s)	R/W	Default	Description
24:0	R/W	0	region1_phase_slope : //signed , default = 0,region1 phase slope

VPP_VSC_REGION3_PHASE_SLOPE 0x1d0b

Bit(s)	R/W	Default	Description
24:0	R/W	0	region3_phase_slope : //signed , default = 0,region3 phase slope

VPP_VSC_REGION4_PHASE_SLOPE 0x1d0c

Bit(s)	R/W	Default	Description
24:0	R/W	0	region4_phase_slope : //signed , default = 0,region4 phase slope

VPP_VSC_PHASE_CTRL 0x1d0d

Bit(s)	R/W	Default	Description
18:17	R/W	0	vsc_double_line_mode : //unsigned , default = 0, double line mode, input/output line width of vscaler becomes 2X, so only 2 line buffer in this case, use for 3D line by line interleave scaling bit1 true, double the input width and half input height, bit0 true, change line buffer 2 lines instead of 4 lines
16	R.O	0	prog_interlace : //unsigned , default = 0,0: progressive output, 1: interlace output
15	R/W	0	vsc_bot_l0_out_en : //unsigned , default = 0,vertical scaler output line0 in advance or not for bottom field
14:13	R/W	1	vsc_bot_rpt_l0_num : //unsigned , default = 1,vertical scaler initial repeat line0 number for bottom field
11:8	R/W	4	vsc_bot_ini_rcv_num : //unsigned , default = 4,vertical scaler initial receiving number for bottom field
7	R/W	0	vsc_top_l0_out_en : //unsigned , default = 0,vertical scaler output line0 in advance or not for top field
6:5	R/W	1	vsc_top_rpt_l0_num : //unsigned , default = 1,vertical scaler initial repeat line0 number for top field
3:0	R/W	4	vsc_top_ini_rcv_num : //unsigned , default = 4,vertical scaler initial receiving number for top field

VPP_VSC_INI_PHASE 0x1d0e

Bit(s)	R/W	Default	Description
31:16	R/W	0,	//unsigned , default = 0,vertical scaler field initial phase for bottom field Bit 15:0 //unsigned , default = 0,vertical scaler field initial phase for top field

VPP_HSC_REGION12_STARTP 0x1d10

Bit(s)	R/W	Default	Description
28:16	R/W	0	region1_startp : //unsigned , default = 0,region1 startp
12:0	R/W	0	region2_startp : //unsigned , default = 0,region2 startp

VPP_HSC_REGION34_STARTP 0x1d11

Bit(s)	R/W	Default	Description
28:16	R/W	0	region3 : startp //unsigned , default = 0x780,region3 startp
12:0	R/W	0	region4 : startp //unsigned , default = 0x780,region4 startp

VPP_HSC_REGION4_ENDP 0x1d12

Bit(s)	R/W	Default	Description
12:0	R/W	13	region4 : startp //unsigned , default = 13'd1919,region4 startp

horizontal start phase step, (source/dest)*(2^24)

VPP_HSC_START_PHASE_STEP 0x1d13

Bit(s)	R/W	Default	Description
27:24	R/W	1	integer_part : //unsigned , default = 1,integer part of step
23:0	R/W	0	fraction_part : //unsigned , default = 0,fraction part of step

VPP_HSC_REGION0_PHASE_SLOPE 0x1d14

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region0 phase slope

VPP_HSC_REGION1_PHASE_SLOPE 0x1d15

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region1 phase slope

VPP_HSC_REGION3_PHASE_SLOPE 0x1d16

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region3 phase slope

VPP_HSC_REGION4_PHASE_SLOPE 0x1d17

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region4 phase slope

VPP_HSC_PHASE_CTRL 0x1d18

Bit(s)	R/W	Default	Description
22:21	R/W	1	hsc_rpt_p0_num0 : //unsigned , default = 1 ,horizontal scaler initial repeat pixel0 number0
19:16	R/W	4	hsc_ini_rcv_num0 : //unsigned , default = 4 ,horizontal scaler initial receiving number0
15:0	R/W	0	hsc_ini_phase0 : //unsigned , default = 0 ,horizontal scaler top field initial phase0

VPP_SC_MISC 0x1d19

Bit(s)	R/W	Default	Description
22	R/W	0	hsc_len_div2_en : //unsigned , default = 0 ,if true, divide VSC line length 2 as the HSC input length, othwise VSC length length is the same as the VSC line length just for special usage, more flexibilityi

21	R/W	0	lbuf_mode : //unsigned , default = 0 ,if true, prevsc uses lin buffer, otherwise prevsc does not use line buffer, it should be same as prevsc_en
20	R/W	0	prehsc_en : //unsigned , default = 0 ,prehsc_en
19	R/W	0	prevsc_en : //unsigned , default = 0 ,prevsc_en
18	R/W	0	vsc_en : //unsigned , default = 0 ,vsc_en
17	R/W	0	hsc_en : //unsigned , default = 0 ,hsc_en
16	R/W	0	sc_top_en : //unsigned , default = 0 ,scale_top_en
15	R/W	0	sc_vd_en : //unsigned , default = 0 ,video1 scale out enable
12	R/W	1	hsc_nonlinear_4region_en : //unsigned , default = 1 ,if true, region0,region4 are nonlinear regions, otherwise they are not scaling regions, for horizontal scaler
10:8	R/W	0	hsc_bank_length : //unsigned , default = 0 ,horizontal scaler bank length
5	R/W	4	vsc_phase_field_mode : //unsigned , default = 4 ,vertical scaler phase field mode, if true, disable the opposite parity line output, more bandwith needed if output 1080i
4	R/W	0	vsc_nonlinear_4region_en : //unsigned , default = 0 ,if true, region0,region4 are nonlinear regions, otherwise they are not scaling regions, for vertical scaler
2:0	R/W	4	vsc_bank_length : //unsigned , default = 4 ,vertical scaler bank length

VPP_SCO_FIFO_CTRL 0x1d33

Bit(s)	R/W	Default	Description
27:16	R/W	0	sco_fifo_line_lenm1 : //unsigned , default = 0xfff, scale out fifo line length minus 1
12:0	R/W	0	sco_fifo_size : //unsigned , default = 0x200, scale out fifo size (actually only bit 11:1 is valid, 11:1, max 1024), always even number

for 3D quincunx sub-sampling and horizontal pixel by pixel 3D interleaving

VPP_HSC_PHASE_CTRL1 0x1d34

Bit(s)	R/W	Default	Description
27:24	R/W	0	prehsc_mode : //unsigned , default = 0,prehsc_mode, bit 3:2, prehsc odd line interp mode, bit 1:0, prehsc even line interp mode, each 2bit, 00: pix0+pix1/2, average, 01: pix1, 10: pix0
23	R/W	0	hsc_double_pix_mode : //unsigned , default = 0,horizontal scaler double pixel mode
22:21	R/W	1	hsc_rpt_p0_num1 : //unsigned , default = 1,horizontal scaler initial repeat pixel0 number1
19:16	R/W	4	hsc_ini_rcv_num1 : //unsigned , default = 4,horizontal scaler initial receiving number1
15:0	R/W	0	hsc_ini_phase1 : //unsigned , default = 0,horizontal scaler top field initial phase1

for 3D quincunx sub-sampling

VPP_HSC_INI_PAT_CTRL 0x1d35

Bit(s)	R/W	Default	Description
31:24	R/W	0	prehsc_pattern : //unsigned , default = 0, prehsc pattern, each patten 1 bit, from lsb -> msb
22:20	R/W	0	prehsc_pat_star : //unsigned , default = 0, prehsc pattern start
18:16	R/W	0	prehsc_pat_end : //unsigned , default = 0, prehsc pattern end
15:8	R/W	0	hsc_pattern : //unsigned , default = 0, hsc pattern, each patten 1 bit, from lsb -> msb
6:4	R/W	0	hsc_pat_start : //unsigned , default = 0, hsc pattern start
2:0	R/W	0	hsc_pat_end : //unsigned , default = 0, hsc pattern end

VPP_SC_GCLK_CTRL 0x1d35

Bit(s)	R/W	Default	Description
15:0	R/W	0	vpp_sc_gclk_ctrl : //unsigned , default = 0 ,scale clock gate

10.2.3.28 VPP VD2 SCALER

Because there are many coefficients used in the vertical filter and horizontal filters, indirect access the coefficients of vertical filter and horizontal filter is used. For vertical filter, there are 33x4 coefficients For horizontal filter, there are 33x4 coefficients

VD2_SCALE_COEF_IDX 0x3943

Bit(s)	R/W	Default	Description
15	R/W	0	index_inc : // unsigned , default = 0x0 ,index increment, if bit9 == 1 then (0: index increase 1, 1: index increase 2) else (index increase 2)
14	R/W	0	rd_cbus_coef_en : // unsigned , default = 0x0 ,1: read coef through cbus enable, just for debug purpose in case when we wanna check the coef in ram in correct or not
13	R/W	0	vf_sep_coef_en : // unsigned , default = 0x0 ,if true, vertical separated coef enable
9	R/W	0	high_reso_en : // unsigned , default = 0x0 ,if true, use 9bit resolution coef, other use 8bit resolution coef
8:7	R/W	0	type_index : // unsigned , default = 0x0 ,type of index, 00: vertical coef, 01: vertical chroma coef: 10: horizontal coef, 11: resevered
6:0	R/W	0	coef_index : // unsigned , default = 0x0 ,coef index

VD2_SCALE_COEF 0x3944

Bit(s)	R/W	Default	Description
31:24	R/W	0	coef0 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter
23:16	R/W	0	coef1 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter
15:8	R/W	0	coef2 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter

7 :0	R/W	0	coef3 : //signed , default = 0x0 , coefficients for vertical filter and horizontal filter

VD2_VSC_REGION12_STARTP 0x3945

Bit(s)	R/W	Default	Description
28:16	R/W	0	region1_startp : //unsigned , default = 0 ,region1 startp
12:0	R/W	0	region2_startp : //unsigned , default = 0 ,region2 startp

VD2_VSC_REGION34_STARTP 0x3946

Bit(s)	R/W	Default	Description
28:16	R/W	0	region3_startp : //unsigned , default = 0x0438,region3 startp
12:0	R/W	0	region4_startp : //unsigned , default = 0x0438,region4 startp

VD2_VSC_REGION4_ENDP 0x3947

Bit(s)	R/W	Default	Description
12:0	R/W	13	region4_endp : //unsigned , default = 13'd1079 ,region4 endp

VD2_VSC_START_PHASE_STEP 0x3948

Bit(s)	R/W	Default	Description
27:24	R/W	1	integer_part : //unsigned , default = 1,vertical start phase step, (source/dest)*(2^24),integer part of step
23:0	R/W	0	fraction_part : //unsigned , default = 0,vertical start phase step, (source/dest)*(2^24),fraction part of step

VD2_VSC_REGION0_PHASE_SLOPE 0x3949

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,vertical scaler region0 phase slope,region0 phase slope

VD2_VSC_REGION1_PHASE_SLOPE 0x394a

Bit(s)	R/W	Default	Description
24:0	R/W	0	region1_phase_slope : //signed , default = 0,region1 phase slope

VD2_VSC_REGION3_PHASE_SLOPE 0x394b

Bit(s)	R/W	Default	Description
24:0	R/W	0	region3_phase_slope : //signed , default = 0,region3 phase slope

VD2_VSC_REGION4_PHASE_SLOPE 0x394c

Bit(s)	R/W	Default	Description
24:0	R/W	0	region4_phase_slope : //signed , default = 0,region4 phase slope

VD2_VSC_PHASE_CTRL 0x394d

Bit(s)	R/W	Default	Description
18:17	R/W	0	vsc_double_line_mode : //unsigned , default = 0, double line mode, input/output line width of vscaler becomes 2X, so only 2 line buffer in this case, use for 3D line by line interleave scaling bit1 true, double the input width and half input height, bit0 true, change line buffer 2 lines instead of 4 lines
16	R.O	0	prog_interlace : //unsigned , default = 0,0: progressive output, 1: interlace output
15	R/W	0	vsc_bot_l0_out_en : //unsigned , default = 0,vertical scaler output line0 in advance or not for bottom field
14:13	R/W	1	vsc_bot_rpt_l0_num : //unsigned , default = 1,vertical scaler initial repeat line0 number for bottom field
11:8	R/W	4	vsc_bot_ini_rcv_num : //unsigned , default = 4,vertical scaler initial receiving number for bottom field
7	R/W	0	vsc_top_l0_out_en : //unsigned , default = 0,vertical scaler output line0 in advance or not for top field
6:5	R/W	1	vsc_top_rpt_l0_num : //unsigned , default = 1,vertical scaler initial repeat line0 number for top field
3:0	R/W	4	vsc_top_ini_rcv_num : //unsigned , default = 4,vertical scaler initial receiving number for top field

VD2_VSC_INI_PHASE 0x394e

Bit(s)	R/W	Default	Description
31:16	R/W	0,	//unsigned , default = 0,vertical scaler field initial phase for bottom field Bit 15:0 //unsigned , default = 0,vertical scaler field initial phase for top field

VD2_HSC_REGION12_STARTP 0x394f

Bit(s)	R/W	Default	Description
28:16	R/W	0	region1_startp : //unsigned , default = 0,region1 startp
12:0	R/W	0	region2_startp : //unsigned , default = 0,region2 startp

VD2_HSC_REGION34_STARTP 0x3950

Bit(s)	R/W	Default	Description
28:16	R/W	0	region3 : startp //unsigned , default = 0x780,region3 startp
12:0	R/W	0	region4 : startp //unsigned , default = 0x780,region4 startp

VD2_HSC_REGION4_ENDP 0x3951

Bit(s)	R/W	Default	Description
12:0	R/W	13	region4 : startp //unsigned , default = 13'd1919,region4 startp

horizontal start phase step, (source/dest)*(2^24)

VD2_HSC_START_PHASE_STEP 0x3952

Bit(s)	R/W	Default	Description
27:24	R/W	1	integer_part : //unsigned , default = 1,integer part of step
23:0	R/W	0	fraction_part : //unsigned , default = 0,fraction part of step

VD2_HSC_REGION0_PHASE_SLOPE 0x3953

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region0 phase slope

VD2_HSC_REGION1_PHASE_SLOPE 0x3954

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region1 phase slope

VD2_HSC_REGION3_PHASE_SLOPE 0x3955

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region3 phase slope

VD2_HSC_REGION4_PHASE_SLOPE 0x3956

Bit(s)	R/W	Default	Description
24:0	R/W	0	region0_phase_slope : //signed , default = 0,region4 phase slope

VD2_HSC_PHASE_CTRL 0x3957

Bit(s)	R/W	Default	Description
22:21	R/W	1	hsc_rpt_p0_num0 : //unsigned , default = 1 ,horizontal scaler initial repeat pixel0 number0
19:16	R/W	4	hsc_ini_rcv_num0 : //unsigned , default = 4 ,horizontal scaler initial receiving number0
15:0	R/W	0	hsc_ini_phase0 : //unsigned , default = 0 ,horizontal scaler top field initial phase0

VD2_SC_MISC 0x3958

Bit(s)	R/W	Default	Description
22	R/W	0	hsc_len_div2_en : //unsigned , default = 0 ,if true, divide VSC line length 2 as the HSC input length, othwise VSC length length is the same as the VSC line length just for special usage, more flexibilityi
21	R/W	0	lbuf_mode : //unsigned , default = 0 ,if true, prevsc uses lin buffer, otherwise prevsc does not use line buffer, it should be same as prevsc_en
20	R/W	0	prehsc_en : //unsigned , default = 0 ,prehsc_en
19	R/W	0	prevsc_en : //unsigned , default = 0 ,prevsc_en

Bit(s)	R/W	Default	Description
18	R/W	0	vsc_en : //unsigned , default = 0 ,vsc_en
17	R/W	0	hsc_en : //unsigned , default = 0 ,hsc_en
16	R/W	0	sc_top_en : //unsigned , default = 0 ,scale_top_en
15	R/W	0	sc_vd_en : //unsigned , default = 0 ,video1 scale out enable
12	R/W	1	hsc_nonlinear_4region_en : //unsigned , default = 1 ,if true, region0,region4 are nonlinear regions, otherwise they are not scaling regions, for horizontal scaler
10:8	R/W	0	hsc_bank_length : //unsigned , default = 0 ,horizontal scaler bank length
5	R/W	4	vsc_phase_field_mode : //unsigned , default = 4 ,vertical scaler phase field mode, if true, disable the opposite parity line output, more bandwith needed if output 1080i
4	R/W	0	vsc_nonlinear_4region_en : //unsigned , default = 0 ,if true, region0,region4 are nonlinear regions, otherwise they are not scaling regions, for vertical scaler
2:0	R/W	4	vsc_bank_length : //unsigned , default = 4 ,vertical scaler bank length

VD2_SCO_FIFO_CTRL 0x3959

Bit(s)	R/W	Default	Description
27:16	R/W	0	sco_fifo_line_lenm1 : //unsigned , default = 0xfff, scale out fifo line length minus 1
12:0	R/W	0	sco_fifo_size : //unsigned , default = 0x200, scale out fifo size (actually only bit 11:1 is valid, 11:1, max 1024), always even number

for 3D quincunx sub-sampling and horizontal pixel by pixel 3D interleaving

VD2_HSC_PHASE_CTRL1 0x395a

Bit(s)	R/W	Default	Description
27:24	R/W	0	prehsc_mode : //unsigned , default = 0,prehsc_mode, bit 3:2, prehsc odd line interp mode, bit 1:0, prehsc even line interp mode, each 2bit, 00: pix0+pix1/2, average, 01: pix1, 10: pix0
23	R/W	0	hsc_double_pix_mode : //unsigned , default = 0,horizontal scaler double pixel mode
22:21	R/W	1	hsc_rpt_p0_num1 : //unsigned , default = 1,horizontal scaler initial repeat pixel0 number1
19:16	R/W	4	hsc_ini_rcv_num1 : //unsigned , default = 4,horizontal scaler initial receiving number1
15:0	R/W	0	hsc_ini_phase1 : //unsigned , default = 0,horizontal scaler top field initial phase1

for 3D quincunx sub-sampling

VD2_HSC_INI_PAT_CTRL 0x395b

Bit(s)	R/W	Default	Description
31:24	R/W	0	prehsc_pattern : //unsigned , default = 0, prehsc pattern, each patten 1 bit, from lsb -> msb
22:20	R/W	0	prehsc_pat_star : //unsigned , default = 0, prehsc pattern start

18:16	R/W	0	prehsc_pat_end : //unsigned , default = 0, prehsc pattern end
15:8	R/W	0	hsc_pattern : //unsigned , default = 0, hsc pattern, each patten 1 bit, from lsb -> msb
6:4	R/W	0	hsc_pat_start : //unsigned , default = 0, hsc pattern start
2:0	R/W	0	hsc_pat_end : //unsigned , default = 0, hsc pattern end

VD2_SC_GCLK_CTRL 0x395c

Bit(s)	R/W	Default	Description
15:0	R/W	0	vpp_sc_gclk_ctrl : //unsigned , default = 0 ,scale clock gate

10.2.3.29 Supper Scale/ Sharpness

SRSHARP0_SHARP_HVSIZE 0x3e00

Bit(s)	R/W	Default	Description
28:16	R/W	0d1920	reg_pknr_hsize : . unsigned , default = 1920
12: 0	R/W	0d1080	reg_pknr_vsize : . unsigned , default = 1080

SRSHARP0_SHARP_HVBLANK_NUM 0x3e01

Bit(s)	R/W	Default	Description
23: 16	R/W	0d8	reg_deband_hblank : . unsigned , default = 8
15: 8	R/W	0d20	reg_pknr_hblank_num : . unsigned , default = 20
7: 0	R/W	0d60	reg_pknr_vblank_num : . unsigned , default = 60

SRSHARP0_NR_GAUSSIAN_MODE 0x3e02

Bit(s)	R/W	Default	Description
4	R/W	0d1	reg_nr_gau_ymode : : 0 3x3 filter; 1: 5x5 filter . unsigned , default = 1
0	R/W	0d1	reg_nr_gau_cmode : : 0 3x3 filter; 1: 5x5 filter . unsigned , default = 1

SRSHARP0_PK_CON_2CIRHPGAIN_TH_RATE 0x3e05

Bit(s)	R/W	Default	Description
31:24	R/W	0d25	reg_pk_cirhpcon2gain0 : : threshold0 of curve to map hpcon to hpgain for circle hp filter (all 8 direction same). 0~255.. unsigned , default = 25
23:16	R/W	0d60	reg_pk_cirhpcon2gain1 : : threshold1 of curve to map hpcon to hpgain for circle hp filter (all 8 direction same). 0~255.. unsigned , default = 60
15: 8	R/W	0d80	reg_pk_cirhpcon2gain5 : : rate0 (for hpcon<th0) of curve to map hpcon to hpgain for circle hp filter (all 8 direction same). 0~255.. unsigned , default = 80
7: 0	R/W	0d20	reg_pk_cirhpcon2gain6 : : rate1 (for hpcon>th1) of curve to map hpcon to hpgain for circle hp filter (all 8 direction same). 0~255.. unsigned , default = 20

SRSHARP0_PK_CON_2CIRHPGAIN_LIMIT 0x3e06

Bit(s)	R/W	Default	Description
31:24	R/W	0d96	reg_pk_cirhpcon2gain2 : : level limit(for hpcon<th0) of curve to map hpcon to hpgain for circle hp filter (all 8 direction same). 0~255.. unsigned , default = 96
23:16	R/W	0d96	reg_pk_cirhpcon2gain3 : : level limit(for th0<hpcon<th1) of curve to map hpcon to hpgain for circle hp filter (all 8 direction same). 0~255.. unsigned , default = 96
15: 8	R/W	0d5	reg_pk_cirhpcon2gain4 : : level limit(for hpcon>th1) of curve to map hpcon to hpgain for circle hp filter (all 8 direction same). 0~255.. unsigned , default = 5

SRSHARP0_PK_CON_2CIRBPGAIN_TH_RATE 0x3e07

Bit(s)	R/W	Default	Description
31:24	R/W	0d20	reg_pk_cirbpcon2gain0 : : threshold0 of curve to map bpcon to bpgain for circle bp filter (all 8 direction same). 0~255.. unsigned , default = 20
23:16	R/W	0d50	reg_pk_cirbpcon2gain1 : : threshold1 of curve to map bpcon to bpgain for circle bp filter (all 8 direction same).. unsigned , default = 50
15: 8	R/W	0d50	reg_pk_cirbpcon2gain5 : : rate0 (for bpcon<th0) of curve to map bpcon to bpgain for circle bp filter (all 8 direction same). 0~255.. unsigned , default = 50
7: 0	R/W	0d25	reg_pk_cirbpcon2gain6 : : rate1 (for bpcon>th1) of curve to map bpcon to bpgain for circle bp filter (all 8 direction same). 0~255.. unsigned , default = 25

SRSHARP0_PK_CON_2CIRBPGAIN_LIMIT 0x3e08

Bit(s)	R/W	Default	Description
31:24	R/W	0d40	reg_pk_cirbpcon2gain2 : : level limit(for bpcon<th0) of curve to map bpcon to bpgain for circle bp filter (all 8 direction same). 0~255.. unsigned , default = 40
23:16	R/W	0d40	reg_pk_cirbpcon2gain3 : : level limit(for th0<bpcon<th1) of curve to map bpcon to bpgain for circle bp filter (all 8 direction same). 0~255.. unsigned , default = 40
15: 8	R/W	0d5	reg_pk_cirbpcon2gain4 : : level limit(for bpcon>th1) of curve to map bpcon to bpgain for circle bp filter (all 8 direction same). 0~255.. unsigned , default = 5

SRSHARP0_PK_CON_2DRTHPGAIN_TH_RATE 0x3e09

Bit(s)	R/W	Default	Description
31:24	R/W	0d25	reg_pk_drthpcon2gain0 : : threshold0 of curve to map hpcon to hpgain for directional hp filter (best direction). 0~255.. unsigned , default = 25
23:16	R/W	0d60	reg_pk_drthpcon2gain1 : : threshold1 of curve to map hpcon to hpgain for directional hp filter (best direction). 0~255.. unsigned , default = 60
15: 8	R/W	0d80	reg_pk_drthpcon2gain5 : : rate0 (for hpcon<th0) of curve to map hpcon to hpgain for directional hp filter (best direction). 0~255.. unsigned , default = 80
7: 0	R/W	0d20	reg_pk_drthpcon2gain6 : : rate1 (for hpcon>th1) of curve to map hpcon to hpgain for directional hp filter (best direction). 0~255.. unsigned , default = 20

SRSHARP0_PK_CON_2DRTHPGAIN_LIMIT 0x3e0a

Bit(s)	R/W	Default	Description
31:24	R/W	0d90	reg_pk_drthpcon2gain2 : : level limit(for hpcon<th0) of curve to map hpcon to hpgain for directional hp filter (best direction).. unsigned , default = 90
23:16	R/W	0d96	reg_pk_drthpcon2gain3 : : level limit(for th0<hpcon<th1) of curve to map hpcon to hpgain for directional hp filter (best direction). 0~255.. unsigned , default = 96
15: 8	R/W	0d5	reg_pk_drthpcon2gain4 : : level limit(for hpcon>th1) of curve to map hpcon to hpgain for directional hp filter (best direction). 0~255.. unsigned , default = 5

SRSHARP0_PK_CON_2DRTBPGAIN_TH_RATE 0x3e0b

Bit(s)	R/W	Default	Description
31:24	R/W	0d20	reg_pk_drtbpcon2gain0 : : threshold0 of curve to map bpcon to bpgain for directional bp filter (best direction). 0~255.. unsigned , default = 20
23:16	R/W	0d50	reg_pk_drtbpcon2gain1 : : threshold1 of curve to map bpcon to bpgain for directional bp filter (best direction). 0~255.. unsigned , default = 50
15: 8	R/W	0d50	reg_pk_drtbpcon2gain5 : : rate0 (for bpcon<th0) of curve to map bpcon to bpgain for directional bp filter (best direction). 0~255.. unsigned , default = 50
7: 0	R/W	0d25	reg_pk_drtbpcon2gain6 : : rate1 (for bpcon>th1) of curve to map bpcon to bpgain for directional bp filter (best direction). 0~255.. unsigned , default = 25

SRSHARP0_PK_CON_2DRTBPGAIN_LIMIT 0x3e0c

Bit(s)	R/W	Default	Description
31:24	R/W	0d40	reg_pk_drtbpcon2gain2 : : level limit(for bpcon<th0) of curve to map bpcon to bpgain for directional bp filter (best direction). 0~255.. unsigned , default = 40
23:16	R/W	0d40	reg_pk_drtbpcon2gain3 : : level limit(for th0<bpcon<th1) of curve to map bpcon to bpgain for directional bp filter (best direction). 0~255.. unsigned , default = 40
15: 8	R/W	0d5	reg_pk_drtbpcon2gain4 : : level limit(for bpcon>th1) of curve to map bpcon to bpgain for directional bp filter (best direction). 0~255.. unsigned , default = 5

SRSHARP0_PK_CIRFB_LPF_MODE 0x3e0d

Bit(s)	R/W	Default	Description
29:28	R/W	0d1	reg_cirhp_horz_mode : : no horz filter on HP; 1: [1 2 1]/4; 2/3: [1 2 2 2 1]/8 . unsigned , default = 1
25:24	R/W	0d1	reg_cirhp_vert_mode : : no vert filter on HP; 1: [1 2 1]/4; 2/3: [1 2 2 2 1]/8 . unsigned , default = 1
21:20	R/W	0d1	reg_cirhp_diag_mode : : filter on HP; 1: [1 2 1]/4; . unsigned , default = 1
13:12	R/W	0d1	reg_cirbp_horz_mode : : no horz filter on BP; 1: [1 2 1]/4; 2/3: [1 2 2 2 1]/8 . unsigned , default = 1
9: 8	R/W	0d1	reg_cirbp_vert_mode : : no vert filter on BP; 1: [1 2 1]/4; 2/3: [1 2 2 2 1]/8 . unsigned , default = 1
5: 4	R/W	0d1	reg_cirbp_diag_mode : : filter on BP; 1: [1 2 1]/4; . unsigned , default = 1

SRSHARP0_PK_DRTFB_LPF_MODE 0x3e0e

Bit(s)	R/W	Default	Description
29:28	R/W	0d1	reg_drthp_horz_mode : : no horz filter on HP; 1: [1 2 1]/4; 2/3: [1 2 2 2 1]/8 2 . unsigned , default = 1
25:24	R/W	0d1	reg_drthp_vert_mode : : no vert filter on HP; 1: [1 2 1]/4; 2/3: [1 2 2 2 1]/8 2 . unsigned , default = 1
21:20	R/W	0d1	reg_drthp_diag_mode : : filter on HP; 1: [1 2 1]/4; 1 . unsigned , default = 1
13:12	R/W	0d1	reg_drtbp_horz_mode : : no horz filter on BP; 1: [1 2 1]/4; 2/3: [1 2 2 2 1]/8 2 . unsigned , default = 1
9: 8	R/W	0d1	reg_drtbp_vert_mode : : no vert filter on BP; 1: [1 2 1]/4; 2/3: [1 2 2 2 1]/8 2 . unsigned , default = 1
5: 4	R/W	0d1	reg_drtbp_diag_mode : : filter on BP; 1: [1 2 1]/4; 1 . unsigned , default = 1

SRSHARP0_PK_CIRFB_HP_CORING 0x3e0f

Bit(s)	R/W	Default	Description
21:16	R/W	0d4	reg_cirhp_horz_core : : coring of HP for Horz . unsigned , default = 4
13: 8	R/W	0d4	reg_cirhp_vert_core : : coring of HP for Vert . unsigned , default = 4
5: 0	R/W	0d4	reg_cirhp_diag_core : : coring of HP for Diag . unsigned , default = 4

SRSHARP0_PK_CIRFB_BP_CORING 0x3e10

Bit(s)	R/W	Default	Description
21:16	R/W	0d4	reg_cirbp_horz_core : : coring of HP for Horz . unsigned , default = 4
13: 8	R/W	0d4	reg_cirbp_vert_core : : coring of HP for Vert . unsigned , default = 4
5: 0	R/W	0d4	reg_cirbp_diag_core : : coring of HP for Diag . unsigned , default = 4

SRSHARP0_PK_DRTFB_HP_CORING 0x3e11

Bit(s)	R/W	Default	Description
21:16	R/W	0d4	reg_drthp_horz_core : : coring of HP for Horz . unsigned , default = 4
13: 8	R/W	0d4	reg_drthp_vert_core : : coring of HP for Vert . unsigned , default = 4
5: 0	R/W	0d4	reg_drthp_diag_core : : coring of HP for Diag . unsigned , default = 4

SRSHARP0_PK_DRTFB_BP_CORING 0x3e12

Bit(s)	R/W	Default	Description
21:16	R/W	0d4	reg_drtbp_horz_core : : coring of HP for Horz . unsigned , default = 4

Bit(s)	R/W	Default	Description
13: 8	R/W	0d4	reg_drtbp_vert_core : : coring of HP for Vert . unsigned , default = 4
5: 0	R/W	0d4	reg_drtbp_diag_core : : coring of HP for Diag . unsigned , default = 4

SRSHARP0_PK_CIRFB_BLEND_GAIN 0x3e13

Bit(s)	R/W	Default	Description
31:28	R/W	0d8	reg_hp_cir_hgain : : normalized 8 as '1' . unsigned , default = 8
27:24	R/W	0d8	reg_hp_cir_vgain : : normalized 8 as '1' . unsigned , default = 8
23:20	R/W	0d8	reg_hp_cir_dgain : : normalized 8 as '1' . unsigned , default = 8
15:12	R/W	0d8	reg_bp_cir_hgain : : normalized 8 as '1' . unsigned , default = 8
11: 8	R/W	0d8	reg_bp_cir_vgain : : normalized 8 as '1' . unsigned , default = 8
7: 4	R/W	0d8	reg_bp_cir_dgain : : normalized 8 as '1' . unsigned , default = 8

SRSHARP0_NR_ALPY_SSD_GAIN_OFST 0x3e14

Bit(s)	R/W	Default	Description
15: 8	R/W	0d16	reg_nr_alp0_ssd_gain : : gain to max ssd normalized 16 as '1' . unsigned , default = 16
5: 0	R/W	0x0	reg_nr_alp0_ssd_ofst : : offset to ssd before dividing to min_err . signed , default = -2

SRSHARP0_NR_ALP0Y_ERR2CURV_TH_RATE 0x3e15

Bit(s)	R/W	Default	Description
31:24	R/W	0d10	reg_nr_alp0_minerr_ypar0 : : threshold0 of curve to map mierr to alp0 for luma channel, this will be set value of flat region mierr that no need blur. 0~255.. unsigned , default = 10
23:16	R/W	0d25	reg_nr_alp0_minerr_ypar1 : : threshold1 of curve to map mierr to alp0 for luma channel,this will be set value of texture region mierr that can not blur.. unsigned , default = 25
15: 8	R/W	0d80	reg_nr_alp0_minerr_ypar5 : : rate0 (for mierr<th0) of curve to map mierr to alp0 for luma channel. the larger of the value, the deep of the slope. 0~255.. unsigned , default = 80
7: 0	R/W	0d64	reg_nr_alp0_minerr_ypar6 : : rate1 (for mierr>th1) of curve to map mierr to alp0 for luma channel. the larger of the value, the deep of the slope. 0~255.. unsigned , default = 64

SRSHARP0_NR_ALP0Y_ERR2CURV_LIMIT 0x3e16

Bit(s)	R/W	Default	Description
31:24	R/W	0d63	reg_nr_alp0_minerr_ypar2 : : level limit(for mierr<th0) of curve to map mierr to alp0 for luma channel, this will be set to alp0 that we can do for flat region. 0~255.. unsigned , default = 63

23:16	R/W	0d0	reg_nr_alp0_minerr_ypar3 : : level limit(for th0<mierr<th1) of curve to map mierr to alp0 for luma channel, this will be set to alp0 that we can do for misc region. 0~255.. unsigned , default = 0
15: 8	R/W	0d63	reg_nr_alp0_minerr_ypar4 : : level limit(for mierr>th1) of curve to map mierr to alp0 for luma channel, this will be set to alp0 that we can do for texture region. 0~255.. unsigned , default = 63

SRSHARP0_NR_ALP0C_ERR2CURV_TH_RATE 0x3e17

Bit(s)	R/W	Default	Description
31:24	R/W	0d10	reg_nr_alp0_minerr_cpar0 : : threshold0 of curve to map mierr to alp0 for chroma channel, this will be set value of flat region mierr that no need blur.. unsigned , default = 10
23:16	R/W	0d25	reg_nr_alp0_minerr_cpar1 : : threshold1 of curve to map mierr to alp0 for chroma channel,this will be set value of texture region mierr that can not blur.. unsigned , default = 25
15: 8	R/W	0d80	reg_nr_alp0_minerr_cpar5 : : rate0 (for mierr<th0) of curve to map mierr to alp0 for chroma channel. the larger of the value, the deep of the slope. 0~255.. unsigned , default = 80
7: 0	R/W	0d64	reg_nr_alp0_minerr_cpar6 : : rate1 (for mierr>th1) of curve to map mierr to alp0 for chroma channel. the larger of the value, the deep of the slope. 0~255.. unsigned , default = 64

SRSHARP0_NR_ALP0C_ERR2CURV_LIMIT 0x3e18

Bit(s)	R/W	Default	Description
31:24	R/W	0d63	reg_nr_alp0_minerr_cpar2 : : level limit(for mierr<th0) of curve to map mierr to alp0 for chroma channel, this will be set to alp0 that we can do for flat region. 0~255.. unsigned , default = 63
23:16	R/W	0d0	reg_nr_alp0_minerr_cpar3 : : level limit(for th0<mierr<th1) of curve to map mierr to alp0 for chroma channel, this will be set to alp0 that we can do for misc region. 0~255.. unsigned , default = 0
15: 8	R/W	0d63	reg_nr_alp0_minerr_cpar4 : : level limit(for mierr>th1) of curve to map mierr to alp0 for chroma channel, this will be set to alp0 that we can do for texture region. 0~255.. unsigned , default = 63

SRSHARP0_NR_ALP0_MIN_MAX 0x3e19

Bit(s)	R/W	Default	Description
29:24	R/W	0d2	reg_nr_alp0_ymin : : normalized to 64 as '1' . unsigned , default = 2
21:16	R/W	0d63	reg_nr_alp0_ymax : : normalized to 64 as '1' . unsigned , default = 63
13: 8	R/W	0d2	reg_nr_alp0_cmin : : normalized to 64 as '1' . unsigned , default = 2
5: 0	R/W	0d63	reg_nr_alp0_cmax : : normalized to 64 as '1' . unsigned , default = 63

SRSHARP0_NR_ALP1_MIERR_CORING 0x3e1a

Bit(s)	R/W	Default	Description

16	R/W	0d0	reg_nr_alp1_maxerr_mode : : 0 max err; 1: xerr . unsigned , default = 0
13: 8	R/W	0d0	reg_nr_alp1_core_rate : : normalized 64 as "1" . unsigned , default = 0
5: 0	R/W	0d3	reg_nr_alp1_core_ofst : : normalized 64 as "1" . signed , default = 3

SRSHARP0_NR_ALP1_ERR2CURV_TH_RATE 0x3e1b

Bit(s)	R/W	Default	Description
31:24	R/W	0d0	reg_nr_alp1_minerr_par0 : : threshold0 of curve to map mierr to alp1 for luma/chroma channel, this will be set value of flat region mierr that no need directional NR. 0~255.. unsigned , default = 0
23:16	R/W	0d24	reg_nr_alp1_minerr_par1 : : threshold1 of curve to map mierr to alp1 for luma/chroma channel,this will be set value of texture region mierr that can not do directional NR. 0~255.. unsigned , default = 24
15: 8	R/W	0d0	reg_nr_alp1_minerr_par5 : : rate0 (for mierr<th0) of curve to map mierr to alp1 for luma/chroma channel. the larger of the value, the deep of the slope.. unsigned , default = 0
7: 0	R/W	0d20	reg_nr_alp1_minerr_par6 : : rate1 (for mierr>th1) of curve to map mierr to alp1 for luma/chroma channel. the larger of the value, the deep of the slope. 0~255. unsigned , default = 20

SRSHARP0_NR_ALP1_ERR2CURV_LIMIT 0x3e1c

Bit(s)	R/W	Default	Description
31:24	R/W	0d0	reg_nr_alp1_minerr_par2 : : level limit(for mierr<th0) of curve to map mierr to alp1 for luma/chroma channel, this will be set to alp1 that we can do for flat region. 0~255.. unsigned , default = 0
23:16	R/W	0d16	reg_nr_alp1_minerr_par3 : : level limit(for th0<mierr<th1) of curve to map mierr to alp1 for luma/chroma channel, this will be set to alp1 that we can do for misc region. 0~255.. unsigned , default = 16
15: 8	R/W	0d63	reg_nr_alp1_minerr_par4 : : level limit(for mierr>th1) of curve to map mierr to alp1 for luma/chroma channel, this will be set to alp1 that we can do for texture region. 0~255.255 before. unsigned , default = 63

SRSHARP0_NR_ALP1_MIN_MAX 0x3e1d

Bit(s)	R/W	Default	Description
29:24	R/W	0d0	reg_nr_alp1_ymin : : normalized to 64 as '1' . unsigned , default = 0
21:16	R/W	0d63	reg_nr_alp1_ymax : : normalized to 64 as '1' . unsigned , default = 63
13: 8	R/W	0d0	reg_nr_alp1_cmin : : normalized to 64 as '1' . unsigned , default = 0
5: 0	R/W	0d63	reg_nr_alp1_cmax : : normalized to 64 as '1' . unsigned , default = 63

SRSHARP0_PK_ALP2_MIERR_CORING 0x3e1e

Bit(s)	R/W	Default	Description
16	R/W	0d1	reg_pk_alp2_maxerr_mode : : 0 max err; 1: xerr . unsigned , default = 1

13: 8	R/W	0d13	reg_pk_alp2_core_rate : : normalized 64 as "1" . unsigned , default = 13
5: 0	R/W	0d1	reg_pk_alp2_core_ofst : : normalized 64 as "1" . signed , default = 1

SRSHARP0_PK_ALP2_ERR2CURV_TH_RATE 0x3e1f

Bit(s)	R/W	Default	Description
31:24	R/W	0d0	reg_pk_alp2_minerr_par0 : : threshold0 of curve to map mierr to alp2 for luma channel, this will be set value of flat region mierr that no need peaking.. unsigned , default = 0
23:16	R/W	0d24	reg_pk_alp2_minerr_par1 : : threshold1 of curve to map mierr to alp2 for luma channel,this will be set value of texture region mierr that can not do peaking. 0~255.. unsigned , default = 24
15: 8	R/W	0d0	reg_pk_alp2_minerr_par5 : : rate0 (for mierr<th0) of curve to map mierr to alp2 for luma channel. the larger of the value, the deep of the slope. 0~255.. unsigned , default = 0
7: 0	R/W	0d20	reg_pk_alp2_minerr_par6 : : rate1 (for mierr>th1) of curve to map mierr to alp2 for luma channel. the larger of the value, the deep of the slope. 0~255.. unsigned , default = 20

SRSHARP0_PK_ALP2_ERR2CURV_LIMIT 0x3e20

Bit(s)	R/W	Default	Description
31:24	R/W	0d0	reg_pk_alp2_minerr_par2 : : level limit(for mierr<th0) of curve to map mierr to alp2 for luma channel, this will be set to alp2 that we can do for flat region. 0~255.. unsigned , default = 0
23:16	R/W	0d16	reg_pk_alp2_minerr_par3 : : level limit(for th0<mierr<th1) of curve to map mierr to alp2 for luma channel, this will be set to alp2 that we can do for misc region. 0~255.. unsigned , default = 16
15: 8	R/W	0d63	reg_pk_alp2_minerr_par4 : : level limit(for mierr>th1) of curve to map mierr to alp2 for luma channel, this will be set to alp2 that we can do for texture region. 0~255. default = 63;. unsigned , default = 255

SRSHARP0_PK_ALP2_MIN_MAX 0x3e21

Bit(s)	R/W	Default	Description
13: 8	R/W	0d0	reg_pk_alp2_min : : normalized to 64 as '1' . unsigned , default = 0
5: 0	R/W	0d63	reg_pk_alp2_max : : normalized to 64 as '1' . unsigned , default = 63

SRSHARP0_PK_FINALGAIN_HP_BP 0x3e22

Bit(s)	R/W	Default	Description
17: 16	R/W	0d0	reg_final_gain_rs : : right shift bits for the gain normalization, 0 normal to 32 as 1; 1 normal to 64 as 1; -2 normal to 8 as 1; -1 normal to 16 as 1 . signed , default = 0
15: 8	R/W	0d40	reg_hp_final_gain : : gain to highpass boost result (including directional/circle blending), normalized 32 as '1', 0~255. 1.25 * 32. unsigned , default = 40

7: 0	R/W	0d30	reg_bp_final_gain : : gain to bandpass boost result (including directional/circle blending), normalized 32 as '1', 0~255. 1.25 * 32. unsigned , default = 30

SRSHARP0_PK_OS_HORZ_CORE_GAIN 0x3e23

Bit(s)	R/W	Default	Description
31:24	R/W	0d8	reg_pk_os_hsidecore : : side coring (not to current pixel) to adaptive overshoot margin in horizontal direction. the larger of this value, the less overshoot admitted 0~255;. unsigned , default = 8
23:16	R/W	0d20	reg_pk_os_hsidegain : : side gain (not to current pixel) to adaptive overshoot margin in horizontal direction. normalized to 32 as '1'. 0~255;. unsigned , default = 20
15: 8	R/W	0d2	reg_pk_os_hmidcore : : midd coring (to current pixel) to adaptive overshoot margin in horizontal direction. the larger of this value, the less overshoot admitted 0~255;. unsigned , default = 2
7: 0	R/W	0d20	reg_pk_os_hmidgain : : midd gain (to current pixel) to adaptive overshoot margin in horizontal direction. normalized to 32 as '1'. 0~255;. unsigned , default = 20

SRSHARP0_PK_OS_VERT_CORE_GAIN 0x3e24

Bit(s)	R/W	Default	Description
31:24	R/W	0d8	reg_pk_os_vsidecore : : side coring (not to current pixel) to adaptive overshoot margin in vertical direction. the larger of this value, the less overshoot admitted 0~255;. unsigned , default = 8
23:16	R/W	0d20	reg_pk_os_vsidegain : : side gain (not to current pixel) to adaptive overshoot margin in vertical direction. normalized to 32 as '1'. 0~255;. unsigned , default = 20
15: 8	R/W	0d2	reg_pk_os_vmidcore : : midd coring (to current pixel) to adaptive overshoot margin in vertical direction. the larger of this value, the less overshoot admitted 0~255;. unsigned , default = 2
7: 0	R/W	0d20	reg_pk_os_vmidgain : : midd gain (to current pixel) to adaptive overshoot margin in vertical direction. normalized to 32 as '1'. 0~255;. unsigned , default = 20

SRSHARP0_PK_OS_ADPT_MISC 0x3e25

Bit(s)	R/W	Default	Description
31:24	R/W	0d40	reg_pk_os_minerr_core : : coring to minerr for adaptive overshoot margin. the larger of this value, the less overshoot admitted 0~255;. unsigned , default = 40
23:16	R/W	0d6	reg_pk_os_minerr_gain : : gain to minerr based adaptive overshoot margin. normalized to 64 as '1'. 0~255;. unsigned , default = 6
15: 8	R/W	0d200	reg_pk_os_adpt_max : : maximum limit adaptive overshoot margin (4x). 0~255; . unsigned , default = 200
7: 0	R/W	0d20	reg_pk_os_adpt_min : : minimun limit adaptive overshoot margin (1x). 0~255; . unsigned , default = 20

SRSHARP0_PK_OS_STATIC 0x3e26

Bit(s)	R/W	Default	Description

29:28	R/W	0d2	reg_pk_osh_mode : : 0~3: (2x+1) window in H direction . unsigned , default = 2
25:24	R/W	0d2	reg_pk_osv_mode : : 0~3: (2x+1) window in V direction . unsigned , default = 2
21:12	R/W	0d200	reg_pk_os_down : : static negative overshoot margin. 0~1023; . unsigned , default = 200
9: 0	R/W	0d200	reg_pk_os_up : : static positive overshoot margin. 0~1023; . unsigned , default = 200

SRSHARP0_PK_NR_ENABLE 0x3e27

Bit(s)	R/W	Default	Description
3: 2	R/W	0d0	reg_3d_mode : , 0: no 3D; 1: L/R; 2: T/B; 3: horizontal interleaved, dft = 0 //. unsigned , default = 0
1	R/W	0d1	reg_pk_en : . unsigned , default = 1
0	R/W	0d1	reg_nr_en : . unsigned , default = 1

SRSHARP0_PK_DRT_SAD_MISC 0x3e28

Bit(s)	R/W	Default	Description
31:24	R/W	0d24	reg_pk_sad_ver_gain : : gain to sad[4], 16 normalized to "1"; . unsigned , default = 24
23:16	R/W	0d24	reg_pk_sad_hor_gain : : gain to sad[0], 16 normalized to "1"; . unsigned , default = 24
10: 9	R/W	0d0	reg_pk_bias_diag : : bias towards diag . unsigned , default = 0
4: 0	R/W	0d24	reg_pk_drt_force : : force direction of drt peaking filter, h2b: 0:hp drt force, 1: bp drt force; 2: bp+hp drt force, 3: no force;. unsigned , default = 24

SRSHARP0_NR_TI_DNLP_BLEND 0x3e29

Bit(s)	R/W	Default	Description
10: 8	R/W	0d4	reg_dnlp_input_mode : : dnlp input options. 0: org_y; 1: gau_y; 2: gauadp_y; 3: edgadplpf_y; 4: nr_y;5: lti_y; 6: pk_y (before os);7: pk_y (after os). unsigned , default = 4
3: 2	R/W	0d1	reg_nr_cti_blend_mode : : blend mode of nr and lti result: 0: nr; 1:cti; 2: (nr+cti)/2; 3:cti + dlt_nr . unsigned , default = 1
1: 0	R/W	0d1	reg_nr_lti_blend_mode : : blend mode of nr and lti result: 0: nr; 1:lti; 2: (nr+lti)/2; 3:lti + dlt_nr . unsigned , default = 1

SRSHARP0_TI_DIR_CORE_ALPHA 0x3e2a

Bit(s)	R/W	Default	Description
29:24	R/W	0d10	reg_adp_lti_dir_alp_core_ofst : : ofst to min_err, alpha = (min_err - (max_err-min_err)*rate + ofst)/max_err*64; dft=10. unsigned , default = 10

19:16	R/W	0d0	reg_adp_lti_dir_alp_core_rate : : ofset to min_err, alpha = (min_err - (max_err-min_err)*rate + ofst)/max_err*64; dft=0/32. unsigned , default = 0
13: 8	R/W	0d0	reg_adp_lti_dir_alpmin : : min value of alpha, alpha = (min_err+x +ofst)/max_err*64; dft=10 . unsigned , default = 0
5: 0	R/W	0d63	reg_adp_lti_dir_alpmax : : max value of alpha, alpha = (min_err+x +ofst)/max_err*64; dft=63 . unsigned , default = 63

SRSHARP0_CTI_DIR_ALPHA 0x3e2b

Bit(s)	R/W	Default	Description
29:24	R/W	0d5	reg_adp_cti_dir_alp_core_ofst : : ofst to min_err, alpha = (min_err - (max_err-min_err)*rate + ofst)/max_err*64; dft=10. unsigned , default = 5
19:16	R/W	0d0	reg_adp_cti_dir_alp_core_rate : : ofset to min_err, alpha = (min_err - (max_err-min_err)*rate + ofst)/max_err*64; dft=0/32. unsigned , default = 0
13: 8	R/W	0d0	reg_adp_cti_dir_alpmin : : min value of alpha, alpha = (min_err +x+ofst)/max_err*64; dft=10 . unsigned , default = 0
5: 0	R/W	0d63	reg_adp_cti_dir_alpmax : : max value of alpha, alpha = (min_err +x+ofst)/max_err*64; dft=63 . unsigned , default = 63

SRSHARP0_LTI_CTI_DF_GAIN 0x3e2c

Bit(s)	R/W	Default	Description
29:24	R/W	0d16	reg_adp_lti_hdf_gain : : 8 normalized to "1"; default = 16 . unsigned , default = 16
21:16	R/W	0d12	reg_adp_lti_vdf_gain : : 8 normalized to "1"; default = 12 . unsigned , default = 12
13: 8	R/W	0d16	reg_adp_cti_hdf_gain : : 8 normalized to "1"; default = 16 . unsigned , default = 16
5: 0	R/W	0d12	reg_adp_cti_vdf_gain : : 8 normalized to "1"; default = 12 . unsigned , default = 12

SRSHARP0_LTI_CTI_DIR_AC_DBG 0x3e2d

Bit(s)	R/W	Default	Description
30	R/W	0d1	reg_adp_lti_dir_lpf : : 0: no lpf; 1: [1 2 2 2 1]/8 lpf . unsigned , default = 1
28	R/W	0d0	reg_adp_lti_dir_difmode : : 0: y_dif; 1: y_dif + (u_dif+v_dif)/2; . unsigned , default = 0
26	R/W	0d1	reg_adp_cti_dir_lpf : : 0: no lpf; 1: [1 2 2 2 1]/8 lpf dft=1 . unsigned , default = 1
25:24	R/W	0d0	reg_adp_cti_dir_difmode : : 0: (u_dif+v_dif); 1: y_dif/2 + (u_dif+v_dif)*3/4; 2: y_dif + (u_dif+v_dif)/2; 3: y_dif*2 (not recomended). unsigned , default = 0

23:22	R/W	0d3	reg_adp_hvlti_dcblend_mode : : 0: hlti_dc; 1:vlti_dc; 2: avg 3; blend on alpha . unsigned , default = 3
21:20	R/W	0d3	reg_adp_hvcti_dcblend_mode : : 0: hcti_dc; 1:vcti_dc; 2: avg 3; blend on alpha . unsigned , default = 3
19:18	R/W	0d3	reg_adp_hvlti_acblend_mode : : hlti_ac; 1:vlti_ac; 2: add 3;:adaptive to alpha . unsigned , default = 3
17:16	R/W	0d3	reg_adp_hvcti_acblend_mode : : hcti_ac; 1:vcti_ac; 2: add 3;: adaptive to alpha . unsigned , default = 3
14:12	R/W	0d0	reg_adp_hlti_debug : , for hlti debug, default = 0 . unsigned , default = 0
10: 8	R/W	0d0	reg_adp_vlti_debug : , for vlti debug, default = 0 . unsigned , default = 0
6: 4	R/W	0d0	reg_adp_hcti_debug : , for hcti debug, default = 0 . unsigned , default = 0
2: 0	R/W	0d0	reg_adp_vcti_debug : , for vcti debug, default = 0 . unsigned , default = 0

SRSHARP0_HCTI_FLT_CLP_DC 0x3e2e

Bit(s)	R/W	Default	Description
28	R/W	0d1	reg_adp_hcti_en : , 0: no cti, 1: new cti, default = 1 . unsigned , default = 1
27:26	R/W	0d3	reg_adp_hcti_vdn_flt : , 0: no lpf; 1:[0,2,4,2,0], 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
25:24	R/W	0d2	reg_adp_hcti_hdn_flt : , 0: no lpf; 1:[0, 0, 0, 4, 8, 4, 0, 0, 0], 2:[0, 0, 2, 4, 4, 4, 2, 0, 0], 3: [1, 2, 2, 2, 2, 2, 2, 2, 1], default = 2. unsigned , default = 2
23:22	R/W	0d3	reg_adp_hcti_ddn_flt : , 0: no lpf; 1:[0,2,4,2,0], 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
21:20	R/W	0d2	reg_adp_hcti_lpf0_flt : , 0:no filter; 1:sigma=0.75, 2: sigma = 1.0, 3: sigma = 1.5, default = 2 . unsigned , default = 2
19:18	R/W	0d2	reg_adp_hcti_lpf1_flt : , 0:no filter; 1:sigma= 2.0, 2: sigma = 3.0, 3: sigma = 4.0, default = 2 . unsigned , default = 2
17:16	R/W	0d2	reg_adp_hcti_lpf2_flt : , 0:no filter; 1:sigma=5.0, 2: sigma = 9.0, 3: sigma = 13.0, default = 2 . unsigned , default = 2
15:12	R/W	0d7	reg_adp_hcti_hard_clp_win : , window size, 0~8, default = 7 . unsigned , default = 7
11: 8	R/W	0d3	reg_adp_hcti_hard_win_min : , window size, 0~8, default = 3 . unsigned , default = 3
4	R/W	0d1	reg_adp_hcti_clp_mode : , 0: hard clip, 1: adaptive clip, default = 1 . unsigned , default = 1
2: 0	R/W	0d0	reg_adp_hcti_dc_mode : , 0:dn, 1:lpf0, 2:lpf1, 3:lpf2, 4: lpf3: 5: vdn result; 6/7:org, default = 0 . unsigned , default = 0

SRSHARP0_HCTI_BST_GAIN 0x3e2f

Bit(s)	R/W	Default	Description

31:24	R/W	0d80	reg_adp_hcti_bst_gain0 : : gain of the bandpass 0 (lpf1-lpf2)- LBP, default = 80 . unsigned , default = 80
23:16	R/W	0d96	reg_adp_hcti_bst_gain1 : : gain of the bandpass 1 (lpf0-lpf1)- BP, default = 96 . unsigned , default = 96
15: 8	R/W	0d64	reg_adp_hcti_bst_gain2 : : gain of the bandpass 2 (hdn-lpf0)- HP, default = 64 . unsigned , default = 64
7: 0	R/W	0d16	reg_adp_hcti_bst_gain3 : : gain of the unsharp band (yuvin-hdn) - US, default = 16 . unsigned , default = 16

SRSHARP0_HCTI_BST_CORE 0x3e30

Bit(s)	R/W	Default	Description
31:24	R/W	0d0	reg_adp_hcti_bst_core0 : : core of the bandpass 0 (lpf1-lpf2)- LBP, default = 0 . unsigned , default = 0
23:16	R/W	0d0	reg_adp_hcti_bst_core1 : : core of the bandpass 1 (lpf0-lpf1)- BP, default = 0 . unsigned , default = 0
15: 8	R/W	0d0	reg_adp_hcti_bst_core2 : : core of the bandpass 2 (hdn-lpf0)- HP, default = 0 . unsigned , default = 0
7: 0	R/W	0d0	reg_adp_hcti_bst_core3 : : core of the unsharp band (yuvin-hdn) - US, default = 0 . unsigned , default = 0

SRSHARP0_HCTI_CON_2_GAIN_0 0x3e31

Bit(s)	R/W	Default	Description
31:29	R/W	0d2	reg_adp_hcti_con_mode : : con mode 0:[0, 0,-1, 1, 0, 0, 0]+[0, 0, 0, 1,-1, 0, 0], 1: [0, 0,-1, 0, 1, 0, 0], 2: [0,-1, 0, 0, 0, 1, 0], 3:[-1, 0, 0, 0, 0, 0, 1], 4: .... default = 2. unsigned , default = 2
28:26	R/W	0d3	reg_adp_hcti_dx_mode : : dx mode 0: [-1 1 0]; 1~7: [-1 (2x+1)"0" 1], default = 3 . unsigned , default = 3
25:24	R/W	0d1	reg_adp_hcti_con_lpf : : lpf mode of the con: 0: [1 2 1]/4; 1:[1 2 2 2 1]/8, default = 1 . unsigned , default = 1
23:16	R/W	0d25	reg_adp_hcti_con_2_gain0 : , default = 25 . unsigned , default = 25
15: 8	R/W	0d60	reg_adp_hcti_con_2_gain1 : , default = 60 . unsigned , default = 60
7: 0	R/W	0d0	reg_adp_hcti_con_2_gain2 : 0;, default = 0 . unsigned , default = 0

SRSHARP0_HCTI_CON_2_GAIN_1 0x3e32

Bit(s)	R/W	Default	Description
31:24	R/W	0d96	reg_adp_hcti_con_2_gain3 : 96;, default = 96 . unsigned , default = 96
23:16	R/W	0d5	reg_adp_hcti_con_2_gain4 : 5;, default = 5 . unsigned , default = 5

15: 8	R/W	0d80	reg_adp_hcti_con_2_gain5 : 80;, default = 80 . unsigned , default = 80
7: 0	R/W	0d20	reg_adp_hcti_con_2_gain6 : 20;, default = 20 . unsigned , default = 20

SRSHARP0_HCTI_OS_MARGIN 0x3e33

Bit(s)	R/W	Default	Description
7: 0	R/W	0d0	reg_adp_hcti_os_margin : : margin for hcti overshoot, default = 0 . unsigned , default = 0

SRSHARP0_HLTI_FLT_CLP_DC 0x3e34

Bit(s)	R/W	Default	Description
28	R/W	0d1	reg_adp_hlti_en : , 0: no cti, 1: new cti, default = 1 . unsigned , default = 1
27:26	R/W	0d2	reg_adp_hlti_vdn_flt : , 0: no lpf; 1:[0,2,4,2,0], 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 2 . unsigned , default = 2
25:24	R/W	0d2	reg_adp_hlti_hdn_flt : , 0: no lpf; 1:[0, 0, 0, 4, 8, 4, 0, 0, 0], 2:[0, 0, 2, 4, 4, 4, 2, 0, 0], 3: [1, 2, 2, 2, 2, 2, 2, 2, 1], default = 2. unsigned , default = 2
23:22	R/W	0d2	reg_adp_hlti_ddn_flt : , 0: no lpf; 1:[0,2,4,2,0], 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 2 . unsigned , default = 2
21:20	R/W	0d2	reg_adp_hlti_lpf0_flt : , 0:no filter; 1:sigma=0.75, 2: sigma = 1.0, 3: sigma = 1.5, default = 2 . unsigned , default = 2
19:18	R/W	0d2	reg_adp_hlti_lpf1_flt : , 0:no filter; 1:sigma= 2.0, 2: sigma = 3.0, 3: sigma = 4.0, default = 2 . unsigned , default = 2
17:16	R/W	0d2	reg_adp_hlti_lpf2_flt : , 0:no filter; 1:sigma=5.0, 2: sigma = 9.0, 3: sigma = 13.0, default = 2 . unsigned , default = 2
15:12	R/W	0d2	reg_adp_hlti_hard_clp_win : , window size, 0~8, default = 2 . unsigned , default = 2
11: 8	R/W	0d1	reg_adp_hlti_hard_win_min : , window size, 0~8, default = 1 . unsigned , default = 1
4	R/W	0d0	reg_adp_hlti_clp_mode : , 0: hard clip, 1: adaptive clip, default = 0 . unsigned , default = 0
2: 0	R/W	0d4	reg_adp_hlti_dc_mode : , 0:dn, 1:lpf0, 2:lpf1, 3:lpf2, 4: lpf3: 5: vdn result; 6/7:org, default = 4 . unsigned , default = 4

SRSHARP0_HLTI_BST_GAIN 0x3e35

Bit(s)	R/W	Default	Description
31:24	R/W	0d40	reg_adp_hlti_bst_gain0 : : gain of the bandpass 0 (lpf1-lpf2)- LBP, default = 40 . unsigned , default = 40
23:16	R/W	0d48	reg_adp_hlti_bst_gain1 : : gain of the bandpass 1 (lpf0-lpf1)- BP, default = 48 . unsigned , default = 48

15: 8	R/W	0d32	reg_adp_hlti_bst_gain2 : : gain of the bandpass 2 (hdn-lpf0)- HP, default = 32 . unsigned , default = 32
7: 0	R/W	0d16	reg_adp_hlti_bst_gain3 : : gain of the unsharp band (yuvin-hdn) - US, default = 16 . unsigned , default = 16

SRSHARP0_HLTI_BST_CORE 0x3e36

Bit(s)	R/W	Default	Description
31:24	R/W	0d5	reg_adp_hlti_bst_core0 : : core of the bandpass 0 (lpf1-lpf2)- LBP, default = 5 . unsigned , default = 5
23:16	R/W	0d5	reg_adp_hlti_bst_core1 : : core of the bandpass 1 (lpf0-lpf1)- BP, default = 5 . unsigned , default = 5
15: 8	R/W	0d5	reg_adp_hlti_bst_core2 : : core of the bandpass 2 (hdn-lpf0)- HP, default = 5 . unsigned , default = 5
7: 0	R/W	0d3	reg_adp_hlti_bst_core3 : : core of the unsharp band (yuvin-hdn) - US, default = 3 . unsigned , default = 3

SRSHARP0_HLTI_CON_2_GAIN_0 0x3e37

Bit(s)	R/W	Default	Description
31:29	R/W	0d2	reg_adp_hlti_con_mode : : con mode 0:[0, 0,-1, 1, 0, 0, 0]+[0, 0, 0, 1,-1, 0, 0], 1: [0, 0,-1, 0, 1, 0, 0], 2: [0,-1, 0, 0, 0, 1, 0], 3:[-1, 0, 0, 0, 0, 0, 1], 4: ....., default = 2. unsigned , default = 2
28:26	R/W	0d3	reg_adp_hlti_dx_mode : : dx mode 0: [-1 1 0]; 1~7: [-1 (2x+1)"0" 1], default = 3 . unsigned , default = 3
25:24	R/W	0d1	reg_adp_hlti_con_lpf : : lpf mode of the con: 0: [1 2 1]/4; 1:[1 2 2 2 1]/8, default = 1 . unsigned , default = 1
23:16	R/W	0d25	reg_adp_hlti_con_2_gain0 : 25;, default = 25 . unsigned , default = 25
15: 8	R/W	0d60	reg_adp_hlti_con_2_gain1 : 60;, default = 60 . unsigned , default = 60
7: 0	R/W	0d90	reg_adp_hlti_con_2_gain2 : 0;, default = 90 . unsigned , default = 90

SRSHARP0_HLTI_CON_2_GAIN_1 0x3e38

Bit(s)	R/W	Default	Description
31:24	R/W	0d96	reg_adp_hlti_con_2_gain3 : 96;, default = 96 . unsigned , default = 96
23:16	R/W	0d95	reg_adp_hlti_con_2_gain4 : 5;, default = 95 . unsigned , default = 95
15: 8	R/W	0d80	reg_adp_hlti_con_2_gain5 : 80;, default = 80 . unsigned , default = 80
7: 0	R/W	0d20	reg_adp_hlti_con_2_gain6 : 20;, default = 20 . unsigned , default = 20

SRSHARP0_HLTI_OS_MARGIN 0x3e39

Bit(s)	R/W	Default	Description
7: 0	R/W	0d0	reg_adp_hlti_os_margin : : margin for hlti overshoot, default = 0 . unsigned , default = 0

SRSHARP0_VLTI_FLT_CON_CLP 0x3e3a

Bit(s)	R/W	Default	Description
14	R/W	0d1	reg_adp_vlti_en : : enable bit of vlti, default = 1 . unsigned , default = 1
13:12	R/W	0d3	reg_adp_vlti_hxn_flt : : 0: no dn; 1: [1 2 1]/4; 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
11:10	R/W	0d3	reg_adp_vlti_dxn_flt : : 0: no dn; 1: [1 2 1]/4; 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
9: 8	R/W	0d3	reg_adp_vlti_han_flt : : 0: no dn; 1: [1 2 1]/4; 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
7: 6	R/W	0d3	reg_adp_vlti_dan_flt : : 0: no dn; 1: [1 2 1]/4; 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
5: 4	R/W	0d2	reg_adp_vlti_dx_mode : : 0:[-1 1] 1:[-1 0 -1]; 2/3: [-1 0 0 0 -1], default = 2 . unsigned , default = 2
2	R/W	0d1	reg_adp_vlti_con_lpf : : lpf mode of the con: 0: [1 2 1]/4; 1:[1 2 2 2 1]/8, default = 1 . unsigned , default = 1
0	R/W	0d1	reg_adp_vlti_hard_clp_win : : window size; 0: 1x3 window; 1: 1x5 window, default = 1 . unsigned , default = 1

SRSHARP0_VLTI_BST_GAIN 0x3e3b

Bit(s)	R/W	Default	Description
23:16	R/W	0d32	reg_adp_vlti_bst_gain0 : : gain to boost filter [-1 2 -1];, default = 32 . unsigned , default = 32
15: 8	R/W	0d32	reg_adp_vlti_bst_gain1 : : gain to boost filter [-1 0 2 0 -1];, default = 32 . unsigned , default = 32
7: 0	R/W	0d32	reg_adp_vlti_bst_gain2 : : gain to boost filter usf, default = 32 . unsigned , default = 32

SRSHARP0_VLTI_BST_CORE 0x3e3c

Bit(s)	R/W	Default	Description
23:16	R/W	0d5	reg_adp_vlti_bst_core0 : : coring to boost filter [-1 2 -1];, default = 5 . unsigned , default = 5
15: 8	R/W	0d5	reg_adp_vlti_bst_core1 : : coring to boost filter [-1 0 2 0 -1];, default = 5 . unsigned , default = 5
7: 0	R/W	0d3	reg_adp_vlti_bst_core2 : : coring to boost filter usf, default = 3 . unsigned , default = 3

SRSHARP0_VLTI_CON_2_GAIN_0 0x3e3d

Bit(s)	R/W	Default	Description
31:24	R/W	0d25	reg_adp_vlti_con_2_gain0 : 25;, default = 25 . unsigned , default = 25
23:16	R/W	0d69	reg_adp_vlti_con_2_gain1 : 60;, default = 69 . unsigned , default = 60
15: 8	R/W	0d90	reg_adp_vlti_con_2_gain2 : 0;, default = 90 . unsigned , default = 90
7: 0	R/W	0d96	reg_adp_vlti_con_2_gain3 : 96;, default = 96 . unsigned , default = 96

SRSHARP0_VLTI_CON_2_GAIN_1 0x3e3e

Bit(s)	R/W	Default	Description
31:24	R/W	0d95	reg_adp_vlti_con_2_gain4 : 5;, default = 95 . unsigned , default = 95
23:16	R/W	0d80	reg_adp_vlti_con_2_gain5 : 80;, default = 80 . unsigned , default = 80
15: 8	R/W	0d20	reg_adp_vlti_con_2_gain6 : 20;, default = 20 . unsigned , default = 20
7: 0	R/W	0d0	reg_adp_vlti_os_margin : : margin for vlti overshoot, default = 0 . unsigned , default = 0

SRSHARP0_VCTI_FLT_CON_CLP 0x3e3f

Bit(s)	R/W	Default	Description
14	R/W	0d1	reg_adp_vcti_en : : enable bit of vlti, default = 1 . unsigned , default = 1
13:12	R/W	0d3	reg_adp_vcti_hxn_flt : : 0: no dn; 1: [1 2 1]/4; 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
11:10	R/W	0d3	reg_adp_vcti_dxn_flt : : 0: no dn; 1: [1 2 1]/4; 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
9: 8	R/W	0d3	reg_adp_vcti_han_flt : : 0: no dn; 1: [1 2 1]/4; 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
7: 6	R/W	0d3	reg_adp_vcti_dan_flt : : 0: no dn; 1: [1 2 1]/4; 2 : [1 2 2 2 1]/8 3:[1 0 2 0 1]/4, default = 3 . unsigned , default = 3
5: 4	R/W	0d2	reg_adp_vcti_dx_mode : : 0:[-1 1] 1:[-1 0 -1]; 2/3: [-1 0 0 0 -1], default = 2 . unsigned , default = 2
2	R/W	0d1	reg_adp_vcti_con_lpf : : lpf mode of the con: 0: [1 2 1]/4; 1:[1 2 2 2 1]/8, default = 1 . unsigned , default = 1
0	R/W	0d1	reg_adp_vcti_hard_clp_win : : window size; 0: 1x3 window; 1: 1x5 window, default = 1 . unsigned , default = 1

SRSHARP0_VCTI_BST_GAIN 0x3e40

Bit(s)	R/W	Default	Description
23:16	R/W	0d0	reg_adp_vcti_bst_gain0 : : gain to boost filter [-1 2 -1];, default = 0 . unsigned , default = 0
15: 8	R/W	0d0	reg_adp_vcti_bst_gain1 : : gain to boost filter [-1 0 2 0 -1];, default = 0 . unsigned , default = 0
7: 0	R/W	0d0	reg_adp_vcti_bst_gain2 : : gain to boost filter usf, default = 0 . unsigned , default = 0

SRSHARP0_VCTI_BST_CORE 0x3e41

Bit(s)	R/W	Default	Description
23:16	R/W	0d0	reg_adp_vcti_bst_core0 : : coring to boost filter [-1 2 -1];, default = 0 . unsigned , default = 0
15: 8	R/W	0d0	reg_adp_vcti_bst_core1 : : coring to boost filter [-1 0 2 0 -1];, default = 0 . unsigned , default = 0
7: 0	R/W	0d0	reg_adp_vcti_bst_core2 : : coring to boost filter usf, default = 0 . unsigned , default = 0

SRSHARP0_VCTI_CON_2_GAIN_0 0x3e42

Bit(s)	R/W	Default	Description
31:24	R/W	0d25	reg_adp_vcti_con_2_gain0 : 25;, default = 25 . unsigned , default = 25
23:16	R/W	0d60	reg_adp_vcti_con_2_gain1 : 60;, default = 60 . unsigned , default = 60
15: 8	R/W	0d90	reg_adp_vcti_con_2_gain2 : 0;, default = 90 . unsigned , default = 90
7: 0	R/W	0d96	reg_adp_vcti_con_2_gain3 : 96;, default = 96 . unsigned , default = 96

SRSHARP0_VCTI_CON_2_GAIN_1 0x3e43

Bit(s)	R/W	Default	Description
31:24	R/W	0d95	reg_adp_vcti_con_2_gain4 : 5;, default = 95 . unsigned , default = 95
23:16	R/W	0d80	reg_adp_vcti_con_2_gain5 : 80;, default = 80 . unsigned , default = 80
15: 8	R/W	0d20	reg_adp_vcti_con_2_gain6 : 20;, default = 20 . unsigned , default = 20
7: 0	R/W	0d0	reg_adp_vcti_os_margin : : margin for vcti overshoot, default = 0 . unsigned , default = 0

SRSHARP0_SHARP_3DLIMIT 0x3e44

Bit(s)	R/W	Default	Description
28:16	R/W	0d0	reg_3d_mid_width : ,width of left part of 3d input, dft = half size of input width default = 0 . unsigned , default = 960
12: 0	R/W	0d0	reg_3d_mid_height : ,height of left part of 3d input, dft = half size of input height default = 0 . unsigned , default = 540

SRSHARP0_DNLP_EN 0x3e45

Bit(s)	R/W	Default	Description
15:8	R/W	0d0	reg_dnlp_hblank: . unsigned , default = 8
0	R/W	0d1	reg_dnlp_en : . unsigned , default = 1

SRSHARP0_DEMO_CRTL 0x3e56

Bit(s)	R/W	Default	Description
18:17	R/W	0d2	demo_disp_position : . unsigned , default = 2
16	R/W	0d0	demo_hsvsharp_enable : . unsigned , default = 0
12: 0	R/W	0d360	demo_left_top_screen_width : : . unsigned , default = 360

SRSHARP0_SHARP_SR2_CTRL 0x3e57

Bit(s)	R/W	Default	Description
24	R/W	0	reg_sr2_bic_pknr_bypass : bypass peaking/TI/Cubic
23:22	R/W		reserved
21:16	R/W	24	sr2_pk_la_err_dis_rate, low angle and high angle error should not be no less than nearby_error* rate/64
15: 8	R/W	16	sr2_pk_sad_diag_gain, gain to sad[2] and sad[6], 16 normalized to 1
7	R/W	0	sr2_vert_outphs, vertical output pixel phase, 0: 0 phase; 1: 1/2 phase
6	R/W	0	sr2_horz_outphs, horizontal output pixel phase, 0: 0 phase; 1: 1/2 phase
5	R/W	0	sr2_vert_ratio , vertical scale ratio, 0-> 1:1; 1-> 1:2
4	R/W	0	sr2_horz_ratio , horizontal scale ratio, 0-> 1:1; 1-> 1:2
3	R/W	1	sr2_ bic_norm , normalization of bicubical: 0: 128; 1: 64
2	R/W	0	sr2_ enable , 1 to enable super scaler
1	R/W	0	sr2_ sharp_prc_lr_hbic,
0	R/W	0	sr2_ sharp_prc_lr, 1: LTI/CTI/NR/Peaking processing using LR grid. 0: on HR grid; 1:on LR grid, horizontally no upscale, but using simple bic.

SRSHARP0_SHARP_SR2_YBIC_HCOEF0 0x3e58

Bit(s)	R/W	Default	Description
31:24	R/W	0	sr2_y_bic_hcoeff03, signed
23:16	R/W	0	sr2_y_bic_hcoeff02, signed
15: 8	R/W	64	sr2_y_bic_hcoeff01, signed
7: 0	R/W	0	sr2_y_bic_hcoeff00, signed

SRSHARP0_SHARP_SR2_YBIC_HCOEF1 0x3e59

Bit(s)	R/W	Default	Description
31:24	R/W	-4	sr2_y_bic_hcoeff13 , signed
23:16	R/W	36	sr2_y_bic_hcoeff12 , signed
15: 8	R/W	36	sr2_y_bic_hcoeff11 , signed
7: 0	R/W	-4	sr2_y_bic_hcoeff10 , signed

SRSHARP0_SHARP_SR2_CBIC_HCOEF0 0x3e5a

Bit(s)	R/W	Default	Description
31:24	R/W	0	sr2_c_bic_hcoeff03 , signed
23:16	R/W	21	sr2_c_bic_hcoeff02 , signed
15: 8	R/W	22	sr2_c_bic_hcoeff01 , signed
7: 0	R/W	21	sr2_c_bic_hcoeff00 , signed

SRSHARP0_SHARP_SR2_CBIC_HCOEF1 0x3e5b

Bit(s)	R/W	Default	Description
31:24	R/W	-4	sr2_c_bic_hcoeff13 , signed
23:16	R/W	36	sr2_c_bic_hcoeff12 , signed
15: 8	R/W	36	sr2_c_bic_hcoeff11 , signed
7: 0	R/W	-4	sr2_c_bic_hcoeff10 , signed

SHARP_SR2_YBIC_VCOEF0 0x3e5c

Bit(s)	R/W	Default	Description
31:24	R/W	0	sr2_y_bic_vcoeff03 , signed
23:16	R/W	0	sr2_y_bic_vcoeff02 , signed
15: 8	R/W	64	sr2_y_bic_vcoeff01 , signed

7: 0	R/W	0	sr2_y_bic_vcoeff00 , signed

SRSHARP0_SHARP_SR2_YBIC_VCOEF1 0x3e5d

Bit(s)	R/W	Default	Description
31:24	R/W	-4	sr2_y_bic_vcoeff13 , signed
23:16	R/W	36	sr2_y_bic_vcoeff12 , signed
15: 8	R/W	36	sr2_y_bic_vcoeff11 , signed
7: 0	R/W	-4	sr2_y_bic_vcoeff10 , signed

SRSHARP0_SHARP_SR2_CBIC_VCOEF0 0x3e5e

Bit(s)	R/W	Default	Description
31:24	R/W	0	sr2_c_bic_vcoeff03 , signed
23:16	R/W	21	sr2_c_bic_vcoeff02 , signed
15: 8	R/W	22	sr2_c_bic_vcoeff01 , signed
7: 0	R/W	21	sr2_c_bic_vcoeff00 , signed

SRSHARP0_SHARP_SR2_CBIC_VCOEF1 0x3e5f

Bit(s)	R/W	Default	Description
31:24	R/W	-4	sr2_c_bic_vcoeff13 , signed
23:16	R/W	36	sr2_c_bic_vcoeff12 , signed
15: 8	R/W	36	sr2_c_bic_vcoeff11 , signed
7: 0	R/W	-4	sr2_c_bic_vcoeff10 , signed

SRSHARP0_SHARP_SR2_MISC 0x3e60

Bit(s)	R/W	Default	Description
31:2	R/W		reserved
1	R/W	0	sr2_cmpmux_bef , 0 : no swap for YUV/RGB; 1: swap for YUV/RGB, YUV/RGB->UVY/GBR
0	R/W	0	sr2_cmpmux_aft , 0 : no swap for YUV/RGB; 1: swap for YUV/RGB, UVY/GBR->YUV/RGB

SRSHARP0_SR3_SAD_CTRL 0x3e61

Bit(s)	R/W	Default	Description
31:30	R/W		reserved
29:24	R/W	0d6	reg_sr3_pk_sad_core_rate : rate of coring.
23:22	R/W		reserved

Bit(s)	R/W	Default	Description
21:16	R/W	0d6	reg_sr3_lti_sad_core_rate : rate of coring.
15:14	R/W		reserved
13:8	R/W	0d6	reg_sr3_cti_sad_core_rate : rate of coring.
7	R/W	0d1	reg_sr3_lti_hsad_mode: mode for hsad of lti calculation, 0:block based; 1: other sharp
6	R/W	0d1	reg_sr3_cti_hsad_mode: mode for hsad of cti calculation, 0:block based; 1: other sharp
5	R/W	0d1	reg_sr3_lti_dsad_mode: mode for dsad of lti calculation, 0:block based; 1: other sharp
4	R/W	0d1	reg_sr3_cti_dsad_mode: mode for dsad of cti calculation, 0:block based; 1: other sharp
3	R/W	0d1	reg_sr3_lti_vsad_mode: mode for vsad of lti calculation, 0:block based; 1: other sharp
2	R/W	0d1	reg_sr3_cti_vsad_mode: mode for vsad of cti calculation, 0:block based; 1: other sharp
1	R/W	0d1	reg_sr3_lti_hsad_hlpf: hlpf for hsad of lti calculation, 0:no hlpf; 1: with [1 2 1] hlpf.
0	R/W	0d1	reg_sr3_cti_hsad_hlpf: hlpf for hsad of cti calculation, 0:no hlpf; 1: with [1 2 1] hlpf.

SRSHARP0_SR3_PK_CTRL0 0x3e62

Bit(s)	R/W	Default	Description
31:12	R/W		reserved
11	R/W	0d1	reg_sr3_pk_sad_mode: mode for sad of peaking calculation, 0: block based; 1: other sharp.
10	R/W	0d1	reg_sr3_pk_hsad_hlpf: hlpf for hsad for peaking calculation,0:no hlpf; 1: with [1 2 2 2 1] hlpf.
9	R/W	0d1	reg_sr3_pk_vsad_hlpf: hlpf for vsad for peaking calculation,0:no hlpf; 1: with [1 2 2 2 1] hlpf.
8	R/W	0d1	reg_sr3_pk_dsad_hlpf: hlpf for dsad for peaking calculation,0:no hlpf; 1: with [1 2 2 2 1] hlpf.
7:6	R/W	0d3	reg_sr3_pk_hpdrt_mode: mode for HPdrt filter
5:4	R/W	0d3	reg_sr3_pk_bpdrt_mode: mode for BPdrt filter
3:2	R/W	0d3	reg_pk_drtbld_range: range of the min2 and min direction distance
1	R/W		reserved
0	R/W	0d0	reg_sr3_pk_ti_blend_mode: blend mode of the TI and PK result

SRSHARP0_SR3_PK_CTRL1 0x3e63

Bit(s)	R/W	Default	Description
31	R/W		reserved
30:28	R/W	0d1	reg_sr3_pk_Hp_hvcon_replace8_maxsad: replace HP hvcon by maxsad

Bit(s)	R/W	Default	Description
27	R/W		reserved
26:24	R/W	0d1	reg_sr3_pk_bp_hvcon_replace8_maxsad: replace BP hvcon by maxsad
23:16	R/W	0d32	reg_sr3_pk_hp_hvcon_replace8lv_gain: gain to local variant before calculating the hv gain for peaking.
15:8	R/W	0d32	reg_sr3_pk_bp_hvcon_replace8lv_gain: gain to local variant before calculating the hv gain for peaking.
7	R/W	0d1	reg_sr3_sad_intlev_mode: interleave detect xerr mode: 0 max; 1: sum
6	R/W	0d1	reg_sr3_sad_intlev_mode1: mode 1 of using diagonal protection: 1: with diagonal protection
5:0	R/W	0d12	reg_sr3_sad_intlev_gain: interleave detection for sad gain applied, normalized to 8 as 1

SRSHARP0_DEJ_CTRL 0x3e64

Bit(s)	R/W	Default	Description
15:8	R/W	8	reg_sr3_dejaggy_hblank
7:4	R/W		reserved
3:2	R/W	0d3	reg_sr3_dejaggy_sameside_prtct: enable of sr3 dejaggy same side curve protect from filter, [0] for proc path; [1] for ctrl path.
1	R/W	0d1	reg_sr3_dejaggy_sameside_mode: mode of sameside flag decision
0	R/W	0d1	reg_sr3_dejaggy_enable: enable of sr3 dejaggy

SRSHARP0_DEJ_ALPHA 0x3e65

Bit(s)	R/W	Default	Description
31:28	R/W	0d0	reg_sr3_dejaggy_ctrlchrm_alpha_1 : alpha for LR video LPF
27:24	R/W	0d15	reg_sr3_dejaggy_ctrlchrm_alpha_0 : alpha for LR video LPF
23:20	R/W	0d0	reg_sr3_dejaggy_ctrlluma_alpha_1 : alpha for LR video LPF
19:16	R/W	0d15	reg_sr3_dejaggy_ctrlluma_alpha_0 : alpha for LR video LPF
15:12	R/W	0d4	reg_sr3_dejaggy_procchrm_alpha_1: alpha for LR video LPF
11:8	R/W	0d6	reg_sr3_dejaggy_procchrm_alpha_0: alpha for LR video LPF
7:4	R/W	0d4	reg_sr3_dejaggy_procluma_alpha_1: alpha for LR video LPF
3:0	R/W	0d6	reg_sr3_dejaggy_procluma_alpha_0: alpha for LR video LPF

SRSHARP0_SR3_DRTLPF_EN 0x3e66

Bit(s)	R/W	Default	Description
31:15	R/W		reserved

Bit(s)	R/W	Default	Description
14:8	R/W	0d0	reg_pk_debug_edge
6:4	R/W	0d0	reg_sr3_drtlpf_theta_en
3	R/W		reserved
2:0	R/W	0d7	reg_sr3_drtlpf_enable: directional lpf on Y/U/V channels

SRSHARP0_SR3_DRTLPF_ALPHA_0 0x3e67

Bit(s)	R/W	Default	Description
31:30	R/W		reserved
29:24	R/W	0d9	reg_sr3_drtlpf_alpha_3
23:22	R/W		reserved
21:16	R/W	0d10	reg_sr3_drtlpf_alpha_2
15:14	R/W		reserved
13:8	R/W	0d11	reg_sr3_drtlpf_alpha_1
7:6	R/W		reserved
5:0	R/W	0d12	reg_sr3_drtlpf_alpha_0: directional lpf alpha coef for min_sad/max_sad compare

SRSHARP0_SR3_DRTLPF_ALPHA_1 0x3e68

Bit(s)	R/W	Default	Description
31:30	R/W		reserved
29:24	R/W	0d1	reg_sr3_drtlpf_alpha_7
23:22	R/W		reserved
21:16	R/W	0d4	reg_sr3_drtlpf_alpha_6
15:14	R/W		reserved
13:8	R/W	0d7	reg_sr3_drtlpf_alpha_5
7:6	R/W		reserved
5:0	R/W	0d8	reg_sr3_drtlpf_alpha_4: directional lpf alpha coef for min_sad/max_sad compare

SRSHARP0_SR3_DRTLPF_ALPHA_2 0x3e69

Bit(s)	R/W	Default	Description
31:30	R/W		reserved
29:24	R/W	0d0	reg_sr3_drtlpf_alpha_11
23:22	R/W		reserved

Bit(s)	R/W	Default	Description
21:16	R/W	0d0	reg_sr3_drtlpf_alpha_10
15:14	R/W		reserved
13:8	R/W	0d0	reg_sr3_drtlpf_alpha_9
7:6	R/W		reserved
5:0	R/W	0d0	reg_sr3_drtlpf_alpha_8: directional lpf alpha coef for min_sad/max_sad compare

SRSHARP0_SR3_DRTLPF_ALPHA_OFST 0x3e6a

Bit(s)	R/W	Default	Description
31:28	R/W	0	reg_sr3_drtlpf_alpha_ofst7
27:24	R/W	0	reg_sr3_drtlpf_alpha_ofst6
23:20	R/W	0	reg_sr3_drtlpf_alpha_ofst5
19:16	R/W	-2	reg_sr3_drtlpf_alpha_ofst4
15:12	R/W	0	reg_sr3_drtlpf_alpha_ofst3
11:8	R/W	0	reg_sr3_drtlpf_alpha_ofst2
7:4	R/W	0	reg_sr3_drtlpf_alpha_ofst1
3:0	R/W	-2	reg_sr3_drtlpf_alpha_ofst0: directional lpf alpha coef offset of each direction.

SRSHARP0_SR3_DERING_CTRL 0x3e6b

Bit(s)	R/W	Default	Description
31	R/W		reserved
30:28	R/W	1	reg_sr3_dering_enable: dering enable
27	R/W		reserved
26:24	R/W	3	reg_sr3_dering_varlpf_mode: local variant LPF mode. 0: no filter; 1: errosion 3x3; 2: 3x3 lpf; 3: 3x3 errosion + lpf
23:20	R/W	9	reg_sr3_dering_maxrange: range of dering in LR resolution.
19:18	R/W		reserved
17:16	R/W	2	reg_sr3_dering_lcvar_blend_mode: mode for lcvar calculation. 0:HV blend; 1:diag blend; 2:HV blend + V; 3: HV blend+Diag blend
15:8	R/W	40	reg_sr3_dering_lcvar_gain: gain to local variant and normalized to 32 as 1
7:0	R/W	28	reg_sr3_dering_lcvar_nearby_maxsad_th: threshold to use near side maxsad if that side is larger than this threshold, otherwise use the max one.

SRSHARP0_SR3_DERING_LUMA2PKGAIN_0TO3 0x3e6c

Bit(s)	R/W	Default	Description
31:24	R/W	255	reg_sr3_dering_luma2pkgain3: level limit(for th0<bpcon<th1) of curve for dering pkgain base on LPF luma level
23:16	R/W	255	reg_sr3_dering_luma2pkgain2: level limit(for bpcon<th0) of curve for dering pkgain base on LPF luma level
15:8	R/W	200	reg_sr3_dering_luma2pkgain1: threshold 1 of curve for dering pkgain based on LPF luma level.
7:0	R/W	30	reg_sr3_dering_luma2pkgain0: threshold 0 of curve for dering pkgain based on LPF luma level.

SRSHARP0_SR3_DERING_LUMA2PKGAIN_4TO6 0x3e6d

Bit(s)	R/W	Default	Description
31:24	R/W		reserved
23:16	R/W	24	reg_sr3_dering_luma2pkgain6: rate1 (for bpcon>th1) of curve for dering pkOS based on LPF luma level.
15:8	R/W	50	reg_sr3_dering_luma2pkgain5: rate0 (for bpcon<th0) of curve for dering pkOS based on LPF luma level.
7:0	R/W	255	reg_sr3_dering_luma2pkgain4: level limit(for bpcon>th1) of curve for dering pkgain base on LPF luma level

SRSHARP0_SR3_DERING_LUMA2PKOS_0TO3 0x3e6e

Bit(s)	R/W	Default	Description
31:24	R/W	255	reg_sr3_dering_luma2pkos3: level limit(for th0<bpcon<th1) of curve for dering pkOS base on LPF luma level
23:16	R/W	255	reg_sr3_dering_luma2pkos2: level limit(for bpcon<th0) of curve for dering pkOS base on LPF luma level
15:8	R/W	200	reg_sr3_dering_luma2pkos1: threshold 1 of curve for dering pkOS based on LPF luma level.
7:0	R/W	30	reg_sr3_dering_luma2pkos0: threshold 0 of curve for dering pkOS based on LPF luma level.

SRSHARP0_SR3_DERING_LUMA2PKOS_4TO6 0x3e6f

Bit(s)	R/W	Default	Description
31:24	R/W		reserved
23:16	R/W	24	reg_sr3_dering_luma2pkos6: rate1 (for bpcon>th1) of curve for dering pkOS based on LPF luma level.
15:8	R/W	50	reg_sr3_dering_luma2pkos5: rate0 (for bpcon<th0) of curve for dering pkOS based on LPF luma level.

7:0	R/W	255	reg_sr3_dering_luma2pkos4: level limit(for bpcon>th1) of curve for dering pkOS base on LPF luma level

SRSHARP0_SR3_DERING_GAINVS_MADSAD 0x3e70

Bit(s)	R/W	Default	Description
31:28	R/W	0	reg_sr3_dering_gainvs_maxsad7
27:24	R/W	0	reg_sr3_dering_gainvs_maxsad6
23:20	R/W	0	reg_sr3_dering_gainvs_maxsad5
19:16	R/W	0	reg_sr3_dering_gainvs_maxsad4
15:12	R/W	0	reg_sr3_dering_gainvs_maxsad3
11:8	R/W	0	reg_sr3_dering_gainvs_maxsad2
7:4	R/W	4	reg_sr3_dering_gainvs_maxsad1
3:0	R/W	8	reg_sr3_dering_gainvs_maxsad0: pkgain vs maxsad value, 8 node interpolations.

SRSHARP0_SR3_DERING_GAINVS_VR2MAX 0x3e71

Bit(s)	R/W	Default	Description
31:28	R/W	15	reg_sr3_dering_gainvs_vr2max7
27:24	R/W	15	reg_sr3_dering_gainvs_vr2max6
23:20	R/W	15	reg_sr3_dering_gainvs_vr2max5
19:16	R/W	15	reg_sr3_dering_gainvs_vr2max4
15:12	R/W	14	reg_sr3_dering_gainvs_vr2max3
11:8	R/W	12	reg_sr3_dering_gainvs_vr2max2
7:4	R/W	2	reg_sr3_dering_gainvs_vr2max1
3:0	R/W	0	reg_sr3_dering_gainvs_vr2max0: pkgain vs ratio

SRSHARP0_SR3_DERING_PARAM0 0x3e72

Bit(s)	R/W	Default	Description
31:24	R/W		reserved
23:16	R/W	10	reg_sr3_dering_lcvar_floor
15:8	R/W	32	reg_sr3_dering_vr2max_gain: gain to max before feeding to LUT
7:6	R/W		reserved
5:0	R/W	16	reg_sr3_dering_vr2max_limt: limit of maxsad

SRSHARP0_SR3_DRTLPF_THETA 0x3e73

Bit(s)	R/W	Default	Description
31:0	R/W	0xfec96420	reg_sr3_drtlpf_theta: u4x8 directional lpf beta coef for min_sad/min2_sad compared to x=0:7 correspond to [1:8]/16; 0 means no drtLPF, 15: 100% alpha dependant drtLPF.

SRSHARP0_SATPRT_CTRL 0x3e74

Bit(s)	R/W	Default	Description
31:28	R/W		reserved
27:16	R/W	5	reg_satprt_sat_core: 4x will be coring to cor(irgb_max-irgb_min) to calculate the oy_delt, the smaller the more protection to color, the larger only the rich color will be protected.
15:8	R/W	64	reg_satprt_sat_rate: rate to cor(irgb_max-irgb_min) to calculate the oy_delt,the larger the more protection to color; norm 16 as 1
7:4	R/W		reserved
3:2	R/W	1	reg_satprt_csc_mode: CSC mode of current yuv input: 0:601; 1:709; 2:BT2020 NCL; 3 reserved
1	R/W	1	reg_satprt_is_lmt: flag telling the YUV is limited range data or full rang data; 1 is limited data
0	R/W	0	reg_satprt_enable: 1 to enable of saturation protection for dnlp adjustments

SRSHARP0_SATPRT_DIVM 0x3e75

Bit(s)	R/W	Default	Description
31:24	R/W		reserved
23:0	R/W	{128,128,128}	reg_satprt_div_m: u8x3, 1/m, normalized to 128 as 1.

