开阳电子

AMT630HV160

Automotive Graphic Display Controller

版本： **V1.3**

日期： **2024-12**

ARKmicro Technologies Inc reserves the right to change or modify the information contained herein without notice. It is the customer's responsibility to obtain the most recent revision of the document. **ARKmicro Technologies Inc** makes no warranty for the use of its products and bears no responsibility for any errors or omissions that may appear in the document.

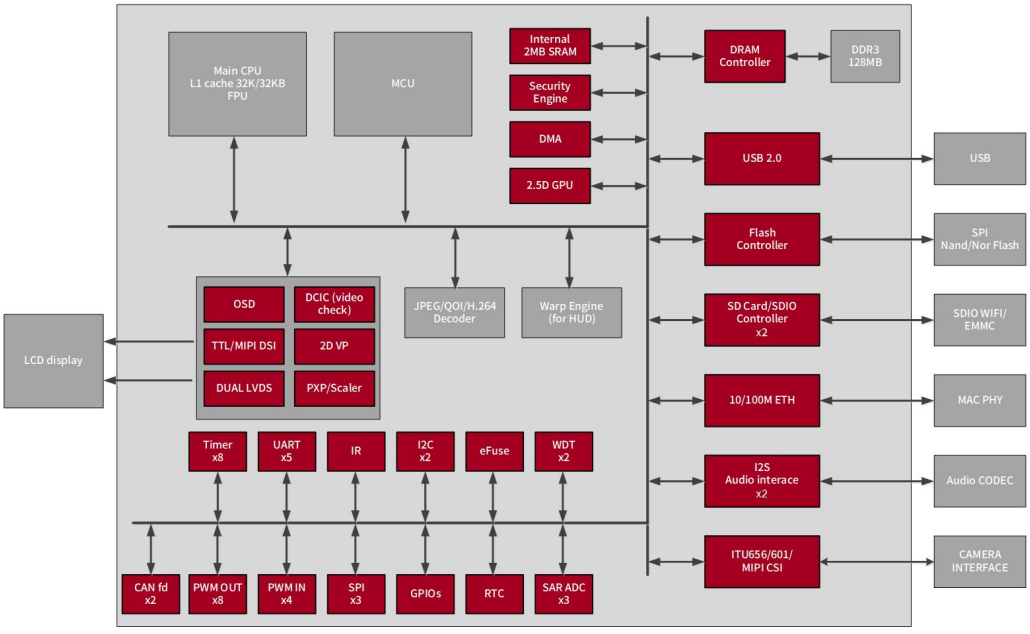
Version Record:

Date	Version	Modified Description
2023-12	V1.0	Initial version
2024-04	V1.1	Add Hud warping engine
2024-11	V1.2	Add Thermal Information，Power Sequence
2024-12	V1.3	Add RDC chapter

1.General Description

AMT630HV160 features state-of-the-art functions and performance for in-vehicle graphic display applications such as digital clusters, central HMI system. It integrates a high-performance CPU core and a power efficient MCU safety core. AMT630HV160 has rich video and audio feature, including 1080p H.264 video decoder, lower power 2.5D graphics and vector engine, MIPI CSI/ITU656/ITU601 video input, TTL/LVDS/MIPI DSI display, I2S interface. AMT630HV160 also provide various memory interface, including 2MB on-chip ram, DDR3 interface, SD/eMMC, SPI Quad NOR FLASH, SPI Quad NAND FLASH and other useful peripherals for smart display applications.

2.System Architecture



3.Features

3.1 CPU

- Single-core Cortex-A7 up to 800MHz
- 32KB instruction cache and 32KB data cache
- Embedded JTAG for software debug

3.2 Security MCU subsystem

- MCU up to 240MHz, with security debug interface
- Program/Data Ram share with 288K ECC Ram
- Embedded SMPU
- Embedded hardware semaphores to access Memory and peripherals resource domain controller.
- Embedded mailbox for message exchange between two Cores.

3.3 Low power 2D and vector graphic

- Supports move operations, in positive and negative direction
- Supports 16 different ROP2 operations
- Porter-Duff composition with/without global alpha
- Integrated optional up/down bitmap scaling
- Integrated vertical and horizontal bitmap flipping
- Color-keyed transparency, source and destination
- Control of pixel alpha blending factors
- Anti-aliased 8-bit font expansion
- 2.5D rendering

- Supported color formats: YUV, YUV420, RGB565, ARGB565, RGB888 and ARGB888

3.4 Memory Subsystem

3.4.1 Boot ROM

- On-chip memory
- Support Security Boot
- Supports system boot from the following devices:
  SD/eMMC SPI Nor Flash
- Supports mandatory upgrade process through SD and USB

3.4.2 Embedded 2MByte VRAM

3.4.3 SDRAM

- Embedded 128MB DDR3 DRAM up to 1600M data rate
- Enter auto or self-refresh mode automatically

3.4.4 SD/MMC

- Two SD/MMC host controller interfaces
- SDMMC controller controls the devices that comply with the Secure Digital (SD3.0) controls the devices that comply with the Multimedia Card (eMMC5.1)
- Support 4-bit bus width
- Supports block size of 1 to 65535 bytes

3.5 MIPI CSI2/ITU656/601 Interface

- Support MIPI CSI/ITU601/656, YUV422/420 input up to 1920x1080
- MIPI CSI2 support YUV420, YUV422, RGB888 format
- Auto-adapt to the format and the resolution of the input
- Support Stepless Scaler
- Support 90、180、270 rotate and flip

3.6 Video engine

3.6.1 video decoder

- Supports multi-formats:

  - H.264 BP/MP/HP@L4.2 up to 1080p@30fps

  - JPEG 8000x8000

  - QOI picture decompress 1080p@30fps

3.7 Warp engine

- Support warp engine for HUD application, up to 1280x720
- Support RGB888 format

3.8 Display

- Support one video layers, one OSD layer.
- Support overlay and per-pixel alpha blending between two channels
- Each channel can be stopped separately
- Support ARGB888/RGB666/RGB565/YUV420/YUV422 RGB Interface
- Support ITU656/601/1120/SRGB/CPU panel interface
- Support 24 RGB Interface LCD/Dual LVDS/MIPI DSI, up to 1080@60fps
- Support Display Content Integrity Checker

3.9 DMAC

- Support eight channels DMA
- 64 DMA request and response line
- Programmable DMA burst size
- Support 8, 16 and 32 bit wide transactions
- Incrementing and non-incrementing addressing for source and destination

3.10 Audio Interface

- Two separate I2S

3.11 USB Interface

- One USB 2.0 Host/Device up to 480Mbps

3.12 EMAC

- One MAC interface for connecting external Ethernet PHY
- 10/100 Mbit/s Ethernet port with RMII interface
- Compliant with IEEE 802.3-2002 standard
- Supports full and half duplex operations

3.13 SAR ADC Interface

- 3x SAR ADC, 12bit resolution,1M sample rate
- 8-CH multiplexed with per ADC
- Built-in comparer to generate interrupts

3.14 Secure engine

- Embedded 2Kbit e-fuse for security manager
- Cipher algorithms: AES, SM4
- Cipher modes: ECB, CBC, CTR, OFB, CFB
- Hash (MAC) algorithms: MD5, SHA-1,SHA-256/512,CRC32
- Hash modes: Raw, HMAC
- Combined modes: AES-GCM, AES-CCM
- Support RSA1024 RSA2048 RSA4096 ECC(include SM2)

- Embedded TRNG

3.15 Can FD

- Two channels Support
- Support CAN 2.0B, CAN FD
- Per channel Support 16 acceptance filter

3.16 Always on RTC block

- Low power with 16KByte PRAM (100uA）
- Support GPIO/CAN wake up

3.17 Other module and Interface

- 7 channels PLL
- 8 channels 32bit timer with diff source clock
- 8 channels PWM out and 4 channels PWM in
- Watchdog timer
- Reset Controller
- 2 channels 4-wire high-speed UART and 4 channels normal UART for GPRS, Bluetooth, GPS etc.
- 3 channels SPI interface: 2 channels 4bit and 1 channel 1bit
- Support IrDA1.1 and Remoter
- 2 channels I2C interface
- 64 GPIOs

3.18 Package

- BGA352

- 3.3V and 0.9V (kernel) power supplies

3.19 Reliability

- Qualification according to AEC-Q100 Grade2 and ASIL-B
- Ambient Temperature (T_A): -40℃~105℃
- Storage Temperature: -55℃~150℃
- ESD (HBM): ±2KV

4.Ball Description

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
A		VSS	uart3_cts	uart3_txd	VSS	VDD	ADC2_AUX4	ADC2_AUX3	ADC1_AUX4	ADC1_AUX3	ADC0_AUX4	ADC0_AUX3	csi_datap2	csi_ckn	csi_datap0	sda0	tms	tdo	GPIO20	GPIO13	GPIO12		A
B	VSS	GPIO24	uart3_rts	uart3_rxd	VSS	VDD	ADC2_AUX5	ADC2_AUX2	ADC1_AUX5	ADC1_AUX2	ADC0_AUX5	ADC0_AUX2	csi_datan2	csi_ckp	csi_datan0	scl0	tdi	GPIO21	GPIO19	GPIO14	GPIO11	GPIO8	B
C	GPIO25	GPIO26	GPIO27	GPIO31	VSS	VDD	ADC2_AUX6	ADC2_AUX1	ADC1_AUX6	ADC1_AUX1	ADC0_AUX6	ADC0_AUX1	csi_datan3	csi_datap1	itu_vsync	mclk	tck	GPIO22	GPIO18	GPIO15	GPIO10	GPIO7	C
D	GPIO30	GPIO29	GPIO28	GPIO34	VSS	VDD	ADC2_AUX7	ADC2_AUX0	ADC1_AUX7	ADC1_AUX0	ADC0_AUX7	ADC0_AUX0	csi_datap3	csi_datan1	AVDD18_CSI	ntrst	rcrt_in	GPIO23	GPIO17	GPIO16	GPIO9	GPIO6	D
E	XI_24M	GPIO33	GPIO32	DVDD_PD4	VSS	VDD	VQPS	VDD	VDD	AVDD18_ADC	AVDD18_ADC	VDD	csi_rext	DVDD_CSI	AVSS_CSI	VSS	VSS	VSS	VSS	VSS	VSS	VSS	E
F	XO_24M	GPIO38	GPIO35	DVDD_PD4	VSS													VDD	VDD	VDD	VDD	VDD	F
G	XO_32K	GPIO39	testmode	VDD	VDD													VDD	GPIO2	GPIO3	GPIO4	GPIO5	G
H	XI_32K	PWR_SEQ_RTC	RSTN_RTC	VDD	AVSS_PLL													VSS	GPIO1	GPIO0	pwmout3	pwmout2	H
J	ACC_RTC	CAN0_RTC	CAN1_RTC	VDD	AVDD09_PLL													VSS	pwmin3	pwmin2	pwmout0	pwmout1	J
K	CAN0_SBY_RTC	GPIO1_WK_RTC	GPIO0_WK_RTC	VDD09_RTC_LDO	VDD33_RTC					VSS	VSS	VSS	VSS					DVDD_PD3	pwmin0	pwmin1	scl1	sda1	K
L	CAN1_SBY_RTC	CAN0_RXD_RTC	CAN1_RXD_RTC	VDD18_RTC_LDO	VSS_RTC					VSS	VSS	VSS	VSS					DVDD_PD3	uart0_txd	uart0_rxd	uart0_cts	uart0_rts	L
M	VDDQ15	VDDQ15	VDDQ15	VDDQ15	VDDQ15					VSS	VSS	VSS	VSS					DVDD_PD3	ssp2_d3	ssp2_d2	ssp2_d1	ssp2_d0	M
N	VSSQ	VSSQ	KGD_PZQ	DDRPHY_ZQ	DDR_VREF					VSS	VSS	VSS	VSS					VSS	ssp2_csn0_out	ssp2_clkout	sspl_csn	sspl_clk	N
P	VSSQ	VSSQ	VSSQ	VSSQ	VSSQ													VSS	ssp0_csn0_out	ssp0_clkout	sspl_txd	sspl_rxd	P
R	VDD	VDD	VDD	VDD	VDD													VDD	ssp0_d3	ssp0_d2	ssp0_d1	ssp0_d0	R
T	can1_tx	can1_rx	can0_tx	VDD	VSS_DDR_PLL													VDD	VDD	VDD	VDD	VDD	T
U	can1_stby	gpio130	DVDD_PD1	can0_rx	VDD18_DDR_PLL													VSS	VSS	i2s0_bclk	i2s0_mclk	i2s0_sync	U
V	gpio131	gpio132	bootsel0	can0_stby	AVDD18_DSI	VSS	VSS	VSS	VSS	VDD	VDD	DVDD_PD2	AVDD18_LVDS	VDD	VDD18_USB	AVSS_USB	DVDD_EMMC	VSS	VSS	sdl_d3	i2s0_sdout	i2s0_sdin	V
W	bootsel2	bootsel1	DVDD_PD2	VSS	AVSS_DSI	GPIO136	DVDD_PD2	GPIO143	pwmout7	OUT2_DP	OUT2_CP	OUT2_BP	AVSS_LVDS	VDD	VDD33_USB	TXRTUNE_USB	VSS	DVDD_SD1	uart1_rxd	sdl_d2	sdl_d1	sdl_d0	W
Y	DSI_DATAN0	DSI_DATAP1	DSI_DATAP2	DSI_DATAN3	GPIO133	GPIO137	GPIO140	pwmout4	GPI036	OUT2_DN	OUT2_CN	OUT2_BN	OUT1_CP	OUT1_CN	OUT1_AP	VBUS_USB	emmc_cd	emmc_cmd	i2s1_sync	uart1_txd	sdl_cmd	sdl_clk	Y
AA	DSI_DATAP0	DSI_DATAN1	DSI_DATAN2	DSI_DATAP3	GPIO134	GPIO138	GPIO141	pwmout5	GPI037	OUT2_CKP	OUT2_AP	OUT2_AN	OUT1_CKN	OUT1_CKP	OUT1_AN	IDO_USB	emmc_clk	emmc_d0	emmc_d3	i2s1_bclk	i2s1_sdout	sdl_cd	AA
AB		DSI_CKN	DSI_CKP	DSI_REXT	GPIO135	GPIO139	GPIO142	pwmout6	uart2_txd	uart2_rxd	OUT2_CKN	OUT1_DP	OUT1_DN	OUT1_BP	OUT1_BN	DP_USB	DM_USB	emmc_dl	emmc_d2	i2s1_mclk	i2s1_sdin		AB
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	

Ball Description

Ball 编号	Ball 名称	类型	复位状态 IO	Pull up/down 配置选择	默认驱动电流(mA)	供电电源
RTC						
E1	XI_24M	I	NA	NA	NA	VDD33_RTC
F1	XO_24M	O	NA	NA	NA	VDD33_RTC
G1	XO_32K	O	NA	NA	NA	VDD33_RTC
H1	XI_32K	I	NA	NA	NA	VDD33_RTC
H2	PWR_SEQ_RTC	O	NA	NA	NA	VDD33_RTC
H3	RSTN_RTC	I	NA	NA	NA	VDD33_RTC
J1	ACC_RTC	I	NA	NA	NA	VDD33_RTC
J2	CAN0_RTC	I	NA	NA	NA	VDD33_RTC
J3	CAN1_RTC	I	NA	NA	NA	VDD33_RTC
K3	GPIO0_WK_RTC	IO	NA	NA	NA	VDD33_RTC
K2	GPIO1_WK_RTC	IO	NA	NA	NA	VDD33_RTC
K1	CAN0_SBY_RTC	IO	NA	NA	NA	VDD33_RTC
L1	CAN1_SBY_RTC	IO	NA	NA	NA	VDD33_RTC
L2	CAN0_RXD_RTC	IO	NA	NA	NA	VDD33_RTC
L3	CAN1_RXD_RTC	IO	NA	NA	NA	VDD33_RTC
K5	VDD33_RTC	P	NA	NA	NA	
K4	VDD09_RTC_LDO	P	NA	NA	NA	
L4	VDD18_RTC_LDO	P	NA	NA	NA	
L5	VSS_RTC	G	NA	NA	NA	
USB						
AA16	ID0_USB	I	I	NA	NA	VDD33_USB
Y16	VBUS_USB	I	I	NA	NA	VDD33_USB
AB16	DP_USB	ANA	NA	NA	NA	VDD33_USB
AB17	DM_USB	ANA	NA	NA	NA	VDD33_USB
W16	TXRTUNE_USB	ANA	NA	NA	NA	VDD33_USB
W15	VDD33_USB	P	NA	NA	NA	
V15	VDD18_USB	P	NA	NA	NA	
V16	AVSS_USB	G	NA	NA	NA	
DDR						
N3	KGD_PZQ	ANA	NA	NA	NA	VDDQ15

N4	DDRPHY_ZQ	ANA	NA	NA	NA	VDDQ15
N5	DDR_VREF	ANA	NA	NA	NA	VDDQ15
U5	VDD18_DDR_PLL	P	NA	NA	NA	
T5	VSS_DDR_PLL	G	NA	NA	NA	
M1,M2,M3,M4,M5	VDDQ15	P	NA	NA	NA	
N1,N2,P1,P2,P3,P4,P5	VSSQ	G	NA	NA	NA	
GPIO Group1						
U4	can0_rx	IO	I	PU/PD	8mA	DVDD_PD1
T3	can0_tx	IO	I	PU/PD	8mA	DVDD_PD1
V4	can0_stby	IO	I	PU/PD	8mA	DVDD_PD1
T2	can1_rx	IO	I	PU/PD	8mA	DVDD_PD1
T1	can1_tx	IO	I	PU/PD	8mA	DVDD_PD1
U1	can1_stby	IO	I	PU/PD	8mA	DVDD_PD1
U2	gpio130	IO	I	PU/PD	8mA	DVDD_PD1
V1	gpio131	IO	I	PU/PD	8mA	DVDD_PD1
V2	gpio132	IO	I	PU/PD	8mA	DVDD_PD1
V3	bootsel0	IO	I	NA	8mA	DVDD_PD1
W2	bootsel1	IO	I	NA	8mA	DVDD_PD1
W1	bootsel2	IO	I	NA	8mA	DVDD_PD1
U3	DVDD_PD1	P	NA	NA	NA	
LCD/MIPI-DSI/LVDS-TX/GPIO Group2						
Y1	DSI_DATAN0	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA1	DSI_DATAP0	ANA/IO	I	PU/PD	8mA	DVDD_PD2
Y2	DSI_DATAP1	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA2	DSI_DATAN1	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AB2	DSI_CKN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AB3	DSI_CKP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
Y3	DSI_DATAP2	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA3	DSI_DATAN2	ANA/IO	I	PU/PD	8mA	DVDD_PD2
Y4	DSI_DATAN3	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA4	DSI_DATAP3	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AB4	DSI_REXT	ANA	NA	NA	NA	DVDD_PD2
V5	AVDD18_DSI	P	NA	NA	NA	DVDD_PD2
W5	AVSS_DSI	G	NA	NA	NA	DVDD_PD2
Y5	GPIO133	IO	I	PU/PD	8mA	DVDD_PD2
AA5	GPIO134	IO	I	PU/PD	8mA	DVDD_PD2

AB5	GPIO135	IO	I	PU/PD	8mA	DVDD_PD2
W6	GPIO136	IO	I	PU/PD	8mA	DVDD_PD2
Y6	GPIO137	IO	I	PU/PD	8mA	DVDD_PD2
AA6	GPIO138	IO	I	PU/PD	8mA	DVDD_PD2
AB6	GPIO139	IO	I	PU/PD	8mA	DVDD_PD2
Y7	GPIO140	IO	I	PU/PD	8mA	DVDD_PD2
AA7	GPIO141	IO	I	PU/PD	8mA	DVDD_PD2
AB7	GPIO142	IO	I	PU/PD	8mA	DVDD_PD2
W8	GPIO143	IO	I	PU/PD	8mA	DVDD_PD2
Y8	pwmout4	IO	I	PU/PD	8mA	DVDD_PD2
AA8	pwmout5	IO	I	PU/PD	8mA	DVDD_PD2
AB8	pwmout6	IO	I	PU/PD	8mA	DVDD_PD2
W9	pwmout7	IO	I	PU/PD	8mA	DVDD_PD2
Y9	GPIO36	IO	I	PU/PD	8mA	DVDD_PD2
AA9	GPIO37	IO	I	PU/PD	8mA	DVDD_PD2
AB9	uart2_txd	IO	I	PU/PD	8mA	DVDD_PD2
AB10	uart2_rxd	IO	I	PU/PD	8mA	DVDD_PD2
W10	OUT2_DP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
Y10	OUT2_DN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA10	OUT2_CKP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AB11	OUT2_CKN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
W11	OUT2_CP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
Y11	OUT2_CN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
W12	OUT2_BP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
Y12	OUT2_BN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA11	OUT2_AP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA12	OUT2_AN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AB12	OUT1_DP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AB13	OUT1_DN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA14	OUT1_CKP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA13	OUT1_CKN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
Y13	OUT1_CP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
Y14	OUT1_CN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AB14	OUT1_BP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AB15	OUT1_BN	ANA/IO	I	PU/PD	8mA	DVDD_PD2
Y15	OUT1_AP	ANA/IO	I	PU/PD	8mA	DVDD_PD2
AA15	OUT1_AN	ANA/IO	I	PU/PD	8mA	DVDD_PD2

V13	AVDD18_LVDS	P	NA	NA	NA	DVDD_PD2
W13	AVSS_LVDS	G	NA	NA	NA	DVDD_PD2
W3,W7,V12	DVDD_PD2	P	NA	NA	NA	
EMMC Group						
Y17	emmc_cd	IO	I	PU/PD	8mA	DVDD_EMMC
AA17	emmc_clk	IO	I	PU/PD	8mA	DVDD_EMMC
Y18	emmc_cmd	IO	I	PU/PD	8mA	DVDD_EMMC
AA18	emmc_d0	IO	I	PU/PD	8mA	DVDD_EMMC
AB18	emmc_d1	IO	I	PU/PD	8mA	DVDD_EMMC
AB19	emmc_d2	IO	I	PU/PD	8mA	DVDD_EMMC
AA19	emmc_d3	IO	I	PU/PD	8mA	DVDD_EMMC
V17	DVDD_EMMC	P	NA	NA	NA	
SD1 Group						
Y19	i2s1_sync	IO	I	PU/PD	8mA	DVDD_SD1
AB20	i2s1_mclk	IO	I	PU/PD	8mA	DVDD_SD1
AA20	i2s1_bclk	IO	I	PU/PD	8mA	DVDD_SD1
AA21	i2s1_sdout	IO	I	PU/PD	8mA	DVDD_SD1
AB21	i2s1_sdin	IO	I	PU/PD	8mA	DVDD_SD1
W19	uart1_rxd	IO	I	PU/PD	8mA	DVDD_SD1
Y20	uart1_txd	IO	I	PU/PD	8mA	DVDD_SD1
AA22	sd1_cd	IO	I	PU/PD	8mA	DVDD_SD1
Y22	sd1_clk	IO	I	PU/PD	8mA	DVDD_SD1
Y21	sd1_cmd	IO	I	PU/PD	8mA	DVDD_SD1
W22	sd1_d0	IO	I	PU/PD	8mA	DVDD_SD1
W21	sd1_d1	IO	I	PU/PD	8mA	DVDD_SD1
W20	sd1_d2	IO	I	PU/PD	8mA	DVDD_SD1
V20	sd1_d3	IO	I	PU/PD	8mA	DVDD_SD1
W18	DVDD_SD1	P	NA	NA	NA	
GPIO Group3						
U22	i2s0_sync	IO	I	PU/PD	8mA	DVDD_PD3
U21	i2s0_mclk	IO	I	PU/PD	8mA	DVDD_PD3
U20	i2s0_bclk	IO	I	PU/PD	8mA	DVDD_PD3
V21	i2s0_sdout	IO	I	PU/PD	8mA	DVDD_PD3
V22	i2s0_sdin	IO	I	PU/PD	8mA	DVDD_PD3
P19	ssp0_csn0_out	IO	I	PU/PD	8mA	DVDD_PD3