SRSHARP0_DB_FLT_CTRL 0x3e77

Bit(s)	R/W	Default	Description
26	R/W	0	reg_nrdeband_reset1 : // unsigned , default = 0 0 : no reset seed 1: reload chroma seed
25	R/W	0	reg_nrdeband_reset0 : // unsigned , default = 0 0 : no reset seed 1: reload luma seed
24	R/W	0	reg_nrdeband_rgb : // unsigned , default = 0 0 : yuv 1: RGB
23	R/W	1	reg_nrdeband_en11 : // unsigned , default = 1 debanding registers of side lines, [0] for luma, same for below
22	R/W	1	reg_nrdeband_en10 : // unsigned , default = 1 debanding registers of side lines, [1] for chroma, same for below
21	R/W	1	reg_nrdeband_siderand : // unsigned , default = 1 options to use side two lines use the rand, instead of use for the YUV three component of middle line, 0: seed[3]/bandrand[3] for middle line yuv; 1: seed[3]/bandrand[3] for nearby three lines Y;

Bit(s)	R/W	Default	Description
20	R/W	0	reg_nrdeband_randmode : // unsigned , default = 0 mode of rand noise adding, 0: same noise strength for all difs; else: strenght of noise will not exceed the difs, MIN((pPKReg->reg_nrdeband_bandrand[m]), noise[m])
19:17	R/W	6	reg_nrdeband_bandrand2 : // unsigned , default = 6
15:13	R/W	6	reg_nrdeband_bandrand1 : // unsigned , default = 6
11: 9	R/W	6	reg_nrdeband_bandrand0 : // unsigned , default = 6
7	R/W	1	reg_nrdeband_hpxor1 : // unsigned , default = 1 debanding random hp portion xor, [0] for luma
6	R/W	1	reg_nrdeband_hpxor0 : // unsigned , default = 1 debanding random hp portion xor, [1] for chroma
5	R/W	1	reg_nrdeband_en1 : // unsigned , default = 1 debanding registers, for luma
4	R/W	1	reg_nrdeband_en0 : // unsigned , default = 1 debanding registers, for chroma
3: 2	R/W	2	reg_nrdeband_lpf_mode1 : // unsigned , default = 2 lpf mode, 0: 3x3, 1:3x5; 2: 5x5; 3:5x7
1: 0	R/W	2	reg_nrdeband_lpf_mode0 : // unsigned , default = 2 lpf mode, 0: 3x3, 1:3x5; 2: 5x5; 3:5x7

SRSHARP0_DB_FLT_YC_THRD 0x3e78

Bit(s)	R/W	Default	Description
31:28	R/W	9	reg_nrdeband_luma_th3 : // unsigned , default = 9 threshold to |Y-Ylpf|, if < th[0] use lpf
27:24	R/W	7	reg_nrdeband_luma_th2 : // unsigned , default = 7 elseif <th[1] use (lpf*3 + y)/4
23:20	R/W	6	reg_nrdeband_luma_th1 : // unsigned , default = 6 elseif <th[1] use (lpf*3 + y)/4elseif <th[2] (lpf*1 + y)/2
19:16	R/W	5	reg_nrdeband_luma_th0 : // unsigned , default = 5 elseif <th[1] use (lpf*3 + y)/4elseif elseif <th[3] (lpf*1 + 3*y)/4; else
15:12	R/W	9	reg_nrdeband_chrm_th3 : // unsigned , default = 9 threshold to |Y-Ylpf|, if < th[0] use lpf
11: 8	R/W	7	reg_nrdeband_chrm_th2 : // unsigned , default = 7 elseif <th[1] use (lpf*3 + y)/4
7: 4	R/W	6	reg_nrdeband_chrm_th1 : // unsigned , default = 6 elseif <th[1] use (lpf*3 + y)/4elseif <th[2] (lpf*1 + y)/2
3: 0	R/W	5	reg_nrdeband_chrm_th0 : // unsigned , default = 5 elseif <th[1] use (lpf*3 + y)/4elseif elseif

SRSHARP0_DB_FLT_RANDLUT 0x3e79

Bit(s)	R/W	Default	Description
23:21	R/W	1	reg_nrdeband_randslut7 : // unsigned , default = 1 lut0

20:18	R/W	1	reg_nrdeband_randslut6 : // unsigned , default = 1 lut0
17:15	R/W	1	reg_nrdeband_randslut5 : // unsigned , default = 1 lut0
14:12	R/W	1	reg_nrdeband_randslut4 : // unsigned , default = 1 lut0
11: 9	R/W	1	reg_nrdeband_randslut3 : // unsigned , default = 1 lut0
8: 6	R/W	1	reg_nrdeband_randslut2 : // unsigned , default = 1 lut0
5: 3	R/W	1	reg_nrdeband_randslut1 : // unsigned , default = 1 lut0
2: 0	R/W	1	reg_nrdeband_randslut0 : // unsigned , default = 1 lut0

SRSHARP0_DB_FLT_PXI_THRD 0x3e7a

Bit(s)	R/W	Default	Description
25:16	R/W	0	reg_nrdeband_yc_th1 : // unsigned , default = 0 to luma/|u/v| for using the denoise
9: 0	R/W	0	reg_nrdeband_yc_th0 : // unsigned , default = 0 to luma/|u/v| for using the denoise

SRSHARP0_DB_FLT_SEED_Y 0x3e7b

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438240	reg_nrdeband_seed0 : // unsigned , default = 1621438240 noise adding seed for Y. seed[0]= 0x60a52f20; as default

SRSHARP0_DB_FLT_SEED_U 0x3e7c

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438247	reg_nrdeband_seed1 : // unsigned , default = 1621438247 noise adding seed for U. seed[0]= 0x60a52f27; as default

SRSHARP0_DB_FLT_SEED_V 0x3e7d

Bit(s)	R/W	Default	Description
31: 0	R/W	1621438242	reg_nrdeband_seed2 : // unsigned , default = 1621438242 noise adding seed for V. seed[0]= 0x60a52f22; as default

SRSHARP0_PKGAIN_VSLUMA_LUT_L 0x3e7e

Bit(s)	R/W	Default	Description
31:28	R/W	5	reg_pkgain_vsluma_lut7
27:24	R/W	6	reg_pkgain_vsluma_lut6
23:20	R/W	6	reg_pkgain_vsluma_lut5
19:16	R/W	6	reg_pkgain_vsluma_lut4
15:12	R/W	7	reg_pkgain_vsluma_lut3
11:8	R/W	10	reg_pkgain_vsluma_lut2

7:4	R/W	12	reg_pkgain_vsluma_lut1
3:0	R/W	8	reg_pkgain_vsluma_lut0

SRSHARP0_PKGAIN_VSLUMA_LUT_H 0x3e7f

Bit(s)	R/W	Default	Description
31:4	R/W		reserved
3:0	R/W	4	reg_pkgain_vsluma_lut8

SRSHARP0_PKOSHT_VSLUMA_LUT_L 0x3e80

Bit(s)	R/W	Default	Description
31:28	R/W	5	reg_pkosht_vsluma_lut7
27:24	R/W	6	reg_pkosht_vsluma_lut6
23:20	R/W	6	reg_pkosht_vsluma_lut5
19:16	R/W	6	reg_pkosht_vsluma_lut4
15:12	R/W	7	reg_pkosht_vsluma_lut3
11:8	R/W	10	reg_pkosht_vsluma_lut2
7:4	R/W	12	reg_pkosht_vsluma_lut1
3:0	R/W	8	reg_pkosht_vsluma_lut0

SRSHARP0_PKOSHT_VSLUMA_LUT_H 0x3e81

Bit(s)	R/W	Default	Description
31:4	R/W		reserved
3:0	R/W	4	reg_pkosht_vsluma_lut8

SRSHARP0_SATPRT_LMT_RGB1 0x3e82

Bit(s)	R/W	Default	Description
27:16	R/W	0d0	reg_satprt_lmt_g:
11: 0	R/W	0d0	reg_satprt_lmt_r: limit of RGB channel, for limited range RGB, 12bits

SRSHARP0_SATPRT_LMT_RGB2 0x3e83

Bit(s)	R/W	Default	Description
31:16	R/W	0d0	reserved
11: 0	R/W	0d0	reg_satprt_lmt_b: limit of RGB channel, for limited range RGB

SRSHARP0_SHARP_GATE_CLK_CTRL_0 0x3e84

Bit(s)	R/W	Default	Description
31:0	R/W	0d0	Gate clock control [01:00]: sharp input control unit [03:02]: deband unit [05:04]: dejaggy unit [07:06]: dnlp unit [09:08]: demo control unit [11:10]: horiz interp unit

SRSHARP0_SHARP_GATE_CLK_CTRL_1 0x3e85

Bit(s)	R/W	Default	Description
31:0	R/W	0d0	Gate clock control [01:00]: sr_top “pipe_ctrl” [03:02]: drt [05:04]: ssd [07:06]: cubic [09:08]: edi [11:10]: pkgainsad [13:12]: bicomux [15:14]: bicin_fifo [17:16]: lpf4pkgain_fifo [19:18]: min2hvgain_fifo [21:20]: sad4pkgain_fifo [23:22]: dirminmax4xtl_fifo [25:24]: drtsad8_fifo [27:26]: ssdmax_fifo [29:28]: pkminmax_fifo [31:30]: cirdrtgain_fifo

SRSHARP0_SHARP_GATE_CLK_CTRL_2 0x3e86

Bit(s)	R/W	Default	Description
31:0	R/W	0d0	Gate clock control [01:00]: bufdiff_fifo [03:02]: osvar_fifo [05:04]: pkhvgain unit [07:06]: pkgain unit [09:08]: TI unit [11:10]: pk unit [13:12]: locvar unit [15:14]: hvconc unit

SRSHARP0_SHARP_GATE_CLK_CTRL_3 0x3e87

Bit(s)	R/W	Default	Description
31:0	R/W	0d0	Gate clock control [01:00]: TI, htl Y [03:02]: TI, vti Y [05:04]: TI, htl U [07:06]: TI, vtl U [09:08]: TI, htl V [11:10]: TI, vtl V [13:12]: TI, lumaminmax_fifo [15:14]: TI, chrmminmax fifo

SRSHARP0_SHARP_DPS_CTRL 0x3e88

Bit(s)	R/W	Default	Description
31:0	R/W	0d0	Power saving control : hvcon : nrssd : os filter : cubic 5 lines to 9 lines : dering : locvar : drtlpf : hlti : hcti : vlti : vcti : lti blend : cti blend : htishort (no used) : nr Y filter : nr C filter : nr belnd : pkti blend : os ctrl : pk HP : pk BP [26:24] dejaggy power saving control [30:28] reserved

SRSHARP0_DNLP_00 0x3e90

Bit(s)	R/W	Default	Description

31: 0	R/W	0x00100008	reg_dnlp_ygrid0 : : dnlp00 . unsigned , default = 32'h00100008

SRSHARP0_DNLP_01 0x3e91

Bit(s)	R/W	Default	Description
31: 0	R/W	0x00200018	reg_dnlp_ygrid1 : : dnlp01 . unsigned , default = 32'h00200018

SRSHARP0_DNLP_02 0x3e92

Bit(s)	R/W	Default	Description
31: 0	R/W	0x00300028	reg_dnlp_ygrid2 : : dnlp02 . unsigned , default = 32'h00300028

SRSHARP0_DNLP_03 0x3e93

Bit(s)	R/W	Default	Description
31: 0	R/W	0x00400038	reg_dnlp_ygrid3 : : dnlp03 . unsigned , default = 32'h00400038

SRSHARP0_DNLP_04 0x3e94

Bit(s)	R/W	Default	Description
31: 0	R/W	0x00500048	reg_dnlp_ygrid4 : : dnlp04 . unsigned , default = 32'h00500048

SRSHARP0_DNLP_05 0x3e95

Bit(s)	R/W	Default	Description
31: 0	R/W	0x0068005c	reg_dnlp_ygrid5 : : dnlp05 . unsigned , default = 32'h0068005c

SRSHARP0_DNLP_06 0x3e96

Bit(s)	R/W	Default	Description
31: 0	R/W	0x00800074	reg_dnlp_ygrid6 : : dnlp06 . unsigned , default = 32'h00800074

SRSHARP0_DNLP_07 0x3e97

Bit(s)	R/W	Default	Description
31: 0	R/W	0x00a00090	reg_dnlp_ygrid7 : : dnlp07 . unsigned , default = 32'h00a00090

SRSHARP0_DNLP_08 0x3e98

Bit(s)	R/W	Default	Description
31: 0	R/W	0x00c000b0	reg_dnlp_ygrid8 : : dnlp08 . unsigned , default = 32'h00c000b0

SRSHARP0_DNLP_09 0x3e99

Bit(s)	R/W	Default	Description
31: 0	R/W	0x00e000d0	reg_dnlp_ygrid9 : : dnlp09 . unsigned , default = 32'h00e000d0

SRSHARP0_DNLP_10 0x3e9a

Bit(s)	R/W	Default	Description
31: 0	R/W	0x010000f0	reg_dnlp_ygrid10 : : dnlp10 . unsigned , default = 32'h010000f0

SRSHARP0_DNLP_11 0x3e9b

Bit(s)	R/W	Default	Description
31: 0	R/W	0x012c0114	reg_dnlp_ygrid11 : : dnlp11 . unsigned , default = 32'h012c0114

SRSHARP0_DNLP_12 0x3e9c

Bit(s)	R/W	Default	Description
31: 0	R/W	0x01540140	reg_dnlp_ygrid12 : : dnlp12 . unsigned , default = 32'h01540140

SRSHARP0_DNLP_13 0x3e9d

Bit(s)	R/W	Default	Description
31: 0	R/W	0x0180016c	reg_dnlp_ygrid13 : : dnlp13 . unsigned , default = 32'h0180016c

SRSHARP0_DNLP_14 0x3e9e

Bit(s)	R/W	Default	Description
31: 0	R/W	0x01c001a0	reg_dnlp_ygrid14 : : dnlp14 . unsigned , default = 32'h01c001a0

SRSHARP0_DNLP_15 0x3e9f

Bit(s)	R/W	Default	Description
31: 0	R/W	0x020001e0	reg_dnlp_ygrid15 : : dnlp15 . unsigned , default = 32'h020001e0

SRSHARP0_DNLP_16 0x3ea0

Bit(s)	R/W	Default	Description
31: 0	R/W	0x02400220	reg_dnlp_ygrid16 : : dnlp16 . unsigned , default = 32'h02400220

SRSHARP0_DNLP_17 0x3ea1

Bit(s)	R/W	Default	Description
31: 0	R/W	0x02800260	reg_dnlp_ygrid17 : : dnlp17 . unsigned , default = 32'h02800260

SRSHARP0_DNLP_18 0x3ea2

Bit(s)	R/W	Default	Description
31: 0	R/W	0x02b00298	reg_dnlp_ygrid18 : : dnlp18 . unsigned , default = 32'h02b00298

SRSHARP0_DNLP_19 0x3ea3

Bit(s)	R/W	Default	Description

31: 0	R/W	0x02e002c8	reg_dnlp_ygrid19 : : dnlp19 . unsigned , default = 32'h02e002c8

SRSHARP0_DNLP_20 0x3ea4

Bit(s)	R/W	Default	Description
31: 0	R/W	0x030002f0	reg_dnlp_ygrid20 : : dnlp20 . unsigned , default = 32'h030002f0

SRSHARP0_DNLP_21 0x3ea5

Bit(s)	R/W	Default	Description
31: 0	R/W	0x03200310	reg_dnlp_ygrid21 : : dnlp21 . unsigned , default = 32'h03200310

SRSHARP0_DNLP_22 0x3ea6

Bit(s)	R/W	Default	Description
31: 0	R/W	0x0338032c	reg_dnlp_ygrid22 : : dnlp22 . unsigned , default = 32'h0338032c

SRSHARP0_DNLP_23 0x3ea7

Bit(s)	R/W	Default	Description
31: 0	R/W	0x03500348	reg_dnlp_ygrid23 : : dnlp23 . unsigned , default = 32'h03500348

SRSHARP0_DNLP_24 0x3ea8

Bit(s)	R/W	Default	Description
31: 0	R/W	0x035c0358	reg_dnlp_ygrid24 : : dnlp24 . unsigned , default = 32'h035c0358

SRSHARP0_DNLP_25 0x3ea9

Bit(s)	R/W	Default	Description
31: 0	R/W	0x03680360	reg_dnlp_ygrid25 : : dnlp25 . unsigned , default = 32'h03680360

SRSHARP0_DNLP_26 0x3eaa

Bit(s)	R/W	Default	Description
31: 0	R/W	0x03780370	reg_dnlp_ygrid26 : : dnlp26 . unsigned , default = 32'h03780370

SRSHARP0_DNLP_27 0x3eab

Bit(s)	R/W	Default	Description
31: 0	R/W	0x03880380	reg_dnlp_ygrid27 : : dnlp27 . unsigned , default = 32'h03880380

SRSHARP0_DNLP_28 0x3eac

Bit(s)	R/W	Default	Description
31: 0	R/W	0x03a00390	reg_dnlp_ygrid28 : : dnlp28 . unsigned , default = 32'h03a00390

SRSHARP0_DNLP_29 0x3ead

Bit(s)	R/W	Default	Description
31: 0	R/W	0x03c003b0	reg_dnlp_ygrid29 : : dnlp29 . unsigned , default = 32'h03c003b0

SRSHARP0_DNLP_30 0x3eae

Bit(s)	R/W	Default	Description
31: 0	R/W	0x03e003d0	reg_dnlp_ygrid30 : : dnlp30 . unsigned , default = 32'h03e003d0

SRSHARP0_DNLP_31 0x3eaf

Bit(s)	R/W	Default	Description
31: 0	R/W	0x03fc03f0	reg_dnlp_ygrid31 : : dnlp31 . unsigned , default = 32'h03fc03f0

10.2.3.30 VKSTONE registers

VKS_CTRL 0x3100

Bit(s)	R/W	Default	Description
31	W	1	reg_vks_en : // unsigned , default = 1 enable signal of the vks function need set high in every frme
30	R/W	1	reg_vks_scl_mode0 : // unsigned , default = 1 : b0 mode of vks ofset mode, 0: offset= offset; 1: offset= offset*step= ofset/scale;
29	R/W	1	reg_vks_scl_mode1 : // unsigned , default = 1 : b0 mode of vks ofset mode, 0: offset= offset; 1: offset= offset*step= ofset/scale;
28	R/W	1	reg_vks_fill_mode : // unsigned , default = 1 mode of out-of-boundary fill, 0 extension, 1: fill with the fill_value
27:26	R/W	1	reg_vks_row_inp_mode : // unsigned , default = 1 , interpolation mode from 16pieces ofset/step to each line ofset and step; 0: linear interpolation; 1: cubic interpolation (using ccoef)
25	R/W	0	reg_vks_border_ext_mode0 : // unsigned , default = 0 , extend mode of the border data of luma and chroma, 0: copy the most border one; 1: extropolate the border one
24	R/W	0	reg_vks_border_ext_mode1 : // unsigned , default = 0 , extend mode of the border data of luma and chroma, 0: copy the most border one; 1: extropolate the border one
23	R/W	1	reg_vks_obuf_mode0 : // unsigned , default = 1 , mode of output buffer left/right side. 0: no precalculate active pixels during output fill region; 1: precaclc active pixels during output fill regions
22	R/W	1	reg_vks_obuf_mode1 : // unsigned , default = 1 , mode of output buffer left/right side. 0: no precalculate active pixels during output fill region; 1: precaclc active pixels during output fill regions
21:20	R/W	3	reg_vks_obuf_mrgn0 : // unsigned , default = 3 , margin pixels for left right most active pixel to the fill pixels to avoid jump
19:18	R/W	3	reg_vks_obuf_mrgn1 : // unsigned , default = 3 , margin pixels for left right most active pixel to the fill pixels to avoid jump

Bit(s)	R/W	Default	Description
17:16	R/W	2	reg_vks_phs_qmode : // unsigned , default = 2 , interpolation mode of the phase, 0: floor to 1/64 phase; 1: round to 1/64 phase; 2/3 linear intp
15: 0	R/W	11651	reg_vks_row_scl : // unsigned , default = 11651 , scale of row to make it fit to the 16 pieces, scl = (2^23)/RowMax

VKS_OUT_WIN_SIZE 0x3101

Bit(s)	R/W	Default	Description
29:16	R/W	1280	reg_vks_ocolmax : // unsigned , default = 1280 output outer window col number, decided by the projector
13: 0	R/W	720	reg_vks_orowmax : // unsigned , default = 720 output outer window row number, decided by the projector

VKS_PRELPF_YCOEF0 0x3102

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_vks_prelpf_ycoef0 : // signed , default = -128 coef of horizontal luma prelpf for Keystone, normalized 128 as '1'
23:16	R/W	0	reg_vks_prelpf_ycoef1 : // signed , default = 0 coef of horizontal luma prelpf for Keystone, normalized 128 as '1'
15: 8	R/W	0	reg_vks_prelpf_ycoef2 : // signed , default = 0 coef of horizontal luma prelpf for Keystone, normalized 128 as '1'
7: 0	R/W	0	reg_vks_prelpf_ycoef3 : // signed , default = 0 coef of horizontal luma prelpf for Keystone, normalized 128 as '1'

VKS_PRELPF_YCOEF1 0x3103

Bit(s)	R/W	Default	Description
15: 8	R/W	0	reg_vks_prelpf_ycoef4 : // signed , default = 0 coef of horizontal luma prelpf for Keystone, normalized 128 as '1'
7: 0	R/W	0	reg_vks_prelpf_ycoef5 : // signed , default = 0 coef of horizontal luma prelpf for Keystone, normalized 128 as '1'

VKS_PRELPF_CCOEF0 0x3104

Bit(s)	R/W	Default	Description
31:24	R/W	0x0	reg_vks_prelpf_ccoef0 : // signed , default = -128 mode of horizontal chroma prelpf for Keystone, normalized 128 as '1'
23:16	R/W	0	reg_vks_prelpf_ccoef1 : // signed , default = 0 mode of horizontal chroma prelpf for Keystone, normalized 128 as '1'
15: 8	R/W	0	reg_vks_prelpf_ccoef2 : // signed , default = 0 mode of horizontal chroma prelpf for Keystone, normalized 128 as '1'
7: 0	R/W	0	reg_vks_prelpf_ccoef3 : // signed , default = 0 mode of horizontal chroma prelpf for Keystone, normalized 128 as '1'

VKS_PRELPF_CCOEF1 0x3105

Bit(s)	R/W	Default	Description
15: 8	R/W	0	reg_vks_prelpf_ccoef4 : // signed , default = 0 mode of horizontal chroma prelpf for Keystone, normalized 128 as '1'
7: 0	R/W	0	reg_vks_prelpf_ccoef5 : // signed , default = 0 mode of horizontal chroma prelpf for Keystone, normalized 128 as '1'

VKS_FILL_VAL 0x3106

Bit(s)	R/W	Default	Description
23:16	R/W	0	reg_vks_fill_value0 : // unsigned , default = 0 , border fill color define. yuv: [0 128 128]; rgb:[0 0 0] ,use 8 bits in 12bit path,6bits in 10bit path
15: 8	R/W	128	reg_vks_fill_value1 : // unsigned , default = 128 , border fill color define. yuv: [0 128 128]; rgb:[0 0 0] ,use 8 bits in 12bit path,6bits in 10bit path
7: 0	R/W	128	reg_vks_fill_value2 : // unsigned , default = 128 , border fill color define. yuv: [0 128 128]; rgb:[0 0 0] ,use 8 bits in 12bit path,6bits in 10bit path

VKS_IWIN_HSIZE 0x3107

Bit(s)	R/W	Default	Description
29:16	R/W	160	reg_vks_iwinx0 : // unsigned , default = 160 , input start-col and end-col;
13: 0	R/W	1279	reg_vks_iwinx1 : // unsigned , default = 1279 , input start-col and end-col;

VKS_IWIN_VSIZE 0x3108

Bit(s)	R/W	Default	Description
29:16	R/W	0	reg_vks_iwiny0 : // unsigned , default = 0 , input start-row and end-row;
13: 0	R/W	719	reg_vks_iwiny1 : // unsigned , default = 719 , input start-row and end-row;

VKS_TOP_MISC 0x3109

Bit(s)	R/W	Default	Description
18	R/W	1	reg_flt_en : // unsigned , default = 1
17	R/W	0	reg_frm_rst : // unsigned , default = 0
16	R/W	0	reg_ctrl_sync : // unsigned , default = 0
15: 8	R/W	4	blank_num : // unsigned , default = 4
7: 0	R/W	9	flt_blank_num : // unsigned , default = 9

VKS_START_CTRL 0x310a

Bit(s)	R/W	Default	Description

16	R/W	0	reg_vks_en_mode : // unsigned , default = 0
15: 0	R/W	5	reg_hold_phnum : // unsigned , default = 5

VKS_LBUF_SIZE 0x310b

Bit(s)	R/W	Default	Description
11: 0	R/W	1024	reg_lbuf_depth : // unsigned , default = 1024

VKS_PARA_ADDR_PORT 0x310e

Bit(s)	R/W	Default	Description
31: 0	R/W	3ff	Access address for vkstone para_lut,start address must be 0x09

VKS_PARA_DATA_PORT 0x310f

Bit(s)	R/W	Default	Description
31: 0	R/W	/	Data access for vkstone para lut

Below are the detail registers of VKS_PARA_DATA_PORT and VKS_PARA_ADDR_PORT control:

VKS_SCL_OFFSET00 ~ VKS_SCL_OFFSET16 0X09~0X19

Bit(s)	R/W	Default	Description
31:20			reserved
19: 0	R/W	/	Left offset of the input pixel offset from the left 12bits pixel and 8 bits float phase

VKS_SCL_STEP00 ~ VKS_SCL_STEP16 0X1A~0X2A

Bit(s)	R/W	Default	Description
24:20			reserved
23: 0	R/W	/	Unsigned reg_scl_stepx for ratio of each line(defined piece),step:4.20 opxium = (ipxium << 20)/step scale:4.2 = 1/step

VKS_PPS_YCOEF00 ~ VKS_PPS_YCOEF32 0X2B~0X4B

Bit(s)	R/W	Default	Description
31:24	R/W	/	Signed reg_vks_ycoef0 Poly_phase scalar coef
23:16	R/W	/	Signed reg_vks_ycoef1 Poly_phase scalar coef
15:8	R/W	/	Signed reg_vks_ycoef2 Poly_phase scalar coef
7: 0	R/W	/	Signed reg_vks_ycoef3 Poly_phase scalar coef

VKS_PPS_CCOEF00 ~ VKS_PPS_CCOEF32 0X4C~0X6C

Bit(s)	R/W	Default	Description
31:24	R/W	/	Signed reg_vks_ccoef0 Poly_phase scalar coef

23:16	R/W	/	Signed reg_vks_ccoef1 Poly_phase scalar coef
15:8	R/W	/	Signed reg_vks_ccoef2 Poly_phase scalar coef
7: 0	R/W	/	Signed reg_vks_ccoef3 Poly_phase scalar coef

10.2.3.31 OSD1 registers

VIU_OSD1_CTRL_STAT 0x1A10

Bit(s)	R/W	Default	Description
31	R/W	0	osd_cfg_syn_en : // unsigned , default =0 1: module enable sync by go_field 0: normal
30	R/W	0	ENABLE_FREE_CLK. 1 = Use free-running clock; 0 = Use gated clock to save power.
29	R	0	TEST_RD_DSR: Applicable only when OSD debug mode is enabled. 1 = A new pixel is ready at register VIU_OSD1_TEST_RDDATA; 0 = No data ready.
/	/	/	/
27-24	R	0	OSD_BLK_MODE: the input pixel format of which the current OSD block is being processed. Dvalin src & normal src have different pixel format value for same format
23-22	R	0	OSD_BLK_PTR: The number of the current OSD block that is being processed.
21	R	0	OSD_ENABLE. 1 = OSD display is enabled; 0 = disabled.
20-12	R/W	0	GLOBAL_ALPHA: legal range 0 – 256. It is a 9-bit value that is multiplied to all output pixel's Alpha value, and then normalized, i.e.: Alpha_tmp = Alpha_internal + (Alpha_internal == 0 ? 0 : 1); Alpha_out = (Alpha_tmp * GLOBAL_ALPHA) / 256;
11	R/W	0	TEST_RD_EN: OSD debug mode enable. 1 = Output pixels are not routed to VPP, instead they are presented on registers VIU_OSD1_TEST_RDDATA, for CPU to read. 0 = Normal mode, pixels are output to VPP.
10-9	R/W	0	unused

Bit(s)	R/W	Default	Description
8-5	R/W	0	CTRL_MTCH_Y: For OSD 444, 422 or 16-bit (COLOR_MATRIX = 0 or 1) mode, the input pixels contain no Alpha information, in order to associated the output pixel with an Alpha value, the following steps are taken: If TC_ALPHA_EN = 0, then all output pixels use a default Alpha value 0xFF; If TC_ALPHA_EN = 1, then the Alpha value is looked up by matching the pixel's Y/Cb/Cr against four Alpha registers' Y/Cb/Cr. If the pixel matches any one of the Alpha registers, then this register's Alpha value is used; If the pixel matches with more than one of the Alpha registers, then the lower Alpha register takes priority, e.g. use Alpha Reg0's value if the pixel match both Alpha Reg0 and Reg1; If no match, then use default Alpha value 0xFF. There are two ways of matching: one way is that the pixel has to compare all Y, Cb and Cr value with the Alpha registers; the other way is that the pixel only has to compare Y value with the Alpha registers. CTRL_MTCH_Y defines which way is used to determine a match. Bit[0] is for matching Alpha register 0: 1 = only need to compare Y; 0 = compare all Y, Cb and Cr. Bit[1] is for matching Alpha register 1: 1 = only need to compare Y; 0 = compare all Y, Cb and Cr. Bit[2] is for matching Alpha register 2: 1 = only need to compare Y; 0 = compare all Y, Cb and Cr. Bit[3] is for matching Alpha register 3: 1 = only need to compare Y; 0 = compare all Y, Cb and Cr.
4	R/W	0	CTRL_422TO444. 1 = Enable conversion of 422 format input to 444 format output; 0 = Disable 422 to 444 conversion.
/	/	/	/
2	R/W	0	osd_mem_mode : // unsigned , default =0 0:canvas_araddr 1: linear_araddr (Dvalin src must use this mode)
1	R/W	0	premult_en : // unsigned , default =0
0	R/W	0	OSD_BLK_ENABLE: Each bit to enable display an OSD block

VIU_OSD1_CTRL_STAT2 0x1A2d

Bit(s)	R/W	Default	Description
31-16	R	0	unused
15	R/W	0	unused
14	R/W	0	Replaced_alpha_en
13-6	R/W	0	Replaced_alpha
5-4	R/W	0	Hold_fifo_lines[6:5]
3	R/W	0	RGBYUV_FULL_RANGE: Select coefficients for applicable output range. 1 = output full range 0-255; 0 = output range 16-235.

Bit(s)	R/W	Default	Description
2	R/W	0	ALPHA_9B_MODE: Define how to expand 8-bit alpha value to 9-bit. 1 = The formula is (Alpha < 128) ? Alpha : Alpha + 1; 0 = The formula is (Alpha == 0) ? Alpha : Alpha + 1.
1	R/W	0	Pedding status cleanup
0	R/W	0	COLOR_EXPAND_MODE. 1 = Expand the color components to 8-bit by padding LSBs with MSBs. E.g. If the input is 5'b11000, the output is expanded to 8'b11000110; 0 = Expand the color components to 8-bit by padding LSBs with 0.

VIU_OSD1_TCOLOR_AG0 0x1A17
VIU_OSD1_TCOLOR_AG1 0x1A18
VIU_OSD1_TCOLOR_AG2 0x1A19
VIU_OSD1_TCOLOR_AG3 0x1A1a

Define Alpha register 0/1/2/3 values.

Bit(s)	R/W	Default	Description
31-24	R/W	0xFF	Y or R.
23-16	R/W	0xFF	CB or G.
15-8	R/W	0xFF	CR or B.
7-0	R/W	0xFF	ALPHA.

VIU_OSD1_BLK0_CFG_W0 0x1A1b

Defines display block 0/1/2/3's property, word 0.

Bit(s)	R/W	Default	Description
31	R/W	0	Reserved
30	R/W	0	mali_src_en 1: read data from Dvalin afbcd decoder 0: read data from DDR directly
29	R/W	0	y_rev: 0=normal read, 1=reverse read in Y direction
28	R/W	0	x_rev: 0=normal read, 1=reverse read in X direction
27-24	R/W	0	Reserved
23-16	R/W	0	TBL_ADDR. Virtual canvas LUT entry.
15	R/W	0	LITTLE_ENDIAN: define the of data stored in DDR. 1 = Data stored in DDR memory are of little endian; 0 = Data stored in DDR memory are of big endian.
14	R/W	0	RPT_Y: For reducing data size stored in DDR.

Bit(s)	R/W	Default	Description
			1 = For each line, OSD will display twice; 0 = No repeat, OSD display once per line.
13-12	R/W	0	INTERP_CTRL: If enabled, interpolate a data after each incoming pixels, in order to save DDR bandwidth. 0 = No interpolation; 1 = unused, no interpolation; 2 = Interpolate with preceding pixel value; 3 = Interpolate with the averaged value between the preceding pixel and the next pixel.
11-8	R/W	0	OSD_BLK_MODE: Define the OSD block's input pixel format. 0-2 = unused; 3 = 4:2:2 mode. Input 32-bit data for 2 pixels. Bit[31:24] is Y0, bit [23:16] is Cb0, bit[15:8] is Y1, bit [7:0] is Cr0, for Alpha value refer to reg VIU_OSD1_CTRL_STAT.CTRL_MTCH_Y; 4 = 16-bit mode. Refer to COLOR_MATRIX; 5 = 32-bit mode. Refer to COLOR_MATRIX; 6 = unused; 7 = 24-bit mode. Refer to COLOR_MATRIX; 8-15 = unused;
6	R/W	0	TC_ALPHA_EN: refer to reg VIU_OSD1_CTRL_STAT.CTRL_MTCH_Y. 1 = Enable alpha register matching. 0 = Disable.
5-2	R/W	0	COLOR_MATRIX: Applicable only to 16-bit color mode (OSD_BLK_MODE=4), 32-bit mode (OSD_BLK_MODE=5) and 24-bit mode (OSD_BLK_MODE=7), defines the bit-field allocation of the pixel data. For expanding the bit-fields to full 8-bit, refer to VIU_OSD1_CTRL_STAT2.color_expand_mode For 16-bit mode (OSD_BLK_MODE=4): 0 = 6:5:5 format. Bit[15:10] is Y[7:2] or R[7:2], bit[9:5] is Cb[7:3] or G[7:3], bit[4:0] is Cr[7:3] or B[7:3], for Alpha value refer to reg VIU_OSD1_CTRL_STAT.CTRL_MTCH_Y; 1 = 8:4:4 format. Bit[15:8] is Y or R, bit[7:4] is Cb[7:4] or G[7:4], bit[3:0] is Cr[7:4] or B[7:4]], for Alpha value refer to reg VIU_OSD1_CTRL_STAT.CTRL_MTCH_Y; 4 = 5:6:5 format. Bit[15:11] is Y[7:3] or R[7:3], bit[10:5] is Cb[7:2] or G[7:2], bit[4:0] is Cr[7:3] or B[7:3], for Alpha value refer to reg VIU_OSD1_CTRL_STAT.CTRL_MTCH_Y; For 32-bit mode (OSD_BLK_MODE=5): 0 = RGBA 8:8:8:8 format. Bit[31:24] is Y or R, bit[23:16] is Cb or G; bit[15:8] is Cr or B; bit[7:0] is Alpha; 1 = ARGB 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Y or R; bit[15:8] is Cb or G; bit[7:0] is Cr or B; 2 = ABGR 8:8:8:8 format. Bit[31:24] is Alpha, bit[23:16] is Cr or B; bit[15:8] is Cb or G; bit[7:0] is Y or R; 3 = BGRA 8:8:8:8 format. Bit[31:24] is Cr or B, bit[23:16] is Cb or G; bit[15:8] is Y or R; bit[7:0] is Alpha. For 24-bit mode (OSD_BLK_MODE=7): 0 = RGB 8:8:8 mode. Bit[23:16] is Y or R, bit[15:8] is Cb or G, bit[7:0] is Cr or B, for Alpha value refer to reg VIU_OSD1_CTRL_STAT.CTRL_MTCH_Y;

Bit(s)	R/W	Default	Description
			5 = BGR 8:8:8 mode. Bit[23:16] is Cr or B, bit[15:8] is Cb or G, bit[7:0] is Y or R, for Alpha value refer to reg VIU_OSD1_CTRL_STAT.CTRL_MTCH_Y.
1	R/W	0	INTERLACE_EN. 1 = Enable interlace mode. 0 = Disable.
0	R/W	0	INTERLACE_SEL_ODD: Applicable only if INTERLACE_EN = 1. 1 = Only output odd lines; 0 = Only output even lines.

VIU_OSD1_BLK0_CFG_W1 0x1A1c

Defines display block 0/1/2/3's property, word 1.

Bit(s)	R/W	Default	Description
31-29	R/W	0	Unused.
28-16	R/W	0	X_END. Virtual canvas co-ordinate.
15-13	R/W	0	Unused.
12-0	R/W	0	X_START. Virtual canvas co-ordinate.

VIU_OSD1_BLK0_CFG_W2 0x1A1d

Defines display block 0/1/2/3's property, word 2.

Bit(s)	R/W	Default	Description
31-29	R/W	0	Unused.
28-16	R/W	0	Y_END. Virtual canvas co-ordinate.
15-13	R/W	0	Unused.
12-0	R/W	0	Y_START. Virtual canvas co-ordinate.

VIU_OSD1_BLK0_CFG_W3 0x1A1e

Defines display block 0/1/2/3's property, word 3.

Bit(s)	R/W	Default	Description
31-28	R/W	0	Unused.
27-16	R/W	0	H_END. Display horizontal co-ordinate.
15-12	R/W	0	Unused.
11-0	R/W	0	H_START. Display horizontal co-ordinate.

VIU_OSD1_BLK0_CFG_W4 0x1A13

Defines display block 0/1/2/3's property, word 4.

Bit(s)	R/W	Default	Description
31-28	R/W	0	Unused.
27-16	R/W	0	V_END. Display vertical co-ordinate.
15-12	R/W	0	Unused.
11-0	R/W	0	V_START. Display vertical co-ordinate.

VIU_OSD1_BLK1_CFG_W4 0x1a14

Bit(s)	R/W	Default	Description
31:0	R/W	0	Frame_addr: // unsigned , default =0 Frame_addr in linear_addr

VIU_OSD1_BLK2_CFG_W4 0x1a15

Bit(s)	R/W	Default	Description
31:0	R/W	0	Line_stride : // unsigned , default =0 Line_stride in linear_addr

VIU_OSD1_FIFO_CTRL_STAT 0x1A2b

Bit(s)	R/W	Default	Description
31	R/W	0	burst_len_sel[2] of [2:0]
30	R/W	0	BYTE_SWAP: In addition to endian control, further define whether to swap upper byte and lower byte within a 16-bit memory word. 1 = Swap, data[15:0] becomes {data[7:0], data[15:8]}; 0 = No swap, data[15:0] is still data[15:0].
29	R/W	0	Div_swap : swap the 2 64bits word in 128bits word
28-24	R/W	0	Fifo_lim : when osd fifo is small than the fifo_lim*16, closed the req port of osd_rd_mif
23-22	R/W	0	Fifo_ctrl: 00 : for 1 word in 1 burst, 01 : for 2words in 1burst, 10 : for 4 words in 1burst, 11: reserved
21-20	R	0	FIFO_ST: State of the FIFO activity. 0 = Idle; 1 = FIFO requesting; 2 = FIFO request aborting.
19	R	0	FIFO_OVERFLOW.
18-12	R/W	32	FIFO_DEPTH_VAL: Define the depth of FIFO which stores 128-bit data from DDR to be FIFO_DEPTH_VAL * 8.

Bit(s)	R/W	Default	Description
11-10	R/W	0	BURST_LEN_SEL[1:0] of [2:0]: Define DDR burst request length. 0 = up to 24 per burst; 1 = up to 32 per burst; 2 = up to 48 per burst;/ 3 = up to 64 per burst. 4 = up to 96 per burst, 5 = up to 128 per burst
9-5	R/W	4	HOLD_FIFO_LINES: The number of lines that OSD must wait after VSYNC, before it starts request data from DDR .
4	R/W	0	CLEAR_ERR: One pulse to clear error status.
3	R/W	0	FIFO_SYNC_RST: Set 1 to reset OSD FIFO.
2-1	R/W	0	ENDIAN: define the endianess of the 64-bit data stored in memory, and how to convert. 0 = No convertion; 1 = Convert to {din[31:0], din[63:32]}; 2 = Convert to {din[15:0], din[31:16], din[47:32], din[63:48]}; 3 = Convert to {din[47:32], din[63:48], din[15:0], din[31:16]};
0	R/W	0	URGENT. 1 = Set DDR request priority to be urgent; 0 = Set DDR request priority to be normal.

VIU_OSD1_TEST_RDDATA 0x1A2c

During OSD debug mode (VIU_OSD1_CTRL_STAT.TEST_RD_EN = 1), the output pixels will be presented at this register.

Bit(s)	R/W	Default	Description
31-24	R	0	Y or R.
23:16	R	0	Cb or G.
15-8	R	0	Cr or B.
7-0	R	0	Alpha[8:1].

VIU_OSD1_PROT_CTRL 0x1a2e

Bit(s)	R/W	Default	Description
31:16	R/W	0	urgent_ctrl : // unsigned , default =0
15	R.O	0	prot_en : // unsigned , default =0; 1=Borrow PROT's FIFO storage, either for rotate or non-rotate.
12: 0	R.O	0	prot_fifo_size : // unsigned , default =0; effective FIFO size when prot_en=1.

VIU_OSD1_MALI_UNPACK_CTRL 0x1a2f

Bit(s)	R/W	Default	Description
31	R/W	0	mali_unpack_en 1: OSD will unpack mali_src 0: OSD will unpack normal src
28			Alpha_div_en: alpha dividor enable
27:26			Alpha_dividor gating clk
25:24			Alpa_mapping_mode In osd,this bit should be set 0 when Alpha_div_en active,means 8 bits alpha mode
17	/	/	/
16	R/W	0	afbcd_swap_64bit:
15: 12	R/W	1	afbcd_r_reorder,change osd output order when use Dvalin src: 1: r_re = r ; 2: r_re = g; 3: r_re = b; 4: r_re = a; default: r_re = 0;
11: 8		2	afbcd_r_reorder,change osd output order when use Dvalin src: 1: g_re = r ; 2: g_re = g; 3: g_re = b; 4: g_re = a; default: r_re = 0;
7: 4		3	afbcd_r_reorder,change osd output order when use Dvalin src: 1: b_re = r ; 2: b_re = g; 3: b_re = b; 4: b_re = a; default: r_re = 0;
3: 0		4	afbcd_r_reorder,change osd output order when use Dvalin src: 1: a_re = r ; 2: a_re = g; 3: a_re = b; 4: a_re = a; default: r_re = 0;

VIU_OSD1_DIMM_CTRL 0x1adf

Bit(s)	R/W	Default	Description
30	R/W	0	OSD dimm enable,osd out will be one color when this bit active
29:0	R/W	0	Osd_dim_rgb_out ,osd out will be this vaaule when bit30 active

10.2.3.32 OSD2 registers

VIU_OSD2_CTRL_STAT 0x1a30

See:VIU_OSD1_CTRL_STAT

VIU_OSD2_CTRL_STAT2 0x1a4d

See:VIU_OSD1_CTRL_STAT2

VIU_OSD2_COLOR_ADDR 0x1a31

See:VIU_OSD1_COLOR_ADDR

VIU_OSD2_COLOR 0x1a32

See:VIU_OSD1_COLOR

VIU_OSD2_TCOLOR_AG0 0x1a37

See:VIU_OSD1_TCOLOR_AG0

VIU_OSD2_TCOLOR_AG1 0x1a38

See:VIU_OSD1_TCOLOR_AG1

VIU_OSD2_TCOLOR_AG2 0x1a39

See:VIU_OSD1_TCOLOR_AG02

VIU_OSD2_TCOLOR_AG3 0x1a3a

See:VIU_OSD1_TCOLOR_AG3

VIU_OSD2_BLK0_CFG_W0 0x1a3b

See:VIU_OSD1_BLK0_CFG_W0

VIU_OSD2_BLK0_CFG_W1 0x1a3c

See:VIU_OSD1_BLK0_CFG_W1

VIU_OSD2_BLK0_CFG_W2 0x1a3d

See:VIU_OSD1_BLK0_CFG_W2

VIU_OSD2_BLK0_CFG_W3 0x1a3e

See:VIU_OSD1_BLK0_CFG_W3

VIU_OSD2_BLK0_CFG_W4 0x1a64

See: VIU_OSD1_BLK0_CFG_W4

VIU_OSD2_BLK1_CFG_W4 0x1a65

See: VIU_OSD2_BLK1_CFG_W4

VIU_OSD2_BLK2_CFG_W4 0x1a66

See: VIU_OSD1_BLK2_CFG_W4

VIU_OSD2_FIFO_CTRL_STAT 0x1a4b

See: VIU_OSD1_FIFO_CTRL_STAT

VIU_OSD2_TEST_RDDATA 0x1a4c

See: VIU_OSD1_TEST_RDDATA

VIU_OSD2_PROT_CTRL 0x1a4e

See: VIU_OSD1_PROT_CTRL

VIU_OSD2_MALI_UNPACK_CTRL 0x1abd

See: VIU_OSD1_MALI_UNPACK_CTRL

VIU_OSD2_DIMM_CTRL 0x1acf

See: VIU_OSD1_DIMM_CTRL

10.2.3.33 OSD3 registers

VIU_OSD3_CTRL_STAT 0x3d80

See:VIU_OSD1_CTRL_STAT

VIU_OSD3_CTRL_STAT2 0x3d81

See:VIU_OSD1_CTRL_STAT2

VIU_OSD3_COLOR_ADDR 0x3d82

See:VIU_OSD1_COLOR_ADDR

VIU_OSD3_COLOR 0x3d83

See:VIU_OSD1_COLOR

VIU_OSD3_TCOLOR_AG0 0x3d84

See:VIU_OSD1_TCOLOR_AG0

VIU_OSD3_TCOLOR_AG1 0x3d85

See:VIU_OSD1_TCOLOR_AG1

VIU_OSD3_TCOLOR_AG2 0x3d86

See:VIU_OSD1_TCOLOR_AG02

VIU_OSD3_TCOLOR_AG3 0x3d87

See:VIU_OSD1_TCOLOR_AG3

VIU_OSD3_BLK0_CFG_W0 0x3d88

See:VIU_OSD1_BLK0_CFG_W0

VIU_OSD3_BLK0_CFG_W1 0x3d8c

See:VIU_OSD1_BLK0_CFG_W1

VIU_OSD3_BLK0_CFG_W2 0x3d90

See:VIU_OSD1_BLK0_CFG_W2

VIU_OSD3_BLK0_CFG_W3 0x3d94

See:VIU_OSD1_BLK0_CFG_W3

VIU_OSD3_BLK0_CFG_W4 0x3d98

See: VIU_OSD1_BLK0_CFG_W4

VIU_OSD3_BLK1_CFG_W4 0x3d99

See: VIU_OSD3_BLK1_CFG_W4

VIU_OSD3_BLK2_CFG_W4 0x3d9a

See: VIU_OSD1_BLK2_CFG_W4

VIU_OSD3_FIFO_CTRL_STAT 0x3d9c

See: VIU_OSD1_FIFO_CTRL_STAT

VIU_OSD3_TEST_RDDATA 0x3d9d

See: VIU_OSD1_TEST_RDDATA

VIU_OSD3_PROT_CTRL 0x3d9e

See: VIU_OSD1_PROT_CTRL

VIU_OSD3_MALI_UNPACK_CTRL 0x3d9f

See: VIU_OSD1_MALI_UNPACK_CTRL

VIU_OSD3_DIMM_CTRL 0x3da0

See: VIU_OSD1_DIMM_CTRL

10.2.3.34 VPP_VD1_MATRIX

VPP_VD1_MATRIX_COEF00_01 0x3290

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coefficient00, signed, 3.10
12-0	R/W	0	Coefficient01, signed, 3.10

VPP_VD1_MATRIX_COEF02_10 0x3291

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coefficient02, signed, 3.10
12-0	R/W	0	Coefficient10, signed, 3.10

VPP_VD1_MATRIX_COEF11_12 0x3292

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coefficient11, signed, 3.10
12-0	R/W	0	Coefficient12, signed, 3.10

VPP_VD1_MATRIX_COEF20_21 0x3293

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coefficient20, signed, 3.10
12-0	R/W	0	Coefficient21, signed, 3.10

VPP_VD1_MATRIX_COEF22 0x3294

Bit(s)	R/W	Default	Description
18-16	R/W	0	convrs
12-0	R/W	0	Coefficient22, signed, 3.10

VPP_VD1_MATRIX_COEF13_14 0x3295

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coefficient13, signed, 3.10
12-0	R/W	0	Coefficient14, signed, 3.10

VPP_VD1_MATRIX_COEF23_24 0x3296

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coefficient23, signed, 3.10
12-0	R/W	0	Coefficient24, signed, 3.10

VPP_VD1_MATRIX_COEF15_25 0x3297

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coefficient15, signed, 3.10
12-0	R/W	0	Coefficient25, signed, 3.10

VPP_VD1_MATRIX_CLIP 0x3298

Bit(s)	R/W	Default	Description
31-8	R/W	0x0	reserved
7-5	R/W	0x1	Matrix rs
4-3	R/W	0x10	Matrix clmod 0: only 3x3, 1: pre_offseted_ch1,ch2> pre_offseted_ch0, use the added 2x3 coef 2: pre_offseted_ch1,ch2 > 0, use the added 2x3 coef 3: pre_offseted_ch1,ch2 > 512, use the added 2x3 coef
2-0	R/W	0x10	Matrix clip enable

VPP_VD1_MATRIX_OFFSET0_1 0x3299

Bit(s)	R/W	Default	Description
27-16	R/W	0	Offset0, signed value
11-0	R/W	0	Offset1, signed value

VPP_VD1_MATRIX_OFFSET2 b 0x329a

Bit(s)	R/W	Default	Description
11-0	R/W	0	Offset2, signed value

VPP_VD1_MATRIX_PRE_OFFSET0_1 0x329b

Bit(s)	R/W	Default	Description
27-16	R/W	0	pre_Offset0, signed value
11-0	R/W	0	Pre_Offset1, signed value

VPP_VD1_MATRIX_PRE_OFFSET2 0x329c

Bit(s)	R/W	Default	Description
11-0	R/W	0	Pre_Offset2, signed value

VPP_VD1_MATRIX_EN_CTRL 0x329d

Bit(s)	R/W	Default	Description
5:4	R/W	0	Gate clock ctrl
1	R/W	0	Enable_sync_sel
0	R/W	0	Conv_en_pre

10.2.3.35 VPP_POST_MATRIX

VPP_POST_MATRIX_COEF00_01 0x32b0

See:VPP_VD1_MATRIX_COEF00_01

VPP_POST_MATRIX_COEF02_10 0x32b1

See:VPP_VD1_MATRIX_COEF02_10

VPP_POST_MATRIX_COEF11_12 0x32b2

See:VPP_VD1_MATRIX_COEF11_12

VPP_POST_MATRIX_COEF20_21 0x32b3

See:VPP_VD1_MATRIX_COEF20_21

VPP_POST_MATRIX_COEF22 0x32b4

See:VPP_VD1_MATRIX_COEF22

VPP_POST_MATRIX_COEF13_14 0x32b5

See:VPP_VD1_MATRIX_COEF13_14

VPP_POST_MATRIX_COEF23_24 0x32b6

See:VPP_VD1_MATRIX_COEF23_24

VPP_POST_MATRIX_COEF15_25 0x32b7

See:VPP_VD1_MATRIX_COEF15_25

VPP_POST_MATRIX_CLIP 0x32b8

See:VPP_VD1_MATRIX_CLIP

VPP_POST_MATRIX_OFFSET0_1 0x32b9

See:VPP_VD1_MATRIX_OFFSET0_1

VPP_POST_MATRIX_OFFSET2 0x32ba

See:VPP_VD1_MATRIX_OFFSET2

VPP_POST_MATRIX_PRE_OFFSET0_1 0x32bb

See:VPP_VD1_MATRIX_PRE_OFFSET0_1

VPP_POST_MATRIX_PRE_OFFSET2 0x32bc

See:VPP_VD1_MATRIX_PRE_OFFSET2

VPP_POST_MATRIX_EN_CTRL 0x32bd

See:VPP_VD1_MATRIX_EN_CTRL

10.2.3.36 VPP_POST2_MATRIX

VPP_POST2_MATRIX_COEF00_01 0x39a0

See:VPP_VD1_MATRIX_C OEF00_01

VPP_POST2_MATRIX_COEF02_10 0x39a1

See:VPP_VD1_MATRIX_COEF02_10

VPP_POST2_MATRIX_COEF11_12 0x39a2

See:VPP_VD1_MATRIX_COEF11_12

VPP_POST2_MATRIX_COEF20_21 0x39a3

See:VPP_VD1_MATRIX_COEF20_21

VPP_POST2_MATRIX_COEF22 0x39a4

See:VPP_VD1_MATRIX_COEF22

VPP_POST2_MATRIX_COEF13_14 0x39a5

See:VPP_VD1_MATRIX_COEF13_14

VPP_POST2_MATRIX_COEF23_24 0x39a6

See:VPP_VD1_MATRIX_COEF23_24

VPP_POST2_MATRIX_COEF15_25 0x39a7

See:VPP_VD1_MATRIX_COEF15_25

VPP_POST2_MATRIX_CLIP 0x39a8

See:VPP_VD1_MATRIX_CLIP

VPP_POST2_MATRIX_OFFSET0_1 0x39a9

See:VPP_VD1_MATRIX_OFFSET0_1

VPP_POST2_MATRIX_OFFSET2 0x39aa

See:VPP_VD1_MATRIX_OFFSET2

VPP_POST2_MATRIX_PRE_OFFSET0_1 0x39ab

See:VPP_VD1_MATRIX_PRE_OFFSET0_1

VPP_POST2_MATRIX_PRE_OFFSET2 0x39ac

See:VPP_VD1_MATRIX_PRE_OFFSET2

VPP_POST2_MATRIX_EN_CTRL 0x39ad

See:VPP_VD1_MATRIX_EN_CTRL

10.2.3.37 VPP_OSD2_MATRIX

VPP_OSD2_MATRIX_COEF00_01 0x3920

See:VPP_VD1_MATRIX_COEF00_01

VPP_OSD2_MATRIX_COEF02_10 0x3921

See:VPP_VD1_MATRIX_COEF02_10

VPP_OSD2_MATRIX_COEF11_12 0x3922

See:VPP_VD1_MATRIX_COEF11_12

VPP_OSD2_MATRIX_COEF20_21 0x3923

See:VPP_VD1_MATRIX_COEF20_21

VPP_OSD2_MATRIX_COEF22 0x3924

See:VPP_VD1_MATRIX_COEF22

VPP_OSD2_MATRIX_COEF13_14 0x3925

See:VPP_VD1_MATRIX_COEF13_14

VPP_OSD2_MATRIX_COEF23_24 0x3926

See:VPP_VD1_MATRIX_COEF23_24

VPP_OSD2_MATRIX_COEF15_25 0x3927

See:VPP_VD1_MATRIX_COEF15_25

VPP_OSD2_MATRIX_CLIP 0x3928

See:VPP_VD1_MATRIX_CLIP

VPP_OSD2_MATRIX_OFFSET0_1 0x3929

See:VPP_VD1_MATRIX_OFFSET0_1

VPP_OSD2_MATRIX_OFFSET2 0x392a

See:VPP_VD1_MATRIX_OFFSET2

VPP_OSD2_MATRIX_PRE_OFFSET0_1 0x392b

See:VPP_VD1_MATRIX_PRE_OFFSET0_1

VPP_OSD2_MATRIX_PRE_OFFSET2 0x392c

See:VPP_VD1_MATRIX_PRE_OFFSET2

VPP_OSD2_MATRIX_EN_CTRL 0x392d

See:VPP_VD1_MATRIX_EN_CTRL

10.2.3.38 VPP_WRAP_OSD1_MATRIX

VPP_WRAP_OSD1_MATRIX_COEF00_01 0x3d60

See:VPP_VD1_MATRIX_COEF00_01

VPP_WRAP_OSD1_MATRIX_COEF02_10 0x3d61

See:VPP_VD1_MATRIX_COEF02_10

VPP_WRAP_OSD1_MATRIX_COEF11_12 0x3d62

See:VPP_VD1_MATRIX_COEF11_12

VPP_WRAP_OSD1_MATRIX_COEF20_21 0x3d63

See:VPP_VD1_MATRIX_COEF20_21

VPP_WRAP_OSD1_MATRIX_COEF22 0x3d64

See:VPP_VD1_MATRIX_COEF22

VPP_WRAP_OSD1_MATRIX_COEF13_14 0x3d65

See:VPP_VD1_MATRIX_COEF13_14

VPP_WRAP_OSD1_MATRIX_COEF23_24 0x3d66

See:VPP_VD1_MATRIX_COEF23_24

VPP_WRAP_OSD1_MATRIX_COEF15_25 0x3d67

See:VPP_VD1_MATRIX_COEF15_25

VPP_WRAP_OSD1_MATRIX_CLIP 0x3d68

See:VPP_VD1_MATRIX_CLIP

VPP_WRAP_OSD1_MATRIX_OFFSET0_1 0x3d69

See:VPP_VD1_MATRIX_OFFSET0_1

VPP_WRAP_OSD1_MATRIX_OFFSET2 0x3d6a

See:VPP_VD1_MATRIX_OFFSET2

VPP_WRAP_OSD1_MATRIX_PRE_OFFSET0_1 0x3d6b

See:VPP_VD1_MATRIX_PRE_OFFSET0_1

VPP_WRAP_OSD1_MATRIX_PRE_OFFSET2 0x3d6c

See:VPP_VD1_MATRIX_PRE_OFFSET2

VPP_WRAP_OSD1_MATRIX_EN_CTRL 0x3d6d

See:VPP_VD1_MATRIX_EN_CTRL

10.2.3.39 VPP_WRAP_OSD2_MATRIX

VPP_WRAP_OSD2_MATRIX_COEF00_01 0x3d70

See:VPP_VD1_MATRIX_COEF00_01

VPP_WRAP_OSD2_MATRIX_COEF02_10 0x3d71

See:VPP_VD1_MATRIX_COEF02_10

VPP_WRAP_OSD2_MATRIX_COEF11_12 0x3d72

See:VPP_VD1_MATRIX_COEF11_12

VPP_WRAP_OSD2_MATRIX_COEF20_21 0x3d73

See:VPP_VD1_MATRIX_COEF20_21

VPP_WRAP_OSD2_MATRIX_COEF22 0x3d74

See:VPP_VD1_MATRIX_COEF22

VPP_WRAP_OSD2_MATRIX_COEF13_14 0x3d75

See:VPP_VD1_MATRIX_COEF13_14

VPP_WRAP_OSD2_MATRIX_COEF23_24 0x3d76

See:VPP_VD1_MATRIX_COEF23_24

VPP_WRAP_OSD2_MATRIX_COEF15_25 0x3d77

See:VPP_VD1_MATRIX_COEF15_25

VPP_WRAP_OSD2_MATRIX_CLIP 0x3d78

See:VPP_VD1_MATRIX_CLIP

VPP_WRAP_OSD2_MATRIX_OFFSET0_1 0x3d79

See:VPP_VD1_MATRIX_OFFSET0_1

VPP_WRAP_OSD2_MATRIX_OFFSET2 0x3d7a

See:VPP_VD1_MATRIX_OFFSET2

VPP_WRAP_OSD2_MATRIX_PRE_OFFSET0_1 0x3d7b

See:VPP_VD1_MATRIX_PRE_OFFSET0_1

VPP_WRAP_OSD2_MATRIX_PRE_OFFSET2 0x3d7c

See:VPP_VD1_MATRIX_PRE_OFFSET2

VPP_WRAP_OSD2_MATRIX_EN_CTRL 0x3d7d

See:VPP_VD1_MATRIX_EN_CTRL

10.2.3.40 VPP_WRAP_OSD3_MATRIX

VPP_WRAP_OSD3_MATRIX_COEF00_01 0x3db0

See:VPP_VD1_MATRIX_COEF00_01

VPP_WRAP_OSD3_MATRIX_COEF02_10 0x3db1

See:VPP_VD1_MATRIX_COEF02_10

VPP_WRAP_OSD3_MATRIX_COEF11_12 0x3db2

See:VPP_VD1_MATRIX_COEF11_12

VPP_WRAP_OSD3_MATRIX_COEF20_21 0x3db3

See:VPP_VD1_MATRIX_COEF20_21

VPP_WRAP_OSD3_MATRIX_COEF22 0x3db4

See:VPP_VD1_MATRIX_COEF22

VPP_WRAP_OSD3_MATRIX_COEF13_14 0x3db5

See:VPP_VD1_MATRIX_COEF13_14

VPP_WRAP_OSD3_MATRIX_COEF23_24 0x3db6

See:VPP_VD1_MATRIX_COEF23_24

VPP_WRAP_OSD3_MATRIX_COEF15_25 0x3db7

See:VPP_VD1_MATRIX_COEF15_25

VPP_WRAP_OSD3_MATRIX_CLIP 0x3db8

See:VPP_VD1_MATRIX_CLIP

VPP_WRAP_OSD3_MATRIX_OFFSET0_1 0x3db9

See:VPP_VD1_MATRIX_OFFSET0_1

VPP_WRAP_OSD3_MATRIX_OFFSET2 0x3dba

See:VPP_VD1_MATRIX_OFFSET2

VPP_WRAP_OSD3_MATRIX_PRE_OFFSET0_1 0x3dbb

See:VPP_VD1_MATRIX_PRE_OFFSET0_1

VPP_WRAP_OSD3_MATRIX_PRE_OFFSET2 0x3dbc

See:VPP_VD1_MATRIX_PRE_OFFSET2

VPP_WRAP_OSD3_MATRIX_EN_CTRL 0x3dbd

See:VPP_VD1_MATRIX_EN_CTRL

10.2.3.41 HDR

VDIN0_HDR2_CTRL 0x1280

Bit(s)	R/W	Default	Description
20:18	R/W	0	reg_din_swap : // unsigned , default = 0
17	R/W	0	reg_out_fmt : // unsigned , default = 0
16	R/W	0	reg_only_mat : // unsigned , default = 0
13	R/W	0	reg_VDIN0_HDR2_top_en : // unsigned , default = 0
12	R/W	1	reg_cgain_mode : // unsigned , default = 1
7: 6	R/W	1	reg_gmut_mode : // unsigned , default = 1
5	R/W	0	reg_in_shift : // unsigned , default = 0
4	R/W	1	reg_in_fmt : // unsigned , default = 1
3	R/W	1	reg_eo_enable : // unsigned , default = 1
2	R/W	1	reg_oe_enable : // unsigned , default = 1
1	R/W	1	reg_ogain_enable : // unsigned , default = 1
0	R/W	1	reg_cgain_enable : // unsigned , default = 1

VDIN0_HDR2_CLK_GATE 0x1281

Bit(s)	R/W	Default	Description
31:30	R/W	0	clk_tm : gate clock ctrl (main clock) // unsigned , default = 0
29:28	R/W	0	output : matrix clock gate ctrl // unsigned , default = 0
25:24	R/W	0	input : matrix clock gate ctrl // unsigned , default = 0
23:22	R/W	0	hdr : top cbus clock gate ctrl // unsigned , default = 0
21:20	R/W	0	eotf : cbus clock gate ctrl // unsigned , default = 0
19:18	R/W	0	oetf : cbus clock gate ctrl // unsigned , default = 0

17:16	R/W	0	gamma : mult cbus clock gate ctrl // unsigned , default = 0
15:14	R/W	0	adaptive : cbus scaler clock gate ctrl // unsigned , default = 0
13:12	R/W	0	cgain : cbus clock gate ctrl // unsigned , default = 0
11:10	R/W	0	eotf : clock gate ctrl // unsigned , default = 0
9:8	R/W	0	oetf : clock gate ctrl // unsigned , default = 0
7:6	R/W	0	gamma : mult clock gate ctrl // unsigned , default = 0
5:4	R/W	0	adaptive : scaler clock gate ctrl // unsigned , default = 0
3:2	R/W	0	uv : gain clock gate ctrl // unsigned , default = 0
1:0	R/W	0	cgain : clock gate ctrl // unsigned , default = 0

VDIN0_HDR2_MATRIXI_COEF00_01 0x1282

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0
12:0	R/W	0	coef01 : // signed , default = 0

VDIN0_HDR2_MATRIXI_COEF02_10 0x1283

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

VDIN0_HDR2_MATRIXI_COEF11_12 0x1284

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

VDIN0_HDR2_MATRIXI_COEF20_21 0x1285

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

VDIN0_HDR2_MATRIXI_COEF22 0x1286

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

VDIN0_HDR2_MATRIXI_COEF30_31 0x1287

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

VDIN0_HDR2_MATRIXI_COEF32_40 0x1288

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

VDIN0_HDR2_MATRIXI_COEF41_42 0x1289

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

VDIN0_HDR2_MATRIXI_OFFSET0_1 0x128A

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

VDIN0_HDR2_MATRIXI_OFFSET2 0x128B

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

VDIN0_HDR2_MATRIXI_PRE_OFFSET0_1 0x128C

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

VDIN0_HDR2_MATRIXI_PRE_OFFSET2 0x128D

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

VDIN0_HDR2_MATRIXO_COEF00_01 0x128E

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0
12:0	R/W	0	coef01 : // signed , default = 0

VDIN0_HDR2_MATRIXO_COEF02_10 0x128F

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

VDIN0_HDR2_MATRIXO_COEF11_12 0x1290

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

VDIN0_HDR2_MATRIXO_COEF20_21 0x1291

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

VDIN0_HDR2_MATRIXO_COEF22 0x1292

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

VDIN0_HDR2_MATRIXO_COEF30_31 0x1293

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

VDIN0_HDR2_MATRIXO_COEF32_40 0x1294

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

VDIN0_HDR2_MATRIXO_COEF41_42 0x1295

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

VDIN0_HDR2_MATRIXO_OFFSET0_1 0x1296

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0

10:0	R/W	0	offset1 : // signed , default = 0

VDIN0_HDR2_MATRIXO_OFFSET2 0x1297

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

VDIN0_HDR2_MATRIXO_PRE_OFFSET0_1 0x1298

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

VDIN0_HDR2_MATRIXO_PRE_OFFSET2 0x1299

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

VDIN0_HDR2_MATRIXI_CLIP 0x129A

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

VDIN0_HDR2_MATRIXO_CLIP 0x129B

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

VDIN0_HDR2_CGAIN_OFFT 0x129C

Bit(s)	R/W	Default	Description
26:16	R/W	0	reg_cgain_oft2 : // signed , default = 0
10:0	R/W	0	reg_cgain_oft1 : // signed , default = 0

VDIN0_EOTF_LUT_ADDR_PORT 0x129E

Bit(s)	R/W	Default	Description
7:0	R/W	0	eotf_lut_addr : // unsigned , default = 0

VDIN0_EOTF_LUT_DATA_PORT 0x129F

Bit(s)	R/W	Default	Description
19:0	R/W	0	eotf_lut_data : // unsigned , default = 0

VDIN0_OETF_LUT_ADDR_PORT 0x12A0

Bit(s)	R/W	Default	Description
7:0	R/W	0	oetf_lut_addr : // unsigned , default = 0

VDIN0_OETF_LUT_DATA_PORT 0x12A1

Bit(s)	R/W	Default	Description
11:0	R/W	0	oetf_lut_data : // unsigned , default = 0

VDIN0_CGAIN_LUT_ADDR_PORT 0x12A2

Bit(s)	R/W	Default	Description
7:0	R/W	0	cgain_lut_addr : // unsigned , default = 0

VDIN0_CGAIN_LUT_DATA_PORT 0x12A3

Bit(s)	R/W	Default	Description
11:0	R/W	0	cgain_lut_data : // unsigned , default = 0

VDIN0_HDR2_CGAIN_COEF0 0x12A4

Bit(s)	R/W	Default	Description
27:16	R/W	0	reg_cgain_coef1 : // unsigned , default = 0
11:0	R/W	0	reg_cgain_coef0 : //unsigned , default = 0

VDIN0_HDR2_CGAIN_COEF1 0x12A5

Bit(s)	R/W	Default	Description
11:0	R/W	0	reg_cgain_coef2 : // unsigned , default = 0

VDIN0_OGAIN_LUT_ADDR_PORT 0x12A6

Bit(s)	R/W	Default	Description
7:0	R/W	0	ogain_lut_addr : // unsigned , default = 0

VDIN0_OGAIN_LUT_DATA_PORT 0x12A7

Bit(s)	R/W	Default	Description
11:0	R/W	0	ogain_lut_data : // unsigned , default = 0

VDIN0_HDR2_ADPS_CTRL 0x12A8

Bit(s)	R/W	Default	Description
6	R/W	1	reg_adpscl_bypass2 : // unsigned , default = 1
5	R/W	1	reg_adpscl_bypass1 : // unsigned , default = 1
4	R/W	1	reg_adpscl_bypass0 : // unsigned , default = 1
1:0	R/W	1	reg_adpscl_mode : // unsigned , default = 1

VDIN0_HDR2_ADPS_ALPHA0 0x12A9

Bit(s)	R/W	Default	Description
29:16	R/W	0x1000	reg_adpscl_alpha1 : // unsigned , default = 0x1000
13:0	R/W	0x1000	reg_adpscl_alpha0 : // unsigned , default = 0x1000

VDIN0_HDR2_ADPS_ALPHA1 0x12AA

Bit(s)	R/W	Default	Description
27:24	R/W	0xc	reg_adpscl_shift0 : // unsigned , default = 0xc
23:20	R/W	0xc	reg_adpscl_shift1 : // unsigned , default = 0xc
19:16	R/W	0xc	reg_adpscl_shift2 : // unsigned , default = 0xc
13:0	R/W	0x1000	reg_adpscl_alpha2 : // unsigned , default = 0x1000

VDIN0_HDR2_ADPS_BETA0 0x12AB

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta0_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta0 : // unsigned , default = 0xfc000

VDIN0_HDR2_ADPS_BETA1 0x12AC

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta1_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta1 : // unsigned , default = 0xfc000

VDIN0_HDR2_ADPS_BETA2 0x12AD

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta2_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta2 : // unsigned , default = 0xfc000

VDIN0_HDR2_ADPS_COEF0 0x12AE

Bit(s)	R/W	Default	Description
27:16	R/W	460	reg_adpscl_ys_coef1 : // unsigned , default = 460
11:0	R/W	1188	reg_adpscl_ys_coef0 : // unsigned , default = 1188

VDIN0_HDR2_ADPS_COEF1 0x12AF

Bit(s)	R/W	Default	Description
11:0	R/W	104	reg_adpscl_ys_coef2 : // unsigned , default = 104

VDIN0_HDR2_GMUT_CTRL 0x12B0

Bit(s)	R/W	Default	Description
3:0	R/W	14	reg_gmut_shift : // unsigned , default = 14

VDIN0_HDR2_GMUT_COEF0 0x12B1

Bit(s)	R/W	Default	Description
31:16	R/W	674	reg_gmut_coef01 : // unsigned , default = 674
15:0	R/W	1285	reg_gmut_coef00 : // unsigned , default = 1285

VDIN0_HDR2_GMUT_COEF1 0x12B2

Bit(s)	R/W	Default	Description
31:16	R/W	142	reg_gmut_coef10 : // unsigned , default = 142
15:0	R/W	89	reg_gmut_coef02 : // unsigned , default = 89

VDIN0_HDR2_GMUT_COEF2 0x12B3

Bit(s)	R/W	Default	Description
31:16	R/W	23	reg_gmut_coef12 : // unsigned , default = 23
15:0	R/W	1883	reg_gmut_coef11 : // unsigned , default = 1883

VDIN0_HDR2_GMUT_COEF3 0x12B4

Bit(s)	R/W	Default	Description
31:16	R/W	180	reg_gmut_coef21 : // unsigned , default = 180
15:0	R/W	34	reg_gmut_coef20 : // unsigned , default = 34

VDIN0_HDR2_GMUT_COEF4 0x12B5

Bit(s)	R/W	Default	Description
15:0	R/W	1834	reg_gmut_coef22 : // unsigned , default = 1834

VDIN0_HDR2_PIPE_CTRL1 0x12B6

Bit(s)	R/W	Default	Description
31:24	R/W	4	vblank_num_oetf : // unsigned , default = 4
23:16	R/W	4	hblank_num_oetf : // unsigned , default = 4
15:8	R/W	10	vblank_num_eotf : // unsigned , default = 10
7:0	R/W	10	hblank_num_eotf : // unsigned , default = 10

VDIN0_HDR2_PIPE_CTRL2 0x12B7

Bit(s)	R/W	Default	Description
31:24	R/W	10	vblank_num_cgain : // unsigned , default = 10
23:16	R/W	10	hblank_num_cgain : // unsigned , default = 10
15:8	R/W	11	vblank_num_gmut : // unsigned , default = 11
7:0	R/W	11	hblank_num_gmut : // unsigned , default = 11

VDIN0_HDR2_PIPE_CTRL3 0x12B8

Bit(s)	R/W	Default	Description
31:24	R/W	22	vblank_num_adps : // unsigned , default = 22
23:16	R/W	2	hblank_num_adps : // unsigned , default = 2
15:8	R/W	4	vblank_num_uv : // unsigned , default = 4
7:0	R/W	4	hblank_num_uv : // unsigned , default = 4

VDIN0_HDR2_PROC_WIN1 0x12B9

Bit(s)	R/W	Default	Description
28:16	R/W	0	proc_win_h_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_h_st : // unsigned , default = 0

VDIN0_HDR2_PROC_WIN2 0x12BA

Bit(s)	R/W	Default	Description
31	R/W	0	proc_win_gmut_en : // unsigned , default = 0
30	R/W	0	proc_win_adps_en : // unsigned , default = 0
29	R/W	0	proc_win_cgain_en : // unsigned , default = 0
28:16	R/W	0	proc_win_v_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_v_st : // unsigned , default = 0

VDIN0_HDR2_MATRIXI_EN_CTRL 0x12BB

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

VDIN0_HDR2_MATRIXO_EN_CTRL 0x12BC

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

VDIN1_HDR2_CTRL 0x1380

Bit(s)	R/W	Default	Description
20:18	R/W	0	reg_din_swap : // unsigned , default = 0
17	R/W	0	reg_out_fmt : // unsigned , default = 0
16	R/W	0	reg_only_mat : // unsigned , default = 0
13	R/W	0	reg_VDIN0_HDR2_top_en : // unsigned , default = 0
12	R/W	1	reg_cgain_mode : // unsigned , default = 1
7: 6	R/W	1	reg_gmut_mode : // unsigned , default = 1
5	R/W	0	reg_in_shift : // unsigned , default = 0
4	R/W	1	reg_in_fmt : // unsigned , default = 1
3	R/W	1	reg_eo_enable : // unsigned , default = 1
2	R/W	1	reg_oe_enable : // unsigned , default = 1
1	R/W	1	reg_ogain_enable : // unsigned , default = 1
0	R/W	1	reg_cgain_enable : // unsigned , default = 1

VDIN1_HDR2_CLK_GATE 0x1381

Bit(s)	R/W	Default	Description
31:30	R/W	0	clk_tm : gate clock ctrl (main clock) // unsigned , default = 0
29:28	R/W	0	output : matrix clock gate ctrl // unsigned , default = 0
25:24	R/W	0	input : matrix clock gate ctrl // unsigned , default = 0
23:22	R/W	0	hdr : top cbus clock gate ctrl // unsigned , default = 0

21:20	R/W	0	eotf : cbus clock gate ctrl // unsigned , default = 0
19:18	R/W	0	oetf : cbus clock gate ctrl // unsigned , default = 0
17:16	R/W	0	gamma : mult cbus clock gate ctrl // unsigned , default = 0
15:14	R/W	0	adaptive : cbus scaler clock gate ctrl // unsigned , default = 0
13:12	R/W	0	cgain : cbus clock gate ctrl // unsigned , default = 0
11:10	R/W	0	eotf : clock gate ctrl // unsigned , default = 0
9:8	R/W	0	oetf : clock gate ctrl // unsigned , default = 0
7:6	R/W	0	gamma : mult clock gate ctrl // unsigned , default = 0
5:4	R/W	0	adaptive : scaler clock gate ctrl // unsigned , default = 0
3:2	R/W	0	uv : gain clock gate ctrl // unsigned , default = 0
1:0	R/W	0	cgain : clock gate ctrl // unsigned , default = 0

VDIN1_HDR2_MATRIXI_COEF00_01 0x1382

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0
12:0	R/W	0	coef01 : // signed , default = 0

VDIN1_HDR2_MATRIXI_COEF02_10 0x1383

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

VDIN1_HDR2_MATRIXI_COEF11_12 0x1384

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

VDIN1_HDR2_MATRIXI_COEF20_21 0x1385

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

VDIN1_HDR2_MATRIXI_COEF22 0x1386

Bit(s)	R/W	Default	Description

12:0	R/W	0	coef22 : // signed , default = 0

VDIN1_HDR2_MATRIXI_COEF30_31 0x1387

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

VDIN1_HDR2_MATRIXI_COEF32_40 0x1388

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

VDIN1_HDR2_MATRIXI_COEF41_42 0x1389

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

VDIN1_HDR2_MATRIXI_OFFSET0_1 0x138A

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

VDIN1_HDR2_MATRIXI_OFFSET2 0x138B

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

VDIN1_HDR2_MATRIXI_PRE_OFFSET0_1 0x138C

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

VDIN1_HDR2_MATRIXI_PRE_OFFSET2 0x138D

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

VDIN1_HDR2_MATRIXO_COEF00_01 0x138E

Bit(s)	R/W	Default	Description

28:16	R/W	0	coef00 : // signed , default = 0
12:0	R/W	0	coef01 : // signed , default = 0

VDIN1_HDR2_MATRIXO_COEF02_10 0x138F

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

VDIN1_HDR2_MATRIXO_COEF11_12 0x1390

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

VDIN1_HDR2_MATRIXO_COEF20_21 0x1391

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

VDIN1_HDR2_MATRIXO_COEF22 0x1392

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

VDIN1_HDR2_MATRIXO_COEF30_31 0x1393

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

VDIN1_HDR2_MATRIXO_COEF32_40 0x1394

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

VDIN1_HDR2_MATRIXO_COEF41_42 0x1395

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

VDIN1_HDR2_MATRIXO_OFFSET0_1 0x1396

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

VDIN1_HDR2_MATRIXO_OFFSET2 0x1397

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

VDIN1_HDR2_MATRIXO_PRE_OFFSET0_1 0x1398

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

VDIN1_HDR2_MATRIXO_PRE_OFFSET2 0x1399

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

VDIN1_HDR2_MATRIXI_CLIP 0x139A

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

VDIN1_HDR2_MATRIXO_CLIP 0x139B

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

VDIN1_HDR2_CGAIN_OFFT 0x139C

Bit(s)	R/W	Default	Description
26:16	R/W	0	reg_cgain_oft2 : // signed , default = 0
10:0	R/W	0	reg_cgain_oft1 : // signed , default = 0

VDIN1_EOTF_LUT_ADDR_PORT 0x139E

Bit(s)	R/W	Default	Description

7:0	R/W	0	eotf_lut_addr : // unsigned , default = 0

VDIN1_EOTF_LUT_DATA_PORT 0x139F

Bit(s)	R/W	Default	Description
19:0	R/W	0	eotf_lut_data : // unsigned , default = 0

VDIN1_OETF_LUT_ADDR_PORT 0x13A0

Bit(s)	R/W	Default	Description
7:0	R/W	0	oetf_lut_addr : // unsigned , default = 0

VDIN1_OETF_LUT_DATA_PORT 0x13A1

Bit(s)	R/W	Default	Description
11:0	R/W	0	oetf_lut_data : // unsigned , default = 0

VDIN1_CGAIN_LUT_ADDR_PORT 0x13A2

Bit(s)	R/W	Default	Description
7:0	R/W	0	cgain_lut_addr : // unsigned , default = 0

VDIN1_CGAIN_LUT_DATA_PORT 0x13A3

Bit(s)	R/W	Default	Description
11:0	R/W	0	cgain_lut_data : // unsigned , default = 0

VDIN1_HDR2_CGAIN_COEF0 0x13A4

Bit(s)	R/W	Default	Description
27:16	R/W	0	reg_cgain_coef1 : // unsigned , default = 0
11:0	R/W	0	reg_cgain_coef0 : //unsigned , default = 0

VDIN1_HDR2_CGAIN_COEF1 0x13A5

Bit(s)	R/W	Default	Description
11:0	R/W	0	reg_cgain_coef2 : // unsigned , default = 0

VDIN1_OGAIN_LUT_ADDR_PORT 0x13A6

Bit(s)	R/W	Default	Description
7:0	R/W	0	ogain_lut_addr : // unsigned , default = 0

VDIN1_OGAIN_LUT_DATA_PORT 0x13A7

Bit(s)	R/W	Default	Description
11:0	R/W	0	ogain_lut_data : // unsigned , default = 0

VDIN1_HDR2_ADPS_CTRL 0x13A8

Bit(s)	R/W	Default	Description
6	R/W	1	reg_adpscl_bypass2 : // unsigned , default = 1
5	R/W	1	reg_adpscl_bypass1 : // unsigned , default = 1
4	R/W	1	reg_adpscl_bypass0 : // unsigned , default = 1
1:0	R/W	1	reg_adpscl_mode : // unsigned , default = 1

VDIN1_HDR2_ADPS_ALPHA0 0x13A9

Bit(s)	R/W	Default	Description
29:16	R/W	0x1000	reg_adpscl_alpha1 : // unsigned , default = 0x1000
13:0	R/W	0x1000	reg_adpscl_alpha0 : // unsigned , default = 0x1000

VDIN1_HDR2_ADPS_ALPHA1 0x13AA

Bit(s)	R/W	Default	Description
27:24	R/W	0xc	reg_adpscl_shift0 : // unsigned , default = 0xc
23:20	R/W	0xc	reg_adpscl_shift1 : // unsigned , default = 0xc
19:16	R/W	0xc	reg_adpscl_shift2 : // unsigned , default = 0xc
13:0	R/W	0x1000	reg_adpscl_alpha2 : // unsigned , default = 0x1000

VDIN1_HDR2_ADPS_BETA0 0x13AB

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta0_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta0 : // unsigned , default = 0xfc000

VDIN1_HDR2_ADPS_BETA1 0x13AC

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta1_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta1 : // unsigned , default = 0xfc000

VDIN1_HDR2_ADPS_BETA2 0x13AD

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta2_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta2 : // unsigned , default = 0xfc000

VDIN1_HDR2_ADPS_COEF0 0x13AE

Bit(s)	R/W	Default	Description
27:16	R/W	460	reg_adpscl_ys_coef1 : // unsigned , default = 460
11:0	R/W	1188	reg_adpscl_ys_coef0 : // unsigned , default = 1188

VDIN1_HDR2_ADPS_COEF1 0x13AF

Bit(s)	R/W	Default	Description
11:0	R/W	104	reg_adpscl_ys_coef2 : // unsigned , default = 104

VDIN1_HDR2_GMUT_CTRL 0x13B0

Bit(s)	R/W	Default	Description
3:0	R/W	14	reg_gmut_shift : // unsigned , default = 14

VDIN1_HDR2_GMUT_COEF0 0x13B1

Bit(s)	R/W	Default	Description
31:16	R/W	674	reg_gmut_coef01 : // unsigned , default = 674
15:0	R/W	1285	reg_gmut_coef00 : // unsigned , default = 1285

VDIN1_HDR2_GMUT_COEF1 0x13B2

Bit(s)	R/W	Default	Description
31:16	R/W	142	reg_gmut_coef10 : // unsigned , default = 142
15:0	R/W	89	reg_gmut_coef02 : // unsigned , default = 89

VDIN1_HDR2_GMUT_COEF2 0x13B3

Bit(s)	R/W	Default	Description
31:16	R/W	23	reg_gmut_coef12 : // unsigned , default = 23
15:0	R/W	1883	reg_gmut_coef11 : // unsigned , default = 1883

VDIN1_HDR2_GMUT_COEF3 0x13B4

Bit(s)	R/W	Default	Description
31:16	R/W	180	reg_gmut_coef21 : // unsigned , default = 180
15:0	R/W	34	reg_gmut_coef20 : // unsigned , default = 34

VDIN1_HDR2_GMUT_COEF4 0x13B5

Bit(s)	R/W	Default	Description
15:0	R/W	1834	reg_gmut_coef22 : // unsigned , default = 1834

VDIN1_HDR2_PIPE_CTRL1 0x13B6

Bit(s)	R/W	Default	Description
31:24	R/W	4	vblank_num_oetf : // unsigned , default = 4
23:16	R/W	4	hblank_num_oetf : // unsigned , default = 4
15:8	R/W	10	vblank_num_eotf : // unsigned , default = 10
7:0	R/W	10	hblank_num_eotf : // unsigned , default = 10

VDIN1_HDR2_PIPE_CTRL2 0x13B7

Bit(s)	R/W	Default	Description
31:24	R/W	10	vblank_num_cgain : // unsigned , default = 10
23:16	R/W	10	hblank_num_cgain : // unsigned , default = 10
15:8	R/W	11	vblank_num_gmut : // unsigned , default = 11
7:0	R/W	11	hblank_num_gmut : // unsigned , default = 11

VDIN1_HDR2_PIPE_CTRL3 0x13B8

Bit(s)	R/W	Default	Description
31:24	R/W	22	vblank_num_adps : // unsigned , default = 22
23:16	R/W	2	hblank_num_adps : // unsigned , default = 2
15:8	R/W	4	vblank_num_uv : // unsigned , default = 4
7:0	R/W	4	hblank_num_uv : // unsigned , default = 4

VDIN1_HDR2_PROC_WIN1 0x13B9

Bit(s)	R/W	Default	Description
28:16	R/W	0	proc_win_h_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_h_st : // unsigned , default = 0

VDIN1_HDR2_PROC_WIN2 0x13BA

Bit(s)	R/W	Default	Description
31	R/W	0	proc_win_gmut_en : // unsigned , default = 0
30	R/W	0	proc_win_adps_en : // unsigned , default = 0
29	R/W	0	proc_win_cgain_en : // unsigned , default = 0
28:16	R/W	0	proc_win_v_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_v_st : // unsigned , default = 0

VDIN1_HDR2_MATRIXI_EN_CTRL 0x13BB

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

VDIN1_HDR2_MATRIXO_EN_CTRL 0x13BC

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

VD1_HDR2_CTRL 0x3800

Bit(s)	R/W	Default	Description
20:18	R/W	0	reg_din_swap : // unsigned , default = 0
17	R/W	0	reg_out_fmt : // unsigned , default = 0
16	R/W	0	reg_only_mat : // unsigned , default = 0
13	R/W	0	reg_VDIN0_HDR2_top_en : // unsigned , default = 0
12	R/W	1	reg_cgain_mode : // unsigned , default = 1
7: 6	R/W	1	reg_gmut_mode : // unsigned , default = 1
5	R/W	0	reg_in_shift : // unsigned , default = 0
4	R/W	1	reg_in_fmt : // unsigned , default = 1
3	R/W	1	reg_eo_enable : // unsigned , default = 1
2	R/W	1	reg_oe_enable : // unsigned , default = 1
1	R/W	1	reg_ogain_enable : // unsigned , default = 1
0	R/W	1	reg_cgain_enable : // unsigned , default = 1

VD1_HDR2_CLK_GATE 0x3801

Bit(s)	R/W	Default	Description
31:30	R/W	0	clk_tm : gate clock ctrl (main clock) // unsigned , default = 0
29:28	R/W	0	output : matrix clock gate ctrl // unsigned , default = 0
25:24	R/W	0	input : matrix clock gate ctrl // unsigned , default = 0
23:22	R/W	0	hdr : top cbus clock gate ctrl // unsigned , default = 0

21:20	R/W	0	eotf : cbus clock gate ctrl // unsigned , default = 0
19:18	R/W	0	oetf : cbus clock gate ctrl // unsigned , default = 0
17:16	R/W	0	gamma : mult cbus clock gate ctrl // unsigned , default = 0
15:14	R/W	0	adaptive : cbus scaler clock gate ctrl // unsigned , default = 0
13:12	R/W	0	cgain : cbus clock gate ctrl // unsigned , default = 0
11:10	R/W	0	eotf : clock gate ctrl // unsigned , default = 0
9:8	R/W	0	oetf : clock gate ctrl // unsigned , default = 0
7:6	R/W	0	gamma : mult clock gate ctrl // unsigned , default = 0
5:4	R/W	0	adaptive : scaler clock gate ctrl // unsigned , default = 0
3:2	R/W	0	uv : gain clock gate ctrl // unsigned , default = 0
1:0	R/W	0	cgain : clock gate ctrl // unsigned , default = 0

VD1_HDR2_MATRIXI_COEF00_01 0x3802

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0
12:0	R/W	0	coef01 : // signed , default = 0

VD1_HDR2_MATRIXI_COEF02_10 0x3803

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

VD1_HDR2_MATRIXI_COEF11_12 0x3804

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

VD1_HDR2_MATRIXI_COEF20_21 0x3805

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

VD1_HDR2_MATRIXI_COEF22 0x3806

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

VD1_HDR2_MATRIXI_COEF30_31 0x3807

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

VD1_HDR2_MATRIXI_COEF32_40 0x3808

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

VD1_HDR2_MATRIXI_COEF41_42 0x3809

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

VD1_HDR2_MATRIXI_OFFSET0_1 0x380A

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

VD1_HDR2_MATRIXI_OFFSET2 0x380B

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

VD1_HDR2_MATRIXI_PRE_OFFSET0_1 0x380C

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

VD1_HDR2_MATRIXI_PRE_OFFSET2 0x380D

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

VD1_HDR2_MATRIXO_COEF00_01 0x380E

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0

12:0	R/W	0	coef01 : // signed , default = 0

VD1_HDR2_MATRIXO_COEF02_10 0x380F

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

VD1_HDR2_MATRIXO_COEF11_12 0x3810

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

VD1_HDR2_MATRIXO_COEF20_21 0x3811

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

VD1_HDR2_MATRIXO_COEF22 0x3812

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

VD1_HDR2_MATRIXO_COEF30_31 0x3813

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

VD1_HDR2_MATRIXO_COEF32_40 0x3814

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

VD1_HDR2_MATRIXO_COEF41_42 0x3815

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

VD1_HDR2_MATRIXO_OFFSET0_1 0x3816

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

VD1_HDR2_MATRIXO_OFFSET2 0x3817

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

VD1_HDR2_MATRIXO_PRE_OFFSET0_1 0x3818

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

VD1_HDR2_MATRIXO_PRE_OFFSET2 0x3819

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

VD1_HDR2_MATRIXI_CLIP 0x381A

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

VD1_HDR2_MATRIXO_CLIP 0x381B

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

VD1_HDR2_CGAIN_OFFT 0x381C

Bit(s)	R/W	Default	Description
26:16	R/W	0	reg_cgain_oft2 : // signed , default = 0
10:0	R/W	0	reg_cgain_oft1 : // signed , default = 0

VD1_EOTF_LUT_ADDR_PORT 0x381E

Bit(s)	R/W	Default	Description

7:0	R/W	0	eotf_lut_addr : // unsigned , default = 0

VD1_EOTF_LUT_DATA_PORT 0x381F

Bit(s)	R/W	Default	Description
19:0	R/W	0	eotf_lut_data : // unsigned , default = 0

VD1_OETF_LUT_ADDR_PORT 0x3820

Bit(s)	R/W	Default	Description
7:0	R/W	0	oetf_lut_addr : // unsigned , default = 0

VD1_OETF_LUT_DATA_PORT 0x3821

Bit(s)	R/W	Default	Description
11:0	R/W	0	oetf_lut_data : // unsigned , default = 0

VD1_CGAIN_LUT_ADDR_PORT 0x3822

Bit(s)	R/W	Default	Description
7:0	R/W	0	cgain_lut_addr : // unsigned , default = 0

VD1_CGAIN_LUT_DATA_PORT 0x3823

Bit(s)	R/W	Default	Description
11:0	R/W	0	cgain_lut_data : // unsigned , default = 0

VD1_HDR2_CGAIN_COEF0 0x3824

Bit(s)	R/W	Default	Description
27:16	R/W	0	reg_cgain_coef1 : // unsigned , default = 0
11:0	R/W	0	reg_cgain_coef0 : //unsigned , default = 0

VD1_HDR2_CGAIN_COEF1 0x3825

Bit(s)	R/W	Default	Description
11:0	R/W	0	reg_cgain_coef2 : // unsigned , default = 0

VD1_OGAIN_LUT_ADDR_PORT 0x3826

Bit(s)	R/W	Default	Description
7:0	R/W	0	ogain_lut_addr : // unsigned , default = 0

VD1_OGAIN_LUT_DATA_PORT 0x3827

Bit(s)	R/W	Default	Description
11:0	R/W	0	ogain_lut_data : // unsigned , default = 0

VD1_HDR2_ADPS_CTRL 0x3828

Bit(s)	R/W	Default	Description
6	R/W	1	reg_adpscl_bypass2 : // unsigned , default = 1
5	R/W	1	reg_adpscl_bypass1 : // unsigned , default = 1
4	R/W	1	reg_adpscl_bypass0 : // unsigned , default = 1
1:0	R/W	1	reg_adpscl_mode : // unsigned , default = 1

VD1_HDR2_ADPS_ALPHA0 0x3829

Bit(s)	R/W	Default	Description
29:16	R/W	0x1000	reg_adpscl_alpha1 : // unsigned , default = 0x1000
13:0	R/W	0x1000	reg_adpscl_alpha0 : // unsigned , default = 0x1000

VD1_HDR2_ADPS_ALPHA1 0x382A

Bit(s)	R/W	Default	Description
27:24	R/W	0xc	reg_adpscl_shift0 : // unsigned , default = 0xc
23:20	R/W	0xc	reg_adpscl_shift1 : // unsigned , default = 0xc
19:16	R/W	0xc	reg_adpscl_shift2 : // unsigned , default = 0xc
13:0	R/W	0x1000	reg_adpscl_alpha2 : // unsigned , default = 0x1000

VD1_HDR2_ADPS_BETA0 0x382B

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta0_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta0 : // unsigned , default = 0xfc000

VD1_HDR2_ADPS_BETA1 0x382C

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta1_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta1 : // unsigned , default = 0xfc000

VD1_HDR2_ADPS_BETA2 0x382D

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta2_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta2 : // unsigned , default = 0xfc000

VD1_HDR2_ADPS_COEF0 0x382E

Bit(s)	R/W	Default	Description
27:16	R/W	460	reg_adpscl_ys_coef1 : // unsigned , default = 460
11:0	R/W	1188	reg_adpscl_ys_coef0 : // unsigned , default = 1188

VD1_HDR2_ADPS_COEF1 0x382F

Bit(s)	R/W	Default	Description
11:0	R/W	104	reg_adpscl_ys_coef2 : // unsigned , default = 104

VD1_HDR2_GMUT_CTRL 0x3830

Bit(s)	R/W	Default	Description
3:0	R/W	14	reg_gmut_shift : // unsigned , default = 14

VD1_HDR2_GMUT_COEF0 0x3831

Bit(s)	R/W	Default	Description
31:16	R/W	674	reg_gmut_coef01 : // unsigned , default = 674
15:0	R/W	1285	reg_gmut_coef00 : // unsigned , default = 1285

VD1_HDR2_GMUT_COEF1 0x3832

Bit(s)	R/W	Default	Description
31:16	R/W	142	reg_gmut_coef10 : // unsigned , default = 142
15:0	R/W	89	reg_gmut_coef02 : // unsigned , default = 89

VD1_HDR2_GMUT_COEF2 0x3833

Bit(s)	R/W	Default	Description
31:16	R/W	23	reg_gmut_coef12 : // unsigned , default = 23
15:0	R/W	1883	reg_gmut_coef11 : // unsigned , default = 1883

VD1_HDR2_GMUT_COEF3 0x3834

Bit(s)	R/W	Default	Description
31:16	R/W	180	reg_gmut_coef21 : // unsigned , default = 180
15:0	R/W	34	reg_gmut_coef20 : // unsigned , default = 34

VD1_HDR2_GMUT_COEF4 0x3835

Bit(s)	R/W	Default	Description
15:0	R/W	1834	reg_gmut_coef22 : // unsigned , default = 1834

VD1_HDR2_PIPE_CTRL1 0x3836

Bit(s)	R/W	Default	Description
31:24	R/W	4	vblank_num_oetf : // unsigned , default = 4
23:16	R/W	4	hblank_num_oetf : // unsigned , default = 4
15:8	R/W	10	vblank_num_eotf : // unsigned , default = 10
7:0	R/W	10	hblank_num_eotf : // unsigned , default = 10

VD1_HDR2_PIPE_CTRL2 0x3837

Bit(s)	R/W	Default	Description
31:24	R/W	10	vblank_num_cgain : // unsigned , default = 10
23:16	R/W	10	hblank_num_cgain : // unsigned , default = 10
15:8	R/W	11	vblank_num_gmut : // unsigned , default = 11
7:0	R/W	11	hblank_num_gmut : // unsigned , default = 11

VD1_HDR2_PIPE_CTRL3 0x3838

Bit(s)	R/W	Default	Description
31:24	R/W	22	vblank_num_adps : // unsigned , default = 22
23:16	R/W	2	hblank_num_adps : // unsigned , default = 2
15:8	R/W	4	vblank_num_uv : // unsigned , default = 4
7:0	R/W	4	hblank_num_uv : // unsigned , default = 4

VD1_HDR2_PROC_WIN1 0x3839

Bit(s)	R/W	Default	Description
28:16	R/W	0	proc_win_h_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_h_st : // unsigned , default = 0

VD1_HDR2_PROC_WIN2 0x383A

Bit(s)	R/W	Default	Description
31	R/W	0	proc_win_gmut_en : // unsigned , default = 0
30	R/W	0	proc_win_adps_en : // unsigned , default = 0
29	R/W	0	proc_win_cgain_en : // unsigned , default = 0
28:16	R/W	0	proc_win_v_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_v_st : // unsigned , default = 0

VD1_HDR2_MATRIXI_EN_CTRL 0x383B

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

VD1_HDR2_MATRIXO_EN_CTRL 0x383C

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

VD2_HDR2_CTRL 0x3850

Bit(s)	R/W	Default	Description
20:18	R/W	0	reg_din_swap : // unsigned , default = 0
17	R/W	0	reg_out_fmt : // unsigned , default = 0
16	R/W	0	reg_only_mat : // unsigned , default = 0
13	R/W	0	reg_VDIN0_HDR2_top_en : // unsigned , default = 0
12	R/W	1	reg_cgain_mode : // unsigned , default = 1
7: 6	R/W	1	reg_gmut_mode : // unsigned , default = 1
5	R/W	0	reg_in_shift : // unsigned , default = 0
4	R/W	1	reg_in_fmt : // unsigned , default = 1
3	R/W	1	reg_eo_enable : // unsigned , default = 1
2	R/W	1	reg_oe_enable : // unsigned , default = 1
1	R/W	1	reg_ogain_enable : // unsigned , default = 1
0	R/W	1	reg_cgain_enable : // unsigned , default = 1

VD2_HDR2_CLK_GATE 0x3851

Bit(s)	R/W	Default	Description
31:30	R/W	0	clk_tm : gate clock ctrl (main clock) // unsigned , default = 0
29:28	R/W	0	output : matrix clock gate ctrl // unsigned , default = 0
25:24	R/W	0	input : matrix clock gate ctrl // unsigned , default = 0
23:22	R/W	0	hdr : top cbus clock gate ctrl // unsigned , default = 0

21:20	R/W	0	eotf : cbus clock gate ctrl // unsigned , default = 0
19:18	R/W	0	oetf : cbus clock gate ctrl // unsigned , default = 0
17:16	R/W	0	gamma : mult cbus clock gate ctrl // unsigned , default = 0
15:14	R/W	0	adaptive : cbus scaler clock gate ctrl // unsigned , default = 0
13:12	R/W	0	cgain : cbus clock gate ctrl // unsigned , default = 0
11:10	R/W	0	eotf : clock gate ctrl // unsigned , default = 0
9:8	R/W	0	oetf : clock gate ctrl // unsigned , default = 0
7:6	R/W	0	gamma : mult clock gate ctrl // unsigned , default = 0
5:4	R/W	0	adaptive : scaler clock gate ctrl // unsigned , default = 0
3:2	R/W	0	uv : gain clock gate ctrl // unsigned , default = 0
1:0	R/W	0	cgain : clock gate ctrl // unsigned , default = 0

VD2_HDR2_MATRIXI_COEF00_01 0x3852

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0
12:0	R/W	0	coef01 : // signed , default = 0

VD2_HDR2_MATRIXI_COEF02_10 0x3853

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

VD2_HDR2_MATRIXI_COEF11_12 0x3854

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

VD2_HDR2_MATRIXI_COEF20_21 0x3855

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

VD2_HDR2_MATRIXI_COEF22 0x3856

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

VD2_HDR2_MATRIXI_COEF30_31 0x3857

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

VD2_HDR2_MATRIXI_COEF32_40 0x3858

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

VD2_HDR2_MATRIXI_COEF41_42 0x3859

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

VD2_HDR2_MATRIXI_OFFSET0_1 0x385A

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

VD2_HDR2_MATRIXI_OFFSET2 0x385B

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

VD2_HDR2_MATRIXI_PRE_OFFSET0_1 0x385C

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

VD2_HDR2_MATRIXI_PRE_OFFSET2 0x385D

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

VD2_HDR2_MATRIXO_COEF00_01 0x385E

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0

12:0	R/W	0	coef01 : // signed , default = 0

VD2_HDR2_MATRIXO_COEF02_10 0x385F

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

VD2_HDR2_MATRIXO_COEF11_12 0x3860

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

VD2_HDR2_MATRIXO_COEF20_21 0x3861

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

VD2_HDR2_MATRIXO_COEF22 0x3862

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

VD2_HDR2_MATRIXO_COEF30_31 0x3863

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

VD2_HDR2_MATRIXO_COEF32_40 0x3864

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

VD2_HDR2_MATRIXO_COEF41_42 0x3865

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

VD2_HDR2_MATRIXO_OFFSET0_1 0x3866

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

VD2_HDR2_MATRIXO_OFFSET2 0x3867

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

VD2_HDR2_MATRIXO_PRE_OFFSET0_1 0x3868

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

VD2_HDR2_MATRIXO_PRE_OFFSET2 0x3869

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

VD2_HDR2_MATRIXI_CLIP 0x386A

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

VD2_HDR2_MATRIXO_CLIP 0x386B

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

VD2_HDR2_CGAIN_OFFT 0x386C

Bit(s)	R/W	Default	Description
26:16	R/W	0	reg_cgain_oft2 : // signed , default = 0
10:0	R/W	0	reg_cgain_oft1 : // signed , default = 0

VD2_EOTF_LUT_ADDR_PORT 0x386E

Bit(s)	R/W	Default	Description

7:0	R/W	0	eotf_lut_addr : // unsigned , default = 0

VD2_EOTF_LUT_DATA_PORT 0x386F

Bit(s)	R/W	Default	Description
19:0	R/W	0	eotf_lut_data : // unsigned , default = 0

VD2_OETF_LUT_ADDR_PORT 0x3870

Bit(s)	R/W	Default	Description
7:0	R/W	0	oetf_lut_addr : // unsigned , default = 0

VD2_OETF_LUT_DATA_PORT 0x3871

Bit(s)	R/W	Default	Description
11:0	R/W	0	oetf_lut_data : // unsigned , default = 0

VD2_CGAIN_LUT_ADDR_PORT 0x3872

Bit(s)	R/W	Default	Description
7:0	R/W	0	cgain_lut_addr : // unsigned , default = 0

VD2_CGAIN_LUT_DATA_PORT 0x3873

Bit(s)	R/W	Default	Description
11:0	R/W	0	cgain_lut_data : // unsigned , default = 0

VD2_HDR2_CGAIN_COEF0 0x3874

Bit(s)	R/W	Default	Description
27:16	R/W	0	reg_cgain_coef1 : // unsigned , default = 0
11:0	R/W	0	reg_cgain_coef0 : //unsigned , default = 0

VD2_HDR2_CGAIN_COEF1 0x3875

Bit(s)	R/W	Default	Description
11:0	R/W	0	reg_cgain_coef2 : // unsigned , default = 0

VD2_OGAIN_LUT_ADDR_PORT 0x3876

Bit(s)	R/W	Default	Description
7:0	R/W	0	ogain_lut_addr : // unsigned , default = 0

VD2_OGAIN_LUT_DATA_PORT 0x3877

Bit(s)	R/W	Default	Description
11:0	R/W	0	ogain_lut_data : // unsigned , default = 0

VD2_HDR2_ADPS_CTRL 0x3878

Bit(s)	R/W	Default	Description
6	R/W	1	reg_adpscl_bypass2 : // unsigned , default = 1
5	R/W	1	reg_adpscl_bypass1 : // unsigned , default = 1
4	R/W	1	reg_adpscl_bypass0 : // unsigned , default = 1
1:0	R/W	1	reg_adpscl_mode : // unsigned , default = 1

VD2_HDR2_ADPS_ALPHA0 0x3879

Bit(s)	R/W	Default	Description
29:16	R/W	0x1000	reg_adpscl_alpha1 : // unsigned , default = 0x1000
13:0	R/W	0x1000	reg_adpscl_alpha0 : // unsigned , default = 0x1000

VD2_HDR2_ADPS_ALPHA1 0x387A

Bit(s)	R/W	Default	Description
27:24	R/W	0xc	reg_adpscl_shift0 : // unsigned , default = 0xc
23:20	R/W	0xc	reg_adpscl_shift1 : // unsigned , default = 0xc
19:16	R/W	0xc	reg_adpscl_shift2 : // unsigned , default = 0xc
13:0	R/W	0x1000	reg_adpscl_alpha2 : // unsigned , default = 0x1000

VD2_HDR2_ADPS_BETA0 0x387B

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta0_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta0 : // unsigned , default = 0xfc000

VD2_HDR2_ADPS_BETA1 0x387C

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta1_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta1 : // unsigned , default = 0xfc000

VD2_HDR2_ADPS_BETA2 0x387D

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta2_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta2 : // unsigned , default = 0xfc000

VD2_HDR2_ADPS_COEF0 0x387E

Bit(s)	R/W	Default	Description
27:16	R/W	460	reg_adpscl_ys_coef1 : // unsigned , default = 460
11:0	R/W	1188	reg_adpscl_ys_coef0 : // unsigned , default = 1188

VD2_HDR2_ADPS_COEF1 0x387F

Bit(s)	R/W	Default	Description
11:0	R/W	104	reg_adpscl_ys_coef2 : // unsigned , default = 104

VD2_HDR2_GMUT_CTRL 0x3880

Bit(s)	R/W	Default	Description
3:0	R/W	14	reg_gmut_shift : // unsigned , default = 14

VD2_HDR2_GMUT_COEF0 0x3881

Bit(s)	R/W	Default	Description
31:16	R/W	674	reg_gmut_coef01 : // unsigned , default = 674
15:0	R/W	1285	reg_gmut_coef00 : // unsigned , default = 1285

VD2_HDR2_GMUT_COEF1 0x3882

Bit(s)	R/W	Default	Description
31:16	R/W	142	reg_gmut_coef10 : // unsigned , default = 142
15:0	R/W	89	reg_gmut_coef02 : // unsigned , default = 89

VD2_HDR2_GMUT_COEF2 0x3883

Bit(s)	R/W	Default	Description
31:16	R/W	23	reg_gmut_coef12 : // unsigned , default = 23
15:0	R/W	1883	reg_gmut_coef11 : // unsigned , default = 1883

VD2_HDR2_GMUT_COEF3 0x3884

Bit(s)	R/W	Default	Description
31:16	R/W	180	reg_gmut_coef21 : // unsigned , default = 180
15:0	R/W	34	reg_gmut_coef20 : // unsigned , default = 34

VD2_HDR2_GMUT_COEF4 0x3885

Bit(s)	R/W	Default	Description
15:0	R/W	1834	reg_gmut_coef22 : // unsigned , default = 1834

VD2_HDR2_PIPE_CTRL1 0x3886

Bit(s)	R/W	Default	Description
31:24	R/W	4	vblank_num_oetf : // unsigned , default = 4
23:16	R/W	4	hblank_num_oetf : // unsigned , default = 4
15:8	R/W	10	vblank_num_eotf : // unsigned , default = 10
7:0	R/W	10	hblank_num_eotf : // unsigned , default = 10

VD2_HDR2_PIPE_CTRL2 0x3887

Bit(s)	R/W	Default	Description
31:24	R/W	10	vblank_num_cgain : // unsigned , default = 10
23:16	R/W	10	hblank_num_cgain : // unsigned , default = 10
15:8	R/W	11	vblank_num_gmut : // unsigned , default = 11
7:0	R/W	11	hblank_num_gmut : // unsigned , default = 11

VD2_HDR2_PIPE_CTRL3 0x3888

Bit(s)	R/W	Default	Description
31:24	R/W	22	vblank_num_adps : // unsigned , default = 22
23:16	R/W	2	hblank_num_adps : // unsigned , default = 2
15:8	R/W	4	vblank_num_uv : // unsigned , default = 4
7:0	R/W	4	hblank_num_uv : // unsigned , default = 4

VD2_HDR2_PROC_WIN1 0x3889

Bit(s)	R/W	Default	Description
28:16	R/W	0	proc_win_h_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_h_st : // unsigned , default = 0

VD2_HDR2_PROC_WIN2 0x388A

Bit(s)	R/W	Default	Description
31	R/W	0	proc_win_gmut_en : // unsigned , default = 0
30	R/W	0	proc_win_adps_en : // unsigned , default = 0
29	R/W	0	proc_win_cgain_en : // unsigned , default = 0
28:16	R/W	0	proc_win_v_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_v_st : // unsigned , default = 0

VD2_HDR2_MATRIXI_EN_CTRL 0x388B

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

VD2_HDR2_MATRIXO_EN_CTRL 0x388C

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

OSD1_HDR2_CTRL 0x38A0

Bit(s)	R/W	Default	Description
20:18	R/W	0	reg_din_swap : // unsigned , default = 0
17	R/W	0	reg_out_fmt : // unsigned , default = 0
16	R/W	0	reg_only_mat : // unsigned , default = 0
13	R/W	0	reg_VDIN0_HDR2_top_en : // unsigned , default = 0
12	R/W	1	reg_cgain_mode : // unsigned , default = 1
7: 6	R/W	1	reg_gmut_mode : // unsigned , default = 1
5	R/W	0	reg_in_shift : // unsigned , default = 0
4	R/W	1	reg_in_fmt : // unsigned , default = 1
3	R/W	1	reg_eo_enable : // unsigned , default = 1
2	R/W	1	reg_oe_enable : // unsigned , default = 1
1	R/W	1	reg_ogain_enable : // unsigned , default = 1
0	R/W	1	reg_cgain_enable : // unsigned , default = 1

OSD1_HDR2_CLK_GATE 0x38A1

Bit(s)	R/W	Default	Description
31:30	R/W	0	clk_tm : gate clock ctrl (main clock) // unsigned , default = 0
29:28	R/W	0	output : matrix clock gate ctrl // unsigned , default = 0
25:24	R/W	0	input : matrix clock gate ctrl // unsigned , default = 0
23:22	R/W	0	hdr : top cbus clock gate ctrl // unsigned , default = 0

21:20	R/W	0	eotf : cbus clock gate ctrl // unsigned , default = 0
19:18	R/W	0	oetf : cbus clock gate ctrl // unsigned , default = 0
17:16	R/W	0	gamma : mult cbus clock gate ctrl // unsigned , default = 0
15:14	R/W	0	adaptive : cbus scaler clock gate ctrl // unsigned , default = 0
13:12	R/W	0	cgain : cbus clock gate ctrl // unsigned , default = 0
11:10	R/W	0	eotf : clock gate ctrl // unsigned , default = 0
9:8	R/W	0	oetf : clock gate ctrl // unsigned , default = 0
7:6	R/W	0	gamma : mult clock gate ctrl // unsigned , default = 0
5:4	R/W	0	adaptive : scaler clock gate ctrl // unsigned , default = 0
3:2	R/W	0	uv : gain clock gate ctrl // unsigned , default = 0
1:0	R/W	0	cgain : clock gate ctrl // unsigned , default = 0

OSD1_HDR2_MATRIXI_COEF00_01 0x38A2

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0
12:0	R/W	0	coef01 : // signed , default = 0

OSD1_HDR2_MATRIXI_COEF02_10 0x38A3

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

OSD1_HDR2_MATRIXI_COEF11_12 0x38A4

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

OSD1_HDR2_MATRIXI_COEF20_21 0x38A5

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

OSD1_HDR2_MATRIXI_COEF22 0x38A6

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

OSD1_HDR2_MATRIXI_COEF30_31 0x38A7

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

OSD1_HDR2_MATRIXI_COEF32_40 0x38A8

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

OSD1_HDR2_MATRIXI_COEF41_42 0x38A9

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

OSD1_HDR2_MATRIXI_OFFSET0_1 0x38AA

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

OSD1_HDR2_MATRIXI_OFFSET2 0x38AB

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

OSD1_HDR2_MATRIXI_PRE_OFFSET0_1 0x38AC

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

OSD1_HDR2_MATRIXI_PRE_OFFSET2 0x38AD

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

OSD1_HDR2_MATRIXO_COEF00_01 0x38AE

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef00 : // signed , default = 0

12:0	R/W	0	coef01 : // signed , default = 0

OSD1_HDR2_MATRIXO_COEF02_10 0x38AF

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef02 : // signed , default = 0
12:0	R/W	0	coef10 : // signed , default = 0

OSD1_HDR2_MATRIXO_COEF11_12 0x38B0

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef11 : // signed , default = 0
12:0	R/W	0	coef12 : // signed , default = 0

OSD1_HDR2_MATRIXO_COEF20_21 0x38B1

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef20 : // signed , default = 0
12:0	R/W	0	coef21 : // signed , default = 0

OSD1_HDR2_MATRIXO_COEF22 0x38B2

Bit(s)	R/W	Default	Description
12:0	R/W	0	coef22 : // signed , default = 0

OSD1_HDR2_MATRIXO_COEF30_31 0x38B3

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef13 : // signed , default = 0
12:0	R/W	0	coef14 : // signed , default = 0

OSD1_HDR2_MATRIXO_COEF32_40 0x38B4

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef23 : // signed , default = 0
12:0	R/W	0	coef24 : // signed , default = 0

OSD1_HDR2_MATRIXO_COEF41_42 0x38B5

Bit(s)	R/W	Default	Description
28:16	R/W	0	coef15 : // signed , default = 0
12:0	R/W	0	coef25 : // signed , default = 0

OSD1_HDR2_MATRIXO_OFFSET0_1 0x38B6

Bit(s)	R/W	Default	Description
26:16	R/W	0	offset0 : // signed , default = 0
10:0	R/W	0	offset1 : // signed , default = 0

OSD1_HDR2_MATRIXO_OFFSET2 0x38B7

Bit(s)	R/W	Default	Description
10:0	R/W	0	offset2 : // signed , default = 0

OSD1_HDR2_MATRIXO_PRE_OFFSET0_1 0x38B8

Bit(s)	R/W	Default	Description
26:16	R/W	0	pre_offset0 : // signed , default = 0
10:0	R/W	0	pre_offset1 : // signed , default = 0

OSD1_HDR2_MATRIXO_PRE_OFFSET2 0x38B9

Bit(s)	R/W	Default	Description
10:0	R/W	0	pre_offset2 : // signed , default = 0

OSD1_HDR2_MATRIXI_CLIP 0x38BA

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

OSD1_HDR2_MATRIXO_CLIP 0x38BB

Bit(s)	R/W	Default	Description
19:8	R/W	0	comp_th : // unsigned , default = 0
7:5	R/W	0	conv_rs : // unsigned , default = 0
4:3	R/W	0	clmod : // unsigned , default = 0

OSD1_HDR2_CGAIN_OFFT 0x38BC

Bit(s)	R/W	Default	Description
26:16	R/W	0	reg_cgain_oft2 : // signed , default = 0
10:0	R/W	0	reg_cgain_oft1 : // signed , default = 0

OSD1_EOTF_LUT_ADDR_PORT 0x389E

Bit(s)	R/W	Default	Description

7:0	R/W	0	eotf_lut_addr : // unsigned , default = 0

OSD1_EOTF_LUT_DATA_PORT 0x389F

Bit(s)	R/W	Default	Description
19:0	R/W	0	eotf_lut_data : // unsigned , default = 0

OSD1_OETF_LUT_ADDR_PORT 0x38A0

Bit(s)	R/W	Default	Description
7:0	R/W	0	oetf_lut_addr : // unsigned , default = 0

OSD1_OETF_LUT_DATA_PORT 0x38A1

Bit(s)	R/W	Default	Description
11:0	R/W	0	oetf_lut_data : // unsigned , default = 0

OSD1_CGAIN_LUT_ADDR_PORT 0x38A2

Bit(s)	R/W	Default	Description
7:0	R/W	0	cgain_lut_addr : // unsigned , default = 0

OSD1_CGAIN_LUT_DATA_PORT 0x38A3

Bit(s)	R/W	Default	Description
11:0	R/W	0	cgain_lut_data : // unsigned , default = 0

OSD1_HDR2_CGAIN_COEF0 0x38A4

Bit(s)	R/W	Default	Description
27:16	R/W	0	reg_cgain_coef1 : // unsigned , default = 0
11:0	R/W	0	reg_cgain_coef0 : //unsigned , default = 0

OSD1_HDR2_CGAIN_COEF1 0x38A5

Bit(s)	R/W	Default	Description
11:0	R/W	0	reg_cgain_coef2 : // unsigned , default = 0

OSD1_OGAIN_LUT_ADDR_PORT 0x38A6

Bit(s)	R/W	Default	Description
7:0	R/W	0	ogain_lut_addr : // unsigned , default = 0

OSD1_OGAIN_LUT_DATA_PORT 0x38A7

Bit(s)	R/W	Default	Description
11:0	R/W	0	ogain_lut_data : // unsigned , default = 0

OSD1_HDR2_ADPS_CTRL 0x38A8

Bit(s)	R/W	Default	Description
6	R/W	1	reg_adpscl_bypass2 : // unsigned , default = 1
5	R/W	1	reg_adpscl_bypass1 : // unsigned , default = 1
4	R/W	1	reg_adpscl_bypass0 : // unsigned , default = 1
1:0	R/W	1	reg_adpscl_mode : // unsigned , default = 1

OSD1_HDR2_ADPS_ALPHA0 0x38A9

Bit(s)	R/W	Default	Description
29:16	R/W	0x1000	reg_adpscl_alpha1 : // unsigned , default = 0x1000
13:0	R/W	0x1000	reg_adpscl_alpha0 : // unsigned , default = 0x1000

OSD1_HDR2_ADPS_ALPHA1 0x38AA

Bit(s)	R/W	Default	Description
27:24	R/W	0xc	reg_adpscl_shift0 : // unsigned , default = 0xc
23:20	R/W	0xc	reg_adpscl_shift1 : // unsigned , default = 0xc
19:16	R/W	0xc	reg_adpscl_shift2 : // unsigned , default = 0xc
13:0	R/W	0x1000	reg_adpscl_alpha2 : // unsigned , default = 0x1000

OSD1_HDR2_ADPS_BETA0 0x38AB

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta0_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta0 : // unsigned , default = 0xfc000

OSD1_HDR2_ADPS_BETA1 0x38AC

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta1_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta1 : // unsigned , default = 0xfc000

OSD1_HDR2_ADPS_BETA2 0x38AD

Bit(s)	R/W	Default	Description
20	R/W	0	reg_adpscl_beta2_s : // unsigned , default = 0
19:0	R/W	0xfc000	reg_adpscl_beta2 : // unsigned , default = 0xfc000

OSD1_HDR2_ADPS_COEF0 0x38AE

Bit(s)	R/W	Default	Description
27:16	R/W	460	reg_adpscl_ys_coef1 : // unsigned , default = 460
11:0	R/W	1188	reg_adpscl_ys_coef0 : // unsigned , default = 1188

OSD1_HDR2_ADPS_COEF1 0x38AF

Bit(s)	R/W	Default	Description
11:0	R/W	104	reg_adpscl_ys_coef2 : // unsigned , default = 104

OSD1_HDR2_GMUT_CTRL 0x38B0

Bit(s)	R/W	Default	Description
3:0	R/W	14	reg_gmut_shift : // unsigned , default = 14

OSD1_HDR2_GMUT_COEF0 0x38B1

Bit(s)	R/W	Default	Description
31:16	R/W	674	reg_gmut_coef01 : // unsigned , default = 674
15:0	R/W	1285	reg_gmut_coef00 : // unsigned , default = 1285

OSD1_HDR2_GMUT_COEF1 0x38B2

Bit(s)	R/W	Default	Description
31:16	R/W	142	reg_gmut_coef10 : // unsigned , default = 142
15:0	R/W	89	reg_gmut_coef02 : // unsigned , default = 89

OSD1_HDR2_GMUT_COEF2 0x38B3

Bit(s)	R/W	Default	Description
31:16	R/W	23	reg_gmut_coef12 : // unsigned , default = 23
15:0	R/W	1883	reg_gmut_coef11 : // unsigned , default = 1883

OSD1_HDR2_GMUT_COEF3 0x38B4

Bit(s)	R/W	Default	Description
31:16	R/W	180	reg_gmut_coef21 : // unsigned , default = 180
15:0	R/W	34	reg_gmut_coef20 : // unsigned , default = 34

OSD1_HDR2_GMUT_COEF4 0x38B5

Bit(s)	R/W	Default	Description
15:0	R/W	1834	reg_gmut_coef22 : // unsigned , default = 1834

OSD1_HDR2_PIPE_CTRL1 0x38B6

Bit(s)	R/W	Default	Description
31:24	R/W	4	vblank_num_oetf : // unsigned , default = 4
23:16	R/W	4	hblank_num_oetf : // unsigned , default = 4
15:8	R/W	10	vblank_num_eotf : // unsigned , default = 10
7:0	R/W	10	hblank_num_eotf : // unsigned , default = 10

OSD1_HDR2_PIPE_CTRL2 0x38B7

Bit(s)	R/W	Default	Description
31:24	R/W	10	vblank_num_cgain : // unsigned , default = 10
23:16	R/W	10	hblank_num_cgain : // unsigned , default = 10
15:8	R/W	11	vblank_num_gmut : // unsigned , default = 11
7:0	R/W	11	hblank_num_gmut : // unsigned , default = 11

OSD1_HDR2_PIPE_CTRL3 0x38B8

Bit(s)	R/W	Default	Description
31:24	R/W	22	vblank_num_adps : // unsigned , default = 22
23:16	R/W	2	hblank_num_adps : // unsigned , default = 2
15:8	R/W	4	vblank_num_uv : // unsigned , default = 4
7:0	R/W	4	hblank_num_uv : // unsigned , default = 4

OSD1_HDR2_PROC_WIN1 0x38B9

Bit(s)	R/W	Default	Description
28:16	R/W	0	proc_win_h_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_h_st : // unsigned , default = 0

OSD1_HDR2_PROC_WIN2 0x38BA

Bit(s)	R/W	Default	Description
31	R/W	0	proc_win_gmut_en : // unsigned , default = 0
30	R/W	0	proc_win_adps_en : // unsigned , default = 0
29	R/W	0	proc_win_cgain_en : // unsigned , default = 0
28:16	R/W	0	proc_win_v_ed : // unsigned , default = 0
12:0	R/W	0	proc_win_v_st : // unsigned , default = 0

OSD1_HDR2_MATRIXI_EN_CTRL 0x38BB

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

OSD1_HDR2_MATRIXO_EN_CTRL 0x38BC

Bit(s)	R/W	Default	Description
5:4	R/W	0	matrix_gclk_ctrl : // unsigned , default = 0
1	R/W	0	enable_sync_sel : // unsigned , default = 0
0	R/W	0	matrix_conv_en : // unsigned , default = 0

10.2.3.42 VDIN

VDIN0_SCALE_COEF_IDX 0x1200

VDIN0_SCALE_COEF 0x1201

VDIN0_COM_CTRL0 0x1202

Bit(s)	R/W	Default	Description
31	R/W	0	mpeg_to_vdin_sel, 0: mpeg source to NR directly, 1: mpeg source pass through here
30	R/W	0	mpeg_field info which can be written by software
29	R/W	0	force go_field, pulse signal
28	R/W	0	force go_line, pulse signal
27	R/W	0	enable mpeg_go_field input signal
26-20	R/W	0	hold lines
19	R/W	0	delay go_field function enable
18-12	R/W	0	delay go_field line number
11-10	R/W	0	component2 output switch, 00: select component0 in, 01: select component1 in, 10: select component2 in
9-8	R/W	0	component1 output switch, 00: select component0 in, 01: select component1 in, 10: select component2 in
7-6	R/W	0	component0 output switch, 00: select component0 in, 01: select component1 in, 10: select component2 in
5	R/W	0	input window selection function enable
4	R/W	0	enable VDIN common data input, otherwise there will be no video data input

Bit(s)	R/W	Default	Description
3-0	R/W	0	vdin selection, 1: mpeg_in from dram; 2: bt656 input; 3: Reserved (component input); 4: Reserved(tvdecoder input); 5: Reserved(hdmi rx input); 6: digtial video input; 7: internal loop back from Viu; 8: Reserved(mipi_csi2) ;9: Reserved(isp); 10: second bt656 input; otherwise no input.

VDIN0_ACTIVE_MAX_PIX_CNT_STATUS 0x1203

Bit(s)	R/W	Default	Description
28-16	R	0	active_max_pix_cnt, readonly
12-0	R	0	active_max_pix_cnt_shadow, readonly

VDIN0_LCNT_STATUS 0x1204

Bit(s)	R/W	Default	Description
28-16	R	0	go_line_cnt, readonly
12-0	R	0	active_line_cnt, readonly

VDIN0_COM_STATUS0 0x1205

Bit(s)	R/W	Default	Description
17	R	0	Vid_wr_pending_ddr_wrrsp
16	R	0	Curr_pic_sec
15	R	0	Curr_pic_sec_sav
14-3	R	0	lfifo_buf_cnt
2	R	0	vdin_direct_done status
1	R	0	vdin_nr_done status
0	R	0	field

VDIN0_COM_STATUS1 0x1206

Bit(s)	R/W	Default	Description
31	R	0	vdi4 fifo overflow
29-24	R	0	vdi3_asfifo_cnt
23	R	0	vdi3 fifo overflow
21-16	R	0	vdi3_asfifo_cnt
15	R	0	vdi2 fifo overflow
13-8	R	0	vdi2_asfifo_cnt
7	R	0	vdi1 fifo overflow

Bit(s)	R/W	Default	Description
5-0	R	0	vdi1_asfifo_cnt

VDIN0_LCNT_SHADOW_STATUS 0x1207

Bit(s)	R/W	Default	Description
28-16	R	0	go_line_cnt_shadow, readonly
12-0	R	0	active_line_cnt_shadow, readonly

VDIN0_ASFIFO_CTRL0 0x1208

Bit(s)	R/W	Default	Description
23	R/W	0	vdi2 DE enable
22	R/W	0	vdi2 go field enable
21	R/W	0	vdi2 go line enable
20	R/W	0	vdi2 if true, negative active input vsync
19	R/W	0	vdi2 if true, negative active input hsync
18	R/W	0	vdi2 vsync soft reset fifo enable
17	R/W	0	vdi2 overflow status clear
16	R/W		vdi2 asfifo soft reset, level signal
7	R/W	0	Vdi1 DE enable
6	R/W	0	Vdi1 go field enable
5	R/W	0	Vdi1 go line enable
4	R/W	0	Vdi1 if true, negative active input vsync
3	R/W	0	Vdi1 if true, negative active input hsync
2	R/W	0	Vdi1 vsync soft reset fifo enable
1	R/W	0	Vdi1 overflow status clear
0	R/W	0	Vdi1 asfifo soft reset, level signal

VDIN0_ASFIFO_CTRL1 0x1209

Bit(s)	R/W	Default	Description
23	R/W	0	Vdi4 DE enable
22	R/W	0	Vdi4 go field enable
21	R/W	0	Vdi4 go line enable
20	R/W	0	Vdi4 if true, negative active input vsync

19	R/W	0	Vdi4 if true, negative active input hsync
18	R/W	0	Vdi4 vsync soft reset fifo enable
17	R/W	0	Vdi4 overflow status clear
16	R/W	0	Vdi4 asfifo soft reset, level signal
7	R/W	0	Vdi3 DE enable
6	R/W	0	Vdi3 go field enable
5	R/W	0	Vdi3 go line enable
4	R/W	0	Vdi3 if true, negative active input vsync
3	R/W	0	Vdi3 if true, negative active input hsync
2	R/W	0	Vdi3 vsync soft reset fifo enable
1	R/W	0	Vdi3 overflow status clear
0	R/W	0	Vdi3 asfifo soft reset, level signal

VDIN0_WIDTHM1I_WIDTHM1O 0x120a

Bit(s)	R/W	Default	Description
28-16	R/W	0	input width minus 1, after the window function
12-0	R/W	0	output width minus 1

VDIN0_SC_MISC_CTRL 0x120b

Bit(s)	R/W	Default	Description
14-8	R/W	0	hsc_ini_pixi_ptr, signed data, only useful when short_lineo_en is true
7	R/W	0	prehsc_en
6	R/W	0	hsc_en
5	R/W	0	hsc_short_lineo_en, short line output enable
4	R/W	0	hsc_nearest_en
3	R/W	0	Hsc_phase0_always_en
2-0	R/W	0	hsc_bank_length

VDIN0_HSC_PHASE_STEP 0x120c

Bit(s)	R/W	Default	Description
28-24	R/W	0	integer portion
23-0	R/W	0	fraction portion

VDIN0_HSC_INI_CTRL 0x120d

Bit(s)	R/W	Default	Description
30-29	R/W	0	hscale rpt_p0_num
28-24	R/W	0	hscale ini_rcv_num
23-0	R/W	0	hscale ini_phase

VDIN0_COM_STATUS2 0x120e

Bit(s)	R/W	Default	Description
23	R	0	Vdi7 fifo overflow
21-16	R	0	Vdi7_asfifo_cnt
15	R	0	Vdi6 fifo overflow
13-8	R	0	Vdi6_asfifo_cnt
7	R	0	vdi5 fifo overflow
5-0	R	0	vdi5_asfifo_cnt

VDIN0_ASFIFO_CTRL2 0x120f

Bit(s)	R/W	Default	Description
25	R/W	0	if true, decimation counter sync with first valid DE in the field, //otherwise the decimation counter is not sync with external signal
24	R/W	0	decimation de enable
23-20	R/W	0	decimation phase, which counter value use to decimate,
19-16	R/W	0	decimation number, 0: not decimation, 1: decimation 2, 2: decimation 3 ....
7	R/W	0	Vdi5 DE enable
6	R/W	0	Vdi5 go field enable
5	R/W	0	Vdi5 go line enable
4	R/W	0	Vdi5 if true, negative active input vsync
3	R/W	0	Vdi5 if true, negative active input hsync
2	R/W	0	Vdi5 vsync soft reset fifo enable
1	R/W	0	Vdi5 overflow status clear
0	R/W	0	Vdi5 asfifo soft reset, level signal

VDIN0_MATRIX_CTRL 0x1210

Bit(s)	R/W	Default	Description

7	R/W	0	highlight_enable
6	R/W	0	probe_post, if true, probe pixel data after matrix, otherwise probe pixel data before matrix
5-4	R/W	0	probe_sel, 00: select matrix 0, 01: select matrix 1, otherwise select nothing
3-2	R/W	0	matrix coef idx selection, 00: select mat0, 01: select mat1, otherwise slect nothing
1	R/W	0	mat1 conversion matrix enable
0	R/W	0	Mat0 conversion matrix enable

VDIN0_MATRIX_COEF00_01 0x1211

Bit(s)	R/W	Default	Description
28-16	R/W	0	coef00
12-0	R/W	0	coef01

VDIN0_MATRIX_COEF02_10 0x1212

Bit(s)	R/W	Default	Description
28-16	R/W	0	coef02
12-0	R/W	0	Coef10

VDIN0_MATRIX_COEF11_12 0x1213

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coef11
12-0	R/W	0	Coef12

VDIN0_MATRIX_COEF20_21 0x1214

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coef20
12-0	R/W	0	coef21

VDIN0_MATRIX_COEF22 0x1215

Bit(s)	R/W	Default	Description
18-16	R/W	0	convrs
7-0	R/W	0	Coef22

VDIN0_MATRIX_OFFSET0_1 0x1216

Bit(s)	R/W	Default	Description
26-16	R/W	0	offset0

10-0	R/W	0	Offset1

VDIN0_MATRIX_OFFSET2 0x1217

Bit(s)	R/W	Default	Description
10-0	R/W	0	Offset2

VDIN0_MATRIX_PRE_OFFSET0_1 0x1218

Bit(s)	R/W	Default	Description
26-16	R/W	0	Pre_offset0
10-0	R/W	0	Pre_Offset1

VDIN0_MATRIX_PRE_OFFSET2 0x1219

Bit(s)	R/W	Default	Description
10-0	R/W	0	Pre_Offset2

VDIN0_LFIFO_CTRL 0x121a

Bit(s)	R/W	Default	Description
11-0	R/W	0	lfifo_buf_size

VDIN0_COM_GCLK_CTRL 0x121b

Bit(s)	R/W	Default	Description
15-14	R/W	0	Gate clock control for blackbar detector
13-12	R/W	0	Gate clock control for hist
11-10	R/W	0	Gate clock control for line fifo
9-8	R/W	0	Gate clock control for matrix
7-6	R/W	0	Gate clock control for horizontal scaler
5-4	R/W	0	Gate clock control for pre scaler
3-2	R/W	0	Gate clock control for vdin_com_proc
1-0	R/W	0	Gate clock control for the vdin reg

VDIN0_INTF_WIDTHM1 0x121c

Bit(s)	R/W	Default	Description
26	R/W	0	VDIN write mif bvalid_sel: 1. Bvalid_singnal from bus, 0: bytes_wr handshakes
25	R/W	0	VDIN write mif burst last sel: 1. All kind of burst last signal include ext_data_last. 0. Used the normal burst last signal
12-0	R/W	0	VDIN input interface width minus 1, before the window function, after the de decimation

VDIN0_WR_CTRL2 0x121f

Bit(s)	R/W	Default	Description
19	R/W	0	Vdin0 wr bit10 mode
18	R/W	0	Data_ext_en 1: send out data if req was interrupt by soft reset 0 : normal mode
17:16	R/W	1	Words lim[1:0] : it would not send out request before Words_lim *16 words were ready
15:12	R/W	1	Burst_ lim : 00 , 1 word in 1burst , 01 ,2 words in 1burst, 10, 4 words in 1burst , 11 reserved
10:9	R/W	0	Words_lim[3:2]
8	R/W	0	1: discard data before line fifo, 0: normal mode
7-0	R/W	0	Write chroma canvas address, for NV12/21 mode.

VDIN0_WR_CTRL 0x1220

Bit(s)	R/W	Default	Description
31:30	R/W	0	vdin0_wr_mif_hconv_mode. Applicable only to vdin_write_format=0 or 2. 0=Output every even pixel's CbCr; 1=Output every odd pixel's CbCr; 2=Output an average value per even&odd pair of pixels; 3=Output all CbCr. Only applys to vdin_write_format =2.
29	R/W	0	vdin0_wr_mif_no_clk_gate. If true, enable free-run clock.
28	R/W	0	clear write response counter in the vdin write memory interface
27	R/W	1	eol_sel, 1: use eol as the line end indication, 0: use width as line end indication in the vdin write memory interface
26	R/W	0	vcp_nr_en. Only used in VDIN0. NOT used in VDIN1
25	R/W	1	vcp_wr_en Only used in VDIN0. NOT used in VDIN1
24	R/W	1	vcp_in_en Only used in VDIN0. NOT used in VDIN1
23	R/W	1	vdin frame reset enble, if true, it will provide frame reset during go_field(vsync) to the modules after that
22	R/W	1	vdin line fifo soft reset enable, meaning, if true line fifo will reset during go_field (vsync)
21	R/W	0	vdin direct write done status clear bit
20	R/W	0	vdin NR write done status clear bit
19	R/W	0	Vdin0_wr words swap : swap the 2 64bits word in 128 words
18	R/W	0	vdin0_wr_mif_swap_cbcr. Applicable only to vdin_write_format =2. 0=Output CbCr (NV12); 1=Output CrCb (NV21);

Bit(s)	R/W	Default	Description
17:16	R/W	0	vdin0_wr_mif_vconv_mode. Applicable only to vdin_write_format=2. 0=Output every even line's CbCr; 1=Output every odd line's CbCr; 2=Reserved; 3=Output all CbCr.
13-12	R/W	0	vdin_write_format, 0: 4:2:2 to one canvas; 1: 4:4:4 to one canvas; 2: Y to luma canvas, CbCr to chroma canvas, for NV12/21; 3: yuv422 full pack mode
11	R/W	0	vdin write canvas double buffer enable, means the canvas address will be latched by vsync before using
9	R/W	0	vdin write request urgent
8	R/W	0	vdin write request enable
7-0	R/W	0	Write canvas address (For NV12/21 mode, it's LUMA canvas)

VDIN0_WR_H_START_END 0x1221

Bit(s)	R/W	Default	Description
29	R/W	0	if true, horizontal reverse
28-16	R/W	0	start
12-0	R/W	0	end

VDIN0_WR_V_START_END 0x1222

Bit(s)	R/W	Default	Description
29	R/W	0	if true, vertical reverse
28-16	R/W	0	start
12-0	R/W	0	end

VDIN0_VSC_PHASE_STEP 0x1223

Bit(s)	R/W	Default	Description
24-16	R/W	0	integer portion
19-0	R/W	0	fraction portion

VDIN0_VSC_INI_CTRL 0x1224

Bit(s)	R/W	Default	Description
23	R/W	0	vsc_en, vertical scaler enable
21	R/W	0	vsc_phase0_always_en, when scale up, you have to set it to 1
20-16	R/W	0	ini skip_line_num

15-0	R/W	0	vscaler ini_phase

VDIN0_SCIN_HEIGHTM1 0x1225

Bit(s)	R/W	Default	Description
12-0	R/W	0x437	scaler input height minus 1

VDIN0_DUMMY_DATA 0x1226

Bit(s)	R/W	Default	Description
23-16	R/W	0	dummy component 0
15-8	R/W	0x80	dummy component 1
7-0	R/W	0x80	dummy component 2

VDIN0_MATRIX_PROBE_COLOR 0x1228

Bit(s)	R/W	Default	Description
29-20	R/W	0	component 0
19-10	R/W	0	omponent 1
9-0	R/W	0	component 2

VDIN0_MATRIX_HL_COLOR 0x1229

Bit(s)	R/W	Default	Description
23-16	R/W	0	component 0
15-8	R/W	0	component 1
7-0	R/W	0	component 2

VDIN0_MATRIX_PROBE_POS 0x122a

Bit(s)	R/W	Default	Description
28-16	R/W	0	probe x, position
12-0	R/W	0	probe y, postion

VDIN0_HIST_CTRL 0x1230

Bit(s)	R/W	Default	Description
31-24	R/W	0	No use
23-16	R/W	0	No use
11	R/W	0	Hist 34bin only mode
10-9	R/W	0	ldim_stts_din_sel, 00: from matrix0 dout, 01: from vsc_dout, 10: from matrix1 dout, 11: form matrix1 din

Bit(s)	R/W	Default	Description
8	R/W	0	ldim_stts_en
6-5	R/W	0	hist_dnlp_low the real pixels in each bins got by VDIN_DNLP_HISTXX should multiple with 2^(dnlp_low+3)
3-2	R/W	0	hist_din_sel the source used for hist statistics. 2'b00: from MAT0_dout; 2'b01: from vsc_dout; 2'b10: from mat1_dout, 3: mat1_din
1	R/W	0	hist_win_en 1'b0: hist used for full picture; 1'b1: hist used for pixels within hist window
0	R/W	0	hist_spl_en 1'b0: disable hist readback; 1'b1: enable hist readback

VDIN0_HIST_H_START_END 0x1231

Bit(s)	R/W	Default	Description
28-16	R/W	0	hist_hstart horizontal start value to define hist window
12-0	R/W	0	hist_hend horizontal end value to define hist window

VDIN0_HIST_V_START_END 0x1232

Bit(s)	R/W	Default	Description
28-16	R/W	0	hist_vstart vertical start value to define hist window
12-0	R/W	0	hist_vend vertical end value to define hist window

VDIN0_HIST_MAX_MIN 0x1233

Bit(s)	R/W	Default	Description
15-8	R	0	hist_max maximum value
7-0	R	0	hist_min minimum value

VDIN0_HIST_SPL_VAL 0x1234

Bit(s)	R/W	Default	Description
31-0	R	0	hist_spl_rd , counts for the total luma value

VDIN0_HIST_SPL_PIX_CNT 0x1235

Bit(s)	R/W	Default	Description
21-0	R	0	hist_spl_pixel_count, counts for the total calculated pixels

VDIN0_HIST_CHROMA_SUM 0x1236

Bit(s)	R/W	Default	Description
31-0	R	0	hist_chroma_sum , counts for the total chroma value

VDIN0_DNLP_HIST00 0x1237

//0-255 are splited to 64 bins evenly, and VDIN_DNLP_HISTXX

//are the statistic number of pixels that within each bin.

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 2nd bin
15-0	R	0	counts for the 1st bin

VDIN0_DNLP_HIST01 0x1238

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 4th bin
15-0	R	0	counts for the 3rd bin

VDIN0_DNLP_HIST02 0x1239

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 6th bin
15-0	R	0	counts for the 5th bin

VDIN0_DNLP_HIST03 0x123a

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 8th bin
15-0	R	0	counts for the 7th bin

VDIN0_DNLP_HIST04 0x123b

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 10th bin
15-0	R	0	counts for the 9th bin

VDIN0_DNLP_HIST05 0x123c

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 12th bin
15-0	R	0	counts for the 11th bin

VDIN0_DNLP_HIST06 0x123d

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 14th bin
15-0	R	0	counts for the 13th bin

VDIN0_DNLP_HIST07 0x123e

Bit(s)	R/W	Default	Description

31-16	R	0	counts for the 16th bin
15-0	R	0	counts for the 15th bin

VDIN0_DNLP_HIST08 0x123f

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 18th bin
15-0	R	0	counts for the 17th bin

VDIN0_DNLP_HIST09 0x1240

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 20th bin
15-0	R	0	counts for the 19th bin

VDIN0_DNLP_HIST10 0x1241

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 22nd bin
15-0	R	0	counts for the 21st bin

VDIN0_DNLP_HIST11 0x1242

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 24th bin
15-0	R	0	counts for the 23rd bin

VDIN0_DNLP_HIST12 0x1243

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 26th bin
15-0	R	0	counts for the 25th bin

VDIN0_DNLP_HIST13 0x1244

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 28th bin
15-0	R	0	counts for the 27th bin

VDIN0_DNLP_HIST14 0x1245

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 30th bin

15-0	R	0	counts for the 29th bin

VDIN0_DNLP_HIST15 0x1246

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 32nd bin
15-0	R	0	counts for the 31st bin

VDIN0_DNLP_HIST16 0x1247

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 34th bin
15-0	R	0	counts for the 33rd bin

VDIN0_DNLP_HIST17 0x1248

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 36th bin
15-0	R	0	counts for the 35th bin

VDIN0_DNLP_HIST18 0x1249

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 38th bin
15-0	R	0	counts for the 37th bin

VDIN0_DNLP_HIST19 0x124a

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 40th bin
15-0	R	0	counts for the 39th bin

VDIN0_DNLP_HIST20 0x124b

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 42nd bin
15-0	R	0	counts for the 41st bin

VDIN0_DNLP_HIST21 0x124c

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 44th bin
15-0	R	0	counts for the 43rd bin

VDIN0_DNLP_HIST22 0x124d

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 46th bin
15-0	R	0	counts for the 45th bin

VDIN0_DNLP_HIST23 0x124e

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 48th bin
15-0	R	0	counts for the 47th bin

VDIN0_DNLP_HIST24 0x124f

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 50th bin
15-0	R	0	counts for the 49th bin

VDIN0_DNLP_HIST25 0x1250

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 52nd bin
15-0	R	0	counts for the 51st bin

VDIN0_DNLP_HIST26 0x1251

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 54th bin
15-0	R	0	counts for the 53rd bin

VDIN0_DNLP_HIST27 0x1252

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 56th bin
15-0	R	0	counts for the 55th bin

VDIN0_DNLP_HIST28 0x1253

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 58th bin
15-0	R	0	counts for the 57th bin

VDIN0_DNLP_HIST29 0x1254

Bit(s)	R/W	Default	Description

31-16	R	0	counts for the 60th bin
15-0	R	0	counts for the 59th bin

VDIN0_DNLP_HIST30 0x1255

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 62nd bin
15-0	R	0	counts for the 61st bin

VDIN0_DNLP_HIST31 0x1256

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 64th bin
15-0	R	0	counts for the 63rd bin

VDIN0_LDIM_STTS_HIST_REGION_IDX 0x1257

Bit(s)	R/W	Default	Description
31	R	0	local dimming max statistic enable
28	R	0	eol enable
27-25	R	0	vertical line overlap number for max finding
24-22	R	0	horizontal pixel overlap number, 0: 17 pix, 1: 9 pix, 2: 5 pix, 3: 3 pix, 4: 0 pix
20	R	0	1,2,1 low pass filter enable before max/hist statistic
19-16	R	0	region H/V position index, refer to VDIN_LDIM_STTS_HIST_SET_REGION
15	R	0	1: region read index auto increase per read to VDIN_LDIM_STTS_HIST_READ_REGION
6-0	R	0	region read index

VDIN0_LDIM_STTS_HIST_SET_REGION 0x1258

Bit(s)	R/W	Default	Description

28:16	R	0	if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h0: read/write hvstart0
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h1: read/write hend01
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h2: read/write vend01
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h3: read/write hend23
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h4: read/write vend23
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h5: read/write hend45
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h6: read/write vend45
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'd7: read/write hend67
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h8: read/write vend67
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h9: read/write hend89
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'ha: read/write vend89
			//hvstart0, Bit 28:16 row0 vstart, Bit 12:0 col0 hstart
			//hend01, Bit 28:16 col1 hend, Bit 12:0 col0 hend
			//vend01, Bit 28:16 row1 vend, Bit 12:0 row0 vend
			//hend23, Bit 28:16 col3 hend, Bit 12:0 col2 hend
			//vend23, Bit 28:16 row3 vend, Bit 12:0 row2 vend
			//hend45, Bit 28:16 col5 hend, Bit 12:0 col4 hend
			//vend45, Bit 28:16 row5 vend, Bit 12:0 row4 vend
			//hend67, Bit 28:16 col7 hend, Bit 12:0 col6 hend
			//vend67, Bit 28:16 row7 vend, Bit 12:0 row6 vend
			//hend89, Bit 28:16 col9 hend, Bit 12:0 col8 hend
			//vend89, Bit 28:16 row9 vend, Bit 12:0 row8 vend

12:0	R	0	

VDIN0_LDIM_STTS_HIST_READ_REGION 0x1259

Bit(s)	R/W	Default	Description
29:20	R	0	Max_comp2
19:10	R	0	Max_comp1
9:0	R	0	Max_comp0

VDIN0_MEAS_CTRL0 0x125a

Bit(s)	R/W	Default	Description
18	R/W	0	reset bit, high active
17	R/W	0	if true, widen hs/vs pulse
16	R/W	0	vsync total counter always accumulating enable
14-12	R/W	0	select hs/vs of video input channel to measure, 0: no selection, 1 vdi1, 2: vid2: 3: vid3, 4:vid4, 5:vdi5, 6:vdi6, 7:vdi7.
11-4	R/W	0	vsync_span, define how many vsync span need to measure

2-0	R/W	0	meas_hs_index, index to select which HS counter/range

VDIN0_MEAS_VS_COUNT_HI 0x125b

Bit(s)	R/W	Default	Description
19-16	R	0	meas_ind_total_count_n, every number of sync_span vsyncs, this count add 1
15-0	R	0	high bit portion of vsync total counter

VDIN0_MEAS_VS_COUNT_LO 0x125c

Bit(s)	R/W	Default	Description
31-0	R	0	low bit portion of vsync total counter

VDIN0_MEAS_HS_RANGE 0x125d

//according to the meas_hs_index in register VDIN_MEAS_CTRL0

//meas_hs_index == 0, first hs range

//meas_hs_index == 1, second hs range

//meas_hs_index == 2, third hs range

//meas_hs_index == 3, fourth hs range

Bit(s)	R/W	Default	Description
28-16	R	0	count_start
12-0	R	0	count_end

VDIN0_MEAS_HS_COUNT 0x125e

//according to the meas_hs_index in register VDIN_MEAS_CTRL0,

//meas_hs_index == 0, first range hs counter,

//meas_hs_index == 1, second range hs

//meas_hs_index == 2, third range hs

//meas_hs_index == 3, fourth range hs

Bit(s)	R/W	Default	Description
23-0	R	0	Hs counter

VDIN0_BLKBAR_CTRL1 0x125f

Bit(s)	R/W	Default	Description
8	R/W	0	white_enable
7-0	R/W	0	blkbar_white_level

VDIN0_BLKBAR_CTRL0 0x1260

Bit(s)	R/W	Default	Description
31-24	R/W	0	blkbar_black_level threshold to judge a black point

20-8	R/W	0	blkbar_hwidth left and right region width
7-5	R/W	0	blkbar_comp_sel select yin or uin or vin to be the valid input
4	R/W	0	blkbar_sw_statistic_en enable software statistic of each block black points number
3	R/W	0	blkbar_det_en
2-1	R/W	0	blkbar_din_sel, 0:mat0_dout, 1:vsc_dout, 2:mat1_dout, 3:mat1_din
0	R/W	0	Blkbar_det_top_en

VDIN0_BLKBAR_H_START_END 0x1261

Bit(s)	R/W	Default	Description
28-16	R/W	0	blkbar_hstart. Left region start
12-0	R/W	0	blkbar_hend. Right region end

VDIN0_BLKBAR_V_START_END 0x1262

Bit(s)	R/W	Default	Description
28-16	R/W	0	blkbar_vstart.
12-0	R/W	0	blkbar_vend.

VDIN0_BLKBAR_CNT_THRESHOLD 0x1263

Bit(s)	R/W	Default	Description
19-0	R/W	0	blkbar_cnt_threshold. threshold to judge whether a block is totally black

VDIN0_BLKBAR_ROW_TH1_TH2 0x1264

Bit(s)	R/W	Default	Description
28-16	R/W	0	blkbar_row_th1. //threshold of the top blackbar
12-0	R/W	0	blkbar_row_th2 //threshold of the bottom blackbar

VDIN0_BLKBAR_IND_LEFT_START_END 0x1265

Bit(s)	R/W	Default	Description
28-16	R	0	blkbar_ind_left_start. horizontal start of the left region in the current searching
12-0	R	0	blkbar_ind_left_end. horizontal end of the left region in the current searching

VDIN0_BLKBAR_IND_RIGHT_START_END 0x1266

Bit(s)	R/W	Default	Description
28-16	R	0	blkbar_ind_right_start.horizontal start of the right region in the current searching
12-0	R	0	blkbar_ind_right_end. horizontal end of the right region in the current searching

VDIN0_BLKBAR_IND_LEFT1_CNT 0x1267

Bit(s)	R/W	Default	Description
19-0	R	0	blkbar_ind_left1_cnt. Black pixel counter. left part of the left region

VDIN0_BLKBAR_IND_LEFT2_CNT 0x1268

Bit(s)	R/W	Default	Description
19-0	R	0	blkbar_ind_left2_cnt. Black pixel counter. right part of the left region

VDIN0_BLKBAR_IND_RIGHT1_CNT 0x1269

Bit(s)	R/W	Default	Description
19-0	R	0	blkbar_ind_right1_cnt. Black pixel counter. left part of the right region

VDIN0_BLKBAR_IND_RIGHT2_CNT 0x126a

Bit(s)	R/W	Default	Description
19-0	R	0	blkbar_ind_right2_cnt. Black pixel counter. right part of the right region

VDIN0_BLKBAR_STATUS0 0x126b

Bit(s)	R/W	Default	Description
29	R	0	blkbar_ind_black_det_done. LEFT/RIGHT Black detection done
28-16	R	0	blkbar_top_pos. Top black bar position
12-0	R	0	blkbar_bot_pos. Bottom black bar position

VDIN0_BLKBAR_STATUS1 0x126c

Bit(s)	R/W	Default	Description
28-16	R	0	blkbar_left_pos. Left black bar posiont
12-0	R	0	blkbar_right_pos. Right black bar position

VDIN0_WIN_H_START_END ？？？ 0x126d

Bit(s)	R/W	Default	Description
28-16	R/W	0	input window H start
12-0	R/W	0	input window H end

VDIN0_WIN_V_START_END ？？？ 0x126e

Bit(s)	R/W	Default	Description
28-16	R/W	0	input window V start

12-0	R/W	0	input window V end

VDIN0_ASFIFO_CTRL3 0x126f

Bit(s)	R/W	Default	Description
31	R/W	0	Vdi9 DE enable
30	R/W	0	Vdi9 go field enable
29	R/W	0	Vdi9 go line enable
28	R/W	0	Vdi9 if true, negative active input vsync
27	R/W	0	Vdi9 if true, negative active input hsync
26	R/W	0	Vdi9 vsync soft reset fifo enable
25	R/W	0	Vdi9 overflow status clear
24	R/W	0	Vdi9 asfifo soft reset, level signal
15	R/W	0	Vdi7 DE enable
14	R/W	0	Vdi7 go field enable
13	R/W	0	Vdi7 go line enable
12	R/W	0	Vdi7 if true, negative active input vsync
11	R/W	0	Vdi7 if true, negative active input hsync
10	R/W	0	Vdi7 vsync soft reset fifo enable
9	R/W	0	Vdi7 overflow status clear
8	R/W	0	Vdi7 asfifo soft reset, level signal
7	R/W	0	Vdi6 DE enable
6	R/W	0	Vdi6 go field enable
5	R/W	0	Vdi6 go line enable
4	R/W	0	Vdi6 if true, negative active input vsync
3	R/W	0	Vdi6 if true, negative active input hsync
2	R/W	0	Vdi6 vsync soft reset fifo enable
1	R/W	0	Vdi6 overflow status clear
0	R/W	0	Vdi6 asfifo soft reset, level signal

VDIN0_DOLBY_DSC_CTRL0 0x1275

Bit(s)	R/W	Default	Description
31	R/W	0	Dolby check enable

Bit(s)	R/W	Default	Description
30	R/W	0	Tunnel swap mode enable
29-24	R/W	0	Soft reset control, 29 : dsc, 28~27: crc check, 26~24, reserved
16	R/W	0	Little endian mode
15-0	R/W	0	Monitor metadata position

VDIN0_DOLBY_DSC_CTRL1 0x1276

Bit(s)	R/W	Default	Description
31-16	R/W	0	Metadata pixel start position
7-0	R/W	0	Crc check control

VDIN0_DOLBY_DSC_CTRL2 0x1277

Bit(s)	R/W	Default	Description
31	R/W	0	Metadata read enable
30	R/W	0	Metadata read address auto-increment
29-20	R/W	0	Metadata sum
17-0	R/W	0	Tunnel mode channel selection

VDIN0_DOLBY_DSC_CTRL3 0x1278

Bit(s)	R/W	Default	Description
15-0	R/W	0	Select metadata position

VDIN0_DOLBY_AXI_CTRL0 0x1279

Bit(s)	R/W	Default	Description
31	R/W	0	Memory read enable
30	R/W	0	AXI bus read enable
29	R/W	0	AXI write channel urgent
28	R/W	0	Pack mode little endian
27	R/W	0	AXI bus soft reset
26	R/W	0	Frame reset enable
25	R/W	0	Memory read protection enable
24-16	R/W	0	Memory read sum
15-8	R/W	0	AXI request hold line
7-6	R/W	0	AXI burst length

Bit(s)	R/W	Default	Description
5	R/W	0	Frame buffer start
4	R/W	0	Buffer start
3-2	R/W	0	AXI status monitor control
1-0	R/W	0	awid

VDIN0_DOLBY_ AXI_CTRL1 0x127a

Bit(s)	R/W	Default	Description
31-0	R/W	0	Buffer start address

VDIN0_DOLBY_ AXI_CTRL2 0x127b

Bit(s)	R/W	Default	Description
31-16	R/W	0	Buffer size
15-0	R/W	0	Frame buffer size

VDIN0_DOLBY_ AXI_CTRL3 0x127c

Bit(s)	R/W	Default	Description
7-0	R/W	0	Hold cycle

VDIN1_SCALE_COEF_IDX 0x1300

VDIN1_SCALE_COEF 0x1301

VDIN1_COM_CTRL0 0x1302

Bit(s)	R/W	Default	Description
31	R/W	0	mpeg_to_vdin_sel, 0: mpeg source to NR directly, 1: mpeg source pass through here
30	R/W	0	mpeg_field info which can be written by software
29	R/W	0	force go_field, pulse signal
28	R/W	0	force go_line, pulse signal
27	R/W	0	enable mpeg_go_field input signal
26-20	R/W	0	hold lines
19	R/W	0	delay go_field function enable
18-12	R/W	0	delay go_field line number
11-10	R/W	0	component2 output switch, 00: select component0 in, 01: select component1 in, 10: select component2 in
9-8	R/W	0	component1 output switch, 00: select component0 in, 01: select component1 in, 10: select component2 in

Bit(s)	R/W	Default	Description
7-6	R/W	0	component0 output switch, 00: select component0 in, 01: select component1 in, 10: select component2 in
5	R/W	0	input window selection function enable
4	R/W	0	enable VDIN common data input, otherwise there will be no video data input
3-0	R/W	0	vdin selection, 1: mpeg_in from dram; 2: bt656 input; 3: Reserved (component input); 4: Reserved(tvdecoder input); 5: Reserved(hdmi rx input); 6: digtial video input; 7: internal loop back from Viu; 8: Reserved(mipi_csi2) ;9: Reserved(isp); 10: second bt656 input; otherwise no input.

VDIN1_ACTIVE_MAX_PIX_CNT_STATUS 0x1303

Bit(s)	R/W	Default	Description
28-16	R	0	active_max_pix_cnt, readonly
12-0	R	0	active_max_pix_cnt_shadow, readonly

VDIN1_LCNT_STATUS 0x1304

Bit(s)	R/W	Default	Description
28-16	R	0	go_line_cnt, readonly
12-0	R	0	active_line_cnt, readonly

VDIN1_COM_STATUS0 0x1305

Bit(s)	R/W	Default	Description
12-3	R	0	lfifo_buf_cnt
2	R	0	vdin_direct_done status
1	R	0	vdin_nr_done status
0	R	0	field

VDIN1_COM_STATUS1 0x1306

Bit(s)	R/W	Default	Description
31	R	0	vdi4 fifo overflow
29-24	R	0	vdi3_asfifo_cnt
23	R	0	vdi3 fifo overflow
21-16	R	0	vdi3_asfifo_cnt
15	R	0	vdi2 fifo overflow
13-8	R	0	vdi2_asfifo_cnt
7	R	0	vdi1 fifo overflow

Bit(s)	R/W	Default	Description
5-0	R	0	vdi1_asfifo_cnt

VDIN1_LCNT_SHADOW_STATUS 0x1307

Bit(s)	R/W	Default	Description
28-16	R	0	go_line_cnt_shadow, readonly
12-0	R	0	active_line_cnt_shadow, readonly

VDIN1_ASFIFO_CTRL0 0x1308

Bit(s)	R/W	Default	Description
23	R/W	0	vdi2 DE enable
22	R/W	0	vdi2 go field enable
21	R/W	0	vdi2 go line enable
20	R/W	0	vdi2 if true, negative active input vsync
19	R/W	0	vdi2 if true, negative active input hsync
18	R/W	0	vdi2 vsync soft reset fifo enable
17	R/W	0	vdi2 overflow status clear
16	R/W		vdi2 asfifo soft reset, level signal
7	R/W	0	Vdi1 DE enable
6	R/W	0	Vdi1 go field enable
5	R/W	0	Vdi1 go line enable
4	R/W	0	Vdi1 if true, negative active input vsync
3	R/W	0	Vdi1 if true, negative active input hsync
2	R/W	0	Vdi1 vsync soft reset fifo enable
1	R/W	0	Vdi1 overflow status clear
0	R/W	0	Vdi1 asfifo soft reset, level signal

VDIN1_ASFIFO_CTRL1 0x1309

Bit(s)	R/W	Default	Description
23	R/W	0	Vdi4 DE enable
22	R/W	0	Vdi4 go field enable
21	R/W	0	Vdi4 go line enable
20	R/W	0	Vdi4 if true, negative active input vsync

19	R/W	0	Vdi4 if true, negative active input hsync
18	R/W	0	Vdi4 vsync soft reset fifo enable
17	R/W	0	Vdi4 overflow status clear
16	R/W	0	Vdi4 asfifo soft reset, level signal
7	R/W	0	Vdi3 DE enable
6	R/W	0	Vdi3 go field enable
5	R/W	0	Vdi3 go line enable
4	R/W	0	Vdi3 if true, negative active input vsync
3	R/W	0	Vdi3 if true, negative active input hsync
2	R/W	0	Vdi3 vsync soft reset fifo enable
1	R/W	0	Vdi3 overflow status clear
0	R/W	0	Vdi3 asfifo soft reset, level signal

VDIN1_WIDTHM1I_WIDTHM1O 0x130a

Bit(s)	R/W	Default	Description
28-16	R/W	0	input width minus 1, after the window function
12-0	R/W	0	output width minus 1

VDIN1_SC_MISC_CTRL 0x130b

Bit(s)	R/W	Default	Description
14-8	R/W	0	hsc_ini_pixi_ptr, signed data, only useful when short_lineo_en is true
7	R/W	0	prehsc_en
6	R/W	0	hsc_en
5	R/W	0	hsc_short_lineo_en, short line output enable
4	R/W	0	hsc_nearest_en
3	R/W	0	Hsc_phase0_always_en
3	R/W	0	phase0_always_en
2-0	R/W	0	hsc_bank_length

VDIN1_HSC_PHASE_STEP 0x130c

Bit(s)	R/W	Default	Description
28-24	R/W	0	integer portion
23-0	R/W	0	fraction portion

VDIN1_HSC_INI_CTRL 0x130d

Bit(s)	R/W	Default	Description
30-29	R/W	0	hscale rpt_p0_num
28-24	R/W	0	hscale ini_rcv_num
23-0	R/W	0	hscale ini_phase

VDIN1_COM_STATUS2 0x130e

Bit(s)	R/W	Default	Description
23	R	0	Vdi7 fifo overflow
21-16	R	0	Vdi7_asfifo_cnt
15	R	0	Vdi6 fifo overflow
13-8	R	0	Vdi6_asfifo_cnt
7	R	0	vdi5 fifo overflow
5-0	R	0	vdi5_asfifo_cnt

VDIN1_ASFIFO_CTRL2 0x130f

Bit(s)	R/W	Default	Description
25	R/W	0	if true, decimation counter sync with first valid DE in the field, //otherwise the decimation counter is not sync with external signal
24	R/W	0	decimation de enable
23-20	R/W	0	decimation phase, which counter value use to decimate,
19-16	R/W	0	decimation number, 0: not decimation, 1: decimation 2, 2: decimation 3 ....
7	R/W	0	Vdi5 DE enable
6	R/W	0	Vdi5 go field enable
5	R/W	0	Vdi5 go line enable
4	R/W	0	Vdi5 if true, negative active input vsync
3	R/W	0	Vdi5 if true, negative active input hsync
2	R/W	0	Vdi5 vsync soft reset fifo enable
1	R/W	0	Vdi5 overflow status clear
0	R/W	0	Vdi5 asfifo soft reset, level signal

VDIN1_MATRIX_CTRL 0x1310

Bit(s)	R/W	Default	Description

0	R/W	0	post conversion matrix enable

VDIN1_MATRIX_COEF00_01 0x1311

Bit(s)	R/W	Default	Description
28-16	R/W	0	coef00
12-0	R/W	0	coef01

VDIN1_MATRIX_COEF02_10 0x1312

Bit(s)	R/W	Default	Description
28-16	R/W	0	coef02
12-0	R/W	0	Coef10

VDIN1_MATRIX_COEF11_12 0x1313

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coef11
12-0	R/W	0	Coef12

VDIN1_MATRIX_COEF20_21 0x1314

Bit(s)	R/W	Default	Description
28-16	R/W	0	Coef20
12-0	R/W	0	coef21

VDIN1_ 0x1315

Bit(s)	R/W	Default	Description
18-16	R/W	0	convrs
7-0	R/W	0	Coef22

VDIN1_MATRIX_OFFSET0_1 0x1316

Bit(s)	R/W	Default	Description
26-16	R/W	0	offset0
10-0	R/W	0	Offset1

VDIN1_MATRIX_OFFSET2 0x1317

Bit(s)	R/W	Default	Description
10-0	R/W	0	Offset2

VDIN1_MATRIX_PRE_OFFSET0_1 0x1318

Bit(s)	R/W	Default	Description
26-16	R/W	0	Pre_offset0
10-0	R/W	0	Pre_Offset1

VDIN1_MATRIX_PRE_OFFSET2 0x1319

Bit(s)	R/W	Default	Description
10-0	R/W	0	Pre_Offset2

VDIN1_LFIFO_CTRL 0x131a

Bit(s)	R/W	Default	Description
11-0	R/W	0	lfifo_buf_size

VDIN1_COM_GCLK_CTRL 0x131b

Bit(s)	R/W	Default	Description
15-14	R/W	0	Gate clock control for blackbar detector
13-12	R/W	0	Gate clock control for hist
11-10	R/W	0	Gate clock control for line fifo
9-8	R/W	0	Gate clock control for matrix
7-6	R/W	0	Gate clock control for horizontal scaler
5-4	R/W	0	Gate clock control for pre scaler
3-2	R/W	0	Gate clock control for vdin_com_proc
1-0	R/W	0	Gate clock control for the vdin reg

VDIN1_INTF_WIDTHM1 0x131c

Bit(s)	R/W	Default	Description
26	R/W	0	VDIN write mif bvalid_sel: 1. Bvalid_singnal from bus, 0: bytes_wr handshakes
25	R/W	0	VDIN write mif burst last sel: 1. All kind of burst last signal include ext_data_last. 0. Used the normal burst last signal
12-0	R/W	0	VDIN input interface width minus 1, before the window function, after the de decimation

VDIN1_WR_CTRL2 0x131f

Bit(s)	R/W	Default	Description
19	R/W	0	Vdin1 wr bit10 mode

18	R/W	0	Data_ext_en 1: send out data if req was interrupt by soft reset 0 : normal mode
17:16	R/W	1	Words lim[1:0]: it would not send out request before Words_lim *16 words were ready
15:12	R/W	1	Burst_ lim : 00 , 1 word in 1burst , 01 ,2 words in 1burst, 10, 4 words in 1burst , 11 reserved
10:9	R/W	0	Words_lim[3:2]
8	R/W	0	1: discard data before line fifo, 0: normal mode
7-0	R/W	0	Write chroma canvas address, for NV12/21 mode.

VDIN1_WR_CTRL 0x1320

Bit(s)	R/W	Default	Description
31-30	R/W	0	vdin1_wr_mif_hconv_mode. Applicable only to vdin_write_format=0 or 2. 0=Output every even pixel's CbCr; 1=Output every odd pixel's CbCr; 2=Output an average value per even&odd pair of pixels; 3=Output all CbCr. Only applys to vdin_write_format =2.
29	R/W	0	vdin1_wr_mif_no_clk_gate. If true, enable free-run clock.
28	R/W	0	clear write response counter in the vdin write memory interface
27	R/W	1	eol_sel, 1: use eol as the line end indication, 0: use width as line end indication in the vdin write memory interface
23	R/W	1	vdin frame reset enble, if true, it will provide frame reset during go_field(vsync) to the modules after that
22	R/W	1	vdin line fifo soft reset enable, meaning, if true line fifo will reset during go_field (vsync)
21	R/W	0	vdin direct write done status clear bit
20	R/W	0	vdin NR write done status clear bit
19	R/W	0	Vdin0_wr words swap : swap the 2 64bits word in 128 words
18	R/W	0	vdin1_wr_mif_swap_cbcr. Applicable only to vdin_write_format =2. 0=Output CbCr (NV12); 1=Output CrCb (NV21);
17:16	R/W	0	vdin1_wr_mif_vconv_mode. Applicable only to vdin_write_format=2. 0=Output every even line's CbCr; 1=Output every odd line's CbCr; 2=Reserved; 3=Output all CbCr.
13-12	R/W	0	vdin_write_format, 0: 4:2:2 to one canvas; 1: 4:4:4 to one canvas; 2: Y to luma canvas, CbCr to chroma canvas, for NV12/21; 3: 422 10 bit full pack mode
11	R/W	0	vdin write canvas double buffer enable, means the canvas address will be latched by vsync before using

Bit(s)	R/W	Default	Description
9	R/W	0	vdin write request urgent
8	R/W	0	vdin write request enable
7-0	R/W	0	Write canvas address (For NV12/21 mode, it's LUMA canvas)

VDIN1_WR_H_START_END 0x1321

Bit(s)	R/W	Default	Description
27-16	R/W	0	start
11-0	R/W	0	end

VDIN1_WR_V_START_END 0x1322

Bit(s)	R/W	Default	Description
27-16	R/W	0	start
11-0	R/W	0	end

VDIN1_VSC_PHASE_STEP 0x1323

Bit(s)	R/W	Default	Description
24-16	R/W	0	integer portion
19-0	R/W	0	fraction portion

VDIN1_VSC_INI_CTRL 0x1324

Bit(s)	R/W	Default	Description
23	R/W	0	vsc_en, vertical scaler enable
21	R/W	0	vsc_phase0_always_en, when scale up, you have to set it to 1
20-16	R/W	0	ini skip_line_num
15-0	R/W	0	vscaler ini_phase

VDIN1_SCIN_HEIGHTM1 0x1325

Bit(s)	R/W	Default	Description
12-0	R/W	0x437	scaler input height minus 1

VDIN1_DUMMY_DATA 0x1326

Bit(s)	R/W	Default	Description
23-16	R/W	0	dummy component 0
15-8	R/W	0x80	dummy component 1

7-0	R/W	0x80	dummy component 2

VDIN1_MATRIX_PROBE_COLOR 0x1328

Bit(s)	R/W	Default	Description
29-20	R/W	0	component 0
19-10	R/W	0	omponent 1
9-0	R/W	0	component 2

VDIN1_MATRIX_HL_COLOR 0x1329

Bit(s)	R/W	Default	Description
23-16	R/W	0	component 0
15-8	R/W	0	component 1
7-0	R/W	0	component 2

VDIN1_MATRIX_PROBE_POS 0x132a

Bit(s)	R/W	Default	Description
28-16	R/W	0	probe x, position
12-0	R/W	0	probe y, postion

VDIN1_HIST_CTRL 0x1330

Bit(s)	R/W	Default	Description
31-24	R/W	0	Hist pixel white threshold, larger than this will be counted as white pixel number
23-16	R/W	0	Hist pixel black threshold, less than this will be counted as black pixel number
11	R/W	0	Hist 32bin only mode
10-9	R/W	0	ldim_stts_din_sel, 00: from matrix0 dout, 01: from vsc_dout, 10: from matrix1 dout, 11: form matrix1 din
8	R/W	0	ldim_stts_en
6-5	R/W	0	hist_dnlp_low the real pixels in each bins got by VDIN_DNLP_HISTXX should multiple with 2^(dnlp_low+3)
3-2	R/W	0	hist_din_sel the source used for hist statistics. 2'b00: from MAT0_dout; 2'b01: from vsc_dout; 2'b10: from mat1_dout, 3: mat1_din
1	R/W	0	hist_win_en 1'b0: hist used for full picture; 1'b1: hist used for pixels within hist window
0	R/W	0	hist_spl_en 1'b0: disable hist readback; 1'b1: enable hist readback

VDIN1_HIST_H_START_END 0x1331

Bit(s)	R/W	Default	Description
28-16	R/W	0	hist_hstart horizontal start value to define hist window
12-0	R/W	0	hist_hend horizontal end value to define hist window

VDIN1_HIST_V_START_END 0x1332

Bit(s)	R/W	Default	Description
28-16	R/W	0	hist_vstart vertical start value to define hist window
12-0	R/W	0	hist_vend vertical end value to define hist window

VDIN1_HIST_MAX_MIN 0x1333

Bit(s)	R/W	Default	Description
15-8	R	0	hist_max maximum value
7-0	R	0	hist_min minimum value

VDIN1_HIST_SPL_VAL 0x1334

Bit(s)	R/W	Default	Description
31-0	R	0	hist_spl_rd , counts for the total luma value

VDIN1_HIST_SPL_PIX_CNT 0x1335

Bit(s)	R/W	Default	Description
21-0	R	0	hist_spl_pixel_count, counts for the total calculated pixels

VDIN1_HIST_CHROMA_SUM 0x1336

Bit(s)	R/W	Default	Description
31-0	R	0	hist_chroma_sum , counts for the total chroma value

VDIN1_DNLP_HIST00 0x1337

//0-255 are splited to 64 bins evenly, and VDIN_DNLP_HISTXX

//are the statistic number of pixels that within each bin.

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 2nd bin
15-0	R	0	counts for the 1st bin

VDIN1_DNLP_HIST01 0x1338

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 4th bin

Bit(s)	R/W	Default	Description
15-0	R	0	counts for the 3rd bin

VDIN1_DNLP_HIST02 0x1339

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 6th bin
15-0	R	0	counts for the 5th bin

VDIN1_DNLP_HIST03 0x133a

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 8th bin
15-0	R	0	counts for the 7th bin

VDIN1_DNLP_HIST04 0x133b

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 10th bin
15-0	R	0	counts for the 9th bin

VDIN1_DNLP_HIST05 0x133c

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 12th bin
15-0	R	0	counts for the 11th bin

VDIN1_DNLP_HIST06 0x133d

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 14th bin
15-0	R	0	counts for the 13th bin

VDIN1_DNLP_HIST07 0x133e

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 16th bin
15-0	R	0	counts for the 15th bin

VDIN1_DNLP_HIST08 0x133f

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 18th bin

15-0	R	0	counts for the 17th bin

VDIN1_DNLP_HIST09 0x1340

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 20th bin
15-0	R	0	counts for the 19th bin

VDIN1_DNLP_HIST10 0x1341

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 22nd bin
15-0	R	0	counts for the 21st bin

VDIN1_DNLP_HIST11 0x1342

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 24th bin
15-0	R	0	counts for the 23rd bin

VDIN1_DNLP_HIST12 0x1343

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 26th bin
15-0	R	0	counts for the 25th bin

VDIN1_DNLP_HIST13 0x1344

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 28th bin
15-0	R	0	counts for the 27th bin

VDIN1_DNLP_HIST14 0x1345

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 30th bin
15-0	R	0	counts for the 29th bin

VDIN1_DNLP_HIST15 0x1346

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 32nd bin
15-0	R	0	counts for the 31st bin

VDIN1_DNLP_HIST16 0x1347

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 34th bin
15-0	R	0	counts for the 33rd bin

VDIN1_DNLP_HIST17 0x1348

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 36th bin
15-0	R	0	counts for the 35th bin

VDIN1_DNLP_HIST18 0x1349

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 38th bin
15-0	R	0	counts for the 37th bin

VDIN1_DNLP_HIST19 0x134a

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 40th bin
15-0	R	0	counts for the 39th bin

VDIN1_DNLP_HIST20 0x134b

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 42nd bin
15-0	R	0	counts for the 41st bin

VDIN1_DNLP_HIST21 0x134c

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 44th bin
15-0	R	0	counts for the 43rd bin

VDIN1_DNLP_HIST22 0x134d

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 46th bin
15-0	R	0	counts for the 45th bin

VDIN1_DNLP_HIST23 0x134e

Bit(s)	R/W	Default	Description

31-16	R	0	counts for the 48th bin
15-0	R	0	counts for the 47th bin

VDIN1_DNLP_HIST24 0x134f

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 50th bin
15-0	R	0	counts for the 49th bin

VDIN1_DNLP_HIST25 0x1350

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 52nd bin
15-0	R	0	counts for the 51st bin

VDIN1_DNLP_HIST26 0x1351

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 54th bin
15-0	R	0	counts for the 53rd bin

VDIN1_DNLP_HIST27 0x1352

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 56th bin
15-0	R	0	counts for the 55th bin

VDIN1_DNLP_HIST28 0x1353

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 58th bin
15-0	R	0	counts for the 57th bin

VDIN1_DNLP_HIST29 0x1354

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 60th bin
15-0	R	0	counts for the 59th bin

VDIN1_DNLP_HIST30 0x1355

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 62nd bin

15-0	R	0	counts for the 61st bin

VDIN1_DNLP_HIST31 0x1356

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 64th bin
15-0	R	0	counts for the 63rd bin

VDIN1_LDIM_STTS_HIST_REGION_IDX 0x1357

Bit(s)	R/W	Default	Description
31	R	0	local dimming max statistic enable
28	R	0	eol enable
27-25	R	0	vertical line overlap number for max finding
24-22	R	0	horizontal pixel overlap number, 0: 17 pix, 1: 9 pix, 2: 5 pix, 3: 3 pix, 4: 0 pix
20	R	0	1,2,1 low pass filter enable before max/hist statistic
19-16	R	0	region H/V position index, refer to VDIN_LDIM_STTS_HIST_SET_REGION
15	R	0	1: region read index auto increase per read to VDIN_LDIM_STTS_HIST_READ_REGION
6-0	R	0	region read index

VDIN1_LDIM_STTS_HIST_SET_REGION 0x1358

Bit(s)	R/W	Default	Description

28:16	R	0	if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h0: read/write hvstart0
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h1: read/write hend01
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h2: read/write vend01
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h3: read/write hend23
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h4: read/write vend23
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h5: read/write hend45
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h6: read/write vend45
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'd7: read/write hend67
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h8: read/write vend67
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'h9: read/write hend89
			if VDIN_LDIM_STTS_HIST_REGION_IDX[19:16] == 5'ha: read/write vend89
			//hvstart0, Bit 28:16 row0 vstart, Bit 12:0 col0 hstart
			//hend01, Bit 28:16 col1 hend, Bit 12:0 col0 hend
			//vend01, Bit 28:16 row1 vend, Bit 12:0 row0 vend
			//hend23, Bit 28:16 col3 hend, Bit 12:0 col2 hend
			//vend23, Bit 28:16 row3 vend, Bit 12:0 row2 vend
			//hend45, Bit 28:16 col5 hend, Bit 12:0 col4 hend
			//vend45, Bit 28:16 row5 vend, Bit 12:0 row4 vend
			//hend67, Bit 28:16 col7 hend, Bit 12:0 col6 hend
			//vend67, Bit 28:16 row7 vend, Bit 12:0 row6 vend
			//hend89, Bit 28:16 col9 hend, Bit 12:0 col8 hend
			//vend89, Bit 28:16 row9 vend, Bit 12:0 row8 vend

12:0	R	0	

VDIN1_LDIM_STTS_HIST_READ_REGION 0x1359

Bit(s)	R/W	Default	Description
29:20	R	0	Max_comp2
19:10	R	0	Max_comp1
9:0	R	0	Max_comp0

VDIN1_MEAS_CTRL0 0x135a

Bit(s)	R/W	Default	Description
18	R/W	0	reset bit, high active
17	R/W	0	if true, widen hs/vs pulse
16	R/W	0	vsync total counter always accumulating enable
14-12	R/W	0	select hs/vs of video input channel to measure, 0: no selection, 1 vdi1, 2: vid2: 3: vid3, 4:vid4, 5:vdi5, 6:vdi6, 7:vdi7.
11-4	R/W	0	vsync_span, define how many vsync span need to measure

2-0	R/W	0	meas_hs_index, index to select which HS counter/range

VDIN1_MEAS_VS_COUNT_HI 0x135b

Bit(s)	R/W	Default	Description
19-16	R	0	meas_ind_total_count_n, every number of sync_span vsyncs, this count add 1
15-0	R	0	high bit portion of vsync total counter

VDIN1_MEAS_VS_COUNT_LO 0x135c

Bit(s)	R/W	Default	Description
31-0	R	0	low bit portion of vsync total counter

VDIN1_MEAS_HS_RANGE 0x135d

//according to the meas_hs_index in register VDIN_MEAS_CTRL0

//meas_hs_index == 0, first hs range

//meas_hs_index == 1, second hs range

//meas_hs_index == 2, third hs range

//meas_hs_index == 3, fourth hs range

Bit(s)	R/W	Default	Description
28-16	R	0	count_start
12-0	R	0	count_end

VDIN1_MEAS_HS_COUNT 0x135e

//according to the meas_hs_index in register VDIN_MEAS_CTRL0,

//meas_hs_index == 0, first range hs counter,

//meas_hs_index == 1, second range hs

//meas_hs_index == 2, third range hs

//meas_hs_index == 3, fourth range hs

Bit(s)	R/W	Default	Description
23-0	R	0	Hs counter

VDIN1_BLKBAR_CTRL1 0x135f

Bit(s)	R/W	Default	Description
8	R/W	0	white_enable
7-0	R/W	0	blkbar_white_level

VDIN1_BLKBAR_CTRL0 0x1360

Bit(s)	R/W	Default	Description
31-24	R/W	0	blkbar_black_level threshold to judge a black point

20-8	R/W	0	blkbar_hwidth left and right region width
7-5	R/W	0	blkbar_comp_sel select yin or uin or vin to be the valid input
4	R/W	0	blkbar_sw_statistic_en enable software statistic of each block black points number
3	R/W	0	blkbar_det_en
2-1	R/W	0	blkbar_din_sel
0	R/W	0	Blkbar_det_top_en

VDIN1_BLKBAR_H_START_END 0x1361

Bit(s)	R/W	Default	Description
28-16	R/W	0	blkbar_hstart. Left region start
12-0	R/W	0	blkbar_hend. Right region end

VDIN1_BLKBAR_V_START_END 0x1362

Bit(s)	R/W	Default	Description
28-16	R/W	0	blkbar_vstart.
12-0	R/W	0	blkbar_vend.

VDIN1_BLKBAR_CNT_THRESHOLD 0x1363

Bit(s)	R/W	Default	Description
19-0	R/W	0	blkbar_cnt_threshold. threshold to judge whether a block is totally black

VDIN1_BLKBAR_ROW_TH1_TH2 0x1364

Bit(s)	R/W	Default	Description
28-16	R/W	0	blkbar_row_th1. //threshold of the top blackbar
12-0	R/W	0	blkbar_row_th2 //threshold of the bottom blackbar

VDIN1_BLKBAR_IND_LEFT_START_END 0x1365

Bit(s)	R/W	Default	Description
28-16	R	0	blkbar_ind_left_start. horizontal start of the left region in the current searching
12-0	R	0	blkbar_ind_left_end. horizontal end of the left region in the current searching

VDIN1_BLKBAR_IND_RIGHT_START_END 0x1366

Bit(s)	R/W	Default	Description
28-16	R	0	blkbar_ind_right_start.horizontal start of the right region in the current searching
12-0	R	0	blkbar_ind_right_end. horizontal end of the right region in the current searching

VDIN1_BLKBAR_IND_LEFT1_CNT 0x1367

Bit(s)	R/W	Default	Description
19-0	R	0	blkbar_ind_left1_cnt. Black pixel counter. left part of the left region

VDIN1_BLKBAR_IND_LEFT2_CNT 0x1368

Bit(s)	R/W	Default	Description
19-0	R	0	blkbar_ind_left2_cnt. Black pixel counter. right part of the left region

VDIN1_BLKBAR_IND_RIGHT1_CNT 0x1369

Bit(s)	R/W	Default	Description
19-0	R	0	blkbar_ind_right1_cnt. Black pixel counter. left part of the right region

VDIN1_BLKBAR_IND_RIGHT2_CNT 0x136a

Bit(s)	R/W	Default	Description
19-0	R	0	blkbar_ind_right2_cnt. Black pixel counter. right part of the right region

VDIN1_BLKBAR_STATUS0 0x136b

Bit(s)	R/W	Default	Description
29	R	0	blkbar_ind_black_det_done. LEFT/RIGHT Black detection done
28-16	R	0	blkbar_top_pos. Top black bar position
12-0	R	0	blkbar_bot_pos. Bottom black bar position

VDIN1_BLKBAR_STATUS1 0x136c

Bit(s)	R/W	Default	Description
28-16	R	0	blkbar_left_pos. Left black bar posiont
12-0	R	0	blkbar_right_pos. Right black bar position

VDIN1_WIN_H_START_END 0x136d

Bit(s)	R/W	Default	Description
28-16	R/W	0	input window H start
12-0	R/W	0	input window H end

VDIN1_WIN_V_START_END 0x136e

Bit(s)	R/W	Default	Description
28-16	R/W	0	input window V start
12-0	R/W	0	input window V end

VDIN1_ASFIFO_CTRL3 0x136f

Bit(s)	R/W	Default	Description
15	R/W	0	Vdi7 DE enable
14	R/W	0	Vdi7 go field enable
13	R/W	0	Vdi7 go line enable
12	R/W	0	Vdi7 if true, negative active input vsync
11	R/W	0	Vdi7 if true, negative active input hsync
10	R/W	0	Vdi7 vsync soft reset fifo enable
9	R/W	0	Vdi7 overflow status clear
8	R/W	0	Vdi7 asfifo soft reset, level signal
7	R/W	0	Vdi6 DE enable
6	R/W	0	Vdi6 go field enable
5	R/W	0	Vdi6 go line enable
4	R/W	0	Vdi6 if true, negative active input vsync
3	R/W	0	Vdi6 if true, negative active input hsync
2	R/W	0	Vdi6 vsync soft reset fifo enable
1	R/W	0	Vdi6 overflow status clear
0	R/W	0	Vdi6 asfifo soft reset, level signal

VDIN1_COM_GCLK_CTRL2 0x1370

Bit(s)	R/W	Default	Description
3-2	R/W	0	Vshrk_clk2 ctrl
1-0	R/W	0	Vshrk_clk1 ctrl

VDIN1_VSHRK_CTRL 0x1371

Bit(s)	R/W	Default	Description
27	R/W	0	Vshrk enable
26:25	R/W	0	Vshrk mode, 0: 1/2 shrink, 1: 1/4 shrink, 2: 1/8 shrink
24	R/W	0	Vshrink lpf mode, 1: 0.5,1.5,1.5,0.5 lpf for 1/4 shrink, 0.5,1.5,1.5...for 1/8 shrink
23:0	R/W	0	Vshrink padding dummy data

VDIN1_HIST32 0x1372

Bit(s)	R/W	Default	Description

31:0	R	0	Hist 32 mode, [31:16] for white pixel number, 15:0 for black pixel number

VDIN1_COM_STATUS3 0x133

Bit(s)	R/W	Default	Description
7	R	0	Vdi9 fifo overflow
5:0	R	0	Vdi9 asfifo cnt

VI_HIST_CTRL 0x2e00

Bit(s)	R/W	Default	Description
17-16	R/W	0	Spl_sft: the splt are right shift by spl_sft, 0: no shift, 1: right shift by 1. 2,3...
14	R/W	0	Hist_34bin_only, bin 32~63 are not valid, there are 34bins, bin0~bin31, and bin 64 for black pixel, bin 65 for white pixel
13-11	R/W	0	Hist_in_sel: 0: vpp_dout, 1: vpp_vd1_din, 2: vpp_vd2_din, 3: osd1, 4:osd2 5: di pre 6: vdin 7: post blend
10-8	R/W	0	Hist_din_comp_mux: mux of each component, din[9:0],[19:10],[29:20] switches
7-5	R/W	0	Hist_dnlp_low: hist number are shift by (hist_dnlp_low + 3). I.e. Dnlp_low =0, >> 3, dnlp_low=1, >> 4
1	R/W	0	Hist_win_en: hist statistic in a window
0	R/W	0	Luma_hist_spl_en, 1: enable the histogram statistic

VI_HIST_H_START_END 0x2e01

Bit(s)	R/W	Default	Description
28-16	R/W	0	Hist_hstart, refer to VI_HIST_CTRL[1]
12-0	R/W	0	Hist_hend

VI_HIST_V_START_END 0x2e02

Bit(s)	R/W	Default	Description
28-16	R/W	0	Hist_vstart, refer to VI_HIST_CTRL[1]
12-0	R/W	0	Hist_vend

VI_HIST_MAX_MIN 0x2e03

Bit(s)	R/W	Default	Description
15-8	R	0	hist_max maximum value
7-0	R	0	hist_min minimum value

VI_HIST_SPL_VAL 0x2e04

Bit(s)	R/W	Default	Description

31-0	R	0	hist_spl_rd , counts for the total luma value

VI_HIST_SPL_PIX_CNT 0x2e05

Bit(s)	R/W	Default	Description
21-0	R	0	hist_spl_pixel_count, counts for the total calculated pixels

VI_HIST_CHROMA_SUM 0x2e06

Bit(s)	R/W	Default	Description
31-0	R	0	hist_chroma_sum , counts for the total chroma value

VI_DNLP_HIST00 0x2e07

//0-255 are splited to 64 bins evenly, and VDIN_DNLP_HISTXX

//are the statistic number of pixels that within each bin.

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 2nd bin
15-0	R	0	counts for the 1st bin

VI_DNLP_HIST01 0x2e08

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 4th bin
15-0	R	0	counts for the 3rd bin

VI_DNLP_HIST02 0x2e09

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 6th bin
15-0	R	0	counts for the 5th bin

VI_DNLP_HIST03 0x2e0a

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 8th bin
15-0	R	0	counts for the 7th bin

VI_DNLP_HIST04 0x2e0b

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 10th bin
15-0	R	0	counts for the 9th bin

VI_DNLP_HIST05 0x2e0c

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 12th bin
15-0	R	0	counts for the 11th bin

VI_DNLP_HIST06 0x2e0d

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 14th bin
15-0	R	0	counts for the 13th bin

VI_DNLP_HIST07 0x2e0e

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 16th bin
15-0	R	0	counts for the 15th bin

VI_DNLP_HIST08 0x2e0f

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 18th bin
15-0	R	0	counts for the 17th bin

VI_DNLP_HIST09 0x2e10

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 20th bin
15-0	R	0	counts for the 19th bin

VI_DNLP_HIST10 0x2e11

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 22nd bin
15-0	R	0	counts for the 21st bin

VI_DNLP_HIST11 0x2e12

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 24th bin
15-0	R	0	counts for the 23rd bin

VI_DNLP_HIST12 0x2e13

Bit(s)	R/W	Default	Description

31-16	R	0	counts for the 26th bin
15-0	R	0	counts for the 25th bin

VI_DNLP_HIST13 0x2e14

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 28th bin
15-0	R	0	counts for the 27th bin

VI_DNLP_HIST14 0x2e15

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 30th bin
15-0	R	0	counts for the 29th bin

VI_DNLP_HIST15 0x2e16

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 32nd bin
15-0	R	0	counts for the 31st bin

VI_DNLP_HIST16 0x2e17

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 34th bin
15-0	R	0	counts for the 33rd bin

VI_DNLP_HIST17 0x2e18

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 36th bin
15-0	R	0	counts for the 35th bin

VI_DNLP_HIST18 0x2e19

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 38th bin
15-0	R	0	counts for the 37th bin

VI_DNLP_HIST19 0x2e1a

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 40th bin

15-0	R	0	counts for the 39th bin

VI_DNLP_HIST20 0x2e1b

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 42nd bin
15-0	R	0	counts for the 41st bin

VI_DNLP_HIST21 0x2e1c

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 44th bin
15-0	R	0	counts for the 43rd bin

VI_DNLP_HIST22 0x2e1d

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 46th bin
15-0	R	0	counts for the 45th bin

VI_DNLP_HIST23 0x2e1e

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 48th bin
15-0	R	0	counts for the 47th bin

VI_DNLP_HIST24 0x2e1f

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 50th bin
15-0	R	0	counts for the 49th bin

VI_DNLP_HIST25 0x2e20

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 52nd bin
15-0	R	0	counts for the 51st bin

VI_DNLP_HIST26 0x2e21

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 54th bin
15-0	R	0	counts for the 53rd bin

VI_DNLP_HIST27 0x2e22

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 56th bin
15-0	R	0	counts for the 55th bin

VI_DNLP_HIST28 0x2e23

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 58th bin
15-0	R	0	counts for the 57th bin

VI_DNLP_HIST29 0x2e24

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 60th bin
15-0	R	0	counts for the 59th bin

VI_DNLP_HIST30 0x2e25

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 62nd bin
15-0	R	0	counts for the 61st bin

VI_DNLP_HIST31 0x2e26

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 64th bin
15-0	R	0	counts for the 63rd bin

VI_DNLP_HIST31 0x2e27

Bit(s)	R/W	Default	Description
31-16	R	0	counts for the 66th bin, for white pix
15-0	R	0	counts for the 65th bin, for black pix

VI_HIST_PIC_SIZE 0x2e28

Bit(s)	R/W	Default	Description
28-16	R/W	0	Hist_pic_height
12-0	R/W	0	Hist_pic_width

VI_HIST_GCLK_CTRL 0x2e2a

Bit(s)	R/W	Default	Description

5-4	R/W	0	Gated clock control of hist_clk
3-2	R/W	0	Gated clock control of clk0
1-0	R/W	0	Gated clock control of hist register clock

10.2.3.43 HDMITX Top-Level and HDMI TX Contoller IP Register Access

Accessing HDMITX Top-Level and TX Controller IP registers is by directly accessing memory addresses. Due to the difference between the data width of Top-level register and IP register – Top-level 4-byte wide and IP register 1-byte wide, below list the correct ways to access Top-level and IP registers.

To access Top-level registers:

```
#define HDMITX_TOP_BASE_OFFSET              0xff608000
#define HDMITX_TOP_REVOCMEM_ADDR_S          0x2000
#define HDMITX_TOP_REVOCMEM_ADDR_E          0x365E
void hdmitx_wr_only_TOP (uint32_t addr, uint32_t data) {
    if ((addr >= HDMITX_TOP_REVOCMEM_ADDR_S) &&
       (addr <= HDMITX_TOP_REVOCMEM_ADDR_E)) {
    *(volatile uint8_t *)(HDMITX_TOP_BASE_OFFSET+addr) = (uint8_t)(data&0xff);
  } else {
    *(volatile uint32_t *)(HDMITX_TOP_BASE_OFFSET+addr) = (data);
  }
}
uint32_t hdmitx_rd_TOP (uint32_t addr) {
  uint32_t  data;
    if ((addr >= HDMITX_TOP_REVOCMEM_ADDR_S) &&
       (addr <= HDMITX_TOP_REVOCMEM_ADDR_E)) {
    data = (uint32_t)(*(volatile uint8_t *)(HDMITX_TOP_BASE_OFFSET+addr));
  } else {
    data = *(volatile uint32_t *)(HDMITX_TOP_BASE_OFFSET+addr);
  }
  return (data);
}
```

10.2.3.44 HDCP2.2 IP Register Access

Table 10-1 HDCP2.2 IP register base address

Absolute Address	Address Mnemonic	Description
0xffe01000	ELP_ESM_HPI_REG_BASE	Address base to HDCP2.2 IP register access.

To write to an HDCP2.2 IP register:

```
  *((volatile uint32_t *) (ELP_ESM_HPI_REG_BASE+addr)) = data;
```

To read from an HDCP2.2 IP register:

*data = *((volatile uint32_t *) (ELP_ESM_HPI_REG_BASE+addr));*

Table 10-2 HDMITX Top-Level Registers

Addr	Name	RW	Function
0x000<<2	HDMITX_TOP_SW_RESET	RW	Software reset sub-modules.
0x001<<2	HDMITX_TOP_CLK_CNTL	RW	Clock gating and inversion.
0x002<<2	HDMITX_TOP_HPD_FILTER	RW	HPD and RxSense input glitch filter.
0x003<<2	HDMITX_TOP_INTR_MASKN	RW	Interrupt mask.
0x004<<2	HDMITX_TOP_INTR_STAT	RW	Interrupt status.
0x005<<2	HDMITX_TOP_INTR_STAT_CLR	W	Interrupt clear.
0x006<<2	HDMITX_TOP_BIST_CNTL	RW	Build-In Self Test(BIST) control.
0x007<<2	HDMITX_TOP_SHIFT_PTTN_012	RW	Shift pattern for BIST.
0x008<<2	HDMITX_TOP_SHIFT_PTTN_345	RW	Shift pattern for BIST.
0x009<<2	HDMITX_TOP_SHIFT_PTTN_67	RW	Shift pattern for BIST.
0x00A<<2	HDMITX_TOP_TMDS_CLK_PTTN_01	RW	TMDS clock pattern for generating /10 or /40 rate clock.
0x00B<<2	HDMITX_TOP_TMDS_CLK_PTTN_23	RW	TMDS clock pattern for generating /10 or /40 rate clock.
0x00C<<2	HDMITX_TOP_TMDS_CLK_PTTN_CNTL	RW	TMDS clock pattern for generating /10 or /40 rate clock.
0x00D<<2	HDMITX_TOP_REVOCMEM_STAT	RW	Revocmem status
0x00E<<2	HDMITX_TOP_STAT0	RW	Status.
0x010<<2	HDMITX_TOP_SKP_CNTL_STAT	RW	SKP interface control for HDCP2.2.
0x011<<2	HDMITX_TOP_NONCE_0	W	Nonce[31:0] for HDCP2.2.
0x012<<2	HDMITX_TOP_NONCE_1	W	Nonce[63:32] for HDCP2.2.
0x013<<2	HDMITX_TOP_NONCE_2	W	Nonce[95:64] for HDCP2.2.
0x014<<2	HDMITX_TOP_NONCE_3	W	Nonce[127:96] for HDCP2.2.
0x015<<2	HDMITX_TOP_PKF_0	W	PKF[31:0] for HDCP2.2.
0x016<<2	HDMITX_TOP_PKF_1	W	PKF[63:32] for HDCP2.2.
0x017<<2	HDMITX_TOP_PKF_2	W	PKF[95:64] for HDCP2.2.
0x018<<2	HDMITX_TOP_PKF_3	W	PKF[127:96] for HDCP2.2.
0x019<<2	HDMITX_TOP_DUK_0	W	DUK [31:0] for HDCP2.2.

Addr	Name	RW	Function
0x01A<<2	HDMITX_TOP_DUK_1	W	DUK [63:32] for HDCP2.2.
0x01B<<2	HDMITX_TOP_DUK_2	W	DUK [95:64] for HDCP2.2.
0x01C<<2	HDMITX_TOP_DUK_3	W	DUK [127:96] for HDCP2.2.
0x01D<<2	HDMITX_TOP_INFILTER	RW	DDC and CEC input glitch filter control.
0x01E<<2	HDMITX_TOP_NSEC_SCRATCH	RW	Scratch register for non-secure access.
0x01F<<2	HDMITX_TOP_SEC_SCRATCH	RW	Scratch register for secure access.
0x020<<2	HDMITX_TOP_EMP_CNTL0	RW	EMP control.
0x021<<2	HDMITX_TOP_EMP_CNTL1	RW	EMP control.
0x022<<2	HDMITX_TOP_EMP_MEMADDR_START	RW	Mem addr pointer for EMP.
0x023<<2	HDMITX_TOP_EMP_STAT0	R	EMP status.
0x024<<2	HDMITX_TOP_EMP_STAT1	R	EMP status.
0x025<<2	HDMITX_TOP_AXI_ASYNC_CNTL0	RW	AXI async control.
0x026<<2	HDMITX_TOP_AXI_ASYNC_CNTL1	RW	AXI async control.
0x027<<2	HDMITX_TOP_AXI_ASYNC_STAT0	R	AXI async status.
0x028<<2	HDMITX_TOP_I2C_BUSY_CNT_MAX	RW	Max I2C idle time after I2C Start.
0x029<<2	HDMITX_TOP_I2C_BUSY_CNT_STAT	RW	I2C idle time status.
0x02A<<2	HDMITX_TOP_HDCP22_BSOD	RW	HDCP22 BSOD override control.
0x02B<<2	HDMITX_TOP_DDC_CNTL	RW	DDC pull down by SW.
0x2000	HDMITX_TOP_REVOCMEM_ADDR_S	RW	Map to third party's IP internal Revoc MEM start address.
0x365E	HDMITX_TOP_REVOCMEM_ADDR_E	RW	Map to third party's IP internal Revoc MEM end address.

HDMITX_TOP_SW_RESET

Bit	R/W	Default	Description
31:8	R	0	Reserved
15:10	RW	0x3f	Reserved
9	RW	0	sw_reset_i2c: to reset third party IP's I2C module. 0=Release from reset; 1=Apply reset.

Bit	R/W	Default	Description
8	RW	1	sw_reset_axiarb: to reset AXI arbiter between HDCP22 and EMP. 0=Release from reset; 1=Apply reset.
7	RW	1	sw_reset_emp: to reset EMP block. 0=Release from reset; 1=Apply reset.
6	RW	1	sw_reset_flt: to reset DDC&CEC input glitch filter. 0=Release from reset; 1=Apply reset.
5	RW	1	sw_reset_hdcp22: to reset HDCP2.2 IP. 0=Release from reset; 1=Apply reset.
4	RW	1	sw_reset_phyif: to reset PHY interface. 0=Release from reset; 1=Apply reset.
3	RW	1	sw_reset_intr: to reset interrupt block. 0=Release from reset; 1=Apply reset.
2	RW	1	sw_reset_mem: to reset KSV/REVOC mem. 0=Release from reset; 1=Apply reset.
1	RW	1	sw_reset_rnd: to reset random number interface to HDCP. 0=Release from reset; 1=Apply reset.
0	RW	1	sw_reset_core: To reset TX Controller IP. 0=Release from reset; 1=Apply reset.

HDMITX_TOP_CLK_CNTL

Bit	R/W	Default	Description
31	RW	0	free_clk_en: 0= Enable clock gating for power saving; 1= Disable clock gating, enable free-run clock.
30:13	RW	0	Reserved
12	RW	0	i2s_ws_inv: 1= Invert i2s_ws.

Bit	R/W	Default	Description
11	RW	0	i2s_clk_inv: 1= Invert i2s_clk.
10	RW	0	spdif_clk_inv: 1= Invert spdif_clk.
9	RW	0	tmds_clk_inv: 1= Invert tmds_clk.
8	RW	0	pixel_clk_inv: 1= Invert pixel_clk.
7	RW	0	hdcp22_skpclk_en: 1= Enable skpclk to HTX_HDCP2.2 IP.
6	RW	0	hdcp22_esmclk_en: 1= Enable esmclk to HTX_HDCP2.2 IP.
5	RW	0	hdcp22_tmdsclk_en: 1= Enable tmds_clk to HDCP2.2 IP.
4	RW	0	Reserved.
3	RW	0	i2s_clk_en: 1= Enable i2s_clk.
2	RW	0	spdif_clk_en: 1= Enable spdif_clk.
1	RW	0	tmds_clk_en: 1= Enable tmds_clk.
0	RW	0	pixel_clk_en: 1= Enable pixel_clk.

HDMITX_TOP_HPD_FILTER

Bit	R/W	Default	Description
31:16	R	0	Reserved
31:28	RW	0	rxsense_glitch_width: Filter out glitch <= rxsense_glitch_width.
27:16	RW	0	rxsense_valid_width: Filter out width <= rxsense_valid_width * 1024.
15:12	RW	0	hpd_glitch_width: Filter out glitch <= hpd_glitch_width.

Bit	R/W	Default	Description
11:0	RW	0	hpd_valid_width: Filter out width <= hpd_valid_width * 1024.

HDMITX_TOP_INTR_MASKN

Interrupt MASKN, one bit per interrupt source.

0= Disable interrupt source;

1= Enable interrupt source.

Bit	R/W	Default	Description
31:8	R	0	Reserved
7	RW	0	rxsense_fall
6	RW	0	rxsense_rise
5	RW	0	err_i2c_timeout
4	RW	0	hdcp22_rndnum_err
3	RW	0	nonce_rfrsh_rise
2	RW	0	hpd_fall
1	RW	0	hpd_rise
0	RW	0	TX Controller IP interrupt.

HDMITX_TOP_INTR_STAT

Interrupt status. For each bit of bit[4:0], write 1 to manually set the interrupt bit, read back the interrupt status.

Bit	R/W	Default	Description
31	R	0	Shadowing TX Controller IP interrupt status flag.
30	R	0	Shadowing HDCP2.2 IP interrupt status flag.
29:58	R	0	Reserved
7	RW	0	rxsense_fall
6	RW	0	rxsense_rise
5	RW	0	err_i2c_timeout
4	RW	0	hdcp22_rndnum_err
3	RW	0	nonce_rfrsh_rise
2	RW	0	hpd_fall
1	RW	0	hpd_rise

Bit	R/W	Default	Description
0	RW	0	TX Controller IP interrupt.

HDMITX_TOP_INTR_STAT_CLR

Interrupt status clear. For each bit, write 1 to clear the interrupt bit.

Bit	R/W	Default	Description
31: 58	R	0	Reserved
7	W	0	rxsense_fall
6	W	0	rxsense_rise
5	W	0	err_i2c_timeout
4	W	0	hdcp22_rndnum_err
3	W	0	nonce_rfrsh_rise
2	W	0	hpd_fall
1	W	0	hpd_rise
0	W	0	TX Controller IP interrupt.

HDMITX_TOP_BIST_CNTL

Bit	R/W	Default	Description
31:16	R	0	Reserved
15	RW	0	Reserved
14:12	RW	0	tmds_sel: 3'b000=Output zero; 3'b001=Output normal TMDS data; 3'b010=Output PRBS data; 3'b100=Output shift pattern.
11: 9	RW	0	shift_pttn_repeat: 0=New pattern every clk cycle; 1=New pattern every 2 clk cycles; ...; 7=New pattern every 8 clk cycles.
8	RW	0	shift_pttn_en: 1= Eanble shift pattern generator; 0=Disable.
7:5	RW	0	Reserved
4: 3	RW	0	prbs_pttn_mode:

Bit	R/W	Default	Description
			0=PRBS11; 1=PRBS15; 2=PRBS7; 3=PRBS31.
2:1	RW	0	prbs_pttn_width: 0=Idle; 1=Output 8-bit pattern; 2=Output 1-bit pattern; 3=Output 10-bit pattern.
0	RW	0	prbs_pttn_en: 1=Enable PRBS generator; 0=Disable.

HDMITX_TOP_SHIFT_PTTN_012

Bit	R/W	Default	Description
31:30	R	0	Reserved
29:20	RW	0	shift_pttn_data[59:50].
19:10	RW	0	shift_pttn_data[69:60].
9:0	RW	0	shift_pttn_data[79:70].

HDMITX_TOP_SHIFT_PTTN_345

Bit	R/W	Default	Description
31:30	R	0	Reserved
29:20	RW	0	shift_pttn_data[29:20].
19:10	RW	0	shift_pttn_data[39:30].
9:0	RW	0	shift_pttn_data[49:40].

HDMITX_TOP_SHIFT_PTTN_67

Bit	R/W	Default	Description
31:20	R	0	Reserved
19:10	RW	0	shift_pttn_data[9:0].
9:0	RW	0	shift_pttn_data[19:10].

HDMITX_TOP_TMDS_CLK_PTTN_01

Bit	R/W	Default	Description

31:26	R	0	Reserved
25:16	RW	0	tmds_clk_pttn[19:10].
15:10	R	0	Reserved
9:0	RW	0	tmds_clk_pttn[9:0].

HDMITX_TOP_TMDS_CLK_PTTN_23

Bit	R/W	Default	Description
31:26	R	0	Reserved
25:16	RW	0	tmds_clk_pttn[39:30].
15:10	R	0	Reserved
9:0	RW	0	tmds_clk_pttn[29:20].

HDMITX_TOP_TMDS_CLK_PTTN_CNTL

Bit	R/W	Default	Description
31:2	R	0	Reserved
1	RW	0	shift_tmds_clk_pttn: 1=Enable shifting clk pattern, used when TMDS CLK rate = TMDS character rate /4.
0	W	0	load_tmds_clk_pttn: Write this bit to 1 to load tmds_clk_pttn to HW. Always read back 0.

HDMITX_TOP_REVOCMEM_STAT

Bit	R/W	Default	Description
31:12	R	0	Reserved
1	RW	0	revocmem_rd_fail: Read back 1 to indicate Host read REVOC MEM failure, Reading this register automatically clear the failure flag.
0	RW	0	revocmem_wr_fail: Read back 1 to indicate Host write REVOC MEM failure, Reading this register automatically clear the failure flag.

HDMITX_TOP_STAT0

Bit	R/W	Default	Description
31: 12	R	0	Reserved
1	R	0	filtered RxSense status: 0= RxSense low; 1= RxSense high.

0	R	0	filtered HPD status: 0= HPD low; 1= HPD high.

HDMITX_TOP_SKP_CNTL_STAT

Bit	R/W	Default	Description
31	R	0	Status of nonce_vld signal.
30:4	R	0	Reserved
3	RW	0	rndnum_hdcp22: 0=Randon number generator if enabled for hdcp1.4; 1= Randon number generator if enabled for hdcp2.2.
2	RW	0	DUK_vld:Set to 1 once DUK is written.
1	RW	0	PKF_vld:Set to 1 once PKF is written.
0	RW	0	nonce_hw_en: 1=Use HW nonce; 0=Use SW nonce from reg HDMITX_TOP_NONCE_0/1/2/3.

HDMITX_TOP_NONCE_0

Bit	R/W	Default	Description
31:0	W	0	nonce[31:0]

HDMITX_TOP_NONCE_1

Bit	R/W	Default	Description
31:0	W	0	nonce[63:32]

HDMITX_TOP_NONCE_2

Bit	R/W	Default	Description
31:0	W	0	nonce[95:64]

HDMITX_TOP_NONCE_3

Bit	R/W	Default	Description
31:0	W	0	nonce[127:96]

HDMITX_TOP_PKF_0

Bit	R/W	Default	Description
31:0	W	0	PKF[31:0]

HDMITX_TOP_PKF_1

Bit	R/W	Default	Description
31:0	W	0	PKF[63:32]

HDMITX_TOP_PKF_2

Bit	R/W	Default	Description
31:0	W	0	PKF[95:64]

HDMITX_TOP_PKF_3

Bit	R/W	Default	Description
31:0	W	0	PKF[127:96]

HDMITX_TOP_DUK_0

Bit	R/W	Default	Description
31:0	W	0	DUK[31:0]

HDMITX_TOP_ DUK_1

Bit	R/W	Default	Description
31:0	W	0	DUK[63:32]

HDMITX_TOP_ DUK_2

Bit	R/W	Default	Description
31:0	W	0	DUK[95:64]

HDMITX_TOP_ DUK_3

Bit	R/W	Default	Description
31:0	W	0	DUK[127:96]

HDMITX_TOP_INFILTER

Bit	R/W	Default	Description
31:27	RW	0	Reserved
26:24	RW	0	For DDC infilter: filter internal clock divider. 0=No divide; 1=Divide by 2; 2=Divide by 3; … 7=Divide by 8.

23:16	RW	0	For DDC infilter: sampling clock divider. 0=No divide; 1=Divide the filter sampling clock by 2; 2=Divide the filter sampling clock by 3; … 255=Divide the filter sampling clock by 256;
15:11	RW	0	Reserved

HDMITX_TOP_NSEC_SCRATCH

Bit	R/W	Default	Description
31:0	RW	0	Scratch register that can be used for either secure or non-secure reg access.

HDMITX_TOP_SEC_SCRATCH

Bit	R/W	Default	Description
31:0	RW	0	Scratch register that can be used for secure reg access only.

HDMITX_TOP_EMP_CNTL0

Bit	R/W	Default	Description
31:16	RW	0	Emppacketsinframe. Number of EMPs to send for the next frame.
15:8	RW	0	Empstartlatency. Defines the number of lines to wait after End-Of-Field, to start sending EMP.
0	RW	0	emp_tx_en. 0=Disable Extended Metadata packet. 1=Enable Extended Metadata packet.

HDMITX_TOP_EMP_CNTL1

Bit	R/W	Default	Description
31:19	R	0	Reserved.
18:17	RW	0	emp_endian[1:0]. Bit[0]: 1=EMP data are stored as little Endian in DDR per 64-bit. 0=Big endian. Bit[1]: 1=For each 128-bit, swap high 64-bit to low 64-bit, and low 64-bit to high 64-bit. 0=No swap.

16	RW	0	emp_autoclr_ar_pending. If at the start of DDR read request for the current frame, there is still outstanding requests remaining from the previous frame, this bit defines whether to auto-clear DDR request state machine. 0=The current request will not start until the previous outstanding requests are cleared by normal operation. 1=The state machine is reset and previous outstanding requests are cancelled. The current request starts immediately.
15:0	RW	128	emp_sampleline. All EMP control registers are buffered internally, at the time defined by this field – the number of lines after Vsync rise. Before this time the EMP control registers does not take effect.

HDMITX_TOP_EMP_MEMADDR_START

Bit	R/W	Default	Description
31:0	RW	0	Start address of EMP memory pointer for next frame.

HDMITX_TOP_EMP_STAT0

Bit	R/W	Default	Description
31:30	RW	0	emp_err[1:0]. Error status of AXI activity. Write 1 to each bit to clear. Bit[0]: 1=AXI RID incorrect. Bit[1]: 1=AXI RRESP error.
29:27	R	0	Reserved.
26:24	R	0	emp_ar_state[2:0]: Status of DDR AXI Address Read state machine. 0=IDLE; 1=WAIT_FIFO_ROOM; 2=WAIT_ARREADY; 3=WAIT_PENDING; 4=DONE.
23:17	R	0	emp_fifo_count[6:0]. FIFO fill level for storing 128-bit DDR data, the result of reading EMP data from DDR.
16:0	R	0	emp_ar_pending. Number of DDR reads that are still outstanding, meaning request has been sent, but Data have yet to be received.

HDMITX_TOP_EMP_STAT1

Bit	R/W	Default	Description
31:16	R	0	empdone_cnt_buf [15:0]: Buffered number of EMP sent.
15:0	R	0	empdone_cnt[15:0]: Current number of EMP sent.

HDMITX_TOP_AXI_ASYNC_CNTL0

Bit	R/W	Default	Description
31:16	R	0	Reserved.
15:8	RW	128	axi_async_waiting_limit[7:0].
7:5	R	0	Reserved.
4:3	RW	0	axi_urgent[1:0]: DDR request urgent level.
2	RW	0	axi_async_disable_clk. 1=Disable clk to AXI-async module. 0=Enable clk.
1	RW	1	axi_async_auto_gclk_en. 1=Auto clk gate AXI-async module. 0=Free-run clk.
0	RW	1	axi_async_req_en. 1=Enable AXI async interface between EMP's tmds_clk domain to AXI's esm_clk domain. If need to sent EMP, this bit must be enabled. 0=Disable.

HDMITX_TOP_AXI_ASYNC_CNTL1

Bit	R/W	Default	Description
31:0	RW	32'h18101810	axi_async_hold_num.

HDMITX_TOP_AXI_ASYNC_STAT0

Bit	R/W	Default	Description
31:1	R	0	Reserved.
0	RW	1	axi_async_chan_idle.

HDMITX_TOP_I2C_BUSY_CNT_MAX

Bit	R/W	Default	Description
31:0	RW	0xffffffff	i2c_busy_cnt_max. After I2C Start bit, if I2C bus is static for more than i2c_busy_cnt_max number of cycles, an error interrupt will happen.

HDMITX_TOP_I2C_BUSY_CNT_STAT

Bit	R/W	Default	Description
31:0	R	0	i2c_busy_cnt. Reflect the number of idle cycles after the I2C Start bit, for the latest I2C transaction.

HDMITX_TOP_HDCP22_BSOD

Bit	R/W	Default	Description
31:26	R	0	Reserved.
25	RW	0	hdcp22_no_bsod. 1=Do not use BSOD to override, no matter from FW or SW. 0=Use the BSOD, original behaviour.
24	RW	0	hdcp22_bsod_override. 1=Overide BSOD value with hdcp_bsod_val. 0=Use the original BSOD value from ESM FW.
23:0	RW	0x800080	hdcp22_bsod_val.

HDMITX_TOP_DDC_CNTL

Bit	R/W	Default	Description
31:2	R	0	Reserved.
1	RW	1	DSDA pull down. 0=Pull down DSDA. 1=No pull down.
0	RW	1	DSCL pull down. 0=Pull down DSCL. 1=No pull down.

Controller Register Updates

The following are the additional updates added to the original Controller. Please read the following in combination with the original databook of the Controller.

Note: DRM stands for Dynamic Range and Mastering InfoFrame.

Added a_hdcpcfg1 bit[5] and hdcp22reg_ctrl bit[7:6] to optional balance path delays between HDCP22 path, HDCP14 path and non-HDCP path.

Add EMP related register bits.

Add I2C reset.

ih_fc_stat2 0x102

Bits	Name	Memory Access	Description
3	EMP	RW	Active after successful transmission of a total of N packets through DDR-EMP interface. N is programmed by HDMITX_TOP_EMP_CNTL0[31:16]. Value After Reset: 0
2	DRM	RW	Active after successful transmission of an DRM InfoFrame packet. Value After Reset: 0

ih_mute_fc_stat2 0x182

Bits	Name	MemoryAccess	Description
3	EMP	RW	When set to 1, mutes ih_fc_stat2[3]. Value After Reset: 1
2	DRM	RW	When set to 1, mutes ih_fc_stat2[2]. Value After Reset: 1

fc_datauto3 0x10b7

Bits	Name	MemoryAccess	Description
6	DRM_auto	RW	Enables DRM packet insertion Value After Reset: 1

fc_rdrb12 0x10c4

Bits	Name	MemoryAccess	Description
7:4	Reserved.		
3:0	DRMframeinterpolation	RW	DRM frame interpolation Value After Reset: 0

fc_rdrb13 0x10c5

Bits	Name	Memory Access	Description
7:4	DRMpacketsinframe	RW	DRM packets per frame Value After Reset: 0
3:0	DRMpacketlinespacing	RW	DRM packets line spacing Value After Reset: 0

fc_mask2 0x10da

Bits	Name	Memory Access	Description
3	EMP	RW	Mask bit for FC_INT2.EMP interrupt bit Value After Reset: 1
2	DRM	RW	Mask bit for FC_INT2.DRM interrupt bit Value After Reset: 1

fc_packet_tx_en 0x10e3

Bits	Name	Memory Access	Description

7	DRM_tx_en	RW	DRM transmission control 1: Transmission enabled 0: Transmission disabled Value After Reset: 0

fc_drm_hb01 0x1168

Bits	Name	Memory Access	Description
7:0	fc_drm_hb0	RW	Frame composer DRM Packet Header Register 1 Value After Reset: 0

fc_drm_hb02 0x1169

Bits	Name	Memory Access	Description
7:0	fc_drm_hb1	RW	Frame composer DRM Packet Header Register 2 Value After Reset: 0

fc_drm_pb[0:26] 0x116a+(i*0x1)

Bits	Name	Memory Access	Description
7:0	fc_drm_pb	RW	Frame composer DRM Packet Body Register Array Value After Reset: 0

fc_dbgforce 0x1200

Bits	Name	Memory Access	Description
7	forcewoo	RW	Per SolvNet case8000767326: 0=Use bug fix method suggested in case8000767326. 1=Revert to original RTL. Not recommended. Value After Reset: 0

a_hdcpcfg1 0x5001

Bits	Name	Memory Access	Description
6	hdcp14_no_short_read	RW	1=Read Ri' with explicit address=0x08. 0=Original IP behaviour -- short-read. Value After Reset: 0
5	hdcp14_delmatch	RW	1=Match path delays between HDCP14 path and non-HDCP path. 0=Original IP behaviour. Value After Reset: 0

hdcp22reg_id 0x7900

Bits	Name	Memory Access	Description

7	hdcp22_not_capable	R	1=HDCP22 Not capable. Value After Reset: 0

hdcp22reg_ctrl 0x7904

Bits	Name	Memory Access	Description
7	hdcp_byp_delmatch	RW	1=Match path delays between HDCP path and non-HDCP path. 0=Original IP behaviour. Value After Reset: 0
6	hdcp22_14_delmatch	RW	1=Match path delays between HDCP22 path and HDCP14 path. 0=Original IP behaviour. Value After Reset: 0

hdcp22reg_sts 0x7908

Bits	Name	Memory Access	Description
7	Hdcp22_capable	R	1=HDCP22 capable. Value After Reset: 0
6	hdcp22_auth_lost	R	1=HDCP22 authentication lost. Value After Reset: 0
5	hdcp22_authenticated	R	1=HDCP22 authenticated. Value After Reset: 0
4	hdcp22_auth_fail	R	1=HDCP22 authentication failed. Value After Reset: 0

i2cm_softrstz 0x7e09

Bits	Name	Memory Access	Description
1	softrst_idle_cnt_en	RW	Active by writing a zero and auto cleared to one in the following cycle. Do this to re-enable IDLE counter again. The counter maybe frozen due to some mis-operation from RX on the I2C bus. Value After Reset: 1

10.2.3.45 MIPI_DSI(base:32'hFF644000)

Each register final address = BASE + address * 4

MIPI_DSI_PHY_CTRL[31:0] 0x00

Bit(s)	R/W	Default	Description
31	R/W	0	soft reset for the phy. 1 = reset. 0 = dessert the reset.
30	R/W	0	clock lane soft reset.