P20	ssp0_clkout	IO	I	PU/PD	8mA	DVDD_PD3
R22	ssp0_d0	IO	I	PU/PD	8mA	DVDD_PD3
R21	ssp0_d1	IO	I	PU/PD	8mA	DVDD_PD3
R20	ssp0_d2	IO	I	PU/PD	8mA	DVDD_PD3
R19	ssp0_d3	IO	I	PU/PD	8mA	DVDD_PD3
N21	ssp1_csn	IO	I	PU/PD	8mA	DVDD_PD3
N22	ssp1_clk	IO	I	PU/PD	8mA	DVDD_PD3
P21	ssp1_txd	IO	I	PU/PD	8mA	DVDD_PD3
P22	ssp1_rxd	IO	I	PU/PD	8mA	DVDD_PD3
N20	ssp2_clkout	IO	I	PU/PD	8mA	DVDD_PD3
N19	ssp2_csn0_out	IO	I	PU/PD	8mA	DVDD_PD3
M22	ssp2_d0	IO	I	PU/PD	8mA	DVDD_PD3
M21	ssp2_d1	IO	I	PU/PD	8mA	DVDD_PD3
M20	ssp2_d2	IO	I	PU/PD	8mA	DVDD_PD3
M19	ssp2_d3	IO	I	PU/PD	8mA	DVDD_PD3
L19	uart0_txd	IO	I	PU/PD	8mA	DVDD_PD3
L20	uart0_rxd	IO	I	PU/PD	8mA	DVDD_PD3
L21	uart0_cts	IO	I	PU/PD	8mA	DVDD_PD3
L22	uart0_rts	IO	I	PU/PD	8mA	DVDD_PD3
K21	scl1	IO	I	PU/PD	8mA	DVDD_PD3
K22	sda1	IO	I	PU/PD	8mA	DVDD_PD3
K19	pwmin0	IO	I	PU/PD	8mA	DVDD_PD3
K20	pwmin1	IO	I	PU/PD	8mA	DVDD_PD3
J20	pwmin2	IO	I	PU/PD	8mA	DVDD_PD3
J19	pwmin3	IO	I	PU/PD	8mA	DVDD_PD3
J21	pwmout0	IO	I	PU/PD	8mA	DVDD_PD3
J22	pwmout1	IO	I	PU/PD	8mA	DVDD_PD3
H22	pwmout2	IO	I	PU/PD	8mA	DVDD_PD3
H21	pwmout3	IO	I	PU/PD	8mA	DVDD_PD3
H20	GPIO0	IO	I	PU/PD	8mA	DVDD_PD3
H19	GPIO1	IO	I	PU/PD	8mA	DVDD_PD3
G19	GPIO2	IO	I	PU/PD	8mA	DVDD_PD3
G20	GPIO3	IO	I	PU/PD	8mA	DVDD_PD3
G21	GPIO4	IO	I	PU/PD	8mA	DVDD_PD3
G22	GPIO5	IO	I	PU/PD	8mA	DVDD_PD3

D22	GPIO6	IO	I	PU/PD	8mA	DVDD_PD3
C22	GPIO7	IO	I	PU/PD	8mA	DVDD_PD3
B22	GPIO8	IO	I	PU/PD	8mA	DVDD_PD3
D21	GPIO9	IO	I	PU/PD	8mA	DVDD_PD3
C21	GPIO10	IO	I	PU/PD	8mA	DVDD_PD3
B21	GPIO11	IO	I	PU/PD	8mA	DVDD_PD3
A21	GPIO12	IO	I	PU/PD	8mA	DVDD_PD3
A20	GPIO13	IO	I	PU/PD	8mA	DVDD_PD3
B20	GPIO14	IO	I	PU/PD	8mA	DVDD_PD3
C20	GPIO15	IO	I	PU/PD	8mA	DVDD_PD3
D20	GPIO16	IO	I	PU/PD	8mA	DVDD_PD3
D19	GPIO17	IO	I	PU/PD	8mA	DVDD_PD3
C19	GPIO18	IO	I	PU/PD	8mA	DVDD_PD3
B19	GPIO19	IO	I	PU/PD	8mA	DVDD_PD3
A19	GPIO20	IO	I	PU/PD	8mA	DVDD_PD3
B18	GPIO21	IO	I	PU/PD	8mA	DVDD_PD3
C18	GPIO22	IO	I	PU/PD	8mA	DVDD_PD3
D18	GPIO23	IO	I	PU/PD	8mA	DVDD_PD3
D17	rcrt_in	IO	I	PU/PD	8mA	DVDD_PD3
C17	tck	IO	I	PU/PD	8mA	DVDD_PD3
B17	tdi	IO	I	PU/PD	8mA	DVDD_PD3
A18	tdo	IO	I	PU/PD	8mA	DVDD_PD3
A17	tms	IO	I	PU/PD	8mA	DVDD_PD3
D16	ntrst	IO	I	PU/PD	8mA	DVDD_PD3
K18,L18,M18	DVDD_PD3	P	NA	NA	NA	
MIPI-CSI/GPIO Group						
B16	scl0	IO	I	PU/PD	8mA	DVDD_CSI
A16	sda0	IO	I	PU/PD	8mA	DVDD_CSI
C16	mclk	IO	I	PU/PD	8mA	DVDD_CSI
C15	itu_vsync	IO	I	PU/PD	8mA	DVDD_CSI
B15	csi_datan0	ANA/IO	I	PU/PD	8mA	DVDD_CSI
A15	csi_datap0	ANA/IO	I	PU/PD	8mA	DVDD_CSI
C14	csi_datap1	ANA/IO	I	PU/PD	8mA	DVDD_CSI
D14	csi_datan1	ANA/IO	I	PU/PD	8mA	DVDD_CSI
A14	csi_ckn	ANA/IO	I	PU/PD	8mA	DVDD_CSI
B14	csi_ckp	ANA/IO	I	PU/PD	8mA	DVDD_CSI

A13	csi_datap2	ANA/IO	I	PU/PD	8mA	DVDD_CSI
B13	csi_datan2	ANA/IO	I	PU/PD	8mA	DVDD_CSI
C13	csi_datan3	ANA/IO	I	PU/PD	8mA	DVDD_CSI
D13	csi_datap3	ANA/IO	I	PU/PD	8mA	DVDD_CSI
E13	csi_rext	ANA	NA	NA	NA	
E14	DVDD_CSI	P	NA	NA	NA	
D15	AVDD18_CSI	P	NA	NA	NA	
E15	AVSS_CSI	G	NA	NA	NA	
ADC/GPIO Group						
D12	ADC0_AUX0	ANA/IO	I	NA	8mA	AVDD18_ADC
C12	ADC0_AUX1	ANA/IO	I	NA	8mA	AVDD18_ADC
B12	ADC0_AUX2	ANA/IO	I	NA	8mA	AVDD18_ADC
A12	ADC0_AUX3	ANA/IO	I	NA	8mA	AVDD18_ADC
A11	ADC0_AUX4	ANA/IO	I	NA	8mA	AVDD18_ADC
B11	ADC0_AUX5	ANA/IO	I	NA	8mA	AVDD18_ADC
C11	ADC0_AUX6	ANA/IO	I	NA	8mA	AVDD18_ADC
D11	ADC0_AUX7	ANA/IO	I	NA	8mA	AVDD18_ADC
D10	ADC1_AUX0	ANA/IO	I	NA	8mA	AVDD18_ADC
C10	ADC1_AUX1	ANA/IO	I	NA	8mA	AVDD18_ADC
B10	ADC1_AUX2	ANA/IO	I	NA	8mA	AVDD18_ADC
A10	ADC1_AUX3	ANA/IO	I	NA	8mA	AVDD18_ADC
A9	ADC1_AUX4	ANA/IO	I	NA	8mA	AVDD18_ADC
B9	ADC1_AUX5	ANA/IO	I	NA	8mA	AVDD18_ADC
C9	ADC1_AUX6	ANA/IO	I	NA	8mA	AVDD18_ADC
D9	ADC1_AUX7	ANA/IO	I	NA	8mA	AVDD18_ADC
D8	ADC2_AUX0	ANA/IO	I	NA	8mA	AVDD18_ADC
C8	ADC2_AUX1	ANA/IO	I	NA	8mA	AVDD18_ADC
B8	ADC2_AUX2	ANA/IO	I	NA	8mA	AVDD18_ADC
A8	ADC2_AUX3	ANA/IO	I	NA	8mA	AVDD18_ADC
A7	ADC2_AUX4	ANA/IO	I	NA	8mA	AVDD18_ADC
B7	ADC2_AUX5	ANA/IO	I	NA	8mA	AVDD18_ADC
C7	ADC2_AUX6	ANA/IO	I	NA	8mA	AVDD18_ADC
D7	ADC2_AUX7	ANA/IO	I	NA	8mA	AVDD18_ADC
E10,E11	AVDD18_ADC	P	NA	NA	NA	
GPIO4 Group						
A4	uart3_txd	IO	I	PU/PD	8mA	DVDD_PD4
B4	uart3_rxd	IO	I	PU/PD	8mA	DVDD_PD4

A3	uart3_cts	IO	I	PU/PD	8mA	DVDD_PD4
B3	uart3_rts	IO	I	PU/PD	8mA	DVDD_PD4
B2	GPIO24	IO	I	PU/PD	8mA	DVDD_PD4
C1	GPIO25	IO	I	PU/PD	8mA	DVDD_PD4
C2	GPIO26	IO	I	PU/PD	8mA	DVDD_PD4
C3	GPIO27	IO	I	PU/PD	8mA	DVDD_PD4
D3	GPIO28	IO	I	PU/PD	8mA	DVDD_PD4
D2	GPIO29	IO	I	PU/PD	8mA	DVDD_PD4
D1	GPIO30	IO	I	PU/PD	8mA	DVDD_PD4
C4	GPIO31	IO	I	PU/PD	8mA	DVDD_PD4
E3	GPIO32	IO	I	PU/PD	8mA	DVDD_PD4
E2	GPIO33	IO	I	PU/PD	8mA	DVDD_PD4
D4	GPIO34	IO	I	PU/PD	8mA	DVDD_PD4
F3	GPIO35	IO	I	PU/PD	8mA	DVDD_PD4
F2	GPIO38	IO	I	PU/PD	8mA	DVDD_PD4
G2	GPIO39	IO	I	PU/PD	8mA	DVDD_PD4
G3	testmode	IO	I	PU/PD	8mA	DVDD_PD4
E4,F4	DVDD_PD4	P	NA	NA	NA	
J5	AVDD09_PLL	P	NA	NA	NA	
H5	AVSS_PLL	G	NA	NA	NA	
E7	VQPS	P	NA	NA	NA	
R1,R2,R3,R4,R5,R18,G4,H4,J4,T4,G5,A6,B6,C6,D6,E6,E8,E9,V10,V11,E12,V14,W14,F18,G18,T18,F19,T19,F20,T20,F21,T21,T22,F22	VDD	P	NA	NA	NA	
B1,A2,W4,A5,B5,C5,D5,E5,F5,V6,V7,V8,V9,K10,L10,M10,N10,K11,L11,M11,N11,K12,L12,M12,N12,K13,L13,M13,N13,E16,E17,E18,E19,E20,E21,E22,W17,H18,J18,N18,P18,U18,V18,U19,V19	VSS	G	NA	NA	NA	

Ball Function Description

Ball Name	Function Description
GPIO0	MCU_gpio0/gpio0/pad_eth_tx_clk
GPIO1	MCU_gpio1/gpio1/eth_phy_txen
GPIO2	MCU_gpio2/gpio2/eth_phy_txd0
GPIO3	MCU_gpio3/gpio3/eth_phy_txd1
GPIO4	MCU_gpio4/gpio4/eth_phy_txd2
GPIO5	MCU_gpio5/gpio5/eth_phy_txd3
GPIO6	MCU_gpio6/gpio6/pad_eth_rx_clk
GPIO7	MCU_gpio7/gpio7/eth_phy_rxdv
GPIO8	MCU_gpio8/gpio8/eth_phy_rxd0
GPIO9	MCU_gpio9/gpio9/eth_phy_rxd1
GPIO10	MCU_gpio10/gpio10/eth_phy_rxd2
GPIO11	MCU_gpio11/gpio11/eth_phy_rxd3
GPIO12	MCU_gpio12/gpio12/pad_eth_rmii_clk
GPIO13	MCU_gpio13/gpio13/eth_rmii_mdc
GPIO14	MCU_gpio14/gpio14/eth_rmii_mdio
GPIO15	MCU_gpio15/gpio15
GPIO16	MCU_gpio16/gpio16
GPIO17	MCU_gpio17/gpio17
GPIO18	MCU_gpio18/gpio18
GPIO19	MCU_gpio19/gpio19
GPIO20	MCU_gpio20/gpio20
GPIO21	MCU_gpio21/gpio21
GPIO22	MCU_gpio22/gpio22
GPIO23	MCU_gpio23/gpio23
GPIO24	MCU_gpio24/gpio24
GPIO25	MCU_gpio25/gpio25
GPIO26	MCU_gpio26/gpio26
GPIO27	MCU_gpio27/gpio27
GPIO28	MCU_gpio28/gpio28
GPIO29	MCU_gpio29/gpio29
GPIO30	MCU_gpio30/gpio30
GPIO31	MCU_gpio31/gpio31
GPIO32	MCU_gpio32/gpio32
GPIO33	MCU_gpio33/gpio33

GPIO34	MCU_gpio34/gpio34
GPIO35	MCU_gpio35/gpio35
GPIO36	MCU_gpio36/gpio36
GPIO37	MCU_gpio37/gpio37
GPIO38	MCU_gpio38/gpio38/MCU_uart0_rxd
GPIO39	MCU_gpio39/gpio39/MCU_uart0_txd
sd1_cd	MCU_gpio58/gpio58/sdmmc1_card_detect_n
sd1_clk	MCU_gpio59/gpio59/sdmmc1_cclk_out
sd1_cmd	MCU_gpio60/gpio60/sdmmc1_ccmd
sd1_d0	MCU_gpio61/gpio61/sdmmc1_cdata0
sd1_d1	MCU_gpio62/gpio62/sdmmc1_cdata1
sd1_d2	MCU_gpio63/gpio63/sdmmc1_cdata2
sd1_d3	MCU_gpio64/gpio64/sdmmc1_cdata3
scl1	MCU_gpio65/gpio65/i2c1_scl
sda1	MCU_gpio66/gpio66/i2c1_sda
emmc_cd	MCU_gpio67/sdmmc_card_detect
emmc_clk	MCU_gpio68/sdmmc_cclk_out
emmc_cmd	MCU_gpio69/sdmmc_ccmd
emmc_d0	MCU_gpio70/sdmmc_cdata0
emmc_d1	MCU_gpio71/sdmmc_cdata1
emmc_d2	MCU_gpio72/sdmmc_cdata2
emmc_d3	MCU_gpio73/sdmmc_cdata3
uart0_txd	MCU_gpio74/uart0_txd
uart0_rxd	MCU_gpio75/uart0_rxd
uart0_cts	MCU_gpio76/uart0_cts
uart0_rts	MCU_gpio77/uart0_rts
uart1_txd	gpio67/uart1_txd
uart1_rxd	gpio68/uart1_rxd
uart2_txd	gpio69/uart2_txd
uart2_rxd	gpio70/uart2_rxd
uart3_txd	gpio71/uart3_txd
uart3_rxd	gpio72/uart3_rxd
uart3_cts	gpio73/uart3_cts
uart3_rts	gpio74/uart3_rts
i2s_sync	MCU_gpio78/i2s_sync
i2s_mclk	MCU_gpio79/i2s_mclk
i2s_bclk	MCU_gpio80/i2s_bitclk

i2s_sdout	MCU_gpio81/i2s_sadata
i2s_sdin	MCU_gpio82/i2s_sadata
i2s1_sync	gpio75/i2s1_sync
i2s1_mclk	gpio76/i2s1_mclk
i2s1_bclk	gpio77/ii2s1_bitclk
i2s1_sdout	gpio78/i2s1_sadata
i2s1_sdin	gpio79/i2s1_sadata
ssp_csn0_out	MCU_gpio83/gpio127/ssp_csn0_out
ssp_clkout	MCU_gpio84/ssp_clkout
ssp_d0	MCU_gpio85/ssp_rxd_tmp0/ssp_txd0
ssp_d1	MCU_gpio86/ssp_rxd_tmp1/ssp_txd1
ssp_d2	MCU_gpio87/ssp_rxd_tmp2/ssp_txd2
ssp_d3	MCU_gpio88/ssp_rxd_tmp3/ssp_txd3
ssp2_csn0_out	MCU_gpio124/gpio80/ssp2_csn0_out
ssp2_clkout	gpio81/ssp2_clkout
ssp2_d0	gpio82/ssp2_rxd_tmp0/ssp2_txd0
ssp2_d1	gpio83/ssp2_rxd_tmp1/ssp2_txd1
ssp2_d2	gpio84/ssp2_rxd_tmp2/ssp2_txd2
ssp2_d3	gpio85/ssp2_rxd_tmp3/ssp2_txd3
ssp1_csn	MCU_gpio125/gpio86/ssp1_csn0_out/ssp1_s_cs
ssp1_clk	gpio87/ssp1_clkout/ssp1_s_clkin
ssp1_txd	gpio88/ssp1_txd/ssp1_s_txd
ssp1_rxd	ssp1_s_txd/gpio89/ssp1_rxd/ssp1_s_rxd_0
can0_rx	MCU_gpio89/mcan0_rx
can0_tx	MCU_gpio90/mcan0_tx
can0_stby	MCU_gpio91/mcan0_stby/uart3_de
can1_rx	MCU_gpio92/mcan1_rx
can1_tx	MCU_gpio93/mcan1_tx
can1_stby	MCU_gpio94/mcan1_stby/uart3_re
pwmout0	MCU_gpio95/pwm_out0
pwmout1	MCU_gpio96/pwm_out1
pwmout2	MCU_gpio97/pwm_out2
pwmout3	MCU_gpio98/pwm_out3
pwmout4	MCU_gpio99/pwm_out4/uart3_de
pwmout5	MCU_gpio100/pwm_out5/uart3_re
pwmout6	gpio90/pwm_out6/uart3_de
pwmout7	gpio91/pwm_out7/uart3_re

pwmin0	gpio92/pwm_in0
pwmin1	gpio93/pwm_in1
pwmin2	gpio94/pwm_in2
pwmin3	gpio95/pwm_in3
ntrst	MCU_gpio119/nTRST/uart1_cts
tms	MCU_gpio120/TMS/uart1_rts
tdi	MCU_gpio121/TDI/uart2_cts
tdo	MCU_gpio122/TDO/uart2_rts
tck	MCU_gpio123/TCK
rcrt_in	gpio96/rcrt_in
scl0	gpio97/i2c_scl
sda0	gpio98/i2c_sda
CSI_GPIO_DATAP3	gpio99/itu_vin0/CSI_GPIO_DATAP3
CSI_GPIO_DATAN3	gpio100/itu_vin1/CSI_GPIO_DATAN3
CSI_GPIO_DATAP2	gpio101/itu_vin2/CSI_GPIO_DATAP2
CSI_GPIO_DATAN2	gpio102/itu_vin3/CSI_GPIO_DATAN2
CSI_GPIO_CLKP	gpio103/itu_vin4/CSI_GPIO_CLKP
CSI_GPIO_CLKN	gpio104/itu_vin5/CSI_GPIO_CLKN
CSI_GPIO_DATAP1	gpio105/itu_vin6/CSI_GPIO_DATAP1
CSI_GPIO_DATAN1	gpio106/itu_vin7/CSI_GPIO_DATAN1
CSI_GPIO_DATAP0	gpio107/itu_clk/CSI_GPIO_DATAP0
CSI_GPIO_DATAN0	gpio108/itu_hsync/CSI_GPIO_DATAN0
itu_vsync	gpio109/itu_vsync
mclk_out	gpio110
OUTAN_P	MCU_gpio40/gpio40/LCD_B0/OUTAN_P
OUTAP_P	MCU_gpio41/gpio41/LCD_B1/OUTAP_P
OUTBN_P	MCU_gpio42/gpio42/LCD_B2/OUTBN_P
OUTBP_P	MCU_gpio43/gpio43/LCD_B3/OUTBP_P
OUTCN_P	MCU_gpio44/gpio44/LCD_B4/OUTCN_P
OUTCP_P	MCU_gpio45/gpio45/LCD_B5/OUTCP_P
OUTCKN_P	MCU_gpio46/gpio46/LCD_B6/OUTCKN_P
OUTCKP_P	MCU_gpio47/gpio47/LCD_B7/OUTCKP_P
OUTDN_P	MCU_gpio101/LCD_G0/OUTDN_P
OUTDP_P	MCU_gpio102/LCD_G1/OUTDP_P
OUTAN2_P	MCU_gpio48/gpio48/LCD_G2/OUTAN2_P
OUTAP2_P	MCU_gpio49/gpio49/LCD_G3/OUTAP2_P
OUTBN2_P	MCU_gpio50/gpio50/LCD_G4/OUTBN2_P