29	R/W	0	data byte lane 3 soft reset.
28	R/W	0	data byte lane 2 soft reset.
27	R/W	0	data byte lane 1 soft reset.
26	R/W	0	data byte lane 0 soft reset.
5	R/W	0	LPDT data endian. 1 = transfer the high bit first. 0 : transfer the low bit first.
4	R/W	0	HS data endian.
3	R/W	0	force data byte lane in stop mode.
2	R/W	0	force data byte lane 0 in reciever mode.
1	R/W	0	write 1 to sync the txclkesc input. the internal logic have to use txclkesc to decide Txvalid and Txready.
0	R/W	0	enalbe the MIPI DSI PHY TxDDRClk.

MIPI_DSI_PHY_CTRL[31:0] 0x01

Bit(s)	R/W	Default	Description
31	R/W	0	clk lane tx_hs_en control selection. 1 = from register. 0 use clk lane state machine.
30	R/W	0	register bit for clock lane tx_hs_en.
29	R/W	0	clk lane tx_lp_en contrl selection. 1 = from register. 0 from clk lane state machine.
28	R/W	0	register bit for clock lane tx_lp_en.
27	R/W	0	chan0 tx_hs_en control selection. 1 = from register. 0 from chan0 state machine.
26	R/W	0	register bit for chan0 tx_hs_en.
25	R/W	0	chan0 tx_lp_en control selection. 1 = from register. 0 from chan0 state machine.
24	R/W	0	register bit from chan0 tx_lp_en.
23	R/W	0	chan0 rx_lp_en control selection. 1 = from register. 0 from chan0 state machine.
22	R/W	0	register bit from chan0 rx_lp_en.
21	R/W	0	chan0 contention detection enable control selection. 1 = from register. 0 from chan0 state machine.
20	R/W	0	register bit from chan0 contention dectection enable.
19	R/W	0	chan1 tx_hs_en control selection. 1 = from register. 0 from chan0 state machine.
18	R/W	0	register bit for chan1 tx_hs_en.
17	R/W	0	chan1 tx_lp_en control selection. 1 = from register. 0 from chan0 state machine.
16	R/W	0	register bit from chan1 tx_lp_en.
15	R/W	0	chan2 tx_hs_en control selection. 1 = from register. 0 from chan0 state machine.

Bit(s)	R/W	Default	Description
14	R/W	0	register bit for chan2 tx_hs_en.
13	R/W	0	chan2 tx_lp_en control selection. 1 = from register. 0 from chan0 state machine.
12	R/W	0	register bit from chan2 tx_lp_en.
11	R/W	0	chan3 tx_hs_en control selection. 1 = from register. 0 from chan0 state machine.
10	R/W	0	register bit for chan3 tx_hs_en.
9	R/W	0	chan3 tx_lp_en control selection. 1 = from register. 0 from chan0 state machine.
8	R/W	0	register bit from chan3 tx_lp_en.
4	R/W	0	clk chan power down. this bit is also used as the power down of the whole MIPI_DSI_PHY.
3	R/W	0	chan3 power down.
2	R/W	0	chan2 power down.
1	R/W	0	chan1 power down.
0	R/W	0	chan0 power down.

MIPI_DSI_CHAN_STS [31:0] 0x02

Bit(s)	R/W	Default	Description
24	R/W	0	chan0 TX->RX turn can't accept the ACK command from slave watch dog triggered. write 1 to clear the status bit.
23	R/W	0	chan0 RX ESC command watch dog triggered. write 1 to clean this bit.

MIPI_DSI_CLK_TIM [31:0] 0x03

Bit(s)	R/W	Default	Description
31:24	R/W	0	TCLK_PREPARE.
23:16	R/W	0	TCLK_ZERO.
15:8	R/W	0	TCLK_POST.
7:0	R/W	0	TCLK_TRAIL.

MIPI_DSI_HS_TIM [31:0] 0x04

Bit(s)	R/W	Default	Description
31:24	R/W	0	THS_PREPARE.
23:16	R/W	0	THS_ZERO.
15:8	R/W	0	THS_TRAIL.
7:0	R/W	0	THS_EXIT.

MIPI_DSI_LP_TIM [31:0] 0x05

Bit(s)	R/W	Default	Description
31:24	R/W	0	tTA_GET.
23:16	R/W	0	tTA_GO.
15:8	R/W	0	tTA_SURE.
7:0	R/W	0	tLPX.

MIPI_DSI_ANA_UP_TIM [31:0] 0x06

Bit(s)	R/W	Default	Description
31:0	R/W	0	wait time to MIPI DIS analog ready.

MIPI_DSI_INIT_TIM [31:0] 0x07

Bit(s)	R/W	Default	Description
31:0	R/W	0	TINIT

MIPI_DSI_WAKEUP_TIM [31:0] 0x08

Bit(s)	R/W	Default	Description
31:0	R/W	0	TWAKEUP

MIPI_DSI_LPOK_TIM [31:0] 0x09

Bit(s)	R/W	Default	Description
31:0	R/W	0	when in RxULPS state, RX reciever is in sleep mode. every MIPI_DSI_ULPS_CHECK period, the reciever would be enabled once, and waiting this timer period to get the stable input.

MIPI_DSI_LP_WCHDOG [31:0] 0x0A

Bit(s)	R/W	Default	Description
31:0	R/W	0	watch dog timer for MIPI DSI LP receive state

MIPI_DSI_ANA_CTRL [31:0] 0x0B

Bit(s)	R/W	Default	Description
31:0	R/W	0	tMBIAS. timer to wait for analog mBIAS voltage stable.

MIPI_DSI_CLK_TIM1 [31:0] 0x0C

Bit(s)	R/W	Default	Description
7:0	R/W	0	tCLK_PRE

MIPI_DSI_TURN_WCHDOG [31:0] 0x0D

Bit(s)	R/W	Default	Description
7:0	R/W	0	watch dog timer for lane 0 LP turn around waiting time.

MIPI_DSI_ULPS_CHECK [31:0] 0x0E

Bit(s)	R/W	Default	Description
31:0	R/W	0	when Lane0 in LP recieve state, if the another side sent Low power command, using this timer to enable Tcheck the another size wakeup nor not

MIPI_DSI_TEST_CTRL0 [31:0] 0x0F

Bit(s)	R/W	Default	Description
31	R	0	lp_rx_ch0p，read from analog
30	R	0	lp_rx_ch0n，read from analog
29	R	0	lp_cd_ch0p，read from analog
28	R	0	lp_cd_ch0n，read from analog
27	R/W	0	test_en，all test function enable
26	R/W	0	prbs_en，1: test_data = prbs; 0 : test_const_value;
25	R/W	0	hs_const_value，hs ch test value
24	R/W	0	lp_p_const_value，lp ch p test value
23	R/W	0	lp_n_const_value，lp ch n test value
22	R/W	0	lp_rx_ch0p，read from analog
21	R/W	0	lp_rx_ch0n，read from analog
19:16	R/W	0	prbs_down_sample(LP only)
11	R/W	0	lp_cd_swap ,swap lp cd P/N
10	R/W	0	lp_rx_swap ,swap lp rx P/N
9	R/W	0	lp_clk_swap,swap lp clk P/N
8	R/W	0	lp_ch3_swap,swap lp ch3 P/N
7	R/W	0	lp_ch2_swap,swap lp ch2 P/N
6	R/W	0	lp_ch1_swap,swap lp ch1 P/N
5	R/W	0	lp_ch0_swap,swap lp ch0 P/N
4	R/W	0	hs_clk_inv ,inv(swap) hs clk
3	R/W	0	hs_ch3_inv ,inv(swap) hs ch3

Bit(s)	R/W	Default	Description
2	R/W	0	hs_ch2_inv ,inv(swap) hs ch2
1	R/W	0	hs_ch1_inv ,inv(swap) hs ch1
0	R/W	0	hs_ch0_inv ,inv(swap) hs ch0

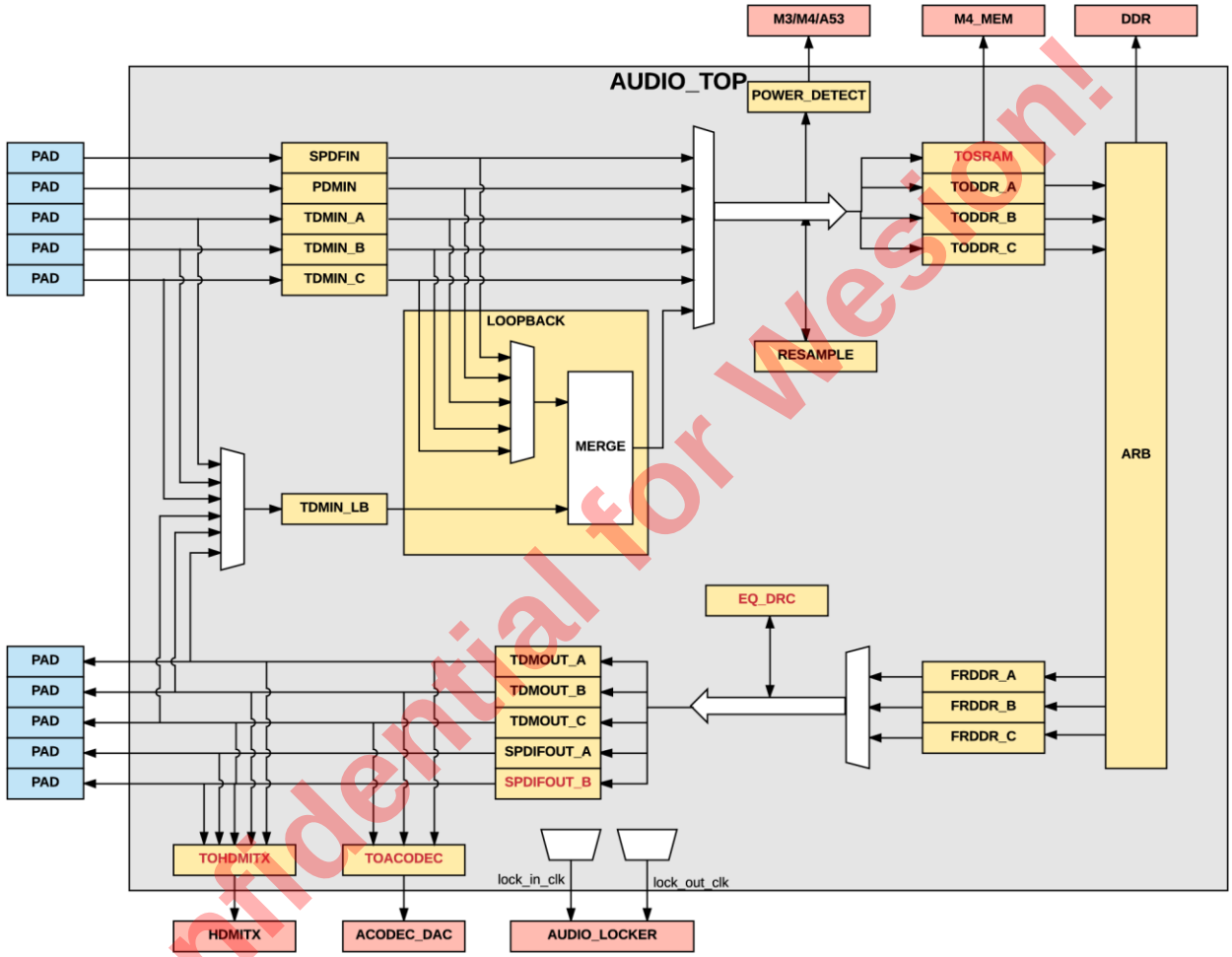
MIPI_DSI_TEST_CTRL1 [31:0] 0x10

Bit(s)	R/W	Default	Description
31:24	R/W	0	trail offset
21	R/W	0	lp_ch0_cd_en, the value when lp_en_const_en = 1
20	R/W	0	lp_ch0_rx_en, the value when lp_en_const_en = 1
19	R/W	0	lp_clk_data_en,1: test_data; 0: original data;
18	R/W	0	lp_clk_en, the value when lp_en_const_en = 1
17	R/W	0	lp_ch3_data_en,1: test_data; 0: original data;
16	R/W	0	lp_ch3_en, the value when lp_en_const_en = 1
15	R/W	0	lp_ch2_data_en,1: test_data; 0: original data;
14	R/W	0	lp_ch2_en, the value when lp_en_const_en = 1
13	R/W	0	lp_ch1_data_en,1: test_data; 0: original data;
12	R/W	0	lp_ch1_en, the value when lp_en_const_en = 1
11	R/W	0	lp_ch0_data_en,1: test_data; 0: original data;
10	R/W	0	lp_ch0_en, the value when lp_en_const_en = 1
9	R/W	0	hs_clk_data_en,1: test_data; 0: original data;
8	R/W	0	hs_clk_en, the value when hs_en_const_en = 1
7	R/W	0	hs_ch3_data_en,1: test_data; 0: original data;
6	R/W	0	hs_ch3_en, the value when hs_en_const_en = 1
5	R/W	0	hs_ch2_data_en,1: test_data; 0: original data;
4	R/W	0	hs_ch2_en, the value when hs_en_const_en = 1
3	R/W	0	hs_ch1_data_en,1: test_data; 0: original data;
2	R/W	0	hs_ch1_en, the value when hs_en_const_en = 1
1	R/W	0	hs_ch0_data_en,1: test_data; 0: original data;
0	R/W	0	hs_ch0_en, the value when hs_en_const_en = 1

11. Audio Path

S922X integrates 3 TDM input/output interface, 1 SPDIF input, 2 SPDIF output interface, 1 PDM interface upto 8 channels, 3 TODDR (FIFO) for transfer input data to DDR, 1 TOSRAM for transfer input data to M4 MEM, 3 FRDDR (FIFO) for transfer data from DDR to output, 1 TDM LB & Loopback for AEC, 1 HW resample for clock synchronization, 1 power detect for voice wake up, and 1 clock locker detect difference of two clock. Below is the diagram for S922X audio path.

Figure 11-1 Diagram of Audio Path



11.1 Audio_Input

11.1.1 Overview

This section describes TDM input interface, SPDIF input interface and PDM input interface..

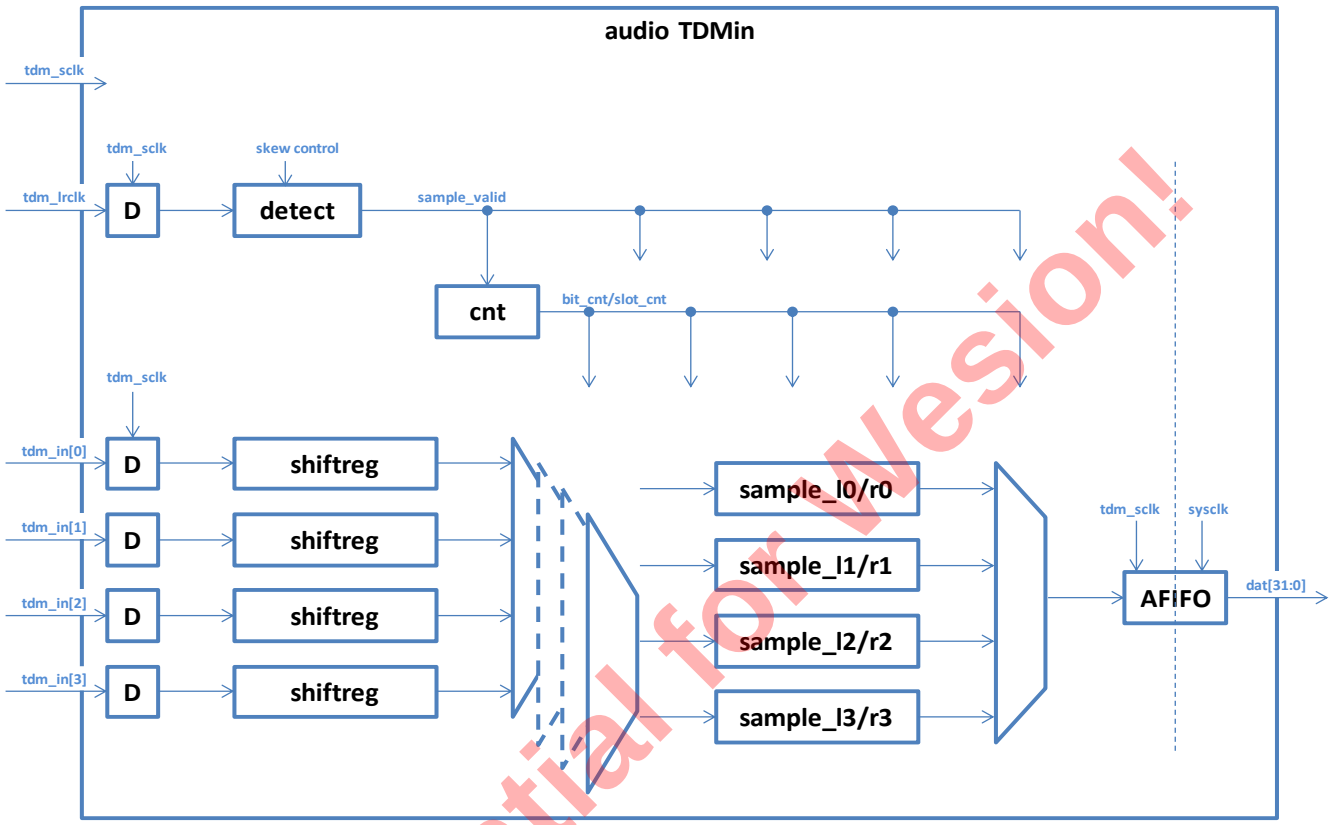
11.1.2 TDM Input Interface

TDM input interface works in the following way:

1. All worked at tdm_sclk before AFIFO,
2. Output is worked at sysclk;
3. Add skew control between lrclk and data;
4. Detect rise or fall edge of lrclk and clear bit_cnt/slot_cnt;

5. Store 4 data in to 4 shift register (max 32bits) ;
6. Swap 4 shift register to 8 sample data;
7. Send 8 sample data out in serials by mask/mute configure;
8. Below is the diagram of TDM input interface.

Figure 11-2 Diagram of SPIDF



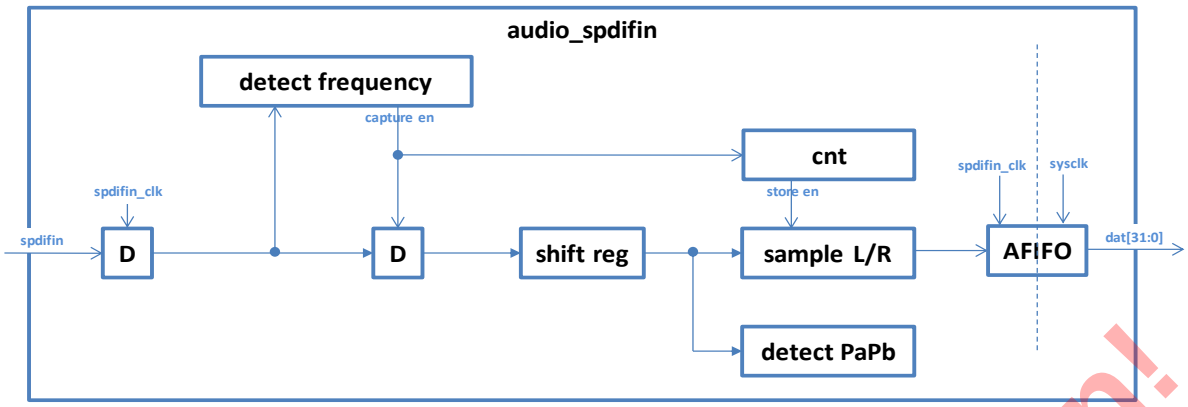
11.1.3 SPDIF Interface

SPDIF input interface works in the following way:

1. All worked at spdifin_clk before AFIFO;
2. After AFIFO worked at sysclk;
3. Detect frequency by rise or fall edge;
4. Capture data to shift reg;
5. Store data to sample L channel or R channel;
6. Detect PaPb if it's IEC60937;

Below is the diagram of SPDIF input interface.

Figure 11-3 SPDIF Input Interface



11.1.4 PDM

PDM input Decimation Filter is a highly programmable multistage decimation filter. It supports 4 inputs or 8 channel PDM digtal micphone interface. Each channel contains one 9 stage CIC filter, 3 low power filters and a high pass filter. The PDM input bit rate is calculate with fs*OSR. OSR can be 64, 128, 192 or 256.

CIC filter may have 3 to 9 CIC stages and the downsample rate also can be programable. There is a multiplier and shifter to adjust the result to match the accuarcy and CIC bit width.

There are 3 low pass filters which shared a 336x24 Coeffient memory and a 336x28 data memory. That means these 3 filters can be configed upto 336 taps together. The coeffient memory need to be programmed through APB bus. For example, if the filter 1 contains 140 taps, filter2 contains 32 taps, filter 3 contain 146 steps, the filter 1 coefficient will use 0~139 of the coeffient memory address; the filter 2 coefficient will use 140~171 of coefficient memory address and the filter 3 coefficient will use 172~317 of the coeffient memory address. The fillter stage controller will based on the downsample rate, filter taps and the rounding mode of each filter to arrange the filter.

The final high pass filter is used to filter the DC curent.

Figure 11-4 PDM Decimation Filter

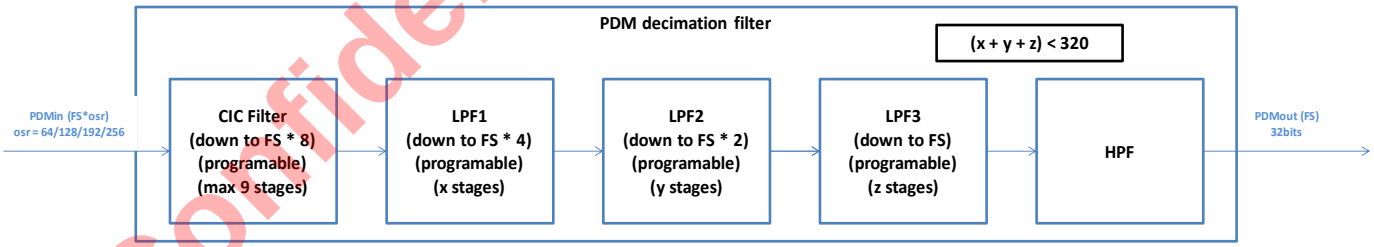
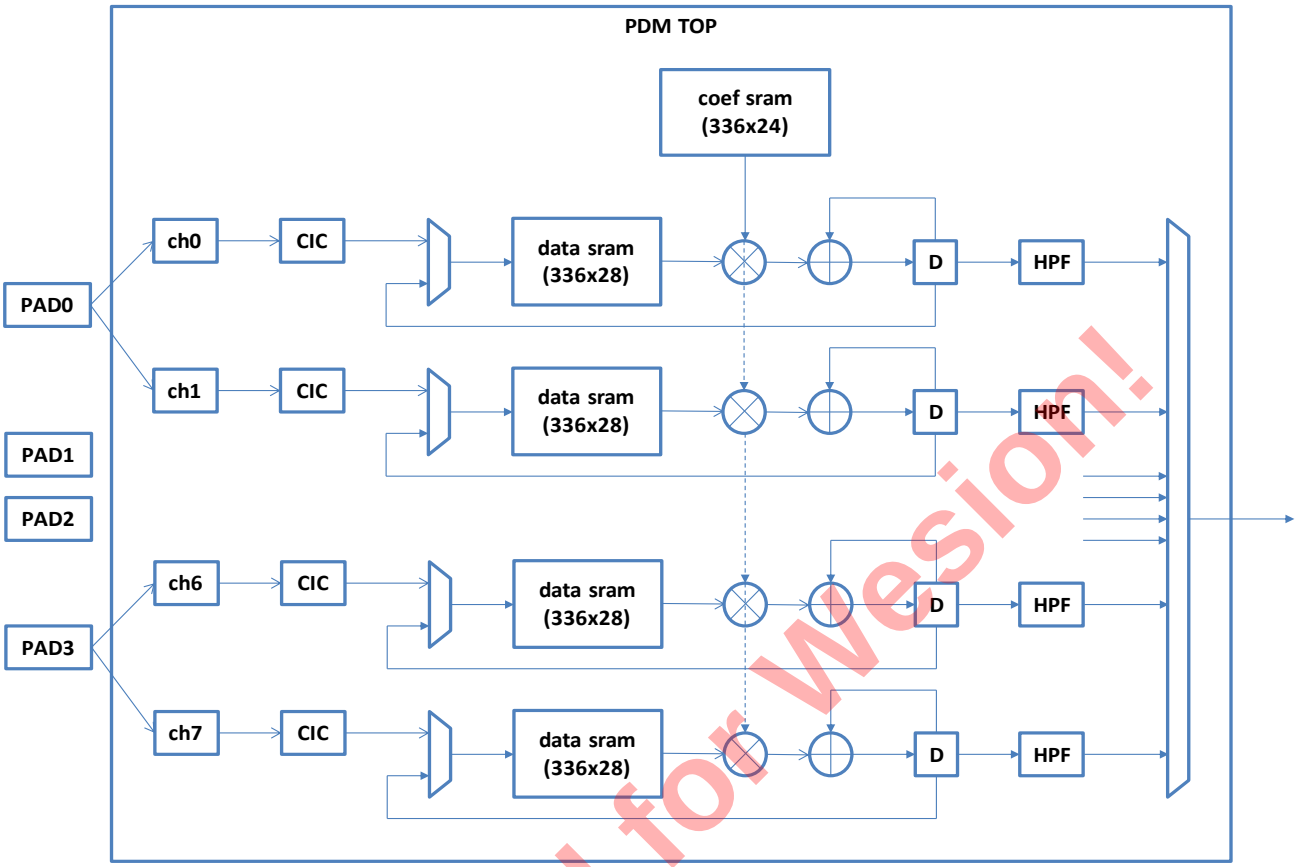


Figure 11-5 PDM Structure



11.2 Audio Output

11.2.1 Overview

This part describes TDM output interface, SPDIF output interface and audioreq_frddr submodule.

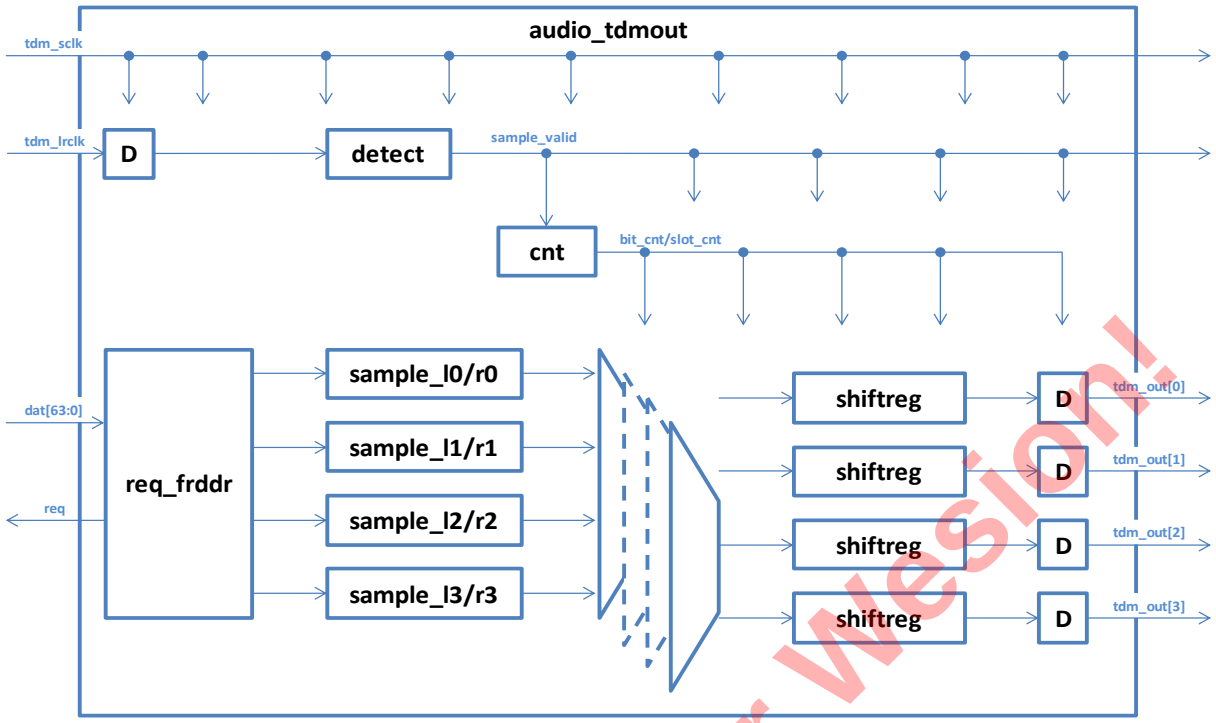
11.2.2 TDM Output Interface

TDM output interface works in the following way:

1. All worked at tdm_sclk;
2. Detect sample valid by tdm_lrclk rise edge and clear bit_cnt/slot_cnt;
3. Request data from FRDDR and store to 8 sample register;
4. Swap 8 sample register;
5. Shift send out data;

Below is the diagram of TDM output interface.

Figure 11-6 TDM Output Interface



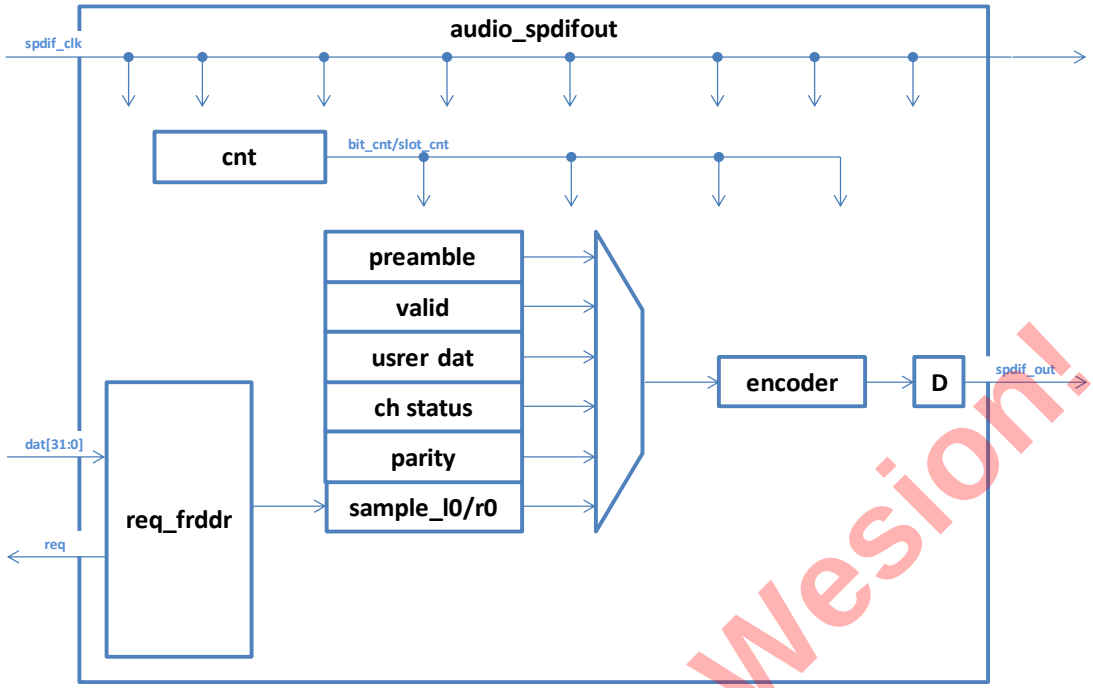
11.2.3 SPDIF Output Interface

The SPDIF output interface works in the following way:

1. All worked at spdifout clk;
2. Request data from FRDDR and store to sample L0/R0;
3. Select data by bit_cnt/slot_cnt and send out;

Below is the diagram of SPDIF output interface.

Figure 11-7 SPDIF Output Interface



The SPDIF(encode) add same source select.The SPDIF out can be the same with i2s0/1/2/3,and can select before or after eq/drc.

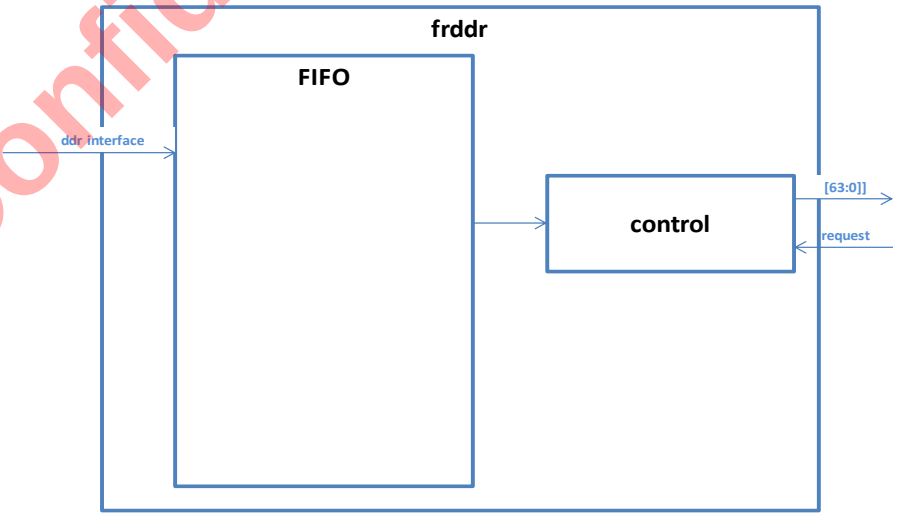
11.2.4 Audioreq_FRDDR

Audioreq_Frddr transfers data from sys clock to out clock and change format in the following way;

1. Wait for OUT enable and FRDDR initial done;
2. Request first time and fill 8 sample;
3. When received update data , it will send request to FRDDR and update 8 sample;

Below is a diagram of Audioreq_Frddr

Figure 11-8 Audioreq_Frddr



11.3 DDR Datapath

11.3.1 Overview

This part describes the datapath between audio module and DDR.

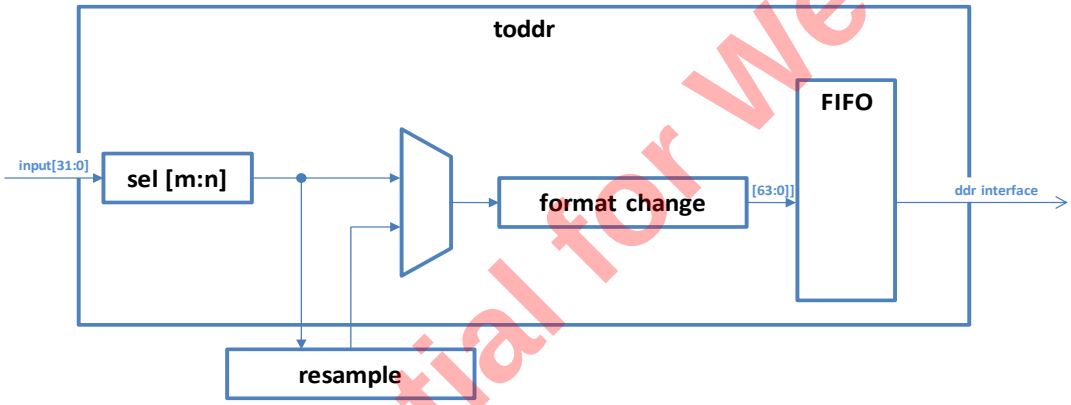
11.3.2 Audio TODDR

S922X has 3 TODDR(FIFO), TODDR_A's FIFO depth is 256x64; B/C are 128x64. TODDR module works in the following way:

1. All worked at sysclk;
2. Resample if need;
3. Change format and package to 64 bits data by configure;
4. Write to fifo;
5. Read data from fifo and send to DDR automatically by configure;

Below is the Diagram of Audio TODDR.

Figure 11-9 Audio TODDR



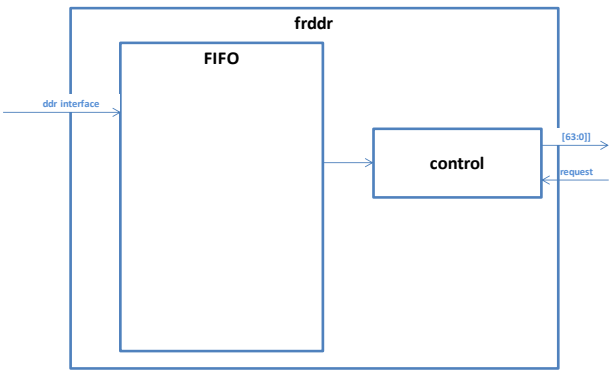
11.3.3 Audio FRDDR

S922X has 3 FRDDR(FIFO), FRDDR_A's FIFO depth is 256x64, B/C are 128x64;

1. All worked at sysclk;
2. When enable FRDDR, it will fill FIFO from DDR first;
3. When FRDDR receive request, it will read data from fifo and send out;
4. FRDDR will fill FIFO automatically by configure;

Below is the Diagram of Audio FRDDR.

Figure 11-10 Audio FRDDR



11.4 Audio TORAM

TORAM is similar as TODDR, just use RAM interface.

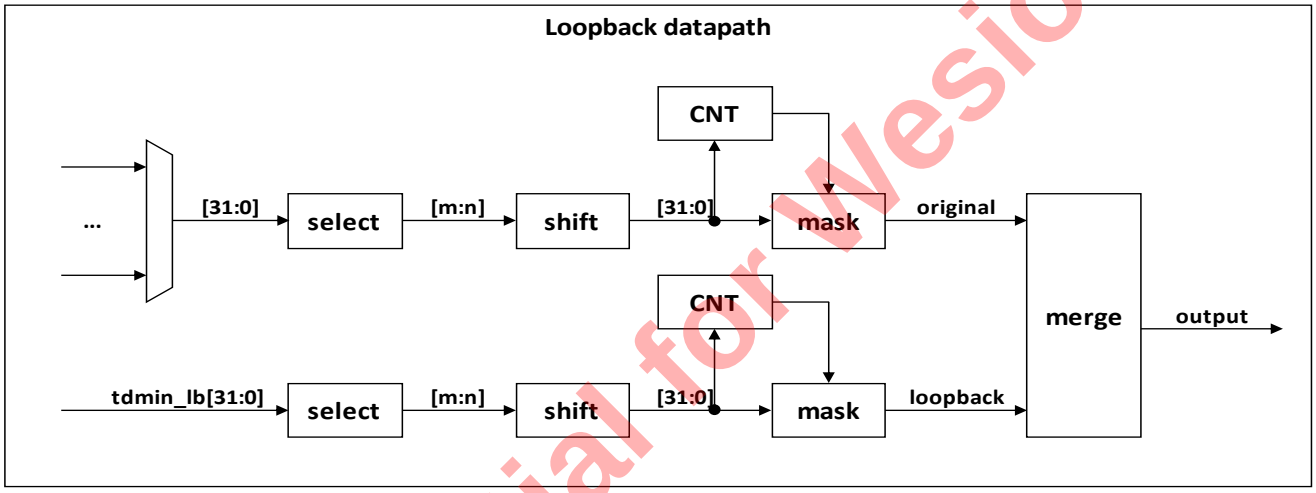
11.5 Audio Loopback

TDMIN_LB can receive one TDMOUT or TDMIN, it can merger with one TDMIN/SPDIFIN/PDMIN in the following way:

1. Store one source to temp register;
2. When another source arrived, send out direct;
3. When finished, send out temp register;

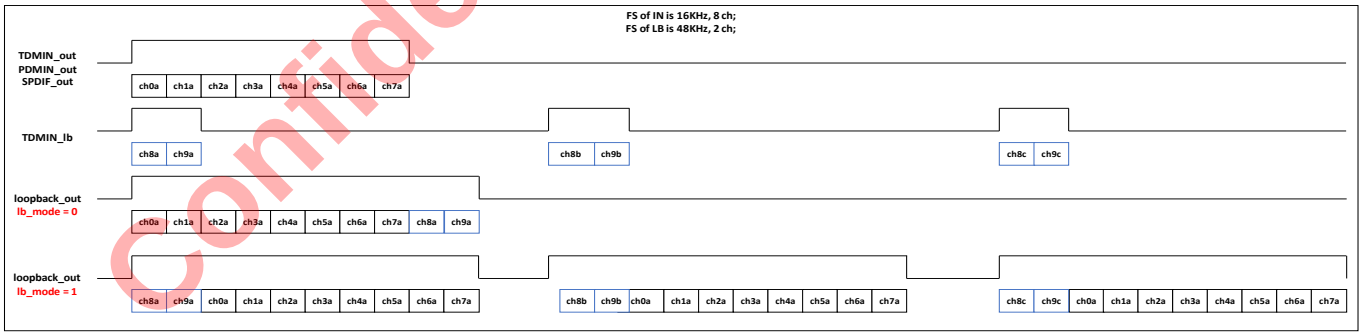
Below is the diagram of audio loopback datapath.

Figure 11-11 Audio Loopback Data Path



Audio loopback wave form are shown below.

Figure 11-12 Audio Loopback Wave Form

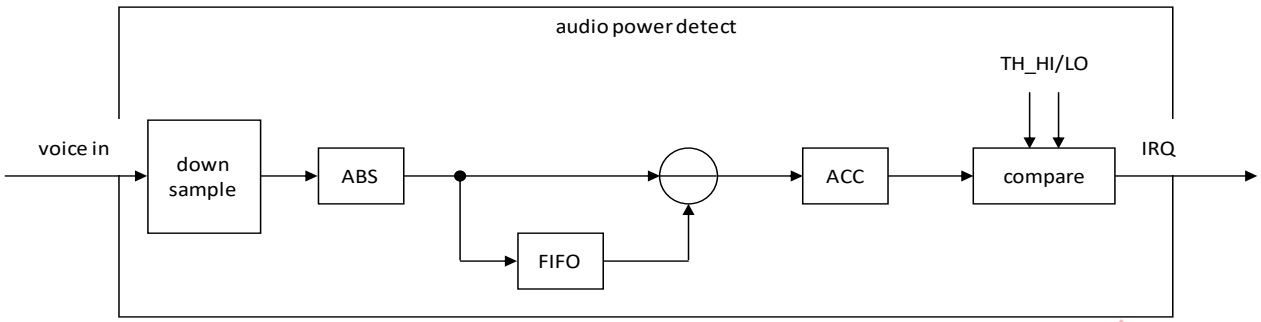


11.6 Audio Power Detect

Audio Power Detect submodule calculate a window of voice and compares it to TH_HI/LO. It generates IRQ when status changes.

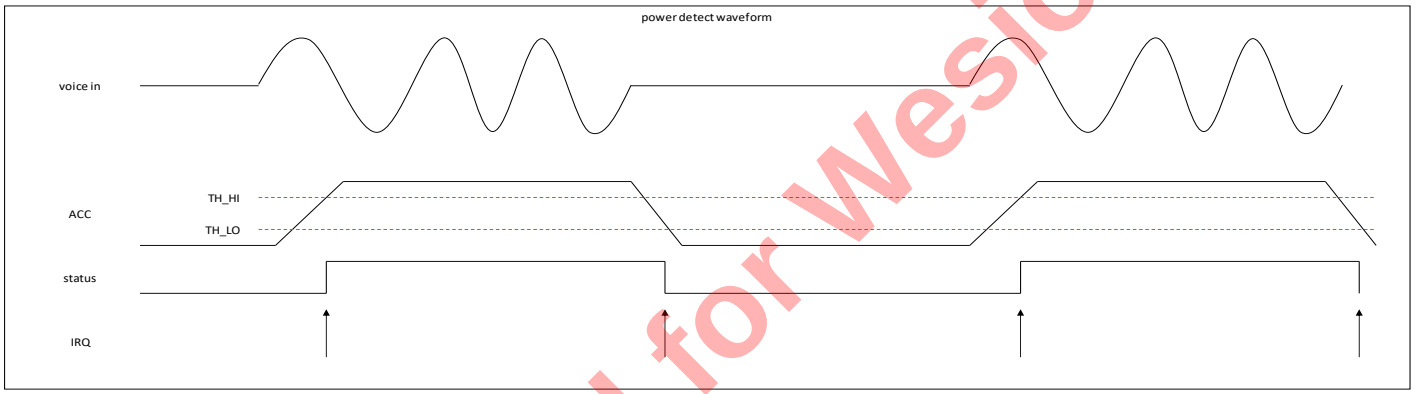
Below is the diagram of audio power detect data path.

Figure 11-13 Audio Power Detect Data Path



Audio power detect wave form is shown below:

Figure 11-14 Audio Power Detect Wave Form

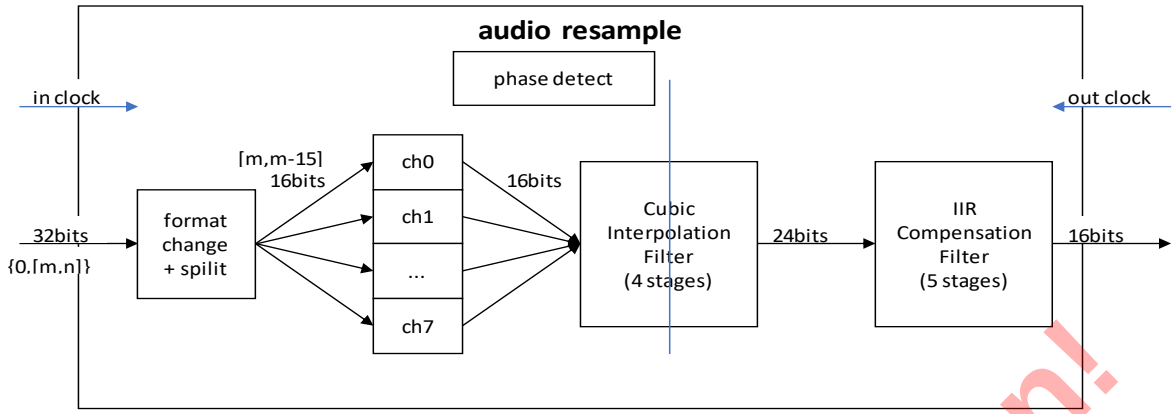


11.7 Audio Resample

Audio resample changes data from one symbol rate to another symbol rate in the following step:

1. Input is from Toddr and selected by toddr's m/n register;
2. ExampleA: toddr_m = 27, toddr_n=4, resample_in[31:0] = {8'd0, toddr_in[27:4]};
3. Cut or extend input from 32bits to 16bits;
4. By exampleA, set resample_M = 23, then filter_in = toddr_in[27:12];
5. Split to each channel (max is 8ch);
6. Interpolation by a cubic filter;
7. Use an IIR filter compensate;

Figure 11-15 Audio Resample

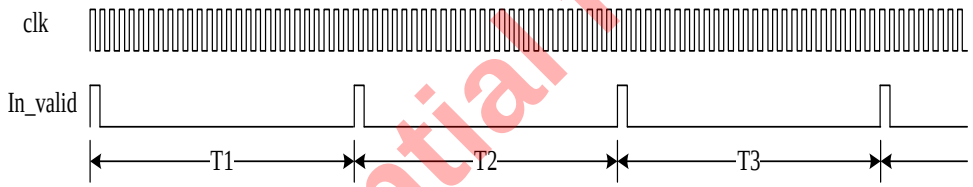


In the diagram, the module phase_det is used to calculate the interpolation phase, the module interp is used to implement the interpolation processing. There are two method which be implemented to calculate the interpolation phase. One is front feedback method, the other is accumulation method. They can be switched by register.

method 0

The method 0 is the direct feedback method. According the out_valid, it interpolates the mapping sample. The operation flow is given below:

Step 1: calculate the input data rate(average the count every input valid signal). For hdmi_rx audio data, the input rate is basically even.



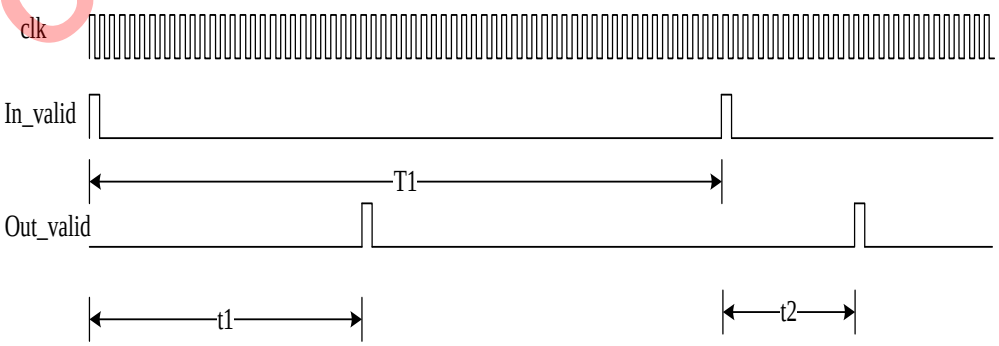
Count the clock number in T1,T2, and so on. get CNT then do average:

T = 0.999*T + 0.001*CNT

For 8 channels signal, the CNT should be count between 8 in_valid signal.

Step 2: in the output clock domain, generate the output valid signal. Because the output clock should be integral times of output frequency, the valid signal can easily generated by counting the output clock. Then change the valid signal to main clock domain.

Step 3: calculate the interpolation phase mu.



Count the clock number in t1 get cnt, then mu can be calculated:

Mu = cnt/T

The bit width of mu is 9 bits.

Step 4: do the interpolation(4 points cubic interpolation)

Basic function is:

Y = ((a*mu + b)*mu + c)*mu + d, which mu is interpolation phase.

and

a = 16*x0 - 48*x1 + 48*x2 - 16*x3

b = 31*x0 - 31*x1 - 31*x2 + 31*x3

c = 23*x0 + 59*x1 - 59*x2 - 23*x3

d = 6*x0 + 58*x1 + 58*x2 + 6*x3

Which x0, x1, x2,x3 is the delay line of input data.

Performance analysis:

The advantage of this method is that the output data rate is generated by output clock and match the acquirement of software absolutely.

The disadvantage is that the interpolation phase has bias. The bias relies on the main clock. More higher the main clock frequency is, more precise the interpolation result is.

Actually the precision of interpolation phase is 9 bits, so the frequency of main clock should be more than 512 times of input data rate.

method 1

The method 1 is accumulation method. It assume the output frequency is accurate and stable. It sets a fixed phase step. The phase step is the ratio of the input frequency to output frequency. Every step, It outputs a sample. About the bias of input frequency, there is a register to report the statistics value. It can be used to adjust the initial phase step by firmware.

Performance analysis:

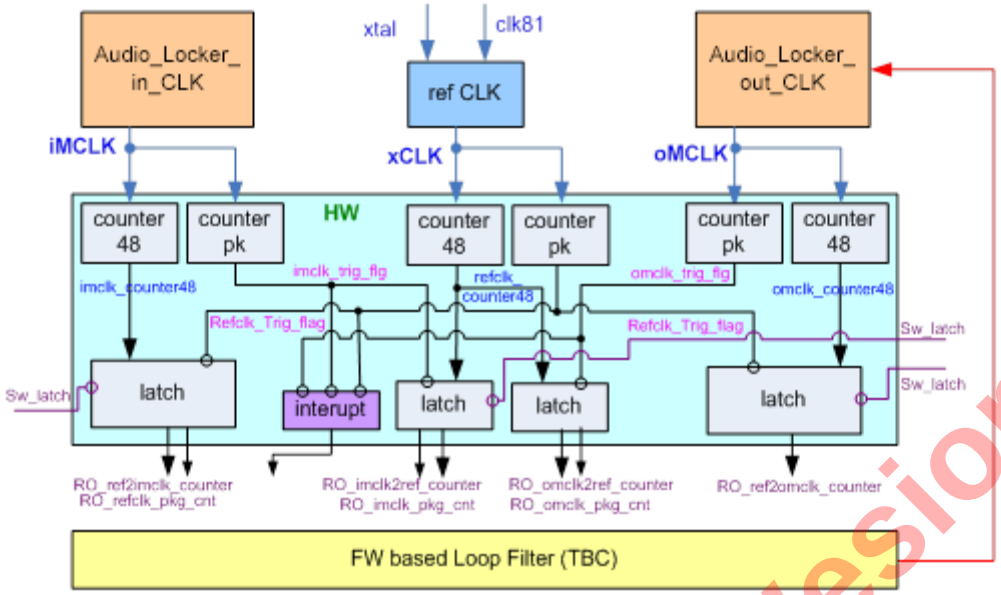
This method requires the audio output frequency precision, because there is no information statistics of output frequency. When the audio output frequency is not stable, it is recommended it not be used.

11.8 Audio Locker

The audio input clock and output clock are in different power domain and may have little difference, the difference leads to FIFO problem, S922X use Audio Locker module to fix this problem, the basic method is to measure the difference between different clock and adjust(software dynamically) the clock frequency to make them match.

Below show the diagram of Audio Locker module.

Figure 11-16 Audio Locker Diagram



11.9 Register Description

11.9.1 PDM Registers

Base Address: 0xFF640000

Each register final address = module base address+ address * 4

PDM_CTRL 0x00

Bits	R/W	Default	Description
31	R/W	0x00000000	PDM enable
30	R/W		invert the PDM_DCLK
29	R/W		output mode: 1: 24bits. 0: 32 bits
28	R/W		bypass mode. 1: bypass all filter. directly output the PDM input to DDR. 0: normal mode.
27:20	R/W		pdm_mute_mask[7:0]; [7] mute pdm ch7; … [0] mute pdm ch0;
17	R/W		chnum_en. 1: start send ch_cnt out.
16	R/W		PDM Asynchronous FIFO soft reset. write 1 to soft reset AFIFO
15:8	R/W		PDM channel reset. 0: to reset each PDM channel. 1: normal mode
7:0	R/W		PDM channel enable. each bit for one channel

PDM_HCIC_CTRL1 0x01

Bits	R/W	Default	Description

31	R/W	0x00000000	hcic filter enable. 1 use sinc filter. 0 bypass input to output
29:24	R/W		hcic final gain shift parameter
23:16	R/W		hcic final gain multiplier
8:4	R/W		hcic down sample rate
3:0	R/W		hcic stage number. must be between 3 to 9

PDM_HCIC_CTRL2 0x02

PDM_F1_CTRL 0x03

Bits	R/W	Default	Description
31	R/W	0x00000000	filter 1 enable
16:15	R/W		f1 round mode. 2'b00 : sign bit at bit 49. 28bits output [49:22] round at bit 21. 32bits output [49:18]. 24bits output [49:26] 2'b01 : sign bit at bit 50. 28bits output [50:23] round at bit 22. 32bits output [49:18]. 24bits output [49:26] 2'b10 : sign bit at bit 51. 28bits output [51:24] round at bit 23 32bits output [49:18]. 24bits output [49:26].
15:12	R/W		filter 1 down sample rate
8:0	R/W		filter 1 stage number

PDM_F2_CTRL 0x04

Bits	R/W	Default	Description
31	R/W	0x00000000	filter 2 enable
16:15	R/W		F2 round mode. 2'b00 : round at bit 21. 2'b01 : round at bit 22.2'b10 : round at bit 23
15:12	R/W		filter 2 down sample rate
8:0	R/W		filter 2 stage number

PDM_F3_CTRL 0x05

Bits	R/W	Default	Description
31	R/W	0x00000000	filter 3 enable
16:15	R/W		F3 round mode. 2'b00 : round at bit 21. 2'b01 : round at bit 22.2'b10 : round at bit 23
15:12	R/W		filter 3 down sample rate
8:0	R/W		filter 3 stage number

PDM_HPF_CTRL 0x06

Bits	R/W	Default	Description
31	R/W	0x00000000	High pass filter enable
20:16	R/W		high pass filter shift steps. 6~19 steps

15:0	R/W		high pass filter output factor

PDM_CHAN_CTRL 0x07

Bits	R/W	Default	Description
31:24	R/W	0x00000000	Chan3 data sample pointer vs edge of the PDM_DCLK
23:16	R/W		Chan2 data sample pointer vs edge of the PDM_DCLK
15:8	R/W		Chan1 data sample pointer vs edge of the PDM_DCLK
7:0	R/W		Chan0 data sample pointer vs edge of the PDM_DCLK

PDM_CHAN_CTRL1 0x08

Bits	R/W	Default	Description
31:24	R/W	0x00000000	Chan7 data sample pointer vs edge of the PDM_DCLK
23:16	R/W		Chan6 data sample pointer vs edge of the PDM_DCLK
15:8	R/W		Chan5 data sample pointer vs edge of the PDM_DCLK
7:0	R/W		Chan4 data sample pointer vs edge of the PDM_DCLK

PDM_COEFF_ADDR 0x09

Bits	R/W	Default	Description
8:0	R/W	0x00000000	address of the write/read of coeff data

PDM_COEFF_DATA 0x0A

Bits	R/W	Default	Description
31:0	R/W	0x00000000	write/read data to coeff memory

PDM_CLKG_CTRL 0x0B

Bits	R/W	Default	Description
6	R/W	0x00000000	filt_ctrl module auto clock gating control
5	R/W		sinc fifo module auto clock gating control
4	R/W		filter module auto clock gating control
3	R/W		apb module auto clock gating control
2	R/W		coeff memory module auto clock gating control
1	R/W		each channel module auto clock gating control
0	R/W		cts_pdm_clk auto clock gating control

PDM_STS 0x0C

Bits	R/W	Default	Description
1	R/W	0x00000000	HPF filter output overflow. means the PCLK is too slow
0	R/W		HCIC filter output overflow. means the CTS_PDM_CLK is too slow. can't finished the filter function.

PDM_MUTE_VALUE 0x0D

Bits	R/W	Default	Description
31:0	R/W	0x00000000	mute value if mute_mask = 1;

11.9.2 CLK Registers

The register signal are connected in the following way, the *_slv_sclk_* signal are connected to only a,b, c from core pinmux, *_slv_sclk-_d is connected to wifi_beeacon signal; only slv_lrclk_a, b, c are connected, slv_lrclk_d is not connected, as shown in the following table:

Table 11-1 Audio Clock Signal Connection

Register Signal	Connected Signal
i_slv_sclk_a	TDMA_SLV_SCLK
i_slv_sclk_b	TDMB_SLV_SCLK
i_slv_sclk_c	TDMC_SLV_SCLK
i_slv_sclk_d	wifi_beeacon
i_slv_lrclk_a	TDMA_SLV_FS
i_slv_lrclk_b	TDMB_SLV_FS
i_slv_lrclk_c	TDMC_SLV_FS
i_slv_lrclk_d	-

For Below registers:

Base Address: 0xFF642000

Each register final address = module base address+ address * 4

EE_AUDIO_CLK_GATE_EN 0x00

Bits	R/W	Default	Description
22	R/W	0x00000000	eqdrc, 0:disable; 1: enable;
21	R/W	0x00000000	spdifoutB, 0:disable; 1: enable;
20	R/W	0x00000000	toram, 0:disable; 1: enable;
19	R/W	0x00000000	power detect, 0:disable; 1: enable;
18	R/W		resample, 0:disable; 1: enable;

Bits	R/W	Default	Description
17	R/W		spdifout, 0:disable; 1: enable;
16	R/W		spdifin, 0:disable; 1: enable;
15	R/W		loopback, 0:disable; 1: enable;
14	R/W		toddrc, 0:disable; 1: enable;
13	R/W		toddrb, 0:disable; 1: enable;
12	R/W		toddra, 0:disable; 1: enable;
11	R/W		frddrc, 0:disable; 1: enable;
10	R/W		frddrb, 0:disable; 1: enable;
9	R/W		frddra, 0:disable; 1: enable;
8	R/W		tdmoutc, 0:disable; 1: enable;
7	R/W		tdmoutb, 0:disable; 1: enable;
6	R/W		tmdouta, 0:disable; 1: enable;
5	R/W		tdminlb, 0:disable; 1: enable;
4	R/W		tdminc, 0:disable; 1: enable;
3	R/W		tdminb, 0:disable; 1: enable;
2	R/W		tdmina, 0:disable; 1: enable;
1	R/W		pdm, 0:disable; 1: enable;
0	R/W		ddr_arb, 0:disable; 1: enable;

EE_AUDIO_MCLK_A_CTRL 0x01

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_en, 0:disable; 1: enable;
30	R/W		Force_oscin, 0:disable; 1: force clock source as oscin(24M);
26:24	R/W		clk_sel, 0:mp0_pll; 1:mp1_pll; 2:mp2_pll; 3:mp3_pll; 4:hifi_pll; 5:fclk_div3(666M); 6:fclk_div4(500M); 7:gp0_pll;
15:0	R/W		clk_div, the frequency of mclk = pll/clk_div;

EE_AUDIO_MCLK_B_CTRL 0x02

Same as EE_AUDIO_MCLK_A_CTRL.

EE_AUDIO_MCLK_C_CTRL 0x03

Same as EE_AUDIO_MCLK_A_CTRL.

EE_AUDIO_MCLK_D_CTRL 0x04

Same as EE_AUDIO_MCLK_A_CTRL.

EE_AUDIO_MCLK_E_CTRL 0x05

Same as EE_AUDIO_MCLK_A_CTRL.

EE_AUDIO_MCLK_F_CTRL 0x06

Same as EE_AUDIO_MCLK_A_CTRL.

EE_AUDIO_MST_PAD_CTRL0 0x07

Bits	R/W	Default	Description
6:4	R/W		Mclk_pad_1_sel: 0: mclk_a; 1: mclk_b; 2: mclk_c; 3: mclk_d; 4: mclk_e; 5: mclk_f;
2:0	R/W		Mclk_pad_0_sel: 0: mclk_a; 1: mclk_b; 2: mclk_c; 3: mclk_d; 4: mclk_e; 5: mclk_f;

EE_AUDIO_MST_PAD_CTRL1 0x08

Bits	R/W	Default	Description
26:24	R/W		lrclk_pad_2_sel: 0: lrclk_a; 1: lrclk_b; 2: lrclk_c; 3: lrclk_d; 4: lrclk_e; 5: lrclk_f;
22:20	R/W		lrclk_pad_1_sel: 0: lrclk_a; 1: lrclk_b; 2: lrclk_c; 3: lrclk_d; 4: lrclk_e; 5: lrclk_f;

Bits	R/W	Default	Description
18:16	R/W		lrclk_pad_0_sel: 0: lrclk_a; 1: lrclk_b; 2: lrclk_c; 3: lrclk_d; 4: lrclk_e; 5: lrclk_f;
10:8	R/W		sclk_pad_2_sel: 0: sclk_a; 1: sclk_b; 2: sclk_c; 3: sclk_d; 4: sclk_e; 5: sclk_f;
6:4	R/W		sclk_pad_1_sel: 0: sclk_a; 1: sclk_b; 2: sclk_c; 3: sclk_d; 4: sclk_e; 5: sclk_f;
2:0	R/W		sclk_pad_0_sel: 0: sclk_a; 1: sclk_b; 2: sclk_c; 3: sclk_d; 4: sclk_e; 5: sclk_f;

EE_AUDIO_SW_RESET 0x09

Bits	R/W	Default	Description
25	R/W		clk tree
24	R/W		tohdmitx
23	R/W		toacodec
22	R/W		toram
21	R/W		powdet
20	R/W		ddrarb

Bits	R/W	Default	Description
19	R/W		resample
18	R/W		eqdrc
17	R/W		spdifin
16	R/W		spdifoutB
15	R/W		spdifout
14	R/W		tdmoutc
13	R/W		tdmoutb
12	R/W		tdmouta
11	R/W		frddrc
10	R/W		frddrb
9	R/W		frddra
8	R/W		toddrc
7	R/W		toddrb
6	R/W		toddra
5	R/W		loopback
4	R/W		tdmin_lb
3	R/W		tdminc
2	R/W		tdminb
1	R/W		tdmina
0	R/W		pdm

EE_AUDIO_MST_A_SCLK_CTRL0 0x10

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_in_en, 0:disable; 1: enable;
30	R/W		clk_out_en, 0:disable; 1: enable;
29:20	R/W		sclk_div, the frequency of sclk = mclk/sclk_div;
19:10	R/W		lrclk_hi, duty cycle of LRCLK, less than lrclk_div; example 0: lrclk_hi = 1, LRCLK will only keep one cycle; example 1: lrclk_hi = lrclk_div/2, LRCLK will be 50/50 duty cycle;
9:0	R/W		lrclk_div, the frequency of lrclk = sclk/lrclk_div;

EE_AUDIO_MST_A_SCLK_CTRL1 0x11

Bits	R/W	Default	Description
31:28	R/W	0x00000000	sclk_ph0_sel, select from sclk_delay_line(depth is 16)
27:24	R/W		lrclk_ph0_sel, select from lrclk_delay_line(depth is 16)
23:20	R/W		sclk_ph1_sel, select from sclk_delay_line(depth is 16)
19:16	R/W		lrclk_ph1_sel, select from lrclk_delay_line(depth is 16)
15:12	R/W		sclk_ph2_sel, select from sclk_delay_line(depth is 16)
11:8	R/W		lrclk_ph2_sel, select from lrclk_delay_line(depth is 16)
5:0	R/W		clk_inv, invert clk; [5]: lrclk_ph2; [4]: sclk_ph2; [3]: lrclk_ph1; [2]: sclk_ph1; [1]: lrclk_ph0; [0]: sclk_ph0;

EE_AUDIO_MST_B_SCLK_CTRL0 0x12

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_B_SCLK_CTRL1 0x13

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_MST_C_SCLK_CTRL0 0x14

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_C_SCLK_CTRL1 0x15

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_MST_D_SCLK_CTRL0 0x16

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_D_SCLK_CTRL1 0x17

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_MST_E_SCLK_CTRL0 0x18

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_E_SCLK_CTRL1 0x19

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_MST_F_SCLK_CTRL0 0x1a

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_F_SCLK_CTRL1 0x1b

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_CLK_TDMIN_A_CTRL 0x20

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_in_en, 0:disable; 1: enable;

30	R/W		clk_out_en, 0:disable; 1: enable;
29	R/W		sclk_inv, 0:not revert; 1: revert clock;
27:24	R/W		sclk_sel, 0:mst_a_sclk_ph1; 1:mst_b_sclk_ph1; 2:mst_c_sclk_ph1; 3:mst_d_sclk_ph1; 4:mst_e_sclk_ph1; 5:mst_f_sclk_ph1; 6:i_slv_sclk_a ; 7:i_slv_sclk_b ; 8:i_slv_sclk_c ; 9::i_slv_sclk_d ; 10:i_slv_sclk_e ; 11:i_slv_sclk_f ; 12:i_slv_sclk_g ; 13:i_slv_sclk_h ; 14:i_slv_sclk_i ; 15:i_slv_sclk_j ; Please refer to Table 11-1
23:20	R/W		lrclk_sel, 0:mst_a_lrclk_ph1; 1:mst_b_lrclk_ph1; 2:mst_c_lrclk_ph1; 3:mst_d_lrclk_ph1; 4:mst_e_lrclk_ph1; 5:mst_f_lrclk_ph1; 6:i_slv_lrclk_a ; 7:i_slv_lrclk_b ; 8:i_slv_lrclk_c ; 9::i_slv_lrclk_d ; 10:i_slv_lrclk_e ; 11:i_slv_lrclk_f ; 12:i_slv_lrclk_g ; 13:i_slv_lrclk_h ; 14:i_slv_lrclk_i ; 15:i_slv_lrclk_j ; Please refer to Table 11-1

EE_AUDIO_CLK_TDMIN_B_CTRL 0x21

Same as EE_AUDIO_CLK_TDMIN_A_CTRL

EE_AUDIO_CLK_TDMIN_C_CTRL 0x22

Same as EE_AUDIO_CLK_TDMIN_A_CTRL

EE_AUDIO_CLK_TDMIN_LB_CTRL 0x23

Same as EE_AUDIO_CLK_TDMIN_A_CTRL

EE_AUDIO_CLK_TDMOUT_A_CTRL 0x24

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_in_en, 0:disable; 1: enable;
30	R/W		clk_out_en, 0:disable; 1: enable;
29	R/W		sclk_inv, 0:not revert; 1: revert clock;
28	R/W		Sclk_ws_inv, for the capture ws sclk; 0: not revert; 1: revert clock;

27:24	R/W		sclk_sel, 0:mst_a_sclk_ph1; 1:mst_b_sclk_ph1; 2:mst_c_sclk_ph1; 3:mst_d_sclk_ph1; 4:mst_e_sclk_ph1; 5:mst_f_sclk_ph1; 6:i_slv_sclk_a ; 7:i_slv_sclk_b ; 8:i_slv_sclk_c ; 9::i_slv_sclk_d ; 10:i_slv_sclk_e ; 11:i_slv_sclk_f ; 12:i_slv_sclk_g ; 13:i_slv_sclk_h ; 14:i_slv_sclk_i ; 15:i_slv_sclk_j ; Please refer to Table 11-1
23:20	R/W		lrclk_sel, 0:mst_a_lrclk_ph1; 1:mst_b_lrclk_ph1; 2:mst_c_lrclk_ph1; 3:mst_d_lrclk_ph1; 4:mst_e_lrclk_ph1; 5:mst_f_lrclk_ph1; 6:i_slv_lrclk_a ; 7:i_slv_lrclk_b ; 8:i_slv_lrclk_c ; 9::i_slv_lrclk_d ; 10:i_slv_lrclk_e ; 11:i_slv_lrclk_f ; 12:i_slv_lrclk_g ; 13:i_slv_lrclk_h ; 14:i_slv_lrclk_i ; 15:i_slv_lrclk_j ; Please refer to Table 11-1

EE_AUDIO_CLK_TDMOUT_B_CTRL 0x25

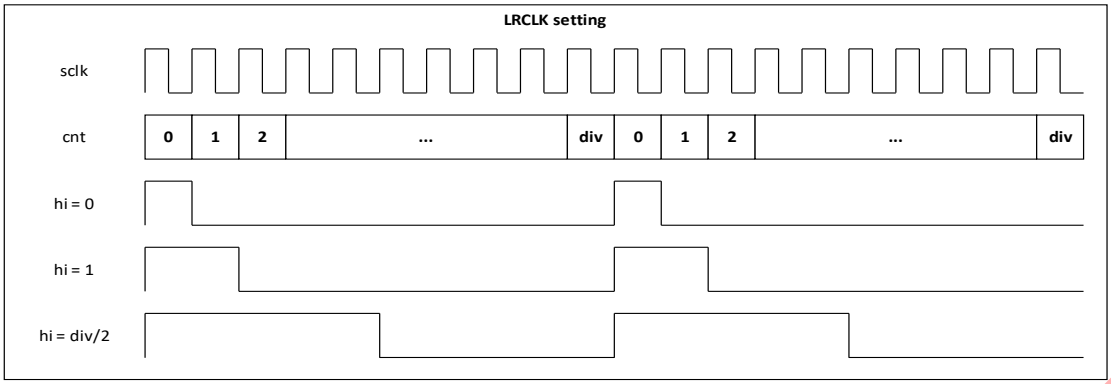
Same as EE_AUDIO_CLK_TDMOUT_A_CTRL

EE_AUDIO_CLK_TDMOUT_C_CTRL 0x26

Same as EE_AUDIO_CLK_TDMOUT_A_CTRL

EE_AUDIO_CLK_SPDIFIN_CTRL 0x27

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_en, 0:disable; 1: enable;
30	R/W		Force_oscin, 0:disable; 1: force clock source as oscin(24M);
26:24	R/W		clk_sel, 0:mp0_pll; 1:mp1_pll; 2:mp2_pll; 3:mp3_pll; 4:hifi_pll; 5:fclk_div3(666M); 6:fclk_div4(500M); 7:gp0_pll;
7:0	R/W		clk_div, the frequency of mclk = pll/clk_div;



EE_AUDIO_MST_B_SCLK_CTRL0 0x12

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_B_SCLK_CTRL1 0x13

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_MST_C_SCLK_CTRL0 0x14

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_C_SCLK_CTRL1 0x15

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_MST_D_SCLK_CTRL0 0x16

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_D_SCLK_CTRL1 0x17

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_MST_E_SCLK_CTRL0 0x18

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_E_SCLK_CTRL1 0x19

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_MST_F_SCLK_CTRL0 0x1a

Same as EE_AUDIO_MST_A_SCLK_CTRL0

EE_AUDIO_MST_F_SCLK_CTRL1 0x1b

Same as EE_AUDIO_MST_A_SCLK_CTRL1

EE_AUDIO_CLK_TDMIN_A_CTRL 0x20

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_in_en, 0:disable; 1: enable;
30	R/W		clk_out_en, 0:disable; 1: enable;
29	R/W		sclk_inv, 0:not revert; 1: revert clock;

27:24	R/W		sclk_sel, 0:mst_a_sclk_ph1; 1:mst_b_sclk_ph1; 2:mst_c_sclk_ph1; 3:mst_d_sclk_ph1; 4:mst_e_sclk_ph1; 5:mst_f_sclk_ph1; 6:i_slv_sclk_a ; 7:i_slv_sclk_b ; 8:i_slv_sclk_c ; 9::i_slv_sclk_d ; 10:i_slv_sclk_e ; 11:i_slv_sclk_f ; 12:i_slv_sclk_g ; 13:i_slv_sclk_h ; 14:i_slv_sclk_i ; 15:i_slv_sclk_j ; Please refer to Table 11-1
23:20	R/W		lrclk_sel, 0:mst_a_lrclk_ph1; 1:mst_b_lrclk_ph1; 2:mst_c_lrclk_ph1; 3:mst_d_lrclk_ph1; 4:mst_e_lrclk_ph1; 5:mst_f_lrclk_ph1; 6:i_slv_lrclk_a ; 7:i_slv_lrclk_b ; 8:i_slv_lrclk_c ; 9::i_slv_lrclk_d ; 10:i_slv_lrclk_e ; 11:i_slv_lrclk_f ; 12:i_slv_lrclk_g ; 13:i_slv_lrclk_h ; 14:i_slv_lrclk_i ; 15:i_slv_lrclk_j ; Please refer to Table 11-1

EE_AUDIO_CLK_TDMIN_B_CTRL 0x21

Same as EE_AUDIO_CLK_TDMIN_A_CTRL

EE_AUDIO_CLK_TDMIN_C_CTRL 0x22

Same as EE_AUDIO_CLK_TDMIN_A_CTRL

EE_AUDIO_CLK_TDMIN_LB_CTRL 0x23

Same as EE_AUDIO_CLK_TDMIN_A_CTRL

EE_AUDIO_CLK_TDMOUT_A_CTRL 0x24

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_in_en, 0:disable; 1: enable;
30	R/W		clk_out_en, 0:disable; 1: enable;
29	R/W		sclk_inv, 0:not revert; 1: revert clock;
28	R/W		Sclk_ws_inv, for the capture ws sclk; 0: not revert; 1: revert clock;

27:24	R/W		sclk_sel, 0:mst_a_sclk_ph1; 1:mst_b_sclk_ph1; 2:mst_c_sclk_ph1; 3:mst_d_sclk_ph1; 4:mst_e_sclk_ph1; 5:mst_f_sclk_ph1; 6:i_slv_sclk_a ; 7:i_slv_sclk_b ; 8:i_slv_sclk_c ; 9::i_slv_sclk_d ; 10:i_slv_sclk_e ; 11:i_slv_sclk_f ; 12:i_slv_sclk_g ; 13:i_slv_sclk_h ; 14:i_slv_sclk_i ; 15:i_slv_sclk_j ; Please refer to Table 11-1
23:20	R/W		lrclk_sel, 0:mst_a_lrclk_ph1; 1:mst_b_lrclk_ph1; 2:mst_c_lrclk_ph1; 3:mst_d_lrclk_ph1; 4:mst_e_lrclk_ph1; 5:mst_f_lrclk_ph1; 6:i_slv_lrclk_a ; 7:i_slv_lrclk_b ; 8:i_slv_lrclk_c ; 9::i_slv_lrclk_d ; 10:i_slv_lrclk_e ; 11:i_slv_lrclk_f ; 12:i_slv_lrclk_g ; 13:i_slv_lrclk_h ; 14:i_slv_lrclk_i ; 15:i_slv_lrclk_j ; Please refer to Table 11-1

EE_AUDIO_CLK_TDMOUT_B_CTRL 0x25

Same as EE_AUDIO_CLK_TDMOUT_A_CTRL

EE_AUDIO_CLK_TDMOUT_C_CTRL 0x26

Same as EE_AUDIO_CLK_TDMOUT_A_CTRL

EE_AUDIO_CLK_SPDIFIN_CTRL 0x27

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_en, 0:disable; 1: enable;
30	R/W		Force_oscin, 0:disable; 1: force clock source as oscin(24M);
26:24	R/W		clk_sel, 0:mp0_pll; 1:mp1_pll; 2:mp2_pll; 3:mp3_pll; 4:hifi_pll; 5:fclk_div3(666M); 6:fclk_div4(500M); 7:gp0_pll;
7:0	R/W		clk_div, the frequency of mclk = pll/clk_div;

EE_AUDIO_CLK_SPDIFOUT_CTRL 0x28

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_en, 0:disable; 1: enable;
30	R/W		Force_oscin, 0:disable; 1: force clock source as oscin(24M);
26:24	R/W		clk_sel, 0:mp0_pll; 1:mp1_pll; 2:mp2_pll; 3:mp3_pll; 4:hifi_pll; 5:fclk_div3(666M); 6:fclk_div4(500M); 7:gp0_pll;
9:0	R/W		clk_div, the frequency of mclk = pll/clk_div;

EE_AUDIO_CLK_RESAMPLE_CTRL 0x29

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_ en, 0:disable; 1: enable;
30	R/W		Force_oscin, 0:disable; 1: force clock source as oscin(24M);
27:24	R/W		sclk_sel, 0:mst_a_sclk_ph1; 1:mst_b_sclk_ph1; 2:mst_c_sclk_ph1; 3:mst_d_sclk_ph1; 4:mst_e_sclk_ph1; 5:mst_f_sclk_ph1; 6:i_slv_sclk_a ; 7:i_slv_sclk_b ; 8:i_slv_sclk_c ; 9::i_slv_sclk_d ; 10:i_slv_sclk_e ; 11:i_slv_sclk_f ; 12:i_slv_sclk_g ; 13:i_slv_sclk_h ; 14:i_slv_sclk_i ; 15:i_slv_sclk_j ; Please refer to Table 11-1
7:0	R/W		Clk_div

EE_AUDIO_CLK_LOCKER_CTRL 0x2a

Bits	R/W	Default	Description
31	R/W	0x00000000	lock_out_clk; 0:disable; 1:enable;
30	R/W		Force_oscin for lock_out_clk, 0:disable; 1: force clock source as oscin(24M);
27:24	R/W		sclk_sel, 0:mst_a_sclk_ph1; 1:mst_b_sclk_ph1; 2:mst_c_sclk_ph1; 3:mst_d_sclk_ph1; 4:mst_e_sclk_ph1; 5:mst_f_sclk_ph1; 6:i_slv_sclk_a ;

Bits	R/W	Default	Description
			7:i_slv_sclk_b ; 8:i_slv_sclk_c ; 9::i_slv_sclk_d ; 10:i_slv_sclk_e ; 11:i_slv_sclk_f ; 12:i_slv_sclk_g ; 13:i_slv_sclk_h ; 14:i_slv_sclk_i ; 15:i_slv_sclk_j ; Please refer to Table 11-1
23:16	R/W		clk_div, lock_out_clk; out = in/clk_div;
15	R/W		clk_en, lock_in_clk; 0:disable; 1:enable;
14	R/W		Force_oscin for lock_in_clk, 0:disable; 1: force clock source as oscin(24M);
11:8	R/W		clk_sel, lock_in_clk; 0:mst_a_mclk; 1:mst_b_mclk; 2:mst_c_mclk; 3:mst_d_mclk; 4:mst_e_mclk; 5:mst_f_mclk; 6:i_slv_sclk_a ; 7:i_slv_sclk_b ; 8:i_slv_sclk_c ; 9::i_slv_sclk_d ; 10:i_slv_sclk_e ; 11:i_slv_sclk_f ; 12:i_slv_sclk_g ; 13:i_slv_sclk_h ; 14:i_slv_sclk_i ; 15:i_slv_sclk_j ; Please refer to Table 11-1
7:0	R/W		clk_div, lock_in_clk; out = in/clk_div;

EE_AUDIO_CLK_PDMIN_CTRL0 0x2b

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_en, 0:disable; 1: enable;
30	R/W		Force_oscin, 0:disable; 1: force clock source as oscin(24M);
26:24	R/W		clk_sel, 0:mp0_pll; 1:mp1_pll; 2:mp2_pll; 3:mp3_pll; 4:hifi_pll; 5:fclk_div3(666M); 6:fclk_div4(500M); 7:gp0_pll;
15:0	R/W		the frequency of pdm_dclk = pll/clk_div;

EE_AUDIO_CLK_PDMIN_CTRL1 0x2c

Bits	R/W	Default	Description

31	R/W	0x00000000	clk_en, 0:disable; 1: enable;
30	R/W		Force_oscin, 0:disable; 1: force clock source as oscin(24M);
26:24	R/W		clk_sel, 0:mp0_pll; 1:mp1_pll; 2:mp2_pll; 3:mp3_pll; 4:hifi_pll; 5:fclk_div3(666M); 6:fclk_div4(500M); 7:gp0_pll;
15:0	R/W		the frequency of pdm_sysclk = pll/clk_div;

EE_AUDIO_CLK_SPDIFOUT_B_CTRL 0x2d

Bits	R/W	Default	Description
31	R/W	0x00000000	clk_en, 0:disable; 1: enable;
30	R/W		Force_oscin, 0:disable; 1: force clock source as oscin(24M);
26:24	R/W		clk_sel, 0:mp0_pll; 1:mp1_pll; 2:mp2_pll; 3:mp3_pll; 4:hifi_pll; 5:fclk_div3(666M); 6:fclk_div4(500M); 7:gp0_pll;
9:0	R/W		clk_div, the frequency of mclk = pll/clk_div;

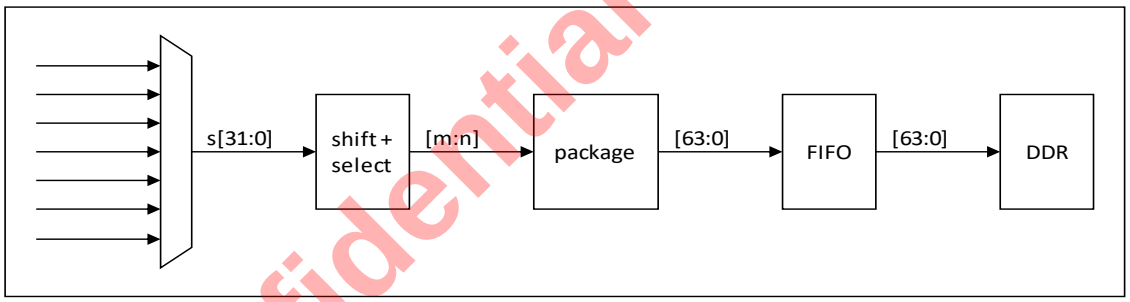
11.9.3 TODDR Registers

EE_AUDIO_TODDR_A_CTRL0 0x40

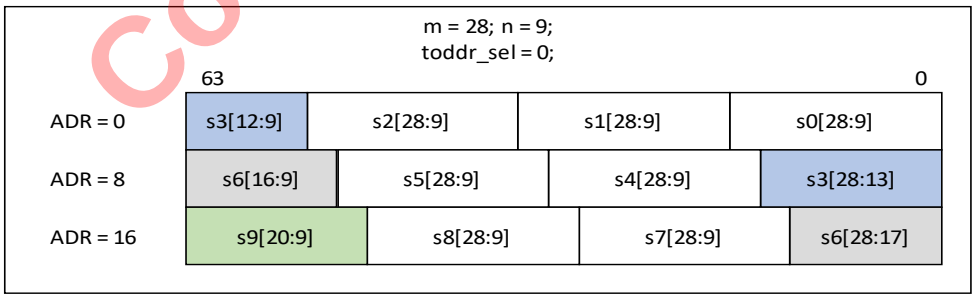
Bits	R/W	Default	Description
31	R/W	0x00000000	reg_toddr_en, 0: disable; 1: enable;
30	R/W		reg_sel_resample, 0: select original data; 1: select reample data;
29	R/W		reg_ext_signed, 0: select write to only one buff (start_addr,finish_addr); 1: select write to two buff (start_addr,finish_addr) (start_addrb, finish_addrb);
28	R/W		reg_toddr_endian
27	R/W		Enable_sync_chnum; 1: start store data when first ch ;
26:24	R/W		reg_toddr_int_en

Bits	R/W	Default	Description
23:16	R/W		[23] : reserved; [22] : reserved; [21] : fifo overflow, write when fifocnt = depth; [20] : fifo overflow, read when fifocnt = 0; [19] : when write to ddr "int_addr" data (only once); [18] : when write to ddr "int_addr" data (repeat); [17] : when write to ddr address match "int_addr"; [16]: when write to ddr address match "finish_addr";
15:13	R/W		reg_toddr_sel, 0: combined data[m:n] without gap; like S0[m:n],S1[m:n],S2[m:n],… 1: combined data[m:n] as 16bits; like {S0[11:0],4'd0},{S1[11:0],4'd0}… 2: combined data[m:n] as 16bits; like {4'd0,S0[11:0]},{4'd0,{S1[11:0]}… 3: combined data[m:n] as 32bits; like {S0[27:4],8'd0},{S1[27:4],8'd0}… 4: combined data[m:n] as 32bits; like {8'd0,S0[27:4]},{8'd0,{S1[27:4]}…
12:8	R/W		reg_toddr_m_sel, the msb positioin in data
7:3	R/W		reg_toddr_n_sel, the lsb position in data
2:0	R/W		reg_toddr_src_sel, [7]: loopback; [6]: tdmin_lb; [5]: reserved; [4]: pdmin; [3]: spdifin; [2]: tdmin_c; [1]: tdmin_b; [0]: tdmin_a;

We can change format before write to DDR:



For example:



m = 8; n = 1;
toddr_sel = 1;

	63			0
ADR = 0	s3[8:1],8'd0	s2[8:1],8'd0	s1[8:1],8'd0	s0[8:1],8'd0
ADR = 8	s7[8:1],8'd0	s6[8:1],8'd0	s5[8:1],8'd0	s4[8:1],8'd0

m = 22; n = 12;
toddr_sel = 2;

	63			0
ADR = 0	5'd0,s3[22:12]	5'd0,s2[22:12]	5'd0,s1[22:12]	5'd0,s0[22:12]
ADR = 8	5'd0,s7[22:12]	5'd0,s6[22:12]	5'd0,s5[22:12]	5'd0,s4[22:12]

m = 18; n = 4;
toddr_sel = 3;

	63	0
ADR = 0	s1[18:4],17'd0	s0[18:4],17'd0
ADR = 8	s3[18:4],17'd0	s2[18:4],17'd0

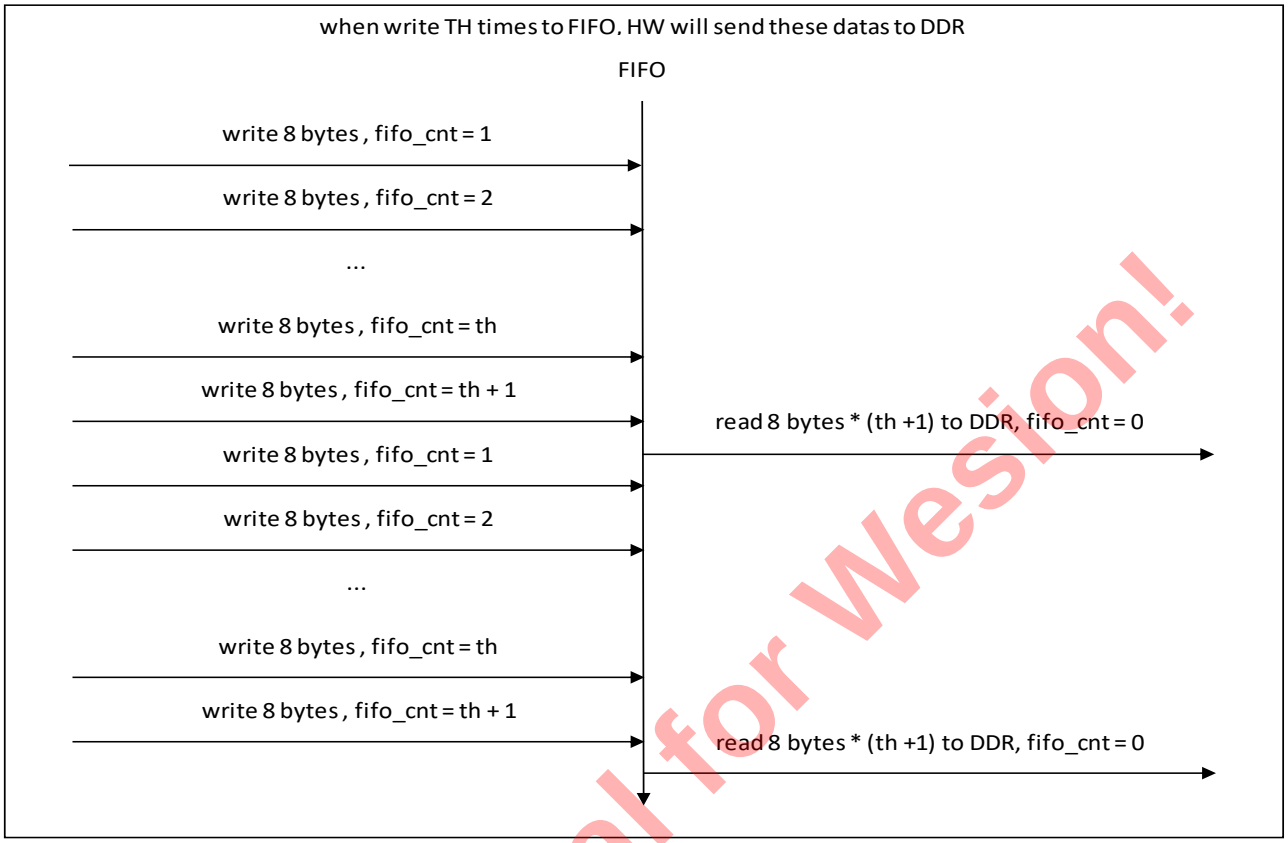
m = 11; n = 3;
toddr_sel = 4;

	63	0
ADR = 0	23'd0,s1[11:3]	23'd0,s0[11:3]
ADR = 8	23'd0,s3[11:3]	23'd0,s2[11:3]

EE_AUDIO_TODDR_A_CTRL1 0x41

Bits	R/W	Default	Description
25	R/W	0x00000000	force_finish; the value from 0-> 1: force finish by current address and jump to start_address;
24	R/W		Insert_chnum; 0: disable; 1: insert chnum[9:0] to data[9:0]
23:16	R/W		reg_fifo_start_rd_th, each time, when fifo_cnt greater than this register, control will start read data from fifo and write to DDR; write length is "reg_fifo_start_rd_th + 1" * 8bytes;
11:8	R/W		reg_status_sel, control status2 source;

7:0	R/W		reg_int_status_clr,clear each bits of int_status register



EE_AUDIO_TODDR_A_START_ADDR 0x42

Bits	R/W	Default	Description
31:0	R/W	0x00000000	start_addr, buff_A start address, ignore [2:0]

EE_AUDIO_TODDR_A_FINISH_ADDR 0x43

Bits	R/W	Default	Description
31:0	R/W	0x00000000	finish_addr, buff_A finish address, ignore [2:0]

EE_AUDIO_TODDR_A_INT_ADDR 0x44

Bits	R/W	Default	Description
31:0	R/W	0x00000000	int_addr, usage A : as an address of interrupt; usage B : as a count of interrupt;

EE_AUDIO_TODDR_A_STATUS1 0x45

Bits	R/W	Default	Description
19	R/W	0x00000000	sel_b_true
18	R/W		sel_b

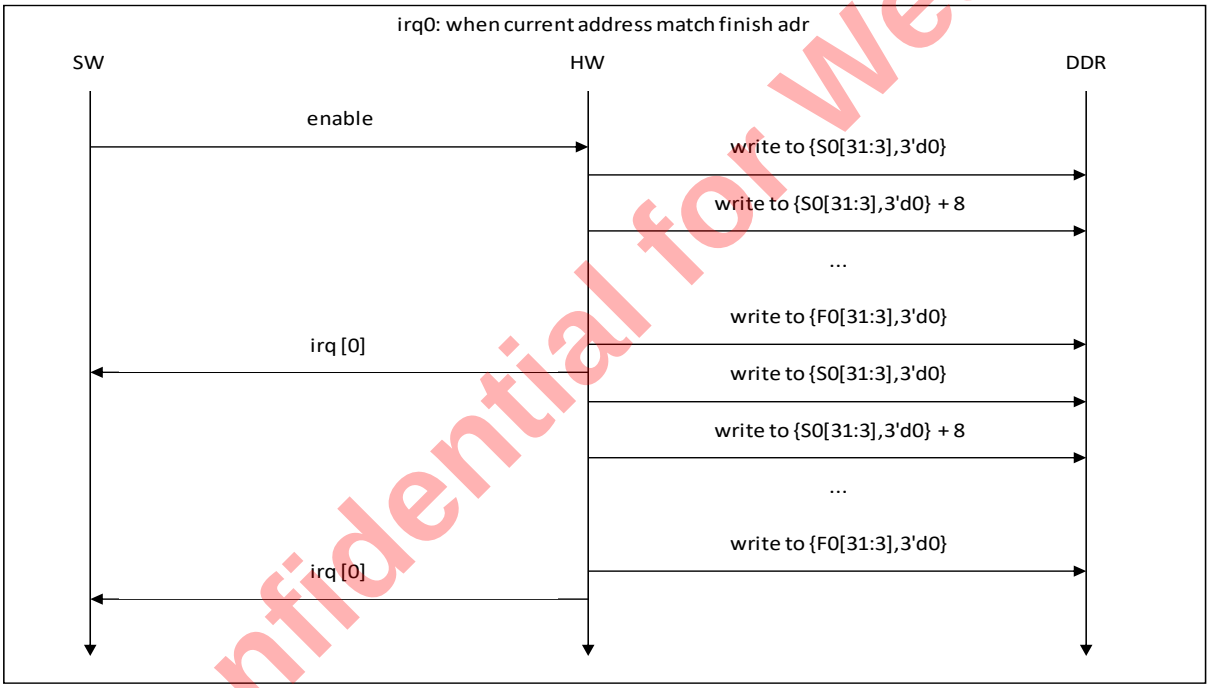
Bits	R/W	Default	Description
16:8	R/W		fifo count, the num in fifo
7:0	R/W		int_status, when irq generate, related bit will changed to 1 and can only clear by reg_int_status_clr

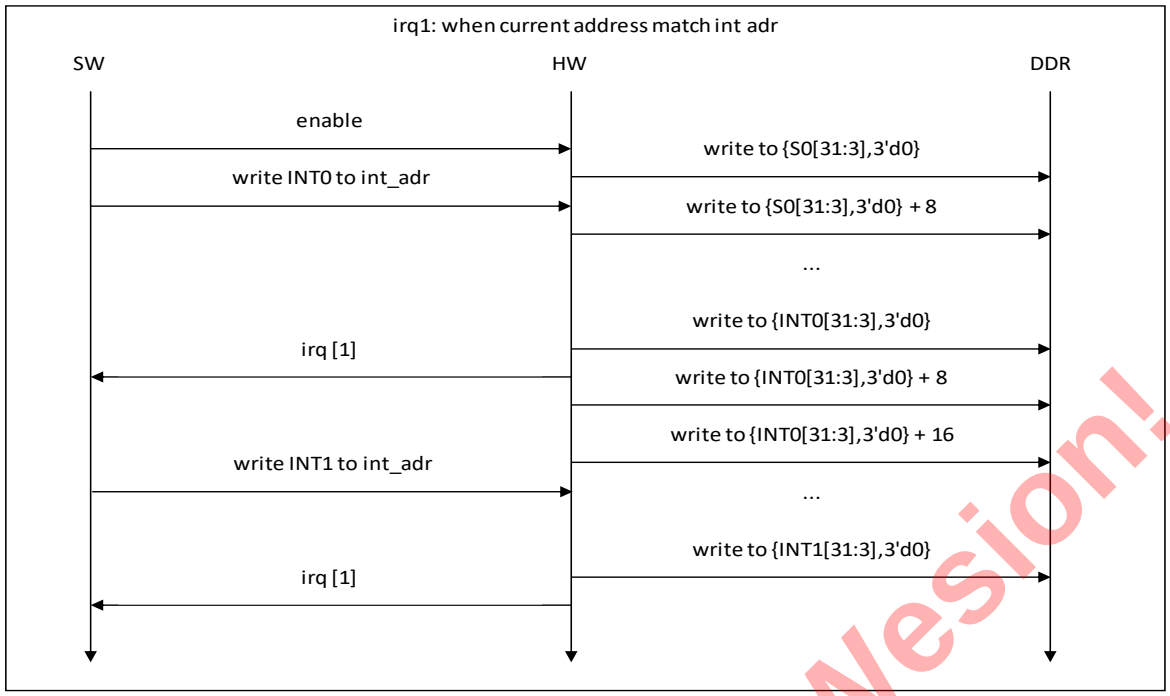
EE_AUDIO_TODDR_A_STATUS2 0x46

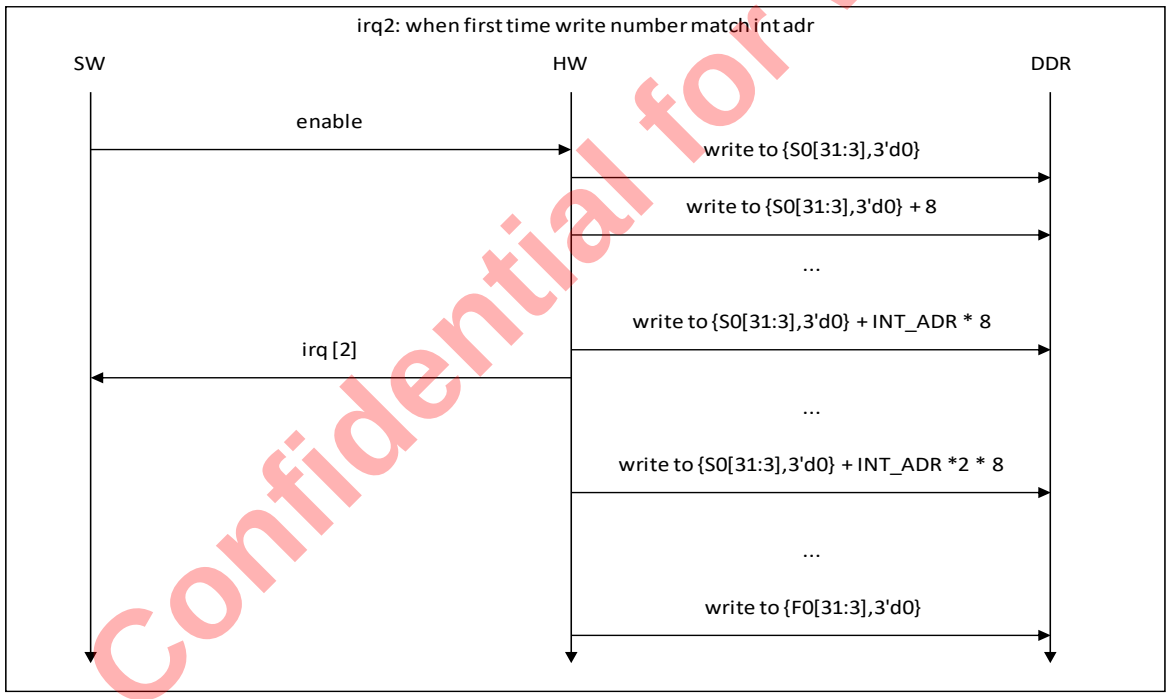
Bits	R/W	Default	Description
31:0	R/W	0x00000000	status2, by reg_status_sel: 0: current ddr write address; 1: next finish address; 2: count by ddr reply, current ddr write address; 3: count by ddr reply, next finish address;

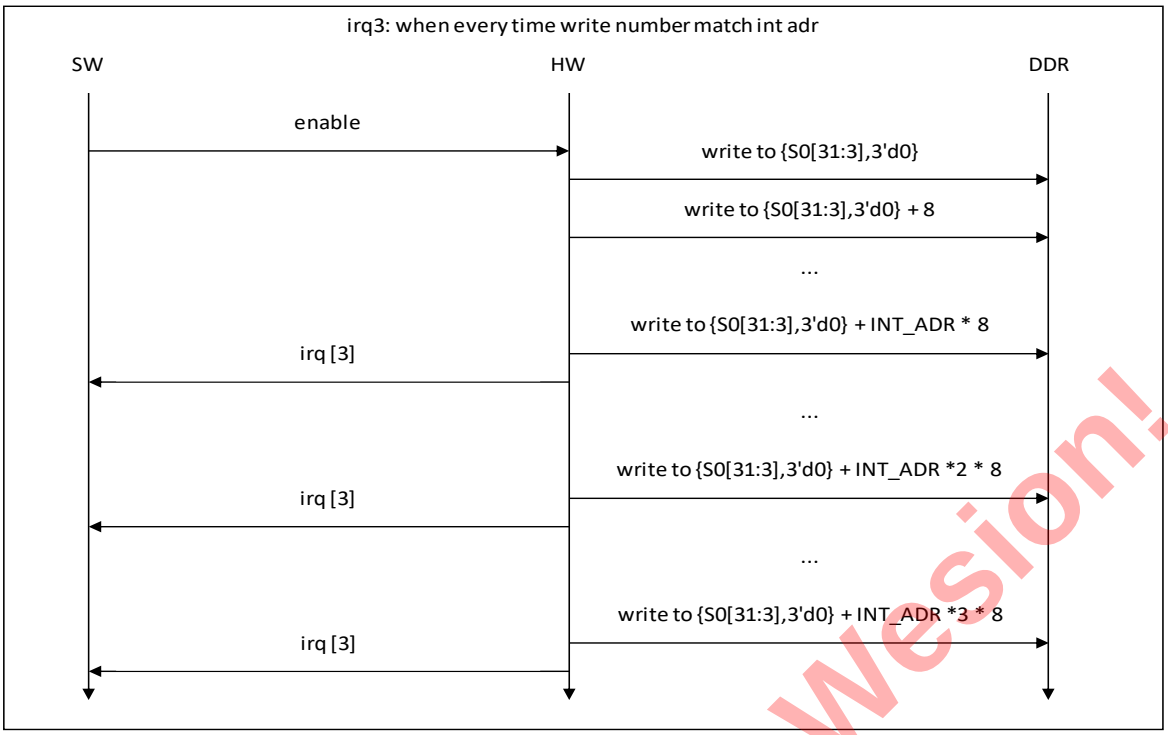
We can generate 8 irq and add them together to CPU.