OUTBP2_P	MCU_gpio51/gpio51/LCD_G5/OUTBP2_P
OUTCN2_P	MCU_gpio52/gpio52/LCD_G6/OUTCN2_P
OUTCP2_P	MCU_gpio53/gpio53/LCD_G7/OUTCP2_P
OUTCKN2_P	MCU_gpio103/LCD_R0/OUTCKN2_P
OUTCKP2_P	MCU_gpio104/LCD_R1/OUTCKP2_P
OUTDN2_P	MCU_gpio105/LCD_R2/OUTDN2_P
OUTDP2_P	MCU_gpio106/LCD_R3/OUTDP2_P
DSI_GPIO_DATAN0	MCU_gpio107/LCD_R4/DSI_GPIO_DATAN0
DSI_GPIO_DATAP0	MCU_gpio108/LCD_R5/DSI_GPIO_DATAP0
DSI_GPIO_DATAN1	MCU_gpio109/LCD_R6/DSI_GPIO_DATAN1
DSI_GPIO_DATAP1	MCU_gpio110/LCD_R7/DSI_GPIO_DATAP1
DSI_GPIO_CLKN	MCU_gpio54/gpio54/LCD_DE/DSI_GPIO_CLKN
DSI_GPIO_CLKP	MCU_gpio55/gpio55/LCD_CLK/DSI_GPIO_CLKP
DSI_GPIO_DATAN2	MCU_gpio56/gpio56/LCD_VSYNC/DSI_GPIO_DATAN2
DSI_GPIO_DATAP2	MCU_gpio57/gpio57/LCD_HSYNC/DSI_GPIO_DATAP2
ADC0_AUX0	gpio111/ADC0_AUX0
ADC0_AUX1	gpio112/ADC0_AUX1
ADC0_AUX2	gpio113/ADC0_AUX2
ADC0_AUX3	gpio114/ADC0_AUX3
ADC0_AUX4	gpio115/ADC0_AUX4
ADC0_AUX5	gpio116/ADC0_AUX5
ADC0_AUX6	gpio117/ADC0_AUX6
ADC0_AUX7	gpio118/ADC0_AUX7
ADC1_AUX0	gpio119/ADC1_AUX0
ADC1_AUX1	gpio120/ADC1_AUX1
ADC1_AUX2	gpio121/ADC1_AUX2
ADC1_AUX3	gpio122/ADC1_AUX3
ADC1_AUX4	gpio123/ADC1_AUX4
ADC1_AUX5	gpio124/ADC1_AUX5
ADC1_AUX6	gpio125/ADC1_AUX6
ADC1_AUX7	gpio126/ADC1_AUX7
ADC2_AUX0	MCU_gpio111/ADC2_AUX0
ADC2_AUX1	MCU_gpio112/ADC2_AUX1
ADC2_AUX2	MCU_gpio113/ADC2_AUX2
ADC2_AUX3	MCU_gpio114/ADC2_AUX3
ADC2_AUX4	MCU_gpio115/ADC2_AUX4
ADC2_AUX5	MCU_gpio116/ADC2_AUX5

ADC2_AUX6	MCU_gpio117/ADC2_AUX6
ADC2_AUX7	MCU_gpio118/ADC2_AUX7
DSI_GPIO_DATAN3	MCU_gpio128/gpio128/DSI_GPIO_DATAN3
DSI_GPIO_DATAP3	MCU_gpio129/gpio129/DSI_GPIO_DATAP3
GPIO130	MCU_gpio130/gpio130
GPIO131	MCU_gpio131/gpio131
GPIO132	MCU_gpio132/gpio132
GPIO133	MCU_gpio133/gpio133
GPIO134	MCU_gpio134/gpio134
GPIO135	MCU_gpio135/gpio135
GPIO136	MCU_gpio136/gpio136
GPIO137	MCU_gpio137/gpio137
GPIO138	MCU_gpio138/gpio138
GPIO139	MCU_gpio139/gpio139
GPIO140	MCU_gpio140/gpio140
GPIO141	MCU_gpio141/gpio141
GPIO142	MCU_gpio142/gpio142
GPIO143	MCU_gpio143/gpio143

5.Memory Map

模块名	MCU	CPU
securit ROM	32'h0000_0000,32K	not access
MCU iram	32'h0010_0000,288K	not access
IRAM	32'h2000_0000,32K	32'h0000_0000,32KB
Axi sram (4bank)	32'h2010_0000,2MB	32'h0010_0000,2MB
AXI		
axi(DDR2/3)	32'h6000_0000 512MB	32'h6000_0000 512MB
AHB		

ahb_usb	32'h4000_0000,1M	32'h4000_0000,1M
ahb_sdmmc	32'h4010_0000,1M	32'h4010_0000,1M
ahb_dma	32'h4020_0000,1M	32'h4020_0000,1M
ahb_lcd	32'h4030_0000,1M	32'h4030_0000,1M
ahb_sdmmc1	32'h4040_0000,1M	32'h4040_0000,1M
ahb_gpu	32'h4050_0000,1M	32'h4050_0000,1M
ahb_pxp	32'h4060_0000,1M	32'h4060_0000,1M
ahb_mfc	32'h4070_0000,1M	32'h4070_0000,1M
ahb_itu	32'h4080_0000,1M	32'h4080_0000,1M
ahb_qoi	32'h4090_0000,1M	32'h4090_0000,1M
ahb_ble	32'h40a0_0000,1M	32'h40a0_0000,1M
ahb_eth	32'h40b0_0000,1M	32'h40b0_0000,1M
ahb_wrap	32'h40c0_0000,1M	32'h40c0_0000,1M
ahb_spi_4bit_mst0	32'h40d0_0000,512K	32'h40d0_0000,512K
ahb_spi_4bit_mst1	32'h40d8_0000,512K	32'h40d8_0000,512K
ahb_crc	32'h40e0_0000,512K	32'h40e0_0000,512K
ahb_mailbox	32'h40e8_0000,512K	32'h40e8_0000,512K
ahb_sema	32'h40f0_0000,512K	32'h40f0_0000,512K
APB0		
apb_sys	32'h5000_0000,1M	32'h5000_0000,1M
apb_ecspi	32'h5020_0000,1M	32'h5020_0000,1M
apb_i2c	32'h5030_0000,1M	32'h5030_0000,1M
apb_i2c1	32'h5040_0000,1M	32'h5040_0000,1M
apb_uart0	32'h5050_0000,1M	32'h5050_0000,1M
apb_uart1	32'h5060_0000,1M	32'h5060_0000,1M
apb_uart2	32'h5070_0000,1M	32'h5070_0000,1M
apb_uart3	32'h5080_0000,1M	32'h5080_0000,1M
apb_gpio	32'h5090_0000,1M	32'h5090_0000,1M
apb_timer	32'h50a0_0000,1M	32'h50a0_0000,1M
apb_pwm	32'h50b0_0000,1M	32'h50b0_0000,1M
apb_wdt	32'h50c0_0000,1M	32'h50c0_0000,1M
apb_i2s	32'h50d0_0000,1M	32'h50d0_0000,1M
apb_i2s1	32'h50f0_0000,1M	32'h50f0_0000,1M

APB1		
apb_rtc	32'h5100_0000,1M	32'h5100_0000,1M

apb_adc	32'h5110_0000,1M	32'h5110_0000,1M
apb_adc1	32'h5120_0000,1M	32'h5120_0000,1M
apb_adc2	32'h5130_0000,1M	32'h5130_0000,1M
apb_ddrctl	32'h5140_0000,1M	32'h5140_0000,1M
apb_ddrphy	32'h5150_0000,1M	32'h5150_0000,1M
apb_vbus	32'h5160_0000,1M	32'h5160_0000,1M
apb_gbus	32'h5170_0000,1M	32'h5170_0000,1M
apb_chbus	32'h5180_0000,1M	32'h5180_0000,1M
apb_cabus	32'h5190_0000,1M	32'h5190_0000,1M
apb_dsi_host	32'h51a0_0000,1M	32'h51a0_0000,1M
apb_csi_host	32'h51b0_0000,1M	32'h51b0_0000,1M
apb_rcrt	32'h51c0_0000,1M	32'h51c0_0000,1M
apb_ddr_rdc	32'h51d0_0000,1M	32'h51d0_0000,1M
apb_sram_rdc	32'h51e0_0000,1M	32'h51e0_0000,1M

APB2(MCU)		
apb_MCU_sys	32'h5500_0000,1M	not access
apb_MCU_wdt	32'h5510_0000,1M	not access
apb_MCU_timer	32'h5520_0000,1M	not access
apb_MCU_gpio	32'h5530_0000,1M	not access
apb_MCU_dcic	32'h5550_0000,1M	not access
apb_MCU_canfd0	32'h5560_0000,1M	not access
apb_MCU_canfd1	32'h5570_0000,1M	not access
apb_MCU_uart0	32'h5580_0000,1M	not access
apb_MCU_pwm	32'h5590_0000,1M	not access

ahb_MCU_pka	32'h5200_0000,1M	not access
ahb_MCU_ehse	32'h5900_0000,1M	not access
ahb_MCU_trng	32'h5a00_0000,1M	not access

*6.*Interrupt Source

MCU:

MCU_int_src	0	MCU_wdt_intr

	1	MCU_timer0_intr
	2	MCU_timer1_intr
	3	MCU_timer2_intr
	4	MCU_timer3_intr
	5	can0_intr
	6	can1_intr
	7	MCU_gpioa_intr
	8	MCU_gpiob_intr
	9	MCU_gpioc_intr
	10	MCU_gpiod_intr
	11	MCU_spka_intr
	12	MCU_ehse_intr
	13	MCU_trng_intr
	14	MCU_uart0_intr
	15	MCU_pwm_intr
	16-20	MCU_dcic_intr[4:0]
	21-23	ecc_intr[2:0]
	24	ddr_rdc_int_MCU
	25	sram_rdc_int_MCU
	26	ahb_periph_prot_MCU
	27	sema_int_MCU
	28	MCU_gpioe_intr
	29-31	reserv
	32	lcd_intr
	33	mfc_intr
	34	gpu_intr
	35	usb_intr
	36	pxp_intr
	37	dma_intr
	38	sdmcc_intr
	39	sdmmc_intr1
	40	ble_intr
	41	qoi_intr
	42	eth_intr
	43	warp_intr
	44	ddr_rdc_int_CPU
	45	sram_rdc_int_CPU

	46	ahb_periph_prot_CPU
	47	sema_intr
	48	spi0_nand(ssp)（4bit）
	49	spi2（4bit）
	50	i2c
	51	i2c1
	52	uart0
	53	uart1
	54	uart2
	55	uart485
	56	gpioa_intr(0-31)
	57	gpiob_intr(32-63)
	58	gpioc_intr(64-95)
	59	gpiod_intr(96-127)
	60	timer0
	61	timer1
	62	timer2
	63	i2s1_intr
	64	itu_intr
	65	wdt_intr
	66	i2s_intr
	67	rtc_alm_intr
	68	rtc_prd_intr
	69	adc_intr
	70	pwm_intr
	71	timer3
	72	rcrt_intr
	73	ddrctl_int_n
	74	ssp1_intr（1bit）
	75	csi_intr
	76	adc1_intr
	77	adc2_intr
	78	gpioe_intr(128-143)
	79	reserv
	80-95	mailbox_intr[0]

CPU:

CPUint_src	0	lcd_intr
	1	mfc_intr
	2	gpu_intr
	3	usb_intr
	4	pxp_intr
	5	dma_intr
	6	sdmcc_intr
	7	sdmmc_intr1
	8	ble_intr
	9	qoi_intr
	10	eth_intr
	11	warp_intr
	12	ddr_rdc_int_CPU
	13	sram_rdc_int_CPU
	14	ahb_periph_prot_CPU
	15	sema_intr
	16	spi0_nand(ssp)（4bit）
	17	spi2（4bit）
	18	i2c
	19	i2c1
	20	uart0
	21	uart1
	22	uart2
	23	uart485
	24	gpioa_intr(0-31)
	25	gpiob_intr(32-63)
	26	gpioc_intr(64-95)
	27	gpiod_intr(96-127)
	28	timer0
	29	timer1
	30	timer2
	31	i2s1_intr
	32	itu_intr
	33	wdt_intr
	34	i2s_intr

	35	rtc_alm_intr
	36	rtc_prd_intr
	37	adc_intr
	38	pwm_intr
	39	timer3
	40	rcrt_intr
	41	ddrctl_int_n
	42	ssp1_intr（1bit）
	43	csi_intr
	44	adc1_intr
	45	adc2_intr
	46	gpioe_intr(128-143)
	47	rev
	48-63	mailbox_intr[1]

7.System Config Controller

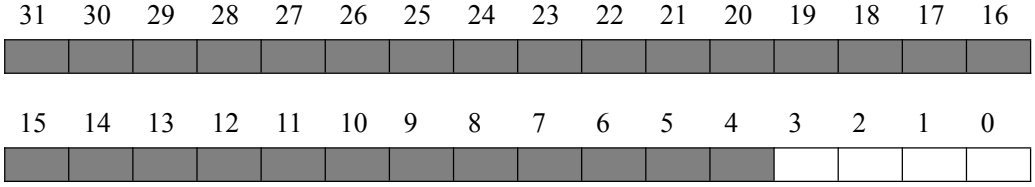
7.1. CPU System Register Descriptions

CPUSYS_BASE: 0x5000_0000

7.1.1 SYS_BOOT_SAMPLE

System Boot Sample

Offset_Address: 0x0000_0000

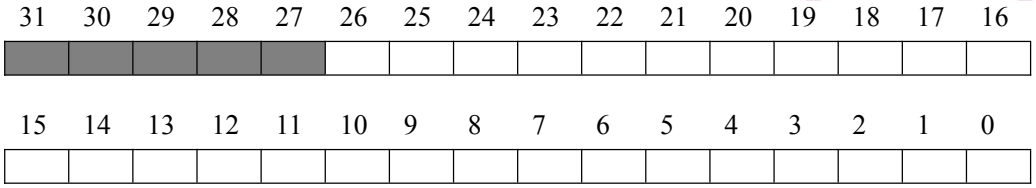


Bit	Type	Reset	Description
31	R	0	lvds lock
30	R	0	cpupll lock
29	R	0	syspll lock
28	R	0	vpupll lock
27	R	0	ddrpll lock
26	R	0	mfcpll lock
25	R	0	ahbpll lock
24	R	0	gpupll lock
23-3	R	0	Not used.

2-0	R		Boot sel

7.1.2 CPUPLL_CFG

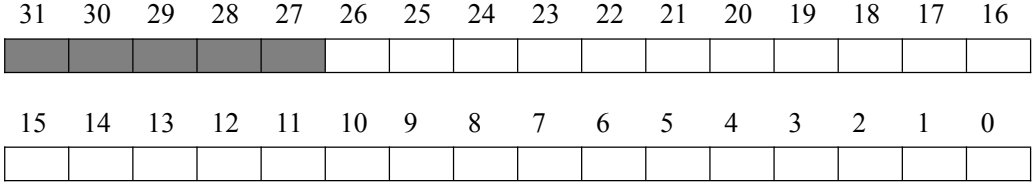
Offset_Address: 0x0000_0004



Bit	Type	Reset	Description
30	R/W	0	**BYPASS**
29	R/W	0	**FOUTVCOPD**
28	R/W	0	**FOUTPOSTDIVPD**
27	R/W	0	**DSMPD**
26	R/W	0	**PD**
25-23	R/W	0	**POSTDIV2**
22-20	R/W	0	**POSTDIV1**
19-8	R/W	0	**FBDIV**
7-6	R	0	**reserved**
5-0	R/W	0	**REFDIV**

7.1.3 SYSPLL_CFG

Offset_Address: 0x0000_0008

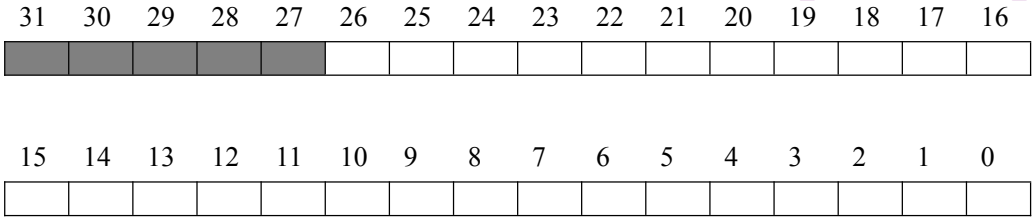


Bit	Type	Reset	Description
30	R/W	0	**BYPASS**
29	R/W	0	**FOUTVCOPD**
28	R/W	0	**FOUTPOSTDIVPD**
27	R/W	0	**DSMPD**
26	R/W	0	**PD**
25-23	R/W	0	**POSTDIV2**
22-20	R/W	0	**POSTDIV1**

19-8	R/W	0	**FBDIV**
7-6	R	0	**reserved**
5-0	R/W	0	**REFDIV**

7.1.4 DDRPLL_CFG

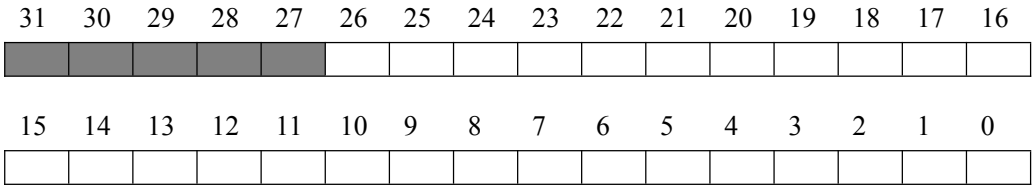
Offset_Address: 0x0000_000C



Bit	Type	Reset	Description
30	R/W	0	**BYPASS**
29	R/W	0	**FOUTVCOPD**
28	R/W	0	**FOUTPOSTDIVPD**
27	R/W	0	**DSMPD**
26	R/W	0	**PD**
25-23	R/W	0	**POSTDIV2**
22-20	R/W	0	**POSTDIV1**
19-8	R/W	0	**FBDIV**
7-6	R	0	**reserved**
5-0	R/W	0	**REFDIV**

7.1.5 VPUPLL_CFG

Offset_Address: 0x0000_0010

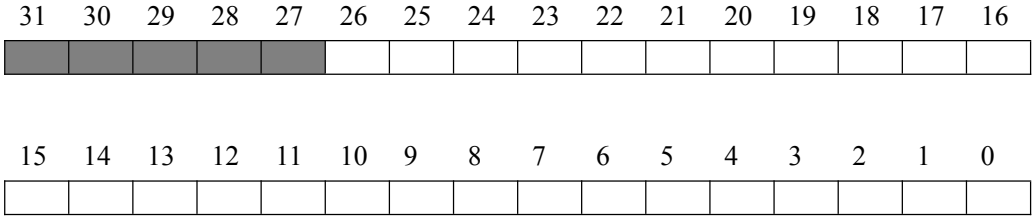


Bit	Type	Reset	Description
30	R/W	0	**BYPASS**
29	R/W	0	**FOUTVCOPD**
28	R/W	0	**FOUTPOSTDIVPD**
27	R/W	0	**DSMPD**

26	R/W	0	**PD**
25-23	R/W	0	**POSTDIV2**
22-20	R/W	0	**POSTDIV1**
19-8	R/W	0	**FBDIV**
7-6	R	0	**reserved**
5-0	R/W	0	**REFDIV**

7.1.6 MFCPLL_CFG

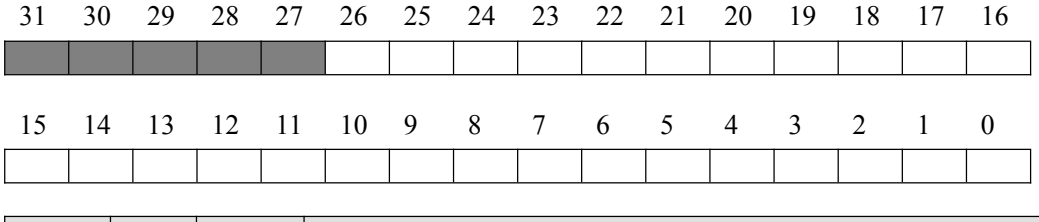
Offset_Address: 0x0000_0014



Bit	Type	Reset	Description
30	R/W	0	**BYPASS**
29	R/W	0	**FOUTVCOPD**
28	R/W	0	**FOUTPOSTDIVPD**
27	R/W	0	**DSMPD**
26	R/W	0	**PD**
25-23	R/W	0	**POSTDIV2**
22-20	R/W	0	**POSTDIV1**
19-8	R/W	0	**FBDIV**
7-6	R	0	**reserved**
5-0	R/W	0	**REFDIV**

7.1.7 AHBPLL_CFG

Offset_Address: 0x0000_0018

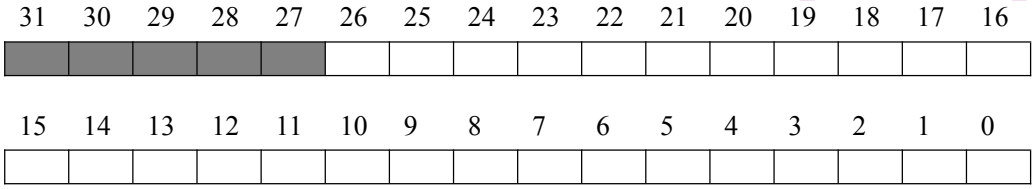


Bit	Type	Reset	Description
30	R/W	0	**BYPASS**
29	R/W	0	**FOUTVCOPD**
28	R/W	0	**FOUTPOSTDIVPD**

27	R/W	0	**DSMPD**
26	R/W	0	**PD**
25-23	R/W	0	**POSTDIV2**
22-20	R/W	0	**POSTDIV1**
19-8	R/W	0	**FBDIV**
7-6	R	0	**reserved**
5-0	R/W	0	**REFDIV**

7.1.8 GPUPLL_CFG

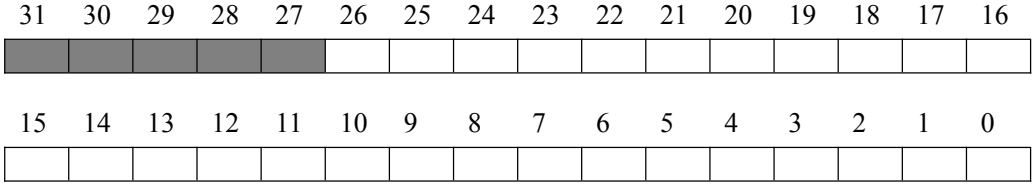
Offset_Address: 0x0000_001C



Bit	Type	Reset	Description
30	R/W	0	**BYPASS**
29	R/W	0	**FOUTVCOPD**
28	R/W	0	**FOUTPOSTDIVPD**
27	R/W	0	**DSMPD**
26	R/W	0	**PD**
25-23	R/W	0	**POSTDIV2**
22-20	R/W	0	**POSTDIV1**
19-8	R/W	0	**FBDIV**
7-6	R	0	**reserved**
5-0	R/W	0	**REFDIV**

7.1.9 SYSPLL_SPRD_CTL

Offset_Address: 0x0000_0020

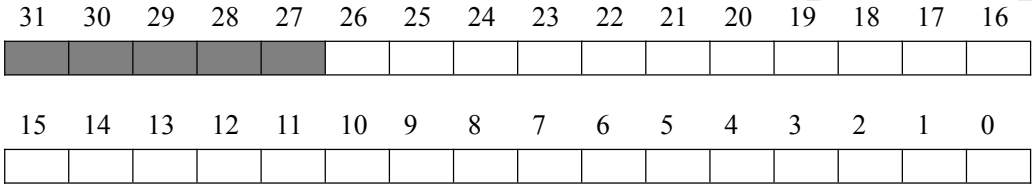


Bit	Type	Reset	Description
31	R/W	0	srst_sprd
30	R/W	0	bypass
29-28	R	0	reserved

27-16	R/W	0	max_offset
15-12	R	0	reserved
11-0	R/W	0	sprd_val

7.1.10 AHBPLL_SPRD_CTL

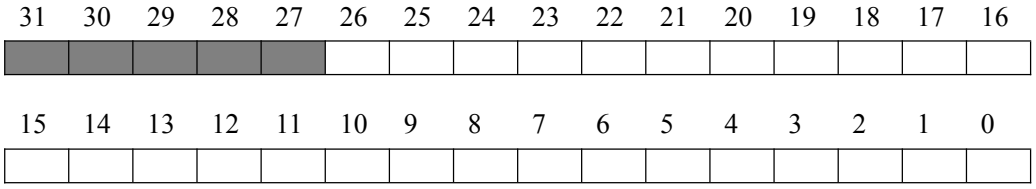
Offset_Address: 0x0000_0024



Bit	Type	Reset	Description
31	R/W	0	srst_sprd
30	R/W	0	bypass
29-28	R	0	reserved
27-16	R/W	0	max_offset
15-12	R	0	reserved
11-0	R/W	0	sprd_val

7.1.11 VPUPLL_SPRD_CTL

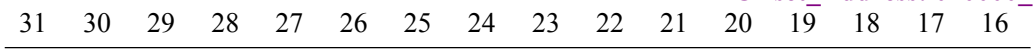
Offset_Address: 0x0000_0028

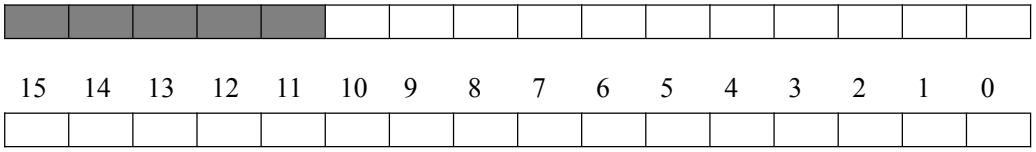


Bit	Type	Reset	Description
31	R/W	0	srst_sprd
30	R/W	0	bypass
29-28	R	0	reserved
27-16	R/W	0	max_offset
15-12	R	0	reserved
11-0	R/W	0	sprd_val

7.1.12 WDTRST_REG

Offset_Address: 0x0000_0030

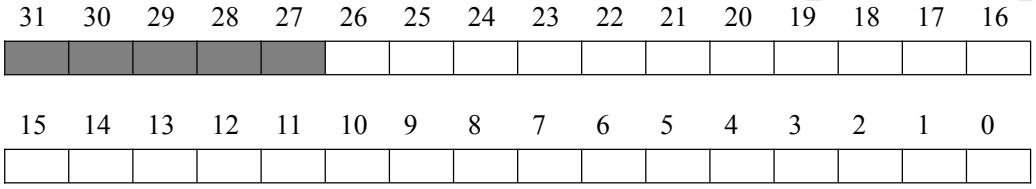




Bit	Type	Reset	Description
31-0	R/W	0	Set by software to identify whether reset is from watchdog or pad.

7.1.13 BUS_CLK_CFG

Offset_Address: 0x0000_0040

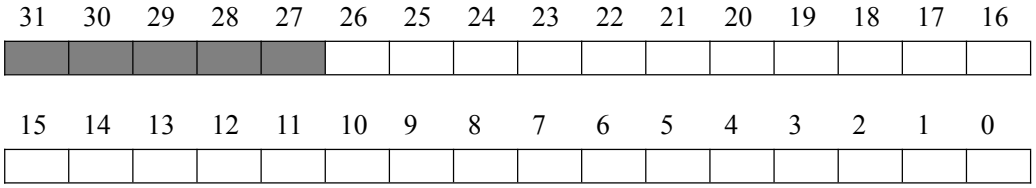


Bit	Type	Reset	Description
31-27	R	0	reserved.
26	R/W	0	**ext2_clk_sel** 0: clk_24m 1: int_ext2_clk_div
25-22	R/W	0	**ext2_clk_div** ext2_clk = int_ext2_clk/(ext2_clk_div+1)
21-20	R/W	0	**Int_ext2_clk_sel** 2'b00: syspll_clk 2'b01: gpupll_clk 2'b10: mfcpll_clk 2'b11: ddrpll_clk
19	R/W	0	**ext1_clk_sel** 0: clk_24m 1: int_ext1_clk_div
18-15	R/W	0	**ext1_clk_div** ext1_clk = int_ext1_clk/(ext1_clk_div+1)
14-13	R/W	0	**Int_ext1_clk_sel** 2'b00: cpupll_clk 2'b01: syspll_clk 2'b10: vpupll_clk 2'b11: ahbpll_clk
12-6	R	0	**reserved**
5-4	R/W	0	**pclk_sel** 2'b00: hclk 2'b01: hclk_div2 2'b10: hclk_div4
3	R/W	0	**hclk_sel**

			1’b0: osc_clk; 1’b1: gated_ahbpll_clk;
2	R/W	0	**xclk_sel** 1’b0: osc_clk; 1’b1: gated_syspll_clk;
1	R/W	0	**dclk_sel** 1’b0: osc_clk; 1’b1: gated_ddrpll_clk;
0	R/W	0	**cclk_sel** 1’b0: osc_clk; 1’b1: gated_cpupll_clk;

7.1.14 BUS_CLK_CFG1

Offset_Address: 0x0000_0180

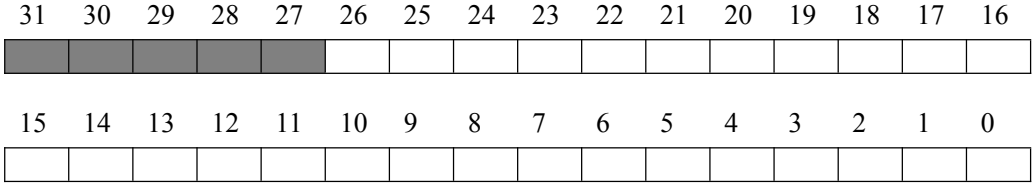


Bit	Type	Reset	Description
31-24	R	0	reserved.
23-19	R/W	0	**qoi_clk_div** qoi_clk = int_qoi_clk/(qoi_clk_div+1)
18	R	0	reserved.
17-16	R/W	0	**qoi_clk_sel** 2’b00: gated_syspll_clk 2’b01: gated_mfcpll_clk 2’b10: osc_clk
15-14	R	0	reserved.
13	R/W	0	**eth_tx_clk_mux_pad** 0: eth_txrx_clk_mux; 1: eth_txrx_clk_mux_inv;
12	R/W	0	**eth_txrx_clk_src_mux** 0: pad_eth_rx_clk; 1: eth_tx_clk_mux
11	R/W	0	**eth_rx_clk_mux** 0: eth_rx_clk_mux; 1: eth_rx_clk_mux_inv;
10	R/W	0	**eth_rx_clk_src_mux** 0: pad_eth_rx_clk; 1: pad_eth_rmii_clk div2;
9	R/W	0	**eth_tx_clk_mux** 0: eth_txrx_clk_mux; 1: eth_txrx_clk_mux_inv;

8	R/W	0	**eth_tx_clk_src_mux** 0: pad_eth_tx_clk; 1: pad_eth_rmii_clk div2;
7	R	0	reserved.
6	R/W	0	**i2s1_clk_sel** 1'b0: input clk from pad; 1'b1: inverted input clk from pad;
5	R	0	reserved.
4	R/W	0	**i2s1_clk_src_sel** 1'b0: gated_syspll_clk; 1'b1: gated_ahbpll_clk;
3-1	R	0	reserved.
0	R/W	0	**dec_pclk_sel** 0: mfc_clk_div2; 1: mfc_clk_div4;

7.1.15 PER_CLK_CFG

Offset_Address: 0x0000_0044

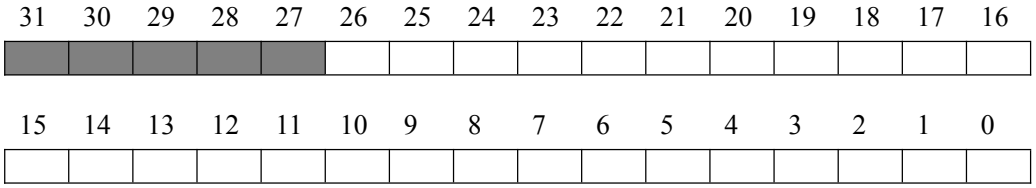


Bit	Type	Reset	Description
31-19	R	0	reserved.
18	R/W	0	**bitclk_i2s_mux0** 1'b0: i2s_codec_blkin; 1'b1: i2s_codec_blkin_inv;
17	R	0	reserved.
16	R/W	0	**i2s_clk_src_mux** 1'b0: gated_syspll_clk; 1'b1: gated_ahbpll_clk;
15-13	R	0	reserved.
12	R/W	0	**gpio_debnc_clk_sel** 1'b0: osc_clk; 1'b1: rtc_clk_32k;
11-9	R	0	reserved.
8	R/W	0	**pwm_src_sel** 1'b0: osc_clk; 1'b1: gated_syspll_clk;
7-4	R/W	0	**pwm_clk_div** pwm_clk = int_pwm_clk/(pwm_clk_div+1)

3	R/W	0	**rcrt_clk_mux** 1'b0: osc_clk; 1'b1: int_rcrt_clk_div;
2-0	R/W	0	**rcrt_clk_div** Int_rcrt_clk_div = osc_clk/(rcrt_clk_div+1)

7.1.16 SDMMC_CLK_CFG

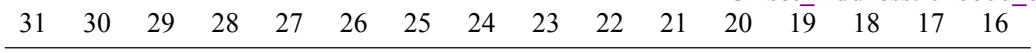
Offset_Address: 0x0000_0048

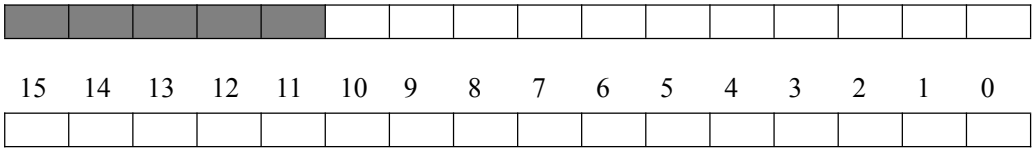


Bit	Type	Reset	Description
31-27	R	0	reserved.
26-20	R/W	0	**sdmmc_drv_dly** delay sel 128 steps, 200ps/step;
19-13	R/W	0	**sdmmc_sample_dly** delay sel 128 steps, 200ps/step;
12-9	R	0	reserved.
8	R/W	0	**sdmmc_clkdrv_edge_mux** int_sdmmc_cclk_in_drv 1'b0: int_sdmmc_clk_in_inv; 1'b1: int_sdmmc_clk_in;
7	R/W	0	**sdmmc_clk_switch_sel** Int_sdmmc_clk_switch 1'b0: osc_clk; 1'b1: gated_ahbpll_clk;
6	R/W	0	**sdmmc_clk_mux** Int_sdmmc_cclk_in 1'b0: int_sdmmc_clk_switch; 1'b1: int_sdmmc_clk_div;
5	R/W	0	**Int_sdmmc_clk_edge_sel** 1'b0: int_sdmmc_cclk_in; 1'b1: int_sdmmc_cclk_in_inv;
4-0	R/W	0	**sdmmc_clk_div** Int_sdmmc_clk_div = int_sdmmc_clk_switch/(sdmmc_clk_div+1)

7.1.17 VOU_CLK_CFG

Offset_Address: 0x0000_004C

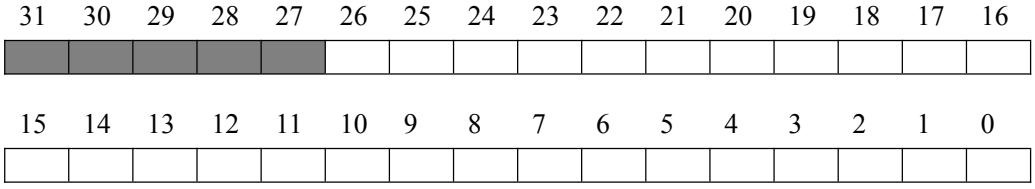




Bit	Type	Reset	Description
31-26	R	0	reserved.
25-19	R/W	0	**tvout_lcd_clk_out_dly** delay sel 128 steps, 200ps/step;
18-16	R/W	0	**tvout_lcd_div** tvout_lcd_clk_pix_wire = tvout_lcd_clk/(tvout_lcd_div+1)
15-10	R	0	reserved.
9	R/W	0	**tv_lcd_pixel_clk_switch** tvout_lcd_pixel_clk 1'b0: tvout_lcd_clk; 1'b1: tvout_lcd_clk_pix_wire;
8	R/W	0	**tv_lcd_clk_switch** tvout_lcd_clk_out_int 1'b0: tvout_lcd_clk; 1'b1: tvout_lcd_clk_inv;
7-3	R/W	0	**tv_clk_div** lcd_clk_div = int_tv_clk_switch/(tv_clk_div+1)
2	R	0	reserved.
1-0	R/W	0	**lcd_clk_src_sel** Int_tv_clk_switch 2'b00: gated_syspll_clk; 2'b01: gated_vpupll_clk; 2'b10: osc_clk;

7.1.18 BUS_CLK_EN

Offset_Address: 0x0000_0050



Bit	Type	Reset	Description
31	R/W	1	**wrap_hclk_en** 1'b0: disable; 1'b1: enable;

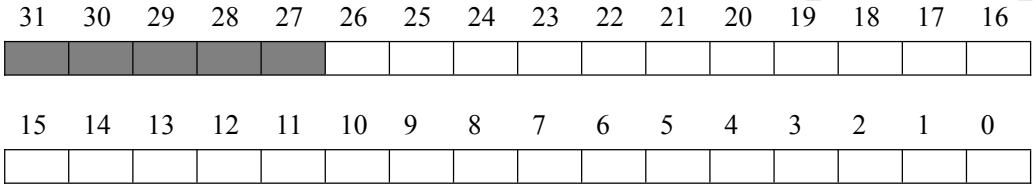
30	R/W	1	**mailbox_hclk_en** 1'b0: disable; 1'b1: enable;
29	R/W	1	**sema_hclk_en** 1'b0: disable; 1'b1: enable;
28	R/W	1	**crc_hclk_en** 1'b0: disable; 1'b1: enable;
27	R/W	1	**ssp2_hclk_en** 1'b0: disable; 1'b1: enable;
26	R/W	1	**ssp_hclk_en** 1'b0: disable; 1'b1: enable;
25	R/W	1	**ddr_3_hclk_en** 1'b0: disable; 1'b1: enable;
24	R/W	1	**ddr_2_hclk_en** 1'b0: disable; 1'b1: enable;
23	R/W	1	**eth_hclk_en** 1'b0: disable; 1'b1: enable;
22	R/W	1	**ble_hclk_en** 1'b0: disable; 1'b1: enable;
21	R/W	1	**mfc_hclk_en** 1'b0: disable; 1'b1: enable;
20	R/W	1	**qoi_hclk_en** 1'b0: disable; 1'b1: enable;
19	R/W	1	**gpu_hclk_en** 1'b0: disable; 1'b1: enable;

18	R/W	1	**itu_hclk_en** 1'b0: disable; 1'b1: enable;
17	R/W	1	**lcd_hclk_en** 1'b0: disable; 1'b1: enable;
16	R/W	1	**sdmmc1_hclk_en** 1'b0: disable; 1'b1: enable;
15	R/W	1	**sdmmc_hclk_en** 1'b0: disable; 1'b1: enable;
14	R/W	1	**usb_hclk_en** 1'b0: disable; 1'b1: enable;
13	R/W	1	**dma_hclk_en** 1'b0: disable; 1'b1: enable;
12	R/W	1	**pxp_hclk_en** 1'b0: disable; 1'b1: enable;
11	R/W	1	**h2xusb_hclk_en** 1'b0: disable; 1'b1: enable;
10	R/W	1	**h2xdma_hclk_en** 1'b0: disable; 1'b1: enable;
9	R/W	1	**ddr_1_xclk_en** 1'b0: disable; 1'b1: enable;
8	R/W	1	**wrap_xclk_en** 1'b0: disable; 1'b1: enable;
7	R/W	1	**pxp_xclk_en** 1'b0: disable; 1'b1: enable;
6	R/W	1	**ble_xclk_en** 1'b0: disable; 1'b1: enable;
5	R/W	1	**qoi_xclk_en** 1'b0: disable; 1'b1: enable;
4	R/W	1	**mfc_xclk_en** 1'b0: disable; 1'b1: enable;
3	R/W	1	**gpu_xclk_en** 1'b0: disable; 1'b1: enable;

2	R/W	1	**ddr_0_xclk_en** 1'b0: disable; 1'b1: enable;
1	R/W	1	**itu_xclk_en** 1'b0: disable; 1'b1: enable;
0	R/W	1	**lcd_xclk_en** 1'b0: disable; 1'b1: enable;

7.1.19 BUS_CLK_EN1

Offset_Address: 0x0000_0054



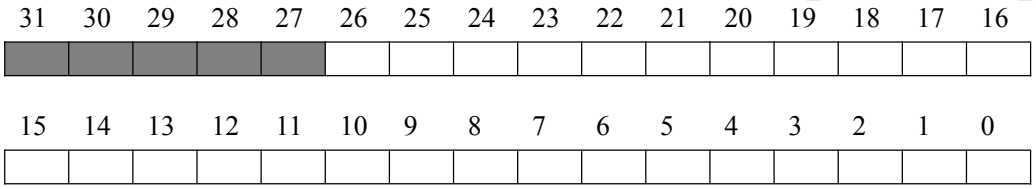
Bit	Type	Reset	Description
31-28	R	1	reserved.
27	R/W	1	**ddrctl_clk_en** 1'b0: disable; 1'b1: enable;
26	R/W	1	**adc2_pclk_en** 1'b0: disable; 1'b1: enable;
25	R/W	1	**adc1_pclk_en** 1'b0: disable; 1'b1: enable;
24	R/W	1	**csi_pclk_en** 1'b0: disable; 1'b1: enable;
23	R/W	1	**dsi_pclk_en** 1'b0: disable; 1'b1: enable;
22	R/W	1	**cabus_pclk_en** 1'b0: disable; 1'b1: enable;

21	R/W	1	**chbus_pclk_en** 1'b0: disable; 1'b1: enable;
20	R/W	1	**gbus_pclk_en** 1'b0: disable; 1'b1: enable;
19	R/W	1	**vbus_pclk_en** 1'b0: disable; 1'b1: enable;
18	R/W	1	**ddrphy_pclk_en** 1'b0: disable; 1'b1: enable;
17	R/W	1	**ddrctl_pclk_en** 1'b0: disable; 1'b1: enable;
16	R	1	reserved.
15	R/W	1	**pwm_pclk_en** 1'b0: disable; 1'b1: enable;
14	R/W	1	**uart3_pclk_en** 1'b0: disable; 1'b1: enable;
13	R/W	1	**uart2_pclk_en** 1'b0: disable; 1'b1: enable;
12	R/W	1	**uart1_pclk_en** 1'b0: disable; 1'b1: enable;
11	R/W	1	**uart0_pclk_en** 1'b0: disable; 1'b1: enable;
10	R/W	1	**gpio_pclk_en** 1'b0: disable; 1'b1: enable;
9	R/W	1	**rcrt_pclk_en** 1'b0: disable; 1'b1: enable;
8	R/W	1	**rtc_pclk_en** 1'b0: disable; 1'b1: enable;
7	R/W	1	**timer_pclk_en** 1'b0: disable; 1'b1: enable;
6	R/W	1	**wdt_pclk_en** 1'b0: disable;

			1’b1: enable;
5	R/W	1	**adc_pclk_en** 1’b0: disable; 1’b1: enable;
4	R/W	1	**i2c1_pclk_en** 1’b0: disable; 1’b1: enable;
3	R/W	1	**i2c_pclk_en** 1’b0: disable; 1’b1: enable;
2	R/W	1	**ssp1_pclk_en** 1’b0: disable; 1’b1: enable;
1	R/W	1	**i2s1_pclk_en** 1’b0: disable; 1’b1: enable;
0	R/W	1	**i2s_pclk_en** 1’b0: disable; 1’b1: enable;

7.1.20 PER_CLK_EN

Offset_Address: 0x0000_0058



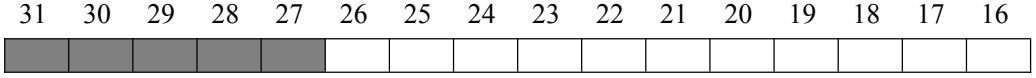
Bit	Type	Reset	Description
31	R/W	1	**sdmmc1_clk_en** 1’b0: disable; 1’b1: enable;
30	R	1	reserved.
29	R/W	1	**ssp2_clk_en** 1’b0: disable; 1’b1: enable;
28	R/W	1	**timer3_clk_en** 1’b0: disable; 1’b1: enable;
27	R/W	1	**timer2_clk_en** 1’b0: disable; 1’b1: enable;

26	R/W	1	**timer1_clk_en** 1'b0: disable; 1'b1: enable;
25	R/W	1	**itu_clk_en** 1'b0: disable; 1'b1: enable;
24	R/W	1	**gpu_clk_en** 1'b0: disable; 1'b1: enable;
23	R/W	1	**mfc_clk_en** 1'b0: disable; 1'b1: enable;
22	R/W	1	**usbphy_clk_en** 1'b0: disable; 1'b1: enable;
21	R/W	1	**usb_12m_en** 1'b0: disable; 1'b1: enable;
20	R/W	1	**bist1_clk_en** 1'b0: disable; 1'b1: enable;
19	R/W	1	**bist_clk_en** 1'b0: disable; 1'b1: enable;
18	R/W	1	**lcd_clk_pix_en** 1'b0: disable; 1'b1: enable;
17	R/W	1	**lcd_clk_out_en** 1'b0: disable; 1'b1: enable;
16	R/W	1	**lcd_clk_en** 1'b0: disable; 1'b1: enable;
15	R/W	1	**sdmmc_clk_en** 1'b0: disable; 1'b1: enable;
14	R/W	1	**rcrt_clk_en** 1'b0: disable; 1'b1: enable;