Then can read int_status to know it's which irq:









EE_AUDIO_TODDR_A_START_ADDRB 0x47

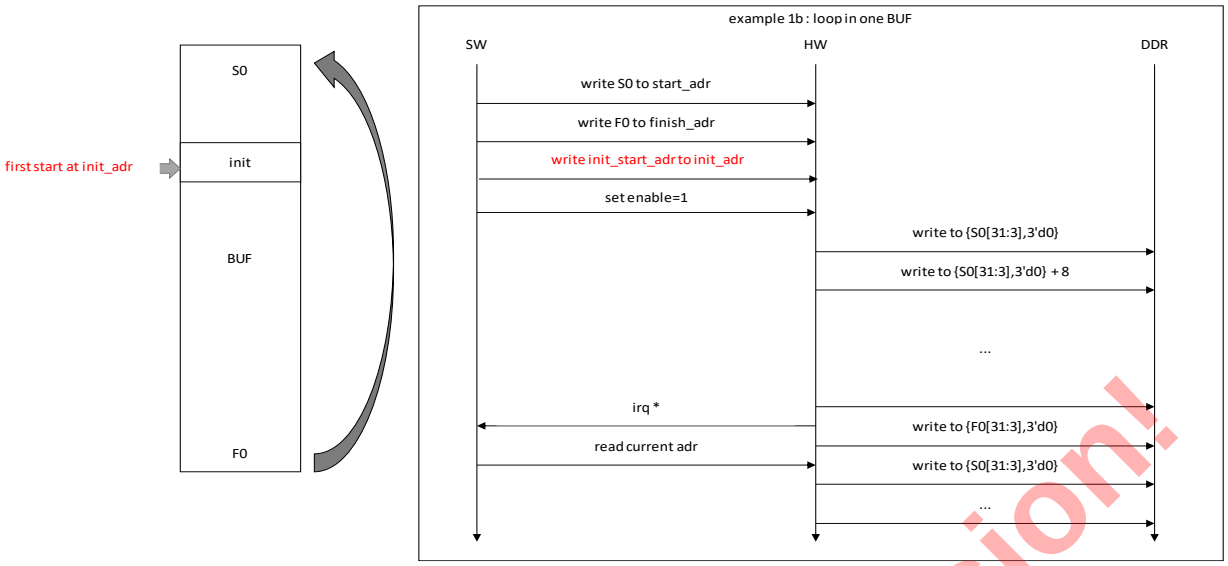
Bits	R/W	Default	Description
31:0	R/W	0x00000000	istart_addrb, buff_B start address, ignore [2:0]

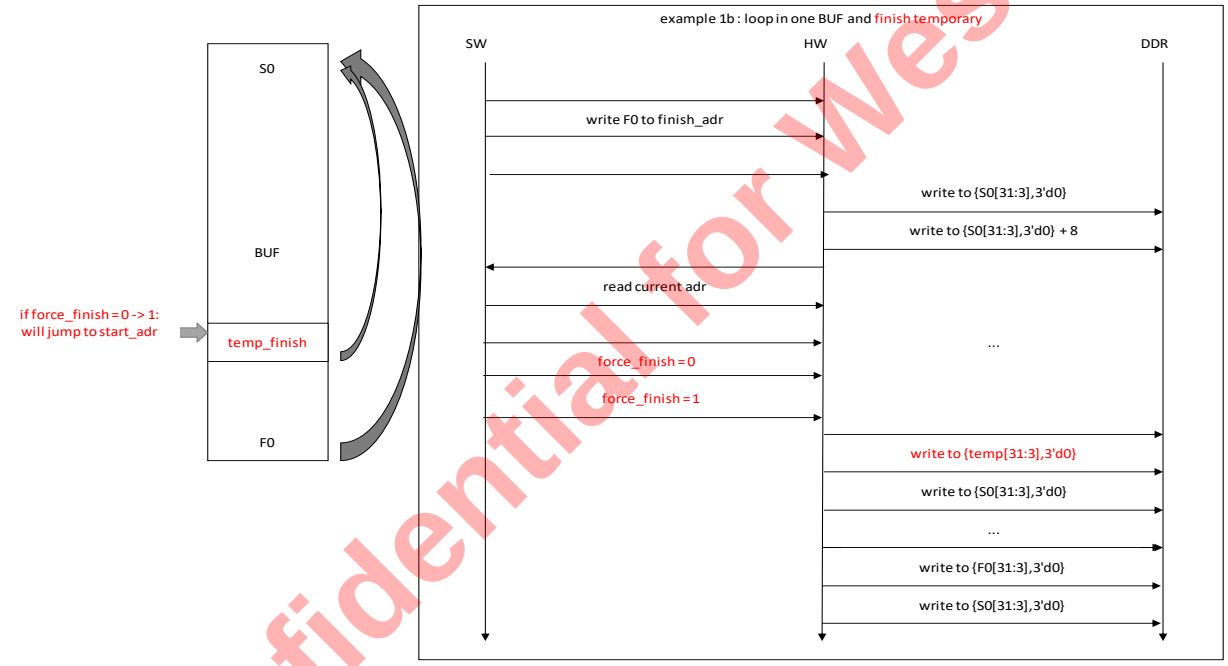
EE_AUDIO_TODDR_A_FINISH_ADDRB 0x48

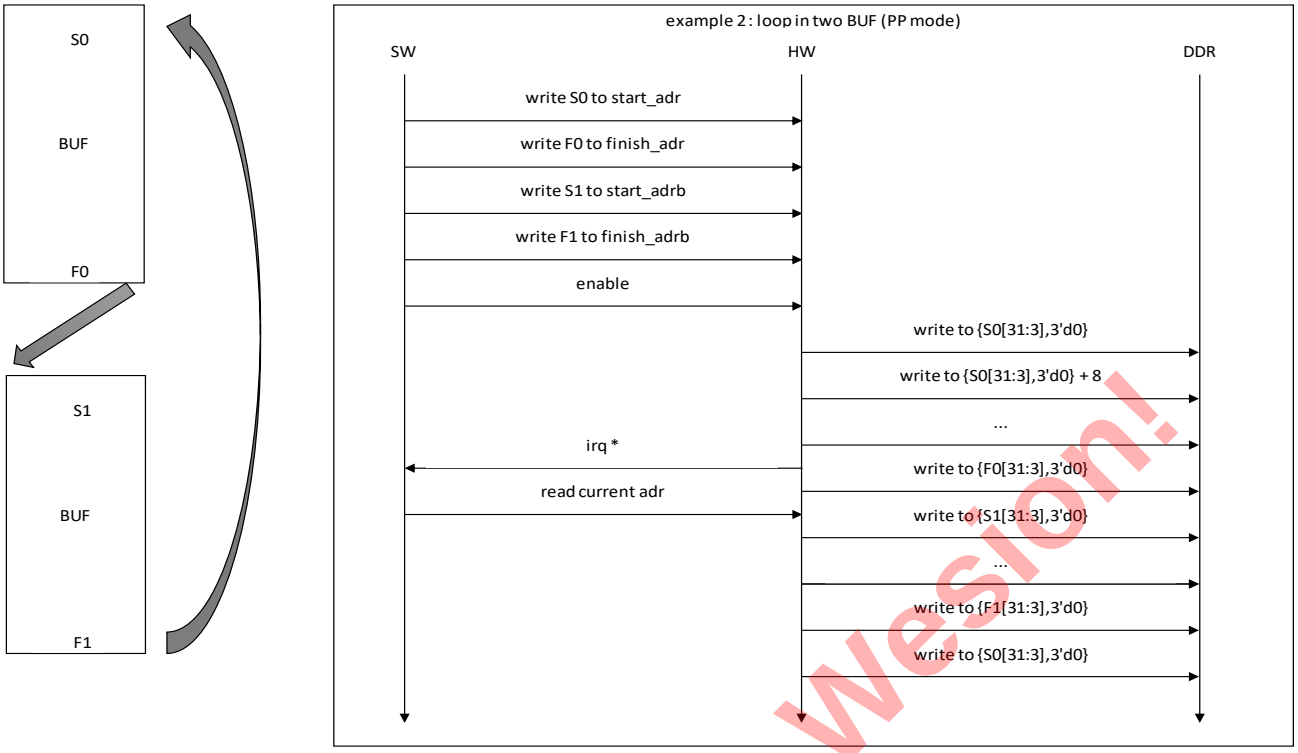
Bits	R/W	Default	Description
31:0	R/W	0x00000000	finish_addrb, buff_B finish address, ignore [2:0]

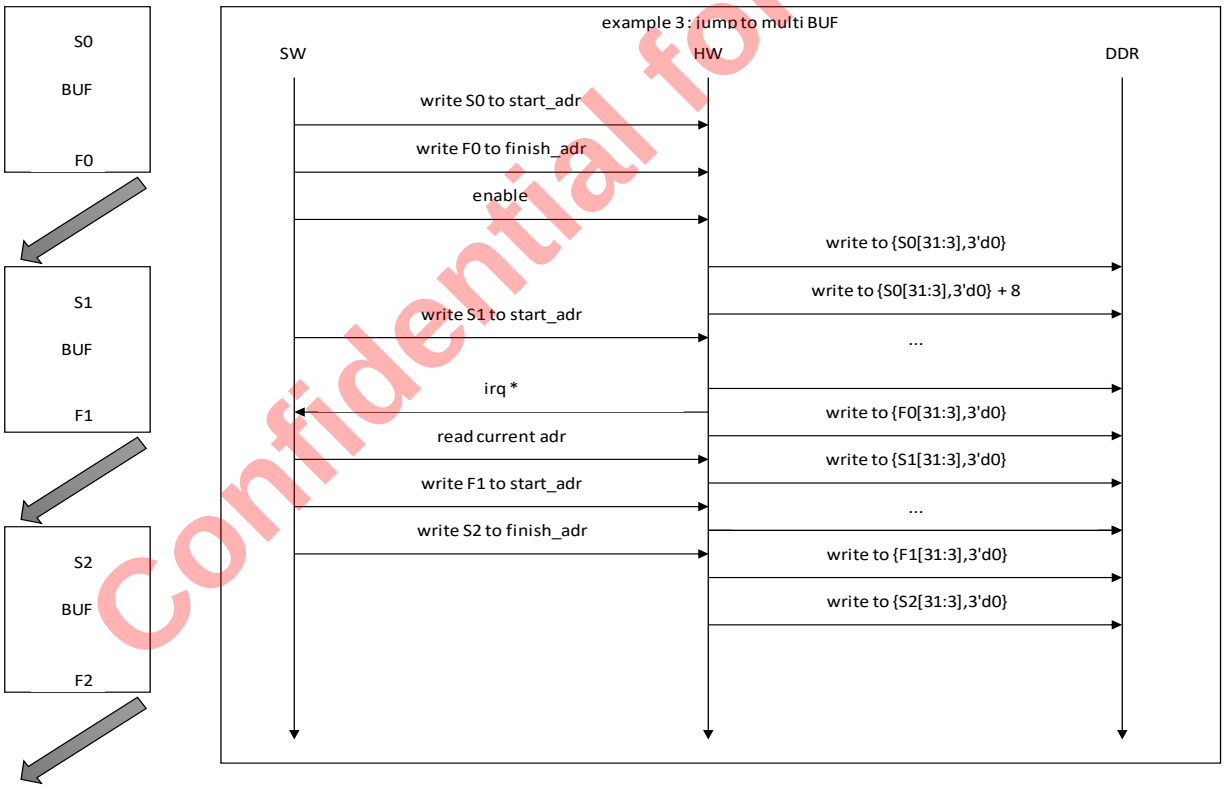
EE_AUDIO_TODDR_A_INIT_ADDR 0x49

Bits	R/W	Default	Description
31:0	R/W	0x00000000	initial address, the first ddr address after enable









EE_AUDIO_TODDR_B_CTRL0 0x50

EE_AUDIO_TODDR_B_CTRL1 0x51

EE_AUDIO_TODDR_B_START_ADDR 0x52

EE_AUDIO_TODDR_B_FINISH_ADDR 0x53

EE_AUDIO_TODDR_B_INT_ADDR 0x54

EE_AUDIO_TODDR_B_STATUS1 0x55

EE_AUDIO_TODDR_B_STATUS2 0x56

EE_AUDIO_TODDR_B_START_ADDRB 0x57

EE_AUDIO_TODDR_B_FINISH_ADDR 0x58

EE_AUDIO_TODDR_B_INIT_ADDR 0x59

EE_AUDIO_TODDR_C_CTRL0 0x60

EE_AUDIO_TODDR_C_CTRL1 0x61

EE_AUDIO_TODDR_C_START_ADDR 0x62

EE_AUDIO_TODDR_C_FINISH_ADDR 0x63

EE_AUDIO_TODDR_C_INT_ADDR 0x64

EE_AUDIO_TODDR_C_STATUS1 0x65

EE_AUDIO_TODDR_C_STATUS2 0x66

EE_AUDIO_TODDR_C_START_ADDRB 0x67

EE_AUDIO_TODDR_C_FINISH_ADDR 0x68

EE_AUDIO_TODDR_C_INIT_ADDR 0x69

11.9.4 FRDDR Registers

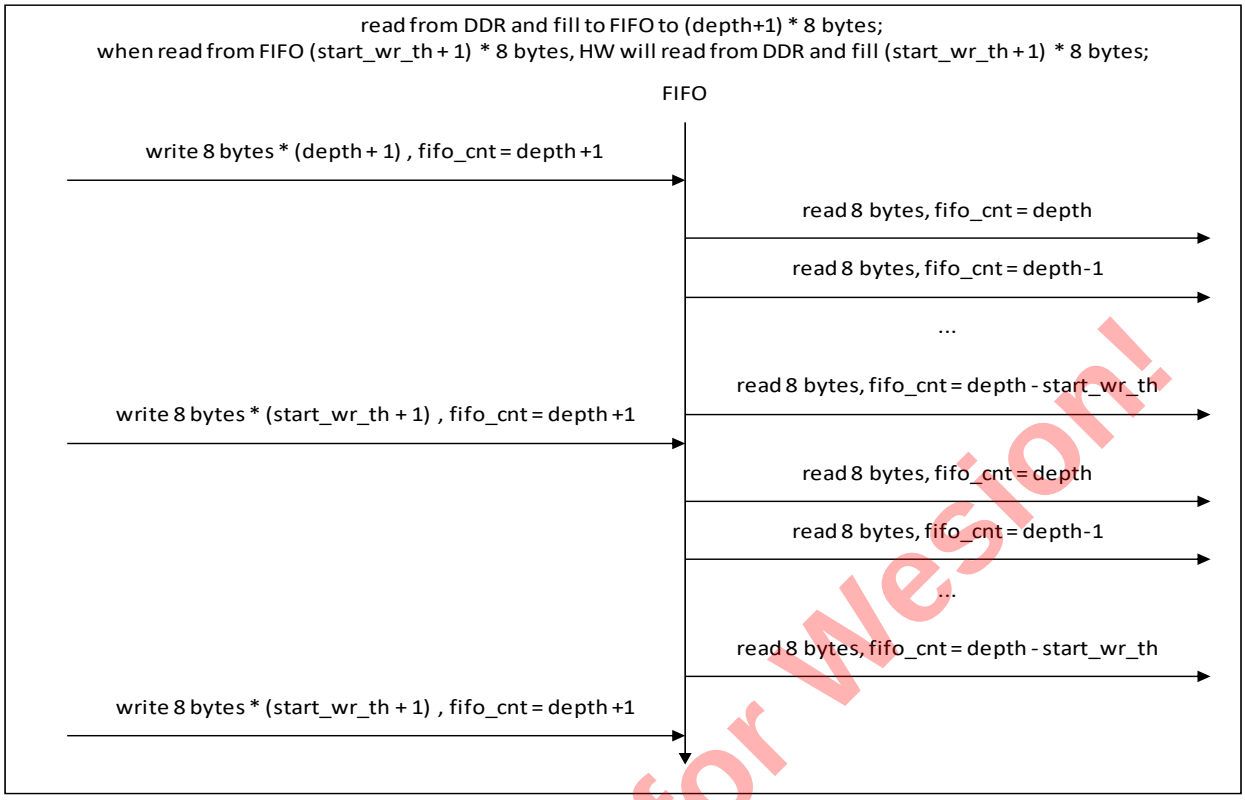
EE_AUDIO_FRDDR_A_CTRL0 0x70

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_frddr_en, 0: disable; 1: enable;
30	R/W		reg_frddr_pp_mode, 0: select write to only one buff (start_addr,finish_addr); 1: select write to two buff (start_addr,finish_addr) (start_addrb, finish_addrb);
26:24	R/W		reg_frdd_endian
23:16	R/W		reg_frddr_int_en, [23] : reserved; [22] : reserved; [21] : fifo overflow, write when fifocnt = depth; [20] : fifo overflow, read when fifocnt = 0; [19] : first time when read from ddr "int_addr" data(only once); [18] : once time when read from ddr "int_addr" data(repeat); [17] : when read from ddr address match "int_addr"; [16]: when read from ddr address match "finish_addr";
15:12	R/W		reg_frddr_ack_dly, add delay to frddr ack
11	R/W		Reg_frddr_src_sel3_en; 0: ignorereq_src_sel3; 1: waitreq_src_sel3;

Bits	R/W	Default	Description
10:8	R/W		reg_frddr_src_sel3, 0: tdmout_a; 1: tdmout_b; 2: tdmout_c; 3: spdifout; 4: spdifout_b; 5: reserved; 6: reserved; 7: reserved;
7	R/W		Reg_frddr_src_sel2_en; 0: ignorereq_src_sel2; 1: waitreq_src_sel2;
6:4	R/W		reg_frddr_src_sel2, 0: tdmout_a; 1: tdmout_b; 2: tdmout_c; 3: spdifout; 4: spdifout_b; 5: reserved; 6: reserved; 7: reserved;
3	R/W		Reg_frddr_src_sel1_en; 0: ignorereq_src_sel1; 1: waitreq_src_sel1;
2:0	R/W		reg_frddr_src_sel1, 0: tdmout_a; 1: tdmout_b; 2: tdmout_c; 3: spdifout; 4: spdifout_b; 5: reserved; 6: reserved; 7: reserved;

EE_AUDIO_FRDDR_A_CTRL1 0x71

Bits	R/W	Default	Description
31:24	R/W	0x00000000	reg_fifo_depth, the max depth of fifo; for high bit rates like 384k*32bits*8ch, set this register higher; for low bit rates like 48k*32bits*2ch, set this register lower;
23:16			reg_fifo_start_wr_th, when the fifo cnt less than "reg_fifo_depth - reg_fifo_start_wr_th", start reqeuest and read data from DDR; each time request "reg_fifo_start_wr_th" * 8 bytes data;
12			force finish; when the value changed from 0 to 1; will finished by current address and jump to start address;
11:8			reg_status_sel, control status2 source;
7:0			reg_int_status_clr, clear each bits of int_status register



EE_AUDIO_FRDDR_A_START_ADDR 0x72

Bits	R/W	Default	Description
31:0	R/W	0x00000000	start_addr,buff_B start address

EE_AUDIO_FRDDR_A_FINISH_ADDR 0x73

Bits	R/W	Default	Description
31:0	R/W	0x00000000	finish_addr,buff_B finish address

EE_AUDIO_FRDDR_A_INT_ADDR 0x74

Bits	R/W	Default	Description
31:0	R/W	0x00000000	int_addr,usage A : as an address of interrupt; usage B : as a count of interrupt;

EE_AUDIO_FRDDR_A_STATUS1 0x75

Bits	R/W	Default	Description
19	R/W	0x00000000	r_selb_true
18	R/W		r_selb
17:8	R/W		fifo count, the num in fifo

7:0	R/W		int_status, when irq generate, related bit will changed to 1 and can only clear by reg_int_status_clr

EE_AUDIO_FRDDR_A_STATUS2 0x76

Bits	R/W	Default	Description
31:0	R/W	0x00000000	status2, by reg_status_sel: 0: current ddr write address; 1: next finish address; 2: count by ddr reply, current ddr write address; 3: count by ddr reply, next finish address;

The same design as TODDR:

We can generate 8 irq and add them together to CPU.

Then can read int_status to know it's which irq.

EE_AUDIO_FRDDR_A_START_ADDRB 0x77

EE_AUDIO_FRDDR_A_FINISH_ADDRB 0x78

The same design as TODDR:

We have an internal register to store reg_start_adr and reg_finish_adr.

We call them as r_start_adr and r_finish_adr;

Each time, when curr_adr match r_finish_adr, curr_adr will jump to r_start_adr , then update r_start_adr and r_finish_adr;

That's mean SW can write new start adr and finish adr before curr_adr match old finish adr .

EE_AUDIO_FRDDR_A_INIT_ADDR 0x79

EE_AUDIO_FRDDR_B_CTRL0 0x80
EE_AUDIO_FRDDR_B_CTRL1 0x81
EE_AUDIO_FRDDR_B_START_ADDR 0x82
EE_AUDIO_FRDDR_B_FINISH_ADDR 0x83
EE_AUDIO_FRDDR_B_INT_ADDR 0x84
EE_AUDIO_FRDDR_B_STATUS1 0x85
EE_AUDIO_FRDDR_B_STATUS2 0x86
EE_AUDIO_FRDDR_B_START_ADDRB 0x87
EE_AUDIO_FRDDR_B_FINISH_ADDR 0x88
EE_AUDIO_FRDDR_B_INIT_ADDR 0x89
EE_AUDIO_FRDDR_C_CTRL0 0x90
EE_AUDIO_FRDDR_C_CTRL1 0x91
EE_AUDIO_FRDDR_C_START_ADDR 0x92
EE_AUDIO_FRDDR_C_FINISH_ADDR 0x93
EE_AUDIO_FRDDR_C_INT_ADDR 0x94
EE_AUDIO_FRDDR_C_STATUS1 0x95
EE_AUDIO_FRDDR_C_STATUS2 0x96
EE_AUDIO_FRDDR_C_START_ADDRB 0x97
EE_AUDIO_FRDDR_C_FINISH_ADDR 0x98
EE_AUDIO_FRDDR_C_INIT_ADDR 0x99

11.9.5 DDR ARB Registers

EE_AUDIO_ARB_CTRL 0xa0

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_arb_en, 0:disable; 1: enable;
7:0	R/W		reg_arb_mask, [7]: reserved; [6]: frddr_c; [5]: frddr_b; [4]: frddr_a; [3]: reserved; [2]: toddr_c; [1]: toddr_b; [0]: toddr_a;

11.9.6 LoopBack Registers

EE_AUDIO_LB_CTRL0 0xb0

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_lb_en, 0:disable; 1: enable;

30	R/W		reg_lb_mode, 0: out rate = in data rate; 1: out rate = loopback data rate;
29	R/W		reg_ext_signed, 0: extend bits as "0"; 1: extend bits as "msb";
28	R/W		Enable_sync_chnum; 1: start store data when ch_num can match ID;
27	R/W		chnum_en; 1: start send new ch num out;
26:24	R/W		reg_dat_chnum, max channel number of in data;
23:16	R/W		reg_dat_mask, mask each channel of in data;
15:13	R/W		reg_dat_sel, shift [m:n] to [31:0]; 0: right justified, out = { ext,[m:n]}; 1: left justified, out = {[m:n],all0}; 2: right justified, out = { ext,[m:n]}; 3: left justified, out = {[m:n],all0}; 4: right justified, out = { ext,[m:n]};
12:8	R/W		reg_dat_m_sel, the msb positioin in data
7:3	R/W		reg_dat_n_sel, the lsb position in data
2:0	R/W		reg_dat_src_sel, [7]: reserved; [6]: reserved; [5]: reserved; [4]: pdmin; [3]: spdifin; [2]: tdmin_c; [1]: tdmin_b; [0]: tdmin_a;

EE_AUDIO_LB_CTRL1 0xb1

Bits	R/W	Default	Description
29	R/W	0x00000000	reg_lb_ext_signed, 0: extend bits as "0"; 1: extend bits as "msb";
28	R/W		Enable_sync_chnum; 1: start store data when ch_num can match ID;
27	R/W		hold_insertion 1: hold insert data store(update) and insert 0 after original data;
26:24			reg_lb_chnum, max channel number of loopback data;
23:16			reg_lb_mask, mask each channel of loopback data;
15:13			reg_lb_sel, shift [m:n] to [31:0]; 0: right justified, out = {ext,[m:n]}; 1: left justified, out = {[m:n],all0}; 2: right justified, out = { ext,[m:n]}; 3: left justified, out = {[m:n],all0}; 4: right justified, out = { ext,[m:n]};

Bits	R/W	Default	Description
12:8			reg_lb_m_sel, the msb positioin loopback data
7:3			reg_lb_n_sel, the lsb position loopback data

EE_AUDIO_DAT_ID0 0xb2

Bits	R/W	Default	Description
31:24	R/W	0x00000000	data_ch7_id
23:16	R/W		data_ch6_id
15:8	R/W		data_ch5_id
7:0	R/W		data_ch4_id

EE_AUDIO_DAT_ID1 0xb3

Bits	R/W	Default	Description
31:24	R/W	0x00000000	data_ch3_id
23:16	R/W		data_ch2_id
15:8	R/W		data_ch1_id
7:0	R/W		data_ch0_id

EE_AUDIO_LB_ID0 0xb4

Bits	R/W	Default	Description
31:24	R/W	0x00000000	lb_data_ch7_id
23:16	R/W		lb_data_ch6_id
15:8	R/W		lb_data_ch5_id
7:0	R/W		lb_data_ch4_id

EE_AUDIO_LB_ID1 0xb5

Bits	R/W	Default	Description
31:24	R/W	0x00000000	lb_data_ch3_id
23:16	R/W		lb_data_ch2_id
15:8	R/W		lb_data_ch1_id
7:0	R/W		lb_data_ch0_id

EE_AUDIO_LB_STS 0xb6

Bits	R/W	Default	Description
27	R	0x00000000	output_en_err; when output, received a new start;

26	R		insert_storing_err2; it it's receving, but received first ch;
25	R		insert_storing_err1, if it's not receving, but received data
24	R		orig_storing_err2; it it's receving, but received first ch;
23	R		orig_storing_err1, if it's not receving, but received data
22	R		orig_updated, mean it's finished updated;
21	R		insert_storing, mean it's receiving insert data now;
20	R		orig_storing, mean it's receiving insert data now;
19	R		insert_store_cnt_err, when first_ch, if store_cnt is not 0, mean error, missed ch or another reason;
18:10	R		insert_store_cnt_debug[8:0], when first_ch, store current store_cnt value;
9	R		orig_store_cnt_err, when first_ch, if store_cnt is not 0, mean error, missed ch or another reason;
8:0	R		orig_store_cnt_debug[8:0], when first_ch, store current store_cnt value;

11.9.7 TDM Registers

TDMin_B/C are the same as TDMin_A.

TDMout_B/C are the same as TDMout_A.

TDMin_LB is the same as TDMin_A.

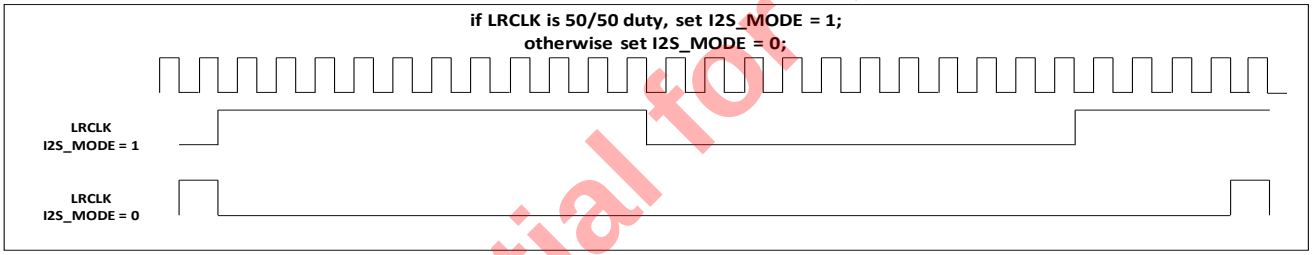
EE_AUDIO_TDMIN_A_CTRL 0xc0

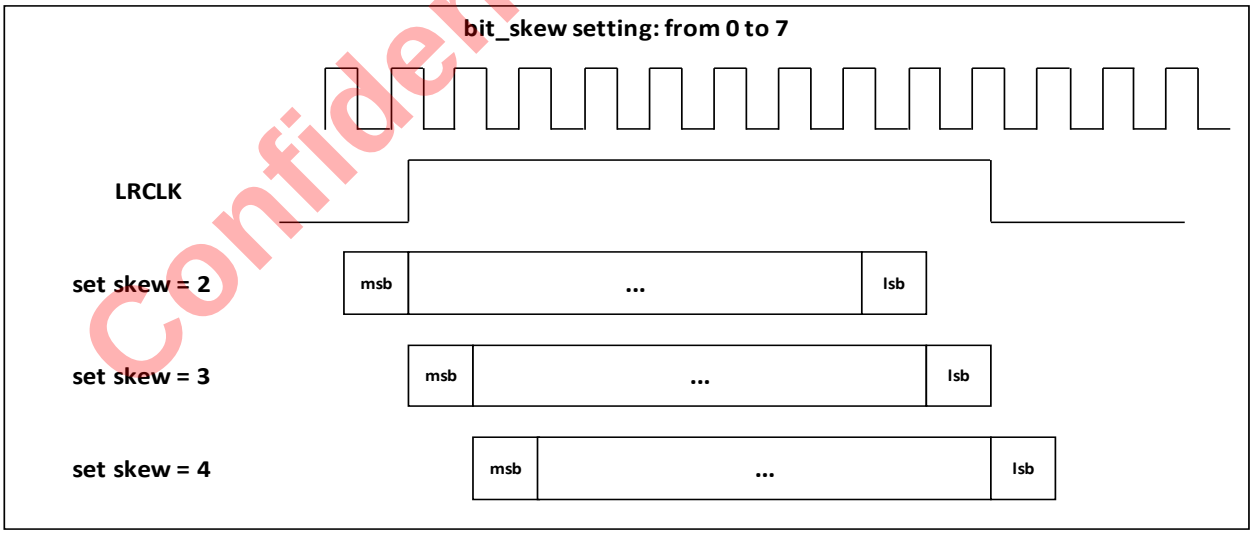
Bits	R/W	Default	Description
31	R/W	0x00000000	reg_work_enable 0:disable; 1:enable;
30	R/W		reg_i2s_mode,0:tdm mode; 1: i2s mode;
29	R/W		reg_rst_afifo_out_n, reset afifo out side; need set to 1 before reg_rst_afifo_in_n;
28	R/W		reg_rst_afifo_in_n, reset afifo in side; need set 1 after set reg_rst_afifo_out_n to 1;
27	R/W		reg_tdmin_in_debug_en, 0: disable debug mode; 1: enable;
26	R/W		reg_tdmin_in_auto_en,0: disalbe; 1: enable detect and store max/min of bit_cnt ;
25	R/W		reg_tdmin_in_rev_ws, revert ws(lrclk); 0 :disable; 1: enable;
24	R/W		reg_tdmin_in_rev_dat, revert data; 0:disable; 1:enable;

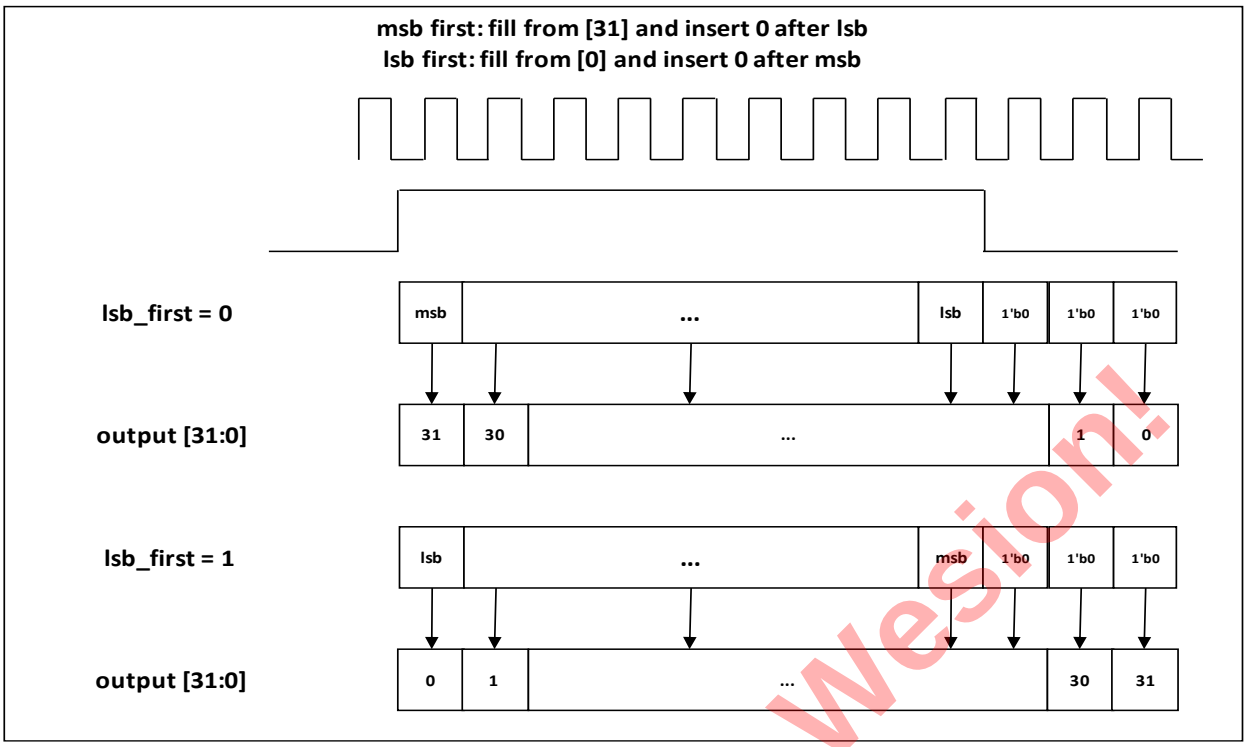
Bits	R/W	Default	Description

23:20	R/W		select tdmin src;
			0: PAD_TDMINA;//for tdmin_lb, it's tdmoutA
			1: PAD_TDMINB; //for tdmin_lb, it's tdmoutB
			2: PAD_TDMINC; //for tdmin_lb, it's tdmoutC
			3: TDMOUTA; //for tdmin_lb, it's pad_tdminA
			4: TDMOUTB; //for tdmin_lb, it's pad_tdminB
			5: TDMOUTC; //for tdmin_lb, it's pad_tdminC

18:16	R/W		reg_tdmin_in_bit_skew, add delay to ws or data for skew modification;
6	R/W		chnum_en; 1: add ch cnt to chnum;
5	R/W		reg_lsb_first, 0: store first bit received to data_store[0]; 1: store first bit received to data_store[31];
4:0	R/W		reg_tdmin_bit_num, bitwidth of each slot, if slot is 16bits, set this register to 15;

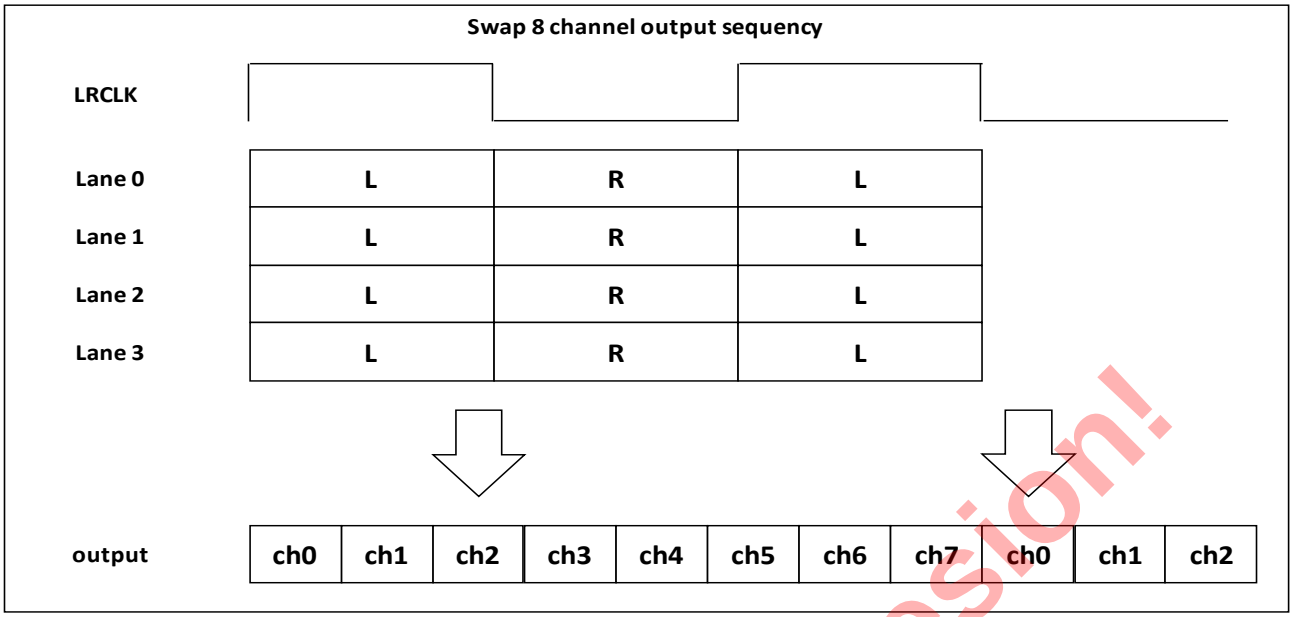






EE_AUDIO_TDMIN_A_SWAP 0xc1

Bits	R/W	Default	Description
30:28	R/W	0x00000000	ch7_sel, 0: lane0 left channel;
26:24	R/W		ch6_sel, 1: lane0 right channel;
22:20	R/W		ch5_sel, 2: lane1 left channel;
18:16	R/W		ch4_sel, 3: lane1 right channel;
14:12	R/W		ch3_sel, 4: lane2 left channel;
10:8	R/W		ch2_sel, 5: lane2 right channel;
6:4	R/W		ch1_sel, 6: lane3 left channel;
2:0	R/W		ch0_sel, 7: lane3 right channel;



EE_AUDIO_TDMIN_A_MASK0 0xc2

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane0_mask, mask each channel in lane0, max is 32 ch;

EE_AUDIO_TDMIN_A_MASK1 0xc3

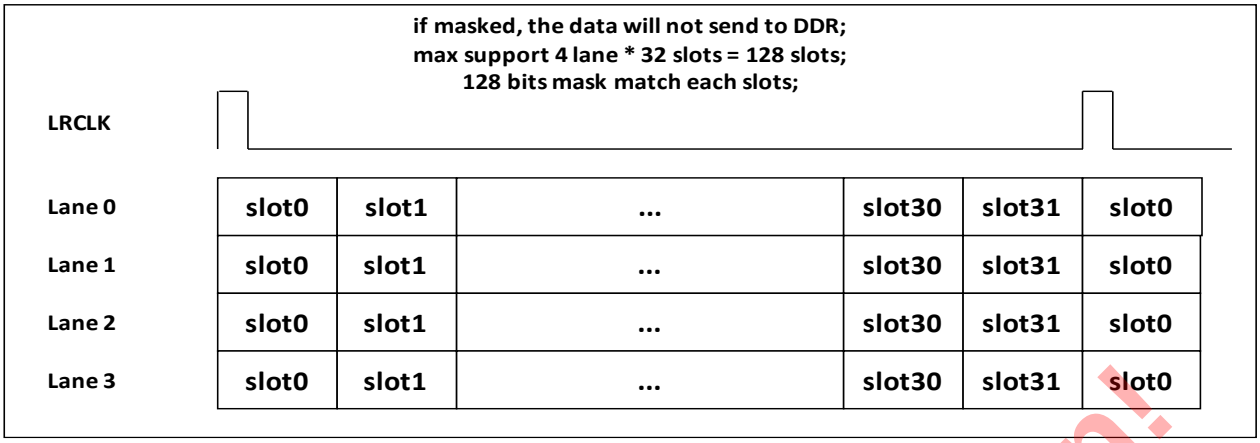
Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane1_mask, mask each channel in lane1, max is 32 ch;

EE_AUDIO_TDMIN_A_MASK2 0xc4

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane2_mask, mask each channel in lane2, max is 32 ch;

EE_AUDIO_TDMIN_A_MASK3 0xc5

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane3_mask, mask each channel in lane3, max is 32 ch;



EE_AUDIO_TDMIN_A_STAT 0xc6

Bits	R/W	Default	Description
31:24	R/W	0x00000000	r_input_slot_cnt_max, the maxnum of slot_cnt
23:16	R/W		r_input_slot_cnt_min, the minnum of slot_cnt
14	R/W		overflow_flag, overflow flag of afifo
13:10	R/W		max_fifo_cnt, the maxnum of afifo_cnt
9:5	R/W		r_input_bit_cnt_max, the maxnum of bit_cnt
4:0	R/W		r_input_bit_cnt_min, the minnum of bit_cnt

EE_AUDIO_TDMIN_A_MUTE_VAL 0xc7

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_tmdin_a_mute_val, when mute , the channel value

EE_AUDIO_TDMIN_A_MUTE0 0xc8

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane0_mute, mute each channel in lane0, max is 32 ch;

EE_AUDIO_TDMIN_A_MUTE1 0xc9

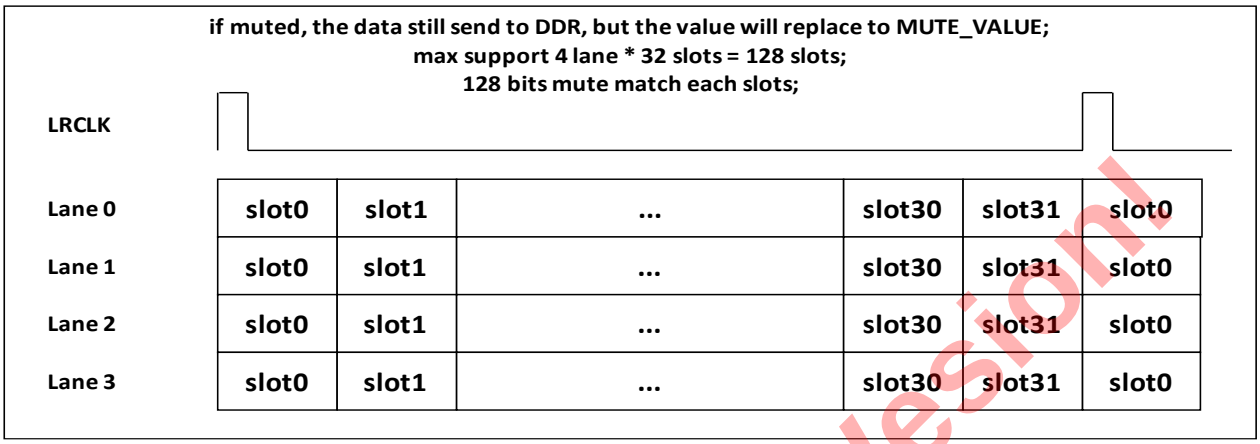
Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane1_mute, mute each channel in lane1, max is 32 ch;

EE_AUDIO_TDMIN_A_MUTE2 0xca

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane2_mute, mute each channel in lane2, max is 32 ch;

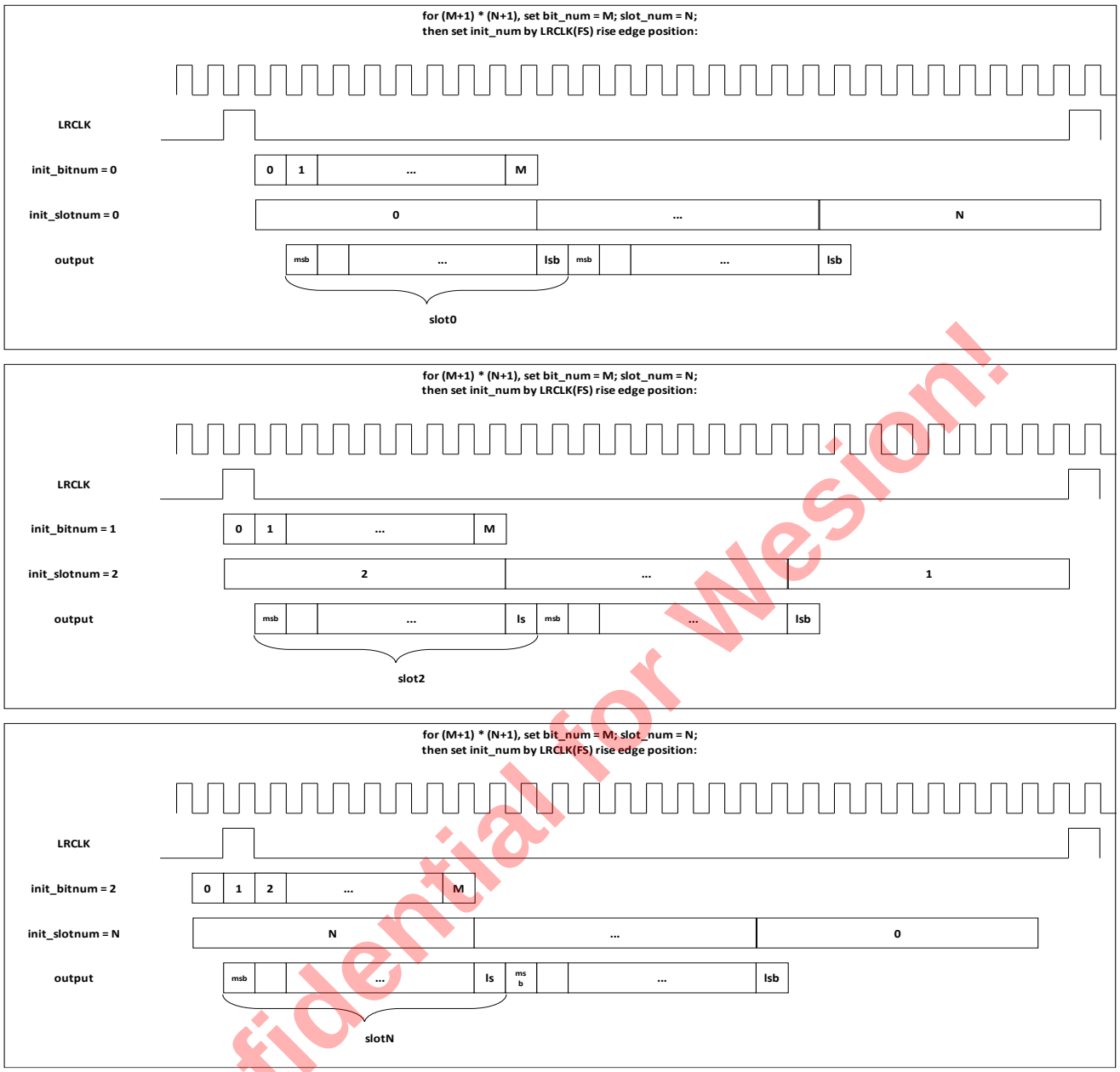
EE_AUDIO_TDMIN_A_MUTE3 0xcb

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane3_mute, mute each channel in lane3, max is 32 ch;



EE_AUDIO_TDMIN_B_CTRL 0xd0
EE_AUDIO_TDMIN_B_SWAP 0xd1
EE_AUDIO_TDMIN_B_MASK0 0xd2
EE_AUDIO_TDMIN_B_MASK1 0xd3
EE_AUDIO_TDMIN_B_MASK2 0xd4
EE_AUDIO_TDMIN_B_MASK3 0xd5
EE_AUDIO_TDMIN_B_STAT 0xd6
EE_AUDIO_TDMIN_B_MUTE_VAL 0xd7
EE_AUDIO_TDMIN_B_MUTE0 0xd8
EE_AUDIO_TDMIN_B_MUTE1 0xd9
EE_AUDIO_TDMIN_B_MUTE2 0xda
EE_AUDIO_TDMIN_B_MUTE3 0xdb
EE_AUDIO_TDMIN_C_CTRL 0xe0
EE_AUDIO_TDMIN_C_SWAP 0xe1
EE_AUDIO_TDMIN_C_MASK0 0xe2
EE_AUDIO_TDMIN_C_MASK1 0xe3
EE_AUDIO_TDMIN_C_MASK2 0xe4
EE_AUDIO_TDMIN_C_MASK3 0xe5
EE_AUDIO_TDMIN_C_STAT 0xe6
EE_AUDIO_TDMIN_C_MUTE_VAL 0xe7
EE_AUDIO_TDMIN_C_MUTE0 0xe8
EE_AUDIO_TDMIN_C_MUTE1 0xe9
EE_AUDIO_TDMIN_C_MUTE2 0xea
EE_AUDIO_TDMIN_C_MUTE3 0xeb
EE_AUDIO_TDMIN_LB_CTRL 0xf0
EE_AUDIO_TDMIN_LB_SWAP 0xf1
EE_AUDIO_TDMIN_LB_MASK0 0xf2
EE_AUDIO_TDMIN_LB_MASK1 0xf3
EE_AUDIO_TDMIN_LB_MASK2 0xf4
EE_AUDIO_TDMIN_LB_MASK3 0xf5
EE_AUDIO_TDMIN_LB_STAT 0xf6
EE_AUDIO_TDMIN_LB_MUTE_VAL 0xf7
EE_AUDIO_TDMIN_LB_MUTE0 0xf8
EE_AUDIO_TDMIN_LB_MUTE1 0xf9
EE_AUDIO_TDMIN_LB_MUTE2 0xfa
EE_AUDIO_TDMIN_LB_MUTE3 0xfb
EE_AUDIO_TDMOUT_A_CTRL0 0x140

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_work_enable, 0:disable; 1:enable;
29	R/W		reg_rst_afifo_out_n, reset afifo out side;
28	R/W		reg_rst_afifo_in_n, reset afifo in side; need set 1 after set reg_rst_afifo_out_n to 1;
27:24	R/W		force_oe[3:0]; 1: oe = force_val; 0: oe = mask_val;
23:20	R/W		reg_tdm_lr_mix, [23]: mix l3 and r3;[22]: mix l2 and r2;[21]: mix l1 and r1;[20]: mix l0 and r0;
19:15	R/W		reg_tdm_init_bitnum, initial count value of bitcnt
14:10	R/W		reg_tdm_init_slotnum, initial count value of slotcnt
9:5	R/W		reg_tdmout_slot_num , max value of slotcnt; if each frame has 8 slots, set it to 7;
4:0	R/W		reg_tdmout_bit_num ,max value of bitcnt; if each slot has 16bits, set it to 15;



EE_AUDIO_TDMOUT_A_CTRL1 0x141

Bits	R/W	Default	Description
31	R/W	0x00000000	eq_drc_sel; 1: select eq_drc output;
30	R/W		reg_debug_en,0:disable debug feature; 1: enable;
29	R/W		reg_out_lsb_first, 0: msb first; 1: lsb first;
28	R/W		reg_rev_ws_in, revert ws; 0: disable; 1: enable;
27	R/W		reg_rev_dat, revert data; 0: disable; 1: enable;
26	R/W		reg_gain_en, 0:disable; 1: enable data * gain;

Bits	R/W	Default	Description
25:24	R/W		reg_frddr_sel, 0:frddr_A;1:frddr_B;2:frddr_C;
23:16	R/W		reg_wait_cnt, wait some time when enable set to 1;then start request data from frddr;
12:8	R/W		reg_frddr_msb, msb position of data
6:4	R/W		reg_frddr_type, 0: split 64bits ddr data to 8 sample, each sample need 8 bits; if bitwidth < 8, left-justified; 1: split 64bits ddr data to 4 sample, each sample need 16 bits; if bitwidth < 16, left-justified ; 2: split 64bits ddr data to 4 sample, each sample need 16 bits; if bitwidth < 16, right-justified ; 3: split 64bits ddr data to 2 sample, each sample need 32 bits; if bitwidth < 32, left-justified; 4: split 64bits ddr data to 2 sample, each sample need 32 bits; if bitwidth < 32, right-justified;
3:0	R/W		force_oe_val[3:0];

For position of sample in DDR:

	ddr_data [7:0] (byte)							
type	byte7	byte6	byte5	byte4	byte3	byte2	byte1	byte0
0	s7	s6	s5	s4	s3	s2	s1	s0
	s15	s14	s13	s12	s11	s10	s9	s8
1/2	s3		s2		s1		s0	
	s7		s6		s5		s4	
3/4	s1				s0			
	s3				s2			

For encoder, it will start output[31] first, then output[30], finished at output[32-bit_num];

If lsb_first, will start s[0], then s[1], finished at s[bit_num+1];

	output [31:0] (bit)																														
type	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	s[7]	s[6]	s[5]	s[4]	s[3]	s[2]	s[1]	s[0]	0																							
1	s[15]	s[14]	s[13]	s[12]	s[11]	s[10]	s[9]	s[8]	s[7]	s[6]	s[5]	s[4]	s[3]	s[2]	s[1]	s[0]	0															
2	s[msb]	s[msb-1]	...			s[1]	s[0]	0																								
3	s[31]	s[30]	s[29]	s[28]	s[27]	s[26]	s[25]	s[24]	s[23]	s[22]	s[21]	s[20]	s[19]	s[18]	s[17]	s[16]	s[15]	s[14]	s[13]	s[12]	s[11]	s[10]	s[9]	s[8]	s[7]	s[6]	s[5]	s[4]	s[3]	s[2]	s[1]	s[0]
4	s[msb]	s[msb-1]	...																			s[1]	s[0]	0								

EE_AUDIO_TDMOUT_A_SWAP 0x142

Bits	R/W	Default	Description
30:28	R/W	0x00000000	lane3 right channel, 0: ch0;
26:24	R/W		lane3 left channel, 1: ch1;
22:20	R/W		lane2 right channel, 2: ch2;
18:16	R/W		lane2 left channel, 3: ch3;
14:12	R/W		lane1 right channel, 4: ch4;
10:8	R/W		lane1 left channel, 5: ch5;
6:4	R/W		lane0 right channel, 6: ch6;
2:0	R/W		lane0 left channel, 7: ch7;

EE_AUDIO_TDMOUT_A_MASK0 0x143

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane0_mask, mask each channel in lane0, max is 32 ch;

EE_AUDIO_TDMOUT_A_MASK1 0x144

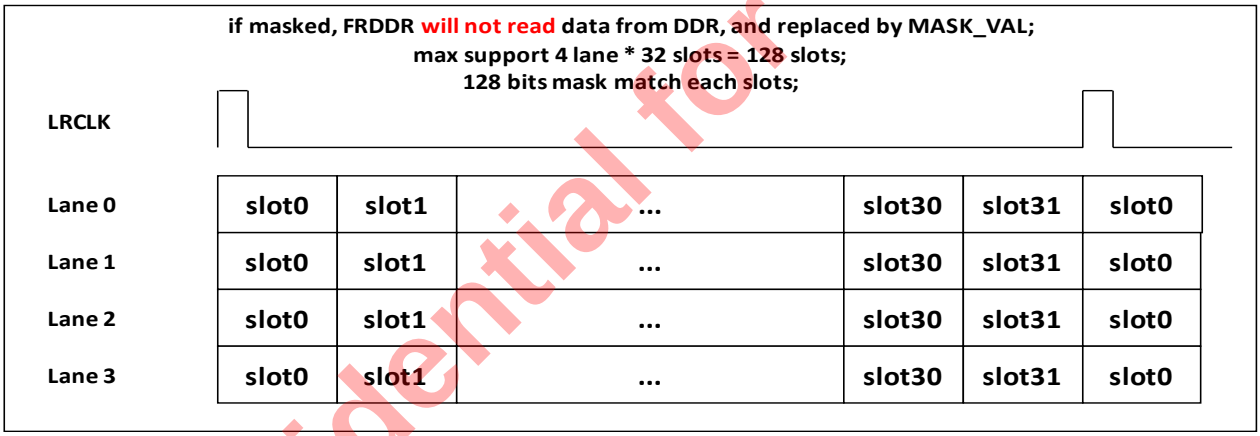
Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane1_mask, mask each channel in lane1, max is 32 ch;

EE_AUDIO_TDMOUT_A_MASK2 0x145

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane2_mask, mask each channel in lane2, max is 32 ch;

EE_AUDIO_TDMOUT_A_MASK3 0x146

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane3_mask, mask each channel in lane3, max is 32 ch;



EE_AUDIO_TDMOUT_A_STAT 0x147

Bits	R/W	Default	Description
31:29	R/W	0x00000000	fifo_cnt, afifo cnt
28	R/W		up_error, change to 1 if overflow
27:24	R/W		req_frdd_fsm_stat
21	R/W		r_slot_cnt_err
20	R/W		r_bit_cnt_err
16:12	R/W		r_max_slot_cnt
8:4	R/W		r_max_bit_cnt
3	R/W		r_out_en

Bits	R/W	Default	Description
2	R/W		r_out_en_pre
1	R/W		r_first_fs
0	R/W		c_frdd_init_finish

EE_AUDIO_TDMOUT_A_GAIN0 0x148

Bits	R/W	Default	Description
31:24	R/W	0x00000000	gain_ch3
23:16	R/W		gain_ch2
15:8	R/W		gain_ch1
7:0	R/W		gain_ch0

EE_AUDIO_TDMOUT_A_GAIN1 0x149

Bits	R/W	Default	Description
31:24	R/W	0x00000000	gain_ch7
23:16	R/W		gain_ch6
15:8	R/W		gain_ch5
7:0	R/W		gain_ch4

EE_AUDIO_TDMOUT_A_MUTE_VAL 0x14a

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_tmdin_a_mute_val, when mute , the channel value

EE_AUDIO_TDMOUT_A_MUTE0 0x14b

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane0_mute, mute each channel in lane0, max is 32 ch;

EE_AUDIO_TDMOUT_A_MUTE1 0x14c

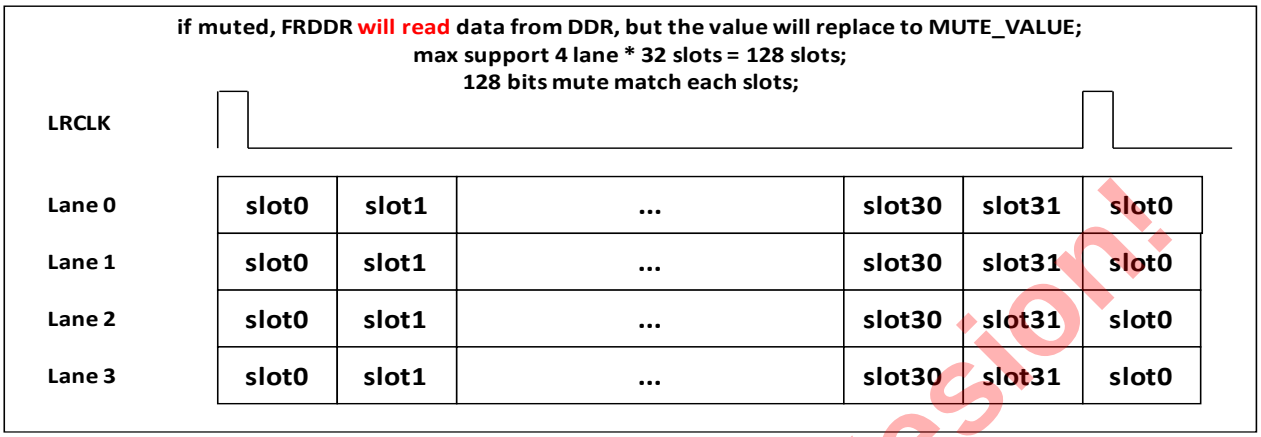
Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane1_mute, mute each channel in lane1, max is 32 ch;

EE_AUDIO_TDMOUT_A_MUTE2 0x14d

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane2_mute, mute each channel in lane2 max is 32 ch;

EE_AUDIO_TDMOUT_A_MUTE3 0x14e

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_lane3_mute, mute each channel in lane3, max is 32 ch;



EE_AUDIO_TDMOUT_A_MASK_VAL 0x14f

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_tdmout_mask_val, when masked, the channel value

EE_AUDIO_TDMOUT_B_CTRL0 0x150
EE_AUDIO_TDMOUT_B_CTRL1 0x151
EE_AUDIO_TDMOUT_B_SWAP 0x152
EE_AUDIO_TDMOUT_B_MASK0 0x153
EE_AUDIO_TDMOUT_B_MASK1 0x154
EE_AUDIO_TDMOUT_B_MASK2 0x155
EE_AUDIO_TDMOUT_B_MASK3 0x156
EE_AUDIO_TDMOUT_B_STAT 0x157
EE_AUDIO_TDMOUT_B_GAIN0 0x158
EE_AUDIO_TDMOUT_B_GAIN1 0x159
EE_AUDIO_TDMOUT_B_MUTE_VAL 0x15a
EE_AUDIO_TDMOUT_B_MUTE0 0x15b
EE_AUDIO_TDMOUT_B_MUTE1 0x15c
EE_AUDIO_TDMOUT_B_MUTE2 0x15d
EE_AUDIO_TDMOUT_B_MUTE3 0x15e
EE_AUDIO_TDMOUT_B_MASK_VAL 0x15f
EE_AUDIO_TDMOUT_C_CTRL0 0x160
EE_AUDIO_TDMOUT_C_CTRL1 0x161
EE_AUDIO_TDMOUT_C_SWAP 0x162
EE_AUDIO_TDMOUT_C_MASK0 0x163
EE_AUDIO_TDMOUT_C_MASK1 0x164
EE_AUDIO_TDMOUT_C_MASK2 0x165
EE_AUDIO_TDMOUT_C_MASK3 0x166
EE_AUDIO_TDMOUT_C_STAT 0x167
EE_AUDIO_TDMOUT_C_GAIN0 0x168
EE_AUDIO_TDMOUT_C_GAIN1 0x169
EE_AUDIO_TDMOUT_C_MUTE_VAL 0x16a
EE_AUDIO_TDMOUT_C_MUTE0 0x16b
EE_AUDIO_TDMOUT_C_MUTE1 0x16c
EE_AUDIO_TDMOUT_C_MUTE2 0x16d
EE_AUDIO_TDMOUT_C_MUTE3 0x16e
EE_AUDIO_TDMOUT_C_MASK_VAL 0x16f

11.9.8 SPDIFRegisters

EE_AUDIO_SPDIFIN_CTRL0 0x100

Bits	R/W	Default	Description

31	R/W	0x00000000	reg_work_enable, 0: disable; 1:enable;
29	R/W		reg_rst_afifo_out_n, reset afifo out side;
28	R/W		reg_rst_afifo_in_n, reset afifo in side; need set 1 after set reg_rst_afifo_out_n to 1;
27	R/W		reg_debug_en, 0:disable debug; 1: enable;
26	R/W		reg_chunm_en; 1: start add ch_cnt to ch_num;
25	R/W		reg_findpapb_en, 0: disable NonPCM mode; 1: enable;
24	R/W		reg_width_sel, 0: detect sample mode by max_width;1: detect sample mode by min_width;
23:12	R/W		reg_nonpcm2pcm_th, when detected NonPcm mode;if long time (z_cnt >= th) didn't detect PaPb again, will generate interrupt to SW: now changed to PCM mode;
11:8	R/W		reg_ch_status_sel, For EE_AUDIO_SPDIFIN_STAT1
7	R/W		reg_mute_l, 0: disable ; 1: mute channel L;
6	R/W		reg_mute_r, 0: disable ; 1: mute channel R;
5:4	R/W		reg_spdifin_src_sel, 0: PAD of spdifin;1: spdifout;
3	R/W		reg_check_valid, 0: disable valid check ; 1: enable;
2	R/W		reg_check_parity, 0: disable parity check ; 1: enable;
1	R/W		reg_invert_data, 0: disalbe; 1: invert [27:4] to [4:27];
0	R/W		reg_spdifin_phase, 0: disable invert; 1: enable;

EE_AUDIO_SPDIFIN_CTRL1 0x101

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_force_sample_mode, 0: auto detect sample mode; 1: force a fixed sample mode;
30:28	R/W		reg_sample_mode
27:20	R/W		reg_interrupt_mask, mask each interrupt;
19:0	R/W		reg_base_timer, define a base timer to detect sample mode changed;

EE_AUDIO_SPDIFIN_CTRL2 0x102

Bits	R/W	Default	Description
29:20	R/W	0x00000000	reg_sample_mode0_timer_th
19:10	R/W		reg_sample_mode1_timer_th
9:0	R/W		reg_sample_mode2_timer_th

EE_AUDIO_SPDIFIN_CTRL3 0x103

Bits	R/W	Default	Description
29:20	R/W	0x00000000	reg_sample_mode3_timer_th
19:10	R/W		reg_sample_mode4_timer_th
9:0	R/W		reg_sample_mode5_timer_th

EE_AUDIO_SPDIFIN_CTRL4 0x104

Bits	R/W	Default	Description
31:24	R/W	0x00000000	reg_sample_mode0_timer
23:16	R/W		reg_sample_mode1_timer
15:8	R/W		reg_sample_mode2_timer
7:0	R/W		reg_sample_mode3_timer

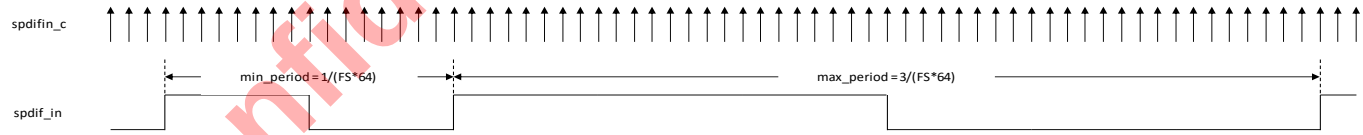
EE_AUDIO_SPDIFIN_CTRL5 0x105

Bits	R/W	Default	Description
31:24	R/W	0x00000000	reg_sample_mode4_timer
23:16	R/W		reg_sample_mode5_timer
15:8	R/W		reg_sample_mode6_timer

We need automatically detect the frequency of sample rate (FS).

It will done by search min_period or max_period of SPDIF IN and compared to thresholds.

We can support 7 mode settings.



For example:

If spdifin_clk is 166MHz, and we want support 48KHz and 96KHz.

The min_period of 48KHz is 1/(48000*64) and the counter number by 166MHz is 166000000/(48000*64) = 54;

96KHz counter number is 166000000/(96000*64) = 27;

Then set:

Width_sel = 1;

Mode0_th = (54+27)/2 = 41;

Mode1_th = 0;

Mode0_timer = 27;

Mode1_timer = 13;

Or:

Width_sel = 0;

Mode0_th = (162+81)/2 = 121;

Mode1_th = 0;

Mode0_timer = 27;

Mode1_timer = 13;

More example:

spdifin_clk = 166MHz								
			width_sel = 1			width_sel = 0		
	FS(Hz)	timer	min_period			max_period		
mode0	24000	54	108.0729167	th0	95	324.21875	th0	283
mode1	32000	40	81.0546875	th1	70	243.1640625	th1	209
mode2	44100	29	58.81519274	th2	58	176.4455782	th2	172
mode3	46000	28	56.38586957	th3	55	169.1576087	th3	165
mode4	48000	27	54.03645833	th4	41	162.109375	th4	121
mode5	96000	13	27.01822917	th5	20	81.0546875	th5	60
mode6	192000	6	13.50911458			40.52734375		
spdifin_clk = 250MHz								
			width_sel = 1			width_sel = 0		
	FS(Hz)	timer	min_period			max_period		
mode0	24000	81	162.7604167	th0	142	488.28125	th0	427
mode1	32000	61	122.0703125	th1	105	366.2109375	th1	315
mode2	44100	44	88.57709751	th2	87	265.7312925	th2	260
mode3	46000	42	84.91847826	th3	83	254.7554348	th3	249
mode4	48000	40	81.38020833	th4	61	244.140625	th4	183
mode5	96000	20	40.69010417	th5	31	122.0703125	th5	91
mode6	192000	10	20.34505208			61.03515625		
spdifin_clk = 400MHz								
			width_sel = 1			width_sel = 0		

	FS(Hz)	timer	min_period			max_period		
mode0	24000	130	260.4166667	th0	228	781.25	th0	683
mode1	32000	97	195.3125	th1	169	585.9375	th1	505
mode2	44100	70	141.723356	th2	139	425.170068	th2	416
mode3	46000	67	135.8695652	th3	133	407.6086957	th3	399
mode4	48000	65	130.2083333	th4	98	390.625	th4	292
mode5	96000	32	65.10416667	th5	49	195.3125	th5	146
mode6	192000	16	32.55208333			97.65625		

EE_AUDIO_SPDIFIN_CTRL6

Bits	R/W	Default	Description
23:16	R/W	0x00000000	Reg_clr_interrupt[7:0] for each bit of irq_status[7:0];
8	R/W		reg_papb_ext_sync, 1: add ext "0" sync check for papb; 0: disable;
7:0	R/W		reg_papb_ext_mask, mask 8 channel "0" sync

EE_AUDIO_SPDIFIN_STAT0 0x107

Bits	R/W	Default	Description
30:28	R/W	0x00000000	r_sample_mode, current sample mode;
27:18	R/W		r_width_min, the min width of two edge;
17:8	R/W		r_width_max, the max width of two edge;
7:0	R/W		r_interrupt_status, interrupt status, need clear by reg_clk_interrupt; [7]: find PaPb; [6]: valid changed; [5]: find nonpcm to pcm (reg_nonpcm2pcm_th) ; [4]: find Pd changed; [3]: find Pc changed; [2]: find sample mode changed; [1]: find parity error; [0]: find overflow;

EE_AUDIO_SPDIFIN_STAT1 0x108

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel status, by reg_ch_status_sel; reg_ch_status_sel[3]: 0: channel A; 1: channel B; reg_ch_status_sel[2:0]: 0: ch_status[31:0]; 1: ch_status[63:32]; 2: ch_status[95:64]; 3: ch_status[127:96]; 4: ch_status[159:128]; 5: ch_status[191:160];

EE_AUDIO_SPDIFIN_STAT2 0x109

Bits	R/W	Default	Description
31:0	R/W	0x00000000	debug status, by reg_ch_status_sel; 0: r_z_width, the width of two preamble Z; 1: {16'd0, frame_cnt_min[7:0], frame_cnt_max[7:0]}; 2: {6'd0,width_min[9:0],6'd0, width_max[9:0]};

EE_AUDIO_SPDIFIN_MUTE_VAL 0x10a

Bits	R/W	Default	Description
31:0	R/W	0x00000000	reg_spdifin_mute_val, when muted, the channel value

EE_AUDIO_SPDIFOUT_STAT 0x120

Bits	R/W	Default	Description
7:5	R/W	0x00000000	fifo_cnt, afifo cnt
4	R/W		up_error, change to 1 if overflow
3:0	R/W		req_frdd_fsm_stat

EE_AUDIO_SPDIFOUT_GAIN0 0x121

Bits	R/W	Default	Description
31:24	R/W	0x00000000	gain_ch3
23:16	R/W		gain_ch2
15:8	R/W		gain_ch1
7:0	R/W		gain_ch0

EE_AUDIO_SPDIFOUT_GAIN1 0x122

Bits	R/W	Default	Description
31:24	R/W	0x00000000	gain_ch7
23:16	R/W		gain_ch6
15:8	R/W		gain_ch5
7:0	R/W		gain_ch4

EE_AUDIO_SPDIFOUT_CTRL0 0x123

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_work_enable, 0: disable ; 1: enable;
29	R/W		reg_rst_afifo_out_n, reset afifo out side;

Bits	R/W	Default	Description
28	R/W		reg_rst_afifo_in_n, reset afifo in side; need set 1 after set reg_rst_afifo_out_n to 1;
27	R/W		reg_hold_start_en; 1: add delay to match TDM out when share buff;
26	R/W		reg_userdata_sel, 0: "user data " = data [29]; 1: "user data" = reg_userdata_set;
25	R/W		reg_userdata_set
24	R/W		reg_chdata_sel, 0: "ch status" = data [30]; 1: "ch status" = reg_chsts0~B
23	R/W		reg_mix_lr, 0: disable; 1: L = (L+R)/2; R = (L+R)/2;
22	R/W		reg_mute_l, 0: disable; 1: ch_l_data = reg_mute_val;
21	R/W		reg_mute_r, 0: disable; 1: ch_r_data = reg_mute_val;
20	R/W		reg_data_sel, 0: insert data from 27bits; 1: insert data from 31bits;
19	R/W		reg_out_msb_first, 0: lsb first; 1: msb first;
18	R/W		reg_valid_sel, 0: "valid flag" = data [28]; 1: "valid flag" = reg_valid_set;
17	R/W		reg_valid_set
11:4	R/W		reg_mask, [11:10]: mask lane3 L/R; [9:8]: mask lane2 L/R; [7:6]: mask lane1 L/R; [5:4]: mask lane0 L/R;
3:0	R/W		reg_parity_mask, [0]: initial parity value;

EE_AUDIO_SPDIFOUT_CTRL1 0x124

Bits	R/W	Default	Description
21	R/W	0x00000000	eq_drc_sel; 1: select eq_drc data;
26	R/W		reg_gain_en, 0:disable; 1: enable data * gain;
25:24	R/W		reg_frddr_sel, 0:frddr_A; 1:frddr_B; 2:frddr_C;
23:16	R/W		reg_wait_cnt, wait some time when enable set to 1; then start request data from frddr;
12:8	R/W		reg_frddr_msb, msb position of data

6:4	R/W		reg_frddr_type, 0: split 64bits ddr data to 8 sample, each sample need 8 bits; if bitwidth < 8, left-justified; 1: split 64bits ddr data to 4 sample, each sample need 16 bits; if bitwidth < 16, left-justified ; 2: split 64bits ddr data to 4 sample, each sample need 16 bits; if bitwidth < 16, right-justified ; 3: split 64bits ddr data to 2 sample, each sample need 32 bits; if bitwidth < 32, left-justified; 4: split 64bits ddr data to 2 sample, each sample need 32 bits; if bitwidth < 32, right-justified;

EE_AUDIO_SPDIFOUT_PREAMB 0x125

Bits	R/W	Default	Description
31	R/W	0x00000000	set preamble Z, 0: preamble Z = 8'b11101000; 1: reg[7:0];
20	R/W		set preamble Y, 0: preamble Z = 8'b11100100; 1: reg[15:8];
29	R/W		set preamble X, 0: preamble Z = 8'b11100010; 1: reg[23:16];
23:16	R/W		preamble X
15:8	R/W		preamble Y
7:0	R/W		preamble Z

EE_AUDIO_SPDIFOUT_SWAP 0x126

Bits	R/W	Default	Description
31:16	R/W	0x00000000	hold start cnt, work when CTRL0[27] = 1;
6:4	R/W		lane0 right ch sel, 0: ch0; 1: ch1; 2: ch2; 3: ch3; 4: ch4; 5: ch5; 6: ch6; 7: ch7;
2:0	R/W		lane0 left ch sel, 0: ch0; 1: ch1; 2: ch2; 3: ch3; 4: ch4; 5: ch5; 6: ch6; 7: ch7;

EE_AUDIO_SPDIFOUT_CHSTS0 0x127

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel A status[31:0]

EE_AUDIO_SPDIFOUT_CHSTS1 0x128

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel A status[63:32]

EE_AUDIO_SPDIFOUT_CHSTS2 0x129

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel A status[95:64]

EE_AUDIO_SPDIFOUT_CHSTS3 0x12a

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel A status[127:96]

EE_AUDIO_SPDIFOUT_CHSTS4 0x12b

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel A status[159:128]

EE_AUDIO_SPDIFOUT_CHSTS5 0x12cv

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel A status[191:160]

EE_AUDIO_SPDIFOUT_CHSTS6 0x12d

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel B status[31:0]

EE_AUDIO_SPDIFOUT_CHSTS7 0x12e

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel B status[63:32]

EE_AUDIO_SPDIFOUT_CHSTS8 0x12f

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel B status[95:64]

EE_AUDIO_SPDIFOUT_CHSTS9 0x130

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel B status[127:96]

EE_AUDIO_SPDIFOUT_CHSTSA 0x131

Bits	R/W	Default	Description

31:0	R/W	0x00000000	channel B status[159:128]

EE_AUDIO_SPDIFOUT_CHSTSB 0x132

Bits	R/W	Default	Description
31:0	R/W	0x00000000	channel B status[191:160]

EE_AUDIO_SPDIFOUT_MUTE_VAL 0x133

EE_AUDIO_SPDIFOUT_B_STAT 0x1a0

EE_AUDIO_SPDIFOUT_B _GAIN0 0x1a1

EE_AUDIO_SPDIFOUT_B _GAIN1 0x1a2

EE_AUDIO_SPDIFOUT_B _CTRL0 0x1a3

EE_AUDIO_SPDIFOUT_B _CTRL1 0x1a4

EE_AUDIO_SPDIFOUT_B _PREAMB 0x1a5

EE_AUDIO_SPDIFOUT_B _SWAP 0x1a6

EE_AUDIO_SPDIFOUT_B _CHSTS0 0x1a7

EE_AUDIO_SPDIFOUT_B _CHSTS1 0x1a8

EE_AUDIO_SPDIFOUT_B _CHSTS2 0x1a9

EE_AUDIO_SPDIFOUT_B _CHSTS3 0x1aa

EE_AUDIO_SPDIFOUT_B _CHSTS4 0x1ab

EE_AUDIO_SPDIFOUT_B _CHSTS5 0x1ac

EE_AUDIO_SPDIFOUT_B _CHSTS6 0x1ad

EE_AUDIO_SPDIFOUT_B _CHSTS7 0x1ae

EE_AUDIO_SPDIFOUT_B _CHSTS8 0x1af

EE_AUDIO_SPDIFOUT_B _CHSTS9 0x1b0

EE_AUDIO_SPDIFOUT_B _CHSTSA 0x1b1

EE_AUDIO_SPDIFOUT_B _CHSTSB 0x1b2

EE_AUDIO_SPDIFOUT_B _MUTE_VAL 0x1b3

11.9.9 Resample Registers

EE_AUDIO_RESAMPLE_CTRL0 0x110

Bits	R/W	Default	Description
31	R/W	0x00000000	Soft_reset
30:29	R/W		Channel_sel
28	R/W		Enable,
27:26	R/W		Method_sel,
25:16	R/W		outrdy_Cnt_ctrl[9:0],

15:0	R/W		Avg_cnt_init,

EE_AUDIO_RESAMPLE_CTRL1 0x111

Bits	R/W	Default	Description
31:0	R/W	0x00000000	Phase_step,It is used to set the phase step of the accumulator. It is equal to fs_in/fs_out*(1<<28).

EE_AUDIO_RESAMPLE_CTRL2 0x112

Bits	R/W	Default	Description
30:28	R/W	0x00000000	outrdy_Cnt_ctrl[12:10],
27	R/W		resample_start_mode;
25	R/W		IIR filter enable
24	R/W		Pause_en
23:0	R/W		Pause_cnt_thd

EE_AUDIO_RESAMPLE_CTRL3 0x113

Bits	R/W	Default	Description
14:12	R/W	0x00000000	resample_st_cnt;
11:8	R/W		Reg_ch_num_sel, The channel number of input
4:0	R/W		Reg_in_msb, the msb of input[31:0]

EE_AUDIO_RESAMPLE_COEF0 0x114

Bits	R/W	Default	Description
25:0	R/W	0x00000000	IIR filter coef0

EE_AUDIO_RESAMPLE_COEF1 0x115

Bits	R/W	Default	Description
25:0	R/W	0x00000000	IIR filter coef1

EE_AUDIO_RESAMPLE_COEF2 0x116

Bits	R/W	Default	Description
25:0	R/W	0x00000000	IIR filter coef2

EE_AUDIO_RESAMPLE_COEF3 0x117

Bits	R/W	Default	Description
25:0	R/W	0x00000000	IIR filter coef3

EE_AUDIO_RESAMPLE_COEF4 0x118

Bits	R/W	Default	Description
25:0	R/W	0x00000000	IIR filter coef4

EE_AUDIO_RESAMPLE_STATUS1 0x119

Bits	R/W	Default	Description
24	R/W	0x00000000	The pause status
21:0	R/W		The real frequency of input data

11.9.10 Power Detect Registers

EE_AUDIO_POW_DET_CTRL0 0x180

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_pow_det_en, 0:disable;1:enable;
30			reg_downsample_en, 0:disable;1:enable;
29			reg_ext_signed, 1:extend signed bit;
28			reg_fast_ram_en, 0:disable(depth=1024);1:enable;
27			reg_sync_ch_en; 1: only select the ch data which ch_num can match ID;
23:20			reg_ram_sel, ram depth = sel * 64;
19:16			reg_downsample_sel, downsample rate = (1<<sel);
15:13			reg_dat_sel, 0: combined data[m:n] without gap; like S0[m:n],S1[m:n],S2[m:n],… 1: combined data[m:n] as 16bits; like {S0[11:0],4'd0},{S1[11:0],4'd0}… 2: combined data[m:n] as 16bits; like {4'd0,S0[11:0]},{4'd0,{S1[11:0]}… 3: combined data[m:n] as 32bits; like {S0[27:4],8'd0},{S1[27:4],8'd0}… 4: combined data[m:n] as 32bits; like {8'd0,S0[27:4]},{8'd0,{S1[27:4]}…
12:8			reg_dat_m_sel, the msb positioin in data
7:3			reg_dat_n_sel, the lsb position in data
2:0			reg_dat_src_sel, [7]: loopback; [6]: tdmin_lb; [5]: reserved; [4]: pdmin; [3]: spdifin; [2]: tdmin_c; [1]: tdmin_b; [0]: tdmin_a;

EE_AUDIO_POW_DET_CTRL1 0x181

Bits	R/W	Default	Description
31:24	R/W	0x00000000	reg_ch3_id

23:16			reg_ch2_id
15:8			reg_ch1_id
7:0			reg_ch0_id

EE_AUDIO_POW_DET_TH_HI 0x182

Bits	R/W	Default	Description
25:0	R/W	0x00000000	hi_th, if acc_value > hi_th, status will be high

EE_AUDIO_POW_DET_TH_LO 0x183

Bits	R/W	Default	Description
25:0	R/W	0x00000000	lo_th, if acc_value < lo_th, status will be low

EE_AUDIO_POW_DET_VALUE 0x184

Bits	R/W	Default	Description
31	R/W	0x00000000	r_status, 0: low status; 1: high status;
25:0			r_acc_data, acc_value

11.9.11 TORAM Registers

EE_AUDIO_TORAM_CTRL0 0x1c0

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_toddr_en, 0: disable; 1: enable;
30	R/W		Reserved ;
29	R/W		reg_ext_signed, 0: select write to only one buff (start_addr,finish_addr); 1: select write to two buff (start_addr,finish_addr) (start_addrb, finish_addrb);
28	R/W		Reserved ;
27	R/W		Enable_sync_chnum; 1: start store data when first ch ;
26:24	R/W		Reserved ;
23:16	R/W		[23:18] : reserved; [17] : when write to ddr address match "int_addr"; [16]: when write to ddr address match "finish_addr";
15:13	R/W		reg_toddr_sel, 0: combined data[m:n] without gap; like S0[m:n],S1[m:n],S2[m:n],… 1: combined data[m:n] as 16bits; like {S0[11:0],4'd0},{S1[11:0],4'd0}… 2: combined data[m:n] as 16bits; like {4'd0,S0[11:0]},{4'd0,{S1[11:0]}… 3: combined data[m:n] as 32bits; like {S0[27:4],8'd0},{S1[27:4],8'd0}… 4: combined data[m:n] as 32bits; like {8'd0,S0[27:4]},{8'd0,{S1[27:4]}…
12:8	R/W		reg_toddr_m_sel, the msb positioin in data

Bits	R/W	Default	Description
7:3	R/W		reg_toddr_n_sel, the lsb position in data
2:0	R/W		reg_toddr_src_sel, [7]: loopback; [6]: tdmin_lb; [5]: reserved; [4]: pdmin; [3]: spdifin; [2]: tdmin_c; [1]: tdmin_b; [0]: tdmin_a;

EE_AUDIO_TORAM_CTRL1 0x1c1

Bits	R/W	Default	Description
24	R/W	0x00000000	Insert_chnum; 0: disable; 1: insert chnum[9:0] to data[9:0]
23:16	R/W		Reserved ;
11:8	R/W		reg_status_sel, control status2 source;
7:0	R/W		reg_int_status_clr,clear each bits of int_status register

EE_AUDIO_TORAM_START_ADDR 0x1c2

Bits	R/W	Default	Description
31:0	R/W	0x00000000	start_addr, buff_A start address, ignore [2:0]

EE_AUDIO_TORAM_FINISH_ADDR 0x1c3

Bits	R/W	Default	Description
31:0	R/W	0x00000000	finish_addr, buff_A finish address, ignore [2:0]

EE_AUDIO_TORAM_INT_ADDR 0x1c4

Bits	R/W	Default	Description
31:0	R/W	0x00000000	int_addr, usage A : as an address of interrupt; usage B : as a count of interrupt;

EE_AUDIO_TORAM_STATUS1 0x1c5

Bits	R/W	Default	Description
7:0	R/W	0x00000000	int_status, when irq generate, related bit will changed to 1 and can only clear by reg_int_status_clr

EE_AUDIO_TORAM_STATUS2 0x1c6

Bits	R/W	Default	Description

31:0	R/W	0x00000000	status2, by reg_status_sel: 0: current write ram times; 1: current write ram address;

EE_AUDIO_TORAM_INIT_ADDR 0x1c7

Bits	R/W	Default	Description
31:0	R/W	0x00000000	first RAM address after enable set to 1;

11.9.12 TOACODEC Registers

EE_AUDIO_TOACODEC_CTRL0 0x1d0

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_toacodec_en, 0: disable; 1: enable;
15:12	R/W		dat_sel: 0: tdmout_a_dat[0]; 1: tdmout_a_dat[1]; 2: tdmout_a_dat[2]; 3: tdmout_a_dat[3]; 4: tdmout_b_dat[0]; 5: tdmout_b_dat[1]; 6: tdmout_b_dat[2]; 7: tdmout_b_dat[3]; 8: tdmout_c_dat[0]; 9: tdmout_c_dat[1]; 10: tdmout_c_dat[2]; 11: tdmout_c_dat[3];
9:8	R/W		lrclk_sel: 0: tdmout_a_lrclk; 1: tdmout_b_lrclk; 2: tdmout_c_lrclk;
7	R/W		Bclk_cap_inv: The dat_o and lrclk_o will captured for timing balance after select; If this bit set to 1, will use invert bclk to capture;
6	R/W		Bclk_o_inv: if set 1, the final bclk connect to acodec will invert;
5:4	R/W		Bclk_sel: 0: tdmout_a_bclk; 1: tdmout_b_bclk; 2: tdmout_c_bclk;

Bits	R/W	Default	Description
2:0	R/W		Mclk_sel: 0:mst_mclk_a; 1:mst_mclk_b; 2:mst_mclk_c; 3:mst_mclk_d; 4:mst_mclk_e; 5:mst_mclk_f;

11.9.13 TOHDMITX Registers

EE_AUDIO_TOHDMITX_CTRL0 0x1d0

Bits	R/W	Default	Description
31	R/W	0x00000000	reg_toacodec_en, 0: disable; 1: enable;
13:12	R/W		dat_sel: 0: tdmout_a_dat; 1: tdmout_b_dat; 2: tdmout_c_dat;
9:8	R/W		lrclk_sel: 0: tdmout_a_lrclk; 1: tdmout_b_lrclk; 2: tdmout_c_lrclk;
7	R/W		Bclk_cap_inv: The dat_o and lrclk_o will captured for timing balance after select; If this bit set to 1, will use invert bclk to capture;
6	R/W		Bclk_o_inv: if set 1, the final bclk connect to acodec will invert;
5:4	R/W		Bclk_sel: 0: tdmout_a_bclk; 1: tdmout_b_bclk; 2: tdmout_c_bclk;
3	R/W		Spdif_clk_cap_inv: The spdif_dat will captured for timing balance after select; If this bit set to 1, will use invert spdif_clk to capture;
2	R/W		Spdif_clk_o_inv: if set 1, the final bclk connect to hdmitx will invert;

Bits	R/W	Default	Description
1	R/W		Spdif_sel: 0:spdif_out; 1:spdif_out_b;
0	R/W		Spdif_clk_sel: 0:spdif_clk; 1:spdif_clk_b;

11.9.14 Audio Locker Register

Base Address: FF64A000

Each register final address = module base address+ address * 4

AUD_LOCK_EN 0X00

Bit(s)	R/W	Default	Description
31:1			reserved
0	R/W	0x0	Audio_lock_en: 0 = disable; 1= enable.

AUD_LOCK_SW_RESET 0X01

Bit(s)	R/W	Default	Description
31:1			reserved
0	R/W	0x0	Audio_lock_soft_reset: 1=generate soft reset pulse

AUD_LOCK_SW_LATCH 0X02

Bit(s)	R/W	Default	Description
31:4			reserved
3	R/W	0x0	omclk2ref software latch: 1= generate the latch pulse
2	R/W	0x0	Ref2omclk software latch: 1= generate the latch pulse
1	R/W	0x0	Ref2imclk software latch: 1= generate the latch pulse
0	R/W	0x0	Imclk2ref software latch: 1= generate the latch pulse

AUD_LOCK_HW_LATCH 0X03

Bit(s)	R/W	Default	Description
31:4			reserved
3	R/W	0x0	omclk2ref hardware latch enble: 1= enable；0= disable

2	R/W	0x0	Ref2omclk hardware latch enble: 1= enable；0= disable
1	R/W	0x0	Ref2imclk hardware latch enble: 1= enable；0= disable
0	R/W	0x0	Imclk2ref hardware latch enble: 1= enable；0= disable

AUD_LOCK_REFCLK_SRC 0X04

Bit(s)	R/W	Default	Description
31:2			reserved
1:0	R/W	0x0	Ref clk source sel: 0= pclk; 1=oscinclk; 2,3,reserved

AUD_LOCK_REFCLK_LAT_INT 0X05

Bit(s)	R/W	Default	Description
31:0			U32 number of reference clock cycles to latch the imclk and omclk

AUD_LOCK_IMCLK_LAT_INT 0X06

Bit(s)	R/W	Default	Description
31:0			U32 number of imclk clock cycles to latch the reference clock

AUD_LOCK_OMCLK_LAT_INT 0X07

Bit(s)	R/W	Default	Description
31:0			U32 number of omclk clock cycles to latch the reference clock

AUD_LOCK_REFCLK_DS_INT 0X08

Bit(s)	R/W	Default	Description
31:10			reserved
9:0	R/W	0x0	U10 downsample step of reference clock for the couter48ds to be measured, module = x+1

AUD_LOCK_IMCLK_DS_INT 0X09

Bit(s)	R/W	Default	Description
31:10			reserved
9:0	R/W	0x0	U10 downsample step of imclk for the couter48ds to be measured, module = x+1

AUD_LOCK_OMCLK_DS_INT 0X0a

Bit(s)	R/W	Default	Description
31:10			reserved
9:0	R/W	0x0	U10 downsample step of omclk for the couter48ds to be measured, module = x+1

AUD_LOCK_INT_CLR 0X0b

Bit(s)	R/W	Default	Description
31:2			reserved
1	R/W	0x0	It is used to generate pulse to clear the interrupt status
0	R/W	0x0	It is used to generate pulse to clear the interrupt

AUD_LOCK_GCLK_CTRL 0X0c

Bit(s)	R/W	Default	Description
31:4			reserved
3:2	R/W	0x0	It is used to gate the module clock
1	R/W	0x0	It is used to gate the register clock
0			reserved

AUD_LOCK_INT_CTRL 0X0d

Bit(s)	R/W	Default	Description
31:4			reserved
3	R/W	0x0	It is used to mask interrupt for omclk_state
2	R/W	0x0	It is used to mask interrupt for imclk_state
1	R/W	0x0	It is used to mask interrupt for refclk_state1
0	R/W	0x0	It is used to mask interrupt for refclk_state0

RO_REF2IMCLK_CNT_L 0X10

Bit(s)	R/W	Default	Description
31:0	R	0x0	U48 latched imclk counter48ds of each reg_refclk_latch_interval; Will start from 0 reaching 2^48

RO_REF2IMCLK_CNT_H 0X11

Bit(s)	R/W	Default	Description
31:16			reserved
15:0	R	0x0	U48 latched imclk counter48ds of each reg_refclk_latch_interval; Will start from 0 reaching 2^48

RO_REF2OMCLK_CNT_L 0X12

Bit(s)	R/W	Default	Description
31:0	R	0x0	U48 latched omclk counter48ds of each reg_refclk_latch_interval; Will start from 0 reaching 2^48

RO_REF2OMCLK_CNT_H 0X13

Bit(s)	R/W	Default	Description
31:16			reserved
15:0	R	0x0	U48 latched omclk counter48ds of each reg_refclk_latch_interval; Will start from 0 reaching 2^48

RO_IMCLK2REF_CNT_L 0X14

Bit(s)	R/W	Default	Description
31:0	R	0x0	U48 latched reference clock counter48ds of each reg_imclk_latch_interval; Will start from 0 reaching 2^48

RO_IMCLK2REF_CNT_H 0X15

Bit(s)	R/W	Default	Description
31:16			reserved
15:0	R	0x0	U48 latched reference clock counter48ds of each reg_imclk_latch_interval; Will start from 0 reaching 2^48

RO_OMCLK2REF_CNT_L 0X16

Bit(s)	R/W	Default	Description
31:0	R	0x0	U48 latched reference clock counter48ds of each reg_omclk_latch_interval; Will start from 0 reaching 2^48

RO_OMCLK2REF_CNT_H 0X17

Bit(s)	R/W	Default	Description
31:16			reserved
15:0	R	0x0	U48 latched reference clock counter48ds of each reg_omclk_latch_interval; Will start from 0 reaching 2^48

RO_REFCLK_PKG_CNT 0X18

Bit(s)	R/W	Default	Description
31:16	R	0x0	U16,number of reference clk latch interval period conter for imclk latch
15:0	R	0x0	U16,number of reference clk latch interval period conter for imclk latch

RO_IMCLK_PKG_CNT 0X19

Bit(s)	R/W	Default	Description
31:16			reserved
15:0	R	0x0	U16,number of imclk latch interval period counter

RO_OMCLK_PKG_CNT 0X1a

Bit(s)	R/W	Default	Description
31:16			reserved
15:0	R	0x0	U16,number of omclk latch interval period counter

RO_AUD_LOCK_INT_STATUS 0X1b

Bit(s)	R/W	Default	Description
31:4			reserved
3	R	0x0	It is used to report interrupt status for omclk_state
2	R	0x0	It is used to report interrupt status for imclk_state
1	R	0x0	It is used to report interrupt status for refclk_state1
0	R	0x0	It is used to report interrupt status for refclk_state0

12. AIFIFO

12.1 Register Description

Base address: 0xffd05000

AIU_AIFIFO_CTRL 0x00

Bit(s)	R/W	Default	Description
13	R/W	0	Aififo request to dcu status
12	R/W	0	Dcu select status
11:5	R/W	0	Aififo word counter number
4:0	R/W	0	How many bits left in the first pop register

AIU_AIFIFO_STATUS 0x01

Bit(s)	R/W	Default	Description
4:0	R	0x0	How many Bits left in the first pop register

AIU_AIFIFO_GBIT 0x02

Bit(s)	R/W	Default	Description
15:0	R	0x0	The gb data

AIU_AIFIFO_CLB 0x03

Bit(s)	R/W	Default	Description
15:0	R	0x0	The gb data

AIU_MEM_AIFIFO_START_PTR 0x04

Bit(s)	R/W	Default	Description
31:0	RW	0x0	The start address from DDR

AIU_MEM_AIFIFO_CURR_PTR 0x05

Bit(s)	R/W	Default	Description
31:0	R	0x0	The current address from DDR

AIU_MEM_AIFIFO_END_PTR 0x06

Bit(s)	R/W	Default	Description
31:0	RW	0x0	The end address from DDR

AIU_MEM_AIFIFO_BYTES_AVAIL 0x07

Bit(s)	R/W	Default	Description
15:11	R/W	0	Unused.
10	R/W	0	Set this bit to 1 to enable filling of the FIFO controlled by the buffer level control. If this bit is 0, then use bit[1] to control the enabling of filling
9	R/W	0	This bit is set when data can be popped
8	R/W	0	This bit will be high when we're fetching data from the DDR memory To reset this module, set cntl_enable = 0, and then wait for busy = 0. After that you can pulse cntl_init to start over
7	R/W	0	Just in case endian. last minute byte swap of the data out of the FIFO to getbit
6	R/W	0	Unused.
5:3	R/W	0	
2	R/W	0	Set to 1 to enable reading the DDR memory FIFO and filling the pipeline to get-bit Set cntl_empty_en = cntl_fill_en = 0 when pulsing cntl_init
1	R/W	0	Set to 1 to enable reading data from DDR memory
0	R/W	0	After setting the read pointers, sizes, channel masks and read masks, set this bit to 1 and then to 0 NOTE: You don't need to pulse cntl_init if only the start address is being changed

AIU_MEM_AIFIFO_CONTROL 0x08

Bit(s)	R/W	Default	Description
15:11		0x0	Unused
10	RW	0x0	Use_level Set This bit to 1 to enable filling of the FIFO controlled by the buffer level control. If This bit is 0, then use Bit[1] to control the enabling of filling
9	RW	0x0	Data Ready. This bit is set when data can be popped
8	RW	0x0	Fill busy This bit will be high when we're fetching data from the DDR memory
7	RW	0x0	Cntl_endian_jic Just in case endian. Last minute byte swap of the data out of the FIFO to get bit
6	RW	0x0	Unused
5：3	RW	0x0	Endian
2	RW	0x0	Cntl_empty_en Set to 1 to enable reading the DDR memory FIFO and filling the pipeline to get-bit
1	RW	0x0	Cntl_fill_en Set to 1 to enable reading data from DDR memory
0	RW	0x0	Cntl_init

AIU_MEM_AIFIFO_MAN_WP 0x09

Bit(s)	R/W	Default	Description
31:0	RW	0x0	Manual write ptr.

AIU_MEM_AIFIFO_MAN_RP 0x0a

Bit(s)	R/W	Default	Description
31:0	RW	0x0	Manual read ptr.

AIU_MEM_AIFIFO_LEVEL 0x0b

Bit(s)	R/W	Default	Description
1	R/W	0	Set to 1 for manual write pointer mode
0	R/W	0	Set high then low after everything has been initialized

AIU_MEM_AIFIFO_BUF_CNTL 0x0c

Bit(s)	R/W	Default	Description
1	RW	0x0	manual mode Set to 1 for manual write pointer mode
0	RW	0x0	Init Set high then low after everything has been initialized

AIU_MEM_AIFIFO_BUF_WRAP_COUNT 0x0d

Bit(s)	R/W	Default	Description
29:24	R/W	0	A_brst_num
21:16	R/W	0	A_id
15:0	R/W	0	level_hold

AIU_MEM_AIFIFO_MEM_CTL 0x0f

Bit(s)	R/W	Default	Description
31:16	R/W	0	drop_bytes
15:14	R/W	0	drop_status (Read-Only)
13:12	R/W	0	sync_match_position (Read-Only)
11:6	R/W	0	reserved
5:4	R/W	0	TIME_STAMP_NUMBER, 0-32bits, 1-64bits, 2-96bits, 3-128bits
3	R/W	0	stamp_soft_reset
2	R/W	0	TIME_STAMP_length_enable
1	R/W	0	TIME_STAMP_sync64_enable
0	R/W	0	TIME_STAMP_enable

AIFIFO_TIME_STAMP_CNTL 0x10

Bit(s)	R/W	Default	Description
31:0	R/W	0	TIME_STAMP_SYNC_CODE_0

AIFIFO_TIME_STAMP_SYNC_0 0x11

Bit(s)	R/W	Default	Description
31:0	R/W	0	TIME_STAMP_SYNC_CODE_1

AIFIFO_TIME_STAMP_SYNC_1 0x12

Bit(s)	R/W	Default	Description
31:0	R/W	0	TIME_STAMP_0

AIFIFO_TIME_STAMP_0 0x13

Bit(s)	R/W	Default	Description
31:0	R/W	0	TIME_STAMP_1

AIFIFO_TIME_STAMP_1 0x14

Bit(s)	R/W	Default	Description
31:0	R/W	0	TIME_STAMP_2

AIFIFO_TIME_STAMP_2 0x15

Bit(s)	R/W	Default	Description
31:0	R/W	0	TIME_STAMP_3

AIFIFO_TIME_STAMP_3 0x16

Bit(s)	R/W	Default	Description
31:0	R/W	0	TIME_STAMP_LENGTH

AIFIFO_TIME_STAMP_LENGTH 0x17

Bit(s)	R/W	Default	Description
31:0	RW	0x0	TIME_STAMP_LENGTH

13. Memory Interface

This part describes S922X's memory interfaces from the following aspects:

- DDR
- NAND
- EMMC/SDIO/SD
- SPICC
- SPIFC

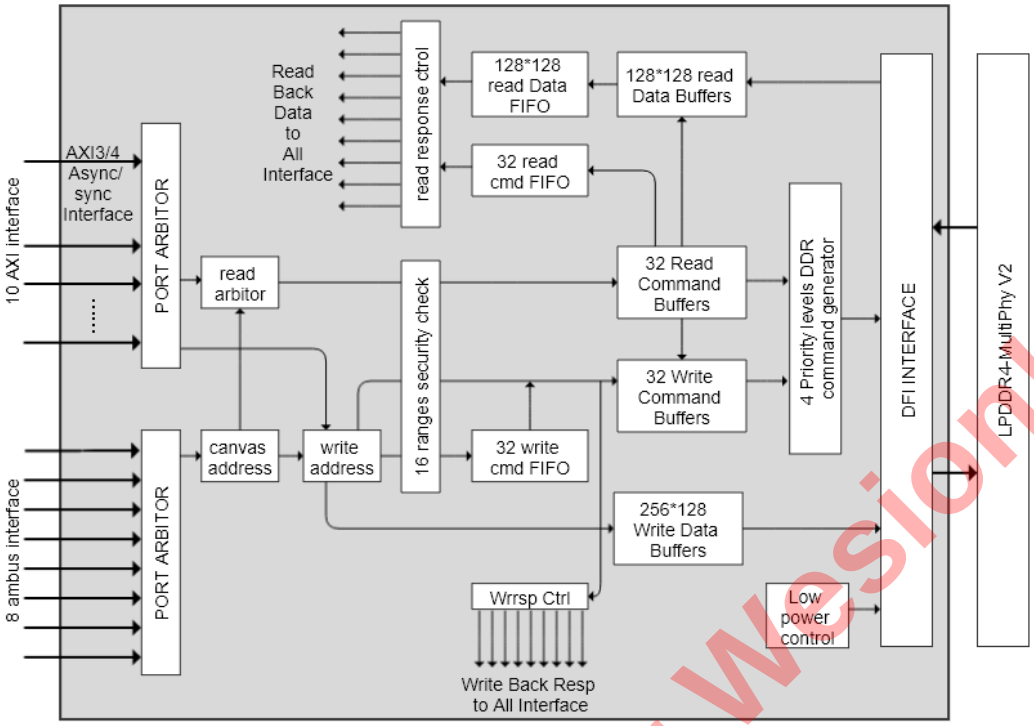
13.1 DDR

13.1.1 Overview

consists of the 2 parts: DDR memory controller (DMC) and DDR PHY controller. The main features of this module are listed below:

- Support DDR3/4 SDRAM, LPDDR3, LPDDR4 SDRAM
- Support 32bits and 16bits data mode.
- Unbalanced 16bits DDR3/4 SDRAM size in 32 bits data bus mode.
- Each DDR command buffer contains 64bytes data.
- 32 write command buffer and 32 read command buffer for DDR command generation.
- READ command reordering regardless ID.
- Write command reordering regardless IDs.
- Write and read data coherence.
- Optimized DDR command reordering based on:
  - Priority
  - Rank
  - Bank
  - Bank group interleaving(DDR4)
  - Read/write
- 4 level priority control to reduce the latency for the urgent request.
- Unblocking urgent control through Asynchronous FIFO and pipelines.
- Additional 128 depth write data FIFO to balance DMC and AXI master write throughputs.
- Additional 128*128 read data FIFO to balance DMC and AXI master read throughputs.
- Per Port Per ID Security Control through the all security ranges.
- Up to 16 security range can be defined.
- Support up to 16 AXI3/4 and 8 AMBUS(Amlogic) interfaces.
- Optimized AMBUS to support 0~3 burst to improve DDR efficiency and reduce
- Low power control with LPDDR4 Multi-PHY V2.
  - Adding auto Self-Refresh mode for super low bandwidth application.
  - Auto power down mode for most of application.
  - Hardware Fast Frequency change less than 10uS.

Figure 13-1 Diagram of DDR Interface



13.1.2 Register Description

DMC Register spec.

DMC unsecure register. Base address 0xFF638000. Each register takes 4 byte address.

Each register's final address = 0xFF638000+ offset * 4.

AM_DDR_PLL_CNTL0 0x0000

Bit(s)	R/W	Default	Description
29			dpll_reset.
28			dpll_en.
27:26			dpll_clk_en
20:19			od1
18:16			od
14:10			dpll_ref_div_n
8:0			dpll_int_num

AM_DDR_PLL_CNTL1 0x0001

Bit(s)	R/W	Default	Description
18:0			ddr_dpll_frac

AM_DDR_PLL_CNTL2 0x0002

Bit(s)	R/W	Default	Description
22:20			fref_sel
17:16			os_ssc
15:12			ssc_str_m
8			ssc_en
7:4			ssc_dep_sel
1:0			dpll ss_mode

AM_DDR_PLL_CNTL3 0x0003

Bit(s)	R/W	Default	Description
31			afc bypass
30			afc clk sel
29			code new
28			dco_m_en
27			dco_sdm_en
26			div2
25			div mode
24			fast_lock mode
23			fb_pre_div
22			filter_mode
21			fix_en
20			freq_shift_en
19			load
18			load_en
17			lock_f
16			pulse_width_en
15			sdmnc_en
14			sdmnc_mode
13			sdmnc_range
12			tdc_en

Bit(s)	R/W	Default	Description
11			tdc_mode_sel
10			wait_en

AM_DDR_PLL_CNTL4 0x0004

Bit(s)	R/W	Default	Description
1:0			pfd_gain
7:4			filter_pvt1
11:8			filter pvt2
13:12			acq_gain
18:16			lambda0
22:20			lambda1
26:24			rou
30:28			alpha

AM_DDR_PLL_CNTL5 0x0005

Bit(s)	R/W	Default	Description
15:0			reve
21:16			lm_s
27:24			lm_w
30:28			adj_vco_ldo

AM_DDR_PLL_CNTL6 0x0006

Bit(s)	R/W	Default	Description
15:0			reve
21:16			lm_s
27:24			lm_w
30:28			adj_vco_ldo

AM_DDR_PLL_STS 0x0007

Bit(s)	R/W	Default	Description
31			DDR_PLL_LOCK
30:19			not used
18			DDR_AFC_DONE

17			DDR_PLL_LOCK
16:7			DDR_DPLL_OUT_RSV
6:0			DDR_SDMNC_MONITOR

DDR_CLK_CNTL 0x0008

Bit(s)	R/W	Default	Description
31			ddr_pll_clk enable. enable the clock from DDR_PLL to clock generateion.
30			ddr_pll_prod_test_en. enable the clock to clock/32 which to clock frequency measurement and production test pin.
29			not used.
28			clock generation logic soft reset. 0 = reset.
27			phy_4xclk phase inverter
26			pll_freq divide/2. 1: use pll div/2 clock as the n_clk. 0: use pll clock as n_clk. this setting is used for the DDR PHY PLL fast lock mode.
2			enable dmc_clk.
1			enable LPDDR4-PHY DfiClk.
0			enable LPDDR4-PHY DfiCtlClk.