13	R/W	1	**adc_clk_en** 1'b0: disable; 1'b1: enable;
12	R/W	1	**i2s_bclk_en** 1'b0: disable; 1'b1: enable;
11	R/W	1	**i2s_mclk_en** 1'b0: disable; 1'b1: enable;
10	R/W	1	**pwm_clk_en** 1'b0: disable; 1'b1: enable;
9	R/W	1	**timer_clk_en** 1'b0: disable; 1'b1: enable;
8	R/W	1	**dbnc_clk_en** 1'b0: disable; 1'b1: enable;
7	R/W	1	**uart3_clk_en** 1'b0: disable; 1'b1: enable;
6	R/W	1	**uart2_clk_en** 1'b0: disable; 1'b1: enable;
5	R/W	1	**uart1_clk_en** 1'b0: disable; 1'b1: enable;
4	R/W	1	**uart0_clk_en** 1'b0: disable; 1'b1: enable;
3	R/W	1	**i2c1_clk_en** 1'b0: disable; 1'b1: enable;
2	R/W	1	**i2c_clk_en** 1'b0: disable; 1'b1: enable;
1	R/W	1	**ssp1_clk_en** 1'b0: disable; 1'b1: enable;
0	R/W	1	**ssp_clk_en** 1'b0: disable; 1'b1: enable;

7.1.21 PER_CLK_EN1

Offset_Address: 0x0000_0184



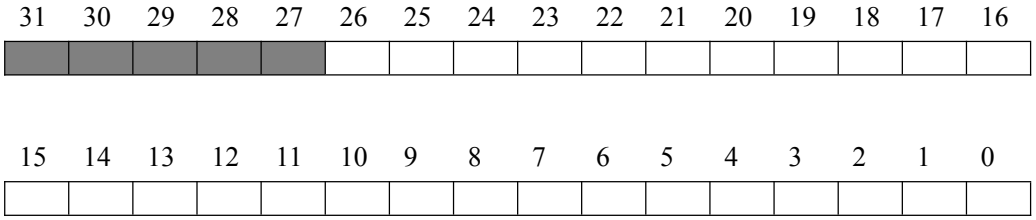
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-16	R	1	reserved.
15	R/W	1	**adc2_clk_en** 1'b0: disable; 1'b1: enable;
14	R/W	1	**adc1_clk_en** 1'b0: disable; 1'b1: enable;
13	R/W	1	**csipix_clk_en** 1'b0: disable; 1'b1: enable;
12	R/W	1	**dsi_refclk_en** 1'b0: disable; 1'b1: enable;
11	R/W	1	**csi_refclk_en** 1'b0: disable; 1'b1: enable;
10	R/W	1	**eth_ptp_refclk_en** 1'b0: disable; 1'b1: enable;
9	R/W	1	**eth_rmii_clk_en** 1'b0: disable; 1'b1: enable;
8	R/W	1	**eth_rx_clk_en** 1'b0: disable; 1'b1: enable;
7	R/W	1	**eth_tx_clk_en** 1'b0: disable; 1'b1: enable;
6	R/W	1	**ble_clk_en** 1'b0: disable; 1'b1: enable;
5	R/W	1	**qoi_clk_en** 1'b0: disable; 1'b1: enable;
4	R/W	1	**bist_clk4_en** 1'b0: disable; 1'b1: enable;
3	R/W	1	**bist_clk3_en** 1'b0: disable; 1'b1: enable;
2	R/W	1	**bist_clk2_en** 1'b0: disable; 1'b1: enable;
1	R/W	1	**i2s1_bclk_en**

			1'b0: disable; 1'b1: enable;
0	R/W	1	**i2s1_mclk_en** 1'b0: disable; 1'b1: enable;

7.1.22 SOFTRESET_CTL

Offset_Address: 0x0000_005C

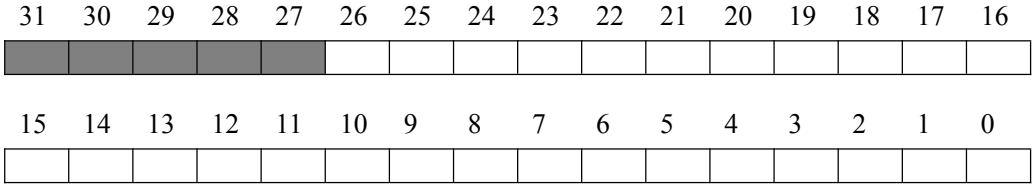


Bit	Type	Reset	Description
31	R/W	1	**softreset_sdmmc1**
30	R/W	1	**softreset_i2s1**
29	R/W	1	**softreset_qoi_core**
28	R/W	1	**softreset_qoi**
27	R	1	**reserved**
26	R/W	1	**softreset_rcrt**
25	R/W	1	**softreset_adc**
24	R/W	1	**softreset_rtc**
23	R/W	1	**softreset_i2s**
22	R/W	1	**softreset_wdt**
21	R/W	1	**softreset_pwm**
20	R/W	1	**softreset_timer3**
19	R/W	1	**softreset_timer2**
18	R/W	1	**softreset_timer1**
17	R/W	1	**softreset_timer0**
16	R/W	1	**softreset_gpio**
15	R/W	1	**softreset_uart3**
14	R/W	1	**softreset_uart2**
13	R/W	1	**softreset_uart1**
12	R/W	1	**softreset_uart0**
11	R/W	1	**softreset_i2c1**
10	R/W	1	**softreset_i2c**

9	R/W	1	**softreset_ssp1**
8	R/W	1	**softreset_ssp**
7	R/W	1	**softreset_gpu_core**
6	R/W	1	**softreset_gpu**
5	R/W	1	**softreset_itu**
4	R/W	1	**softreset_sdmmc**
3	R/W	1	**softreset_usb**
2	R/W	1	**softreset_pxp**
1	R/W	1	**softreset_dma**
0	R/W	1	**softreset_lcd**

7.1.23 SOFTRESET_CTL1

Offset_Address: 0x0000_0060

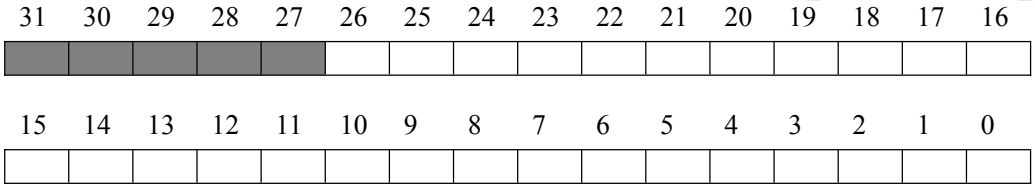


Bit	Type	Reset	Description
31	R/W	1	**softreset_cabuspclk**
30	R/W	1	**softreset_chbuspclk**
29	R/W	1	**softreset_gbuspclk**
28	R/W	1	**softreset_mailbox**
27	R/W	1	**softreset_sema**
26	R/W	1	**softreset_crc**
25	R/W	1	**softreset_buspclk**
24	R/W	1	**softreset_adc1**
23	R/W	1	**softreset_CPU_cnt**
22	R/W	1	**softreset_usbphy**
21	R/W	1	**softreset_ddrphy**
20	R/W	1	**softreset_ddrctl**
19	R/W	1	**softreset_ddr3**
18	R/W	1	**softreset_ddr2**

17	R/W	1	**softreset_ddr1**
16	R/W	1	**softreset_ddr0**
15-10	R	1	**reserved**
9-7	R/W	1	**softreset_codec**
6	R/W	1	**softreset_eth**
5	R/W	1	**softreset_utmi**
4	R/W	1	**softreset_ble**
3	R/W	1	**softreset_h2xusb**
2	R/W	1	**softreset_h2xdma**
1	R/W	1	**softreset_ssp2**
0	R/W	1	**softreset_mfc**

7.1.24 SOFTRESET_CTL2

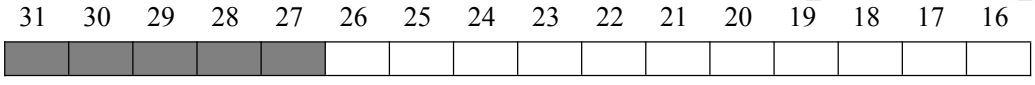
Offset_Address: 0x0000_0188

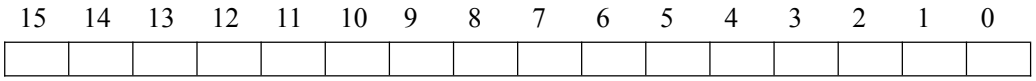


Bit	Type	Reset	Description
31-7	R	1	**reserved**
6	R/W	1	**softreset_csi_pix2xclk**
5	R/W	1	**softreset_csi_pixclk**
4	R/W	1	**softreset_wrap_sram**
3	R/W	1	**softreset_wrap**
2	R/W	1	**softreset_adc2**
1	R/W	1	**softreset_csi**
0	R/W	1	**softreset_dsi**

7.1.25 SSP_CLK_CFG

Offset_Address: 0x0000_0064

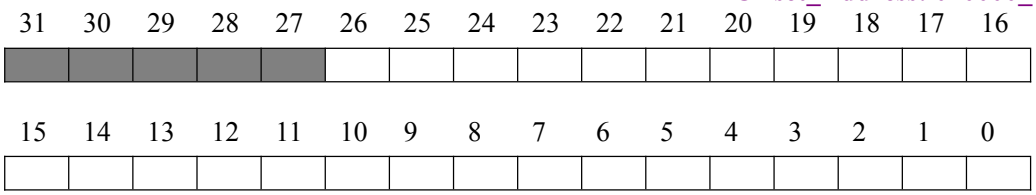




Bit	Type	Reset	Description
31-29	R	0	**reserved**
28	R/W	0	**ssp2_clk_sel** 1'b0: osc_clk; 1'b1: gated_cpupll_clk;
27-24	R/W	0	**ssp2_clk_div** ssp2_clk = int_ssp2_clk/(ssp2_clk_div+1)
23-21	R	0	**reserved**
20	R/W	0	**ssp1_clk_sel** 1'b0: osc_clk; 1'b1: gated_cpupll_clk;
19-16	R/W	0	**ssp1_clk_div** ssp1_clk = int_ssp1_clk/(ssp1_clk_div+1)
15-5	R	0	**reserved**
4	R/W	0	**ssp_clk_sel** 1'b0: osc_clk; 1'b1: gated_cpupll_clk;
3-0	R/W	0	**ssp_clk_div** ssp_clk = int_ssp_clk/(ssp_clk_div+1)

7.1.26 TIMER_CLK_CFG

Offset_Address: 0x0000_0068

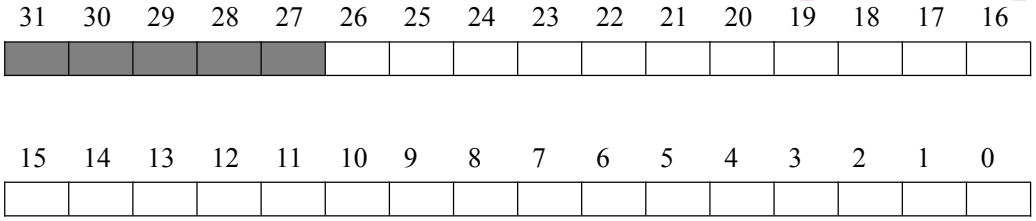


Bit	Type	Reset	Description
31	R	0	**reserved**
30-16	R/W	0	**adc_clk_div** adc_clk = osc_clk/(adc_clk_div +1)*2
15-11	R/W	0	**mfc_clk_div** mfc_clk = int_mfc_clk/(mfc_clk_div+1)
10	R	0	**reserved**
9-8	R/W	0	**mfc_clk_sel** 2'b00: gated_syspll_clk; 2'b01: gated_mfcpll_clk; 2'b10: osc_clk;

7-5	R	0	**reserved**
4	R/W	0	**timer_clk_sel** 1'b0: osc_clk; 1'b1: gated_ahbpll_clk;
3-0	R/W	0	**timer_clk_div** timer_clk = int_timer_clk/(timer_clk_div+1)

7.1.27 I2S_NCO_CFG

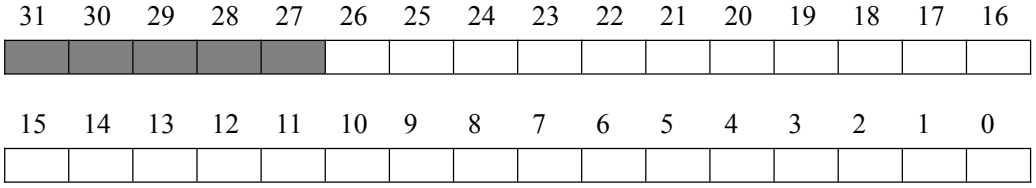
Offset_Address: 0x0000_006C



Bit	Type	Reset	Description
31-16	R/W	0	**step**
15-0	R/W	0	**mod**

7.1.28 SDMMC1_CLK_CFG

Offset_Address: 0x0000_0070

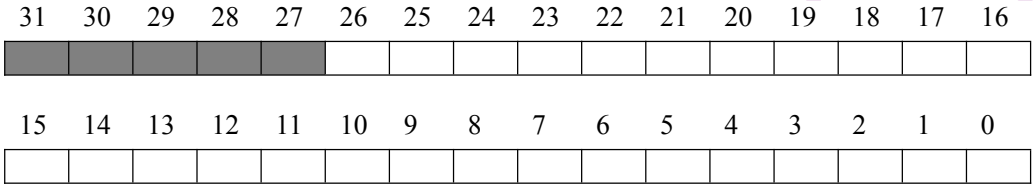


Bit	Type	Reset	Description
31-27	R	0	reserved.
26-20	R/W	0	**sdmmc1_drv_dly** delay sel 128 steps, 200ps/step;
19-13	R/W	0	**sdmmc1_sample_dly** delay sel 128 steps, 200ps/step;
12-9	R	0	reserved.
8	R/W	0	**sdmmc1_clkdrv_edge_mux** int_sdmmc1_cclk_in_drv 1'b0: int_sdmmc1_clk_in_inv; 1'b1: int_sdmmc1_clk_in;
7	R/W	0	**sdmmc1_clk_switch_sel**

			int_sdmmc1_clk_switch 1'b0: osc_clk; 1'b1: gated_ahbpll_clk;
6	R/W	0	**sdmmc1_clk_mux** int_sdmmc1_cclk_in 1'b0: int_sdmmc1_clk_switch; 1'b1: int_sdmmc1_clk_div;
5	R/W	0	**int_sdmmc1_clk_edge_sel** 1'b0: int_sdmmc1_cclk_in; 1'b1: int_sdmmc1_cclk_in_inv;
4-0	R/W	0	**sdmmc1_clk_div** sdmmc1_clk = sdmmc1_clk_switch/(sdmmc1_clk_div +1)

7.1.29 DDR_CTL1_CFG

Offset_Address: 0x0000_0074

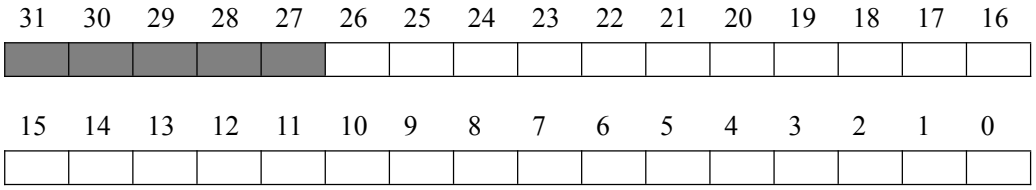


Bit	Type	Reset	Description
31-28	R	0	reserved
27	R	0	ddrphy_byp_stat
26-25	R	0	stat_ddrc_reg_selfref_type
24	R	0	ddrctl_temp_intr_fault
23-18	R	0	reserved
17	R/W	0	**mfc_pclk_en** 1'b0: disable; 1'b1: enable;
16-14	R	0	reserved
13	R/W	0	**gpupll_clk_en** 1'b0: disable; 1'b1: enable;
12	R	0	reserved
11	R/W	0	**ahbpll_clk_en** 1'b0: disable; 1'b1: enable;
10	R	0	reserved
9	R/W	0	**mfcpll_clk_en** 1'b0: disable; 1'b1: enable;
8	R	0	reserved
7	R/W	0	**vpupll_clk_en** 1'b0: disable;

			1’b1: enable;
6	R	0	reserved
5	R/W	0	**ddrpll_clk_en** 1’b0: disable; 1’b1: enable;
4	R	0	reserved
3	R/W	0	**syspll_clk_en** 1’b0: disable; 1’b1: enable;
2	R	0	reserved
1	R/W	0	**cpupll_clk_en** 1’b0: disable; 1’b1: enable;
0	R	0	reserved

7.1.30 PER_CTL_CFG

Offset_Address: 0x0000_0078

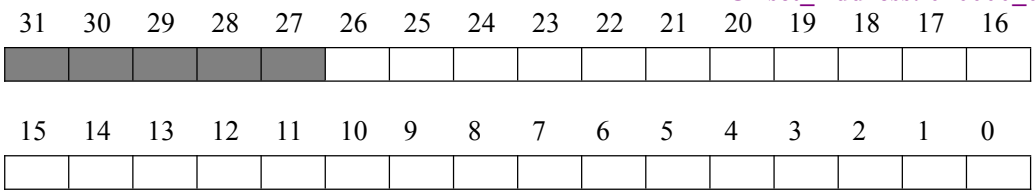


Bit	Type	Reset	Description
31-29	R	0	reserved
28-25	R/W	0	**uart3_clk_div** uart3_clk = int_uart3_clk/(uart3_clk_div+1)
24	R/W	0	**uart3_clk_sel** 1’b0: osc_clk; 1’b1: gated_ahbpll_clk;
23-21	R	0	reserved
20-17	R/W	0	**uart1_clk_div** uart1_clk = int_uart1_clk/(uart1_clk_div+1)
16	R/W	0	**uart1_clk_sel** 1’b0: osc_clk; 1’b1: gated_ahbpll_clk;
15-13	R	0	reserved
12-9	R/W	0	**uart2_clk_div** uart2_clk = int_uart2_clk/(uart2_clk_div+1)
8	R/W	0	**uart2_clk_sel** 1’b0: osc_clk; 1’b1: gated_ahbpll_clk;

7-5	R	0	reserved
4-1	R/W	0	**uart0_clk_div** uart0_clk = int_uart0_clk/(uart0_clk_div+1)
0	R/W	0	**uart0_clk_sel** 1'b0: osc_clk; 1'b1: gated_ahbpll_clk;

7.1.31 TIMER1_CLK_CFG

Offset_Address: 0x0000_007C

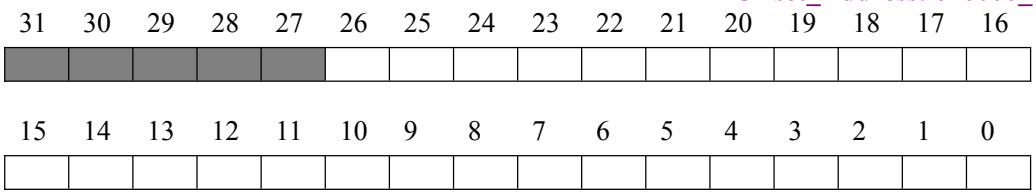


Bit	Type	Reset	Description
31-29	R	0	**reserved**
28	R/W	0	**timer23_clk_sel** 1'b0: osc_clk; 1'b1: gated_syspll_clk;
27-24	R/W	0	**timer23_clk_div** timer23_clk = int_timer23_clk/(timer23_clk_div+1)
23-21	R	0	**reserved**
20	R/W	0	**timer1_clk_sel** 1'b0: osc_clk; 1'b1: gated_ahbpll_clk;
19-16	R/W	0	**timer1_clk_div** timer1_clk = int_timer1_clk/(timer1_clk_div+1)
15-14	R	0	**reserved**
13-12	R/W	0	**itu_data_src_sel** 2'b00: normal; 2'b01: from pad for test; 2'b10: from lcd for test;
11	R	0	**reserved**
10	R/W	0	**itu_clk_padin_sel1** 1'b0: itu_clk_padin_mux0; 1'b1: lcd_clk_pix for test;
9	R/W	0	**itu_clk_padin_sel0** 1'b0: itu_clk_padin; 1'b1: CLK_IN_pad for test;
8	R/W	0	**itu_scaleclk_sel** 1'b0: gated_gpupll_clk; 1'b1: gated_mfcpll_clk;
7-3	R/W	0	**gpu_clk_div**

			gpu_clk = int_gpu_clk/(gpu_clk_div+1)
1-0	R/W	0	**gpu_clk_sel** 2'b00: gated_syspll_clk; 2'b01: gated_gpupll_clk; 2'b10: osc_clk;

7.1.32 ANA_REG

Offset_Address: 0x0000_0080

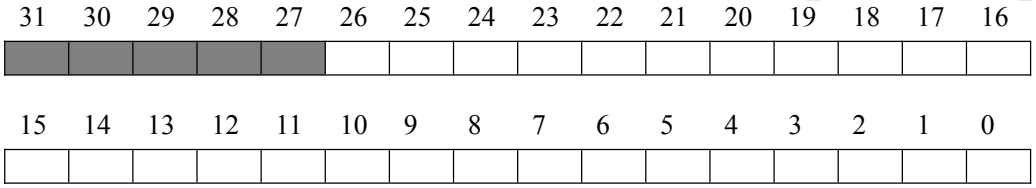


Bit	Type	Reset	Description
31-15	R	0	**reserved**
14	R/W	0	**PD_VREF_TP**
13	R/W	0	**SEL_IRQ_0V9_2**
12	R/W	0	**SEL_IRQ_0V9_1**
11	R/W	0	**SEL_IRQ_0V9_0**
10	R/W	0	**VREF_SWITCH_TH_2**
9	R/W	0	**VREF_SWITCH_TH_1**
8	R/W	0	**VREF_SWITCH_TH_0**
7-6	R	0	**reserved**
5	R/W	0	**usb id reg**
4	R/W	0	**usb phy signal switch**
3	R	0	**reserved**
2	R/W	0	**LPR_TH1_V0P9**
1	R/W	0	**LPR_TH0_V0P9**
0	R/W	0	**LPR_EN_V0P9**

7.1.33 ANA_REG1

Configure usb phy.

Offset_Address: 0x0000_0084

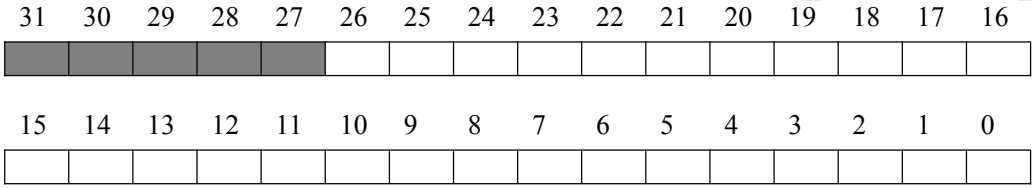


Bit	Type	Reset	Description
31-30	R/W	0	**TXHSXVTUNE0**
29-27	R/W	0	**OTGTUNE0**
26-24	R/W	0	**SQRXTUNE0**
23	R	0	**reserved**
22-20	R/W	0	**COMPDISTUNE0**
19	R/W	0	**BYPASSSEL0**
18	R/W	0	**BYPASSDMEN0**
17	R/W	0	**BYPASSDPEN0**
16	R/W	0	**BYPASSDMDATA0**
15	R/W	0	**BYPASSDPDATA0**
14	R/W	0	**TXBITSTUFFENH0**
13	R/W	0	**TXBITSTUFFEN0**
12	R/W	0	**SLEEPM0**
11	R/W	0	**VBUSVLDEXT0**
10	R/W	0	**VBUSVLDEXTSEL0**
9	R/W	0	**OTGDISABLE0**
8	R/W	0	**PORTRESET0**
7	R/W	0	**COMMONONN**
6-5	R/W	0	**REFCLKSEL**
4-2	R/W	0	**FSEL**
1-0	R/W	0	**VATESTENB**

7.1.34 ANA_REG2

Configure usb phy.

Offset_Address: 0x0000_0088



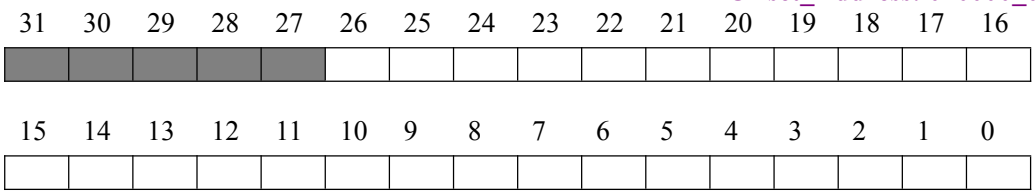
Bit	Type	Reset	Description
31-22	R	0	**reserved**
21	R/W	0	**LOOPBACKENB0**
20	R/W	0	**TESTBURNIN**
19	R	0	**reserved**

18	R/W	0	**DCDENB0**
17	R/W	0	**VDATSRCENB0**
16	R/W	0	**VDATDETENB0**
15	R/W	0	**CHRGSEL0**
14	R/W	0	**TXPREEMPPULSETUNE0**
13-12	R/W	0	**TXPREEMPAMPTUNE0**
11-10	R/W	0	**TXRESTUNE0**
9-8	R/W	0	**TXRISETUNE0**
7-4	R/W	0	**TXVREFTUNE0**
3-0	R/W	0	**TXFSLSTUNE0**

7.1.35 ANA_REG3

Configure usb phy.

Offset_Address: 0x0000_008C

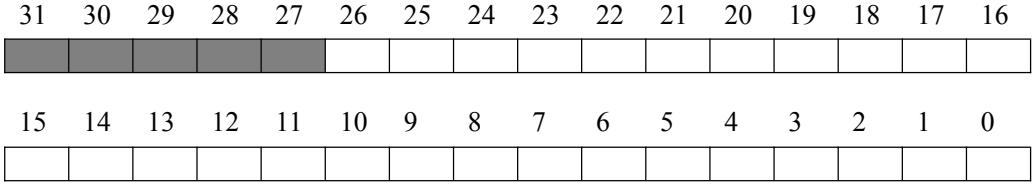


Bit	Type	Reset	Description
31-24	R/W	0	**USB_TESTDATAIN0**
23-22	R	0	**reserved**
21	R/W	0	**USB_TESTDATAOUTSEL0**
20	R/W	0	**USB_TESTCLK0**
19-16	R/W	0	**USB_TESTADDR0**
15-5	R	0	**reserved**
4	R	0	**USB_OTGSESSVLD0**
3-0	R	0	**USB_TESTDATAOUT0**

7.1.36 ANA_REG4

Configure usb phy.

Offset_Address: 0x0000_0090

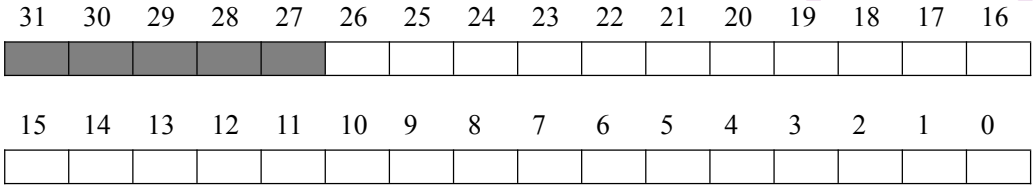


Bit	Type	Reset	Description
31-13	R	0	**reserved**
12	R/W	0	**SIDDQ**
11-10	R/W	0	**VDATREFTUNE0**
9	R/W	0	**HSXCVREXTCTL0**
8	R	0	**reserved**
7	R/W	0	**VREGBYPASS**
6	R/W	0	**PLLBTUNE**
5-2	R/W	0	**PLLPTUNE**
1-0	R/W	0	**PLLITUNE**

7.1.37 ANA_REG5

Configure lvds.

Offset_Address: 0x0000_0094



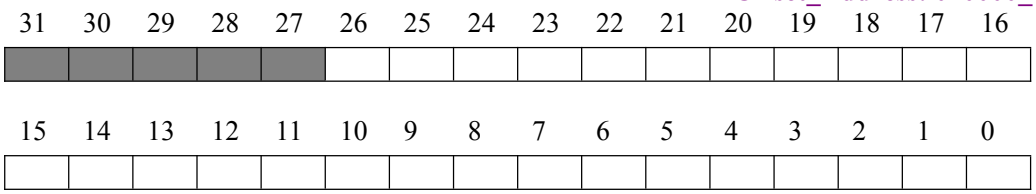
Bit	Type	Reset	Description
31	R/W	0	**TCLKIN_SEL** 1'b0: lcd_clk_pix; 1'b1: lcd_clk_pix div2;
30	R/W	0	**TXCLKSELB2**
29	R/W	0	**TXCLKSELB1**
28	R/W	0	**TXCLKSELB0**
27	R/W	0	**PWDN1**
26	R/W	0	**LVDS_GPIO_SEL_1**
25	R/W	0	**EN1**
24	R/W	0	**BITSEL1**
23	R	0	**reserved**
22	R/W	0	**TXCLKSELA2**
21	R/W	0	**TXCLKSELA1**
20	R/W	0	**TXCLKSELA0**
19	R/W	0	**PWDN0**

18	R/W	0	**LVDS_GPIO_SEL_0**
17	R/W	0	**EN0**
16	R/W	0	**BITSEL0**
15-7	R	0	**reserved**
6	R/W	0	**LVDS_ODD_INV**
5	R/W	0	**LVDS_HSYN_INV**
4	R/W	0	**LVDS_HSYN_SEL**
3	R/W	0	**LVDS_CH_SEL**
2	R/W	0	**LVDS_ODD_CH**
1	R/W	0	**LVDS_TI_SEL**
0	R/W	0	**LVDS_6BITS_SEL**

7.1.38 ANA_REG6

Configure lvds.

Offset_Address: 0x0000_0098



Bit	Type	Reset	Description
31-23	R	0	**reserved**
22	R/W	0	**WSEL**
21	R/W	0	**VREFEB**
20	R/W	0	**TX_LCK_EN**
19	R/W	0	**SRSA**
18	R/W	0	**SEL18_LVDS**
17	R/W	0	**S1_IREF**
16	R/W	0	**S0_IREF**
15	R/W	0	**PN_SWP**
14	R/W	0	**ICONT2**
13	R/W	0	**ICONT1**

12	R/W	0	**ICONT0**
11	R/W	0	**PLEB**
10	R/W	0	**CPEB**
9	R/W	0	**BW1**
8	R/W	0	**BW0**
7	R/W	0	**CLKEDGE**
6	R/W	0	**DRV_TRM_09**
5-3	R/W	0	**TRM_VCM**
2-0	R/W	0	**TRM_IREF**

7.1.39 ANA_REG8

Configure MIPI.

Offset_Address: 0x0000_00A0

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

Bit	Type	Reset	Description
31	R	0	**reserved**
12-10	R/W	0	**DSI_RREF_SEL**
9	R/W	0	**SEL18_DSI**
8	R/W	0	**DSI_MIPI_EN**
7-5	R/W	0	**reserved**
4-2	R/W	0	**CSI_RREF_SEL**
1	R/W	0	**SEL18_CSI**
0	R/W	0	**CSI_MIPI_EN**

7.1.40 ADC_CLK1_CFG

Configure ADC1 ADC2 clock.

Offset_Address: 0x0000_00A4

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

Bit	Type	Reset	Description
31	R	0	**reserved**
30-16	R/W	0	**adc2_clk_div** adc2_clk = osc_clk/(adc2_clk_div +1)*2
15	R/W	0	**reserved**
14-0	R/W	0	**adc1_clk_div** adc1_clk = osc_clk/(adc1_clk_div +1)*2

7.1.40 PADCTL0

PADCTL Register0

Offset_Address: 0x0000_00C0

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

Bit	Type	Reset	Description
31-30	R/W	0	2’b00: MCU_gpio15 2’b01: gpio15 2’b10: eth_rmii_mdo_o_e 2’b11: reserved
29-28	R/W	0	2’b00: MCU_gpio14 2’b01: gpio14 2’b10: eth_rmii_mdo 2’b11: reserved
27-26	R/W	0	2’b00: MCU_gpio13 2’b01: gpio13 2’b10: eth_rmii_mdc 2’b11: reserved
25-24	R/W	0	2’b00: MCU_gpio12 2’b01: gpio12 2’b10: pad_eth_rmii_clk 2’b11: reserved
23-22	R/W	0	2’b00: MCU_gpio11 2’b01: gpio11 2’b10: eth_phy_rxd3 2’b11: reserved
21-20	R/W	0	2’b00: MCU_gpio10 2’b01: gpio10 2’b10: eth_phy_rxd2 2’b11: reserved
19-18	R/W	0	2’b00: MCU_gpio9

			2’b01: gpio9 2’b10: eth_phy_rxd1 2’b11: reserved
17-16	R/W	0	2’b00: MCU_gpio8 2’b01: gpio8 2’b10: eth_phy_rxd0 2’b11: reserved
15-14	R/W	0	2’b00: MCU_gpio7 2’b01: gpio7 2’b10: eth_phy_rxdv 2’b11: reserved
13-12	R/W	0	2’b00: MCU_gpio6 2’b01: gpio6 2’b10: eth_phy_rx_clk 2’b11: reserved
11-10	R/W	0	2’b00: MCU_gpio5 2’b01: gpio5 2’b10: eth_phy_txd3 2’b11: reserved
9-8	R/W	0	2’b00: MCU_gpio4 2’b01: gpio4 2’b10: eth_phy_txd2 2’b11: reserved
7-6	R/W	0	2’b00: MCU_gpio3 2’b01: gpio3 2’b10: eth_phy_txd1 2’b11: reserved
5-4	R/W	0	2’b00: MCU_gpio2 2’b01: gpio2 2’b10: eth_phy_txd0 2’b11: reserved
3-2	R/W	0	2’b00: MCU_gpio1 2’b01: gpio1 2’b10: eth_phy_txen 2’b11: reserved
1-0	R/W	0	2’b00: MCU_gpio0 2’b01: gpio0 2’b10: pad_eth_tx_clk 2’b11: eth_tx_clk_to_pad

7.1.41 PADCTL1

PADCTL Register1

Offset_Address: 0x0000_00C4

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

Bit	Type	Reset	Description
31-30	R/W	0	2'b00: MCU_gpio31 2'b01: gpio31 2'b10: reserved 2'b11: reserved
29-28	R/W	0	2'b00: MCU_gpio30 2'b01: gpio30 2'b10: reserved 2'b11: reserved
27-26	R/W	0	2'b00: MCU_gpio29 2'b01: gpio29 2'b10: reserved 2'b11: reserved
25-24	R/W	0	2'b00: MCU_gpio28 2'b01: gpio28 2'b10: reserved 2'b11: reserved
23-22	R/W	0	2'b00: MCU_gpio27 2'b01: gpio27 2'b10: reserved 2'b11: reserved
21-20	R/W	0	2'b00: MCU_gpio26 2'b01: gpio26 2'b10: reserved 2'b11: reserved
19-18	R/W	0	2'b00: MCU_gpio25 2'b01: gpio25 2'b10: reserved 2'b11: reserved
17-16	R/W	0	2'b00: MCU_gpio24 2'b01: gpio24 2'b10: reserved 2'b11: reserved
15-14	R/W	0	2'b00: MCU_gpio23 2'b01: gpio23 2'b10: ext2_clk 2'b11: reserved
13-12	R/W	0	2'b00: MCU_gpio22 2'b01: gpio22 2'b10: ext2_clk 2'b11: reserved
11-10	R/W	0	2'b00: MCU_gpio21 2'b01: gpio21 2'b10: ext2_clk 2'b11: reserved
9-8	R/W	0	2'b00: MCU_gpio20 2'b01: gpio20

			2'b10: ext1_clk 2'b11: reserved
7-6	R/W	0	2'b00: MCU_gpio19 2'b01: gpio19 2'b10: ext1_clk 2'b11: reserved
5-4	R/W	0	2'b00: MCU_gpio18 2'b01: gpio18 2'b10: ext1_clk 2'b11: reserved
3-2	R/W	0	2'b00: MCU_gpio17 2'b01: gpio17 2'b10: ext1_clk 2'b11: reserved
1-0	R/W	0	2'b00: MCU_gpio16 2'b01: gpio16 2'b10: ext1_clk 2'b11: reserved

7.1.42 PADCTL2

PADCTL Register2

Offset_Address: 0x0000_00C8

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

Bit	Type	Reset	Description
31-30	R/W	0	2'b00: MCU_gpio65 2'b01: gpio65 2'b10: i2c1_scl 2'b11: reserved
29-28	R/W	0	2'b00: MCU_gpio64 2'b01: gpio64 2'b10: sdmmc1_d3 2'b11: reserved
27-26	R/W	0	2'b00: MCU_gpio63 2'b01: gpio63 2'b10: sdmmc1_d2 2'b11: reserved
25-24	R/W	0	2'b00: MCU_gpio62 2'b01: gpio62 2'b10: sdmmc1_d1 2'b11: reserved
23-22	R/W	0	2'b00: MCU_gpio61 2'b01: gpio61 2'b10: sdmmc1_d0

			2'b11: reserved
21-20	R/W	0	2'b00: MCU_gpio60 2'b01: gpio60 2'b10: sdmmc1_cmd 2'b11: reserved
19-18	R/W	0	2'b00: MCU_gpio59 2'b01: gpio59 2'b10: sdmmc1_clk_out 2'b11: reserved
17-16	R/W	0	2'b00: MCU_gpio58 2'b01: gpio58 2'b10: sdmmc1_cd_n 2'b11: reserved
15-14	R/W	0	2'b00: MCU_gpio39 2'b01: gpio39 2'b10: MCU_uart0_txd 2'b11: reserved
13-12	R/W	0	2'b00: MCU_gpio38 2'b01: gpio38 2'b10: MCU_uart0_rxd 2'b11: reserved
11-10	R/W	0	2'b00: MCU_gpio37 2'b01: gpio37 2'b10: reserved 2'b11: reserved
9-8	R/W	0	2'b00: MCU_gpio36 2'b01: gpio36 2'b10: reserved 2'b11: reserved
7-6	R/W	0	2'b00: MCU_gpio35 2'b01: gpio35 2'b10: reserved 2'b11: reserved
5-4	R/W	0	2'b00: MCU_gpio34 2'b01: gpio34 2'b10: reserved 2'b11: reserved
3-2	R/W	0	2'b00: MCU_gpio33 2'b01: gpio33 2'b10: reserved 2'b11: reserved
1-0	R/W	0	2'b00: MCU_gpio32 2'b01: gpio32 2'b10: reserved 2'b11: reserved

7.1.43 PADCTL3

PADCTL Register3

Offset_Address: 0x0000_00CC

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

Bit	Type	Reset	Description
31-30	R/W	0	2'b00: reserved 2'b01: gpio70 2'b10: uart2_rxd 2'b11: reserved
29-28	R/W	0	2'b00: reserved 2'b01: gpio69 2'b10: uart2_txd 2'b11: reserved
27-26	R/W	0	2'b00: reserved 2'b01: gpio68 2'b10: uart1_rxd 2'b11: reserved
25-24	R/W	0	2'b00: reserved 2'b01: gpio67 2'b10: uart1_txd 2'b11: reserved
23-22	R/W	0	2'b00: MCU_gpio77 2'b01: reserved 2'b10: uart0_rts 2'b11: reserved
21-20	R/W	0	2'b00: MCU_gpio76 2'b01: reserved 2'b10: uart0_cts 2'b11: reserved
19-18	R/W	0	2'b00: MCU_gpio75 2'b01: reserved 2'b10: uart0_rxd 2'b11: reserved
17-16	R/W	0	2'b00: MCU_gpio74 2'b01: reserved 2'b10: uart0_txd 2'b11: reserved
15-14	R/W	0	2'b00: MCU_gpio73 2'b01: reserved 2'b10: sdmmc_d3 2'b11: reserved
13-12	R/W	0	2'b00: MCU_gpio72 2'b01: reserved 2'b10: sdmmc_d2 2'b11: reserved

11-10	R/W	0	2'b00: MCU_gpio71 2'b01: reserved 2'b10: sdmmc_d1 2'b11: reserved
9-8	R/W	0	2'b00: MCU_gpio70 2'b01: reserved 2'b10: sdmmc_d0 2'b11: reserved
7-6	R/W	0	2'b00: MCU_gpio69 2'b01: reserved 2'b10: sdmmc_cmd 2'b11: reserved
5-4	R/W	0	2'b00: MCU_gpio68 2'b01: reserved 2'b10: sdmmc_clk 2'b11: reserved
3-2	R/W	0	2'b00: MCU_gpio67 2'b01: reserved 2'b10: sdmmc_cd_n 2'b11: reserved
1-0	R/W	0	2'b00: MCU_gpio66 2'b01: gpio66 2'b10: i2c1_sda 2'b11: reserved

7.1.44 PADCTL4

PADCTL Register4

Offset_Address: 0x0000_00D0

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

Bit	Type	Reset	Description
31-30	R/W	0	2'b00: MCU_gpio84 2'b01: reserved 2'b10: ssp_clk 2'b11: reserved
29-28	R/W	0	2'b00: MCU_gpio83 2'b01: gpio127 2'b10: ssp_csn 2'b11: reserved
27-26	R/W	0	2'b00: reserved 2'b01: gpio79 2'b10: i2s1_sdata 2'b11: reserved
25-24	R/W	0	2'b00: reserved

			2'b01: gpio78 2'b10: i2s1_sdata 2'b11: reserved
23-22	R/W	0	2'b00: reserved 2'b01: gpio77 2'b10: i2s1_bclk 2'b11: reserved
21-20	R/W	0	2'b00: reserved 2'b01: gpio76 2'b10: i2s1_mclk 2'b11: reserved
19-18	R/W	0	2'b00: reserved 2'b01: gpio75 2'b10: i2s1_sync 2'b11: reserved
17-16	R/W	0	2'b00: MCU_gpio82 2'b01: reserved 2'b10: i2s_sdata 2'b11: reserved
15-14	R/W	0	2'b00: MCU_gpio81 2'b01: reserved 2'b10: i2s_sdata 2'b11: reserved
13-12	R/W	0	2'b00: MCU_gpio80 2'b01: reserved 2'b10: i2s_bitclk 2'b11: reserved
11-10	R/W	0	2'b00: MCU_gpio79 2'b01: reserved 2'b10: i2s_mclk 2'b11: reserved
9-8	R/W	0	2'b00: MCU_gpio78 2'b01: reserved 2'b10: i2s_syncin 2'b11: reserved
7-6	R/W	0	2'b00: reserved 2'b01: gpio74 2'b10: uart3_rts 2'b11: reserved
5-4	R/W	0	2'b00: reserved 2'b01: gpio73 2'b10: uart3_cts 2'b11: reserved
3-2	R/W	0	2'b00: reserved 2'b01: gpio72 2'b10: uart3_rxd 2'b11: reserved
1-0	R/W	0	2'b00: reserved 2'b01: gpio71

			2'b10: uart3_txd 2'b11: reserved

7.1.45 PADCTL5

PADCTL Register5

Offset_Address: 0x0000_00D4

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

Bit	Type	Reset	Description
31-30	R/W	0	2'b00: MCU_gpio90 2'b01: reserved 2'b10: can0_tx 2'b11: reserved
29-28	R/W	0	2'b00: MCU_gpio89 2'b01: reserved 2'b10: can0_rx 2'b11: reserved
27-26	R/W	0	2'b00: reserved 2'b01: gpio89 2'b10: ssp1_rxd 2'b11: ssp1_s_rxd
25-24	R/W	0	2'b00: ssp1_s_rxd 2'b01: gpio88 2'b10: ssp1_txd 2'b11: ssp1_s_txd
23-22	R/W	0	2'b00: reserved 2'b01: gpio87 2'b10: ssp1_clkout 2'b11: ssp1_s_clkin
21-20	R/W	0	2'b00: MCU_gpio125 2'b01: gpio86 2'b10: ssp1_csn 2'b11: ssp1_s_cs
19-18	R/W	0	2'b00: reserved 2'b01: gpio85 2'b10: ssp2_d3 2'b11: reserved
17-16	R/W	0	2'b00: reserved 2'b01: gpio84 2'b10: ssp2_d2 2'b11: reserved
15-14	R/W	0	2'b00: reserved 2'b01: gpio83 2'b10: ssp2_d1

			2’b11: reserved
13-12	R/W	0	2’b00: reserved 2’b01: gpio82 2’b10: ssp2_d0 2’b11: reserved
11-10	R/W	0	2’b00: reserved 2’b01: gpio81 2’b10: ssp2_clk 2’b11: reserved
9-8	R/W	0	2’b00: MCU_gpio124 2’b01: gpio80 2’b10: ssp2_csn 2’b11: reserved
7-6	R/W	0	2’b00: MCU_gpio88 2’b01: reserved 2’b10: ssp_d3 2’b11: reserved
5-4	R/W	0	2’b00: MCU_gpio87 2’b01: reserved 2’b10: ssp_d2 2’b11: reserved
3-2	R/W	0	2’b00: MCU_gpio86 2’b01: reserved 2’b10: ssp_d1 2’b11: reserved
1-0	R/W	0	2’b00: MCU_gpio85 2’b01: reserved 2’b10: ssp_d0 2’b11: reserved