DDR_PHY_CTRL 0x0009

Bit(s)	R/W	Default	Description
4			DDR PHY PwrOkIn pin.
1			DDR PHY APB soft reset_n.
0			phy_reset_n.

AM_DDR_PLL_FREQ1_OD 0x000c

Bit(s)	R/W	Default	Description
8			currunt FREQ selection. it can forced to change to select which freqency to select, or it can auto changed by FREQ change hardware.
5:4			OD1.
2:0			OD.

The following registers' base address is 0xff638000. Each register takes 4 byte address.

Each register's final address = 0xff638000 + offset * 4.

DMC_REQ_CTRL 0x0000

Bit(s)	R/W	Default	Description
23			enable dmc request of ambus chan 7. Reserved for GE2D interface. Async interface.

22			enable dmc request of ambus chan 6. DOS HCODEC interface Sync interface.
21			enable dmc request of ambus chan 5. DOS VDEC interface Sync interface.
20			enable dmc request of ambus chan 4. VPU write interface 1 Sync interface.
19			enable dmc request of ambus chan 3. VPU write interface 0 Sync interface.
18			enable dmc request of ambus chan 2. VPU read interface 2. Sync interface.
17			enable dmc request of ambus chan 1. VPU read interface 1. Sync interface.
16			enable dmc request of ambus chan 0. VPU read interface 0. Sync interface.
9			enable dmc request of axibus chan 9. wave async interface.
8			enable dmc request of axibus chan 8 hevc_b async interface.
7			enable dmc request of axibus chan 7. DEVICE. Async interface.
6			enable dmc request of axibus chan 6. USB Async interface.
5			enable dmc request of axibus chan 5. reserved for dmc_test.
4			enable dmc request of axibus chan 4. hevc front Async interface.
3			enable dmc request of axibus chan 3. HDCP/HDMI Async interface.
2			enable dmc request of axibus chan 2. pcie async
1			enable dmc request of axibus chan 1. Dvalin . async interface.
0			enable dmc request of axibus chan 0. CPU/A53 async interface.

DMC_SOFT_RST 0x0001

Bit(s)	R/W	Default	Description
31:24			Reserved.
23:16			8 AMBUS input interface n_clk domain reset_n signal. 0 : reset. 1: normal working mode.
15:0			16 AXI BUS input intefaces n_clk domain reset_n signal. 0: reset. 1: normal working mode. each bit for one interface.

DMC_SOFT_RST1 0x0002

Bit(s)	R/W	Default	Description
31:24			Not used.
23.16			8 am bus interfaces master clock domain reset_n signal. 0 : reset : 1 normal working mode.
15:0.			16 AXI bus interfaces master clock domain reset_n signal. 0 : reset : 1 normal working mode.

DMC_SOFT_RST2 0x0003

Bit(s)	R/W	Default	Description
31~11			Reserved.
10			DMC DFI cmd soft reset_n
9			DMC DFI MISC soft reset_n
8			DMC DFI data soft reset_n
7			DMC DFI dcu soft reset_n
6			DMC siu soft reset_n
5			DMC test soft reset_n. 0 : reset. 1 : normal working mode.
4			DMC low power control moudle soft reset_n. 0 : reset. 1 : normal working mode.
3			DMC QOS monitor module soft reset_n. 0 : reset. 1 : normal working mode.
2			DMC register modle soft reset_n. 0 : reset. 1 : normal working mode.
1			DMC canvas transfer module soft reset_n. 0 : reset. 1 : normal working mode.
0			DMC command buffers and command generation modules soft reset. 0 = reset. 1:

DMC_RST_STS1 0x0004

Bit(s)	R/W	Default	Description
31:24			Not used.
23.0			Read only. The DMC_SOFT_RST1 signal in n_clk domain. When one of the 2 clocks is too slow or too fast, we can read this register to make sure another clock domain reset is done.

DMC_VERSION 0x0005

DMC version number.

DMC_RAM_PD 0x0006

Bit(s)	R/W	Default	Description
5:4			DDR channel 1 READ/WRITE data path SRAMS in power down control. 2'b11: power down. 2'b00: working mode.
3:2			DDR channel 0 READ/WRITE data path SRAMS in power down control. 2'b11: power down. 2'b00: working mode.
1:0			CANVAS LUT memories in power down mode control. 2'b11: power down. 2'b00: working mode.

DC_CAV_LUT_DATAL 0x0012

low 32 bits of canvas data which need to be configured to canvas memory.

DC_CAV_LUT_DATAH 0x0013

Bit(s)	R/W	Default	Description
60:58			Canvas block mode. 2: 64x32, 1: 32x32; 0: linear mode.
57:56			canvas Y direction wrap control. 1: wrap back in y. 0: not wrap back.
55			canvas X direction wrap control. 1: wrap back in X. 0: not wrap back.
54			canvas Hight.
53:41			canvas Width, unit: 8bytes. must in 32bytes boundary. that means last 2 bits must be 0.
40:29			cavnas start address. unit. 8 bytes. must be in 32bytes boundary. that means last 2bits must be 0.
28:0			Canvas block mode. 2 : 64x32, 1: 32x32; 0 : linear mode.

DC_CAV_LUT_ADDR 0x0014

Bit(s)	R/W	Default	Description
9:8			Write 9:8 2'b10. the canvas data will saved in canvas memory with addres 7:0.
7:0			256 canvas Look up table address.

DC_CAV_LUT_RDATAL 0x0015

CBUS low 32bytes canvas read back data from LUT.

DC_CAV_LUT_RDATAH 0x0016

CBUS low 32bytes canvas read back data from LUT.

DC_CAV_BLK_CTRL0 0x0018

Bit(s)	R/W	Default	Description
31:0			blkmode. 1 : 32x32. 0 : others.

DC_CAV_BLK_CTRL1 0x0019

Bit(s)	R/W	Default	Description
63:32			blkmode. 1 : 32x32. 0 : others.

DC_CAV_BLK_CTRL2 0x001a

Bit(s)	R/W	Default	Description
95:64			blkmode. 1 : 32x32. 0 : others.

DC_CAV_BLK_CTRL3 0x001b

Bit(s)	R/W	Default	Description
127:96			blkmode. 1 : 32x32. 0 : others.

DC_CAV_BLK_CTRL4 0x001c

Bit(s)	R/W	Default	Description
159:128			blkmode. 1 : 32x32. 0 : others.

DC_CAV_BLK_CTRL5 0x001d

Bit(s)	R/W	Default	Description
191:160			blkmode. 1 : 32x32. 0 : others.

DC_CAV_BLK_CTRL6 0x001e

Bit(s)	R/W	Default	Description
223:192			blkmode. 1 : 32x32. 0 : others.

DC_CAV_BLK_CTRL7 0x001f

Bit(s)	R/W	Default	Description
255:224			blkmode. 1 : 32x32. 0 : others.

DMC_MON_CTRL0 0x0020

Bit(s)	R/W	Default	Description
31			qos_mon_en. write 1 to trigger the enable. polling this bit 0, means finished. or use interrupt to check finish.
30			qos_mon interrupt clear. clear the qos monitor result. read 1 = qos mon finish interrupt.
3			qos monitor 3 enable.
2			qos monitor 2 enable.
1			qos monitor 1 enable.
0			qos monitor 0 enable.

DMC_MON_CTRL1 0x0021

Bit(s)	R/W	Default	Description
23:0			qos monitor 0 channel select. 8 ambus port and 16 AXI port selection. 1 bit for one port.

DMC_MON_CTRL2 0x0022

Bit(s)	R/W	Default	Description
15:0			port select for the selected channel.

DMC_MON_CTRL3 0x0023

Bit(s)	R/W	Default	Description

23:0			qos monitor 0 channel select. 8 ambus port and 16 AXI port selection. 1 bit for one port.

DMC_MON_CTRL4 0x0024

Bit(s)	R/W	Default	Description
15:0			port select for the selected channel.

DMC_MON_CTRL5 0x0025

Bit(s)	R/W	Default	Description
23:0			qos monitor 0 channel select. 8 ambus port and 16 AXI port selection. 1 bit for one port.

DMC_MON_CTRL6 0x0026

Bit(s)	R/W	Default	Description
15:0			port select for the selected channel.

DMC_MON_CTRL7 0x0027

Bit(s)	R/W	Default	Description
23:0			port select for the selected channel.

DMC_MON_CTRL8 0x0028

Bit(s)	R/W	Default	Description
15:0			port select for the selected channel.

DMC_MON_ALL_REQ_CNT 0x0029

At the test period, the whole MMC request time.

DMC_MON_ALL_GRANT_CNT 0x002a

At the test period, the whole MMC granted data cycles. 64bits unit.

DMC_MON_ONE_GRANT_CNT 0x002b

At the test period, the granted data cycles for the selected channel and ports.

DMC_MON_SEC_GRANT_CNT 0x002c

At the test period, the granted data cycles for the selected channel and ports.

DMC_MON_THD_GRANT_CNT 0x002d

At the test period, the granted data cycles for the selected channel and ports.

DMC_MON_FOR_GRANT_CNT 0x002e

At the test period, the granted data cycles for the selected channel and ports.

DMC_MON_TIMER 0x002f

Timer for the monitor period.

DMC_CLKG_CTRL0 0x0030

Bit(s)	R/W	Default	Description
23:16			Enable the 8 ambus interfaces both main and n_clk auto clock gating function. each 1 bit for one interface.
15:0			Enable the 16 axi interfaces both main and n_clk auto clock gating function. each 1 bit for one interface.

DMC_CLKG_CTRL1 0x0031

Bit(s)	R/W	Default	Description
23:16			Force to disable the 8 ambus interfaces both main and n_clk. each 1 bit for one interface.
15:0			Force to disable the 16 axi interfaces both main and n_clk. each 1 bit for one interface.

DMC_CLKG_CTRL2 0x0032

Bit(s)	R/W	Default	Description
7			Force to disalbe the clock of write rsp generation.
6			Force to disalbe the clock of read rsp generation.
5			Force to disalbe the clock of command filter.
4			Force to disalbe the clock of write reorder buffer.
3			Force to disalbe the clock of write data buffer.
2			Force to disalbe the clock of read reorder buffer.
1			Force to disalbe the clock of read canvas.
0			Force to disalbe the clock of write canvas.

DMC_CLKG_CTRL3 0x0033

Bit(s)	R/W	Default	Description
7			Enalbe auto clock gating for write rsp generation.
6			Enalbe auto clock gating for read rsp generation.
5			Enalbe auto clock gating for ddr0 command filter.
4			Enalbe auto clock gating for ddr0 write reorder buffer.
3			Enalbe auto clock gating for ddr0 write data buffer.
2			Enalbe auto clock gating for ddr0 read reorder buffer.
1			Enalbe auto clock gating for read canvas.
0			Enalbe auto clock gating for write canvas.

DMC_CHAN_STS 0x0036

Bit(s)	R/W	Default	Description
31:28	RO		Not used.
27	RO		always 1
26	RO		ddr0 write data buffer idle. 1 : idle 0: busy.
25	RO		always 1.
24	RO		ddr0 wbuf idle. 1 : idle 0: busy.
23:16	RO		ambus channel idle. 1 : idle 0: busy.
15:0.	RO		axibus channel idle. 1 : idle 0: busy.

DMC_2ARB_CTRL 0x0038

Bit(s)	R/W	Default	Description
32:24			Not used.
22.			always 1
21:12			ddr0 write data buffer idle. 1 : idle 0: busy.
11:6.			always 1.
5:0.			ddr0 wbuf idle. 1 : idle 0: busy.

DMC_CMD_FILTER_CTRL1 0x0040

Bit(s)	R/W	Default	Description
30			1 : use DDR4 special filter.

DMC_CMD_FILTER_CTRL2 0x0041

Not used.

DMC_CMD_FILTER_CTRL3 0x0042

Bit(s)	R/W	Default	Description
31.			force wbuf empty.
30:26			wbuf high level number
25:21			wbuf mid level number
20:16			wbuf low level number
14:10			rbuf high level number
9:5			rbuf middle level number
4:0			rbuf low level number

DMC_CMD_FILTER_CTRL4 0x0043

Bit(s)	R/W	Default	Description
29:25			tITW.long
24:20			tITW. short
19:12			tAP auto precharge the bank not used if idle that long time.
11:6			write to read accesses if there write hit request.
5:0			read to write accesses if there write hit request.

DMC_CMD_FILTER_CTRL5 0x0044

Bit(s)	R/W	Default	Description
31:24			Once ddr data bus switch to read, the maxmum read command number to give up the bus when there's write request pending for write buffer.
23:16			Once ddr data bus switch to write, the maxmum write command number to give up the bus when there's read request pending too long.
15:8.			Once ddr data bus switch to read, the minimum read command number to transfer back to write stage if there's still pending read request.
7:0.			Once ddr data bus switch to write, the minimum write command number to transfer back to read stage if there's still pending write request.

DMC_CMD_BUFFER_CTRL 0x0045

Bit(s)	R/W	Default	Description
31:26			Total write buffer number. Default 32.
25:20			Total read buffer number. Default 32.
19:8			Reserved
7:0			aw_pending_inc_num. incease write ugent level 1 when write command waiting to in write buffer that long.

DMC_CMD_BUFFER_CTRL1 0x0046

Bit(s)	R/W	Default	Description
31:26			Read buffer number in non-urgent request.
25:20			Read buffer bank miss watch dog threshold.
19:8			Read buffer urgent level 3 counter inc weight.
7:0			Read buffer urgent level 2 counter inc weight.
3:0			Read buffer urgent level 2 counter inc weight.

DMC_CMD_FILTER_CTRL6 0x0047

Bit(s)	R/W	Default	Description
31:24			write urgent 3 request pending hold num.
23:16			write urgent 2 request pending hold num.
15:8			write urgent 1 request pending hold num.
7:0			write urgent 0 request pending hold num.

DMC_RDDBUF_CTRL 0x0048

Not used.

DMC_CMD_FILTER_CTRL7 0x0049

Bit(s)	R/W	Default	Description
15:8			Write to read waiting cycles if there write hit request.
7:0			Read to write waiting cycles if there write hit request.

DMC_AM0_CHAN_CTRL 0x0060

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non-urgent request.
13:4.			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AM0_HOLD_CTRL 0x0061

Bit(s)	R/W	Default	Description
15:8			Read hold num. Max outstanding request number.
7:0			Read hold release num. If the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AM0_CHAN_CTRL1 0x0062

Bit(s)	R/W	Default	Description
31			Side band signal used as block other request.

Bit(s)	R/W	Default	Description
30			Side band urgent increase enable.
29			Side band urgent decrease urgent enable.
23:16			When bit 31 enabled, block the ambus related bits read request.
15:0			When bit 31 enabled, block the axi bus related bits read request.

DMC_AM0_CHAN_CTRL2 0x0063

Bit(s)	R/W	Default	Description
31:24			Not used.
23:16			When side band signal used as block other request, and side bank signal is high, block the ambus related bits write request.
15:0			When side band signal used as block other request, and side bank signal is high, block the axi bus related bits write request.

DMC_AM1_CHAN_CTRL 0x0064

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non urgent request.
13:4.			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AM1_HOLD_CTRL 0x0065

Bit(s)	R/W	Default	Description
15:8			Read hold num. Max outstanding request number.
7:0			Read hold release num. If the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AM1_CHAN_CTRL1 0x0066

Bit(s)	R/W	Default	Description
31:			Side band signal used as block other request.

30 :			Side band urgent increase enable.
29 :			Side band urgent decrease urgent enable.
23:16			When bit 31 enabled, block the ambus related bits read request.
15:0			When bit 31 enabled, block the axi bus related bits read request.

DMC_AM1_CHAN_CTRL2 0x0067

Bit(s)	R/W	Default	Description
31:24			Not used.
23:16			When side band signal used as block other request, and side bank signal is high, block the ambus related bits write request.
15:0			When side band signal used as block other request, and side bank signal is high, block the axi bus related bits write request.

DMC_AM2_CHAN_CTRL 0x0068

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non urgent request.
13:4.			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AM2_HOLD_CTRL 0x0069

Bit(s)	R/W	Default	Description
15:8			Read hold num. Max outstanding request number.
7:0			Read hold release num. If the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AM2_CHAN_CTRL1 0x006a

Bit(s)	R/W	Default	Description
31:			Side band signal used as block other request.
30 :			Side band urgent increase enable.
29 :			Side band urgent decrease urgent enable.

23:16			When bit 31 enabled, block the ambus related bits read request.
15:0			When bit 31 enabled, block the axi bus related bits read request.

DMC_AM2_CHAN_CTRL1 0x006b

Bit(s)	R/W	Default	Description
31:24			Not used.
23:16			When side band signal used as block other request, and side bank signal is high, block the ambus related bits write request.
15:0			When side band signal used as block other request, and side bank signal is high, block the axi bus related bits write request.

DMC_AM3_CHAN_CTRL 0x006c

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non urgent request.
13:4.			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AM3_HOLD_CTRL 0x006d

Bit(s)	R/W	Default	Description
31:24			Write hold num. Max outstanding request number.
23:16			Write hold release num. If the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AM3_CHAN_CTRL1 0x006e

Bit(s)	R/W	Default	Description
31			Side band signal used as block other request.
30			Side band urgent increase enable.
29			Side band urgent decrease urgent enable.
23:16			When bit 31 enabled, block the ambus related bits read request.
15:0			When bit 31 enabled, block the axi bus related bits read request.

DMC_AM3_CHAN_CTRL2 0x006f

Bit(s)	R/W	Default	Description
31:24			Not used.
23:16			When side band signal used as block other request, and side bank signal is high, block the ambus related bits write request.
15:0			When side band signal used as block other request, and side bank signal is high, block the axi bus related bits write request.

DMC_AM4_CHAN_CTRL 0x0070

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non urgent request.
13:4.			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AM4_HOLD_CTRL 0x0071

Bit(s)	R/W	Default	Description
31:24			Write hold num. Max outstanding request number.
23:16			Write hold release num. If the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AM4_CHAN_CTRL1 0x0072

Bit(s)	R/W	Default	Description
31:			Side band signal used as block other request.
30 :			Side band urgent increase enable.
29 :			Side band urgent decrease urgent enable.
23:16			When bit 31 enabled, block the ambus related bits read request.
15:0			When bit 31 enabled, block the axi bus related bits read request.

DMC_AM4_CHAN_CTRL2 0x0073

Bit(s)	R/W	Default	Description

31:24			Not used.
23:16			When bit 31 enabled, block the ambus related bits write request.
15:0			When bit 31 enabled, block the axi bus related bits write request.

DMC_AM5_CHAN_CTRL 0x0074

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non urgent request.
13:4.			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AM5_HOLD_CTRL 0x0075

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AM6_CHAN_CTRL 0x0078

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
18			Force this channel all request to be super urgent request.
17			Force this channel all request to be urgent request.
16			Force this channel all request to be non urgent request.
13:4			Read request pending cycle number to inc urgent level if not granted.

3:0			Canvas arbiter arbiter weight

DMC_AM6_HOLD_CTRL 0x0079

Bit(s)	R/W	Default	Description
31:24			Enable to incr 2 urgent levels if the pending cycles is doubled.
23:16			Enable to incr 3 urgent levels.
15:8			Write request pending cycle number to inc urgent level if not granted.
7:0			Force this channel all request to be super urgent request.

DMC_AM7_CHAN_CTRL 0x007c

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non urgent request.
13:4.			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AM7_HOLD_CTRL 0x007d

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI0_CHAN_CTRL 0x0080

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.

19			Axi0 default urgent control : 1 use AWUGT/ARUGT pins in the port. 0 : use bit[15:14] of this register.
18			Force this channel all request to be super urgent request.
17			Force this channel all request to be urgent request.
16			Force this channel all request to be non urgent request.
15:14			Axi0 default urgent level.
13:4			Read request pending cycle number to inc urgent level if not granted.
3:0			Arbiter weight

DMC_AXI0_HOLD_CTRL 0x0081

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI0_CHAN_CTRL1 0x0082

Bit(s)	R/W	Default	Description
19:16			FIQ status
15:12			IRQ status.
11			ARM FIQ controlled super urgent enable.
10			ARM FIQ controlled urgent enable.
9			ARM IRQ controlled super urgent enable.
8			ARM IRQ controlled urgent enable.
7			IRQ/FIQ controll enable.
6:5			Not used.
4			Enable AXI0 auto urgent enable. When there's no other request, treat the AXI0 as super urgent request. Other wise, use the bit3:0 to set the urgent.
3:2			A9 urgent if there's VIU request.
1:0			A9 urgent if there's request other than VIU

DMC_AXI1_CHAN_CTRL 0x0084

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
19.			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non-urgent request.
15:14			Axi1 default urgent level.
13:4.			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI1_HOLD_CTRL 0x0085

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI2_CHAN_CTRL 0x0088

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
19.			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non-urgent request.

15:14			Axi1 default urgent level.
13:4.			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI2_HOLD_CTRL 0x0089

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI3_CHAN_CTRL 0x008c

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
19.			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non-urgent request.
15:14			Axi1 default urgent level.
13:4			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI3_HOLD_CTRL 0x008d

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.

7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI4_CHAN_CTRL 0x0090

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
19.			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18.			Force this channel all request to be super urgent request.
17.			Force this channel all request to be urgent request.
16.			Force this channel all request to be non-urgent request.
15:14			Axi1 default urgent level.
13:4			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI4_HOLD_CTRL 0x0091

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI5_CHAN_CTRL 0x0094

Not used.

DMC_AXI5_HOLD_CTRL 0x0095

Not used.

DMC_AXI6_CHAN_CTRL 0x0096

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.

Bit(s)	R/W	Default	Description
29:20			Write request pending cycle number to inc urgent level if not granted.
19			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18			Force this channel all request to be super urgent request.
17			Force this channel all request to be urgent request.
16			Force this channel all request to be non-urgent request.
15:14			Axi1 default urgent level.
13:4			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI6_HOLD_CTRL 0x0097

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI7_CHAN_CTRL 0x009c

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
19			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18			Force this channel all request to be super urgent request.
17			Force this channel all request to be urgent request.
16			Force this channel all request to be non-urgent request.
15:14			Axi1 default urgent level.
13:4			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI7_HOLD_CTRL 0x009d

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI8_CHAN_CTRL 0x00a0

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
19			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18			Force this channel all request to be super urgent request.
17			Force this channel all request to be urgent request.
16			Force this channel all request to be non-urgent request.
15:14			Axi1 default urgent level.
13:4			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI8_HOLD_CTRL 0x00a1

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI9_CHAN_CTRL 0x00a4

Bit(s)	R/W	Default	Description

31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
19			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18			Force this channel all request to be super urgent request.
17			Force this channel all request to be urgent request.
16			Force this channel all request to be non-urgent request.
15:14			Axi1 default urgent level.
13:4			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI9_HOLD_CTRL 0x00a5

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI10_CHAN_CTRL 0x00a8

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
19			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18			Force this channel all request to be super urgent request.
17			Force this channel all request to be urgent request.
16			Force this channel all request to be non-urgent request.
15:14			Axi1 default urgent level.

13:4			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI10_HOLD_CTRL 0x00a9

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

DMC_AXI11_CHAN_CTRL 0x00ac

Bit(s)	R/W	Default	Description
31			Enable to incr 2 urgent levels if the pending cycles is doubled.
30			Enable to incr 3 urgent levels.
29:20			Write request pending cycle number to inc urgent level if not granted.
19			Axi0 default urgent control: 1 use AWUGT/ARUGT pins in the port. 0: use bit[15:14] of this register.
18			Force this channel all request to be super urgent request.
17			Force this channel all request to be urgent request.
16			Force this channel all request to be non-urgent request.
15:14			Axi1 default urgent level.
13:4			Read request pending cycle number to inc urgent level if not granted.
3:0			Canvas arbiter arbiter weight

DMC_AXI11_HOLD_CTRL 0x00ad

Bit(s)	R/W	Default	Description
31:24			Write hold num. max outstanding request number.
23:16			Write hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.
15:8			Read hold num. max outstanding request number.
7:0			Read hold release num. if the outstanding request == hold num, then hold this request unless the outstanding request number below the hold release number, then continue to request.

The following registers' base address is 0xff639000. Each register takes 4 byte address.

Each register's final address = 0xff639000 + offset * 4.

DMC_SEC_RANGE_CTRL 0x0000

Bit(s)	R/W	Default	Description
31:16			16 range security level. each
15:0			16 range enable. each bit for one range to indentify the range is enabled or not.

DMC_SEC_RANGE0_CTRL 0x0001

Bit(s)	R/W	Default	Description
31:16			range 0 end address higher 16bits.
15:0			range 0 start address higher 16bits.

DMC_SEC_RANGE1_CTRL 0x0002

DMC_SEC_RANGE2_CTRL 0x0003

DMC_SEC_RANGE3_CTRL 0x0004

DMC_SEC_RANGE4_CTRL 0x0005

DMC_SEC_RANGE5_CTRL 0x0006

DMC_SEC_RANGE6_CTRL 0x0007

DMC_SEC_RANGE7_CTRL 0x0008

DMC_SEC_RANGE8_CTRL 0x0009

DMC_SEC_RANGE9_CTRL 0x000a

DMC_SEC_RANGE10_CTRL 0x000b

DMC_SEC_RANGE11_CTRL 0x000c

DMC_SEC_RANGE12_CTRL 0x000d

DMC_SEC_RANGE13_CTRL 0x000e

DMC_SEC_RANGE14_CTRL 0x000f

DMC_SEC_RANGE0_RID_CTRL0 0x0010

Bit(s)	R/W	Default	Description
31:0			range 0 end address higher 16bits.

DMC_SEC_RANGE0_RID_CTRL1 0x0011

Bit(s)	R/W	Default	Description
63:32			range 0 end address higher 16bits.

DMC_SEC_RANGE0_RID_CTRL2 0x0012

Bit(s)	R/W	Default	Description

95:64			range 0 end address higher 16bits.

DMC_SEC_RANGE0_RID_CTRL3 0x0013

Bit(s)	R/W	Default	Description
127:96			range 0 end address higher 16bits.

DMC_SEC_RANGE1_RID_CTRL0 0x0014
DMC_SEC_RANGE1_RID_CTRL1 0x0015
DMC_SEC_RANGE1_RID_CTRL2 0x0016
DMC_SEC_RANGE1_RID_CTRL3 0x0017
DMC_SEC_RANGE2_RID_CTRL0 0x0018
DMC_SEC_RANGE2_RID_CTRL1 0x0019
DMC_SEC_RANGE2_RID_CTRL2 0x001a
DMC_SEC_RANGE2_RID_CTRL3 0x001b
DMC_SEC_RANGE3_RID_CTRL0 0x001c
DMC_SEC_RANGE3_RID_CTRL1 0x001d
DMC_SEC_RANGE3_RID_CTRL2 0x001e
DMC_SEC_RANGE3_RID_CTRL3 0x001f
DMC_SEC_RANGE4_RID_CTRL0 0x0020
DMC_SEC_RANGE4_RID_CTRL1 0x0021
DMC_SEC_RANGE4_RID_CTRL2 0x0022
DMC_SEC_RANGE4_RID_CTRL3 0x0023
DMC_SEC_RANGE5_RID_CTRL0 0x0024
DMC_SEC_RANGE5_RID_CTRL1 0x0025
DMC_SEC_RANGE5_RID_CTRL2 0x0026
DMC_SEC_RANGE5_RID_CTRL3 0x0027
DMC_SEC_RANGE6_RID_CTRL0 0x0028
DMC_SEC_RANGE6_RID_CTRL1 0x0029
DMC_SEC_RANGE6_RID_CTRL2 0x002a
DMC_SEC_RANGE6_RID_CTRL3 0x002b
DMC_SEC_RANGE7_RID_CTRL0 0x002c
DMC_SEC_RANGE7_RID_CTRL1 0x002d
DMC_SEC_RANGE7_RID_CTRL2 0x002e
DMC_SEC_RANGE7_RID_CTRL3 0x002f
DMC_SEC_RANGE8_RID_CTRL0 0x0030
DMC_SEC_RANGE8_RID_CTRL1 0x0031
DMC_SEC_RANGE8_RID_CTRL2 0x0032
DMC_SEC_RANGE8_RID_CTRL3 0x0033
DMC_SEC_RANGE9_RID_CTRL0 0x0034
DMC_SEC_RANGE9_RID_CTRL1 0x0035
DMC_SEC_RANGE9_RID_CTRL2 0x0036
DMC_SEC_RANGE9_RID_CTRL3 0x0037
DMC_SEC_RANGE10_RID_CTRL0 0x0038

DMC_SEC_RANGE10_RID_CTRL1 0x0039
DMC_SEC_RANGE10_RID_CTRL2 0x003a
DMC_SEC_RANGE10_RID_CTRL3 0x003b
DMC_SEC_RANGE11_RID_CTRL0 0x003c
DMC_SEC_RANGE11_RID_CTRL1 0x003d
DMC_SEC_RANGE11_RID_CTRL2 0x003e
DMC_SEC_RANGE11_RID_CTRL3 0x003f
DMC_SEC_RANGE12_RID_CTRL0 0x0040
DMC_SEC_RANGE12_RID_CTRL1 0x0041
DMC_SEC_RANGE12_RID_CTRL2 0x0042
DMC_SEC_RANGE12_RID_CTRL3 0x0043
DMC_SEC_RANGE13_RID_CTRL0 0x0044
DMC_SEC_RANGE13_RID_CTRL1 0x0045
DMC_SEC_RANGE13_RID_CTRL2 0x0046
DMC_SEC_RANGE13_RID_CTRL3 0x0047
DMC_SEC_RANGE14_RID_CTRL0 0x0048
DMC_SEC_RANGE14_RID_CTRL1 0x0049
DMC_SEC_RANGE14_RID_CTRL2 0x004a
DMC_SEC_RANGE14_RID_CTRL3 0x004b
DMC_SEC_RANGE15_RID_CTRL0 0x004c
DMC_SEC_RANGE15_RID_CTRL1 0x004d
DMC_SEC_RANGE15_RID_CTRL2 0x004e
DMC_SEC_RANGE15_RID_CTRL3 0x004f
DMC_SEC_RANGE0_WID_CTRL0 0x0050
DMC_SEC_RANGE0_WID_CTRL1 0x0051
DMC_SEC_RANGE0_WID_CTRL2 0x0052
DMC_SEC_RANGE0_WID_CTRL3 0x0053
DMC_SEC_RANGE1_WID_CTRL0 0x0054
DMC_SEC_RANGE1_WID_CTRL1 0x0055
DMC_SEC_RANGE1_WID_CTRL2 0x0056
DMC_SEC_RANGE1_WID_CTRL3 0x0057
DMC_SEC_RANGE2_WID_CTRL0 0x0058
DMC_SEC_RANGE2_WID_CTRL1 0x0059
DMC_SEC_RANGE2_WID_CTRL2 0x005a
DMC_SEC_RANGE2_WID_CTRL3 0x005b
DMC_SEC_RANGE3_WID_CTRL0 0x005c
DMC_SEC_RANGE3_WID_CTRL1 0x005d

DMC_SEC_RANGE3_WID_CTRL2 0x005e
DMC_SEC_RANGE3_WID_CTRL3 0x005f
DMC_SEC_RANGE4_WID_CTRL0 0x0060
DMC_SEC_RANGE4_WID_CTRL1 0x0061
DMC_SEC_RANGE4_WID_CTRL2 0x0062
DMC_SEC_RANGE4_WID_CTRL3 0x0063
DMC_SEC_RANGE5_WID_CTRL0 0x0064
DMC_SEC_RANGE5_WID_CTRL1 0x0065
DMC_SEC_RANGE5_WID_CTRL2 0x0066
DMC_SEC_RANGE5_WID_CTRL3 0x0067
DMC_SEC_RANGE6_WID_CTRL0 0x0068
DMC_SEC_RANGE6_WID_CTRL1 0x0069
DMC_SEC_RANGE6_WID_CTRL2 0x006a
DMC_SEC_RANGE6_WID_CTRL3 0x006b
DMC_SEC_RANGE7_WID_CTRL0 0x006c
DMC_SEC_RANGE7_WID_CTRL1 0x006d
DMC_SEC_RANGE7_WID_CTRL2 0x006e
DMC_SEC_RANGE7_WID_CTRL3 0x006f
DMC_SEC_RANGE8_WID_CTRL0 0x0070
DMC_SEC_RANGE8_WID_CTRL1 0x0071
DMC_SEC_RANGE8_WID_CTRL2 0x0072
DMC_SEC_RANGE8_WID_CTRL3 0x0073
DMC_SEC_RANGE9_WID_CTRL0 0x0074
DMC_SEC_RANGE9_WID_CTRL1 0x0075
DMC_SEC_RANGE9_WID_CTRL2 0x0076
DMC_SEC_RANGE9_WID_CTRL3 0x0077
DMC_SEC_RANGE10_WID_CTRL0 0x0078
DMC_SEC_RANGE10_WID_CTRL1 0x0079
DMC_SEC_RANGE10_WID_CTRL2 0x007a
DMC_SEC_RANGE10_WID_CTRL3 0x007b
DMC_SEC_RANGE11_WID_CTRL0 0x007c
DMC_SEC_RANGE11_WID_CTRL1 0x007d
DMC_SEC_RANGE11_WID_CTRL2 0x007e
DMC_SEC_RANGE11_WID_CTRL3 0x007f
DMC_SEC_RANGE12_WID_CTRL0 0x0080
DMC_SEC_RANGE12_WID_CTRL1 0x0081
DMC_SEC_RANGE12_WID_CTRL2 0x0082

DMC_SEC_RANGE12_WID_CTRL3 0x0083
DMC_SEC_RANGE13_WID_CTRL0 0x0084
DMC_SEC_RANGE13_WID_CTRL1 0x0085
DMC_SEC_RANGE13_WID_CTRL2 0x0086
DMC_SEC_RANGE13_WID_CTRL3 0x0087
DMC_SEC_RANGE14_WID_CTRL0 0x0088
DMC_SEC_RANGE14_WID_CTRL1 0x0089
DMC_SEC_RANGE14_WID_CTRL2 0x008a
DMC_SEC_RANGE14_WID_CTRL3 0x008b
DMC_SEC_RANGE15_WID_CTRL0 0x008c
DMC_SEC_RANGE15_WID_CTRL1 0x008d
DMC_SEC_RANGE15_WID_CTRL2 0x008e
DMC_SEC_RANGE15_WID_CTRL3 0x008f
DMC_DES_KEY0_H 0x0090
DMC_DES_KEY0_L 0x0091
DMC_DES_KEY1_H 0x0092
DMC_DES_KEY1_L 0x0093
DMC_DES_PADDING 0x0094
DMC_DES_CTRL 0x0095

Bit(s)	R/W	Default	Description
1			DES enable. 1: DES enable. 0: DES disable. default is 1.
0			DES register mask. if write 1 only. after write 1, DES_CTRL, DES_KEY, DES_padding, and CFG_CA_REMAP regsiter can't be write and read.

DMC_CA_REMAP_L 0x009b

Bit(s)	R/W	Default	Description
63:60			New address for index 15
59:56			New address for index 14
55:52			New address for index 13
51:48			New address for index 12
47:44			New address for index 11
43:40			New address for index 10
39:36			New address for index 9
35:32			New address for index 8
31:28			New address for index 7

Bit(s)	R/W	Default	Description
27:24			New address for index 6
23:20			New address for index 5
19:16			New address for index 4
15:12			New address for index 3
11:8			New address for index 2
7:4			New address for index 1
3:0			New address for index 0

DMC_CA_REMAP_H 0x009c

Bit(s)	R/W	Default	Description
63:60			New address for index 15
59:56			New address for index 14
55:52			New address for index 13
51:48			New address for index 12
47:44			New address for index 11
43:40			New address for index 10
39:36			New address for index 9
35:32			New address for index 8
31:28			New address for index 7
27:24			New address for index 6
23:20			New address for index 5
19:16			New address for index 4
15:12			New address for index 3
11:8			New address for index 2
7:4			New address for index 1
3:0			New address for index 0

DMC_PROT0_RANGE 0x00a0

Bit(s)	R/W	Default	Description
31:16			Range end address.
15:0			Range end address.

DMC_PROT0_CTRL 0x00a1

Bit(s)	R/W	Default	Description
25			Protection 0 write access block function. If enabled, the access wouldn't write to the ddr sdram. If not enabled only generate a interrupt, but the access still wrote to ddr.
24			Protection range 0 enable.
23:16			Each bit to eanble one of the 8 ambus channal for the protection function.
15:0			Each bit to enable one of the 15 channel input for the protection function.

DMC_PROT1_RANGE 0x00a2

Bit(s)	R/W	Default	Description
31:16			Range end address.
15:0			Range end address.

DMC_PROT1_CTRL 0x00a3

Bit(s)	R/W	Default	Description
25			Protection 1 write access block function. If enabled, the access wouldn't write to the ddr sdram. If not enabled only generate a interrupt, but the access still wrote to ddr.
24			Protection range 1 enable bit.
23:16			Each bit to eanble one of the 8 ambus channal for the protection function.
15:0			Each bit to enable one of the 15 channel input for the protection function.

DMC_WTCH0_D0 0x00a4
DMC_WTCH0_D1 0x00a5
DMC_WTCH0_D2 0x00a6
DMC_WTCH0_D3 0x00a7
DMC_WTCH0_RANGE 0x00a8

Bit(s)	R/W	Default	Description
31:16			Start address high 16
15:0			Start address high 16

DMC_WTCH0_CTRL 0x00a9

Bit(s)	R/W	Default	Description
23:0			8 ambus and 16 axibus input channels select.

DMC_WTCH0_CTRL1 0x00aa

Bit(s)	R/W	Default	Description

31:16			Start address high 16.
2			Watch point 0 enable.
1:0			Watch point0 type. 2'b00: double bytes. Only watchpoint data 15:0 and data strb 1:0 is valid. 2'b01: 4 bytes. 2'b10: 8 bytes. 2'b11, all 16bytes.

DMC_WTCH1_D0 0x00ab

DMC_WTCH1_D1 0x00ac

DMC_WTCH1_D2 0x00ad

DMC_WTCH1_D3 0x00ae

DMC_WTCH1_RANGE 0x00af

Bit(s)	R/W	Default	Description
15:0			Start address high 16

DMC_WTCH1_CTRL 0x00b0

Bit(s)	R/W	Default	Description
23:0			8 ambus and 16 axibus input channels select.

DMC_WTCH1_CTRL1 0x00b1

Bit(s)	R/W	Default	Description
31:16			Start address high 16.
2			Watch point 0 enable.
1:0			Watch point0 type. 2'b00: double bytes. Only watchpoint data 15:0 and data strb 1:0 is valid. 2'b01: 4 bytes. 2'b10: 8 bytes. 2'b11, all 16bytes.

DMC_TRAP0_RANGE 0x00b2

Bit(s)	R/W	Default	Description
31:16			Trap0 end address
2			Start0 address.

DMC_TRAP0_CTRL 0x00b3

Bit(s)	R/W	Default	Description
31			Trap0 port ID 3 enable.
30			Trap0 port ID 2 enable.
29			Trap0 port ID 1 enable.
28			Trap0 port ID 0 enable.
27			Trap0 port ID 3 subid enable.

26			Trap0 port ID 2 subid enable.
25			Trap0 port ID 1 subid enable.
24			Trap0 port ID 0 subid enable.
16:20			Trap0 port ID 1 ID number.
14:11			Trap0 port ID 1 subid ID number.
8:4			Trap0 port ID 0 ID number.
3:0			Trap0 port ID 0 subid ID number.

DMC_TRAP0_CTRL2 0x00b4

Bit(s)	R/W	Default	Description
31:17			Not used.
16:20			Trap0 port ID 3 ID number.
14:11			Trap0 port ID 3 subid ID number.
8:4.			Trap0 port ID 2 ID number.
3:0			Trap0 port ID 2 subid ID number.

DMC_TRAP1_RANGE 0x00b5

Bit(s)	R/W	Default	Description
31:16			Trap end address
15:0			Start address

DMC_TRAP1_CTRL 0x00b6

Bit(s)	R/W	Default	Description
31			Trap0 port ID 3 enable.
30			Trap0 port ID 2 enable.
29			Trap0 port ID 1 enable.
28			Trap0 port ID 0 enable.
27			Trap0 port ID 3 subid enable.
26			Trap0 port ID 2 subid enable.
25			Trap0 port ID 1 subid enable.
24			Trap0 port ID 0 subid enable.
16:20			Trap0 port ID 1 ID number.
14:11			Trap0 port ID 1 subid ID number.

8:4			Trap0 port ID 0 ID number.
3:0			Trap0 port ID 0 subid ID number.

DMC_TRAP1_CTRL2 0x00b7

Bit(s)	R/W	Default	Description
31:17			Not used.
16:20			Trap1 port ID 1 ID number.
14:11			Trap1 port ID 1 subid ID number.
8:4.			Trap1 port ID 0 ID number.
3:0.			Trap1 port ID 0 subid ID number.

DMC_SEC_STATUS 0x00b8

Bit(s)	R/W	Default	Description
31:2			Not used.
1			Write security violation
0			Read security violation.

DMC_VIO_ADDR0 0x00b9
DMC_VIO_ADDR1 0x00ba

Bit(s)	R/W	Default	Description
31:24			Not used. Always 0.
23			ddr0 secure check violation.
22			ddr0 protection 1 vilation.
21			ddr0 protection 0 vilation.
20			ddr0 watch 1 catch
19.			ddr0 watch 0 catch.
18			ddr0 write address overflow. write out of DDR size.
17:15			ddr0 write violation AWPROT bits.
14:0			ddr0_write violation ID.

DMC_VIO_ADDR2 0x00bb
DMC_VIO_ADDR3 0x00bc

Bit(s)	R/W	Default	Description

23			ddr0 read secure check violation.
22			ddr0 read protection 1 violation.
21			ddr0 read protection 0 violation.
20			ddr0 read trap1 violation
19			ddr0 read trap0 violation
18			ddr 0 read address overflow. write out of DDR size.
17:15			ddr 0 read violation ARPROT bits.
14:0			ddr 0 read violation ID.

DDR0_ADDRMAP_0 0x00d0

Bit(s)	R/W	Default	Description
29:25			ca8
24:20			ca7
19:15			ca6
14:10			ca5
9:5			ca4
4:0			ca3

DDR0_ADDRMAP_1 0x00d1

Bit(s)	R/W	Default	Description
29:25			ra2
24:20			ra1
19:15			ra0
14:10			ca11
9:5			ca10
4:0			ca9

DDR0_ADDRMAP_2 0x00d2

Bit(s)	R/W	Default	Description
29:25			ra8
24:20			ra7
19:15			ra6
14:10			ra5

9:5			ra4
4:0			ra3

DDR0_ADDRMAP_3 0x00d3

Bit(s)	R/W	Default	Description
29:25			ra14
24:20			ra13
19:15			ra12
14:10			ra11
9:5			ra10
4:0			ra9

DDR0_ADDRMAP_4 0x00d4

Bit(s)	R/W	Default	Description
29:25			ra16 for DDR4 SDRAM
24:20			bg1 for DDR4 SDRAM.
19:15			ba2. or bg0 for DDR4.
14:10			ba1.
9:5			ba0.
4:0			ra15.

DDR1_ADDRMAP_0 0x00d5

Bit(s)	R/W	Default	Description
29:25			ca8
24:20			ca7
19:15			ca6
14:10			ca5
9:5			ca4
4:0			ca3

DDR1_ADDRMAP_1 0x00d6

Bit(s)	R/W	Default	Description
29:25			ra2
24:20			ra1

Bit(s)	R/W	Default	Description
19:15			ra0
14:10			ca11
9:5			ca10
4:0			ca9

DDR1_ADDRMAP_2 0x00d7

Bit(s)	R/W	Default	Description
29:25			ra8
24:20			ra7
19:15			ra6
14:10			ra5
9:5			ra4
4:0			ra3

DDR1_ADDRMAP_3 0x00d8

Bit(s)	R/W	Default	Description
29:25			ra14
24:20			ra13
19:15			ra12
14:10			ra11
9:5			ra10
4:0			ra9

DDR1_ADDRMAP_4 0x00d9

Bit(s)	R/W	Default	Description
29:25			ra16 for DDR4 SDRAM
24:20			bg1 for DDR4 SDRAM.
19:15			ba2. or bg0 for DDR4 SDRAM.
14:10			ba1.
9:5			ba0.
4:0			ra15.

DMC_DDR_CTRL 0x00da

Bit(s)	R/W	Default	Description
31:28			16bit selection for DDR3/4 not balanced mode. 4'b1000: 3Gbyte mode. low 2Gbyte is in 32bits mode. 2G~3G is 16bits mode. 4'b0100: 1.5Gbyte mode. low 1Gbyte is in 32bits mode. 1G~1.5G is in 16bits mode. 4'b0010: 768Mbyte mode. low 512Mbyte is in 32bits mode. 512M~768M is in 16bits mode. 4'b0001: 384Mbyte mode. low 25Mbyte is in 32bits mode. 256M~384M is in 16bits m0doe. 4'b000 or others balance mode. ether 32bits mode or 16bits mode depends on bit
27			0 : cavnas use 64bytes boundary 1 : canvas use 32bytes boundary.
24:22			3'b000 : ddr3 mode. 3'b001 : ddr4 mode. 3'b010 : lpddr3 mode. 3'b011 : lpddr4 mode.
21			rank1 enable bit. if 1, rank1 used the address map is as bit 5:3 defined.
20			DDR4 BG1 enable bit.
18			always 0.
16			1 only use 16bits data in a 32bits phy data interface. 0 : normal data interface.
5:3			DDR rank 1 size. 3'b000 : DDR rank 1 is 128Mbyte. 3'b001 : DDR rank 1 is 256Mbyte. 3'b010 : DDR rank 1 is 512Mbyte. 3'b011 : DDR rank 1 is 1Gbyte. 3'b100 : DDR rank 1 is 2Gbyte. 3'b101 : DDR rank 1 is 4Gbyte. others : reserved.
2:0			DDR rank 0 size. 3'b000 : DDR rank 0 is 128Mbyte. 3'b001 : DDR rank 0 is 256Mbyte. 3'b010 : DDR rank 0 is 512Mbyte. 3'b011 : DDR rank 0 is 1Gbyte. 3'b100 : DDR rank 0 is 2Gbyte. 3'b101 : DDR rank 0 is 4Gbyte. others : reserved.

DDR_APB_SEC_CTRL 0x00db

Bit(s)	R/W	Default	Description
19:16			DMC normal APB register secure control.
19:			1: all can write those register. 0: the APB_PROT[0] must match the bit 0 to write those regsiter.
15:12			DMC sticky APB regsiter secure control.
15.			1: All APB bus can write those registers. 0: The APB_PROT[0] must match the bit 12 to write those register.
11:8.			DMC DDR PLL clock related APB regsiter secure control.
11.			1: All APB bus can write those registers. 0: The APB_PROT[0] must match the bit 8 to write those register.
7:4.			DMC DDR SDRAM protocal contorl register contorl
11.			1: All APB bus can write those registers. 0: The APB_PROT[0] must match the bit 4 to write those register.
3:0.			LPDDR4 PHY APB regsiter secure control.
3			1: All APB bus can write those registers. 0: The APB_PROT[0] must match the bit 0 to write those register.

DMC_TEST_WRCMD_ADDR 0x00dc

DMC_TEST_RDRSP_ADDR 0x00dd

DMC_TEST_RDCMD_ADDR 0x00de

DMC_TEST_WDG 0x00df

Bit(s)	R/W	Default	Description
31:16			Write response watch dog.
15:0			Read response watch dog.

DMC_TEST_STA 0x00e0

Test start address.

For non-sha mode, the last 5 bits would be ignored. The test address at 32bytes boundary.

For sha mode, address must be in 64 bytes boundary. That mean the last 6 bits must be 0.

DMC_TEST_EDA 0x00e1

Test end address.

For non-sha mode, the last 5 bits would be ignored. The test address at 32bytes boundary.

For sha mode, address must be in 64 bytes boundary. That mean the last 6 bits must be 1.

DMC_TEST_CTRL 0x00e2

Bit(s)	R/W	Default	Description
31			enable test.

30			when enable test, enable the write to DDR function.
29			when enable test, enable the read from DDR function.
28			when enable test, enable the sha calculation function must be same as read enable but without write function.
27			enabe to compare data. when do the read enable to enable the error comparaion. suppose the read data should be same as the data in the write buffer.
26			reserved.
25			address generation type. 0: continuous increase the address in the range of test start address and test end address. 1: test module would pick the random address from test start address and test end address.
24			0 : use the DMC_TEST_NUM register as the counter of test numbers. For write if the write command number == the DMC_TEST_NUM, the write is done. For read if the read command number == the DMC TEST_num, the read id done. For one read command can be repeated repeat number times. 1 : finshed at end address.
23			1 : the first write is {WD3, WD2,WD1,WD0}, then the latter is the previous data plus a pattern.({ + WD7, + WD6, + WD5, + WD4}). 0 : the WDATA is the data in write register.
23			reserved.
22:20			read repeat times. for non-sha function, we can define multi times of the read. the test module would repeat the same adddress repeat times.
19			limit write. 0: no outstanding write request limitation. 1: limit the outstanding write commands to the number of bits [15:8]
18			limit read. 0. no outstanding read request limitation. 1. limit the read outstanding request to the number of bits[7:0].
17:16			sha mode for sha function enabled. 00 : not used. 01 : sha1. 2: sha2-256. 3: sha2_224. not used in GXL fixed to be Sha 2.
15:8.			write outstanding commands limit.
7:0			read outstanding commands limit.

DMC_TEST_NUM 0x00e3
DMC_TEST_ WD0 0x00e4
DMC_TEST_ WD1 0x00e5
DMC_TEST_ WD2 0x00e6
DMC_TEST_ WD3 0x00e7
DMC_TEST_ WD4 0x00e8
DMC_TEST_ WD5 0x00e9
DMC_TEST_ WD6 0x00ea
DMC_TEST_ WD7 0x00eb
DMC_TEST_ RD0 0x00ec
DMC_TEST_ RD1 0x00ed
DMC_TEST_ RD2 0x00ee
DMC_TEST_ RD3 0x00ef
DMC_TEST_ RD4 0x00f0
DMC_TEST_ RD5 0x00f1
DMC_TEST_ RD6 0x00f2
DMC_TEST_ RD7 0x00f3
DMC_TEST_ERR_ADDR 0x00f4
DMC_TEST_ERR_CNT 0x00f5
DMC_TEST_STS 0x00f6

Bit(s)	R/W	Default	Description
31			Test done bit. Write 1 to clean.
30			Indicate address err
29:5			Not used.
4			Sha done. Write 1 to clean.
3			Write done. Write 1 to clean.
2			Read done. Write 1 to clean
1			Write watchdog triggered. Write 1 to clean
0			Read watchdog triggered. Write 1 to clean.

DMC_TEST_SHA_MSG0 0x00f8

DMC_TEST_SHA_MSG1 0x00f9

DMC_TEST_SHA_MSG2 0x00fa

DMC_TEST_SHA_MSG3 0x00fb

DMC_TEST_SHA_MSG4 0x00fc

DMC_TEST_SHA_MSG5 0x00fd

DMC_TEST_SHA_MSG6 0x00fe

DMC_TEST_SHA_MSG7 0x00ff

The following registers' base address is 0xff638400. Each register takes 4 byte address.

Each register's final address = 0xff638400 + offset * 4.

DMC_DRAM_TMRD 0x0000

Bit(s)	R/W	Default	Description
3:0			tMRD.

DMC_DRAM_TRFC 0x0001

Bit(s)	R/W	Default	Description
9:0			tRFC

DMC_DRAM_TRP 0x0002

Bit(s)	R/W	Default	Description
21:16			tRP for precharge all banks.
5:0			tRP for precharge one bank.

DMC_DRAM_TRTW 0x0003

Bit(s)	R/W	Default	Description
5:0			tRTW

DMC_DRAM_TCL 0x0004

Bit(s)	R/W	Default	Description
5:0			CL/tRL. read latency.

DMC_DRAM_TCWL 0x0005

Bit(s)	R/W	Default	Description
5:0			CWL: write latency.

DMC_DRAM_TRAS 0x0006

Bit(s)	R/W	Default	Description

7:0			tRAS. minimum active to precharge time for same bank.

DMC_DRAM_TRC 0x0007

Bit(s)	R/W	Default	Description
7:0			tRC. minimum active to active time for same bank.

DMC_DRAM_TRCD 0x0008

Bit(s)	R/W	Default	Description
7:0			tRCD active to read/write timing for same bank.

DMC_DRAM_TRRD 0x0009

Bit(s)	R/W	Default	Description
21:16			tRRD_l active bank A to active B in same band group for DDR4.
5:0			tRRD/tRRD_s active bank A to active bank b time. tRRD_s: active bank A to active bank b in different bank grousp for DDR4.

DMC_DRAM_TFAW 0x000a

Bit(s)	R/W	Default	Description
8:0			tFAW. four active command windows

DMC_DRAM_TRTP 0x000b

Bit(s)	R/W	Default	Description
5:0			tRTP.

DMC_DRAM_TWR 0x000c

Bit(s)	R/W	Default	Description
5:0			tWR.

DMC_DRAM_TWTR 0x000d

Bit(s)	R/W	Default	Description
5:0			tWTR.

DMC_DRAM_TCCD 0x000e

Bit(s)	R/W	Default	Description
19:16			tCCD/tCCD_l.
3:0			tCCD/tCCD_s read to read command time or write to write command time.

DMC_DRAM_TEXSR 0x000f

Bit(s)	R/W	Default	Description
9:0			tEXSR. EXIT SELF-REFRESH to read/write command.

DMC_DRAM_TXS 0x0010

Bit(s)	R/W	Default	Description
9:0			tEXSR. EXIT SELF-REFRESH to read/write command.

DMC_DRAM_TXP 0x0011

Bit(s)	R/W	Default	Description
3:0			tXP. EXIT power down to other command time

DMC_DRAM_TXPDLL 0x0012

Bit(s)	R/W	Default	Description
9:0			tXPDLL, EXIT power down to read/write command time(need to relock PLL).

DMC_DRAM_TZQCS 0x0013

Bit(s)	R/W	Default	Description
7:0			ZQCS command to other command time.

DMC_DRAM_TCKSRE 0x0014

Bit(s)	R/W	Default	Description
4:0			enter self-refresh to disable clock time.

DMC_DRAM_TCKSRX 0x0015

Bit(s)	R/W	Default	Description
4:0			enable clock to exit self-refresh time.

DMC_DRAM_TCKE 0x0016

Bit(s)	R/W	Default	Description
4:0			CKE high or low minimum time.

DMC_DRAM_TMOD 0x0017

Bit(s)	R/W	Default	Description
4:0			tMOD. MRR/MRW to other command time.

DMC_DRAM_TDQS 0x0018

Bit(s)	R/W	Default	Description

3:0			tDQS. the delay to access different rank.

DMC_DRAM_TRSTL 0x0019

Not used.

DMC_DRAM_TZQLAT 0x001a

Bit(s)	R/W	Default	Description
5:0			ZQ LATCH command to other comand timing in LPDDR4 mode.

DMC_DRAM_TMRR 0x001b

Bit(s)	R/W	Default	Description
7:0			tMRR not used in DMC. not support MR READ.

DMC_DRAM_TCKESR 0x001c

Bit(s)	R/W	Default	Description
9:0			tCKESR. CKE low minimum pulse in self refresh mode.

DMC_DRAM_TDPD 0x001d

Not support.

DMC_DRAM_DFITCTRLDELAY 0x001e

Bit(s)	R/W	Default	Description
3:0			DFI_t_ctrldealy

DMC_DRAM_DFITPHYWRDATA 0x001f

Bit(s)	R/W	Default	Description
5:0			dfi_t_phy_wrdata.

DMC_DRAM_DFITPHYWRLAT 0x0020

Bit(s)	R/W	Default	Description
5:0			dfi_t_phy_wrlat. in DDR3/4/LPDDR3 mode: WL -5. in LPDDR4 mode: WL -5 + 2.

DMC_DRAM_DFITRDDATAEN 0x0021

Bit(s)	R/W	Default	Description
5:0			dfi_t_rddata_en. in DDR3/4/LPDDR3 mode: RL -5. in LPDDR4 mode : RL -5 + 1.

DMC_DRAM_DFITPHYRDLAT 0x0022

Bit(s)	R/W	Default	Description
5:0			dfi_t_rdlat.

DMC_DRAM_DFITCTRLUPDMIN 0x0023

Bit(s)	R/W	Default	Description
7:0			CTRLUPD_MIN minimux clock cycle to maintain CTRLUPD_REQ.

DMC_DRAM_DFITCTRLUPDMAX 0x0024

Bit(s)	R/W	Default	Description
7:0			CTRLUPD_MAX. maxmum clock cycle to maintain CTRLUPD_REQ if no CTRLUPD_ACK response.

DMC_DRAM_DFITMSTRRESP 0x0025

Not used.

DMC_DRAM_DFITREFMSKI 0x0026

Not used.

DMC_DRAM_DFITCTRLUPDI 0x0027

Not used.

DMC_DRAM_DFITDRAMCLK 0x0028

Bit(s)	R/W	Default	Description
17			dram clk1 enable.
16			dram clk0 enable.
15:8			DRAM CLK disable waiting time
7:0			DRAM CLK enable enable timer

DMC_DRAM_DFITLPRESP 0x002a

Bit(s)	R/W	Default	Description
3:0			dfi_lp_ctrl_req response time. after dfi_lp_ctrl_req asserted, and after response time if there's still no dfi_lp_ack respone, then drop the dfi_lp_ctrl_req.

DMC_DRAM_DFITPHYMSTR 0x002b

Bit(s)	R/W	Default	Description
15			1: enable DFIPHYMASTER INTERFACE 0 disable DFIPHYMSTR en response. always 0 we would use DFI_RETRAIN function for PHY retrain required from LPDDR4.

DMC_DRAM_TCKECK 0x002c

Bit(s)	R/W	Default	Description
4:0			tCKECK from CKE low to assert dfi_dram_clk_disable time. this time + dfi_t_ctrl_delay

DMC_DRAM_TREFI 0x002d

Bit(s)	R/W	Default	Description
31:24			tZQCI dmc send zqci period. unit is how much auto refresh period.
23:16			pvti dmc send dfi_ctrlupd_req period. unit is one auto refresh period.
15:8			tREFI.dmc send auto refresh command period. unit is 100ns.
7:0			t100ns period. unit is dmc clock cycles

DMC_DRAM_TSR 0x002e

Bit(s)	R/W	Default	Description
5:0			tSR. self resfresh enter to exit time.

DMC_DRAM_TCCDMW 0x002f

Bit(s)	R/W	Default	Description
5:0			4*tCCD in LPDDR4 mask write.

DMC_DRAM_TESCKE 0x0030

Bit(s)	R/W	Default	Description
5:0			tESCKE. enter self refresh to power time for LPDDR4.

DMC_DRAM_TREFI_DDR3 0x0031

Bit(s)	R/W	Default	Description
7:0			8*DDR3 SDRAM tREFI time . the unit is t100ns. use this to check in 8*tREFI time, the DMC should sent more than 16 auto REFRESH command.

DMC_NFQ_TMRD 0x0040
DMC_NFQ_TRFC 0x0041
DMC_NFQ_TRP 0x0042
DMC_NFQ_TRTW 0x0043
DMC_NFQ_TCL 0x0044
DMC_NFQ_TCWL 0x0045
DMC_NFQ_TRAS 0x0046
DMC_NFQ_TRC 0x0047
DMC_NFQ_TRCD 0x0048
DMC_NFQ_TRRD 0x0049
DMC_NFQ_TFAW 0x004a
DMC_NFQ_TRTP 0x004b
DMC_NFQ_TWR 0x004c
DMC_NFQ_TWTR 0x004d
DMC_NFQ_TCCD 0x004e
DMC_NFQ_TEXSR 0x004f
DMC_NFQ_TXS 0x0050
DMC_NFQ_TXP 0x0051
DMC_NFQ_TXPDLL 0x0052
DMC_NFQ_TZQCS 0x0053
DMC_NFQ_TCKSRE 0x0054
DMC_NFQ_TCKSRX 0x0055
DMC_NFQ_TCKE 0x0056
DMC_NFQ_TMOD 0x0057
DMC_NFQ_TDQS 0x0058
DMC_NFQ_TRSTL 0x0059
DMC_NFQ_TZQLAT 0x005a
DMC_NFQ_TMRR 0x005b
DMC_NFQ_TCKESR 0x005c

DMC_NFQ_TDPD 0x005d

DMC_NFQ_DFITCTRLDELAY 0x005e

DMC_NFQ_DFITPHYWRDATA 0x005f

DMC_NFQ_DFITPHYWRLAT 0x0060

DMC_NFQ_DFITRDDATAEN 0x0061

DMC_NFQ_DFITPHYRDLAT 0x0062

DMC_NFQ_DFITCTRLUPDMIN 0x0063

DMC_NFQ_DFITCTRLUPDMAX 0x0064

DMC_NFQ_DFITMSTRRESP 0x0065

DMC_NFQ_DFITREFMSKI 0x0066

DMC_NFQ_DFITCTRLUPDI 0x0067

DMC_NFQ_DFITDRAMCLK 0x0068

DMC_NFQ_DFITLPRESP 0x006a

DMC_NFQ_DFITPHYMSTR 0x006b

DMC_NFQ_TCKECK 0x006c

DMC_NFQ_TREFI 0x006d

DMC_NFQ_TSR 0x006e

DMC_NFQ_TCCDMW 0x006f

DMC_NFQ_TESCKE 0x0070

DMC_NFQ_TREFI_DDR3 0x0071

DMC_DRAM_DFITPHYUPDTYPE0 0x0080

dfi_phyupd_ack hold time for dfi_phyupd_req type = 0.

DMC_DRAM_DFITPHYUPDTYPE1 0x0081

dfi_phyupd_ack hold time for dfi_phyupd_req type = 1.

DMC_DRAM_DFITPHYUPDTYPE2 0x0082

dfi_phyupd_ack hold time for dfi_phyupd_req type = 2.

DMC_DRAM_DFITPHYUPDTYPE3 0x0083

dfi_phyupd_ack hold time for dfi_phyupd_req type = 3.

DMC_DRAM_DFIODTCFG 0x0084

Bit(s)	R/W	Default	Description
12			rank1 ODT default. default vulue for ODT[1] pins if theres no read/write activity.
11			rank1 ODT write sel. enable ODT[1] if there's write occur in rank1.
10			rank1 ODT write nsel. enable ODT[1] if theres's write occur in rank0.
9			rank1 odt read sel. enable ODT[1] if there's read occur in rank1.
8			rank1 odt read nsel. enable ODT[1] if there's read occure in rank0.

Bit(s)	R/W	Default	Description
4			rank0 ODT default. default vulue for ODT[0] pins if theres no read/write activity.
3			rank0 ODT write sel. enable ODT[0] if there's write occur in rank0.
2			rank0 ODT write nsel. enable ODT[0] if theres's write occur in rank1.
1			rank0 odt read sel. enable ODT[0] if there's read occur in rank0.
0			rank0 odt read nsel. enable ODT[0] if there's read occure in rank1.

DMC_DRAM_DFIODTCFG1 0x0085

Bit(s)	R/W	Default	Description
27:24			ODT length for BL8 read transfer.
19:16			ODT length for BL8 write transfer.
12:8.			ODT latency for reads. suppose to be 0.
4:0.			ODT latency for writes. suppose to be 0

DMC_DRAM_MCFG 0x0086

Bit(s)	R/W	Default	Description
12			1: dbi inversion. 0: dbi high inversion.
11			1: dbi read enable. 0: dbi not enabled.
10			1: enable staggered chip select for 2 ranks DRAM.
9			1: enable send auto refresh command to DDR SDRAM when PCTL is in CFG/STOP state.
8			send auto refr cmd before enter register triggered self refresh
4			send auto refr cmd after exit regsiter triggered self refresh mode.
3			disable dram clock after enter register triggered self refresh.
2			send DFI_LP_REQ to PHY after enter register triggered elf refresh mode.
1			send DRAM to power down mode after enter self refresh. ONLY for LPDDR4.
0			send DFI_CTRLUPD_REQ after exit register triggered self refresh.

DMC_DRAM_DFI_CTRL 0x0089

Bit(s)	R/W	Default	Description
20			siu_dfi1_phymstr_ack_en
19			siu_dfi_phymstr_req_and

18			siu_dfi_phymstr_req_or
17			siu_dfi_phymstr_type_sel
16			siu_dfi_phymstr_cs_sel
15			siu_dfi1_lp_en
14			siu_dfi_lp_ack_and
13			siu_dfi_lp_ack_or
12			siu_dfi1_init_start_en
11			siu_dfi_init_com_and
10			siu_dfi_init_com_or
9			siu_dfi1_freq_en
8			siu_dfi1_dram_clk_dis_en
7			siu_dfi_phyupd_type_sel
6			siu_dfi1_phyupd_ack_en
5			siu_dfi_phyupd_req_and
4			siu_dfi_phyupd_req_or
3			siu_dfi_ctrlupd_ack_and
2			siu_dfi_ctrlupd_ack_or
1			siu_dfi1_ctrlupd_req_en
0			siu_dfi1_cmd_en

DMC_DRAM_DFIINITCFG 0x008a

Bit(s)	R/W	Default	Description
31.			dfi_init_complete status. read only.
15:14			Frequency set 1 dfi_freq_ratio value.
12:8			Frequency set 1 dfi_freq value.
7:6			Frequency set 0 dfi_freq_ratio value.
5:1			Frequency set 0 dfi_freq value.
0.			dfi_init_start value can be use manually config dfi_init_start signal.

DMC_DRAM_ZQ_CTRL 0x008b

Bit(s)	R/W	Default	Description
2			send ZQCS command to RANK0 then send comand to RANK1.

1			send ZQCS command to both RANK0 and RANK1 together.
0			send ZQCS command to only rank0.

DMC_DRAM_APD_CTRL 0x008c

Bit(s)	R/W	Default	Description
19:16			DFI_LP_WAKEUP value in APD DFI_LP_REQ mode
12			1: exit power down slow mode(waiting PLL LOCK). 0 : fast mode.
11			enable DFI_LP_REQ when enter Auto power down mode.
10			disable DFI_clk_disable when enter auto power down mode.

DMC_DRAM_ASR_CTRL 0x008d

Bit(s)	R/W	Default	Description
23:20			DFI_LP_WAKEUP value in self refresh DFI_LP_REQ mode.
17			send REFRESH command after exit from auto self refersh mode(ASR).
16			send REFERSH command before enter to Auto self refresh mode(ASR).
15			send ZQCS command after exit from Auto self refresh mode(ASR).
14			send dfi_ctrl_upd after exit from ASR mode
13			send power down command when enter ASR mode. //for LPDDR4 only.
12			set the PHY enter LP2 mode after enter ASR mode.
11			send DFI_LP_REQ after enter ASR mode.
10			set DFI_CLK_DISABLE after enter ASR mode.
9:0			0 disable auto ASR mode.

DMC_DRAM_PHYMSTR_CTRL 0x0090

Not used.

DMC_DRAM_DFIODTRANKMAP 0x0091

Not used.

DMC_DRAM_REFR_CTRL 0x0092

Bit(s)	R/W	Default	Description
17:8			auto refresh request pending cnt if there's page hit request.
6			Disabled auto refresh command if over 16 auto refresh command sent in 2 TREFI_DDR3 period
5			enable dmc send ZQCS command .

4.			enable dmc send DFI_CTRUPD_REQ.
3:1			how many refresh command send for one period. = this number + 1
0			enable dmc send auto refresh command.

DMC_DRAM_FREQ_CTRL 0x0093

Bit(s)	R/W	Default	Description
31 .			Wiret 1 to change freqency read 0: finished.
30:9			Not used.
9.			1 : FREQ MRW done. Let FREQ change machine continue.
8			Freq wait. 1 when freq change finishes, state machine stop at self refresh state in case there's something need to handle.
			// 0 after freq change finishes the state machine go back to access state.
7			When change PLL setting, disable dmc clock
6			When change PLL setting, disable PHY dficlk and dfictlclk.
5			Check vpu_sleep_en ==1 when do FREQ change. If vpu_sleep_en == 0, just wait.
4			Nxt frequency selection. 1 = freq1. 0 = freq0.
3:1.			Not used.
0.			Current frequency selection.

DMC_DRAM_SCFG 0x0094

Bit(s)	R/W	Default	Description
2:0			only one bit can be high at same time.
2			1 : to ask PCTL enter ACCESS STATE. 0 : deassert the request.
1			1 : to ask PCTL enter SELF REFRESH STATE. 0 : deassert the request.
0			1 : to ask PCTL enter STOP/CONFIG STATE . 0 : deassert the request.

DMC_DRAM_STAT 0x0095

Bit(s)	R/W	Default	Description
27:23			dram_sr_state
22:20			stop_st
19:15			sleep_st
14:12			ACCESS STATUS 0 : ACCESS is in normal working mode. 1 : ACCESS sending precharege command. 2 : ACCESS sending AUTO REFESH command.

Bit(s)	R/W	Default	Description
			3 : ACCESS sending DIF_CTRLUPD_REQ command. 4 : ACCESS sending ZQCS command to DDR DRAM(ZQCAL for LPDDR4). 5 : ACCESS sending ZQLATCH command to LPDDR4 only.
11:8			APD STATUS: 0 : APD_IDLE 1 : APD sending PRECHARGE command 2 : APD sending CKE low command 3 : APD sending DISABLE DRAM CLOCK command 4 : APD sending DFI_LP_CTRL_REQ 5 : APD in Auto Power down mode. 6 : APD deassert DFI_LP_CTRL_REQ 7 : APD sending enable DRAM CLOCK command 8 : APD sending out CKE high command.
7:4			DRAM_STATUS: 0 : DRAM IDLE 1 : DRAM_STOP/DRAM_CFG 2 : DRAM_ACCESS 3 : DRAM_SLEEP 4 : DRAM APD(AUTO POWER DOWN). 5 : IDLE -> STOP/CONFIG 6 : STOP -> SLEEP 7 : STOP -> ACCESS 8 : ACCESS -> SLEEP. 9 : ACCESS -> STOP A : ACCESS -> APD B : SLEEP -> STOP C : SLEEP -> ACCESS D : APD -> ACCESS
3			Reserved.
2			1 : DRAM enter normal working state.
1			1 : DRAM enter sleep state. self refresh state.
0			1 : dram enter cfg state.

DMC_DRAM_STAT1 0x0096

Bit(s)	R/W	Default	Description
11:8			freq_st.
7:5			train_st
4:0			dram_phy_st

DMC_PHY_RETRAINING_CTRL 0x0097

Bit(s)	R/W	Default	Description
31			phy_retraining enable.
30			check vpu sleep_en.
23:0			retraining period unit : 100ns.

DMC_DFI_ERR_STAT 0x0098

Bit(s)	R/W	Default	Description
31:20			not used.
9			ddr0_dfi_error
8:5			ddr0_dfi_error_info.
4			ddr1_dfi_error.
3:0			ddr1_dfi_error_info.

DMC_DRAM_MRW_CTRL 0x0099

Bit(s)	R/W	Default	Description
31			write 1 to initial a MRW command read 0 finished.
23:16			MR addr. DDR4 case : 18:16 ba[2:0]. 29:19 BG[1:0].
15:0			opcode.

DMC_DRAM_DFI_SWAP_0 0x00a0

Bit(s)	R/W	Default	Description
5:0			dfi_act_n function select

DMC_DRAM_DFI_SWAP_1 0x00a1

Bit(s)	R/W	Default	Description
5:0			dfi_ras_n function select

DMC_DRAM_DFI_SWAP_2 0x00a2

Bit(s)	R/W	Default	Description
5:0			dfi_cas_n function select

DMC_DRAM_DFI_SWAP_3 0x00a3

Bit(s)	R/W	Default	Description
5:0			dfi_we_n function select

DMC_DRAM_DFI_SWAP_4 0x00a4

Bit(s)	R/W	Default	Description
5:0			dfi_bg0 function select

DMC_DRAM_DFI_SWAP_5 0x00a5

Bit(s)	R/W	Default	Description
5:0			dfi_bg[1] function select

DMC_DRAM_DFI_SWAP_6 0x00a6

Bit(s)	R/W	Default	Description
5:0			dfi_ba[0] function select

DMC_DRAM_DFI_SWAP_7 0x00a7

Bit(s)	R/W	Default	Description
5:0			dfi_ba[1] function select

DMC_DRAM_DFI_SWAP_8 0x00a8

Bit(s)	R/W	Default	Description
5:0			dfi_ba[2] function select

DMC_DRAM_DFI_SWAP_9 0x00a9

Bit(s)	R/W	Default	Description
5:0			dfi_a[0] function select

DMC_DRAM_DFI_SWAP_10 0x00aa

Bit(s)	R/W	Default	Description
5:0			dfi_a[1] function select

DMC_DRAM_DFI_SWAP_11 0x00ab

Bit(s)	R/W	Default	Description
5:0			dfi_a[2] function select

DMC_DRAM_DFI_SWAP_12 0x00ac

Bit(s)	R/W	Default	Description
5:0			dfi_a[3] function select

DMC_DRAM_DFI_SWAP_13 0x00ad

Bit(s)	R/W	Default	Description

5:0			dfi_a[4] function select

DMC_DRAM_DFI_SWAP_14 0x00ae

Bit(s)	R/W	Default	Description
5:0			dfi_a[5] function select

DMC_DRAM_DFI_SWAP_15 0x00af

Bit(s)	R/W	Default	Description
5:0			dfi_a[6] function select

DMC_DRAM_DFI_SWAP_16 0x00b0

Bit(s)	R/W	Default	Description
5:0			dfi_a[7] function select

DMC_DRAM_DFI_SWAP_17 0x00b1

Bit(s)	R/W	Default	Description
5:0			dfi_a[8] function select

DMC_DRAM_DFI_SWAP_18 0x00b2

Bit(s)	R/W	Default	Description
5:0			dfi_a[9] function select

DMC_DRAM_DFI_SWAP_19 0x00b3

Bit(s)	R/W	Default	Description
5:0			dfi_a[10] function select

DMC_DRAM_DFI_SWAP_20 0x00b4

Bit(s)	R/W	Default	Description
5:0			dfi_a[11] function select

DMC_DRAM_DFI_SWAP_21 0x00b5

Bit(s)	R/W	Default	Description
5:0			dfi_a[12] function select

DMC_DRAM_DFI_SWAP_22 0x00b6

Bit(s)	R/W	Default	Description
5:0			dfi_a[13] function select

DMC_DRAM_DFI_SWAP_23 0x00b7

Bit(s)	R/W	Default	Description
5:0			dfi_a[14] function select

DMC_DRAM_DFI_SWAP_24 0x00b8

Bit(s)	R/W	Default	Description
5:0			dfi_a[15] function select

DMC_DRAM_DFI_SWAP_25 0x00b9

Bit(s)	R/W	Default	Description
5:0			dfi_a[16] function select

DMC_DRAM_DFI_SWAP_26 0x00bb

Bit(s)	R/W	Default	Description
5:0			dfi_a[17] function select

The following registers' base address is 0xff638800. Each register takes 4 byte address.

Each register's final address = 0xff638800 + offset * 4.

DMC_STICKY_0	**0x0000**
DMC_STICKY_1	**0x0001**
DMC_STICKY_2	**0x0002**
DMC_STICKY_3	**0x0003**
DMC_STICKY_4	**0x0004**
DMC_STICKY_5	**0x0005**
DMC_STICKY_6	**0x0006**
DMC_STICKY_7	**0x0007**
DMC_STICKY_8	**0x0008**
DMC_STICKY_9	**0x0009**
DMC_STICKY_10	**0x000a**
DMC_STICKY_11	**0x000b**
DMC_STICKY_12	**0x000c**
DMC_STICKY_13	**0x000d**
DMC_STICKY_14	**0x000e**
DMC_STICKY_15	**0x000f**
DMC_STICKY_16	**0x0010**
DMC_STICKY_17	**0x0011**
DMC_STICKY_18	**0x0012**
DMC_STICKY_19	**0x0013**
DMC_STICKY_20	**0x0014**
DMC_STICKY_21	**0x0015**
DMC_STICKY_22	**0x0016**
DMC_STICKY_23	**0x0017**

DMC_STICKY_24	**0x0018**
DMC_STICKY_25	**0x0019**
DMC_STICKY_26	**0x001a**
DMC_STICKY_27	**0x001b**
DMC_STICKY_28	**0x001c**
DMC_STICKY_29	**0x001d**
DMC_STICKY_30	**0x001e**
DMC_STICKY_31	**0x001f**
DMC_STICKY_32	**0x0020**
DMC_STICKY_33	**0x0021**
DMC_STICKY_34	**0x0022**
DMC_STICKY_35	**0x0023**
DMC_STICKY_36	**0x0024**
DMC_STICKY_37	**0x0025**
DMC_STICKY_38	**0x0026**
DMC_STICKY_39	**0x0027**
DMC_STICKY_40	**0x0028**
DMC_STICKY_41	**0x0029**
DMC_STICKY_42	**0x002a**
DMC_STICKY_43	**0x002b**
DMC_STICKY_44	**0x002c**
DMC_STICKY_45	**0x002d**
DMC_STICKY_46	**0x002e**
DMC_STICKY_47	**0x002f**
DMC_STICKY_48	**0x0030**
DMC_STICKY_49	**0x0031**
DMC_STICKY_50	**0x0032**
DMC_STICKY_51	**0x0033**
DMC_STICKY_52	**0x0034**
DMC_STICKY_53	**0x0035**
DMC_STICKY_54	**0x0036**
DMC_STICKY_55	**0x0037**
DMC_STICKY_56	**0x0038**
DMC_STICKY_57	**0x0039**
DMC_STICKY_58	**0x003a**
DMC_STICKY_59	**0x003b**
DMC_STICKY_60	**0x003c**

DMC_STICKY_61 0x003d

DMC_STICKY_62 0x003e

DMC_STICKY_63 0x003f

13.2 NAND

13.2.1 Overview

S922X supports SLC/MLC/TLC NAND Flash with 60-bit ECC.

13.2.2 Descriptor Commands

Command “Standby”

All CEs are high, all other signals are “don’t care”, the bits [9:0] specify how many extra NAND cycles the controller stays in “standby”, if the number is “0”, only one NAND cycle, if the number is 10, the controller will stay in “standby” for 11 NAND cycles.

Command “Idle”

CE is low, ALE and CLE is low, WE and RE is high, the NAND bus is taken over by controller, but the bus is let to idle for one NAND cycle, if extra NAND cycles number is set, then extra NAND cycles is idled, useful when NAND controller needs time to switch to another mode, or wait for RB.

Command “Command”

NAND command is sent, CE is low, ALE is low, CLE is high, WE is low and RE is high, usually only one NAND CE is low, all the other 3 should be high, otherwise the NAND command is accepted by multiple Dies and may cause conflict problem, but it is fine when used to reset the NAND, the command itself is sent by bit 7 to bit 0, one command sends one NAND command.

Command “Address”

NAND address cycle, the CE is low, ALE is high, CLE is low, WE is low and RE is high. Usually only one NAND CE is low, all the other 3 CEs should be high. The address is sent in the bit 7 to bit 0, it lasts one NAND cycle.

Command “Data to NAND”

Write NAND bus directly by this command, CE is low, ALE and CLE are low, WE is low and RE is high, the low bit7 to bit 0 is write to NAND flash when this command is done, it lasts one NAND command cycle. Usually used to program NAND features, it is very low efficient to program NAND by this command, could be used to debug NAND flash software.

Command “NAND to Data”

Read NAND bus and save the data to registers, CE is low, ALE and CLE are low, WE is high and RE is low, the NAND output is locked in registers at perfect timing, the extra NAND cycles will make this command repeats extra NAND cycle, if it is “0”, only one byte is locked in registers, since the registers used to lock the data is only 32 bits, extra NAND cycles number larger than 3 will cause the data overflow. This command is used to read status and features from NAND, not for read NAND data, it is very low efficient to read NAND by this command, it could be used to debug NAND software.

Command “Sync Read”

Same as “NAND to Data”, used in synchronous mode, each NAND cycle will read in two bytes in synchronous mode.

Command “RB pin”

Hardware ready/busy detect with timer and interrupt, the timer is bit 4 to bit 0, it is power of 2 NAND cycle, for example, if timer is set to 10, then 2^10 or 1024 NAND cycles is set to timer out the RB command. The maximum setting is 31, that is 2^31 NAND cycles. Or 42 seconds when NAND cycle is 20 ns. When the RB waiting is timed out, the hardware is still in “waiting”, an interrupt is sent to CPU, the “RB” command can be programmed to send out interrupt when RB is high when the correspond INT bits is set to high, once RB is high, the command queue advanced to next command, no matter interrupt or not, interrupt setting is “one time setting”, it means one RB command can be programmed with interrupt, the next can be programmed without interrupt. There are few bits in the RB command for software debugging purpose.

it can be set to different numbers to tell which RB command is interrupted. CPU can read out the NAND status registers to tell which RB causes interrupt.

Command “RB IO”

assume the NAND controller is used without ready/busy pin, the ready/busy is checked through status register, This command should be used with NAND read status command, like the old RB command, this command is designed with option to check which IO bits as ready/busy, “IO6”, “IO5” and “IO4”, and “INT6”, “INT5” and “INT4” to issue “IRQ” if “Ready”, time out logic is the same as old RB, in case of “Read” NAND, an extra “0x00” command is needed to switch NAND flash back to “Read” mode.

The RB IO command is associated with CE state, there is no CE option in RB IO command, valid CE from previous command is used.

NAND status is checked every 16 NAND cycles, if not ready when time out, an “IRQ” is issued and timer is reset.

Change to read status mode.

Set proper timer out parameter.

Monitor which IO bits, IO6, IO5 or IO4.

Set IRQ option, set NAND_CFG cmd_irq_en bit.

Return back to read mode if needed.

Command “MEM to NAND”

DMA command to send data from memory to NAND, this command consist of ECC mode, the data is read from DDR memory by NAND controller at 64 bits/system cycle, BCH encoded by hardware and sent to NAND in the fly. The NAND data address can be programmed by register or command “set address”.

The data size can be any number, if ECC is off, the exact number of data bytes are read from DDR memory and sent to NAND without ECC encode, if ECC is on, and final page is not 512 bytes, then “0xff” is appended to the data to make it 512 bytes and send to BCH encoded and send to NAND.

When ECC is off, no spare data bytes are read from DDR, the data is sent exactly the same as DDR. When ECC is on, 2/16/0 bytes of spare bytes are read from DDR and appended to 512/1024 main data, encoded and parity bytes all sent to NAND.

The final BCH code word is rounded to integer number of bytes, see BCH table.

If ECC is OFF, the command writes [13:0] number of bytes to NAND.

If ECC is ON, the command writes “pages” of bytes to NAND.

If “short” is 0, the page size is BCH code data size, 512 or 1024.

If “short” is 1, shortened BCH mode, the page size is “page size”*8 bytes.

The number of pages is up to 63.

Data randomization option, if cmd[19] is set to “1” and the random seed is not equal to “0”, the data and parity written to NAND is randomized with PRBS random number, the same seed must be used to read the page back.

Random seed is set by command “Seed” cmd[14:0], suggest to use NAND physical address as “seed”.

Command “NAND to MEM”

DMA command to read data from NAND and save to DDR, this command consist of ECC mode, and data size, this command will correct the data by hardware and save the decode result information to DDR too. The data address in DDR is programmed by “set address” command, if not programmed the address increases to next available data space.

When ECC is off, the data is save to DDR without change, the information bytes is also saved in DDR with ECC mode bits off, no spare bytes, when ECC is on, the data is BCH decoded and corrected by hardware,

if the data size is less than 512/1024, more data will be read out from NAND, but only the required number of bytes are sent to DDR.

If ECC is OFF, the command reads [13:0] number of bytes from NAND.

If ECC is ON, the command reads “pages” of bytes to NAND.

If “short” is 0, the page size is BCH code data size, 512 or 1024.

If “short” is 1, shortened BCH mode, the page size is “page size”*8 bytes.

The number of pages is up to 63.

Data randomization option is the same as “MEM to NAND”.

Command “Set Address”

DDR address setup command, used to change the memory address when read or write NAND, the D/S bits is for data or spare, “0” for data, “1” for spare, L/H is for low and high 16 bits of address, “0” for low 16 bits, “1” for high 16 bits, the DDR address is 32 bits.

This command has nothing to do with NAND bus, but it will take one NAND command to execute, one NAND idle command is used in time.

Mode	Cmd[18:17]	Description
Data	00	Set Data Address
Info	01	Set Info Address
Status	10	Set Status Address

Command “STS”

Read status and save to DDR memory, the status registers is 32 bits, with

Field	Name	Description
15:0	Sts	Status
23:16	Sts counter	Sequence number
31	Done	This status is valid.

STS 1: read status once, without IRQ.

STS 2: read status twice, with IRQ.

If NAND_CFG[20] Sts_irq_en is set to “1”, the STS 2 command will issue an IRQ to system, if used after NAND read command “N2M”, it guarantees the DMA is done. Because the STS itself is a DMA command, it will wait till the previous DMA command done before its DMA.

STS 2 command resets the sequence number, the sequence number in itself is the number of STS 1’s before IRQ, wrap around to 8 bits.

Command “Set Seed”

Random seed for data randomizer, any number except 0 starts random number generator, if “N2M” or “M2N” random bit is set, the random number is XORed with data.

Set none zero random seed, in cmd[14:0].

Set cmd[19] of “N2M” or “M2N” to 1, enable scrambler.

Use same seed to program/read the same NAND page.

13.2.3 Register Definitions

The base address of NAND registers is 0xffe07000, and the final address of each register is listed below:

Table 13-1 NAND Register List

Register Name	Description	Address	R/W
P_NAND_CMD	Write Command and Read Status	Base + 0x00	R/W
P_NAND_CFG	Configuration	Base + 0x04	R/W
P_NAND_DADR	Data Address	Base + 0x08	R/W
P_NAND_IADR	Information Address	Base + 0x0c	R/W
P_NAND_BUF	Read Data Buffer	Base + 0x10	R
P_NAND_INFO	Information	Base + 0x14	R
P_NAND_DC	DDR interface	Base + 0x18	R
P_NAND_ADR	DDR Address	Base + 0x1c	R
P_NAND_DL	DDR Low 32 Bits Data	Base + 0x20	R/W
P_NAND_DH	DDR High 32 Bits Data	Base + 0x24	R/W
P_NAND_CADR	Command Queus Address	Base + 0x28	R/W
P_NAND_SADR	Status Address	Base + 0x2c	R/W
P_NAND_PINS	CS2: SDRAM/NAND pin sharing	Base + 0x30	R/W
P_NAND_VER	Version number	Base + 0x38	R

P_NAND_CMD

Write : Send NAND command to controller, the command format is specified in previous section.

Bit(s)	Name	Description
21:0	Cmd	NAND command sent to NAND queue buffer
30	Cmd_go	When 1, and NAND bus is in waiting Rb mode, due to time out or longer than expected Rb waiting, the command queue will move on by disable RB waiting in current command.
31	Cmd_reset	When 1 the NAND command queue buffer is reset to zero.

Read : Read NAND controller status

Bit(s)	Name	Description
19:0	Cmd_curr	NAND command current on NAND bus, still going, not finished.
24:20	Cmd_cnt	Number of NAND commands still in NAND command queue buffer, the buffer size is 32.
25	Timer out	When 1 wait Rb command timed out.

26	Rb0	Current Rb0 status, 1: ready, 0: busy.
27	Rb1	Current Rb1 status, 1: ready, 0: busy.
28	Rb2	Current Rb2 status, 1: ready, 0: busy.
29	Rb3	Current Rb3 status, 1: ready, 0: busy.
30	Mem_rdy	When 1, DDR interface is idle and ready to accept memory movement request.
31	Ecc_rdy	When 1, ECC BCH encoder/decoder is idle and read to accept encode or decode activity.

P_NAND_CFG

Bit(s)	Name	Description
4:0	Bus_cyc	The number of system clock cycles in one NAND cycle – 1, for example, if the bus_cyc is 3, then the NAND cycle is 4 system clock cycles, the minimum setting is 3, the maximum setting is 31. program this register according NAND timing mode.
9:5	Bus_tim	The timing to lock the NAND data when read NAND data or status, please refer to “Timing Calculator” for details.
11:10	Sync	00: Async mode 01: Micron Sync mode 10: Toshiba-Samsung toggle mode
12	Cmd_start	When set to “1”, if the NAND controller internal 32 command buffer has less than 16 commands, the command DMA starts reading commands from DDR and saves them to internal buffer, the DMA keeps watching the internal buffer, reads whenever there are less than 16 commands left, if an all “zero” command is met, This bit is cleared and current command DMA is done.
13	Cmd_auto	When set to ‘1’, the command DMA will check the previous command queue end location, whether it is changed from all “zero” back to valid command, the auto check period is 1 ms.
14	Apb_mode	Special NAND mode for ROM boot or debug, when 1, DDR interface is redirected to APB register, all the read/write activities are through APB registers. When used in ROM boot and DDR is not ready.
15	Spare_only	When 1, the NAND controller read NAND with/without ECC, but only save the information bytes into DDR memory, the main data is discarded, designed for software to survey the NAND flash spare bytes and prepare NAND programming.
16	Sync_adj	Used to adjust data timing in sync or toggle mode, 0: default timing, 1: delay 1 system clock cycle.
20	Sts_irq_en	Enable STS IRQ.
21	Cmd_irq_en	Enable RB pin or RB IO IRQ.
26	Oob_on	Set to 1 oob_mode 16/0, Set to 0 no oob bytes.
27	Oob_mode	New in M8 v2, Set to 1 enable new oob mode. First page 16 bytes, all other pages 0 byte.

Bit(s)	Name	Description
28	Dc_ugt	Set NAND controller DDR interface to Urgent mode.
29	Nand_wpn	When 1, the NAND wpn pin is set to low, the NAND is in write protection mode, default to 0, the NAND is not protected.
30	Core_power	When 1, internal NAND controller core clock gating is override to always on. The clock gating is disabled.
31	Top_power	When 1, internal NAND top clock gating is override to always on, the clock gating for top is disabled.

P_NAND_DADR

Set DDR data address by registers, the address is 32 Bits, since the DDR data address can also be set by NAND commands, when both happens at the same time, register setting is ignored, the NAND command setting takes effect. Software should avoid conflict address setting.

P_NAND_IADR

Set DDR information (spare bytes) address by registers, the address is 32 Bits, since the DDR information address can also be set by NAND commands, when both happens at the same time, register setting is ignored, the NAND command setting takes effect. Software should avoid conflict address setting.

P_NAND_BUF

When read NAND status, features or data, the results are buffer in this register, the register is 32 Bits, it can only hold 4 bytes, if the host not doesn't read out the results, it will be over written by the following "read".

P_NAND_INFO

One 32 Bits information per each 512 bytes in ECC mode.

Bit(s)	Name	Description
7:0	Info 0	Information (spare) byte 0, errors already corrected by BCH.
15:8	Info 1	Information (spare) byte 1, errors already corrected by BCH
21:16	Pages	Count down page number in current DMA read, starts from the total page size, count down to 1.
28:24	Errcnt	Number of errors corrected by BCH in current page, 0 means no error in current page, 0x1f means this page is uncorrectable.
29	Unc	When 1, this page is uncorrectable by BCH, this page is bad.
30	Ecc	When 1, current NAND read is with ECC on.
31	Done	When 1, the information content and data read from NAND are valid, otherwise the "read" is not done.

P_NAND_DC

Used for apb_mode, internal NAND controller still uses DDR interface, only the DDR request and DDR grant are redirected from DDR to apb registers, this enables the host to read NAND without DDR, for ROM boot and debug.

Bit(s)	Name	Description
7:0	Dc_wr_dm	DDR write data mask, Each Bit masks one byte.
8	Dc_wr	When 1, write data from NAND to DDR. When 0, read data from DDR to NAND.
9	Dc_lbrst	When 1, the 64 Bits data is the last in current DDR burst, used with Dc_req.
10	Dc_ugt	When 1, current DDR request of read/write is urgent, in NAND controller, the Dc_ugt is set to 0, none-urgent.
11	Dc_req	When 1, the NAND controller send request to DDR to read or write data, in case of apb_mode, when dc_req is “1”, the host is responsible for send to or receive from NAND controller. Note: this is the only signal the host needs to check when in apb_mode.

P_NAND_ADR

32 Bits DDR address when NAND controller read or write to DDR memory, any address space within the DDR memory installed in the system is valid.

P_NAND_DL

The DDR interface uses 64 Bits width bus, this register is for low 32 Bits, [31:0].

P_NAND_DH

The DDR interface uses 64 Bits width bus, this register is for high 32 Bits, [63:32].

Read and write to this register will generate “grant” and “dc_wr_avail” or “dc_rd_avail”.

Always read or write low 32 Bits first, then read or write to high 32 Bits and NAND controller hardware will generate “grant” and “dc_wr_avail” or “dc_rd_avail”, combined with the data, the DDR address advances to next address.

NAND controller programs NAND flash in apb_mode:

- Check P_NAND_DC till “dc_req” is high.
- Write low 32 Bits to P_NAND_DL.
- Write high 32 Bits to P_NAND_DH.
- Go back to beginning till all the data is written.

NAND controller reads data from NAND flash in apb_mode:

- Check P_NAND_DC till “dc_req” is high.
- Read low 32 Bits from P_NAND_DL.
- Read high 32 Bits from P_NAND_DH.
- Go back to beginning till all the data is read.

Since the DDR interface in NAND controller process the data in group of 16 double words (64 Bits), the software can only check “dc_req” at the beginning of each 16 double words.

P_NAND_CADR

Set command queue memory address, 32 Bits, any memory location.

This address can only be programmed by APB bus.

P_NAND_SADR

Set status memory location, 32 Bits, any memory location.

This address can also be programmed through command queue.

P_NAND_PINS

Bit(s)	Name	Description
13:0	Pins_len	When pins are acquired by NAND, it will use Pins_len number of NAND bus cycles before releasing pins. Defult is 8.
27:14	Pins_off	When pins are released by NAND, it will wait Pins_off number of NAND bus cycles before sending next request. Default is 2.
31	Not shared	When 1 the pins is not shared, default is 0, pins are shared.

13.3 eMMC/SD

13.3.1 Overview

S922X has the following features of eMMC/SD

- Supports SDSC/SDHC/SDXC card
- Supports eMMC and MMC card specification version 5.0 up to HS200 with data content DES crypto
- 1 bit, 4 Bits, 8 Bits date lines supported (8 Bits only for MMC)
- Descriptor chain architecture, timing tuning and adjustment
- Supports descriptor-based internal DMA controller

This module uses eMMC/SD CONTROLLERs to connect varied SD/MMC Card, or eMMC protocol compatible memory with high throughput.

13.3.2 PinDescription

Table 13-2 PinDescription of eMMC/SD/SDIO Module

Name	Type	Description	Speed (MHz)
CLK	Output	SD eMMC clock, 0~200MHz	200
DS	Reserverd	-	-
DAT[7:0]	SD Card 4 Bits, eMMC 8 Bits Input/Output/Push-Pull Internal pull-up for pins not used	Data, 1,4,8 mode	200
CMD	Input/Output/Push-Pull/Open-Drain Open-drain for initialization Push-pull for fast command transfer ROD is connected when in open-drain mode.	Command Response	100
Rst_n	eMMC required	Hardware reset	Low
IRQ	SD Card or eMMC not used.	Device interrupt can be replaced by DAT[1]	Low

13.3.3 eMMC/SD Mode

eMMC Mode:

Table 13-3 eMMC Mode

Mode Name	Data Rate	IO Voltage	Bus Width	Frequency	Max Data Transfer
Legacy MMC card	Single	3/1.8V	1, 4, 8	0-26MHz	26MB/s
High Speed SDR	Single	3/1.8V	1,4, 8	0-52MHz	52MB/s
High Speed DDR	Dual	3/1.8V	4, 8	0-52MHz	104MB/s

Mode Name	Data Rate	IO Voltage	Bus Width	Frequency	Max Data Transfer
HS200	Single	1.8V	4, 8	0-200MHz	200MB/s

The HS200 mode offers the following features:

- SDR Data sampling method
- CLK frequency up to 200MHz Data rate – up to 200MB/s
- 4 or 8-bits bus width supported
- Signaling levels of 1.8V
- Tuning concept for Read Operations

SD Mode:

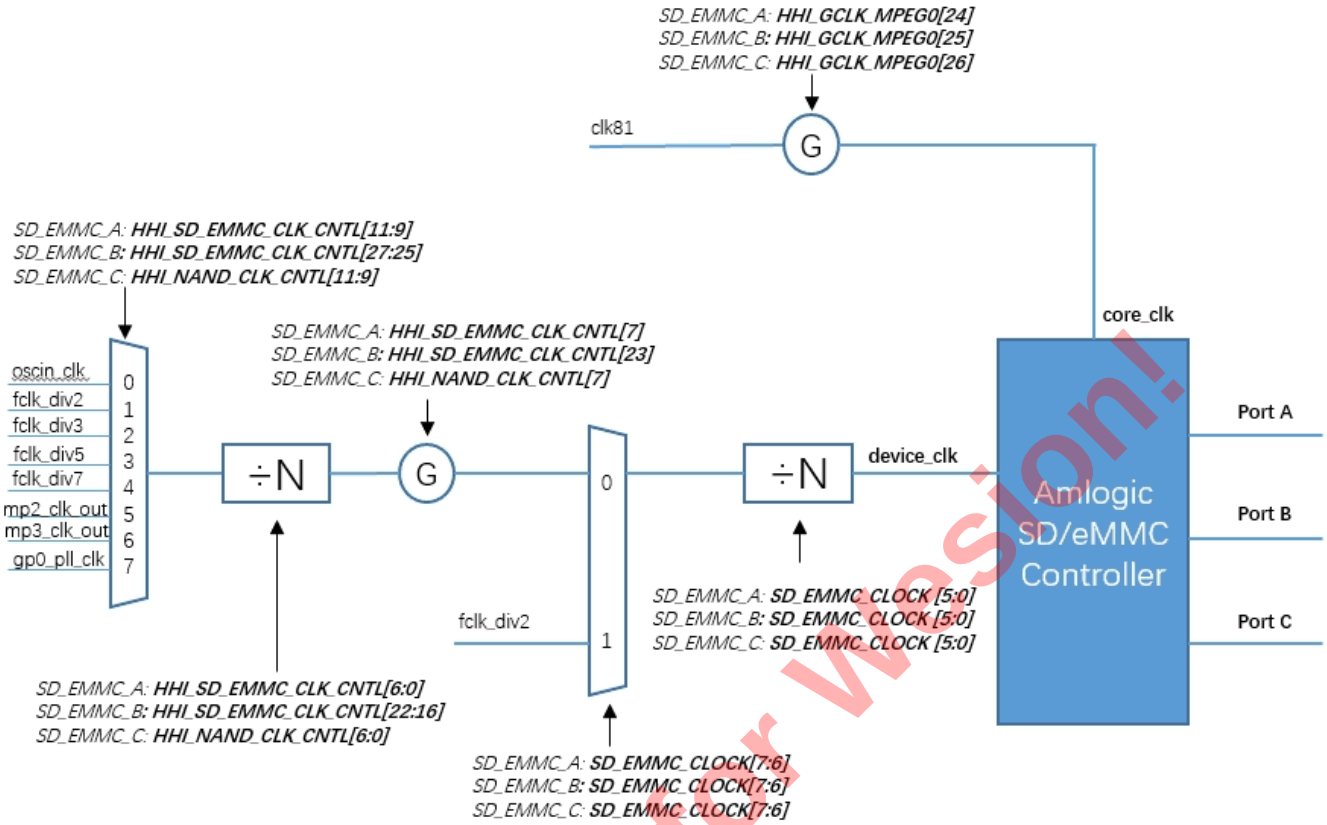
Table 13-4 SD Mode

Mode Name	Data Rate	IO Voltage	Bus Width	Frequency	Max Data Transfer
Default Speed	Single	3.3V	1, 4	0-25MHz	12.5MB/s
High Speed	Single	3.3V	1, 4	0-50MHz	25MB/s
SDR12	Single	1.8V	1,4	0-25MHz	12.5MB/s
SDR25	Single	1.8V	1,4	0-50MHz	25MB/s
SDR50	Single	1.8V	1,4	0-100MHz	50MB/s
SDR104 (highest)	Single	1.8V	1,4	0-208MHz	104MB/s
DDR50	Dual	1.8V	4	0-50MHz	50MB/s

13.3.4 Clock

In S922X, clock source has the options to select from the figure below.

Figure 13-2 EMMC Clock Diagram



13.3.5 Descriptor

13.3.5.1 Descriptor Structure

The descriptor has a size of 4x32 Bits.

Table 13-5 Descriptor Structure

byte	7	6	5	4	3	2	1	0
0	length[7:0]							
1	Timeout 4 Bits				End of chain	R1b	block mode	length[8]
2	data num	resp num	resp 128	resp nocrc	Data wr	Data io	No cmd	No resp
3	owner	error	cmd index 6 Bits					
4	cmd argument 32 Bits							
5								
6								
7								
8	data address 32 Bits or data 0-4 bytes							
9	[1]Big Endian, [0]SRAM							

byte	7	6	5	4	3	2	1	0
10								
11								
12	response address 32 Bits or response irq en [0]SRAM							
13								
14								
15								

13.3.5.2 Descriptor Definition

Table 13-6 Descriptor Definition

Name	Bits	Description
Length	Cmd_cfg[8:0]	same as spec, copy the content from command argument into this field, different byte size and 512 bytes, different number of blocks and infinite blocks. If the command is operating on bytes, block mode = 0, this field contains the number of bytes to read or write, A value of 0 shall cause 512 bytes to be read to written, if the command is operating on blocks, block mode = 1, this field contains the number of blocks, a value of 0 is infinite number of blocks.
Block_mode	Cmd_cfg[9]	1: the read or write shall be performed on block basis. The block size is from SD/eMMC device, and saved in APB3 register in module. 0: the read or write is byte based.
R1b	Cmd_cfg[10]	1: check the DAT0 busy after received response R1 0: do not check the DAT0 busy state.
End_of_chain	Cmd_cfg[11]	1: it is the end of descriptor chain, the host stops and issues IRQ after this descriptor is done. 0: the host reads next descriptor and continues. The command chain execution is started by write an APB3 register and stopped by the “end of chain” or clear a APB3 start register, or found one descriptor with owner is set to 0.
Timeout	Cmd_cfg[15:12]	2timeout ms when timeout != 0, max timeout 32.768s, when over the timeout limit, error bit is set, IRQ is issued. When timeout is 0, no time limit.
No_resp	Cmd_cfg[16]	1: this command doesn’t have response, used with command doesn’t have response. 0: there is a response. The module waits for response, the response timeout setting is in APB3 register.
No_cmd	Cmd_cfg[17]	1: this descriptor doesn’t have command in it, it does data DMA only, used with command to read or write SD/eMMC with data from multiple locations.
Data_io	Cmd_cfg[18]	1: there is data action in this descriptor, used with command have data process. 0: there is no data read/write action.

Name	Bits	Description
Data_wr	Cmd_cfg[19]	1: host writes data to SD/eMMC 0: host read data from SD/eMMC
Resp_nocrc	Cmd_cfg[20]	1: R3 response doesn't have CRC. 0: host does CRC check.
Resp_128	Cmd_cfg[21]	1: R3 response with 128 Bits information. 0: 32 Bits responses.
Resp_num	Cmd_cfg[22]	1: the resp_addr is the IRQ enable Bits, used to check the response error status, when there is an error, IRQ[14] is issued, the first 4 bytes of response is saved into resp_addr. 0: save response into SRAM or DDR location.
Data_num	Cmd_cfg[23]	1: save 4 bytes of data back into descriptor itself at bytes 8~11.
Cmd_index	Cmd_cfg[29:24]	The SD/eMMC command index. Desc REG wr: 4 reg44, 12 reg4c.
Error	Cmd_cfg[30]	Write back by host. The combined error from command, response, data, includes CRC error and timeout. When it is set the descriptor execution is stopped and an IRQ is issued. The CPU can read SD_EMMC_STATUS register to get detail information.
Owner	Cmd_cfg[31]	Programmed by CPU to 1, cleared by host to 0. 1: the descriptor is valid and owned by host, after it is done, even it has error, the owner bit is cleared, the descriptor is owned by CPU. In case of descriptor chain execution when host found a descriptor with "0" owner bit, it will stop.
Cmd_arg	Desc 4~7 bytes	32 Bits. The actual command argument some of the previous fields are copied from this command argument, the software need to make sure they are consistent. Desc REG wr: new value Data_addr: write mask, 1: change, 0: no change.
Data_addr	Desc 8~11 bytes	32 Bits. If the data_num is 0, the content is data address. If the data_num is 1, ths content is 4 data bytes. When it is an address: Data_addr[0]: 1: SRAM address, 0: DDR address. If the data_addr[31:12] matches with SD_EMMC_BASE, it is SRAM address. Data_addr[1]: 1: 4 bytes big endian, 0: little endian(default).