7.1.46 PADCTL6

PADCTL Register6

Offset_Address: 0x0000_00D8

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

Bit	Type	Reset	Description
31-30	R/W	0	2’b00: reserved 2’b01: gpio95 2’b10: pwm_in3 2’b11: reserved
29-28	R/W	0	2’b00: reserved 2’b01: gpio94 2’b10: pwm_in2 2’b11: reserved

27-26	R/W	0	2'b00: reserved 2'b01: gpio93 2'b10: pwm_in1 2'b11: reserved
25-24	R/W	0	2'b00: reserved 2'b01: gpio92 2'b10: pwm_in0 2'b11: reserved
23-22	R/W	0	2'b00: reserved 2'b01: gpio91 2'b10: pwm_out7 2'b11: uart3_re
21-20	R/W	0	2'b00: reserved 2'b01: gpio90 2'b10: pwm_out6 2'b11: uart3_de
19-18	R/W	0	2'b00: MCU_gpio100 2'b01: reserved 2'b10: pwm_out5 2'b11: uart3_re
17-16	R/W	0	2'b00: MCU_gpio99 2'b01: reserved 2'b10: pwm_out4 2'b11: uart3_de
15-14	R/W	0	2'b00: MCU_gpio98 2'b01: reserved 2'b10: pwm_out3 2'b11: reserved
13-12	R/W	0	2'b00: MCU_gpio97 2'b01: reserved 2'b10: pwm_out2 2'b11: reserved
11-10	R/W	0	2'b00: MCU_gpio96 2'b01: reserved 2'b10: pwm_out1 2'b11: reserved
9-8	R/W	0	2'b00: MCU_gpio95 2'b01: reserved 2'b10: pwm_out0 2'b11: reserved
7-6	R/W	0	2'b00: MCU_gpio94 2'b01: reserved 2'b10: can1_stby 2'b11: reserved
5-4	R/W	0	2'b00: MCU_gpio93 2'b01: reserved 2'b10: can1_tx 2'b11: reserved
3-2	R/W	0	2'b00: MCU_gpio92

			2’b01: reserved 2’b10: can1_rx 2’b11: reserved
1-0	R/W	0	2’b00: MCU_gpio91 2’b01: reserved 2’b10: can0_stby 2’b11: reserved

7.1.47 PADCTL7

PADCTL Register7

Offset_Address: 0x0000_00DC

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

Bit	Type	Reset	Description
31-30	R/W	0	2’b00: reserved 2’b01: gpio106 2’b10: itu_vin7 2’b11: reserved
29-28	R/W	0	2’b00: reserved 2’b01: gpio105 2’b10: itu_vin6 2’b11: ext2_clk
27-26	R/W	0	2’b00: reserved 2’b01: gpio104 2’b10: itu_vin5 2’b11: ext2_clk
25-24	R/W	0	2’b00: reserved 2’b01: gpio103 2’b10: itu_vin4 2’b11: ext2_clk
23-22	R/W	0	2’b00: reserved 2’b01: gpio102 2’b10: itu_vin3 2’b11: ext1_clk
21-20	R/W	0	2’b00: reserved 2’b01: gpio101 2’b10: itu_vin2 2’b11: ext1_clk
19-18	R/W	0	2’b00: reserved 2’b01: gpio100 2’b10: itu_vin1 2’b11: ext1_clk
17-16	R/W	0	2’b00: reserved 2’b01: gpio99

			2’b10: itu_vin0 2’b11: ext1_clk
15-14	R/W	0	2’b00: reserved 2’b01: gpio98 2’b10: i2c0_sda 2’b11: reserved
13-12	R/W	0	2’b00: reserved 2’b01: gpio97 2’b10: i2c0_scl 2’b11: reserved
11-10	R/W	0	2’b00: reserved 2’b01: gpio96 2’b10: rcrt_in 2’b11: reserved
9-8	R/W	0	2’b00: MCU_gpio123 2’b01: reserved 2’b10: TCK 2’b11: reserved
7-6	R/W	0	2’b00: MCU_gpio122 2’b01: reserved 2’b10: TDO 2’b11: uart2_rts
5-4	R/W	0	2’b00: MCU_gpio121 2’b01: reserved 2’b10: TDI 2’b11: uart2_cts
3-2	R/W	0	2’b00: MCU_gpio120 2’b01: reserved 2’b10: TMS 2’b11: uart1_rts
1-0	R/W	0	2’b00: MCU_gpio119 2’b01: reserved 2’b10: nTRST 2’b11: uart1_cts

7.1.48 PADCTL8

PADCTL Register8

Offset_Address: 0x0000_00E0

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

Bit	Type	Reset	Description
31-8	R	0	reserved
7-6	R/W	0	2’b00: reserved

			2'b01: gpio110 2'b10: ext2_clk 2'b11: reserved
5-4	R/W	0	2'b00: reserved 2'b01: gpio109 2'b10: itu_vsync 2'b11: reserved
3-2	R/W	0	2'b00: reserved 2'b01: gpio108 2'b10: itu_hsync 2'b11: reserved
1-0	R/W	0	2'b00: reserved 2'b01: gpio107 2'b10: itu_clk 2'b11: reserved

7.1.49 PADCTL9

PADCTL Register9

Offset_Address: 0x0000_00E4

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

Bit	Type	Reset	Description
31-30	R/W	0	2'b00: MCU_gpio53 2'b01: gpio53 2'b10: lcd_d15 2'b11: reserved
29-28	R/W	0	2'b00: MCU_gpio52 2'b01: gpio52 2'b10: lcd_d14 2'b11: reserved
27-26	R/W	0	2'b00: MCU_gpio51 2'b01: gpio51 2'b10: lcd_d13 2'b11: reserved
25-24	R/W	0	2'b00: MCU_gpio50 2'b01: gpio50 2'b10: lcd_d12 2'b11: reserved
23-22	R/W	0	2'b00: MCU_gpio49 2'b01: gpio49 2'b10: lcd_d11 2'b11: reserved
21-20	R/W	0	2'b00: MCU_gpio48 2'b01: gpio48

			2'b10: lcd_d10 2'b11: reserved
19-18	R/W	0	2'b00: MCU_gpio102 2'b01: reserved 2'b10: lcd_d9 2'b11: reserved
17-16	R/W	0	2'b00: MCU_gpio101 2'b01: reserved 2'b10: lcd_d8 2'b11: reserved
15-14	R/W	0	2'b00: MCU_gpio47 2'b01: gpio47 2'b10: lcd_d7 2'b11: reserved
13-12	R/W	0	2'b00: MCU_gpio46 2'b01: gpio46 2'b10: lcd_d6 2'b11: reserved
11-10	R/W	0	2'b00: MCU_gpio45 2'b01: gpio45 2'b10: lcd_d5 2'b11: reserved
9-8	R/W	0	2'b00: MCU_gpio44 2'b01: gpio44 2'b10: lcd_d4 2'b11: reserved
7-6	R/W	0	2'b00: MCU_gpio43 2'b01: gpio43 2'b10: lcd_d3 2'b11: reserved
5-4	R/W	0	2'b00: MCU_gpio42 2'b01: gpio42 2'b10: lcd_d2 2'b11: reserved
3-2	R/W	0	2'b00: MCU_gpio41 2'b01: gpio41 2'b10: lcd_d1 2'b11: reserved
1-0	R/W	0	2'b00: MCU_gpio40 2'b01: gpio40 2'b10: lcd_d0 2'b11: reserved

7.1.50 PADCTL10

PADCTL Register10

Offset_Address: 0x0000_00E8

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

Bit	Type	Reset	Description
31-24	R	0	reserved
23-22	R/W	0	2'b00: MCU_gpio57 2'b01: gpio57 2'b10: lcd_hsync 2'b11: reserved
21-20	R/W	0	2'b00: MCU_gpio56 2'b01: gpio56 2'b10: lcd_vsync 2'b11: reserved
19-18	R/W	0	2'b00: MCU_gpio55 2'b01: gpio55 2'b10: lcd_clk 2'b11: reserved
17-16	R/W	0	2'b00: MCU_gpio54 2'b01: gpio54 2'b10: lcd_de 2'b11: reserved
15-14	R/W	0	2'b00: MCU_gpio110 2'b01: reserved 2'b10: lcd_d23 2'b11: reserved
13-12	R/W	0	2'b00: MCU_gpio109 2'b01: reserved 2'b10: lcd_d22 2'b11: CPU_screen_sel
11-10	R/W	0	2'b00: MCU_gpio108 2'b01: reserved 2'b10: lcd_d21 2'b11: reserved
9-8	R/W	0	2'b00: MCU_gpio107 2'b01: reserved 2'b10: lcd_d20 2'b11: reserved
7-6	R/W	0	2'b00: MCU_gpio106 2'b01: reserved 2'b10: lcd_d19 2'b11: reserved
5-4	R/W	0	2'b00: MCU_gpio105 2'b01: reserved 2'b10: lcd_d18 2'b11: reserved
3-2	R/W	0	2'b00: MCU_gpio104 2'b01: reserved

			2’b10: lcd_d17 2’b11: reserved
1-0	R/W	0	2’b00: MCU_gpio103 2’b01: reserved 2’b10: lcd_d16 2’b11: reserved

7.1.51 PADCTL11

PADCTL Register11

Offset_Address: 0x0000_00EC

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

Bit	Type	Reset	Description
31-30	R/W	0	2’b00: resrved 2’b01: gpio126 2’b10: reserved 2’b11: reserved
29-28	R/W	0	2’b00: resrved 2’b01: gpio125 2’b10: reserved 2’b11: reserved
27-26	R/W	0	2’b00: resrved 2’b01: gpio124 2’b10: reserved 2’b11: reserved
25-24	R/W	0	2’b00: resrved 2’b01: gpio123 2’b10: reserved 2’b11: reserved
23-22	R/W	0	2’b00: resrved 2’b01: gpio122 2’b10: reserved 2’b11: reserved
21-20	R/W	0	2’b00: resrved 2’b01: gpio121 2’b10: reserved 2’b11: reserved
19-18	R/W	0	2’b00: resrved 2’b01: gpio120 2’b10: reserved 2’b11: reserved
17-16	R/W	0	2’b00: resrved 2’b01: gpio119

			2'b10: reserved 2'b11: reserved
15-14	R/W	0	2'b00: resrved 2'b01: gpio118 2'b10: reserved 2'b11: reserved
13-12	R/W	0	2'b00: resrved 2'b01: gpio117 2'b10: reserved 2'b11: reserved
11-10	R/W	0	2'b00: resrved 2'b01: gpio116 2'b10: reserved 2'b11: reserved
9-8	R/W	0	2'b00: resrved 2'b01: gpio115 2'b10: reserved 2'b11: reserved
7-6	R/W	0	2'b00: resrved 2'b01: gpio114 2'b10: reserved 2'b11: reserved
5-4	R/W	0	2'b00: resrved 2'b01: gpio113 2'b10: reserved 2'b11: reserved
3-2	R/W	0	2'b00: resrved 2'b01: gpio112 2'b10: reserved 2'b11: reserved
1-0	R/W	0	2'b00: resrved 2'b01: gpio111 2'b10: reserved 2'b11: reserved

7.1.52 PADCTL12

PADCTL Register12

Offset_Address: 0x0000_00F0

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

Bit	Type	Reset	Description
31-16	R	0	reserved
15-14	R/W	0	2'b00: MCU_gpio118 2'b01: reserved

			2'b10: reserved 2'b11: reserved
13-12	R/W	0	2'b00: MCU_gpio117 2'b01: reserved 2'b10: reserved 2'b11: reserved
11-10	R/W	0	2'b00: MCU_gpio116 2'b01: reserved 2'b10: reserved 2'b11: reserved
9-8	R/W	0	2'b00: MCU_gpio115 2'b01: reserved 2'b10: reserved 2'b11: reserved
7-6	R/W	0	2'b00: MCU_gpio114 2'b01: reserved 2'b10: reserved 2'b11: reserved
5-4	R/W	0	2'b00: MCU_gpio113 2'b01: reserved 2'b10: reserved 2'b11: reserved
3-2	R/W	0	2'b00: MCU_gpio112 2'b01: reserved 2'b10: reserved 2'b11: reserved
1-0	R/W	0	2'b00: MCU_gpio111 2'b01: reserved 2'b10: reserved 2'b11: reserved

7.1.53 PADCTL13

PADCTL Register13

Offset_Address: 0x0000_00F4

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

Bit	Type	Reset	Description
31-18	R	0	Reserved
17	R/W	0	SEL25 for PADCELL
16	R/W	0	SEL18 for PADCELL
15	R/W	0	can1_rx_fromRTC to can1_rx
14	R/W	0	can0_rx_fromRTC to can0_rx
13-8	R	0	reserved

7	R/W	0	i2s1_sdin_oe
6	R/W	0	i2s_sdin_oe
5	R/W	0	i2s1_sdout_oe
4	R/W	0	i2s_sdout_oe
3	R/W	0	ssp1_s_rxd sel
2	R/W	0	i2s1_sadata_in sel
1	R/W	0	i2s_sadata_in sel
0	R/W	0	1'b0: lcd_dout 888; 1'b1: lcd_dout 666;

7.1.54 PADCTL14

PADCTL Register14

Offset_Address: 0x0000_00F8

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

Bit	Type	Reset	Description
31-30	R/W	0	2'b00: MCU_gpio143 2'b01: gpio143 2'b10: reserved 2'b11: reserved
29-28	R/W	0	2'b00: MCU_gpio142 2'b01: gpio142 2'b10: reserved 2'b11: reserved
27-26	R/W	0	2'b00: MCU_gpio141 2'b01: gpio141 2'b10: reserved 2'b11: reserved
25-24	R/W	0	2'b00: MCU_gpio140 2'b01: gpio140 2'b10: reserved 2'b11: reserved
23-22	R/W	0	2'b00: MCU_gpio139 2'b01: gpio139 2'b10: reserved 2'b11: reserved
21-20	R/W	0	2'b00: MCU_gpio138 2'b01: gpio138 2'b10: reserved 2'b11: reserved
19-18	R/W	0	2'b00: MCU_gpio137 2'b01: gpio137 2'b10: reserved

			2’b11: reserved
17-16	R/W	0	2’b00: MCU_gpio136 2’b01: gpio136 2’b10: reserved 2’b11: reserved
15-14	R/W	0	2’b00: MCU_gpio135 2’b01: gpio135 2’b10: reserved 2’b11: reserved
13-12	R/W	0	2’b00: MCU_gpio134 2’b01: gpio134 2’b10: reserved 2’b11: reserved
11-10	R/W	0	2’b00: MCU_gpio133 2’b01: gpio133 2’b10: reserved 2’b11: reserved
9-8	R/W	0	2’b00: MCU_gpio132 2’b01: gpio132 2’b10: reserved 2’b11: reserved
7-6	R/W	0	2’b00: MCU_gpio131 2’b01: gpio131 2’b10: reserved 2’b11: reserved
5-4	R/W	0	2’b00: MCU_gpio130 2’b01: gpio130 2’b10: reserved 2’b11: reserved
3-2	R/W	0	2’b00: MCU_gpio129 2’b01: gpio129 2’b10: reserved 2’b11: reserved
1-0	R/W	0	2’b00: MCU_gpio128 2’b01: gpio128 2’b10: reserved 2’b11: reserved

7.1.55 PAD_DRV0

PAD Drive Register0

Offset_Address: 0x0000_0100

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

Bit	Type	Reset	Description
31-30	R/W	0	GPIO15_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
29-28	R/W	0	GPIO14_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
27-26	R/W	0	GPIO13_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
25-24	R/W	0	GPIO12_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
23-22	R/W	0	GPIO11_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	GPIO10_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	GPIO9_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	GPIO8_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	GPIO7_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	GPIO6_pad drive strength 2'b00: 2mA;

			2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	GPIO5_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	GPIO4_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	GPIO3_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	GPIO2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	GPIO1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	GPIO0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.56 PAD_DRV1

PAD Drive Register1

Offset_Address: 0x0000_0104

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

Bit	Type	Reset	Description
31-30	R/W	0	GPIO31_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA;

			2'b11: 12mA;
29-28	R/W	0	GPIO30_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
27-26	R/W	0	GPIO29_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
25-24	R/W	0	GPIO28_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
23-22	R/W	0	GPIO27_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	GPIO26_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	GPIO25_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	GPIO24_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	GPIO23_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	GPIO22_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	GPIO21_pad drive strength 2'b00: 2mA; 2'b01: 4mA;

			2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	GPIO20_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	GPIO19_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	GPIO18_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	GPIO17_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	GPIO16_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.57 PAD_DRV2

PAD Drive Register2

Offset_Address: 0x0000_0108

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

Bit	Type	Reset	Description
31-30	R/W	0	Scl1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
29-28	R/W	0	Sd1_d3_pad drive strength 2'b00: 2mA; 2'b01: 4mA;

			2'b10: 8mA; 2'b11: 12mA;
27-26	R/W	0	Sd1_d2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
25-24	R/W	0	Sd1_d1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
23-22	R/W	0	Sd1_d0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	Sd1_cmd_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	Sd1_clk_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	Sd1_cd_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	GPIO39_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	GPIO38_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	GPIO37_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	GPIO36_pad drive strength 2'b00: 2mA;

			2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	GPIO35_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	GPIO34_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	GPIO33_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	GPIO32_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.58 PAD_DRV3

PAD Drive Register3

Offset_Address: 0x0000_010C

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

Bit	Type	Reset	Description
31-30	R/W	0	Uart2_rxd_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
29-28	R/W	0	Uart2_txd_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
27-26	R/W	0	Uart1_rxd_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA;

			2'b11: 12mA;
25-24	R/W	0	Uart1_txd_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
23-22	R/W	0	Uart0_rts_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	Uart0_cts_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	Uart0_rxd_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	Uart0_txd_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	Emmc_d3_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	Emmc_d2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	Emmc_d1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	Emmc_d0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	Emmc_cmd_pad drive strength 2'b00: 2mA; 2'b01: 4mA;

			2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	Emmc_clk_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	Emmc_cd_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	Sda1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.59 PAD_DRV4

PAD Drive Register4

Offset_Address: 0x0000_0110

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

Bit	Type	Reset	Description
31-30	R/W	0	Ssp_clkout_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
29-28	R/W	0	Ssp_csn_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
27-26	R/W	0	I2s1_sdin_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
25-24	R/W	0	I2s1_sdout_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

23-22	R/W	0	I2s1_bclk_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	I2s1_mclk_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	I2s1_sync_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	I2s_sdin_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	I2s_sdout_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	I2s_bclk_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	I2s_mclk_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	I2s_sync_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	Uart3_rts_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	Uart3_cts_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA;

			2’b11: 12mA;
3-2	R/W	0	Uart3_rxd_pad drive strength 2’b00: 2mA; 2’b01: 4mA; 2’b10: 8mA; 2’b11: 12mA;
1-0	R/W	0	Uart3_txd_pad drive strength 2’b00: 2mA; 2’b01: 4mA; 2’b10: 8mA; 2’b11: 12mA;

7.1.60 PAD_DRV5

PAD Drive Register5

Offset_Address: 0x0000_0114

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

Bit	Type	Reset	Description
31-30	R/W	0	Can0_tx_pad drive strength 2’b00: 2mA; 2’b01: 4mA; 2’b10: 8mA; 2’b11: 12mA;
29-28	R/W	0	Can0_rx_pad drive strength 2’b00: 2mA; 2’b01: 4mA; 2’b10: 8mA; 2’b11: 12mA;
27-26	R/W	0	Ssp1_rxd_pad drive strength 2’b00: 2mA; 2’b01: 4mA; 2’b10: 8mA; 2’b11: 12mA;
25-24	R/W	0	Ssp1_txd_pad drive strength 2’b00: 2mA; 2’b01: 4mA; 2’b10: 8mA; 2’b11: 12mA;
23-22	R/W	0	Ssp1_clk_pad drive strength 2’b00: 2mA; 2’b01: 4mA; 2’b10: 8mA; 2’b11: 12mA;
21-20	R/W	0	Ssp1_csn_pad drive strength

			2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	Ssp2_d3_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	Ssp2_d2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	Ssp2_d1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	Ssp2_d0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	Ssp2_clkout_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	Ssp2_csn_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	Ssp_d3_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	Ssp_d2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	Ssp_d1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

1-0	R/W	0	Ssp_d0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.61 PAD_DRV6

PAD Drive Register6

Offset_Address: 0x0000_0118

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

Bit	Type	Reset	Description
31-30	R/W	0	Pwmin3_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
29-28	R/W	0	Pwmin2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
27-26	R/W	0	Pwmin1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
25-24	R/W	0	Pwmin0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
23-22	R/W	0	Pwmout7_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	pwmout6_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	Pwmout5_pad drive strength 2'b00: 2mA;

			2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	Pwmout4_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	Pwmout3_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	pwmout2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	Pwmout1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	Pwmout0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	Can1_stby_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	Can1_tx_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	Can1_rx_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	Can0_stby_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.62 PAD_DRV7

PAD Drive Register7

Offset_Address: 0x0000_011C

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

Bit	Type	Reset	Description
31-30	R/W	0	CSI_GPIO_DATAN2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
29-28	R/W	0	CSI_GPIO_DATAP2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
27-26	R/W	0	CSI_GPIO_DATAN1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
25-24	R/W	0	CSI_GPIO_DATAP1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
23-22	R/W	0	CSI_GPIO_DATAN0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	CSI_GPIO_DATAP0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	CSI_GPIO_CLKN_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	CSI_GPIO_CLKP_pad drive strength 2'b00: 2mA; 2'b01: 4mA;

			2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	Sda0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	Scl0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	Rcrt_in_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	tck_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	tdo_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	tdi_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	tms_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	ntrst_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.63 PAD_DRV8

PAD Drive Register8

Offset_Address: 0x0000_0120

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

Bit	Type	Reset	Description
31-8	R	0	reserved
7-6	R/W	0	mclk_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	Itu_vsync_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	CSI_GPIO_DATAN3_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	CSI_GPIO_DATAP3_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.64 PAD_DRV9

PAD Drive Register9

Offset_Address: 0x0000_0124

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

Bit	Type	Reset	Description
31-30	R/W	0	Lcd_d15_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
29-28	R/W	0	Lcd_d14_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

27-26	R/W	0	Lcd_d13_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
25-24	R/W	0	Lcd_d12_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
23-22	R/W	0	Lcd_d11_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	Lcd_d10_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	Lcd_d9_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	Lcd_d8_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	Lcd_d7_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	Lcd_d6_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	Lcd_d5_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	Lcd_d4_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA;

			2'b11: 12mA;
7-6	R/W	0	Lcd_d3_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	Lcd_d2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	Lcd_d1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	lcd_d0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.65 PAD_DRV10