Name	Bits	Description
Resp_addr	Desc 12~15 bytes	32 Bits If the resp_num is 0, the content is resp address. If the resp_num is 1, before execution, it is the response IRQ enable Bits, after execution, it is the first 4 response bytes. When it is an address: Resp_addr[0]: 1: SRAM address, 0: DDR address. If the resp_addr[31:12] matches with SD_EMMC_BASE, it is SRAM address.

13.3.6 Register Description

Each register final address = module base address+ address * 4

Where module address addresses are 0xFFE07000 for portB(eMMC), 0xFFE05000 for port C (eMMC).

SD_EMMC_CLOCK 0x0

Bit(s)	R/W	Default	Description
30	R/W	0	Cfg_irq_sdio_sleep_ds : 1: select DS as IRQ source during sleep.
29	R/W	0	Cfg_irq_sdio_sleep: 1: enable IRQ sdio when in sleep mode. When DAT1 IRQ, the controller uses PCLK to detect DAT1 level and starts core clock, the core initials IRQ_period and detect DAT1 IRQ.
28	R/W	0	Cfg_always_on: 1: Keep clock always on 0: Clock on/off controlled by activities. Any APB3 access or descriptor execution will turn clock on. Recommended value: 0
27:22	R/W	0	Cfg_rx_delay: RX clock delay line, 6 bits 0: no delay, n: delay n*50ps Maximum delay 3150ps.
21:16	R/W	0	Cfg_tx_delay: TX clock delay line, 6 bits 0: no delay, n: delay n*50ps Maximum delay 63*50ps =3150ps.
15:14	R/W	0	Cfg_sram_pd: Sram power down
13:12			Cfg_rx_phase: RX clock phase 0: 0 phase, 1: 90 phase, 2: 180 phase, 3: 270 phase. Recommended value: 0

Bit(s)	R/W	Default	Description
11:10	R/W	0	Cfg_tx_phase: TX clock phase 0: 0 phase, 1: 90 phase, 2: 180 phase, 3: 270 phase. Recommended value: 2
9:8	R/W	0	Cfg_co_phase: Core clock phase 0: 0 phase, 1: 90 phase, 2: 180 phase, 3: 270 phase. Recommended value: 2
7:6	R/W	0	Cfg_src: Clock source 0: Crystal 24MHz or other frequencies selected by clock reset test control register. 1: Fix PLL, 1000MHz Recommended value: 1
5:0	R/W	0	Cfg_div: Clock divider Frequency = clock source/cfg_div Clock off: cfg_div==0, the clock is disabled Divider bypass: cfg_div==1, clock source is used as core clock without divider Maximum divider 63.

SD_EMMC_DELAY1 0x4

Bit(s)	R/W	Default	Description
31:30	R/W	0	Reserved.
29:24	R/W	0	Dly[4]: Data 4 delay line
23:18	R/W	0	Dly[3]: Data 3 delay line
17:12	R/W	0	Dly[2]: Data 2 delay line
11:6	R/W	0	Dly[1]: Data 1 delay line
5:0	R/W	0	Dly[0]: Data 0 delay line Total delay = 50ps * Dly When Dly == 0, no delay. When Dly ==63, 3150ps delay. NOTE: the 50ps is typical delay, actually delay may vary from chip to chip, from different temperature.

SD_EMMC_DELAY2 0x8

Bit(s)	R/W	Default	Description
31:30			Reserved.
29:24			Dly[9]: Data 9 delay line
23:18			Dly[8]: Data 8 delay line
17:12			Dly[7]: Data 7 delay line

Bit(s)	R/W	Default	Description
11:6			Dly[6]: Data 6 delay line
5:0			Dly[5]: Data 5 delay line Total delay = 50ps * Dly When Dly == 0, no delay. When Dly ==63, 3150ps delay. NOTE: the 50ps is typical delay, actually delay may vary from chip to chip, from different temperature.

SD_EMMC_ADJUST 0xC

Bit(s)	R/W	Default	Description
31:23			Unused
22	R/W	0	Adj_auto 1: Use cali_dut's first falling edge to adjust the timing, set cali_enable to 1 to use this function, simulation shows it can tracking 2.5ns range with 800ppm. 0: disable Working for HS200 mode, set Cali_enable to 1. Enhanced after gxlx project. Use RESP and DAT0 as reference, Separate RESP and DAT0, adjust the timing whenever there is a transition, insert a sample when there is no transition.
21:16	R/W	0	Adj_delay: Resample the input signals when clock index==adj_delay
15	R/W	0	Reserved
14	R/W	0	Cali_rise: 1: test the rising edge, recording rising edge location only. 0: test the falling edge
13	R/W	0	Adj_fixed: Adjust interface timing by resampling the input signals
12	R/W	0	Cali_enable: 1: Enable calibration 0: shut off to save power.
11:8	R/W	0	Cali_sel: Select one signal to be tested Signals are labeled from 0 to 9 the same as delay lines. Only one signal is tested at anytime. For example: Cali_sel == 9, test CMD line.

SD_EMMC_CALOUT 0x10

Bit(s)	R/W	Default	Description
31:16			Unused

15:8	R		Cali_setup
7	R		Cali_vld: The reading is valid When there is no rising edge or falling edge event, the valid is low, this reading is not valid.
5:0	R		Cali_idx: The event happens at this index, The index starts from rising edge of core clock from 0, 1, 2, …

SD_EMMC_ADJ_IDX_LOG 0x20

Bit(s)	R/W	Default	Description
31:30			Current, Auto_adj mode, Current 6 bits adj_idx
29:24			Previous 1
23:18			Previous 2
17:12			Previous 3
11:6			Previous 4
5:0			Previous 5: Last two bits of previous 5

SD_EMMC_CLKTEST_LOG 0x24

Bit(s)	R/W	Default	Description
31			Clktest_done, Test done
30:0			Clktest_times, Test clock core for 2^Clktest_exp

SD_EMMC_CLKTEST_OUT 0x28

Bit(s)	R/W	Default	Description
31:0			Clktest_out, All 2^clktest_exp test results add up. Note: divided by 2^clktest_exp and get average clock core period length measured by 50ps delay cells.

SD_EMMC_EYETEST_LOG 0x2c

Bit(s)	R/W	Default	Description
31			eyetest_done, Test done
30:0			eyetest_times, Test eye for 2^eyetest_exp

SD_EMMC_EYETEST_OUT0 0x30

Bit(s)	R/W	Default	Description
31:0			All 2^eyetest_exp test results “OR” together.

SD_EMMC_EYETEST_OUT1 0x34

Bit(s)	R/W	Default	Description

31:0			eyetest_out1, All 2^eyetest_exp test results “OR” together. Total eyeout [62:0] = {Eyetest_out1[30:0], eyetest_out0[31:0]} EYEtest output changed to 64 bits after TXLX and A113 {eyetest_out1, eyetest_out0}

SD_EMMC_INTF3 0x38

Bit(s)	R/W	Default	Description
26			Eyetest_sel, 0 : select core clock as eyetest clock. 1 : select DS after delay line as eyetest clock. Eyetest point is DS after delay line.
25:23			NAND_EDO, NAND Async interface EDO position after RE rising edge. [not for SD_eMMC]
22			Sd_intf3, Using SD interface 3
21:18			Ds_sht_exp, 0: DS shift setting never expires, always using the DS_sht_m as shift length. None-zero: 2^ds_sht_exp after 2^ds_sht_exp “ms”, the setting expired The internal FSM will automatically change the Ds shift setting.
17:12			Ds_sht_m, Shift DS by number of 50ps delay cells. If using auto FSM mode, it is the initial value.
11			Eyetest_on, 1: Turn on eye test After eyetest_done, set this bit to 0 to reset eyetest internal registers.
10:6			Eyetest_exp, Repeat the eye test for 2^eyetest_exp times, Or the test results together, report the final results.
5			Clktest_on_m, Manual turn on clock test
4:0			Clktest_exp, Repeat the clock test for 2^clktest_exp times. Add the clock length together, report the sum.

SD_EMMC_START 0x40

Bit(s)	R/W	Default	Description
31:2	R/W	0	Desc_addr[31:2]: Descriptor address, the last 2 Bits are 0, SRAM: 4 bytes aligned, the valid address range is from 0x200~0x3ff DDR: 8 bytes aligned the valid address is anywhere in DDR, the length of chain is unlimited. Desc_addr = ADDR>>2.

1	R/W	0	Desc_busy: Start/Stop 1: Start command chain execution process. 0: Stop Write 1 to this register starts execution. Write 0 to this register stops execution.
0	R/W	0	Desc_int: SRAM/DDR 1: Read descriptor from internal SRAM, limited to 32 descriptors. 0: Read descriptor from external DDR

SD_EMMC_CFG 0x44

Bit(s)	R/W	Default	Description
31:28	R/W	0	Cfg_ip_txd_adj: Data 1 interrupt, when in TXD mode, the data 1 irq is a input signal, the round trip delay is uncertain factor, change this cfg to compensate the delay.
27	R/W	0	Cfg_err_abort: 1: abort current read/write and issue IRQ 0: continue on current read/write blocks.
26	R/W	0	Cfg_irq_ds: 1: Use DS pin as SDIO IRQ input, 0: Use DAT1 pin as SDIO IRQ input.
25	R/W	0	Cfg_txd_retry: When TXD CRC error, host sends the block again. The total number of retries of one descriptor is limited to 15, after 15 retries, the TXD_err is set to high.
24	R/W	0	Cfg_txd_add_err: TXD add error test. Test feature, should not be used in normal condition. It will inverted the first CRC Bits of the 3rd block. Block index starts from 0, 1, 2, …
23	R/W	0	Cfg_auto_clk: SD/eMMC Clock Control 1: when BUS is idle and no descriptor is available, automatically turn off clock, to save power. 0:.whenever core clock is on the SD/eMMC clock is ON, it is still on/off during read data from SD/eMMC.
22	R/W	0	Cfg_stop_clk: SD/eMMC Clock Control 1: no clock for external SD/eMMC, used in voltage switch. 0: normal clock, the clock is automatically on/off during reading mode to back off reading in case of DDR slow response.
21	R/W	0	Cfg_cmd_low: Hold CMD as output Low eMMC boot mode.
20	R/W	0	Cfg_chk_ds

Bit(s)	R/W	Default	Description
19	R/W	0	Cfg_ignore_owner: Use this descriptor even if its owner bit is “0”.
18	R/W	0	Cfg_sdclk_always_on: 1: SD/eMMC clock is always ON 0: SD/eMMC clock is controlled by host. WARNING: Set SD/eMMC clock to always ON, host may lose data when DDR is slow.
17	R/W	0	Cfg_blk_gap_ip: 1: Enable SDIO data block gap interrupt period 0: Disabled.
16	R/W	0	Cfg_out_fall: DDR mode only The command and TXD start from rising edge. Set 1 to start from falling edge.
15:12	R/W	0	Cfg_rc_cc: Wait response-command, command-command gap before next command, 2cfg_rc_cc core clock cycles.
11:8	R/W	0	Cfg_resp_timeout: Wait response till 2cfg_resp_timeout core clock cycles. Maximum 32768 core cycles.
7:4	R/W	0	Cfg_bl_len: Block length 2cfg_bl_len, because internal buffer size is limited to 512 bytes, the cfg_bl_len <=9.
3	R/W	0	Cfg_dc_ugt: 1: DDR access urgent 0: DDR access normal
2	R/W	0	Cfg_ddr: 1: DDR mode 0: SDR mode
1:0	R/W	0	Cfg_bus_width: 0: 1 bit 1: 4 Bits 2: 8 Bits 3: 2 Bits (not supported)

SD_EMMC_STATUS 0x48

Bit(s)	R/W	Default	Description
31	R		Core_busy: 1: core is busy, desc_busy or sd_emmc_irq or bus_fsm is not idle. 0: core is idle.
30	R		Desc_busy: 1: Desc input process is busy, more descriptors in chain. 0: no more descriptor in chain or desc_err.

Bit(s)	R/W	Default	Description
29:26	R		Bus_fsm: BUS fsm
25	R		DS: Input data strobe
24	R		CMD_i: Input response signal
23:16	R		DAT_i: Input data signals
15	R/W		IRQ_sdio: SDIO device uses DAT[1] to request IRQ
14	R/W		Resp_status: When resp_num is set to 1, the resp_addr is the response status IRQ enable Bits, if there is an error.
13	R/W		End_of_Chain: End of Chain IRQ, Normal IRQ
12	R/W		Desc_timeout: Descriptor execution time over time limit. The timeout limit is set by descriptor itself. Consider the multiple block read/write, set the proper timeout limits.
11	R/W		Resp_timeout: No response received before time limit. The timeout limit is set by cfg_resp_timeout.
10	R/W		Resp_err: Response CRC error
9	R/W		Desc_err: SD/eMMC controller doesn't own descriptor. The owner bit is "0", set cfg_ignore_owner to ignore this error.
8	R/W		Txd_err: TX data CRC error, For multiple block write, any one of blocks CRC error.
7:0	R/W		Rxd_err: RX data CRC error per wire, for multiple block read, the CRC errors are Ored together.

SD_EMMC_IRQ_EN 0x4c

Bit(s)	R/W	Default	Description
31:18			unused
17			Cfg_cmd_setup, 1: improve CMD setup time by half SD_CLK cycle. 0: CMD is two cycle aligned with DATA.
16	R/W	0	Cfg_secure: Data read/write with crypto DES
15	R/W	0	en_IRQ_sdio: Enable sdio interrupt.
14	R/W	0	En_resp_status: Response status error.
13	R/W	0	en_End_of_Chain: End of Chain IRQ
12	R/W	0	en_Desc_timeout: Descriptor execution time over time limit.
11	R/W	0	en_Resp_timeout: No response received before time limit.

10	R/W	0	en_Resp_err: Response CRC error
9	R/W	0	en_Desc_err: SD/eMMC controller doesn't own descriptor.
8	R/W	0	En_txd_err: TX data CRC error
7:0	R/W	0	en_Rxd_err: RX data CRC error per wire.

Descriptor_REG0 0x50

Bit(s)	R/W	Default	Description
31:0	R/W		SD_EMMC_CMD_CFG APB read wait Same as descriptor first word, resp_num = 1, response saved back into descriptor only. Read from this APB will hold APB bus

Descriptor_REG1 0x54

Bit(s)	R/W	Default	Description
31:0	R/W		SD_EMMC_CMD_ARG APB write start Same as descriptor second word. Write to this APB address starts execution. If the current desc is busy, it will be executed after current descriptor is done.

Descriptor_REG2 0x58

Bit(s)	R/W	Default	Description
31:0	R/W		SD_EMMC_CMD_DAT : Same as descriptor third word, 32 Bits data.

Descriptor_REG3 0x5c

Bit(s)	R/W	Default	Description
31:0	R/W		SD_EMMC_CMD_RSP: Write: response status IRQ enable Bits. Read: Response Bit 31:0

Descriptor_REG4 0x60

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_CMD_RSP1: Response bit 63:32

Descriptor_REG5 0x64

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_CMD_RSP2: Response bit 95:64

Descriptor_REG6 0x68

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_CMD_RSP3: Response bit 127:96

Descriptor_REG7 0x6c

Bit(s)	R/W	Default	Description
31:0			Reserved

Current_Next_Descriptor_REG0 0x70

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_CURR_CFG: Current descriptor under execution.

Current_Next_Descriptor_REG1 0x74

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_CURR_ARG

Current_Next_Descriptor_REG2 0x78

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_CURR_DAT

Current_Next_Descriptor_REG3 0x7c

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_CURR_RSP

Current_Next_Descriptor_REG4 0x80

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_NEXT_CFG: Next descriptor waiting for execution, already read out from SRAM or DDR, can’t be changed.

Current_Next_Descriptor_REG5 0x84

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_NEXT_ARG

Current_Next_Descriptor_REG6 0x88

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_NEXT_DAT

Current_Next_Descriptor_REG7 0x8c

Bit(s)	R/W	Default	Description
31:0	R		SD_EMMC_NEXT_RSP

SD_EMMC_RXD 0x90

Bit(s)	R/W	Default	Description
31:25	R		Unused
24:16	R		Data_blk: Rxd Blocks received from BUS Txd blocks received from DDR.
15:10			unused
9:0	R		Data_cnt: Rxd words received from BUS. Txd words received from DDR.

SD_EMMC_TXD 0x94

Bit(s)	R/W	Default	Description
31:25			Unused
24:16	R		Txd_blk: Txd BUS block counter
15			unused
14:0	R		Txd_cnt: Txd BUS cycle counter

13.4 Serial Peripheral Interface Communication Controller

13.4.1 Overview

SPI Communication Controller is designed for connecting general SPI protocol compatible module. This controller allows rapid data communication with less software interrupts than conventional serial communications.

13.4.2 Features

- Full-duplex synchronous serial interface
- Master/Slave configurable
- Four chip selects to support multiple peripherals
- Transfer continuation function allows unlimited length data transfers
- 64-bit wide by 16-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SS) and SPI Clock (SCLK) are configurable
- Both PIO(Programming In/Out interface) and DMA(Direct Memory Access interface) supported

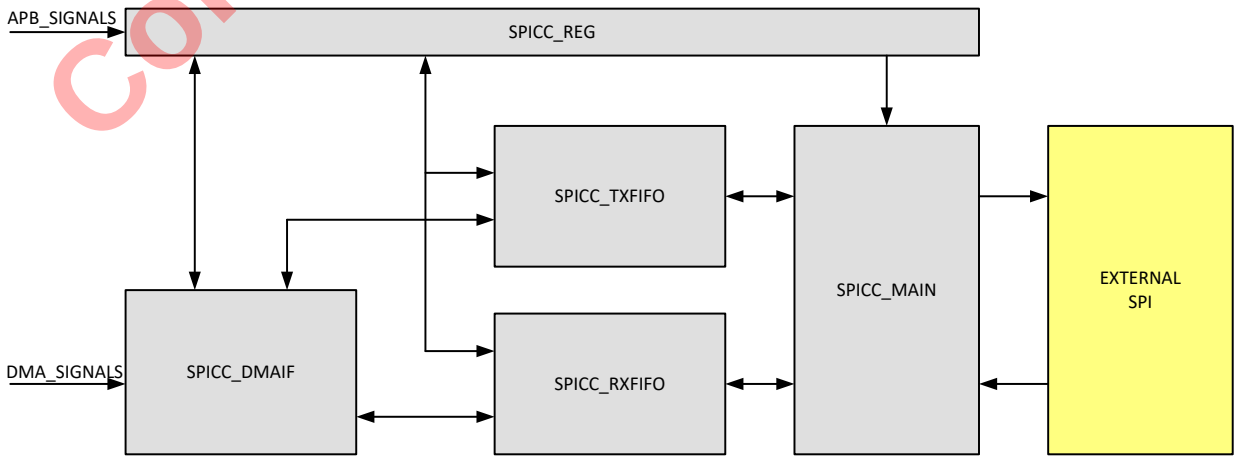
13.4.3 FunctionalDescription

The following are two SPICC modes of operation:

- Master Mode—When the SPICC module is configured as a master, it uses a serial link to transfer data between the SPICC and an external device. A chip-enable signal and a clock signal are used to transfer data between these two devices. If the external device is a transmit-only device, the SPICC master's output port can be ignored and used for other purposes. To use the internal TXFIFO and RXFIFO, two auxiliary output signals, SS and SPI_RDY, are used for data transfer rate control. The user can also program the sample period control register to a fixed data transfer rate.
- Slave Mode–When the SPICC module is configured as a slave, the user can configure the SPICC Control register to match the external SPI master's timing. In this configuration, SS becomes an input signal, and is used to control data transfers through the Shift register, as well as to load/store the data FIFO.

There are 5 sub-modules in spi communication controller, i.e. spicc_reg, spicc_dmaif, spicc_txfifo, spicc_rxfifo, and spicc_main. Transmitting and receiving are using different channel, that means they have different buffer.

Figure 13-3 SPICC Block Diagram



- spicc_reg is driven by host cpu, and spicc_reg is responsible for configuring other modules.

- spicc_dmaif is responsible for dealing with DMA operatoins.
- spicc_txfifo contains a transmission FIFO.
- spicc_rxfifo contains a receiving FIFO.
- spicc_main is responsible for main control of basic spi operation.

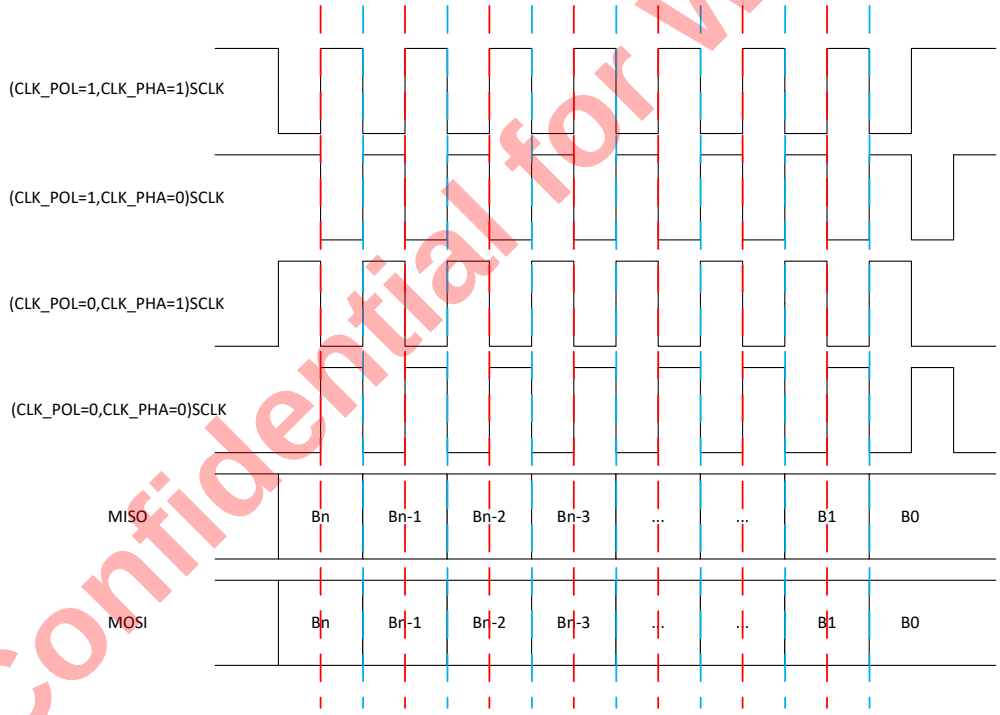
Here are SPI External Signals:

Table 13-7 SPI External Signals

Signal Name	I/O	Description
spicc_sclk	IO	SCLK, SPI Clock
spicc_miso	IO	MISO, Master Input Slave Output
spicc_mosi	IO	MOSI, Master Output Slave In
spicc_ss[3:0]	IO	SS, SPI chip Select, Supports up to 4 slaves.

And here is SPI Generic Timing:

Figure 13-4 SPI Generic Timing



13.4.4 Register Description

RXDATA 0xffd13000

Bit(s)	R/W	Default	Description
31:0	R	0	Rx Data

Note1: when PIO mode, programmer can get data from this register.

TXDATA 0xffd13004

Bit(s)	R/W	Default	Description
31:0	W	0	Tx Data

Note1: when PIO mode, programmer need send data to this register.

CONREG 0xffd13008

Bit(s)	R/W	Default	Description
31:19	RW	0	[13]burst_length ([5:0]bit number of one word/package, [12:6]burst length-1)
18:16	RW	0	[3]data_rate (sclk will be divided by system clock with equation: 2^(data_rate+2), Example: if system clock = 128MHz and data_rate=2, sclk's frequency equals 8MHz)
15:14			Reserved
13:12	RW	0	[2]chip_select (00:select ss_0, 01:select ss_1, 10:select ss_2, 11:select ss_3,)
11:10			Reserved
9:8	RW	0	[2]drctl (0:ignore RDY input, 1:Data ready using pin rdy_i's failing edge, 2:Data ready using pin rdy_i's low level, 3:reserved)
7	RW	0	[1]sspol (0:SS polarity Low active,1:High active)
6	RW	0	[1]ssctl (see details in Note1)
5	RW	0	[1]pha (clock/data phase control, see section 2.2)
4	RW	0	[1]pol (clock polarity control, see section 2.2)
3	RW	0	[1]smc (start mode control, see Note2)
2	RW	0	[1]xch(exchange bit, ATTN:will automatically cleared when burst finished, see Note3)

Bit(s)	R/W	Default	Description
1	RW	0	[1]mode (0:slave,1:master)
0	RW	0	[1]en (0:spicc disable,1:enable)

Note1: In one burst of master mode, if ssctl ==1, ss will output 1 between each spi transition. And if ssctl ==0, ss will output 0.

Note2: smc is for start mode control. If smc ==0, burst will start when xch is set to 1'b1; if smc==1, burst will start when txfifo is not empty.

Note3: setting xch will issue a burst when smc==0, and This bit will be self-cleared after burst is finished.

INTREG 0xffd1300c

Bit(s)	R/W	Default	Description
31:8			Reserved
7	RW	0	[1]tcen(transfer completed interrrupt enable)
6	RW	0	Reserved
5	RW	0	[1]rfen(rxfifo full interrrupt enable)
4	RW	0	Reserved
3	RW	0	[1]rren(rxfifo ready interrrupt enable)
2	RW	0	[1]tfen(txfifo full interrrupt enable)
1	RW	0	Reserved
0	RW	0	[1]teen(txfifo empty interrrupt enable)

Note1: Interrupt Status presents in STATREG.

DMAREG 0xffd13010

Bit(s)	R/W	Default	Description
31:26	RW	0	[6]DMA Burst Number
25:20	RW	0	[6]DMA Thread ID
19	RW	0	[1]DMA Urgent
18:15	RW	0x7	[4]Number in one Write request burst(0:1,1:2...)
14:11	RW	0x7	[4]Number in one Read request burst(0:1,1:2...)
10:6	RW	0x8	[5]RxFIFO threshold(RxFIFO's count>=thres, will request write)
5:1	RW	0	[5]TxFIFO threshold(TxFIFO's count<=thres, will request read)

0	RW	0	[1]DMA Enable

STATREG 0xffd13014

Bits	R/W	Defaults	Description
31:8			Reserved
7	RW	0	[1]tc(transfer completed, w1c, see Note1)
6	R	0	Reserved
5	R	0	[1]rf(rxfifo full)
4	R	0	Reserved
3	R	0	[1]rr(rxfifo ready)
2	R	0	[1]tf(txfifo full)
1	R	0	Reserved
0	R	0	[1]te(txfifo empty)

Note1: tc is the status bit which indicates a burst transfer is completed. And a burst transfer should be started by writing xch 1'b1. This bit supports w1c(Write 1 clear).

PERIODREG 0xffd13018

Bit(s)	R/W	Default	Description
31:15			Reserved
14:0	RW	0	[15]period(wait cycles, see Note1)

Note1: Programmer can add wait cycles through this register if transmission rate need to be controlled.

TESTREG 0xffd1301c

Bit(s)	R/W	Default	Description
31:23	RW	0	Reserved
23:22	RW	0	[2]fiforst(fifo soft reset)
21:16	RW	0x15	[6]dlyctl(delay control)
15	RW	0	[1]swap(data swap for reading rxfifo)
14	RW	0	[1]lbc(loop back control)
12:10	R	0	[3]smstatus(internal state machine status)
9:5	R	0	[5]rxcnt(internal RxFIFO counter)
4:0	R	0	[5]txcnt(internal TxFIFO counter)

Note1: Programmer can only use the TESTREG[9:0], rxcnt(internal RxFIFO counter) and txcnt(internal TxFIFO counter) , and other Bits just for test.

DRADDR 0xffd13020

Bit(s)	R/W	Default	Description
31:0	RW	0	Read Address of DMA

DWADDR 0xffd13024

Bit(s)	R/W	Default	Description
31:0	RW	0	Write Address of DMA

LD_CNTL0 (0xffd13028)

Bits	R/W	Defaults	Description
31:9	RW	0	Reserved
8	RW	0	dma raddr/waddr load by dma_enable signal
7	RW	0	dma waddr load by vsync irq
6	RW	0	dma raddr load by vsync irq
5	RW	0	dma write counter enable
4	RW	0	dma read counter enable
3	RW	0	xch enable set by vsync irq
2	RW	0	dma enable set by vsync irq
1	RW	0	Reserved
0	RW	0	Vsync irq source select

LD_CNTL1 0xffd1302c

Bits	R/W	Defaults	Description
31:16	RW	0	dma write counter
15:0	RW	0	dma read counter

LD_RADDR 0xffd13030

Bits	R/W	Defaults	Description
31:0	RW	0	shadow dma read address for load

LD_WADDR 0xffd13034

Bits	R/W	Defaults	Description
31:0	RW	0	shadow dma write address for load

ENHANCE_CNTL 0xffd13038

Bits	R/W	Defaults	Description
31:30	R	0	Reserved
29	R/W	0	main clock always on
28	R/W	0	clk-cs delay enable
27	R/W	0	cs_oen enhance enable
26	R/W	0	clk_oen enhance enable
25	R/W	0	mosi_oen enhance enable
24	R/W	0	spi clk select 0: controlled by data_rate in CONREG 1: controlled by enhance_clk_div in ENHANCE_CNTL
23:16	R/W	0	enhance_clk_div
15:0	R/W	0	clk-cs delay value

ENHANCE_CNTL1 (0xffd1303c)

Bits	R/W	Defaults	Description
31:29	R/W	0	enhance_fclk_mosi_oen_dlyctl: mosi_oen delay control in fclk
28	R/W	0	enhance_fclk_mosi_oen_dlyctl_en: enable dlyctl
27:25	R/W	0	enhance_fclk_mosi_o_dlyctl: mosi_o delay control in fclk
24	R/W	0	enhance_fclk_mosi_o_dlyctl_en: enable dlyctl
23:21	R/W	0	enhance_fclk_miso_i_dlyctl: miso_i delay control in fclk
20	R/W	0	enhance_fclk_miso_i_dlyctl_en: enable dlyctl
19:17	R/W	0	enhance_fclk_mosi_i_dlyctl: mosi_i delay control in fclk
16	R/W	0	enhance_fclk_mosi_i_dlyctl_en: enable dlyctl
15	R/W	0	enhance_fclk_en: fclk gate enable
14	R/W	0	enhance_mosi_i_capture_en: enable, 1=select enhance capture function for mosi_i (slave mode)
9:1	R/W	0	enhance_clk_tcnt: adjust capturing timing for miso_i data (master mode). when clk_cnt=enhance_clk_tcnt, capture input data. the value of enhance_clk_tcnt must be less than the most value of clk_cnt. sclk will be divided by system clock with clk_cnt.
0	R/W	0	enhance_miso_i_capture_en: enable, 1=select enhance capture function for miso_i (master mode)

ENHANCE_CNTL2 0xffd13040

Bits	R/W	Defaults	Description

31	R/W	0	clk_cs_tt delay enable
30:16	R/W	0	clk_cs_tt delay value
15	R/W	0	clk_cs_ti delay enable
14:0	R/W	0	clk_cs_ti delay value

SPICC1: 0xffd15000~0xffd15038

See the registers for SPICC0

.

13.5 Serial Peripheral Interface Flash Controller

13.5.1 Overview

SPI Flash Controller is designed for connecting varied SPI Flash memory.

13.5.2 Features

- Support three operation modes, NOR Flash mode, Master mode, and Slave mode.
- Support read/write buffer up to 64bytes.
- Support no clock toggling during DUMMY state.
- Support hold by an external pin during a transition.
- AHB read support byte and halfword.
- Support bit-number rather than byte-number for each stage.
- Support 2/4 wire writing like fast reading
- Support both rising-edge and falling-edge for SPI slave sampling and SPI master sampling.
- Support 1 wire for SPI_D and SPI_Q.
- Support SPI_CK setup and hold time by cycles
- Support 8 bit clock divider, so SPI_CK can be low as 1/256 HCLK
- Support byte-order in a word
- Support no command state, so the command is sent/received in address state by 2/4 wires.
- Support both data input and data output in a transition. SPI_DOUT->(SPI_DUMMY)->SPI_DIN

13.5.3 Register Description

Note that: If the bit2 “Backward Compatible” in “SPI User Register” is 1, the registers and functions of this SPI controller are as same as Apollo SPI controller.

To use SPI Flash commands, please set the bit2 “Backward Compatible” in “SPI User Register” as “1”.

SPI FLASH Command register 0xffd14000

Bit(s)	R/W	Default	Description
31	R/W	0	READ command. 1 = read. When it becames 0, the read command is finished. The READ command could be (0xEB, 0x6B, 0xBB, 0x3B, 0x0B, 0x03). By default is 0x0B. One read command will read 968erilog968968968 32x8bits data. And saved in data cache.
30	R/W	0	WREN command. (0x06)
29	R/W	0	WRDI command. (0x04).
28	R/W	0	RDID command. (0x9f).
27	R/W	0	RDSR command. (0x05).
26	R/W	0	WRSR command. (0x01).
25	R/W	0	Page program command. (0xAD or 0x02).
24	R/W	0	SE command (0x20).
23	R/W	0	BE command (0xD8).

Bit(s)	R/W	Default	Description
22	R/W	0	CE command.(0xC7).
21	R/W	0	Deep Power Down command(0xB9).
20	R/W	0	RES command. (0xAB).
19	R/W	0	HPM command.(0xA3). (Just For winbond SPI flash).
18	R/W	0	USER defined command.
17:0	R/W	0	Reserved for future.

If the bit2 “Backward Compatible” in “SPI User Register” is 1, the bit15:0 of this registers are defined as same as Apollo SPI controller.

Bit(s)	R/W	Default	Description
15	R/W	0	USER command address bit. 1 = user command includes address. 0 = no address.
14	R/W	0	USER command dummy bit. 1= user command includes Dummy bytes.
13	R/W	0	USER command DIN bit. 1 = use command includes data in. 0 = no data in.
12	R/W	0	USER command DO bit. 1 -= use command includes data output. 0 = no data output. Only DIN or DO support for one User command. Not both.
11:10	R/W	0	User command dummy byte number. How many dummy bytes for user command.
9:8	R/W	0	Reserved for future.
7:0	R/W	0	command value for User command bit[7:0].

SPI address register 0xffd14004

Bit(s)	R/W	Default	Description
31:0	R/W	0	The address[31:0] of the user command

If the bit2 “Backward Compatible” in “SPI User Register” is 1, the bit15:0 of this registers are defined as same as Apollo SPI controller.

Bit(s)	R/W	Default	Description
31:30	R/W	0	Reserved.
29:24	R/W	0	DIN/DO data bytes number.
23:0	R/W	0	24 bits address.

SPI control register 0xffd14008

Bit(s)	R/W	Default	Description
31:27	R/W	0	Reserved.

Bit(s)	R/W	Default	Description
26	R/W	0	Write bit order. 1 = 0, 1, 2, 3, 4, 5, 6, 7. 0 = 7, 6, 5, 4, 3, 2, 1, 0.
25	R/W	0	Read bit order. . 1 = 0, 1, 2, 3, 4, 5, 6, 7. 0 = 7, 6, 5, 4, 3, 2, 1, 0.
24	R/W	0	Fast read QIO mode.
23	R/W	0	Fast read DIO mode.
22	R/W	0	Write 2 bytes status mode. For some of winbond SPI flash, the status register is 16bits.
21	R/W	1	SPI flash WP pin value if use SPI flash WP pin as write protection.
20	R/W	0	Fast read QOUT mode.
19	R/W	1	1 = SPI share pins with SDRAM. 0 = doesn't share.
18	R/W	0	SPI hold mode. 1=SPI controller would use SPI hold function. 0 = SPI controller won't use hold function. The SPI flash hold pin can be tie high on the board. Or SPI controller can use hold pin as QIO/QOUT mode.
17	R/W	1	1 = enable AHB request. 0 = disable AHB request when you reconfigure SPI controller or running APB bus commands.
16	R/W	0	1 =enable SST SPI Flash aai command. The APB bus PP command will send AAI command.
15	R/W	1	1 = release from Deep Power-Down command is with read electronic signature.
14	R/W	0	Fast read DOUT mode.
13	R/W	1	Fast read mode. AHB bus read requirement and APB bus read command use the command 0x0Bh.
12:0	R/W	0	Reserved for future.

If the bit2 "Backward Compatible" in "SPI User Register" is 1, the bit12:0 of this registers are defined as same as Apollo SPI controller.

Bit(s)	R/W	Default	Description
12	R/W	1	1=SPI clock frequency is same as system clock. 0 = SPI clock frequency will use clock divider.
11:8	R/W	0	Clock counter for clock divider.
7:4	R/W	0	Clock high counter.
3:0	R/W	0	Clock low counter. If the SPI clock frequcncy = sys_clock_frequency / n. Then the clock divider counter = n – 1; the clock high counter = n /2 – 1; the clock low counter = n– 1; For example, if you want to SPI clock 970erilog970970 is divided by 2 of the system clock. The clock divider counter = 1, clock high counter = 0, clock low counter = 1. For SPI clock frequency = system clock / 4.

Bit(s)	R/W	Default	Description
			The clock divider counter = 3, clock high counter = 1, clock low counter = 3.

SPI control register 1 0xffd1400c

Bit(s)	R/W	Default	Description
31:28	R/W	5	SPI Clock cycles for SPI flash timing requirement tCSH.
27:16	R/W	0xfff	SPI Clock cycles for SPI flash timing requirement tRES.
15:0	R/W	0x0120	System clock cycles for SPI bus timer. In SPI share bus and SPI hold function mode. SPI bus timer used , if SPI use the bus for a limit time, SPI controller will diassert SPI hold pin to halt the SPI Flash, and give the bus control to SDRAM.

SPI status register 0xffd14010

Bit(s)	R/W	Default	Description
31:24	R/W	0	Reserved.
23:16	R/W	0	For winbond SPI flash, this 8 bits used for DIOmode M7~M0,
15:0	R/W	0	SPI status register value. WRSR command will write this value to SPI flash status. RDSR or RES command will save the read result to this register.

When SPI controller in the slave mode, this register are the status for the SPI master to read out.

Bit(s)	R/W	Default	Description
31:0	R/W	0	In SPI Slave mode, the read status of the user command

SPI control register 2 0xffd14014

Bit(s)	R/W	Default	Description
31:28	R/W	0	Delay cycle number of SPI_CS input in SPI slave mode or SPI_CS output in SPI master mode 0= not delay, 1 = delayed by 1 cycle of system clock, 2 = delayed by 2 cycles of system clock , …
27:26	R/W	0	delay mode of SPI_CS input in SPI slave mode or SPI_CS output in SPI master mode 0= not latched by the edges of SPI_CK 1= latched by the falling edges of SPI_CK 2 = latched by the rising edges of SPI_CK
25:23	R/W	0	Delay cycle number of SPI Data from SPI Master to SPI Slave In SPI master mode, it is for data outputs; in SPI slave mode, it is for data inputs. 0= not delay, 1 = delayed by 1 cycle of system clock, 2 = delayed by 2 cycles of system clock , …
22:21	R/W	0	Delay mode of SPI Data from SPI Master to SPI Slave In SPI master mode, it is for data outputs; in SPI slave mode, it is for data inputs. 0= not latched by the edges of SPI_CK

Bit(s)	R/W	Default	Description
			1= latched by the falling edges of SPI_CK 2 = latched by the rising edges of SPI_CK
20:18	R/W	0	Delay cycle number of SPI Data from SPI Slave to SPI Master. In SPI master mode, it is for data inputs; in SPI slave mode, it is for data outputs. 0= not delay, 1 = delayed by 1 cycle of system clock, 2 = delayed by 2 cycles of system clock , …
17:16	R/W	0	Delay mode of SPI Data from SPI Slave to SPI Master. In SPI master mode, it is for data inputs; in SPI slave mode, it is for data outputs. 0= not latched by the edges of SPI_CK 1= latched by the falling edges of SPI_CK 2 = latched by the rising edges of SPI_CK
15:12	R/W	0	In SPI master mode, SPI_CK rising edge mode 4'b1000 = later by 1/4 cycle of SPI_CK 4'b1001 = later by 1/8 cycle of SPI_CK 4'b1010 = later by 1/16 cycle of SPI_CK 4'b1011 = later by 1/32 cycle of SPI_CK 4'b1100 = earlier by 1/4 cycle of SPI_CK 4'b1101 = earlier by 1/8 cycle of SPI_CK 4'b1110 = earlier by 1/16 cycle of SPI_CK 4'b1111 = earlier by 1/32 cycle of SPI_CK Others = Normal
11:8	R/W	0	In SPI master mode, SPI_CK falling edge mode 4'b1000 = later by 1/4 cycle of SPI_CK 4'b1001 = later by 1/8 cycle of SPI_CK 4'b1010 = later by 1/16 cycle of SPI_CK 4'b1011 = later by 1/32 cycle of SPI_CK 4'b1100 = earlier by 1/4 cycle of SPI_CK 4'b1101 = earlier by 1/8 cycle of SPI_CK 4'b1110 = earlier by 1/16 cycle of SPI_CK 4'b1111 = earlier by 1/32 cycle of SPI_CK Others = Normal
7:4	R/W	1	In master mode, SPI clock cycles for SPI hold timing.
3:0	R/W	1	In master mode, SPI clock sycles for SPI setup timing. SPI setup time and SPI hold time is used to configure how soon the controller can enable spi_cs_n after the controller get the bus and how long the controller still keep the bus after the spi_cs_n become to be high.

SPI Clock register 0xffd14018

Bit(s)	R/W	Default	Description

31	R/W	1	1=SPI clock frequency is same as system clock. 0 = SPI clock frequency will use clock divider.
30:18	R/W	0	Clock counter for Pre-scale divider: 0= not pre-scale divider, 1= pre-scale divided by 2, 2= pre-scale divided by 3, ….,
17:12	R/W	0	Clock counter for clock divider.
11:6	R/W	0	Clock high counter in SPI master mode. In SPI slave mode, it is for the delay counter for the rising edges of spi_ck_i
5:0	R/W	0	Clock low counter, in SPI master mode. In SPI slave mode, it is for the delay counter for the falling edges of spi_ck_i If the SPI clock frequcncy = sys_clock_frequency / n. Then the clock divider counter = n – 1; the clock high counter = n /2 – 1; the clock low counter = n– 1; For example, if you want to SPI clock 973erilog973973 is divided by 2 of the system clock. The clock divider counter = 1, clock high counter = 0, clock low counter = 1. For SPI clock frequency = system clock / 4. The clock divider counter = 3, clock high counter = 1, clock low counter = 3.

SPI User register 0xffd1401c

Bit(s)	R/W	Default	Description
31	R/W	1	USER command COMMAND bit. 1 = user command includes command. 0 = no command. If some SPI slaves may support 2/4 IO at the first cycle, clear this bit.
30	R/W	0	USER command ADDRESS bit. 1 = user command includes address. 0 = no address.
29	R/W	0	USER command DUMMY bit. 1= user command includes Dummy bytes.
28	R/W	0	USER command DIN bit. 1 = user command includes data in. 0 = no data in.
27	R/W	0	USER command DO bit. 1 = user command includes data output. 0 = no data output. If both DIN and DO are valid, SPI master is firstly in data output state and then in data input state. If all of DUMMY, DO and DIN are valid, SPI master is firstly in data output state and then in dummy state, finally in data input state.
26	R/W	0	USER command dummy idle bit. 1= no SPI clock toggling in dummy state. 0= normal
25	R/W	0	USER command highpart bit for SPI_DOUT stage. It is for data-output in spi master mode and for data-input in spi slave mode. 1 = only high half part of buffer are used. 0 = low half part or the whole 64bytes are used.
24	R/W	0	USER command highpart bit for SPI_DIN stage. It is for data-input in spi master mode and for data-output in spi slave mode. 1 = only high half part of buffer are used. 0 = low half part or the whole 64bytes are used.

Bit(s)	R/W	Default	Description
23	R/W	0	User command external hold bit for prep. 1 = in prep state, SPI master controller can be hold by the external pin SPI_HOLD
22	R/W	0	User command external hold bit for command. 1 = in command state, SPI master controller can be hold by the external pin SPI_HOLD
21	R/W	0	User command external hold bit for address. 1 = in address state, SPI master controller can be hold by the external pin SPI_HOLD
20	R/W	0	User command external hold bit for dummy. 1 = in dummy state, SPI master controller can be hold by the external pin SPI_HOLD
19	R/W	0	User command external hold bit for data input. 1 = in data input state, SPI master controller can be hold by the external pin SPI_HOLD
18	R/W	0	User command external hold bit for data output. 1 = in data output state, SPI master controller can be hold by the external pin SPI_HOLD
17	R/W	1	User command external hold polarity bit. 1 = high is valid for hold, 0 = low is valid for hold.
16	R/W	0	Single DIO mode: Data output and input apply only 1 wire.
15	R/W	0	Fast write QIO mode.
14	R/W	0	Fast write DIO mode.
13	R/W	0	Fast write QOUT mode.
12	R/W	0	Fast write DOUT mode.
11	R/W	0	Write byte order. 0 = d[7:0], d[15:8], d[23:16], d[31:24]. 1 = d[31:24], d[23:16], d[15:8], d[7:0]
10	R/W	0	Read byte order. . 0 = d[7:0], d[15:8], d[23:16], d[31:24]. 1 = d[31:24], d[23:16], d[15:8], d[7:0]
9:8	R/W	0	AHB endian mode: 0= little-endian; 1= big-endian; 2~3 reserved
7	R/W	0	In SPI master mode, the clock output edge bit: 0 = SPI_CK is inverted, 1 = SPI_CK is not inverted
6	R/W	1	In SPI slave mode, the clock input edge bit: 0 = SPI_CK_I is inverted, 1 = SPI_CK_I is not inverted
5	R/W	0	SPI CS setup bit: 1 = valid in prep state
4	R/W	0	SPI CS hold bit: 1 = valid in done state
3	R/W	0	AHB-read apply the configurations of user-command, such as command value, bit-length,…
2	R/W	1	Backward Compatible: 1 = compatible to Apollo SPI This bit affect the three registers: “SPI Flash Commmand Register”, “SPI Address Register” and “SPI Control Register”

Bit(s)	R/W	Default	Description
1	R/W	0	AHB-read support 4byte address, when AHB-read apply the configurations of user-command. 1 = 4byte address, 0 = 3byte address
0	R/W	0	In SPI master mode, Enable bit for Data input during SPI_DOUT stage. 1 = enable; 0 = disable This bit shall not be used in 2/4wire or SIO. When this bit is 1, during SPI_DOUT stage, data input are stored into cacheline/buffer from address 0, i.e., Bit24 is not controlling the start address. The data output can be specified by bit25

SPI User register 1 0xffd14020

Bit(s)	R/W	Default	Description
31:26	R/W	0	USER command bit number for address state 0 = 1 bit, 1= 2 bits, …
25:17	R/W	0	USER command bit number for data output state 0 = 1 bit, 1= 2 bits, …
16:8	R/W	0	USER command bit number for data input state 0 = 1 bit, 1= 2 bits, …
7:0	R/W	0	USER command cycle number for dummy state 0 = 1 cycle, 1= 2 cycles, …

SPI User register 2 0xffd14024

Bit(s)	R/W	Default	Description
31:28	R/W	0	USER command bit number for command state 0 = 1 bit, 1= 2 bits, …
27:16	R/W	0	Reserved
15:0	R/W	0	The command content of the user command

SPI User register 3 0xffd14028

Bit(s)	R/W	Default	Description
31:0	R/W	0	In SPI Master mode, the address[63:32] of the user command In SPI Slave mode, the write status of the user command

SPI PIN register 0xffd1402c

Bit(s)	R/W	Default	Description
31	R/W	0	Pin swap when it is in DIN stage and SPI data input are 4wire. This feature is for Ubec Zigbee chips. 1 = swap between {spi_q, spi_d_i} and {spi_hold_i, spi_wp_i} 0 = normal
30	R/W	0	In SPI Master mode, CS keep active after a transition. 1 = enable; 0 = disable
29	R/W	0	Idle edge of SPI_CK 0 = low when it is idle 1 = high when it is idle
28:24	R/W	0	Reserved
23	R/W	0	In the SPI slave mode, spi_cs_i polarity: 1= high voltage is active 0= low voltage is active
22:21	R/W	0	In the SPI slave mode, spi_ck_i and spi_cs_i source pins 0=SPI_CK and SPI_CS pins, respectively 1=SPI_CS2 and SPI_CS1 pins, respectively 2=SPI_HOLD and SPI_WP pins, respectively
20	R/W	0	SPI_CS2 and SPI_CS1 pin function MUX 0= spi_ck and spi_cs in the SPI master mode 1= input pads for spi_ck_i and spi_cs_i, respectively, in the SPI slave mode
19	R/W	0	SPI_CK and SPI_CS pin function MUX 0= spi_ck and spi_cs in the SPI master mode 1= input pads for spi_ck_i and spi_cs_i, respectively, in the SPI slave mode
18:17	R/W	0	SPI_HOLD and SPI_WP pin function MUX 0= normal 1= spi_q and spi_d, respectively 2= spi_cs3 and spi_cs2, repectively 3= input pads for spi_ck_i and spi_cs_i, respectively, in the SPI slave mode
16	R/W	0	SPI_D and SPI_Q switch 1= SPI_D and SPI_Q pin-functions are swapped 0= normal
15:11	R/W	0	In Master mode, these are spi_ck MUX bit[4:0] for spi_cs4 (SPI_HOLD pin), spi_cs3 (SPI_WP pin), spi_cs2, spi_cs1 and spi_cs, respectively 1= this pin is spi_ck, if this pin is not idle 0= this pin is spi_cs, if this pin is not idle

Bit(s)	R/W	Default	Description
10:6	R/W	0	In Master mode, these are polarity bit[4:0] for spi_cs4 (SPI_HOLD pin), spi_cs3 (SPI_WP pin), spi_cs2, spi_cs1 and spi_cs, respectively 1= high voltage is active 0= low voltage is active
5:0	R/W	0x1E	In Master mode, these are idle bit[5:0] for SPI_CK, for spi_cs4 (SPI_HOLD pin), spi_cs3 (SPI_WP pin), spi_cs2, spi_cs1 and spi_cs, respectively 1= idle, i.e., the spi_ck signal is 0 or the spi_cs is at the inactive level 0= active if SPI controller is working

SPI Slave register 0xffd14030

Bit(s)	R/W	Default	Description
31	R/W	0	SPI controller SW reset: 1 = reset, 0 = none
30	R/W	0	SPI slave mode: 1 = slave, 0 = master
29	R/W	0	In SPI slave mode, the enable bit for the command of write-buffer-and-read-buffer 0= disable, 1=enable
28	R/W	0	In SPI slave mode, the enable bit for the command of write-status-and-read-status 0= disable, 1=enable
27	R/W	0	SPI slave command define enable 0=Apply the last 3bits of Flash commands and two extra command. 1=Apply the user defined command in SPI Slave register 3
26:4	R/W	0	Reserved
11:10	R/W	0	spi_cs_i recovery mode: 0= and 1= or 2= normal 3= delayed
9:5	R/W	0	Interrupt enable for bit4:0 1= enable, 0=disable
4	R/W	0	A SPI transition is done. (whatever it is in SPI master mode or SPI slave mode)
3	R/W	0	In SPI slave mode, a status write is done
2	R/W	0	In SPI slave mode, a status read is done
1	R/W	0	In SPI slave mode, a buffer write is done
0	R/W	0	In SPI slave mode, a buffer read is done

SPI Slave register 1 0xffd14034

Bit(s)	R/W	Default	Description
31:27	R/W	0	In SPI slave mode, status bit number 0 = 1 bit, 1= 2 bits, …
26	R/W	0	In SPI slave mode, status fast read/write enable bit: 1 = enable, 0 = disable
25	R/W	0	In SPI slave mode, status read back enable bit: 1 = reading status is written status in SPI User register 3, 0 = reading status is in SPI Status register.
24:16	R/W	0	In SPI slave mode, buffer bit number 0 = 1 bit, 1= 2 bits, …
15:10	R/W	0	In SPI slave mode, address bit number for reading buffer 0 = 1 bit, 1= 2 bits, …
9:4	R/W	0	In SPI slave mode, address bit number for writing buffer 0 = 1 bit, 1= 2 bits, …
3	R/W	0	In SPI slave mode, dummy enable bit for writing status 1=enable, 0=disable
2	R/W	0	In SPI slave mode, Dummy enable bit for reading status 1=enable, 0=disable
1	R/W	0	In SPI slave mode, Dummy enable bit for writing buffer 1=enable, 0=disable
0	R/W	0	In SPI slave mode, Dummy enable bit for reading buffer 1=enable, 0=disable

SPI Slave register 2 0xffd14038

Bit(s)	R/W	Default	Description
31:24	R/W	0	In SPI slave mode, Dummy cycle number for writing buffer 0 = 1 cycle, 1= 2 cycles, …
23:16	R/W	0	In SPI slave mode, Dummy cycle number for reading buffer 0 = 1 cycle, 1= 2 cycles, …
15:8	R/W	0	In SPI slave mode, Dummy cycle number for writing status 0 = 1 cycle, 1= 2 cycles, …
7:0	R/W	0	In SPI slave mode, Dummy cycle number for reading status 0 = 1 cycle, 1= 2 cycles, …

SPI Slave register 3 0xffd1403C

Bit(s)	R/W	Default	Description
31:24	R/W	0	In SPI slave mode, Command value for writing status, when bit27 “SPI slave command define enable” in SPI Slave register is 1

23:16	R/W	0	In SPI slave mode, Command value for reading status, when bit27 “SPI slave command define enable” in SPI Slave register is 1
15:8	R/W	0	In SPI slave mode, Command value for writing buffer, when bit27 “SPI slave command define enable” in SPI Slave register is 1
7:0	R/W	0	In SPI slave mode, Command value for reading buffer, when bit27 “SPI slave command define enable” in SPI Slave register is 1

SPI controller cache 0~7 0xffd14040~0xffd1405c

Bit(s)	R/W	Default	Description
31:0	R/W	0	Cache line Word 0~7. Cache is used to read data both for AHB or APB read command. Cache is also used for APB page programming etc.

SPI controller buffer 8~15 0xffd14060~0xffd1407c

Bit(s)	R/W	Default	Description
31:0	R/W	0	Buffer Word 8. Buffer is used to read/write data only for APB read/write user commands.

14. I/O Interface

This part describes S922X's I/O interfaces from the following aspects:

- USB
- Ethernet
- SDIO
- Transport Interface and Transport Stream Demux
- IR Remote
- SAR ADC
- I2C
- UART
- PWM

14.1 Universal Serial Bus

14.1.1 Overview

The chip integrates one USB XHCI OTG 2.0 ports, one USB3.0 and PCIe 2.0 combo interface up to 5Gbps, supports 2 configurations:

- 1 USB2.0 OTG + 1 USB 2.0 Host + 1 PCIe
- 1 USB2.0 OTG + 1 USB3.0 (No PCIe)

14.1.2 Features

The USB2.0 OTG controller features:

- Support for the following speeds: High-Speed (HS, 480-Mbps),Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes
- Multiple DMA/non DMA mode access support on the application side
- Supports up to 16 bidirectional endpoints, including control endpoint 0.
- Supports Session Request Protocol (SRP) and Host Negotiation Protocol (HNP)
- Supports up to 16 host channels.
- Supports ACA ID detector, refer to Amlogic USB2.0 OTG ID Detector Specification.pdf for detail

The USB2.0 Host controller features:

- Support for the following speeds: High-Speed (HS, 480-Mbps),Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes
- Multiple DMA/non DMA mode access support on the application side
- Supports up to 16 host channels.

The USB2.0 PHY features:

- Support for the following speeds: High-Speed (HS, 480-Mbps),Full-Speed (FS, 12-Mbps) and Low-Speed (LS, 1.5-Mbps) modes
- The USB3.0 Host controller features:
- Support for the following speed: Super-Speed, High-Speed, Full-Speed and Low-Speed
- Compliant with the xHCI specification

The USB3.0 SS PHY features:

- 5-Gbps SuperSpeed data transmission rate over 3-m USB 3 cable
- Integrated PHY includes transmitter, receiver, spread spectrum clock (SSC) generation, PLL, digital core, and electrostatic discharge (ESD) protection circuits
- Supports legacy Half-rate mode for power-saving

14.1.3 Register Description

Base address:

PHY 20: 0xff63_6000.

PHY 21: 0xff63_A000.

For the following registers, each register’s final address= base address + offset *4.

reg32_00 0x00

Bit(s)	R/W	Default	Description
31	R/W	0	PLL_Fine_Tuning_2
30	R/W	1	PLL_Fine_Tuning_1
29	R/W	0	PLL_Fine_Tuning_0
28	R/W	0	PLL_Bypass_Enable
27	R/W	0	PLL_Lock_over_ride
26	R/W	0	pll_clock_divide_5
25	R/W	0	pll_clock_divide_4
24	R/W	0	pll_clock_divide_3
23	R/W	0	pll_clock_divide_2
22	R/W	0	pll_clock_divide_1
21	R/W	0	pll_clock_divide_0
20	R/W	0	refclk_multiplier_5
19	R/W	0	refclk_multiplier_4
18	R/W	1	refclk_multiplier_3
17	R/W	0	refclk_multiplier_2
16	R/W	1	refclk_multiplier_1
15	R/W	0	refclk_multiplier_0
14	R/W	1	reset_FS_LS_Clock_Divider
13	R/W	1	reset_HS_CDR
12	R/W	1	reset_FS_LS_CDR
11:10	R/W	0	reg32_00_11_10_reserved
9	R/W	0	BIAS_Power_Down
8	R/W	0	BGR_Power_Down
7	R/W	0	Calibration_Power_Down
6	R/W	0	HS_Disconnect_Power_Down
5	R/W	0	HS_Squelch_Power_Down
4	R/W	0	FS_LS_RX_Power_Down
3	R/W	0	HS_RX_Power_Down

Bit(s)	R/W	Default	Description
2	R/W	0	FS_LS_Driver_Power_Down
1	R/W	0	HS_TX_Driver_Power_Down
0	R/W	0	PLL_Power_Down

reg32_01 0x04

Bit(s)	R/W	Default	Description
31:24	R/W	0	Reserved
23:16	R/W	0	Reserved
15:10	R/W	0	Reserved
9	R/W	0	hs_en_mode
8	R/W	0	spare
7:4	R/W	0	Reserved
3:2	R/W	0	bypass_en
1:0	R/W	0x3	slew_control

reg32_02 0x08

Bit(s)	R/W	Default	Description
31	RO	0	Phy_status
30	RO	0	cal_en_flag
29:27	RO	0	reg32_02_29_27_reserved
26	RO	0	HS_Disconnect_Status
25	RO	0	HS_Squelch_Status
24	RO	0	PRBS_Sync_Out
23:16	RO	0	Calibration_code_Value_23_16
15:8	RO	0	Calibration_code_Value_15_8
7:0	RO	0	Calibration_code_Value_7_0

reg32_03 0x0c

Bit(s)	R/W	Default	Description
31:24	R/W	0	Reserved
23:16	R/W	0	Reserved

15:8	R/W	0	Reserved
7:4	R/W	0x2	Reserved
3:2	R/W	0x1	hsdic_ref
1:0	R/W	0	squelch_ref

reg32_04 0x10

Bit(s)	R/W	Default	Description
31:30	R/W	0	i_c2l_bias_trim_3_2
29:28	R/W	0	i_c2l_bias_trim_1_0
27	R/W	0	TEST_Bypass_mode_enable
26	RO	0	i_c2l_cal_done
25	R/W	0	i_c2l_cal_reset_n
24	R/W	0	i_c2l_cal_en
23:16	R/W	0	Calibration_code_Value_23_16
15:8	R/W	0	Calibration_code_Value_15_8
7:0	R/W	0	Calibration_code_Value_7_0

reg32_05 0x14

Bit(s)	R/W	Default	Description
31:24	R/W	0	i_c2l_obs_7_0
23:22	R/W	0	reg32_05_23_22_reserved
21:20	R/W	0	i_c2l_pll_perfcfg_21_20
19	R/W	0x1	i_c2l_pll_perfcfg_19
18:16	R/W	0	i_c2l_pll_perfcfg_18_16
15:12	R/W	0	i_c2l_pll_perfcfg_15_12
11:10	R/W	0	i_c2l_pll_perfcfg_11_10
9	R/W	0x1	i_c2l_pll_perfcfg_9
8	R/W	0x1	i_c2l_pll_perfcfg_8
7:4	R/W	0x7	i_c2l_pll_perfcfg_7_4
3:2	R/W	0x3	i_c2l_pll_perfcfg_3_2
1:0	R/W	0	i_c2l_pll_perfcfg_1_0

reg32_06 0x18

Bit(s)	R/W	Default	Description
31	R/W	0x1	hub_extra_bit_cntr
30:24	R/W	0	cntr_timeout
23	R/W	0	Internal_loopback
22	R/W	0	reg32_06_22_reserved
21	R/W	0	PCS_Reset_Transmit_State_machine
20	R/W	0	PCS_Reset_Receive_State_machine
19:16	R/W	0xf	fsls_farend_device_disconnect_micro_second_count_11_8
15:12	R/W	0xa	reg32_06_15_12_reserved
11:8	R/W	0	bypass_disc_cntr_3_0
7:0	R/W	0x17	PCS_microsecond_timer_done_count_value_7_0

reg32_07 0x1c

Bit(s)	R/W	Default	Description
31:24	RO	0	Prbs_Error_count
23:21	R/W	0	reg32_07_23_21_reserved
20:17	R/W	0xf	RX_ERROR_Turn_Around_Timer_Count
16	R/W	0	acceptable_bit_drops
15	R/W	0	host_tristate
14:12	R/W	0x4	fs_ls_minimum_count
11:8	R/W	0xf	cntr_done_value_7_4
7:4	R/W	0xf	cntr_done_value_3_0
3:0	R/W	0	HS_CDR_internal_tap_select

reg32_08 0x20

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_2
23:16	R/W	0	Custom_Pattern_1
15:8	R/W	0x4	Custom_Pattern_0
7	R/W	0	Enable_RX_ERROR_Timeout_Mode
6	R/W	0	reset_us_timer

5	R/W	0	PRBS_ERROR_Insert
4	R/W	0	PRBS_comparison_enable
3	R/W	0	PRBS_Enable
2:0	R/W	0	pattern

reg32_09 0x24

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_6
23:16	R/W	0	Custom_Pattern_5
15:8	R/W	0	Custom_Pattern_4
7:0	R/W	0	Custom_Pattern_3

reg32_10 0x28

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_10
23:16	R/W	0	Custom_Pattern_9
15:8	R/W	0	Custom_Pattern_8
7:0	R/W	0	Custom_Pattern_7

reg32_11 0x2c

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_14
23:16	R/W	0	Custom_Pattern_13
15:8	R/W	0	Custom_Pattern_12
7:0	R/W	0	Custom_Pattern_11

reg32_12 0x30

Bit(s)	R/W	Default	Description
31:24	R/W	0	Custom_Pattern_18
23:16	R/W	0	Custom_Pattern_17
15:8	R/W	0	Custom_Pattern_16
7:0	R/W	0	Custom_Pattern_15

reg32_13 0x34

Bit(s)	R/W	Default	Description
31	R/W	0	reg32_13_31_reserved
30	R/W	0	i_c2l_fsls_rx_en
29	R/W	0	i_c2l_hs_rx_en
28	R/W	0	i_c2l_fs_oe
27	R/W	0	i_c2l_hs_oe
26	R/W	0	i_c2l_ls_en
25	R/W	0	i_c2l_fs_en
24	R/W	0	i_c2l_hs_en
23	R/W	0	Bypass_Host_Disconnect_Enable
22	R/W	0	Bypass_Host_Disconnect_Value
21	R/W	0	Clear_Hold_HS_disconnect
20:16	R/W	0xc	minimum_count_for_sync_detection
15	R/W	0	Update_PMA_signals
14	R/W	0	load_stat
13:8	R/W	0	reg32_13_13_8_reserved
7:0	R/W	0	Custom_Pattern_19

reg32_14 0x38

Bit(s)	R/W	Default	Description
31:24	R/W	0	reg32_14_31_24_reserved
23:16	R/W	0	Bypass_ctrl_15_8
15:8	R/W	0	Bypass_ctrl_7_0
7	R/W	0	reg32_14_7_reserved
6	R/W	0	i_c2l_assert_single_enable_zero
5	R/W	0	i_c2l_data_16_8
4	R/W	0	pg_rstn
3:2	R/W	0	i_rpu_sw2_en
1	R/W	0	i_rpu_sw1_en
0	R/W	0	i_rpd_en

reg32_15 0x3c

Bit(s)	R/W	Default	Description
31:29	R/W	0	reg32_15_31_29_reserved
28:16	R/W	0xfa0	ms_4_cntr
15:8	R/W	0x3c	non_se0_cntr
7:0	R/W	0x3c	se0_cntr

reg32_16 0x40

Bit(s)	R/W	Default	Description
31	RO	0	usb2_mppll_lock_dig
30	RO	0	usb2_mppll_lock
29	R/W	0	usb2_mppll_reset
28	R/W	0	usb2_mppll_en
27	R/W	0x1	usb2_mppll_fast_lock
26	R/W	0	usb2_mppll_lock_f
25:24	R/W	0x1	usb2_mppll_lock_long
23	R/W	0	usb2_mppll_dco_sdm_en
22	R/W	0x1	usb2_mppll_load
21	R/W	0	usb2_mppll_sdm_en
20	R/W	0	usb2_mppll_tdc_mode
19:15	R/W	0	reg32_16_19_15_reserved
14:10	R/W	0x1	usb2_mppll_n
9	R/W	0	reg32_16_9_reserved
8:0	R/W	0x14	usb2_mppll_m

reg32_17 0x44

Bit(s)	R/W	Default	Description
31:28	R/W	0x9	usb2_mppll_filter_pvt1
27:24	R/W	0x2	usb2_mppll_filter_pvt2
23	R/W	0	usb2_mppll_filter_mode
22:20	R/W	0x7	usb2_mppll_lambda0
19:17	R/W	0x7	usb2_mppll_lambda1

16	R/W	0	usb2_mppll_fix_en
15:14	R/W	0	reg32_17_15_14_reserved
13:0	R/W	0	usb2_mppll_frac_in

reg32_18 0x48

Bit(s)	R/W	Default	Description
31	R/W	0	usb2_mppll_acg_range
30:29	R/W	0x3	usb2_mppll_adj_ldo
28:26	R/W	0x3	usb2_mppll_alpha
25:24	R/W	0x1	usb2_mppll_bb_mode
23:22	R/W	0x1	usb2_mppll_bias_adj
21:19	R/W	0x3	usb2_mppll_data_sel
18:16	R/W	0x3	usb2_mppll_rou
15:14	R/W	0	usb2_mppll_pfd_gain
13	R/W	0x1	usb2_mppll_dco_clk_sel
12	R/W	0	usb2_mppll_dco_m_en
11:6	R/W	0x27	usb2_mppll_lk_s
5:2	R/W	0x9	usb2_mppll_lk_w
1:0	R/W	0x1	usb2_mppll_lkw_sel

reg32_19 0x4c

Bit(s)	R/W	Default	Description
31	RO	0	usb2_mppll_lock_dig
30	RO	0	usb2_mppll_lock
29:10	RO	0	reg32_19_29_10_reserved
9:0	RO	0	usb2_mppll_reg_out

reg32_20 0x50

Bit(s)	R/W	Default	Description
31	R/W	0	bypass_cal_done_r5
30:29	R/W	0	usb2_bgr_dbg_1_0
28:24	R/W	0	usb2_bgr_vref_4_0
23:22	R/W	0	reg32_20_23_22_reserved

Bit(s)	R/W	Default	Description
21	R/W	0	usb2_bgr_start
20:16	R/W	0	usb2_bgr_adj_4_0
15:14	R/W	0	usb2_edgedrv_trim_1_0
13	R/W	0	usb2_edgedrv_en
12:9	R/W	0xf	usb2_dmon_sel_3_0
8	R/W	0	usb2_dmon_en
7	R/W	0	bypass_otg_det
6	R/W	0	usb2_cal_code_r5
5	R/W	0	usb2_amon_en
4	R/W	0x1	usb2_otg_vbusdet_en
3:1	R/W	0x4	usb2_otg_vbus_trim_2_0
0	R/W	0	usb2_otg_iddet_en

reg32_21 0x54

Bit(s)	R/W	Default	Description
31:26	R/W	0	reg32_21_31_26_reserved
25:20	R/W	0	bypass_utmi_reg
19:16	R/W	0	bypass_utmi_cntr
15:6	R/W	0	reg32_21_15_6_reserved
5:4	R/W	0x2	usb2_otg_aca_trim_1_0
3	R/W	0	usb2_tx_strg_pd
2	R/W	0x1	usb2_otg_aca_en
1	R/W	0x1	usb2_cal_ack_en
0	R/W	0	usb2_bgr_force

reg32_22 0x58

Bit(s)	R/W	Default	Description
31:6	RO	0	reg32_22_31_6_reserved
5:3	RO	0	usb2_otg_aca_iddig
2	RO	0	usb2_otg_vbus_vld
1	RO	0	usb2_otg_sess_vld

0	RO	0x2	usb2_otg_id_dig

reg32_23 0x5c

Bit(s)	R/W	Default	Description
31:16	RO	0	test_bus_data_int_15_0
15:8	R/W	0	reg32_23_15_8_reserved
7	R/W	0	orw_test_bus_en
6:1	R/W	0	orw_test_bus_sel_5_0
0	R/W	0	orw_usb2_bgr_en

14.2 PCIE

PCIE module includes PCIE controller and PCIE PHY.

14.3 Ethernet

14.3.1 Overview

The Ethernet MAC controller provides a complete Ethernet interface from the chip to a Reduced Gigabit Media Independent Interface(RGMII) compliant Ethernet PHY.

14.3.2 Ethernet MAC

14.3.2.1 Features

Ethernet MAC has the following features:

- 10/100/1000 MAC 3.70a
- RGMII/RMII
- AHB 32 Bits internal bus
- RX FIFO 4KB, TX FIFO 2KB
- 2 MAC addresses
- EEE
- Power Management

14.3.2.2 Register Description

PRG_ETH_REG0 0xff634540

Bit(s)	R/W	Default	Description
31	R/W	0	Set AHB to DDR interface as urgent.
30	R/W	0	RGMII mode Use RX_CLK as TX_CLK.
29-27	R/W	0	RMII & RGMII mode Select one signal from {RXDV, RXD[3:0]} to calibrate.
26	R/W	0	RMII & RGMII mode 0: test falling edge 1: test rising edge
25	R/W	0	RMII & RGMII mode Start calibration logic
24-20	R/W	0	RMII & RGMII mode 5 Bits correspondent to {RXDV, RXD[3:0]}, set to 1 will delay the data capture by 1 cycle.
19-15	R/W	0	Set bit14 to 0. RMII & RGMII mode Capture input data at clock index equal to adj_delay.
14	R/W	0	Set RXDV and RXD setup time, data is aligned with index 0. When set to 1, auto delay and skew
13	R/W	0	RMII & RGMII mode Enable data delay adjustment and calibration logic.

Bit(s)	R/W	Default	Description
12	R/W	0	RMII & RGMII mode Enable TX_CLK and PHY_REF_CLK generator.
11	R/W	0	RMII mode Use inverted internal clk_rmii_i to generate 25/2.5 tx_rx_clk.
10	R/W	0	Generate 25MHz clock for PHY
9-7	R/W	0	RMII & RGMII mode, 000: invalid value. 001: mp2_clk_out is 250MHz. 010: mp2_clk_out is 500MHz. … Mp2_clk_out is “ratio” *250MHz.
6-5	R/W	0	RGMII mode, TX_CLK related to TXD 00: clock delay 0 cycle. 01: clock delay ¼ cycle. 10: clock delay ½ cycle. 11: clock delay ¾ cycle.
4	R	0	Unused
3	R/W	0	RMII mode CLK_RMII RGMII mode RX_CLK Use inverted signal when set to 1.
2	R/W	0	Sideband Descriptor Endianness Control Function: When set high, this signal configures the DMA to transfer descriptors in reverse endianness of the data format. When low (by default), the descriptors are transferred in the same endian format as the data. This signal is sampled during active reset (including soft-reset) only and ignored after reset is de-asserted.
1	R/W	0	Sideband Data Endianness Control Function: When set high, this signal configures the DMA to transfer data in big-endian format. When low (by default), the data is transferred in little-endian format. This signal is sampled during active reset (including soft-reset) only and ignored after reset is de-asserted.
0	R/W	0	PHY Interface Select Function: These pins select one of the multiple PHY interfaces of MAC. This is sampled only during reset assertion and ignored after that. 1: internal value 001: RGMII 0: internal value 100: RMII

PRG_ETH_REG1 0xff634544

Bit(s)	R/W	Default	Description
31-16	R	0	Unused

15	R/W	0	The result is valid
14	R/W	0	The results is rising edge test or falling edge test.
13-11	R/W	0	The signal under test.
10	R/W	0	The Calibration logic is waiting for event.
9-5	R/W	0	The RX_CLK length in 1ns.
4-0	R/W	0	Signal switch position in 1ns.

14.3.3 Ethernet PHY

14.3.3.1 Features

Ethernet PHY has the following features:

- Integrated IEEE 802.3/802.3u compliant 10/100Mbps Ethernet PHY
- Supporting both full and half-duplex for either 10 or 100 Mb/s data rate
- Auto MDIX capable
- Supports wake-on-LAN,
- 100Base-T support
- MII/RMII/SMII interface
- Supports auto-negotiation
- Full set of power down modes
- Interface available to 100Base-FX Fiber-PMD
- Serial Management Interface (SMI)
- Fix configurations for LED status indicators
- Supporting military temperature range -20C to 80C
- Perfect mix of analog and digital lends itself to robustness, portability, and performance
- Multiple input clock options
- Stand-alone core

14.3.3.2 Register Description

ETH_PHY_DBG_CTL0 0xFF64C000

Bit(s)	R/W	Default	Description
[7:0]	R/W	0	ETH digital debug registers

ETH_PHY_DBG_CTL1 0xFF64C004

Bit(s)	R/W	Default	Description
[23:0]	R/W	0	ETH digital debug registers

ETH_PHY_DBG_CFG0 0xFF64C008

Bit(s)	R/W	Default	Description
[31:0]	R/W	0	ETH digital debug registers

ETH_PHY_DBG_CFG1 0xFF64C00c

Bit(s)	R/W	Default	Description
[31:0]	R/W	0	ETH digital debug registers

ETH_PHY_DBG_CFG2 0xFF64C010

Bit(s)	R/W	Default	Description
[31:0]	R/W	0	ETH digital debug registers

ETH_PHY_DBG_CFG3 0xFF64C014

Bit(s)	R/W	Default	Description
[31:0]	R/W	0	ETH digital debug registers

ETH_PHY_DBG_CFG4 0xFF64C018

Bit(s)	R/W	Default	Description
[31:0]	R/W	0	ETH digital debug registers

ETH_PLL_STS 0xFF64C040

Bit(s)	R/W	Default	Description
[31]	R		eth_mppll_lock
[30]	R		eth_mppll_lock_dig
[29:10]	R		reserved
[9:0]	R		eth_mppll_reg_out

ETH_PLL_CTL0 0xFF64C044

Bit(s)	R/W	Default	Description
[31]	R		eth_mppll_lock
[30]	R		eth_mppll_lock_dig
[29]	R/W		eth_mppll_reset
[28]	R/W		eth_mppll_en
[27]	R/W		eth_mppll_fast_lock
[26]	R/W		eth_mppll_lock_f
[25:24]	R/W		eth_mppll_lock_long
[23]	R/W		eth_mppll_sel_ref
[22]	R/W		eth_mppll_load

Bit(s)	R/W	Default	Description
[21]	R/W		eth_mppll_sdm_en
[20]	R/W		eth_mppll_tdc_mode
[19:15]	R/W		reserved
[14:10]	R/W		eth_mppll_n
[9]	R/W		reserved
[8:0]	R/W		eth_mppll_m

ETH_PLL_CTL1 0xFF64C048

Bit(s)	R/W	Default	Description
[31:28]	R/W		eth_mppll_filter_pvt1
[27:24]	R/W		eth_mppll_filter_pvt2
[23]	R/W		eth_mppll_filter_mode
[22:20]	R/W		eth_mppll_lambda0
[19:17]	R/W		eth_mppll_lambda1
[16]	R/W		eth_mppll_fix_en
[15:14]	R/W		reserved
[13:0]	R/W		eth_mppll_frac_in

ETH_PLL_CTL2 0xFF64C04C

Bit(s)	R/W	Default	Description
[31]	R/W		eth_mppll_acq_range
[30:29]	R/W		eth_mppll_adj_ldo
[28:26]	R/W		eth_mppll_alpha
[25:24]	R/W		eth_mppll_bb_mode
[23:22]	R/W		eth_mppll_bias_adj
[21:19]	R/W		eth_mppll_data_sel
[18:16]	R/W		eth_mppll_rou
[15:14]	R/W		eth_mppll_pfd_gain
[13]	R/W		eth_mppll_dco_clk_sel
[12]	R/W		eth_mppll_dco_m_en
[11:6]	R/W		eth_mppll_lk_s

Bit(s)	R/W	Default	Description
[5:2]	R/W		eth_mppll_lk_w
[1:0]	R/W		eth_mppll_lkw_sel

ETH_PLL_CTL3 0xFF64C050

Bit(s)	R/W	Default	Description
[31:1]	R		reserved
[0]	R/W		eth_mppll_dco_sdm_en

ETH_PLL_CTL4 0xFF64C054

Bit(s)	R/W	Default	Description
[31:0]	R/W		reserved

ETH_PLL_CTL5 0xFF64C058

Bit(s)	R/W	Default	Description
[31:16]	R/W	0	ETH_PHY_RXADC0
[15:3]	R/W	0	ETH_PHY_CTLIO
[2:0]	R/W	0	reserved

ETH_PLL_CTL6 0xFF64C05C

Bit(s)	R/W	Default	Description
[31:16]	R/W	0	reserved
[15:0]	R/W	0	ETH_PHY_RXSQLD

ETH_PLL_CTL7 0xFF64C060

Bit(s)	R/W	Default	Description
[31:16]	R/W	0	reserved
[15:0]	R/W	0	ETH_PHY_RXADC1

ETH_PHY_CNTL0 0xFF64C080

Bit(s)	R/W	Default	Description
[31:16]	R/W	0	co_reg3_oui_in, SMI register 3 default value
[15:0]	R/W	0	co_reg2_oui_in, SMI register 2 default value

ETH_PHY_CNTL1 0xFF64C084

Bit(s)	R/W	Default	Description
[31:24]	R/W	0	led_cfg, led_cfg[7:5]: co_activity_led output select 0: link_led & (~activity) 1: link_led 2: link_led & activity 3: link_led | activity 4: activity 5: rx_led 6: tx_led 7: duplex_led led_cfg[4:2]: co_link_speed_led output select 0: speed100_led 1: activity & speed100_led 2: link_led & speed100_led 3: link_led & activity & speed100_led 4: speed10_led 5: activity & speed10_led 6: link_led & speed10_led 7: link_led & activity & speed10_led led_cfg[1]: invert co_link_speed_led when output to gpio led_cfg[0]: invert all led signal from phy
[23]	R/W	0	invert co_activity_led invert co_activity_led when output to gpio
[22]	R/W	0	co_pwruprst_byp
[21]	R/W	0	co_clk_ext
[20]	R/W	0	co_st_scan
[19]	R/W	0	co_rxclk_inv
[18]	R/W	0	co_phy_enb
[17]	R/W	0	co_clkfreq
[16]	R/W	0	eth_clk_enable
[15:14]	R/W	0	co_st_miimode[1:0]
[13]	R/W	0	co_smii_source_sync
[12]	R/W	0	co_st_pllbp
[11]	R/W	0	co_st_adcbp
[10]	R/W	0	co_st_fxmode

Bit(s)	R/W	Default	Description
[9]	R/W	0	co_en_high
[8]	R/W	0	co_automdix_en
[7:3]	R/W	0	co_st_phyadd[4:0]
[2:0]	R/W	0	co_st_mode[2:0]

ETH_PHY_CNTL2 0xFF64C088

Bit(s)	R/W	Default	Description
[31]	R/W	0	reserved
[30]	R/W	0	eth_phy_clk25min_en: enable 125M clock in test/debug mode
[29:28]	R/W	0	reserved
[27:24]	R/W	0	debug output bus select
[23:13]	R/W	0	reserved
[12]	R/W	0	analog production test mode enable
[11]	R/W	0	mdi source select 0: normal 1: debug in
[10]	R/W	0	reserved
[9]	R/W	0	source select for rx_clk to mac 0: gpio 1: co_clkin from ephy
[8]	R/W	0	source select for tx_clk output to gpio 0: tx_clk from mac 1: co_tx_clk from ephy
[7]	R/W	0	rx_dv/col switch for mac
[6]	R/W	0	ephy smi source select 1: from mac
[5]	R/W	0	use internal phy
[4]	R/W	0	ephy smi source select 1: from debug in
[3]	R/W	0	co_clkin source select 1: from debug in
[2]	R/W	0	co_mdclk source select 1: from debug in
[1]	R/W	0	enable reset in test mode
[0]	R/W	0	ephy loopback mode enable, in/out through debug in/out gpio

ETH_PHY_STS0 0xFF64C094

Bit(s)	R/W	Default	Description
[31:28]	R		reserved
[27:19]	R		co_int_vec[8:0]
[19]	R		reserved
[18:0]	R		ETH_PLL_STS[18:0]

ETH_PHY_STS1 0xFF64C098

Bit(s)	R/W	Default	Description
[31:0]	R		eth_phy_dbg_prb

ETH_PHY_STS2 0xFF64C09C

Bit(s)	R/W	Default	Description
[31:26]	R		eth_phy_rxda[5:0]
[25]	R		eth_phy_rxadcoflw
[24]	R		eth_phy_rxadcuflw
[23]	R		eth_phy_rxprepga
[22:20]	R		eth_phy_rxgain[2:0]
[19:0]	R		eth_test_status

ETH_PHY_DBG_REG 0xFF64C0A0

Bit(s)	R/W	Default	Description
[31:16]	R		eth_phy_dbg_reg
[15:8]			reserved
[7:0]	R/W	0	eth_phy_dbg_reg_mux

14.4 Inter-Integrated Circuit (I2C)

14.4.1 Overview

Inter-Integrated Circuit (IIC or I2C) is a multi-slave serial communication bus between ICs. S922X integrates the I2C interface and signals allowing communications with other I2C peripheral devices.

14.4.2 Features

The I^2C Master Module has the following features:

- Support for 7-bit and 10-bit addressable devices
- Programmable bus speed including standard speed (100kBits/s) and fast speed (400kBits/sec)
- Error transfer detection
- “Transfer complete” indication by polling or interrupt (Interrupts handled by the ISA module. See the ISA module for details).
- Internal buffer holding up to 8 bytes for transfer (in either direction)
- Flexible architecture allowing the software to dictate the format of the I^2C bit streams
- Manual setting of the I^2C bus to accommodate a software only mode

14.4.3 Register Description

For I2C module in EE domain, each register final address = 0x FF805000 + offset * 4 for master mode, final address = 0x FF806000 + offset * 4 for slave mode.

I2C_M_0_CONTROL_REG 0x7c00

Bit(s)	R/W	Default	Description
31	R/W	0	CNTL_JIC: There is internal logic to dynamically enable the gated clocks. If this gated clock logic doesn’t work, you can set This bit to always enable the clock. Setting This bit wastes power.
30	R	0	Unused
29-28	R/W	0	QTR_CLK_EXT: These two Bits extend the clock divider to 12 Bits: QTR_CLK = {[29:28],[21:12]}
27	R	0	unused
26	R	0	Read back level of the SDA line
25	R	0	Read back level of the SCL line
24	R/W	0	Sets the level of the SDA line if manual mode is enabled. If This bit is ‘0’, then the SDA line is pulled low. If This bit is ‘1’ then the SDA line is tri-stated.
23	R/W	0	Sets the level of the SCL line if manual mode is enabled. If This bit is ‘0’, then the SCL line is pulled low. If This bit is ‘1’ then the SCL line is tri-stated.
22	R/W	0	This bit is used to enable manual mode. Manual I2C mode is controlled by Bits 12,13,14 and 15 above.
21:12	R/W	0x142	QTR_CLK_DLY. This value corresponds to period of the SCL clock divided by 4 Quarter Clock Delay = * System Clock Frequency For example, if the system clock is 133Mhz, and the I2C clock period is 10uS (100khz), then Quarter Clock Delay = * 133 Mhz = 332

Bit(s)	R/W	Default	Description
11:8	R	-	READ_DATA_COUNT: This value corresponds to the number of bytes READ over the I2C bus. If this value is zero, then no data has been read. If this value is 1, then Bits [7:0] in TOKEN_RDATA_REG0 contains valid data. The software can read this register after an I2C transaction to get the number of bytes to read from the I2C device.
7:4	R	-	CURRENT_TOKEN: This value reflects the current token being processed. In the event of an error, the software can use this value to determine the error location.
3	R	-	ERROR: This read only Bit is set if the I2C device generates a NACK during writing. This bit is cleared at on the clock cycle after the START Bit is set to 1 indicating the start of list processing. Errors can be ignored by setting the ACK_IGNORE Bit(s) below. Errors will be generated on Writes to devices that return NACK instead of ACK. A NACK is returned by a device if it is unable to accept any more data (for example because it is processing some other real-time function). In the event of an ERROR, the I2C module will automatically generate a STOP condition on the bus.
2	R	-	STATUS: This bit reflects the status of the List processor: 0: IDLE 1: Running. The list processor will enter this state on the clock cycle after the START Bit is set. The software can poll the status register to determine when processing is complete.
1	R/W	0	ACK_IGNORE: Set to 1 to disable I2C ACK detection. The I2C bus uses an ACK signal after every byte transfer to detect problems during the transfer. Current Software implementations of the I2C bus ignore this ACK. This bit is for compatibility with the current Amlogic software. This bit should be set to 0 to allow NACK operations to abort I2C bus transactions. If a NACK occurs, the ERROR bit above will be set.
0	R/W	0	START: Set to 1 to start list processing. Setting This bit to 0 while the list processor is operating causes the list processor to abort the current I2C operation and generate an I2C STOP command on the I2C bus. Normally This bit is set to 1 and left high until processing is complete. To re-start the list processor with a new list (after a previous list has been exhausted), simply set This bit to zero then to one.

I2C_M_0_SLAVE ADDRESS 0x7c01

Bit(s)	R/W	Default	Description
31:29	R	0	Reserved
28	R/W	0	USE_CNTL_SCL_LOW: If This bit is set to 1, then Bits[27:16] control the SCL low time.
27:16	R/W	0	SCL Low delay.
15:14	R	0	Unused
13-11	R/W	0	SCL_FILTER: A filter was added in the SCL input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
10:8	R/W	0	SDA FILTER: A filter was added in the SDA input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
7:0	R/W	0x00	SLAVE_ADDRESS. This is a 7-bit value for a 7-bit I2C device, or (0xF0 | {A9,A8}) for a 10-bit I2C device. By convention, the slave address is typically stored in by first left shifting it so that it's MSB is D7 (The I2C bus assumes the 7-bit address is left shifted one). Additionally, since the SLAVE address is always an 7-bit value, D0 is always 0. NOTE: The I2C always transfers 8-bits even for address. The I2C hardware will use D0 to dictate the direction of the bus. Therefore, D0 should always be '0' when this register is set.

I2C_M_0_TOKEN_LIST_REG0 0x7c02

The register below describes the first 8 tokens in the token list.

Bit(s)	R/W	Default	Description
31:28	R/W	0x00	8th token in the list to process
27:24	R/W	0x00	7th token in the list to process
23:20	R/W	0x00	6th token in the list to process
19:16	R/W	0x00	5th token in the list to process
15:12	R/W	0x00	4th token in the list to process
11:8	R/W	0x00	3rd token in the list to process
7:4	R/W	0x00	2nd token in the list to process
3:0	R/W	0x00	1st token in the list to process (See the table below for token definitions)

Table 14-1 Token Definitions

Command Token	Value	Data	Description
END	0x0	N/A	Used to tell the I2C module that this is the end of the Token list. This token is not associated with the I2C bus, but rather with the state-machine that drives the token list processor.
START	0x1	N/A	The START Token is used to tell an I2C device that this is the beginning of an I2C transfer
SLAVE_ADDR-WRITE	0x2	7-bits	This bit-sequence is used to address a device and tell the device it is being WRITTEN
SLAVE_ADDR-READ	0x3	7-bits	This bit sequence is used to address a device and tell the device it is being READ.
DATA	0x4	8-bits	This 8-bit byte sequence is a byte transfer (READ or WRITE). The DATA token corresponds to a WRITE if it follows a SLAVE_ADDR-WRITE token. The DATA token corresponds to a READ if it follows a SLAVE_ADDR-READ token.
DATA-LAST	0x5	8-bits	Used to indicate the last 8-bit byte transfer is a byte transfer of a READ.
STOP	0x6	N/A	This tells the I2C device it is no longer being addressed

Write data associated with the DATA token should be placed into the I2C_TOKEN_WDATA_REG0 or I2C_TOKEN_WDATA_REG1 registers. Read data associated with the DATA or DATA-LAST token can be read from the I2C_TOKEN_RDATA_REG0 or I2C_TOKEN_RDATA_REG1 registers.

I2C_M_0_TOKEN_LIST_REG1 0x7c03

Bit(s)	R/W	Default	Description
31:28	R/W	0x00	16th token in the list to process

27:24	R/W	0x00	15th token in the list to process
23:20	R/W	0x00	14th token in the list to process
19:16	R/W	0x00	13th token in the list to process
15:12	R/W	0x00	12th token in the list to process
11:8	R/W	0x00	11th token in the list to process
7:4	R/W	0x00	10th token in the list to process
3:0	R/W	0x00	9th token in the list to process

I2C_M_0_TOKEN_WDATA_REG0 0x7c04

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte written for a DATA (write) token.
23:16	R/W	0x00	3rd data byte written for a DATA (write) token.
15:8	R/W	0x00	2nd data byte written for a DATA (write) token.
7:0	R/W	0x00	1st data byte written for a DATA (write) token.

I2C_M_0_TOKEN_WDATA_REG1 0x7c05

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte written for a DATA (write) token.
23:16	R/W	0x00	7th data byte written for a DATA (write) token.
15:8	R/W	0x00	6th data byte written for a DATA (write) token.
7:0	R/W	0x00	5th data byte written for a DATA (write) token.

I2C_M_0_TOKEN_RDATA_REG0 0x7c06

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	3rd data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	2nd data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	1st data byte read for a DATA or DATA-LAST (READ) token.

I2C_M_0_TOKEN_RDATA_REG1 0x7c07

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	7th data byte read for a DATA or DATA-LAST (READ) token.

15:8	R/W	0x00	6th data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	5th data byte read for a DATA or DATA-LAST (READ) token.

I2C_M_1_ 0x7800~0x7807

See I2C_M_0

I2C_M_2_ 0x7400~0x7407

See I2C_M_0

I2C_M_3_ 0x7000~0x7007

See I2C_M_0

For I2C module in AO domain, each register final address = 0x FF805000 + offset * 4 for master mode, final address = 0x FF806000 + offset * 4 for slave mode.

AO_I2C_M_0_CONTROL_REG 0x0

Bit(s)	R/W	Default	Description
31	R/W	0	CNTL_JIC: There is internal logic to dynamically enable the gated clocks. If this gated clock logic doesn't work, you can set this bit to always enable the clock. Setting this bit wastes power.
30	R	0	Unused
29-28	R/W	0	QTR_CLK_EXT: These two bits extend the clock divider to 12 bits: QTR_CLK = {[29:28],[21:12]}
27	R	0	unused
26	R	0	Read back level of the SDA line
25	R	0	Read back level of the SCL line
24	R/W	0	Sets the level of the SDA line if manual mode is enabled. If this bit is '0', then the SDA line is pulled low. If this bit is '1' then the SDA line is tri-stated.
23	R/W	0	Sets the level of the SCL line if manual mode is enabled. If this bit is '0', then the SCL line is pulled low. If this bit is '1' then the SCL line is tri-stated.
22	R/W	0	This bit is used to enable manual mode. Manual I2C mode is controlled by bits 12,13,14 and 15 above.
21:12	R/W	0x142	QTR_CLK_DLY. This value corresponds to period of the SCL clock divided by 4 Quarter Clock Delay = * System Clock Frequency For example, if the system clock is 133Mhz, and the I2C clock period is 10uS (100khz), then Quarter Clock Delay = * 133 Mhz = 332
11:8	R	-	READ_DATA_COUNT: This value corresponds to the number of bytes READ over the I2C bus. If this value is zero, then no data has been read. If this value is 1, then bits [7:0] in TOKEN_RDATA_REG0 contains valid data. The software can read this register after an I2C transaction to get the number of bytes to read from the I2C device.
7:4	R	-	CURRENT_TOKEN: This value reflects the current token being processed. In the event of an error, the software can use this value to determine the error location.

Bit(s)	R/W	Default	Description
3	R	-	ERROR: This read only bit is set if the I2C device generates a NACK during writing. This bit is cleared at on the clock cycle after the START bit is set to 1 indicating the start of list processing. Errors can be ignored by setting the ACK_IGNORE bit below. Errors will be generated on Writes to devices that return NACK instead of ACK. A NACK is returned by a device if it is unable to accept any more data (for example because it is processing some other real-time function). In the event of an ERROR, the I2C module will automatically generate a STOP condition on the bus.
2	R	-	STATUS: This bit reflects the status of the List processor: 0: IDLE 1: Running. The list processor will enter this state on the clock cycle after the START bit is set. The software can poll the status register to determine when processing is complete.
1	R/W	0	ACK_IGNORE: Set to 1 to disable I2C ACK detection. The I2C bus uses an ACK signal after every byte transfer to detect problems during the transfer. Current Software implementations of the I2C bus ignore this ACK. This bit is for compatibility with the current Amlogic software. This bit should be set to 0 to allow NACK operations to abort I2C bus transactions. If a NACK occurs, the ERROR bit above will be set.
0	R/W	0	START: Set to 1 to start list processing. Setting this bit to 0 while the list processor is operating causes the list processor to abort the current I2C operation and generate an I2C STOP command on the I2C bus. Normally this bit is set to 1 and left high until processing is complete. To re-start the list processor with a new list (after a previous list has been exhausted), simply set this bit to zero then to one.

AO_I2C_M_0_SLAVE_ADDR 0x1

Bit(s)	R/W	Default	Description
31:29	R	0	Unused
28	R/W	0	USE_CNTL_SCL_LOW: If this bit is set to 1, then bits[27:16] control the SCL low time.
27:16	R/W	0	SCL Low delay. This is a new feature in M8baby. In the previous M8baby design, the SCL low time was controlled by bits[21:12] of the register above. In this design, the SCL delay is controlled independently by these bits.
15:14	R	0	Unused
13:11	R/W	0	SCL_FILTER: A filter was added in the SCL input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
10:8	R/W	0	SDA FILTER: A filter was added in the SDA input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
7:0	R/W	0x00	SLAVE_ADDRESS. This is a 7-bit value for a 7-bit I2C device, or (0xF0 | {A9,A8}) for a 10-bit I2C device. By convention, the slave address is typically stored in by first left shifting it so that it's MSB is D7 (The I2C bus assumes the 7-bit address is left shifted one). Additionally, since the SLAVE address is always an 7-bit value, D0 is always 0. NOTE: The I2C always transfers 8-bits even for address. The I2C hardware will use D0 to dictate the direction of the bus. Therefore, D0 should always be '0' when this register is set.

AO_I2C_M_0_TOKEN_LIST0 0x2

The register below describes the first 8 tokens in the token list.

Bit(s)	R/W	Default	Description

31:28	R/W	0x00	8th token in the list to process
27:24	R/W	0x00	7th token in the list to process
23:20	R/W	0x00	6th token in the list to process
19:16	R/W	0x00	5th token in the list to process
15:12	R/W	0x00	4th token in the list to process
11:8	R/W	0x00	3rd token in the list to process
7:4	R/W	0x00	2nd token in the list to process
3:0	R/W	0x00	1st token in the list to process (See the table below for token definitions)

Table 14-2 Token Definitions

Command Token	Value	Data	Description
END	0x0	N/A	Used to tell the I2C module that this is the end of the Token list. This token is not associated with the I2C bus, but rather with the state-machine that drives the token list processor.
START	0x1	N/A	The START Token is used to tell an I2C device that this is the beginning of an I2C transfer
SLAVE_ADDR-WRITE	0x2	7-bits	This bit-sequence is used to address a device and tell the device it is being WRITTEN
SLAVE_ADDR-READ	0x3	7-bits	This bit sequence is used to address a device and tell the device it is being READ.
DATA	0x4	8-bits	This 8-bit byte sequence is a byte transfer (READ or WRITE). The DATA token corresponds to a WRITE if it follows a SLAVE_ADDR-WRITE token. The DATA token corresponds to a READ if it follows a SLAVE_ADDR-READ token.
DATA-LAST	0x5	8-bits	Used to indicate the last 8-bit byte transfer is a byte transfer of a READ.
STOP	0x6	N/A	This tells the I2C device it is no longer being addressed

Write data associated with the DATA token should be placed into the I2C_TOKEN_WDATA_REG0 or I2C_TOKEN_WDATA_REG1 registers. Read data associated with the DATA or DATA-LAST token can be read from the I2C_TOKEN_RDATA_REG0 or I2C_TOKEN_RDATA_REG1 registers.

AO_I2C_M_0_TOKEN_LIST1 0x3

Bit(s)	R/W	Default	Description
31:28	R/W	0x00	16th token in the list to process
27:24	R/W	0x00	15th token in the list to process
23:20	R/W	0x00	14th token in the list to process
19:16	R/W	0x00	13th token in the list to process
15:12	R/W	0x00	12th token in the list to process

11:8	R/W	0x00	11th token in the list to process
7:4	R/W	0x00	10th token in the list to process
3:0	R/W	0x00	9th token in the list to process

AO_I2C_M_0_WDATA_REG0 0x4

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte written for a DATA (write) token.
23:16	R/W	0x00	3rd data byte written for a DATA (write) token.
15:8	R/W	0x00	2nd data byte written for a DATA (write) token.
7:0	R/W	0x00	1st data byte written for a DATA (write) token.

AO_I2C_M_0_WDATA_REG1 0x5

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte written for a DATA (write) token.
23:16	R/W	0x00	7th data byte written for a DATA (write) token.
15:8	R/W	0x00	6th data byte written for a DATA (write) token.
7:0	R/W	0x00	5th data byte written for a DATA (write) token.

AO_I2C_M_0_RDATA_REG0 0x6

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	4th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	3rd data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	2nd data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	1st data byte read for a DATA or DATA-LAST (READ) token.

AO_I2C_M_0_RDATA_REG1 0x7

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte read for a DATA or DATA-LAST (READ) token.
23:16	R/W	0x00	7th data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	6th data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	5th data byte read for a DATA or DATA-LAST (READ) token.

AO_I2C_M_0_TIMEOUT_TH 0x8

Bit(s)	R/W	Default	Description
31:24	R/W	0x00	8th data byte read for a DATA or DATA-LAST (READ) token.

Bit(s)	R/W	Default	Description
23:16	R/W	0x00	7th data byte read for a DATA or DATA-LAST (READ) token.
15:8	R/W	0x00	6th data byte read for a DATA or DATA-LAST (READ) token.
7:0	R/W	0x00	5th data byte read for a DATA or DATA-LAST (READ) token.

AO_I2C_S_CONTROL_REG 0x0

Bit(s)	R/W	Default	Description
31-29	R	0	REG POINTER: There are 5 internal registers inside the I2C slave module. The I2C Master sets this value using the byte that follows the address byte in the I2C data stream. Register 4 (numbered 0,1,…4) is the status register.
28	R/W	0	SEND READY: This bit is set to '1' by the ARC to indicate to the slave machine that the I2C slave module is ready to send data. This bit is cleared by the I2C module when it has sent 4 bytes to the I2C master. This bit is also available in the status register that can be read by the I2C master. The I2C master can read the status register to see when the I2C slave module has data to send.
27	R/W	0	RECEIVE READY: This bit is set to '1' by the ARC to indicate to the slave machine that the I2C slave module is ready to receive data. This bit is cleared by the I2C module when it has received 4 bytes from the I2C master. This bit is also available in the status register that can be read by the I2C master. The I2C master can read the status register to see when the I2C slave module is ready to receive data.
26	R	0	BUSY: Read only status bit. '1' indicates that the I2C slave module is sending or receiving data.
25	R/W	0	IRQ_EN: If this bit is set, then an interrupt will be sent to the ARC whenever 4 bytes have been read or 4 bytes have been written to the I2C slave module.
24	R/W	0	ACK Always: Typically the ACK of a slave I2C device is dependent upon the availability of data (if reading) and room to store data (when we are being written). Our I2C module has a status register that can be read continuously. This bit can be set if the I2C master wants to continually read the status register.
23-16	R/W	0	Slave Address: Bits [7:1] are used to identify the device. Bit [0] is ignored since this corresponds to the R/W bit.
15-8	R/W	0x27	HOLD TIME: Data hold time after the falling edge of SCL. This hold time is computed as Hold time = (MPEG system clock period) * (value + 1).
7	R/W	0	Enable: A '1' enables the I2C slave state machine.
6-0	R/W	0x06	Sampling rate. Defined as MPEG system clock / (value + 1). The SDA and SCL inputs into the slave module are sampled as a way of filtering the inputs. A rising or falling edge is determined by when 3 successive samples are either high or low respectively.

AO_I2C_S_SEND_REG: Send Data 0x1

Bit(s)	R/W	Default	Description
31-0	R/W	0	The I2C slave module can send up to 4 bytes of data starting with the byte located at bits [7:0]

AO_I2C_S_RECV_REG: Received Data 0x2

Bit(s)	R/W	Default	Description
31-0	R	0	This read only register corresponds to the 4 bytes of data written to the I2C slave module by an external I2C master. Bits [7:0] correspond to the first byte written by the I2C master.

AO_I2C_S_CNTL1_REG 0x3

Bit(s)	R/W	Default	Description
31-6	R	0	Unused
5-3	R/W	0	SCL_FILTER: A filter was added in the SCL input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering
2-0	R/W	0x00	SDA FILTER: A filter was added in the SDA input path to allow for filtering of slow rise times. 0 = no filtering, 7 = max filtering

14.5 Universal Asynchronous Receiver And Transmitter

14.5.1 Overview

There are a number of UART's in the chip that offer 2-wire (RX/TX) and 4-wire (RX/TX, CTS/RTS) connections at the digital I/O pins. Each UART contains one transmit FIFO and a receive FIFO (see depths below). The FIFO's are filled by the CPU and read by the CPU. In some cases, the receive FIFO can be configured to be pushed directly to DDR memory without CPU intervention.

Table 14-3 UART List

UART	RX/TX FIFO depths	RX FIFO DMA to DDR	Comment
UART_EE_A	128 bytes	Yes	Located in the EE domain
UART_EE_B	64 bytes	Yes	Located in the EE domain
UART_EE_C	64 bytes	Yes	Located in the EE domain
UART0-AO	64 bytes	No	Located in the Always On domain
UART2-AO	64 bytes	No	Located in the Always On domain

14.5.2 Features

Input filters: The CTS (clear to send) and RX (receive) input paths have input filters to deal with slow rise times. The filters are configurable to use a 125nS or 1uS sampling mechanism. There is an implied 3 system clock cycle delay (15nS for a typical system clock of 200Mhz) that is used to synchronize and detect the rising/falling edge of the RXD signal. The RXD signal may be passed through an optional filter to deglitch the external signal in noisy conditions. The deglitch filter has two settings which add to the ¡§detection delay¡¨ of the RXD signal by the internal logic:

- Filter setting 1 (125nS strobe): 375nS ~ 2.6uS
- Filter setting 2 (1uS strobe): 3uS ~ 21uS

The filter is described in the register specification. If the filter is disabled, the shortest RXD low time and high time is 12 system clock cycles (60nS for a system clock of 200Mhz).

Clear to Send: CTS is a signal sent from the receiver UART back to the transmitting UART to tell the transmitting UART to stop sending data. The CTS signal must be received before the next START symbol is sent. The transmitting UART is allowed to send one more byte after the CTS signal is recognized. The CTS signal coming into the chip goes through some synchronization and detection which adds an additional 5 system clocks (typically 25nS for a 200Mhz system clock). This setup time for CTS detection is called CTSstop. The CTS input also has an optional filter can be used to deglitch the incoming CTS signal. If the filter is disabled, the CTS signal must be de-asserted 5 system clock cycles before the start of the next BYTE transfer. If the CTS filter is enabled, then additional time must be added to the 25nS requirement. There are two programmable filter settings that effectively delay CTS being seen by the internal logic:

- Filter setting 1 (125nS strobe): 375nS ~ 2.6uS
- Filter setting 2 (1uS strobe): 3uS ~ 21uS

Interrupts: The UARTs can generate interrupts if the receive FIFO exceeds a pre-programmed threshold. An interrupt can also be generated if there is a frame or parity error.

Clock independent operation: Because the system clock can be altered to accommodate dynamic frequency scaling, the UARTs have an option in which they use the 24Mhz crystal clock as the source for the UART.

14.5.3 FunctionalDescription

The UART requires that a Baud Rate be established. The UART supports rates as slow as 1Hz up to rates as high as 8 MBits/Sec. Once the baud rate has been established, bytes are transmitted as they are written to the transmit-FIFO by the CPU. A large transmit-FIFO exists to allow the CPU to pre-load a transmit package because the CPU can often write faster than the UART can transmit the data.

Data this automatically received by the UART is placed into the receive FIFO one byte at a time. The receive-FIFO decouples the UART from the CPU allowing the CPU to read the UART byte data at a rate not dictated by the UART.

14.5.4 Register Description

The following Register Description is uniformly applied to all UART instantiations in the chip.

Base_adr:0xffd00000

Final_adr = base_adr + offset *4

UART_EE_A = 128x8 FIFOs

UART_EE_B = 64x8 FIFOs

UART_EE_C = 64x8 FIFOs

UART_EE_A_WFIFO: Write data 0x9000

Bit(s)	R/W	Default	Description
31-8	R	0	unused
7-0	R/W	-	Write FIFO data. The Write FIFO holds 128 or 64 bytes. The Write FIFO can be written as long as it is not full.

UART_EE_A_RFIFO: Read Data 0x9001

Bit(s)	R/W	Default	Description
31-8	R	0	unused
7-0	R/W	-	Read FIFO data. The Read FIFO holds 128 or 64 bytes. The empty flag can be used to determine if data is available

UART_EE_A_CONTROL: UART Mode 0x9002

Bit(s)	R/W	Default	Description
31	R/W	0	Invert the RTS signal
30	R/W	0	Mask Error: Set to 1 to mask errors
29	R/W	0	Invert the CTS signal
28	R/W	0	Transmit byte Interrupt: Set to 1 to enable the generation an interrupt whenever a byte is read from the transmit FIFO
27	R/W	0	Receive byte Interrupt: Set to 1 to enable the generation an interrupt whenever a byte is written to the receive FIFO
26	R/W	0	Set to 1 to invert the TX pin
25	R/W	0	Set to 1 to invert the RX pin

24	R/W	0	Clear Error
23	R/W	0	Reset the receive state machine
22	R/W	0	Reset the transmit state machine
21-20	R/W	0	Character length: 00 = 8 Bits, 01 = 7 Bits, 10 = 6 Bits, 11 = 5 Bits
19	R/W	1	Parity Enable: Set to 1 to enable parity
18	R/W	0	Parity type: 0 = even, 1 = odd
17-16	R/W	0	Stop bit length: 00 = 1 bit, 01 = 2 Bits
15	R/W	0	Two Wire mode:
14	R/W	0	Unused
13	R/W	0	Receive Enable. Set to 1 to enable the UART receive function
12	R/W	0	Transmit Enable. Set to 1 to enable the UART transmit function
11-0	R/W	0x120	Old Baud rate

UART_EE_A_STATUS: UART Status 0x9003

Bit(s)	R/W	Default	Description
31-27	R	0	Unused
26	R	0	UART_RECV_BUSY: This bit will be 1 if the uart receive state machine is busy
25	R	0	UART_XMIT_BUSY: This bit will be 1 if the uart transmit state machine is busy
24	R	0	RECV_FIFO_OVERFLOW:
23	R	0	CTS Level
22	R	0	Transmit FIFO Empty
21	R	0	Transmit FIFO Full
20	R	0	Receive FIFO empty
19	R	0	Receive FIFO full
18	R	0	This bit is set if the FIFO is written when it is full. To clear This bit, wriTE bit 24 of register 0x2132
17	R	0	Frame error. To clear This bit, wriTE bit 24 of register UART_EE_A_CONTROL
16	R	0	Parity error. To clear This bit, wriTE bit 24 of register UART_EE_A_CONTROL
15	R	0	Unused
14-8	R	0	Transmit FIFO count. Number of bytes in the transmit FIFO
7	R	0	Unused

6-0	R	0	Receive FIFO count. Number of bytes in the receive FIFO

UART_EE_A_MISC: UART IRQ CONTROL 0x9004

Bit(s)	R/W	Default	Description
31	R/W	0	Added a “just in case” bit that can be set to 1 to enable clocks always. The default is 0 meaning the auto-clock gating logic is enabled.
30	R/W	0	USE old Rx Baud: There was a bug in the RX baud rate generator. The Rx baud rate generator was re-designed to compute a baud rate correctly. If you want to use the old (stupid) logic, you can set This bit to 1.
29	R/W	0	ASYNC_FIFO_PURGE: This bit can be set to 1 after all UART bytes have been received in order to purge the data into the Async FIFO. This bit is needed because the UART receives 8-bit data, but the ASYNC FIFO can only be written with 16-bit data. In this case there might be a residual byte if the UART is not receiving an even number of bytes.
28	R/W	0	ASYNC_FIFO_EN: If This bit is set to 1, then the UART received data is automatically sent to the Async FIFO module which will in turn automatically send the data to DDR memory
27	R/W	0	CTS: Filter Timebase select: 1 = 1uS, 0 = 111nS timebase. A filter was added to the CTS input to allow for a little digital filtering. The amount of filtering is controlled by this timebase (longer = more filtering) and the value in Bits FILTER_SEL below.
26-24	R/W	0	CTS: FILTER_SEL: 0 = no filter, 7 = max filtering
23-20	R/W	0	old BAUD_RATE_EXT: These 4 Bits extend the baud rate divider to 16-bits: Baud Rate = {Reg4[23:20],Reg2[11:0]}
19	R/W	0	RX: Filter Timebase select: 1 = 1uS, 0 = 111nS timebase. A filter was added to the RX input to allow for a little digital filtering. The amount of filtering is controlled by this timebase (longer = more filtering) and the value in Bits FILTER_SEL below.
18-16	R/W	0	RX: FILTER_SEL: 0 = no filter, 7 = max filtering
15-8	R/W	32	XMIT_IRQ_CNT: The UART can be configured to generate an interrupt if the number of bytes in the transmit FIFO drops below this value.
7:0	R/W	15	RECV_IRQ_CNT: The UART can be configured to generate an interrupt after a certain number of bytes have been received by the UART.

UART_EE_A_REG5 0x9005

Bit(s)	R/W	Default	Description
31-28	R/W	0	unused
27	R/W	0	Xtal2_clk_sel: 0: see Xtal_clk_sel 1: xtal_div2(12M)
26	R/W	0	Xtal_clk_sel: 0: xtal_div3(8M); 1: xtal(24M);(need set xtal_tick_en =1 first);
24	R/W	0	USE_XTAL_CLK: If this bit is set, then the clock for generating the UART Baud rate comes from the crystal pad. This allows the UART to operate independent of clk81.

23	R/W	0	USE New Baud rate. Over the years, the baud rate has been extended by concatenating bits from different registers. To take advantage of the full 23-bit baud rate generate (extended to 23 bits to accommodate very low baud rates), you must set this bit. If this bit is set, then the baud rate is configured using bits [22:0] below
22:0	R/W	15	NEW_BAUD_RATE: If bit[23] = 1 above, then the baud rate for the UART is computed using these bits. This was added in M6 to accommodate lower baud rates.

UART_EE_B_WFIFO: Write data 0x8c00

See UART_EE_A bitDescriptions

UART_EE_B_RFIFO: Read Data 0x8c01

See UART_EE_A bitDescriptions

UART_EE_B_CONTROL: UART Mode 0x8c02

See UART_EE_A bitDescriptions

UART_EE_B_STATUS: UART Status 0x8c03

See UART_EE_A bitDescriptions

UART_EE_B_MISC: UART IRQ CONTROL 0x8c04

See UART_EE_A bitDescriptions

UART_EE_B_REG5 0x8c05

See UART_EE_A bitDescriptions

UART_EE_C_YFIFO: write data 0x8800

See UART_EE_A bitDescriptions

UART_EE_C_RFIFO: Read Data 0x8801

See UART_EE_A bitDescriptions

UART_EE_C_CONTROL: UART Mode 0x8802

See UART_EE_A bitDescriptions

UART_EE_C_STATUS: UART Status 0x8803

See UART_EE_A bitDescriptions

UART_EE_C_MISC: UART IRQ CONTROL 0x8804

See UART_EE_A bitDescriptions

UART_EE_C_REG5 0x8805

See UART_EE_A bitDescriptions

14.6 Infrared Remote

14.6.1 Overview

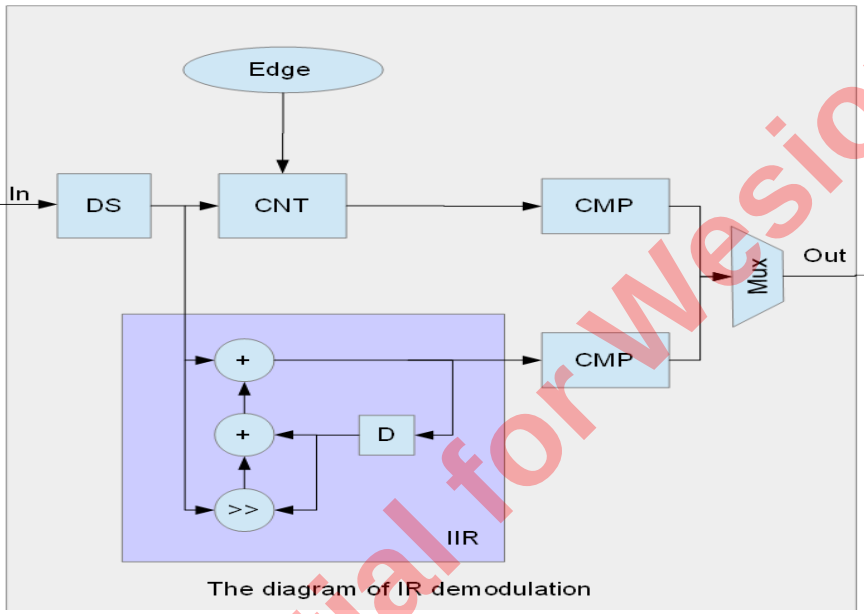
IR module includes 3 sub modules: IR demodulation, Legacy IR remote control and Multi-format IR remote control.

14.6.2 IR Demodulation

IR demolation module demodulate the modulate signal to get the envelop signal.

IR demulation diagram is shown below.

Figure 14-1 IR Demulation Diagram



It implements two methods for demodulation. The input data is an one-bit stream.

The input data are down-sampled firstly. The rate of downsample are configured by register, ranges from 0 to 255.

One method is implemented in the upper path.

The CNT is used to counte the time for ‘0’ and ‘1’. It is cleared when input data changes(from 1 to 0 or from 0 to 1).

When (CNT>REG_IR_PROCT1 && in==1) , it outputs ‘1’. REG_IR_PROCT1 is used to canceled the glitch. Actually, the operation of downsample also have some utility to cancel the glitch.

When (CNT>REG_IR_DECT0 && in==0) , it outputs ‘0’. REG_IR_DECT0 is used to confirm the ‘0’. The value should be bigger than one carrier period.

The other method is implemented in the lower path.

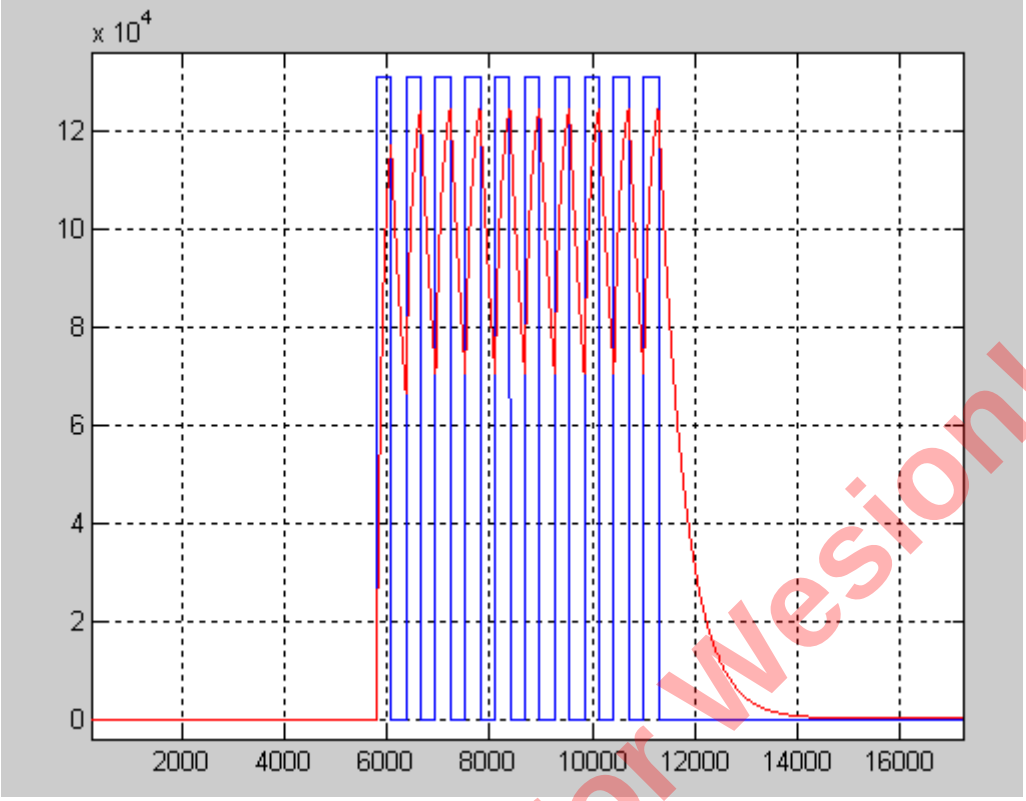
It is a one tap IIR filter. The input data control the integrating factor and the factor can be configured from registers. When 1 inputs, the accumulator can grow quickly. When 0 input, the accumulator decays slowly, and when the number of 0 is bigger enough, it decays soon.

When (accum> REG_IR_IIR_THD1), it outputs ‘1’.

When (accum< REG_IR_IIR_THD0), it outputs ‘0’.

The diagram shows the simple simulation result. The blue is the input data, the red is the filter out.

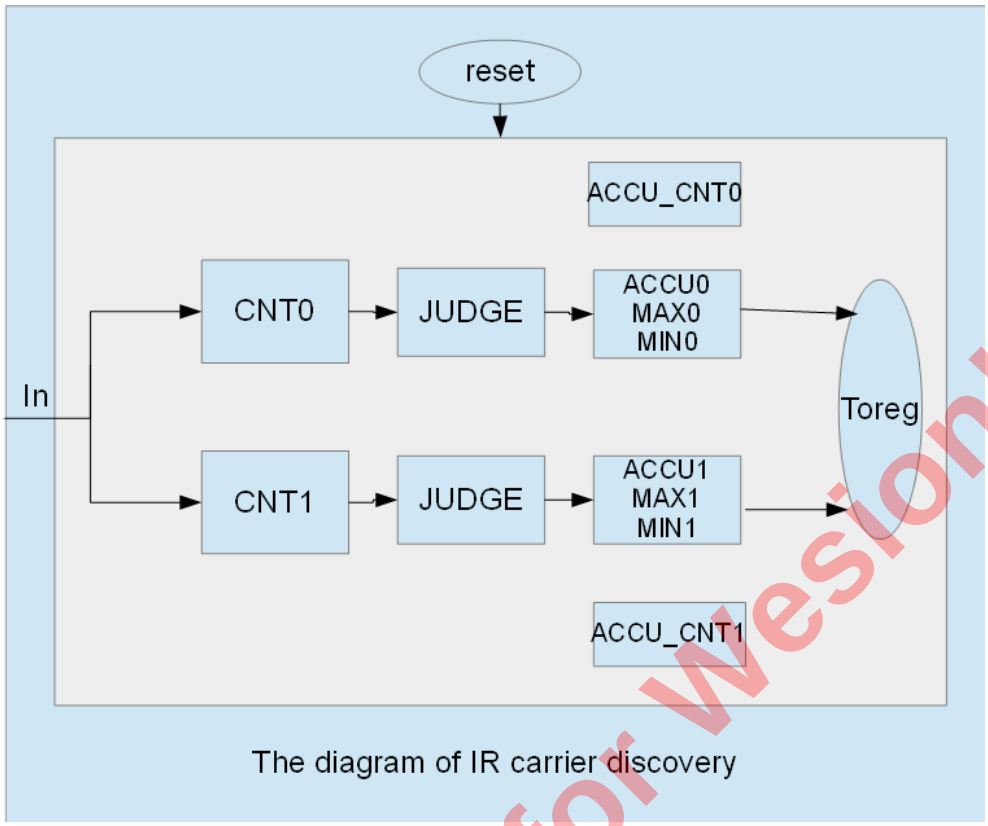
Figure 14-2 Simulation Result of IR Demulation



Because the input carrier signal is a rectangular wave signal, the carrier can be estimated easily by count the wave period. For estimating precisely, the counter for the long “0” and start period and and glitch should be removed.

The simple diagram is shown below:

Figure 14-3 IR Carrier Discovery



CNT is the counter for time of “0” and “1”.

JUDGE is used to remove the long “0” and glitch.

ACC_CNT is the counter for number of accumulating. If ACC_CNT== REG_IR_ST_CNT_THD, then write the value to registers.

For example:

Set REG_IR_ST_CNT_THD to 256, then accumulates 256 times.

The carrier can be calculated:

Fc = Fsys/(RO_IR_SUM_CNT0 +RO_IR_SUM_CNT1)*256;

DUT = RO_IR_SUM_CNT1/(RO_IR_SUM_CNT0 +RO_IR_SUM_CNT1);

Where Fc is the carrier frequency, Fsys is the system clock frequency, DUT is the carrier duty ratio.

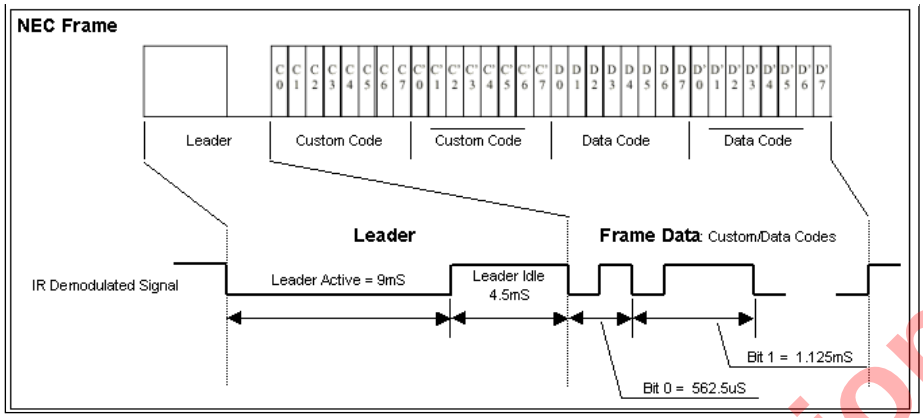
14.6.3 Legacy IR Control

The Legacy IR Remote control module has two modes of operation:

- NEC Frame decoder mode
- General Time Measurement mode

NEC Frame Decoder: The NEC Frame Decoder mode operates by analyzing the waveform of a TV

Figure 14-4 NEC Frame Decoder



remote.

The waveform has a number of components, each of which must fit within a time window to be considered valid. If the entire waveform meets the specifications described by the registers below, then the TV remote codes are captured and an interrupt is generated.

General Time Measurement: Some remotes don't follow the standard NEC format, so additional registers provide the ability to measure the time between rising and/or falling edges of the IR signal. Since the time measurement is done in hardware, the software only needs to read a "width" measurement from a register for every rising and/or falling edge event.

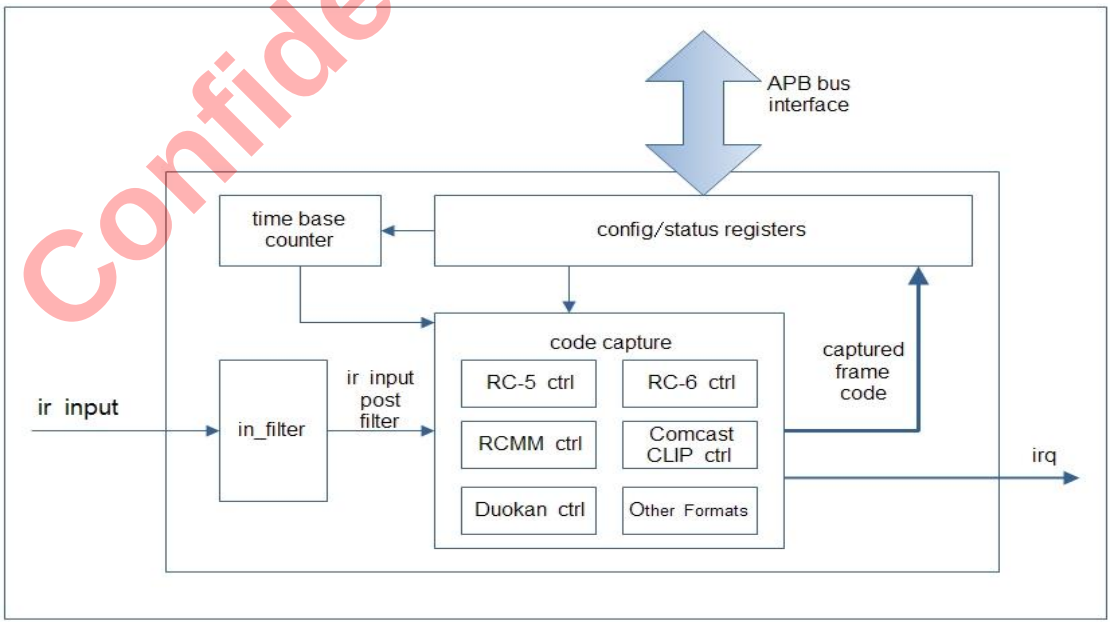
14.6.4 Multi Format IR Control

The decoder mainly consisted of two blocks:

- Decoder with input filter
- A set of registers including control & clock, data and tuning

The function diagram of IR decoder is illustrated in the figure below.

Figure 14-5 IR Decoder Function Block



IR Decoder decodes the IR remote control input signal. 13 operation modes are supported:

- Hardware Decode IR transmission protocol compatible frame decoder mode (NEC MITSUBISHI Thomson Toshiba Sony SIRC RC5 RC6 RCMM Duokan Comcast Sanyo Modes)
- General programmable time measurement frame decoder mode (General Mode)

In Hardware Decode Mode, the Decoder uses signal pattern search mechanism to decode data frame. It can detect logical ‘0’ ,‘1’ ,“00”,”01”,”10” and ”11”, as well as data frame start and end. Whenever Decoder detects and decodes the data frame, the data are kept in data register.

In General Mode, the Decoder uses edge detection mechanism to decode data frame. It can detect each input signal edge and record the time between two edges. The time measurement result is kept in control register.

The user should set proper operation mode corresponding to the selection of remote controller.

There is a simple time-based signal Filter between the signal input and the Decoder. The Filter is programmable and helps to improve signal integrity.

14.6.5 Register Description

The following Register Description is uniformly applied to all UART instantiations in the chip.

Base_adr: 0xff808000

Final_adr = base_adr + offset *4

AO_IR_DEC_DEMOD_CNTL0 0x30

Bits	R/W	Default	Description
31			Ir demodulor soft_reset:write 1 to reset ir_demodulor,will auto clr to 0
30			Reg_ir_fd_reset :It is used to reset the carrier detection module.write 1 to reset ,will auto clr to 0
29			Reg_ir_demod_mode:It is uset to set the demod mode. 0 = count mode; 1 = iir mode.default is 0
28			Ir demodulor clk gate bypass:write 1 will bypass clk gate,default is 0
27:16			Reg_ir_st_cnt_thd:It is used to set the statistics number for carrier detection.default is 0x40
15:8			Reg_ir_ds_rate:It is uset to set the downsample rate.default is 0x8
7:4			Reg_ir_fsft_1:It is used to set the shift value for input data “1”.default is 0x7
3:0			Reg_ir_fsft_0:It is used to set the shift value for input data “0”.default is 0x9

AO_IR_DEC_DEMOD_CNTL1 0x31

Bits	R/W	Default	Description
31			Reg_ir_demod_en:ir demod enable ,default is 0
30			Reg_ir_invt:ir input invert, invert input at 1.default is 0
29:16			Reg_ir_proctect1:It is used to set the protection threshold for 1. It is used to filter the glitch.default is 0xa
13:0			Reg_ir_detect0:It is used to set the detection threshold for signal “0”.default is 0x3e8

AO_IR_DEC_DEMOD_IIR_THD 0x32

Bits	R/W	Default	Description
31:16			Reg_ir_iir_thd1:It is used to set the detection threshold for “1” in iir mode.default is 0xdac
15:0			Reg_ir_iir_thd0:It is used to set the detection threshold for “0” in iir mode.default is 0xdac

AO_IR_DEC_DEMOD_THD0 0x33

Bits	R/W	Default	Description
29:16			Reg_ir_thd0_low:It is used to set the low threshold for “0” when statistics.default is 0x12c
13:0			Reg_ir_thd0_high:It is used to set the high threshold for “0” when statistics.default is 0x1770

AO_IR_DEC_DEMOD_THD1 0x34

Bits	R/W	Default	Description
29:16			Reg_ir_thd1_low:It is used to set the low threshold for “1” when statistics.default is 0x12c
13:0			Reg_ir_thd1_high:It is used to set the high threshold for “1” when statistics.default is 0x1770

AO_IR_DEC_DEMOD_SUM_CNT0 0x35

Bits	R/W	Default	Description
25:0			Ro_ir_sum_cnt0:It is used to report the sum value for the statistics data “0”.READ ONLY

AO_IR_DEC_DEMOD_SUM_CNT1 0x36

Bits	R/W	Default	Description
25:0			Ro_ir_sum_cnt1:It is used to report the sum value for the statistics data “1”.READ ONLY

AO_IR_DEC_DEMOD_CNT0 0x37

Bits	R/W	Default	Description
29:16			Ro_ir_max_cnt0:It is used to report the maximum value for the statistics data “0”.READ ONLY
13:0			Ro_ir_min_cnt0:It is used to report the minimum value for the statistics data “0”.READ ONLY

AO_IR_DEC_DEMOD_CNT1 0x38

Bits	R/W	Default	Description
29:16			Ro_ir_max_cnt1:It is used to report the maximum value for the statistics data “1”.READ ONLY
13:0			Ro_ir_min_cnt1:It is used to report the minimum value for the statistics data “1”.READ ONLY

AO_IR_DEC_LDR_ACTIVE: Leader Active Time 0x00

Bit(s)	R/W	Default	Description
31-29	R	0	unused
28-16	R/W	0x 1d8	Max Leader ACTIVE time: 9.44mS assuming base rate = 20uS
15-13	R	0	unused
12-0	R/W	0x1ac	Min Leader ACTIVE time: 8.56mS assuming base rate = 20uS

This register controls the min/max leader active time window. For most TV remote controls, the leader active time is about 9mS. The values in this register correspond to counts of the base rate programmed by register 0x2124

AO_IR_DEC_LDR_IDLE: Leader Idle Time 0x01

Bit(s)	R/W	Default	Description
31-26	R	0	unused
28-16	R/W	0xf8	Max Leader IDLE time: 4.96mS assuming base rate = 20uS
15-13	R	0	unused
12-0	R/W	0xca	Min Leader IDLE time: 4.04mS assuming base rate = 20uS

This register controls the min/max leader IDLE time window. For most TV remote controls, the leader idle time is about 4.5mS. The values in this register correspond to counts of the base rate programmed by register: 0x2124

AO_IR_DEC_LDR_REPEAT: Repeat Leader Idle Time 0x02

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0x7a	Max REPEAT Leader IDLE time: 2.44mS assuming base rate = 20uS
15-10	R	0	unused
9-0	R/W	0x66	Min REPEAT Leader IDLE time: 2.04mS assuming base rate = 20uS

This register controls the repeat leader IDLE time window. The repeat key uses the standard leader active time (9ms) but a shorter leader idle time.

AO_IR_DEC_BIT_0: BIT 0 Identification Time 0x03

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0x42	Max BIT 0 time: 1.32mS assuming base rate = 20uS
15-10	R	0	Unused
9-0	R/W	0x2e	Min BIT 0 time: 0.92mS assuming base rate = 20uS

This register controls the min/max BIT 0 time window. For most TV remote controls, the bit 0 time is about 1.125mS. The values in this register correspond to counts of the base rate programmed by register: 0x2124

AO_IR_DEC_REG0: Base Rate Generator 0x04

Bit(s)	R/W	Default	Description
31	R	0	Just in case bit. Normally this bit is set to 0 so that the auto-clock gating is enabled. If there is a problem related to the auto-clock gating, then this bit can be set to one to disable the auto-clock gating.
30-28	R/W	0	FILTER_COUNT: This is a new feature to Nike. The IR remote input now has a simple filter that accommodates slow rise times by providing a little hysteresis. The logic works as follows. If the input is low, then an input signal will only be considered HIGH if it remains high for (FILTER_COUNT * 111nS). Similarly, if the input is currently high, it will only be considered LOW if the input signal remains low for (FILTER_COUNT * 111nS).
27-25	R	0	Unused
24-12	R/w	0xFA0	Max Frame Time. 80mS assuming base rate = 20uS This value is used to determine if a code is a repeat code (e.g. leader followed by no data for this amount of time). This value can also be used to catch slow remote codes (i.e. code sequences that are longer than expected).
11-0	R/W	0x13	This value dictates the base rate time for all measurements associated with the IR decoder. In the past, the base rate was divided from the system clock. In the current design, the base rate is divided from a fixed 1uS timer. This 1uS timer is constant and doesn't change (even when the system clock does) Base rate = (count + 1) * 1uS

This register controls the master rate generator for all width measurements made by the IR decoder module.

AO_IR_DEC_FRAME: Frame Data 0x05

Bit(s)	R/W	Default	Description
31-0	R	0	Frame Data. Format: {custom code, ~custom code, data, ~data}

Note: New keys will be ignored until this register is read if the *hold first key* bit is set in the decode control register. Reading this register resets an internal hold first flag.

AO_IR_DEC_STATUS: Frame Status 0x06

Bit(s)	R/W	Default	Description
31	R/W	0	Sim faster: Reserved
30	R/W	0	BIT_1_MATCH_EN: Set this bit to 1 to enable qualification of bit 1 times. In the previous IR decoder module, frame detection only looked at the BIT 0 time to identify a zero bit. If a zero bit time wasn't found, then it was assumed that the bit was a 1. In the updated IR decoder module, the module will look at the BIT 0 time to find zero bits, and the BIT 1 time to find 1 bits. If the width of a pulse doesn't match the zero or the one bit width time, then the frame is considered invalid.
29-20	R/W	0x89	Max BIT 1 time: 2.74mS assuming base rate = 20uS
19-10	R/W	0x57	Min BIT 1 time: 1.74mS assuming base rate = 20uS

9	R	-	IRQ Status. 1 if there is an interrupt
8	R	-	IR Decoder input. This is the level of the digital signal coming into the IR module for decoding. This is the same as reading the I/O pad level.
7	R	-	BUSY. 1 if the decoder isbusy
6-4	R	0	Decoder Status: For debug only 000: OK 001: last frame timed out 010: leader time error (invalid IR signal) 011: repeat error (repeat leader, but other IR transitions found). 100: Invalid bit
3-0	R	0	Frame Status Bit 3: Frame data valid Bit 2: data code error (data != ~data in IR bit stream) Bit 1: custom code error (custom_code != ~custom_code in IR bit stream) Bit 0: 1 = repeat key, 0 = standard key

AO_IR_DEC_REG1: Decode / Interrupt Control 0x07

Bit(s)	R/W	Default	Description
31	R	0	Unused
30	R/W	0	CNTL_1uS_EQ_CLK: This bit should be set to 1 if the clk81 (system clock) is less than 50 Mhz.
29	R/W	0	CNTL_111nS_EQ_CLK: This bit should be set to 1 if the clk81 (system clock) is less than 50 Mhz.
28-16	R	0	Time measurement since the last time the internal time counter was reset by the rising and/or falling edge of the IR signal. The selection of reset on rising and/or falling edge is determined by the selection of the IRQ (Bits 3-2 below)
15	R/W	1	ENABLE: If this bit is 1, then the state-machines are enabled. If this bit is zero, then the state-machines cleanup and immediately return to idle.
14	R/W	0	USE SYSTEM CLOCK: This is a new feature. 1 = use the system clock, 0 = use the 1uS timebase tick. During normal operation, the module is setup to create a 20uS tick from the 1uS internal timebase of the chip. If the chip is configured to operate using the 32khz RTC oscillator, the 1uS timebase is invalid and therefore the 20uS timebase is invalid. In order to measure time correctly, the IR remote circuit can use the system clock (which in this case is the 32khz oscillator clock) as the master timebase.
13-9	R/W	1f	Number of bits in the IR frame (N-1)

Bit(s)	R/W	Default	Description
8-7	R/W	1f	Decoder mode 00: NEC Frames: Decode Leader and 32 bits 01: Only accumulate bits (skip the leader) 10: Measure Mode: The internal width measuring counter is reset on the rising and/or falling edge of the IR remote signal based on the settings of IRQ Selection below. Just before being reset, the measured width is captured and stored so that it can be read in bits [28:16] of this register. 11: NEC Frames: Decode Leader and 32 bits
6	R/W	1	Hold First Key. If this bit is set true, then the frame data register (0x2125) will only be updated if hasn't already been updated. Once updated, the frame data register will not be updated again until it has been read. This bit can be used to guarantee the first TV remote code captured will not be overwritten by subsequent transmissions from a TV remote. NOTE: You must read the frame data register to clear an internal hold first flag if this bit is set.
5-4	R/W	11	Frame mask. These bits are used to qualify frames for capture. 00: Capture all frames good or bad 01: Capture only frames where data=~data. Ignore custom codes 10: Capture only frames where custom_code = ~custom_code. Ignore data codes 11: Capture only frames where (data=~data) and (custom_code = ~custom_code)
3-2	R/W	0	IRQ Selection and width measurement reset: 00: IR Decoder done 01: IR input rising or falling edge detected 10: IR input falling edge detected 11: IR rising edge detected
1	R/W	0	IR Polarity. Polarity of the input signal (VD[0])
0	R/W	0	Set to 1 to reset the IR decoder. This is useful because the IR remote state machine thinks in terms of milliseconds and may take tens of milliseconds to return to idle by itself.

AO_MF_IR_DEC_LDR_ACTIVE: Leader Active control 0x10

This register controls the min/max Leader Active time window. For example, for NEC format, the Leader Active time is about 9mS. To identify a Leader Active time between 8.60 mS and 9.40 mS (assuming base resolution = 20uS), user can set Max duration = 0x1d6 ('d470) to represent 9.40 mS, and set Min duration = 0x1ae ('d430) to represent 8.60 mS.

Bit(s)	R/W	Default	Description
31-29	R	0	Unused
28-16	R/W	0	Max duration of Leader's active part
15-13	R	0	Unused
12-0	R/W	0	Min duration of Leader's active part

AO_MF_IR_DEC_LDR_IDLE: Leader Idle control 0x11

Bit(s)	R/W	Default	Description

31-29	R	0	Unused
28-16	R/W	0	Max duration of Leader's idle part
15-13	R	0	Unused
12-0	R/W	0	Min duration of Leader's idle part

AO_MF_IR_DEC_LDR_REPEAT: Repeat Leader Idle Time 0x12

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0	Max duration of Repeat Code's Leader. In NECformat, it defines for the repeat leader's idle part. In Toshiba format, it defines for the repeat leader's second idle part (In Toshiba format, the repeat leader's first idle part has the same duration time as the normal leader idle part.)
15-10	R	0	Unused
9-0	R/W	0	Min duration of Repeat Code's Leader

AO_MF_IR_DEC_BIT_0 0x13

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0	Max duration of Duration Setting Regiser 0. It defines max tining duration for: Logic"0" for NEC/Toshiba/Sony/Thomas format or Half trailer bit for RC6 format (RC6's half trailer bit typically 888.89us) or time of Duokan/RCMM/4ppm format's Logic "00"
15-10	R	0	Unused
9-0	R/W	0	Min duration of Duration Setting Regiser 0.

AO_MF_IR_DEC_REG0 0x14

Bit(s)	R/W	Default	Description
31	R/W	0	Clock gating control just in case. Set 1 can force clock gating disabled.
30-28	R/W	0	Filter ctrl. Set the monitor timing for input filter, bigger value means longer monitor time. Value 0 = no filtering.
27-25	R	0	Unused

24-12	R/W	0	Max frame time. Max duration of one whole frame.
11-0	R/W	0	Base time parameter. Used to generate the timing resolution. Resolution = (base_time_paramter + 1) * (1/ Freq_sys_clk). For example, if Frequency of sys_clk is 1Mhz, and base_time_parameter=19, Then resolution = (19+1)*(1uS) = 20uS.

AO_MF_IR_DEC_STATUS 0x16

Bit(s)	R/W	Default	Description
31	R/W	0	Frame data valid 1. (This bit is set to 1 when a captured frame is updated/stored into “FrameBody_1” register. A read of “FrameBody_1” register will clear This bit. “FrameBody_1” register is used to store the over 32bit MSBs of the formats whose length is more than 32 bit)
30	R/W	0	bit_1_match_en. Set to 1 to enable the check of whether logic”1” bit matches timing configure during the frame input process.
29-20	R/W	0	Max Duration 1. Max duration of Duration Setting Regiser 1. It defines max duration for: Logic”1” for NEC/Toshiba/Sony format or Whole trailer bit for RC6 format (RC6’s whole trailer bit typically 1777.78us) or time of Duokan/RCMM/4ppm format’s Logic “01”
19-10	R/W	0	Min Duration 1. Min duration of Duration Setting Regiser 1.
9	R	0	irq_status. Appear as 1 if there is an interrupt.
8	R	0	ir_i_sync. IR remote serial input after synchronization. This is the level of the digital signal coming into the IR module for decoding. This is the same as reading the I/O pad level.
7	R	0	Busy. When =1, means state machine is active.
6-4	R	0	Decoder_status (for debug only). 000: OK 001: last frame timed out 010: leader time error (invalid IR signal) 011: repeat error (repeat leader, but other IR transitions found). 100: Invalid bit
3-0	R	0	Frame status. bit 3: Frame data valid (This bit is set to 1 when a captured frame is updated/stored into “FrameBody” register. A read of “FrameBody” register will clear This bit.

Bit(s)	R/W	Default	Description
			If store and read occurs at the same time, This bit is set to 1 in common, But if “Hold first” is set to true and this valid Bit is already 1, a read clear takes precedance and This bit is clear to 0.) bit 2: data code error (data != ~data in IR bit stream) bit 1: custom code error (custom_code != ~custom_code in IR bit stream) bit 0: 1 = received frame is repeat key, 0 = received frame is normal key

AO_MF_IR_DEC_REG1 0x17

Bit(s)	R/W	Default	Description
31	R/W	0	Set to 1 to use faster timebase. .
30	R/W	0	cntl_1us_eq_clk. Just use sys_clk to relace 1uS tick.
29	R/W	0	cntl_xtal3_eq_clk. Just use sys_clk to relace 111ns tick.
28-16	R	0	Pulse Width Counter. It stores the internal counter of pulse width duration. Commonly used as time measurement when decode_mode is set to measure width mode (software decode). Time measurement starts at the last time the internal time counter was reset by the rising and/or falling edge of the IR signal. The selection of reset on rising and/or falling edge is determined by the IRQ Selection field (Bits 3-2 below)
15	R/W	0	Enable. 1 = enable the state machine of IR decoder. 0 = disable the state machine of IR decoder.
14	R/W	0	cntl_use_sys_clk. Use sys_clk for the timebase. It’s useful when sys_clk at low frequency (such as 32Khz) and cannot create 1uS timebase tick. 1 = use the system clock as timebase. 0 = use the 1uS timebase tick as timebase.
13-8	R/W	0	bit_length minus 1. (N-1). Used to set the value of frame body’s bit length (frame body commonly includes address and data code part). If a format has 24 bit frame body, this value shall be set to 23.
7	R/W	0	Record_at_error. 1= record the frame body and status forcibly, even if data/custom code error check enabled by frame_mask and relative error occurs. 0 = if data/custom code error check enabled by frame_mask and relative error occurs, not record the frame body and status forcibly
6	R/W	0	Hold_first Used to hold the first captured frame data. If This bit is set to 1, then the “FrameBody/FrameBody_1” register will only be updated if hasn’t already been updated. Once updated, the “FrameBody/FrameBody_1” register will not be updated again until it has been read. This bit can be used to guarantee the first TV remote code captured will not be overwritten by subsequent transmissions from a TV remote. NOTE: Read the”FrameBody” register can clear the internal “Frame data valid” flag, and read the “FrameBody_1” register can clear the “Frame data valid 1” flag.

Bit(s)	R/W	Default	Description
5-4	R/W	0	Frame_mask. Some formats' body include bit-inversed data or custom/address code for error check. 00 = ignore error check from either data or custom/address code 01= check if data code matches its inverse values, ignore error check from custom/address code 10= check if custom/address code matches its inverse values, ignore error check from data code 11= check if data and custom codes match their inverse values
3-2	R/W	0	Irq_sel. IRQ Selection and width measurement reset: 00: IR Decoder done 01: IR input rising or falling edge detected 10: IR input falling edge detected 11: IR rising edge detected
1	R/W	0	IR input polarity selection. Used to adjust/invert the polarity of IR input waveform.
0	R/W	0	Decoder Reset. Set to 1 to reset the IR decoder. This is useful because the IR remote state machine thinks in terms of milliseconds and may take tens of milliseconds to return to idle by itself.

AO_MF_IR_DEC_REG2 0x18

Bit(s)	R/W	Default	Description
31-27	R	0	Unused
26	R/W	0	Width_low_enable. Enable counter record of low pulse width duration. 0 = do not force enable of width low counter record 1 = force enable of width low counter record Some IR formats' decoding need to use internal width low counter record. By default, the width low counter record is enabled automatically for related formats. This bit is used for enable forcibly just in case. Besides, if "leader plus stop bit" method is enabled for repeat detection, This bit is also need to be enabled.
25	R/W	0	Width_high_enable. Enable counter record of high pulse width duration. 0 = do not force enable of width high counter record 1 = force enable of width high counter record Some IR formats' decoding need to use internal width high counter record. By default, the width high counter record is enabled automatically for related formats. This bit is used for enable forcibly just in case.

Bit(s)	R/W	Default	Description
24	R/W	0	Enable “leader plus stop bit” method for repeat detection. 0 = “leader plus stop bit” method disabled 1 = “leader plus stop bit” method enabled Some IR formats use one normal frame's leader followed by a stop bit to represet repeat. There is no frame data in this kind repeat frame. To use this method, width_low_enable (Bit 26 of 0x20 offset register) shall be set to 1, and max_duration_3 and min_duration_3 in 0x28 offset register shall be set to appropriate value for stop bit's timing duration.
23-22	R	0	Unused
21-16	R/W	0	Repeat_Bit_index. These Bits are used for compare bit method to set the index of the bit that is used as repeat flag. The index value can be 0 to 63. Compare bit method is one of the methods for repeat detection . Some IR formats use one bit in frame to represet whether the frame is repeat.
15	R/W	0	Running_count_tick_mode. This bit is only valid when use_clock_to_counter Bit is 0. 0 = use 100uS as increasing time unit of frame-to-frame counter 1 = use 10uS as increasing time unit of frame-to-frame counter
14	R/W	0	Use_clock_to_counter. If This bit is set to 1, the running_count_tick_mode Bit is ignored. 0 = do not use system clock as increasing time unit of frame-to-frame counter 1 = use system clock as increasing time unit of frame-to-frame counter
13	R/W	0	Enable frame-to-frame time counter (running-counter). 0 = frame-to-frame time counter disabled 1 = frame-to-frame time counter enabled If enabled, the frame-to-frame counter increases every 100uS or 10uS until it reaches its max value(all Bits are 1) or it is reset. When it reaches its max value, it keeps the value until it is reset. When it is reset, it becomes zero and then begin increasing again. The counter can be reset even when it has not reached its max value. The increasing time unit can be 100uS or 10uS or system clock frequency which is set by running_count_tick_mode and use_clock_to_counter settings. When a frame’s data are capured and stored into FrameBody/FrameBody_1 register, frame-to-frame counter is reset to zero. After reset to zero, the frame-to-frame counter will begin increasing again, until it reaches its max value or it is reset. For repeat frame detection, users can use hardware detection by enabling compare frame or compare bit method, or users can read frame-to-frame counter to let software to make the decision.

Bit(s)	R/W	Default	Description
12	R/W	0	Enable repeat time check for repeat detection. This bit is valid only when compare frame method or compare Bit method is enabled. 0 = repeat time check disabled 1 = repeat time check enabled When repeat frame detection is enabled by enabling compare frame or compare Bit method, the frame time interval may need to be checked in order to decide whether the frames are repeat (key pressed without release) or not. You can configure the repeat_time_max value by setting 0x38 offset register. If frame interval is smaller than the “repeat time max”, it may considered as repeat. If frame interval is bigger than the “repeat time max”, it is considered as not repeat.
11	R/W	0	Enable compare frame method for repeat detection. 0 = compare frame method disabled 1 = compare frame method enabled Some IR formats transer the same data frame as repeat frame when the key is kept pressed without release. For repeat detection, compare frame method can be used. If a new frame and the old received frame are the same and the repeat time is under the limit(frame-to-frame time counter value is smaller than the repeat_time_max), the status register's frame_status0 is set to 1 automatically as repeat detected flag. You can configure the repeat_time_max value by setting 0x38 offset register.
10	R/W	0	Enable compare Bit method for repeat detection. 0 = compare Bit method disabled 1 = compare Bit method enabled Some IR formats use only one bit to represet whether the frame is repeat. You can compare only one bit instead of compare the whole frame for repeat detection. If compare frame method is enabled, then This bit is ignored.
9	R/W	0	Disable read-clear of FrameBody/FrameBody_1. 0 = read-clear enabled 1 = read-clear disabled FrameBody/FrameBody_1 registers are read-cleared in default. When these register are read, they are cleared to zero. This bit is used to disable this read-clear feature. (FrameBody/FrameBody_1 registers are used to store captured frame data).
8	R/W	0	input stream bit order. 0 = LSB first mode (first bit in input stream is considered as LSB) 1= MSB first mode (first bit in input stream is considered as MSB) Note: Commonly the following formats shall set 1 to enable MSB first mode (unless you insist on LSBfirst mode for your specified use): RC5, RC5 extend, RC6, RCMM, Duokan, Comcast
7:4	R	0	Unused

Bit(s)	R/W	Default	Description
3:0	R/W	0	Decode_mode.(format selection) 0x0 =NEC 0x1= skip leader (just Bits, without leader) 0x2=General time measurement (measure width, software decode) 0x3=MITSUBISHI 0x4=Thomson 0x5=Toshiba 0x6=Sony SIRC 0x7=RC5 0x8=Reserved 0x9=RC6 0xA=RCMM 0xB=Duokan 0xC=Reserved 0xD=Reserved 0xE=Comcast 0xF=Sanyo

AO_MF_IR_DEC_DURATN2 0x19

Bit(s)	R/W	Default	Description
31-26	R	0	Unused
25-16	R/W	0	Max duration of Duration Setting Regiser 2. It defines max duration for: Half bit for RC5/6 format (RC5 typically 888.89us for half bit, RC6 typically 444.44us) or time of Duokan/RCMM/4ppm format's Logic "10" or time of Comcast/16ppm's base duration
15-10	R	0	Unused
9-0	R/W	0	Min duration of Duration Setting Regiser 2.

AO_MF_IR_DEC_DURATN3 0x1a

Bit(s)	R/W	Default	Description
31-26	R	0	Unused

25-16	R/W	0	Max duration of Duration Setting Regiser 3. It defines max duration for: Whole bit for RC5/6 format (RC5 typically 1777.78us for whole bit, RC6 typically 888.89us) or time of Duokan/RCMM/4ppm format's Logic "11" or time of Comcast/16ppm's offset duration
15-10	R	0	Unused
9-0	R/W	0	Min duration of Duration Setting Regiser 3.

AO_MF_IR_DEC_FRAME: Frame Body (Frame Data, LSB 32Bit) 0x1b

Note: New keys will be ignored until ***FrameBody*** register is read if the *hold first key* Bit is set in the decode control register. Reading this register resets an internal frame data valid flag.

Bit(s)	R/W	Default	Description
31-0	R	0	32 bit Read-Only register stores frame body (LSB 32 bit) captured from IR remote data flow, commonly includes custom/address code and data code.

AO_MF_IR_DEC_FRAME: Frame Body 1 (Frame Data, MSB 32Bit) 0x1b

Note: New keys will be ignored until ***FrameBody*** register is read if the *hold first key* Bit is set in the decode control register. Reading this register resets an internal frame data valid flag.

Bit(s)	R/W	Default	Description
31-0	R	0	Stores frame body excess 32 bit range. (MSB 32 bit)

AO_MF_IR_DEC_STATUS_1 0x1c

Bit(s)	R/W	Default	Description
31-20	R	0	Unused
19-0	R	0	Stores the last frame-to-frame counter value before the last counter reset caused by the last frame data record/update.

AO_MF_IR_DEC_STATUS_2 0x1d

Bit(s)	R/W	Default	Description
31-20	R	0	Unused
19-0	R	0	Stores the value of the frame-to-frame counter which is running currently.

14.7 Pulse-Width Modulation

14.7.1 Overview

The chip has 5 PWM modules that can be connected to various digital I/O pins, among which 3 are in EE domain and 2 is in AO domain. Each PWM is driven by a programmable divider driven by a 4:1 clock selector. The PWM signal is generated using two 16-bit counters. One is the High and Low counter, which is individually programmable with values between 1 and 65535. Using a combination of the divided clock (divide by N) and the HIGH and LOW counters, a wide number of PWM configurations are possible. The other is delta-sigma couter, generate 18-bit sigma, the PWM-out is the highest sigma. The PWM outputs vs counters are also illustrate below.

Figure 14-6 PWM Block Diagram

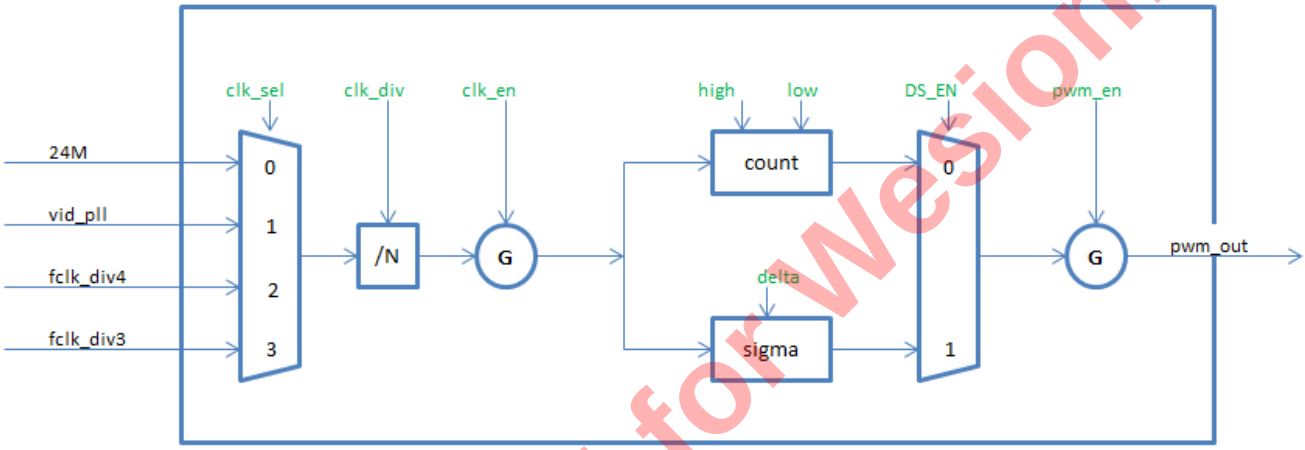


Figure 14-7 High/Low counter

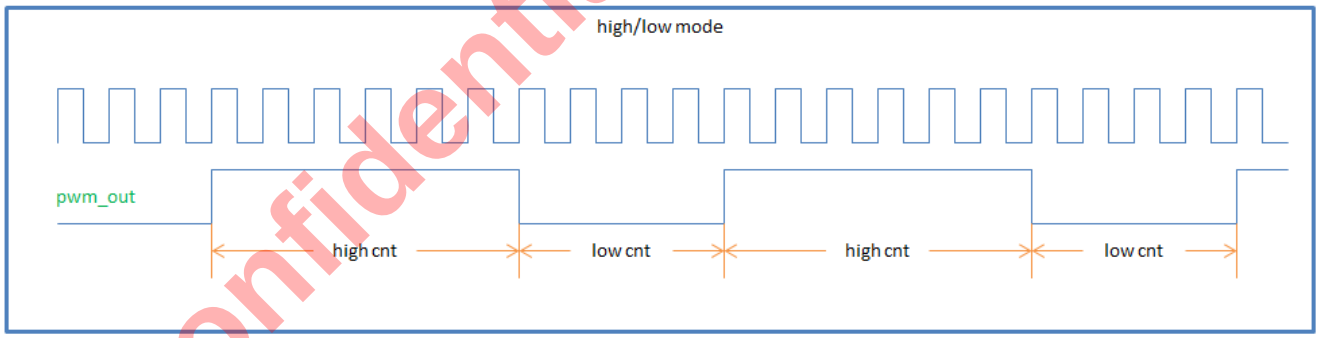
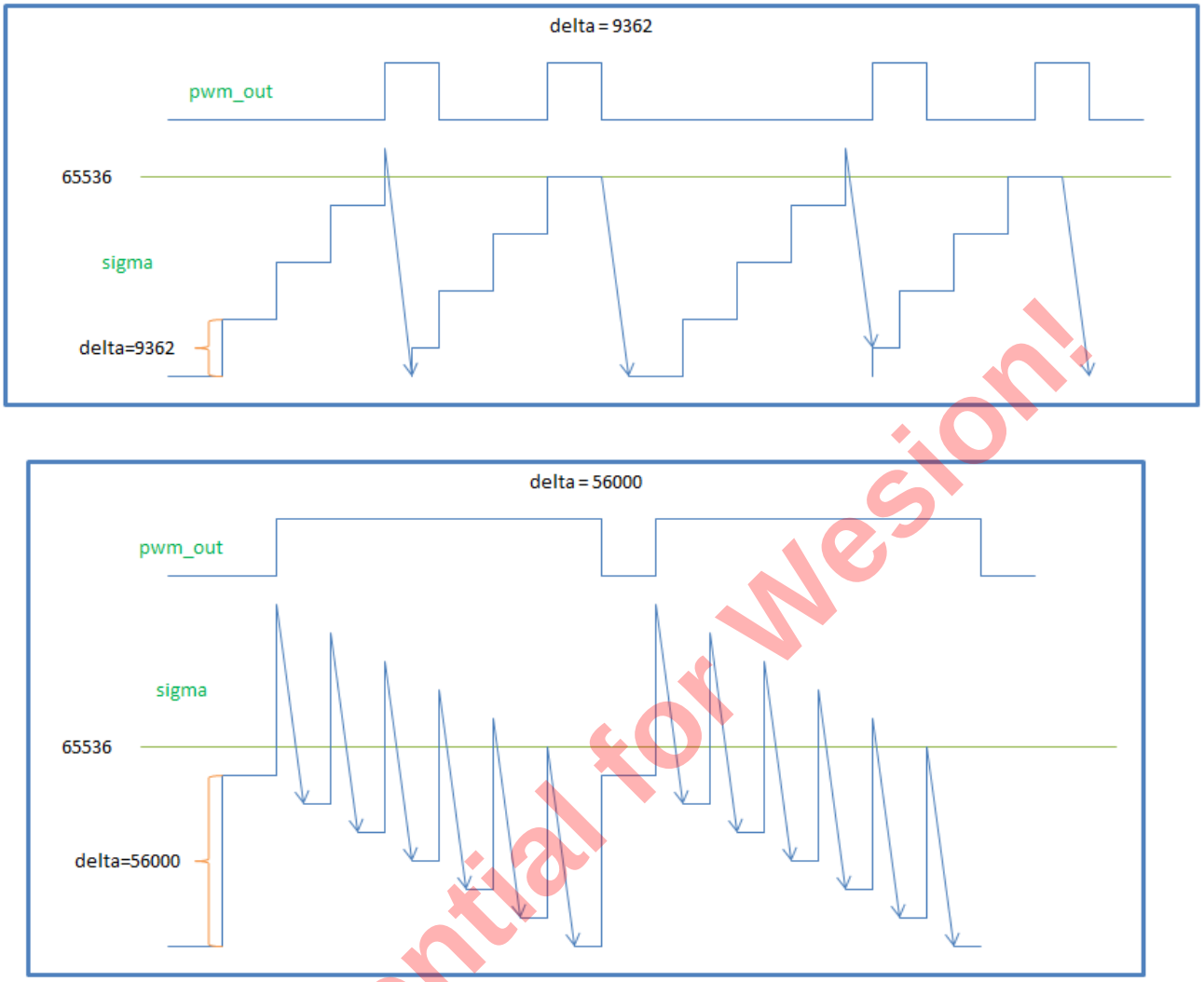
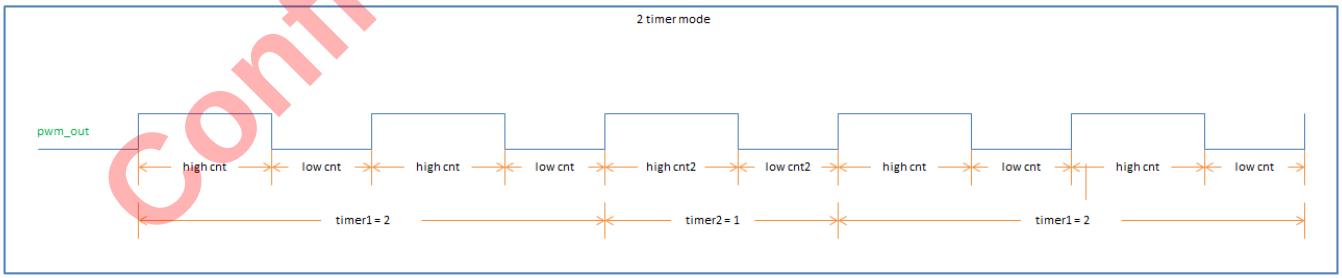


Figure 14-8 Delta-sigma conter, delta = 9362 & delta = 56000



PWM 2 timer mode is illustrated as following:

Figure 14-9 2 timer mode



14.7.2 Register Description

Each PWM module contains two PWM generators call A and B and controlled by the following registers.

For PWM modules in EE domain, the each register's final address =0xffd00000 + offset*4

PWM_PWM_A 0x6c00

Bit(s)	R/W	Default	Description

31-15	R/W	0	PWM_A_HIGH: This sets the high time (in clock counts) for the PWM_A generator output
15-0	R/W	0	PWM_A_LOW: This sets the high time (in clock counts) for the PWM_A generator output

PWM_PWM_B 0x6c01

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_B_HIGH: This sets the high time (in clock counts) for the PWM_B generator output
15-0	R/W	0	PWM_B_LOW: This sets the high time (in clock counts) for the PWM_B generator output

PWM_MISC_REG_AB 0x6c02

Bit(s)	R/W	Default	Description
31	R/W	0	pwm_B_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
30	R/W	0	pwm_A_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
29	R/W	0	pwm_B_constant_en
28	R/W	0	pwm_A_constant_en
27	R/W	0	pwm_B_inv_en
26	R/W	0	pwm_A_inv_en
25	R/W	0	cntl_pwm_a2_en
24	R/W	0	cntl_pwm_b2_en
23	R/W	0	PWM_B_CLK_EN: Set this bit to 1 to enable PWM B clock
22-16	R/W	0	PWM_B_CLK_DIV: Selects the divider (N+1) for the PWM B clock. See the clock tress document
15	R/W	0	PWM_A_CLK_EN: Set this bit to 1 to enable PWM A clock
14-8	R/W	0	PWM_A_CLK_DIV: Selects the divider (N+1) for the PWM A clock. See the clock tress document
7-6	R/W	0	PWM_B_CLK_SEL: Select the clock for the PWM B. See the clock tress document
5-4	R/W	0	PWM_A_CLK_SEL: Select the clock for the PWM A. See the clock tress document
3	R/W	0	DS_B_EN: This bit is only valid if PWM_B_EN is 0: if this bit is set to 1, then the PWM_B output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_B output is set low.
2	R/W	0	DS_A_EN: This bit is only valid if PWM_A_EN is 0: if this bit is set to 1, then the PWM_A output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_A output is set low.
1	R/W	0	PWM_B_EN: If this bit is set to 1, then the PWM_B output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_B output is controlled by DS_B_EN above.

Bit(s)	R/W	Default	Description
0	R/W	0	PWM_A_EN: If this bit is set to 1, then the PWM_A output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_A output is controlled by DS_A_EN above.

DS_A_B 0x6c03

Bit(s)	R/W	Default	Description
31-15	R/W	0	DS_B_VAL: This value represents the delta sigma setting for channel B (PWM_B)
15-0	R/W	0	DS_A_VAL: This value represents the delta sigma setting for channel A (PWM_A)

PWM_TIME_AB 0x6c04

Bit(s)	R/W	Default	Description
31-24	R/W	0	A_timer1
23:16	R/W	0	A_timer2
15:8	R/W	0	B_timer1
7:0	R/W	0	B_timer2

PWM_A2 0x6c05

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_A2_HIGH: This sets the high time (in clock counts) for the PWM_A2 generator output
15-0	R/W	0	PWM_A2_LOW: This sets the high time (in clock counts) for the PWM_A2 generator output

PWM_B2 0x6c06

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_B2_HIGH: This sets the high time (in clock counts) for the PWM_B2 generator output
15-0	R/W	0	PWM_B2_LOW: This sets the high time (in clock counts) for the PWM_B2 generator output

PWM_BLINK_AB 0x6c07

Bit(s)	R/W	Default	Description
31-10	R	0	Reserved
9	R/W	0	blink enable for pwm B
8	R/W	0	blink enable for pwm A
7-4	R/W	0	blink times for pwm B
3-0	R/W	0	blink times for pwm A

PWM_PWM_C_D: 0x6800~0x6807

See the registers for PWM A/B

PWM_PWM_E_F: 0x6400~0x6407

See the registers for PWM A/B

AO PWM' clock sources are xtal, clk81, fclk_div3, fclk_div4.

For the following register, each register's final address =0xFF807000 + offset*4

AO_PWM_PWM_A: PWM_A_DUTY_CYCLE 0x0

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_A_HIGH: This sets the high time (in clock counts) for the PWM_A generator output
15-0	R/W	0	PWM_A_LOW: This sets the high time (in clock counts) for the PWM_A generator output

AO_PWM_PWM_B: PWM_B_DUTY_CYCLE 0x1

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_B_HIGH: This sets the high time (in clock counts) for the PWM_B generator output
15-0	R/W	0	PWM_B_LOW: This sets the high time (in clock counts) for the PWM_B generator output

AO_PWM_MISC_REG_AB: 0x2

Bit(s)	R/W	Default	Description
31	R/W	0	pwm_B_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
30	R/W	0	pwm_A_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
29	R/W	0	pwm_B_constant_en set this bit to 1, then pwm can support 0%(100%) duty output
28	R/W	0	pwm_A_constant_en set this bit to 1, then pwm can support 0%(100%) duty output
27	R/W	0	pwm_B_inv_en set this bit to 1, pwm output is inverted
26	R/W	0	pwm_A_inv_en set this bit to 1, pwm output is inverted
25	R/W	0	Pwm_a2_en
24	R/W	0	Pwm_b2_en
23	R/W	0	PWM_B_CLK_EN: Set this bit to 1 to enable PWM B clock
22-16	R/W	0	PWM_B_CLK_DIV: Selects the divider (N+1) for the PWM B clock. See the clock tress document
15	R/W	0	PWM_A_CLK_EN: Set this bit to 1 to enable PWM A clock
14-8	R/W	0	PWM_A_CLK_DIV: Selects the divider (N+1) for the PWM A clock. See the clock tress document

Bit(s)	R/W	Default	Description
7-6	R/W	0	PWM_B_CLK_SEL: Select the clock for the PWM B. See the clock tress document
5-4	R/W	0	PWM_A_CLK_SEL: Select the clock for the PWM A. See the clock tress document
3	R/W	0	DS_B_EN: This bit is only valid if PWM_B_EN is 0: if this bit is set to 1, then the PWM_B output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_B output is set low.
2	R/W	0	DS_A_EN: This bit is only valid if PWM_A_EN is 0: if this bit is set to 1, then the PWM_A output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_A output is set low.
1	R/W	0	PWM_B_EN: If this bit is set to 1, then the PWM_B output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_B output is controlled by DS_B_EN above.
0	R/W	0	PWM_A_EN: If this bit is set to 1, then the PWM_A output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_A output is controlled by DS_A_EN above.

AO_PWM_DELTA_SIGMA_AB 0x3

Bit(s)	R/W	Default	Description
31-15	R/W	0	DS_B_VAL: This value represents the delta sigma setting for channel B (PWM_B)
15-0	R/W	0	DS_A_VAL: This value represents the delta sigma setting for channel A (PWM_A)

AO_PWM_TIME_AB 0x4

Bit(s)	R/W	Default	Description
31-24	R/W	0	A1_timer
23:16	R/W	0	A2_timer
15:8	R/W	0	B1_timer
7:0	R/W	0	B2_timer

AO_PWM_A2 0x5

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_A2_HIGH: This sets the high time (in clock counts) for the PWM_A2 generator output
15-0	R/W	0	PWM_A2_LOW: This sets the high time (in clock counts) for the PWM_A2 generator output

AO_PWM_B2 0x6

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_B2_HIGH: This sets the high time (in clock counts) for the PWM_B2 generator output
15-0	R/W	0	PWM_B2_LOW: This sets the high time (in clock counts) for the PWM_B2 generator output

AO_PWM_BLINK_AB 0x7

Bit(s)	R/W	Default	Description
31-10	R	0	Reserved
9	R/W	0	blink enable for pwm B
8	R/W	0	blink enable for pwm A
7-4	R/W	0	blink times for pwm B
3-0	R/W	0	blink times for pwm A

For the following register, each register's final address= 0xFF802000 + offset * 4

AO_PWM_PWM_C: PWM_C_DUTY_CYCLE 0x0

This is a new module to Nike. It replaces the older delta sigma (PWM like) generator in the HIU. This module allows the software to select either a PWM or delta-sigma output using the same module. There are two outputs: PWM_C and PWM_D. Either of these can be programmed to be PWM outputs or delta sigma outputs.

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_C_HIGH: This sets the high time (in clock counts) for the PWM_C generator output
15-0	R/W	0	PWM_C_LOW: This sets the high time (in clock counts) for the PWM_C generator output

AO_PWM_PWM_D: PWM_D_DUTY_CYCLE 0x1

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_D_HIGH: This sets the high time (in clock counts) for the PWM_D generator output
15-0	R/W	0	PWM_D_LOW: This sets the high time (in clock counts) for the PWM_D generator output

AO_PWM_MISC_REG_CD: 0x2

Bit(s)	R/W	Default	Description
31	R/W	0	pwm_D_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
30	R/W	0	pwm_C_hiz when hiz mode, pwm_o will connect to pad_oe, then pad_o will equal this bit
29	R/W	0	pwm_D_constant_en set this bit to 1, then pwm can support 0%(100%) duty output
28	R/W	0	pwm_C_constant_en set this bit to 1, then pwm can support 0%(100%) duty output
27	R/W	0	pwm_D_inv_en set this bit to 1, pwm output is inverted

Bit(s)	R/W	Default	Description
26	R/W	0	pwm_C_inv_en set this bit to 1, pwm output is inverted
25	R/W	0	Pwm_C2_en
24	R/W	0	Pwm_D2_en
23	R/W	0	PWM_D_CLK_EN: Set this bit to 1 to enable PWM Dclock
22-16	R/W	0	PWM_D_CLK_DIV: Selects the divider (N+1) for the PWM Dclock. See the clock tress document
15	R/W	0	PWM_C_CLK_EN: Set this bit to 1 to enable PWM C clock
14-8	R/W	0	PWM_C_CLK_DIV: Selects the divider (N+1) for the PWM C clock. See the clock tress document
7-6	R/W	0	PWM_D_CLK_SEL: Select the clock for the PWM D. See the clock tress document
5-4	R/W	0	PWM_C_CLK_SEL: Select the clock for the PWM C. See the clock tress document
3	R/W	0	DS_D_EN: This bit is only valid if PWM_D_EN is 0: if this bit is set to 1, then the PWM_D output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_D output is set low.
2	R/W	0	DS_C_EN: This bit is only valid if PWM_C_EN is 0: if this bit is set to 1, then the PWM_C output is configured to generate a delta sigma output based on the settings in the register below. If this bit is set to 0, then the PWM_C output is set low.
1	R/W	0	PWM_D_EN: If this bit is set to 1, then the PWM_D output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_D output is controlled by DS_D_EN above.
0	R/W	0	PWM_C_EN: If this bit is set to 1, then the PWM_C output is configured to generate a PWM output based on the register above. If this bit is 0, then the PWM_C output is controlled by DS_C_EN above.

AO_PWM_DELTA_SIGMA_CD 0x3

Bit(s)	R/W	Default	Description
31-15	R/W	0	DS_D_VAL: This value represents the delta sigma setting for channel D (PWM_D)
15-0	R/W	0	DS_C_VAL: This value represents the delta sigma setting for channel C (PWM_C)

AO_PWM_TIME_CD 0x4

Bit(s)	R/W	Default	Description
31-24	R/W	0	C1_timer
23:16	R/W	0	C2_timer
15:8	R/W	0	D1_timer
7:0	R/W	0	D2_timer

AO_PWM_C2 0x5

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_C2_HIGH: This sets the high time (in clock counts) for the PWM_C generator output
15-0	R/W	0	PWM_C2_LOW: This sets the high time (in clock counts) for the PWM_C generator output

AO_PWM_D2 0x6

Bit(s)	R/W	Default	Description
31-15	R/W	0	PWM_D2_HIGH: This sets the high time (in clock counts) for the PWM_D generator output
15-0	R/W	0	PWM_D2_LOW: This sets the high time (in clock counts) for the PWM_D generator output

AO_PWM_BLINK_CD 0x7

Bit(s)	R/W	Default	Description
31-10	R	0	Reserved
9	R/W	0	blink enable for pwm D
8	R/W	0	blink enable for pwm C
7-4	R/W	0	blink times for pwm D
3-0	R/W	0	blink times for pwm C

14.8 SAR ADC

14.8.1 Overview

This SAR ADC is a general purpose ADC for measuring analog signals. The module can make RAW ADC measurements or average a number of measurements to introduce filtering. The SAR ADC is a single block so an analog mux is placed in front to allow multiple different measurements to be made sequentially. Timing of the samples, and delays between muxing are all programmable as is the averaging to be applied to the SAR ADC.

14.8.2 Register Description

Each register final address = 0xff809000 + offset * 4

SAR_ADC_REG0: Control Register #0 0x80

Bit(s)	R/W	Default	Description
31	R	0	PANEL DETECT level.
30	R/W	0	DELTA_BUSY: If This bit is 1, then it indicates the delta processing engine is busy
29	R/W	0	AVG_BUSY: If This bit is 1, then it indicates the averaging engine is busy
28	R/W	0	SAMPLE_BUSY: If This bit is 1, then it indicates the sampling engine is busy
27	R/W	0	FIFO_FULL:
26	R/W	0	FIFO_EMPTY:
25-21	R/W	4	FIFO_COUNT: Current count of samples in the acquisition FIFO
20-19	R/W	0	ADC_BIAS_CTRL
18-16	R/W	0	CURR_CHAN_ID: These Bits represent the current channel (0..7) that is being sampled.
15	R/W	0	ADC_TEMP_SEN_SEL
14	R/W	0	SAMPLING_STOP: This bit can be used to cleanly stop the sampling process in the event that continuous sampling is enabled. To stop sampling, simply set This bit and wait for all processing modules to no longer indicate that they are busy.
13-12	R/W	0	CHAN_DELTA_EN: There are two Bits corresponding to Channels 0 and 1. Channel 0 and channel 1 can be individually enabled to take advantage of the delta processing module.
11	R/W	0	Unused
10	R/W	0	DETECT_IRQ_POL: This bit sets the polarity of the detect signal. The detect signal is used during X/Y panel applications to detect if the panel is touched
9	R/W	0	DETECT_IRQ_EN: If This bit is set to 1, then an interrupt will be generated if the DETECT signal is low/high. The polarity is set in the bit above.
8-4	R/W	0	FIFO_CNT_IRQ: When the FIFO contains N samples, then generate an interrupt (if bit 3 is set below).
3	R/W	0	FIFO_IRQ_EN: Set This bit to 1 to enable an IRQ when the acquisition FIFO reaches a certain level.
2	W	0	SAMPLE_START: This bit should be written to 1 to start sampling.

Bit(s)	R/W	Default	Description
1	R/W	0	CONTINUOUS_EN: If This bit is set to 1, then the channel list will be continually processed
0	R/W	0	SAMPLING_ENABLE: Setting This bit to '1' enables the touch panel controller sampling engine, averaging module, XY processing engine and the FIFO.

SAR_ADC_CHAN_LIST:Channel List 0x81

Bit(s)	R/W	Default	Description
31-27	R/W	0	unused
26-24	R/W	2	Length of the list of channels to process. If this value is 2, then only channels in Bits [8:0] below are processed.
23-21	R/W	7	8th channel
20-18	R/W	6	7th channel
17-15	R/W	5	6th channel
14-12	R/W	4	5th channel
11-9	R/W	3	4th channel
8-6	R/W	2	3rd channel
5-3	R/W	1	2nd channel
2-0	R/W	0	First channel in the list of channels to process

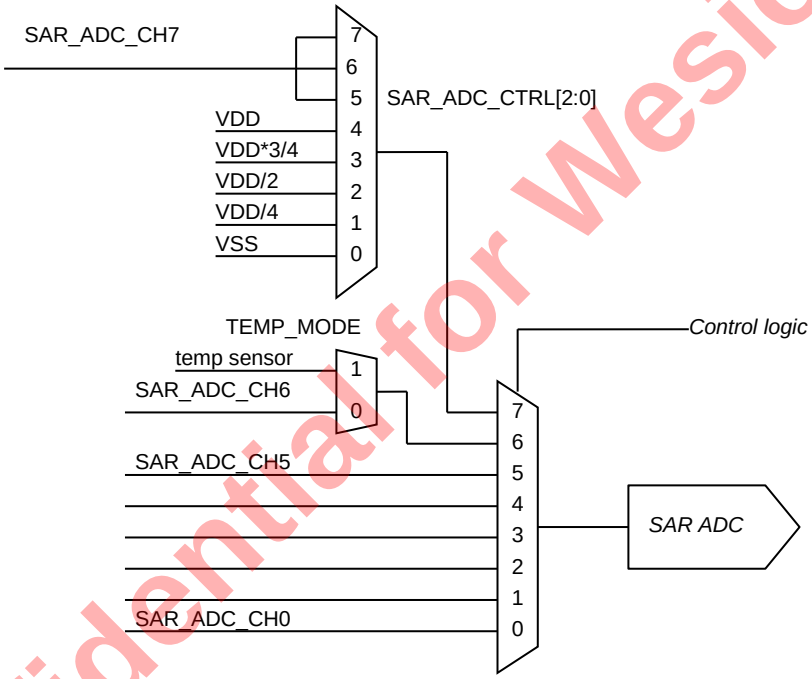
SAR_ADC_AVG_CNTL:Sampling/Averaging Modes 0x82

Each channel listed in the CHANNEL_LIST is given independent control of the number of samples to acquire and averaging mode

Bit(s)	R/W	Default	Description
31-30	R/W	0	Channel 7: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
29-28	R/W	0	Channel 6: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
27-26	R/W	0	Channel 5: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
25-24	R/W	0	Channel 4: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
23-22	R/W	0	Channel 3: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
21-20	R/W	0	Channel 2: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
19-18	R/W	0	Channel 1: Averaging mode: 0 = no averaging, 1 = simple averaging, 2 = median averaging.
17-16	R/W	0	Channel 0: Averaging mode: 0 = no averaging. 1 = simple averaging of the number of samples acquired (1,2,4 or 8). 2 = median averaging. NOTE: If these Bits are set to 2, then you must set the number of samples to acquire below to 8.
15-13	R/W	0	Channel 7: Number of samples to acquire 2N:
13-12	R/W	0	Channel 6: Number of samples to acquire 2N:

Bit(s)	R/W	Default	Description
11-10	R/W	0	Channel 5: Number of samples to acquire 2N:
9-8	R/W	0	Channel 4: Number of samples to acquire 2N:
7-6	R/W	0	Channel 3: Number of samples to acquire 2N:
5-4	R/W	0	Channel 2: Number of samples to acquire 2N:
3-2	R/W	0	Channel 1: Number of samples to acquire 2N:
1-0	R/W	0	Channel 0: Number of samples to acquire 2N: 0 = 1, 1 = 2, 2 = 4, 4 = 8.

SAR_ADC_REG3: Control Register #3 0x83



Bit(s)	R/W	Default	Description
31	R/W	0	CNTL_USE_SC_DLY: hold time delay was added to the start conversion clock. Unfortunately, it appears that the analog ADC design requires that we use the inverted clock so This bit is meaningless.
30	R/W	0	SAR ADC_CLK_EN: 1 = enable the SAR ADC clock
29	R/W	0	reserved
28	R/W	0	reserved
27	R/W	0	SARADC_CTRL[4]: is used to control the internal ring counter. 1 = enable the continuous ring counter. 0 = disable
26	R/W	0	SARADC_CTRL[3]: used to select the internal sampling clock phase

Bit(s)	R/W	Default	Description
25~23	R/W	0	SARADC_CTRL[2:0]: 000 ssa 001 vdda/4 010 vdda/2 011 vdda*3/4 100 vdda 101, 110, 111 unused
22	R/W	0	DETECT_EN: This bit controls the analog switch that connects a 50k resistor to the X+ signal. Setting This bit to 1 closes the analog switch
21	R/W	0	ADC_EN: Set This bit to 1 to enable the ADC
20-18	R/W	2	PANEL_DETECT_COUNT: Increasing this value increases the filtering on the panel detect signal using the timebase settings in Bits [17:16] below.
17-16	R/W	0	PANEL_DETECT_FILTER_TB: 0 = count 1uS ticks, 1 = count 10uS ticks, 2 = count 100uS ticks. 3 = count 1mS ticks
15-10	R/W	20	ADC_CLK_DIV: The ADC clock is derived by dividing the 27Mhz crystal by N+1. This value divides the 27Mhz clock to generate an ADC clock. A value of 20 for example divides the 27Mhz clock by 21 to generate an equivalent 1.28Mhz clock.
9-8	R/W	1	BLOCK_DLY_SEL: 0 = count 1uS ticks, 1 = count 10uS ticks, 2 = count 100uS ticks. 3 = count 1mS ticks
7-0	R/W	10	BLOCK_DLY: After all channels in the CHANNEL_LIST have been processed, the sampling engine will delay for an amount of time before re-processing the CHANNEL_LIST again. Combined with Bits [9:8] above, this value is used to generate a delay between processing blocks of channels.

SAR_ADC_DELAY:INPUT / SAMPLING DELAY 0x84

As the CHANNEL_LIST is process, the input switches are set according to the requirements of the channel. After setting the switches there is a programmable delay before sampling begins. Additionally, each channel specifies the number of samples for that particular channel. The sampling rate is programmed below.

Bit(s)	R/W	Default	Description
31-29	R	0	unused
28	R/W	0	CNTL_EOC_BY_CNT: ADC dout valid controlled by counter
27	R/W	0	CNTL_USE_LATCHED_DATA: ADC dout be latched first, then be sampled
26	R	0	unused
25-24	R/W	0	INPUT_DLY_SEL: 0 = 111nS ticks, 1 = count 1uS ticks, 2 = count 10uS ticks, 3 = count 100uS ticks
23-16	R/W	3	INPUT_DLY_CNY: For channels that acquire 2,4 or 8 samples, the delay between two samples is controlled by this count (N+1) combined with the delay selection in the two bits above.
15-10	R/W	14	CNTL_EOC_DLY_CNT: the delay between SC and ADC output data ready for latch/sample

Bit(s)	R/W	Default	Description
9-8	R/W	0	SAMPLE_DLY_SEL: 0 = count 1uS ticks, 1 = count 10uS ticks, 2 = count 100uS ticks. 3 = count 1mS ticks
7-0	R/W	9	SAMPLE_DLY_CNY: For channels that acquire 2,4 or 8 samples, the delay between two samples is controlled by this count (N+1) combined with the delay selection in the two bits above.

SAR_ADC_LAST_RD: Last Sample 0x85

For channel 0 and channel 1, (the special X/Y channels) the last sample pushed into the FIFO for each channel is saved in a register. This allows the software to see the last sample for channel 0 and channel 1 even when the FIFO overflows. For example, if we are sampling quickly and there is a gesture on the screen, we can use the contents of the FIFO to see the direction of the gesture and use the last sample values to see where the pen finally came to rest.

Bit(s)	R/W	Default	Description
31-24	R	0	unused
23-16	R	0	LAST_CHANNEL1
15-10	R	0	unused
9-0	R	0	LAST_CHANNEL0

SAR_ADC_FIFO_RD: Control Register #6 (FIFO RD) 0x86

Bit(s)	R/W	Default	Description
31-16	R	0	Unused
15	R	0	Unused
14-12	R	0	Channel ID. This value identifies the channel associated with the data in Bits [9:0] below
11-10	R	0	Unused
9-0	R	0	Sample value: 9-bit raw or averaged ADC sample written to the FIFO.

SAR_ADC_AUX_SW:Channel 2~7 ADC MUX, Switch Controls 0x87

Channels 2 ~ 7 can program the ADC input mux to any selection between 0 and 7. This register allows the software to associate a mux selection with a particular channel. In addition to the ADC mux, there are a number of switches that can be set in any particular state. Channels 2 ~ 7 share a common switch setting. Channels 0 and 1 on the other hand have programmable switch settings (see other registers below).

Bit(s)	R/W	Default	Description
31-26	R	0	unused
25-23	R/W	7	Channel 7 ADC_MUX setting when channel 7 is being measured.
22-20	R/W	7	Channel 6 ADC_MUX setting when channel 6 is being measured.
19-17	R/W	7	Channel 5 ADC_MUX setting when channel 5 is being measured.
16-14	R/W	6	Channel 4 ADC_MUX setting when channel 4 is being measured.

13-11	R/W	0	Channel 3 ADC_MUX setting when channel 3 is being measured.
10-8	R/W	1	Channel 2 ADC_MUX setting when channel 2 is being measured.
7	R	0	unused
6	R/W	0	VREF_P_MUX setting when channel 2,3..7 is being measured
5	R/W	0	VREF_N_MUX setting when channel 2,3..7 is being measured
4	R/W	0	MODE_SEL setting when channel 2,3..7 is being measured
3	R/W	1	YP_DRIVE_SW setting when channel 2,3..7 is being measured
2	R/W	1	XP_DRIVE_SW setting when channel 2,3..7 is being measured
1	R/W	0	YM_DRIVE_SW setting when channel 2,3..7 is being measured
0	R/W	0	YM_DRIVE_SW setting when channel 2,3..7 is being measured

SAR_ADC_CHAN_10_SW:Channel 0, 1 ADC MUX, Switch Controls 0x88

Channels 0 and 1 have independent programmable switch settings when either/both of these channels are being measured.

Bit(s)	R/W	Default	Description
31-26	R	0	unused
25-23	R/W	2	Channel 1 ADC MUX setting
22	R/W	0	Channel 1 VREF_P_MUX
21	R/W	0	Channel 1 VREF_N_MUX
20	R/W	0	Channel 1 MODE_SEL
19	R/W	1	Channel 1 YP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
18	R/W	1	Channel 1 XP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
17	R/W	0	Channel 1 YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
16	R/W	0	Channel 1 YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
15-10	R		unused
9-7	R/W	3	Channel 0 ADC MUX setting

Bit(s)	R/W	Default	Description
6	R/W	0	Channel 0 VREF_P_MUX
5	R/W	0	Channel 0 VREF_N_MUX
4	R/W	0	Channel 0 MODE_SEL
3	R/W	1	Channel 0 YP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
2	R/W	1	Channel 0 XP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
1	R/W	0	Channel 0 YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
0	R/W	0	Channel 0 YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND

SAR_ADC_DETECT_IDLE_SW:DETECT / IDLE Mode switches 0x89

IDLE MODE:

When nothing is being measured, the switches should be put into a safe state. This safe state is accomplished using Bits [9:0] below.

DETECT MODE:

When bit [26] is set, the input muxes / switches are configured according to the Bits below. Typically the software configures the switches below to correspond to the detect touch mode. That is, Y- internal MOSFET is closed so that the Y plane of the touch screen is connected to Ground. Additionally, the DETECT_EN bit(different register) set to 1 so that the 50k resistor to VDD is connected to X+. In this configuration, the detect comparator connected to the 50k resistor will be weakly pulled up to VDD through the 50k resistor. If the user touches the screen, the X and Y planes of the touch screen will contact causing the X+ signal to be pulled to ground.

Bit(s)	R/W	Default	Description
31-27	R	0	unused
26	R/W	0	DETECT_SW_EN: If This bit is set, then Bits [25:16] below are applied to the analog muxes/switches of the touch panel controller.
25-23	R/W	5	DETECT MODE ADC MUX setting
22	R/W	0	DETECT MODE VREF_P_MUX setting
21	R/W	0	DETECT MODE VREF_N_MUX setting
20	R/W	0	DETECT MODE MODE_SEL setting

Bit(s)	R/W	Default	Description
19	R/W	1	DETECT MODE YP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
18	R/W	1	DETECT MODE XP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
17	R/W	0	DETECT MODE YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
16	R/W	0	DETECT MODE YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
15-10	R		Unused
9-7	R/W	5	IDLE MODE ADC MUX setting
6	R/W	0	IDLE MODE VREF_P_MUX setting
5	R/W	0	IDLE MODE VREF_N_MUX setting
4	R/W	0	IDLE MODE MODE_SEL setting
3	R/W	1	IDLE MODE YP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
2	R/W	1	IDLE MODE XP_DRIVE_SW setting: 0: TADC_CH6N = 3.3v 1: TADC_CH6N = floating
1	R/W	0	IDLE MODE YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND
0	R/W	0	IDLE MODE YM_DRIVE_SW setting: 0: TADC_CH4N = floating 1: TADC_CH4N = GND

SAR_ADC_DELTA_10:Delta Mode Deltas 0x8a

Bit(s)	R/W	Default	Description
31-28	R	0	unused
27	R/W		TEMP_SEL

26	R/W		TS_REVE[1]
25-16	R/W	0	Channel 1 delta value when delta processing for channel 1 is enabled.
15	R/W		TS_REVE[0]
14-11	R/W		TS_C[3:0]
10	R/W	0	TS_VBG_EN
9-0	R/W	0	Channel 0 delta value when delta processing for channel 0 is enabled.

SAR_ADC_REG11: 0x8b

Bit(s)	R/W	Default	Description
31-30	R/W	0	unused
29-13	R/W	0	ts_cntl_int
12-0	R/W	0	sar_bg_cntl

SAR_ADC_REG12: 0x8c

Bit(s)	R/W	Default	Description
31-0	R/W	0	reserved

SAR_ADC_REG13: 0x8d

Bit(s)	R/W	Default	Description
15-8	R/W	0	SARADC_RSV2
7-0	R/W	0	reserved

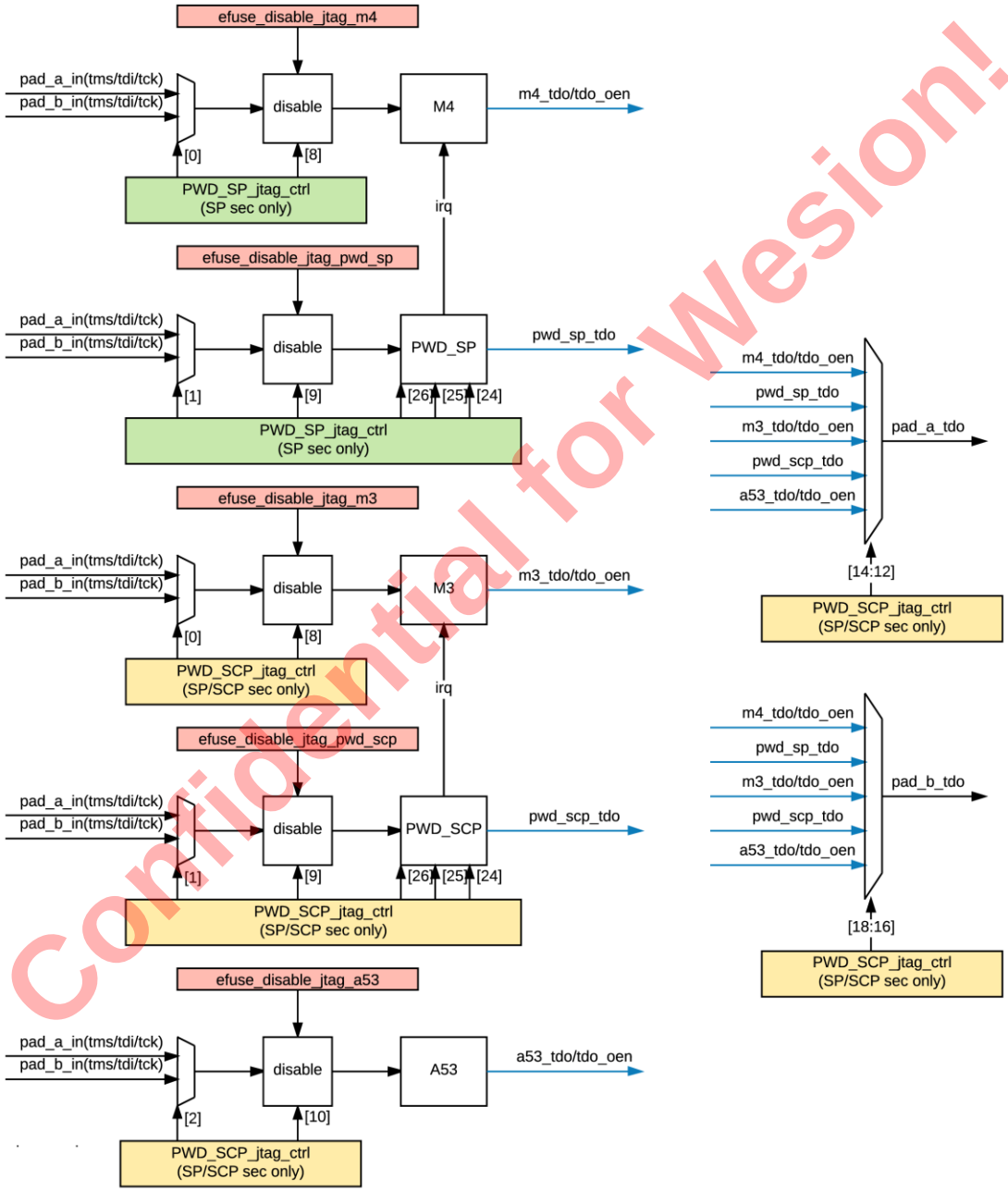
15. System Interface

15.1 JTAG

15.1.1 Overview

JTAG is an interface for internal test. The structure of S922X JTAG module is shown in the following diagram:

Figure 15-1 Diagram of JTAG



15.1.2 Register Description

Base address: 0xFF800000

Register address: 0xFF800000 + offset * 4

AO_SEC_JTAG_SP_CTRL 0xac

Bit(s)	R/W	Default	Description
31	R	0	jtag_pwd_sp_data_valid
30	R	0	jtag_pwd_sp_timeout
26	R/W	0	jtag_pwd_sp_wd_resetn
25	R/W	0	jtag_pwd_sp_clr
24	R/W	0	jtag_pwd_sp_en
9	R/W	0	jtag_disable_pwd_sp_in
8	R/W	0	jtag_disable_m4_in
4	R/W	0	jtag_sel_force_m4_tdo_en_low
1	R/W	0	jtag_sel_pwd_sp_in
0	R/W	0	jtag_sel_m4_in

AO_SEC_JTAG_PWD_SP_0~3 0xad~b0

Bit(s)	R/W	Default	Description
31:0	R	0	jtag pwd sp received data

AO_SEC_JTAG_PWD_SP_CTRL 0xb1

Bit(s)	R/W	Default	Description
31:5	R/W	0	jtag_cnt_tc
4	R/W	0	use tdo invert as trigger
3	R/W	0	use tck invert as trigger
2	R/W	0	use tms invert as trigger
1	R/W	0	use tdi invert as trigger
0	R/W	0	jtag timeout en

AO_SEC_JTAG_PWD_SP_ADDR0~3 0xb2~b5

Bit(s)	R/W	Default	Description
31:0	R/W	0	jtag pwd sp address

AO_SEC_JTAG_SCP_CTRL 0xb6

Bit(s)	R/W	Default	Description
31	R	0	jtag_pwd_scp_data_valid

Bit(s)	R/W	Default	Description
30	R	0	jtag_pwd_scp_timeout
26	R/W	0	jtag_pwd_scp_wd_resetn
25	R/W	0	jtag_pwd_scp_clr
24	R/W	0	jtag_pwd_scp_en
18:16	R/W	0	jtag_pad_b_out sel: 0: none; 1:m4; 2:jtag_pwd_sp; 3:m3; 4:jtag_pwd_scp; 5:a53
14:12	R/W	0	jtag_pad_a_out_sel: 0: none; 1:m4; 2:jtag_pwd_sp; 3:m3; 4:jtag_pwd_scp; 5:a53
10	R/W	0	jtag_disable_a53_in
9	R/W	0	jtag_disable_pwd_scp_in
8	R/W	0	jtag_disable_m3_in
5	R/W	0	jtag_sel_force_a53_tdo_en_low
4	R/W	0	jtag_sel_force_m3_tdo_en_low
2	R/W	0	jtag_sel_a53_in
1	R/W	0	jtag_sel_pwd_scp_in
0	R/W	0	jtag_sel_m3_in

AO_SEC_JTAG_PWD_SCP_0~3 0xb7~ba

Bit(s)	R/W	Default	Description
31:0	R	0	jtag pwd scp received data

AO_SEC_JTAG_PWD_SCP_CTRL 0xbb

Bit(s)	R/W	Default	Description
31:5	R/W	0	jtag_cnt_tc

4	R/W	0	use tdo invert as trigger
3	R/W	0	use tck invert as trigger
2	R/W	0	use tms invert as trigger
1	R/W	0	use tdi invert as trigger
0	R/W	0	jtag timeout en

AO_SEC_JTAG_PWD_SCP_ADDR0~3 0xbc~bf

Bit(s)	R/W	Default	Description
31:0	R/W	0	jtag pwd scp address

15.2 Temp Sensor

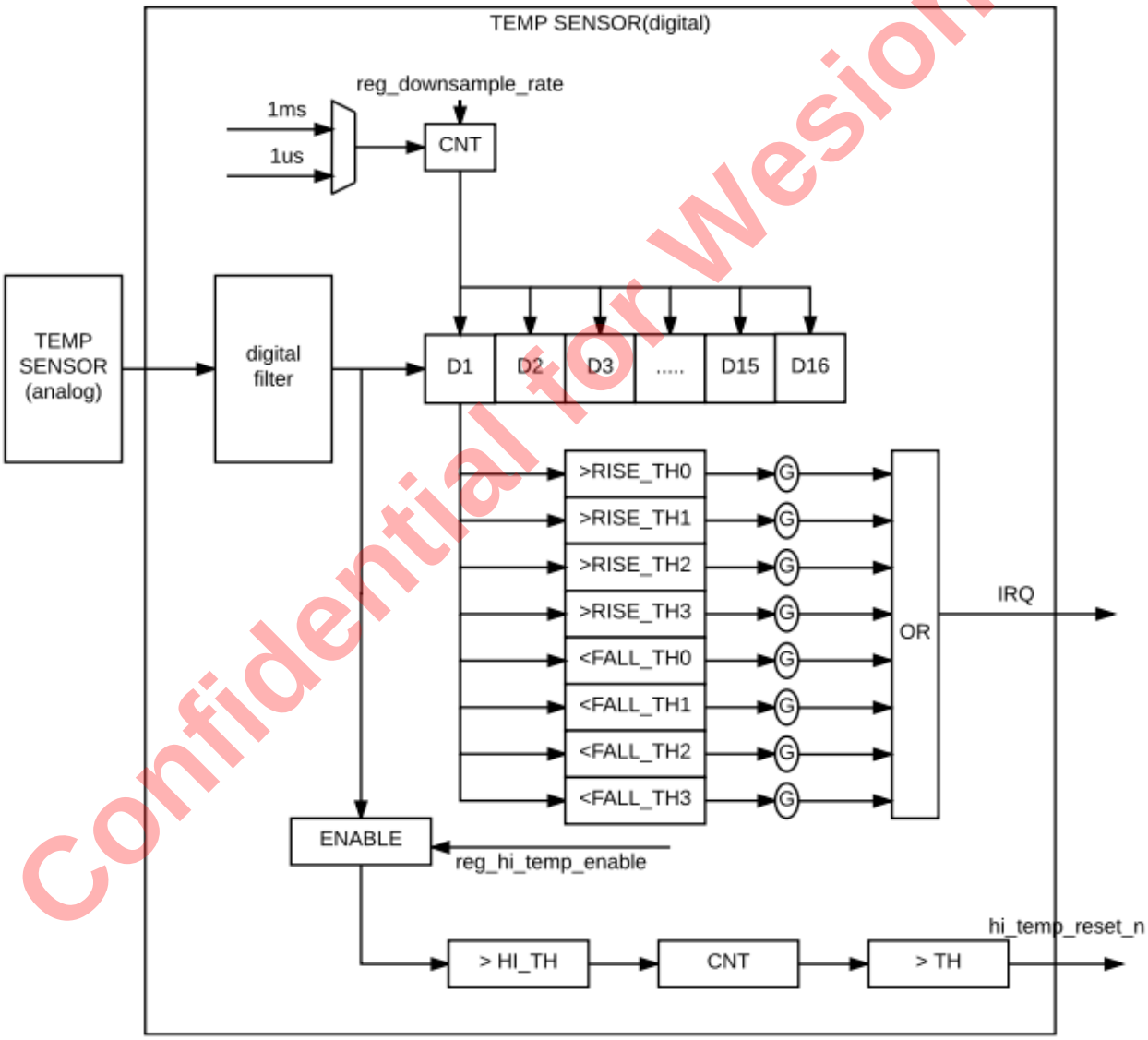
15.2.1 Overview

S922X integrates 2 Temp Sensor, one is close to DDR, one is close to PLL (between CPU and GPU).

Each Temp Sensor are the same design.

9. Capture temperature by programable tick;
10. Store 16 temperature value;
11. Can reset all chip if detect high temperature;
12. Can generate IRQ by 8 threshold;

Figure 15-2 Temperature Sensor Diagram



15.2.2 Register Description

15.2.2.1 Temp Sensor PLL Registers

Base Address: 0xFF634800

Each register final address = module base address+ address * 4

15.2.2.2 Temp Sensor DDR Registers

Base Address: 0xFF634C00

Each register final address = module base address+ address * 4

TS_CFG_REG1 0x001

Bits	R/W	Default	Description
31	R/W	0x00000000	fall_th3_irq_en
30	R/W		fall_th2_irq_en
29	R/W		fall_th1_irq_en
28	R/W		fall_th0_irq_en
27	R/W		rise_th3_irq_en
26	R/W		rise_th2_irq_en
25	R/W		rise_th1_irq_en
24	R/W		rise_th0_irq_en
23	R/W		fall_th3_irq_stat_clr
22	R/W		fall_th2_irq_stat_clr
21	R/W		fall_th1_irq_stat_clr
20	R/W		fall_th0_irq_stat_clr
19	R/W		rise_th3_irq_stat_clr
18	R/W		rise_th2_irq_stat_clr
17	R/W		rise_th1_irq_stat_clr
16	R/W		rise_th0_irq_stat_clr
15	R/W		1: enable IRQ related function.
14	R/W		fast_mode: 0 : downsample unit = 1ms; 1: downsample unit = 1us;
13	R/W		clr_hi_temp_stat
12	R/W		ts_ana_rset_vbg reset vbg(set 0, if have error place set it plus 01000..)
11	R/W		ts_ana_rst_sd reset adc(set 0, if have error place set it plus 01000..)

Bits	R/W	Default	Description
10	R/W		ts_ana_en_vcm enable vcm (disable:0; enable:1)
9	R/W		ts_ana_en_vbg enable vbg (disable:0; enable: 1)
8:7	R/W		filter hcic mode 0: downsample rate = 128; 1: downsample rate = 256; 2/3: downsample rate = 512;
6	R/W		filter ts_out_ctrl; 1: add more delay for filter lock;
5	R/W		filter en(disable:0; enable:1)
4	R/W		ts_ana_en_iptat, useless.
3	R/W		Temp Sensor DEM enable. (disable:0; enable:1)
2:0	R/W		Bipolar bias current input control. recommend value : 3. ts_ana_ch_sel; 0: 8'b00000001; 1: 8'b00000011; 2: 8'b00000111; 3: 8'b00001111; 4: 8'b00011111; 5: 8'b00111111; 6: 8'b01111111; 7: 8'b11111111;

TS_CFG_REG2 0x002

Bits	R/W	Default	Description
31	R/W	0x00000000	hi_temp_enable
30	R/W		reset_en, if = 0, will not reset all chip;
27:16	R/W		high temperature times, if continuous detect high temperature, then will reset all chip.
15:0	R/W		high temperature threshold, if temperature value > this th , mean detected once high temperature

TS_CFG_REG3 0x03

Bits	R/W	Default	Description

31:16	R/W	0x00000000	
15:0	R/W		down_sample rate

TS_CFG_REG4 0x04

Bits	R/W	Default	Description
23:12	R/W	0x00000000	rise_th0
11:0	R/W		rise_th1

TS_CFG_REG5 0x05

Bits	R/W	Default	Description
23:12	R/W	0x00000000	rise_th2
11:0	R/W		rise_th3

TS_CFG_REG6 0x06

Bits	R/W	Default	Description
23:12	R/W	0x00000000	fall_th0
11:0	R/W		fall_th1

TS_CFG_REG7 0x07

Bits	R/W	Default	Description
23:12	R/W	0x00000000	fall_th2
11:0	R/W		fall_th3

TS_STAT0 0x10

Bits	R/W	Default	Description
31:18	R	0x00000000	detect_hi_temp_cnt
17	R		detected_hi_temp_stat
16	R		filter lock
15:0	R		filter out

TS_STAT1 0x11

Bits	R/W	Default	Description
31:9	R	0x00000000	
8	R		hi_temp_stat
7	R		fall_th3_irq

6	R		fall_th2_irq
5	R		fall_th1_irq
4	R		fall_th0_irq
3	R		rise_th3_irq
2	R		rise_th2_irq
1	R		rise_th1_irq
0	R		rise_th0_irq

TS_STAT2 0x12

Bits	R/W	Default	Description
31:16	R	0x00000000	temperature value D2
15:0	R		temperature value D1

TS_STAT3 0x13

Bits	R/W	Default	Description
31:16	R	0x00000000	temperature value D4
15:0	R		temperature value D3

TS_STAT4 0x14

Bits	R/W	Default	Description
31:16	R	0x00000000	temperature value D6
15:0	R		temperature value D5

TS_STAT5 0x15

Bits	R/W	Default	Description
31:16	R	0x00000000	temperature value D8
15:0	R		temperature value D7

TS_STAT6 0x16

Bits	R/W	Default	Description
31:16	R	0x00000000	temperature value D10
15:0	R		temperature value D9

TS_STAT7 0x17

Bits	R/W	Default	Description

31:16	R	0x00000000	temperature value D12
15:0	R		temperature value D11

TS_STAT8 0x18

Bits	R/W	Default	Description
31:16	R	0x00000000	temperature value D14
15:0	R		temperature value D13

TS_STAT9 0x19

Bits	R/W	Default	Description
31:16	R	0x00000000	temperature value D16
15:0	R		temperature value D15