PAD Drive Register10

Offset_Address: 0x0000_0128

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

Bit	Type	Reset	Description
31-24	R	0	reserved
23-22	R/W	0	Lcd_hsync_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	Lcd_vsync_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	Lcd_clk_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

17-16	R/W	0	Lcd_de_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	Lcd_d23_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	Lcd_d22_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	Lcd_d21_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	Lcd_d20_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	Lcd_d19_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	Lcd_d18_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	Lcd_d17_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	Lcd_d16_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.66 PAD_DRV11

PAD Drive Register11

Offset_Address: 0x0000_012C

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

Bit	Type	Reset	Description
31-30	R/W	0	GPIO143_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
29-28	R/W	0	GPIO142_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
27-26	R/W	0	GPIO141_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
25-24	R/W	0	GPIO140_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
23-22	R/W	0	GPIO139_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	GPIO138_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	GPIO137_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	GPIO136_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	GPIO135_pad drive strength 2'b00: 2mA;

			2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	GPIO134_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	GPIO133_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	GPIO132_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	GPIO131_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	GPIO130_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	GPIO129_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	GPIO128_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.67 PAD_DRV14

PAD Drive Register14

Offset_Address: 0x0000_0138

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

Bit	Type	Reset	Description
31-30	R/W	0	AUX7_1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
29-28	R/W	0	AUX6_1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
27-26	R/W	0	AUX5_1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
25-24	R/W	0	AUX4_1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
23-22	R/W	0	AUX3_1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
21-20	R/W	0	AUX2_1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
19-18	R/W	0	AUX1_1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
17-16	R/W	0	AUX0_1_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
15-14	R/W	0	AUX7_0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
13-12	R/W	0	AUX6_0_pad drive strength 2'b00: 2mA; 2'b01: 4mA;

			2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	AUX5_0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	AUX4_0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	AUX3_0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	AUX2_0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	AUX1_0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	AUX0_0_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.68 PAD_DRV15

PAD Drive Register15

Offset_Address: 0x0000_013C

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

Bit	Type	Reset	Description
31-16	R	0	reserved
15-14	R/W	0	AUX7_2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA;

			2'b11: 12mA;
13-12	R/W	0	AUX6_2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
11-10	R/W	0	AUX5_2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
9-8	R/W	0	AUX4_2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
7-6	R/W	0	AUX3_2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
5-4	R/W	0	AUX2_2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
3-2	R/W	0	AUX1_2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;
1-0	R/W	0	AUX0_2_pad drive strength 2'b00: 2mA; 2'b01: 4mA; 2'b10: 8mA; 2'b11: 12mA;

7.1.69 PAD_PUDCTL0

PAD Pull-up Pull-down Register0

Offset_Address: 0x0000_0140

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

Bit	Type	Reset	Description
31-30	R/W	0	GPIO15_pad

			2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
29-28	R/W	0	GPIO14_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
27-26	R/W	0	GPIO13_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
25-24	R/W	0	GPIO12_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
23-22	R/W	0	GPIO11_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
21-20	R/W	0	GPIO10_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
19-18	R/W	0	GPIO9_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
17-16	R/W	0	GPIO8_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
15-14	R/W	0	GPIO7_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
13-12	R/W	0	GPIO6_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;

11-10	R/W	0	GPIO5_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
9-8	R/W	0	GPIO4_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
7-6	R/W	0	GPIO3_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
5-4	R/W	0	GPIO2_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
3-2	R/W	0	GPIO1_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
1-0	R/W	0	GPIO0_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;

7.1.70 PAD_PUDCTL1

PAD Pull-up Pull-down Register1

Offset_Address: 0x0000_0144

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

Bit	Type	Reset	Description
31-30	R/W	0	GPIO31_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
29-28	R/W	0	GPIO30_pad 2'b00: normal;

			2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
27-26	R/W	0	GPIO29_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
25-24	R/W	0	GPIO28_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
23-22	R/W	0	GPIO27_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
21-20	R/W	0	GPIO26_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
19-18	R/W	0	GPIO25_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
17-16	R/W	0	GPIO24_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
15-14	R/W	0	GPIO23_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
13-12	R/W	0	GPIO22_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
11-10	R/W	0	GPIO21_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
9-8	R/W	0	GPIO20_pad

			2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
7-6	R/W	0	GPIO19_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
5-4	R/W	0	GPIO18_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
3-2	R/W	0	GPIO17_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
1-0	R/W	0	GPIO16_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;

7.1.71 PAD_PUDCTL2

PAD Pull-up Pull-down Register2

Offset_Address: 0x0000_0148

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

Bit	Type	Reset	Description
31-30	R/W	0	Scl1_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
29-28	R/W	0	Sd1_d3_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
27-26	R/W	0	Sd1_d2_pad 2’b00: normal;

			2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
25-24	R/W	0	Sd1_d1_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
23-22	R/W	0	Sd1_d0_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
21-20	R/W	0	Sd1_cmd_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
19-18	R/W	0	Sd1_clk_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
17-16	R/W	0	Sd1_cd_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
15-14	R/W	0	GPIO39_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
13-12	R/W	0	GPIO38_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
11-10	R/W	0	GPIO37_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
9-8	R/W	0	GPIO36_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
7-6	R/W	0	GPIO35_pad

			2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
5-4	R/W	0	GPIO34_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
3-2	R/W	0	GPIO33_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
1-0	R/W	0	GPIO32_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;

7.1.72 PAD_PUDCTL3

PAD Pull-up Pull-down Register3

Offset_Address: 0x0000_014C

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

Bit	Type	Reset	Description
31-30	R/W	0	Uart2_rxd_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
29-28	R/W	0	Uart2_txd_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
27-26	R/W	0	Uart1_rxd_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
25-24	R/W	0	Uart1_txd_pad 2'b00: normal; 2'b01: pull-up;

			2’b10: pull-down; 2’b11: repeater;
23-22	R/W	0	Uart0_rts_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
21-20	R/W	0	Uart0_cts_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
19-18	R/W	0	Uart0_rxd_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
17-16	R/W	0	Uart0_txd_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
15-14	R/W	0	Emmc_d3_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
13-12	R/W	0	Emmc_d2_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
11-10	R/W	0	Emmc_d1_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
9-8	R/W	0	Emmc_d0_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
7-6	R/W	0	Emmc_cmd_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
5-4	R/W	0	Emmc_clk_pad 2’b00: normal;

			2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
3-2	R/W	0	Emmc_cd_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
1-0	R/W	0	Sda1_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;

7.1.73 PAD_PUDCTL4

PAD Pull-up Pull-down Register4

Offset_Address: 0x0000_0150

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

Bit	Type	Reset	Description
31-30	R/W	0	Ssp_clkout_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
29-28	R/W	0	Ssp_csn_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
27-26	R/W	0	I2s1_sdin_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
25-24	R/W	0	I2s1_sdout_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
23-22	R/W	0	I2s1_bclk_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down;

			2’b11: repeater;
21-20	R/W	0	I2s1_mclk_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
19-18	R/W	0	I2s1_sync_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
17-16	R/W	0	I2s_sdin_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
15-14	R/W	0	I2s_sdout_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
13-12	R/W	0	I2s_bclk_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
11-10	R/W	0	I2s_mclk_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
9-8	R/W	0	I2s_sync_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
7-6	R/W	0	Uart3_rts_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
5-4	R/W	0	Uart3_cts_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
3-2	R/W	0	Uart3_rxd_pad 2’b00: normal; 2’b01: pull-up;

			2’b10: pull-down; 2’b11: repeater;
1-0	R/W	0	Uart3_txd_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;

7.1.74 PAD_PUDCTL5

PAD Pull-up Pull-down Register5

Offset_Address: 0x0000_0154

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

Bit	Type	Reset	Description
31-30	R/W	0	Can0_tx_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
29-28	R/W	0	Can0_rx_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
27-26	R/W	0	Ssp1_rxd_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
25-24	R/W	0	Ssp1_txd_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
23-22	R/W	0	Ssp1_clk_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
21-20	R/W	0	Ssp1_csn_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;

19-18	R/W	0	Ssp2_d3_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
17-16	R/W	0	Ssp2_d2_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
15-14	R/W	0	Ssp2_d1_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
13-12	R/W	0	Ssp2_d0_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
11-10	R/W	0	Ssp2_clkout_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
9-8	R/W	0	Ssp2_csn_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
7-6	R/W	0	Ssp_d3_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
5-4	R/W	0	Ssp_d2_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
3-2	R/W	0	Ssp_d1_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
1-0	R/W	0	Ssp_d0_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down;

			2’b11: repeater;

7.1.75 PAD_PUDCTL6

PAD Pull-up Pull-down Register6

Offset_Address: 0x0000_0158

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

Bit	Type	Reset	Description
31-30	R/W	0	Pwmin3_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
29-28	R/W	0	Pwmin2_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
27-26	R/W	0	Pwmin1_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
25-24	R/W	0	Pwmin0_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
23-22	R/W	0	Pwmout7_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
21-20	R/W	0	pwmout6_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
19-18	R/W	0	Pwmout5_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
17-16	R/W	0	Pwmout4_pad

			2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
15-14	R/W	0	Pwmout3_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
13-12	R/W	0	pwmout2_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
11-10	R/W	0	Pwmout1_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
9-8	R/W	0	Pwmout0_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
7-6	R/W	0	Can1_stby_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
5-4	R/W	0	Can1_tx_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
3-2	R/W	0	Can1_rx_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
1-0	R/W	0	Can0_stby_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;

7.1.76 PAD_PUDCTL7

PAD Pull-up Pull-down Register7

Offset_Address: 0x0000_015C

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

Bit	Type	Reset	Description
31-30	R/W	0	CSI_GPIO_DATAN2_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
29-28	R/W	0	CSI_GPIO_DATAP2_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
27-26	R/W	0	CSI_GPIO_DATAN1_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
25-24	R/W	0	CSI_GPIO_DATAP1_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
23-22	R/W	0	CSI_GPIO_DATAN0_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
21-20	R/W	0	CSI_GPIO_DATAP0_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
19-18	R/W	0	CSI_GPIO_CLKN_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
17-16	R/W	0	CSI_GPIO_CLKP_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
15-14	R/W	0	Sda0_pad 2'b00: normal;

			2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
13-12	R/W	0	Scl0_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
11-10	R/W	0	Rcrt_in_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
9-8	R/W	0	tck_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
7-6	R/W	0	tdo_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
5-4	R/W	0	tdi_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
3-2	R/W	0	tms_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
1-0	R/W	0	ntrst_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;

7.1.77 PAD_PUDCTL8

PAD Pull-up Pull-down Register8

Offset_Address: 0x0000_0160

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

Bit	Type	Reset	Description
31-8	R	0	reserved
7-6	R/W	0	mclk_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
5-4	R/W	0	Itu_vsync_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
3-2	R/W	0	CSI_GPIO_DATAN3_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
1-0	R/W	0	CSI_GPIO_DATAP3_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;

7.1.78 PAD_PUDCTL9

PAD Pull-up Pull-down Register9

Offset_Address: 0x0000_0154

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

Bit	Type	Reset	Description
31-30	R/W	0	Lcd_d15_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
29-28	R/W	0	Lcd_d14_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
27-26	R/W	0	Lcd_d13_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;

25-24	R/W	0	Lcd_d12_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
23-22	R/W	0	Lcd_d11_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
21-20	R/W	0	Lcd_d10_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
19-18	R/W	0	Lcd_d9_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
17-16	R/W	0	Lcd_d8_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
15-14	R/W	0	Lcd_d7_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
13-12	R/W	0	Lcd_d6_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
11-10	R/W	0	Lcd_d5_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
9-8	R/W	0	Lcd_d4_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
7-6	R/W	0	Lcd_d3_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down;

			2’b11: repeater;
5-4	R/W	0	Lcd_d2_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
3-2	R/W	0	Lcd_d1_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
1-0	R/W	0	lcd_d0_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;

7.1.79 PAD_PUDCTL10

PAD Pull-up Pull-down Register10

Offset_Address: 0x0000_0168

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

Bit	Type	Reset	Description
31-24	R	0	reserved
23-22	R/W	0	Lcd_hsync_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
21-20	R/W	0	Lcd_vsync_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
19-18	R/W	0	Lcd_clk_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;
17-16	R/W	0	Lcd_de_pad 2’b00: normal; 2’b01: pull-up; 2’b10: pull-down; 2’b11: repeater;

15-14	R/W	0	Lcd_d23_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
13-12	R/W	0	Lcd_d22_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
11-10	R/W	0	Lcd_d21_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
9-8	R/W	0	Lcd_d20_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
7-6	R/W	0	Lcd_d19_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
5-4	R/W	0	Lcd_d18_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
3-2	R/W	0	Lcd_d17_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
1-0	R/W	0	Lcd_d16_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;

7.1.80 PAD_PUDCTL11

PAD Pull-up Pull-down Register11

Offset_Address: 0x0000_016C

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

Bit	Type	Reset	Description
31-30	R/W	0	GPIO143_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
29-28	R/W	0	GPIO142_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
27-26	R/W	0	GPIO141_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
25-24	R/W	0	GPIO140_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
23-22	R/W	0	GPIO139_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
21-20	R/W	0	GPIO138_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
19-18	R/W	0	GPIO137_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
17-16	R/W	0	GPIO136_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
15-14	R/W	0	GPIO135_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;

13-12	R/W	0	GPIO134_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
11-10	R/W	0	GPIO133_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
9-8	R/W	0	GPIO132_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
7-6	R/W	0	GPIO131_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
5-4	R/W	0	GPIO130_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
3-2	R/W	0	GPIO129_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;
1-0	R/W	0	GPIO128_pad 2'b00: normal; 2'b01: pull-up; 2'b10: pull-down; 2'b11: repeater;

7.1.81 PAD_PUDCTL13

PAD Pull-up Pull-down Register13

Offset_Address: 0x0000_0174

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

Bit	Type	Reset	Description
31-20	R	0	reserved

19	R/W	0	Lcd_d23_pad REN
18	R/W	0	Lcd_d22_pad REN
17	R/W	0	Lcd_d21_pad REN
16	R/W	0	Lcd_d20_pad REN
15	R/W	0	Lcd_d19_pad REN
14	R/W	0	Lcd_d18_pad REN
13	R/W	0	Lcd_d17_pad REN
12	R/W	0	Lcd_d16_pad REN
11	R/W	0	Lcd_d15_pad REN
10	R/W	0	Lcd_d14_pad REN
9	R/W	0	Lcd_d13_pad REN
8	R/W	0	Lcd_d12_pad REN
7	R/W	0	Lcd_d11_pad REN
6	R/W	0	Lcd_d10_pad REN
5	R/W	0	Lcd_d9_pad REN
4	R/W	0	Lcd_d8_pad REN
3	R/W	0	Lcd_d3_pad REN
2	R/W	0	Lcd_d2_pad REN
1	R/W	0	Lcd_d1_pad REN
0	R/W	0	lcd_d0_pad REN

7.1.82 PAD_PUDCTL14

PAD Pull-up Pull-down Register14

Offset_Address: 0x0000_0178

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

Bit	Type	Reset	Description
31-24	R	0	reserved
23	R/W	0	AUX7_2_IE
22	R/W	0	AUX6_2_IE
21	R/W	0	AUX5_2_IE
20	R/W	0	AUX4_2_IE
19	R/W	0	AUX3_2_IE
18	R/W	0	AUX2_2_IE
17	R/W	0	AUX1_2_IE
16	R/W	0	AUX0_2_IE
15	R/W	0	AUX7_1_IE
14	R/W	0	AUX6_1_IE
13	R/W	0	AUX5_1_IE
12	R/W	0	AUX4_1_IE
11	R/W	0	AUX3_1_IE
10	R/W	0	AUX2_1_IE

9	R/W	0	AUX1_1_IE
8	R/W	0	AUX0_1_IE
7	R/W	0	AUX7_0_IE
6	R/W	0	AUX6_0_IE
5	R/W	0	AUX5_0_IE
4	R/W	0	AUX4_0_IE
3	R/W	0	AUX3_0_IE
2	R/W	0	AUX2_0_IE
1	R/W	0	AUX1_0_IE
0	R/W	0	AUX0_0_IE

7.1.83 SOFTRESET_CTL2

Offset_Address: 0x0000_0188

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

Bit	Type	Reset	Description
31-4	R	1	**reserved**
3	R/W	1	**softreset_wrap**
2	R/W	1	**softreset_adc2**
1	R/W	1	**softreset_csi**
0	R/W	1	**softreset_dsi**

7.1.84 I2S_NCO_CFG1

Offset_Address: 0x0000_018C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

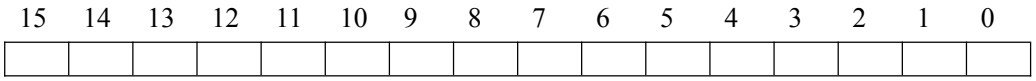
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-16	R/W	0	**step**
15-0	R/W	0	**mod**

7.1.85 MIPI_CTL

Offset_Address: 0x0000_0190

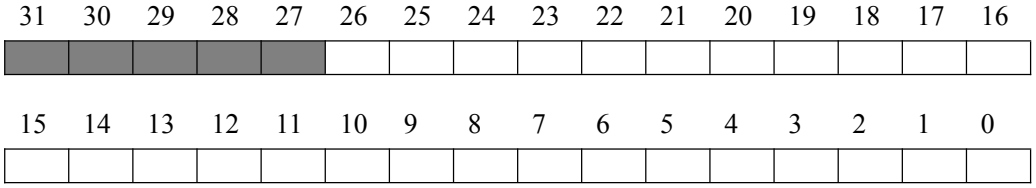
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



Bit	Type	Reset	Description
31-21	R	0	**reserved**
20	R/W	0	**dpiupdatecfg**
19	R/W	0	**dpicolorm**
18	R/W	0	**dpishutdn**
17	R/W	0	**dsi_bist_mode**
16	R/W	0	**mipidata_mode** 1'b0: rgb666; 1'b1: rgb888;
15-7	R	0	**reserved**
6	R/W	0	**csi_bist_mode**
5	R/W	0	**csi_hsync_mode** 1'b0: 根据前端 hsync 清 hcnt; 1'b1: 根据 total pix 清 hcnt;
4	R/W	0	**csi_de_inv**
3	R/W	0	**csi_hsync_inv**
2	R/W	0	**csi_vsync_inv**
1-0	R/W	0	**csi_mipi_data_type** 2'b00: yuv420; 2'b01: yuv420 legacy; 2'b10: yuv422; 2'b11: rgb888;

7.1.86 MIPI_BIST

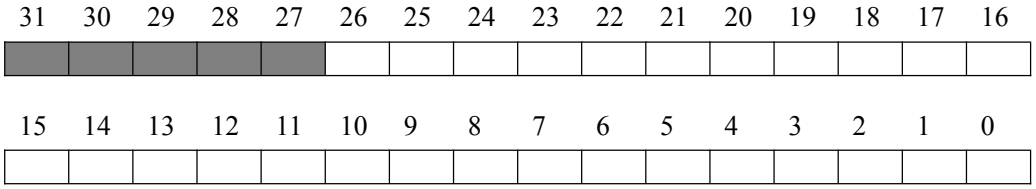
Offset_Address: 0x0000_0194



Bit	Type	Reset	Description
31-2	R	0	**reserved**
1	R	0	**csi_bist_result**
0	R	0	**dsi_bist_result**

7.1.87 MIPI_CLK_CFG

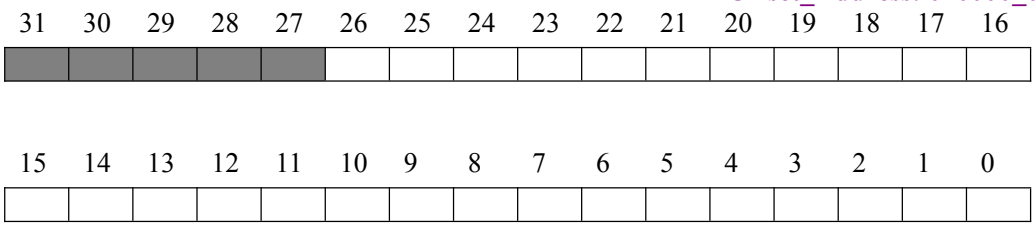
Offset_Address: 0x0000_0198



Bit	Type	Reset	Description
31-10	R	0	**reserved**
9	R/W	0	**csi_pixclk_sel** 0: csi_pix2xclk; 1: csi_pixclk_div2;
8	R/W	0	**csi_pix2xclk_sel** 0: gated_gpupll_clk; 1: gated_syspll_clk;
7-5	R	0	**reserved**
4-0	R/W	0	**csi_clk_div** csi_clk = int_csi_clk/(csi_clk_div+1)

7.1.88 MIPI_CSI_CFG1

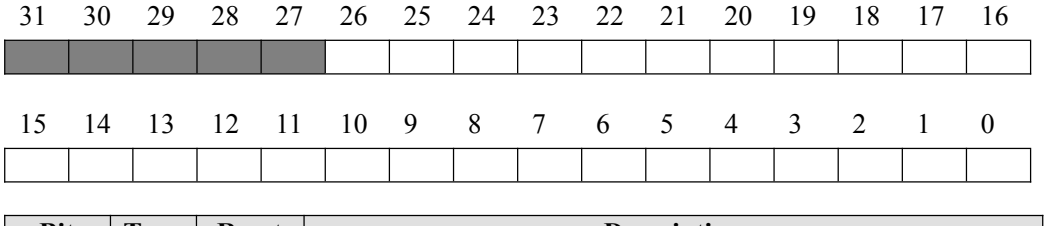
Offset_Address: 0x0000_019C



Bit	Type	Reset	Description
31-15	R	0	**reserved**
14-0	R/W	0	**csi_total_pix**

7.1.89 MIPI_CSI_CFG2

Offset_Address: 0x0000_01A0

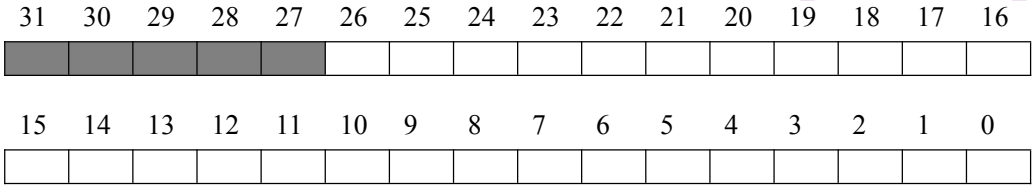


Bit	Type	Reset	Description
31	R	0	**reserved**
30-16	R/W	0	**csi_de_last_line**
15	R	0	**reserved**

14-0	R/W	0	**csi_act_line**

7.1.90 MIPI_CSI_CFG3

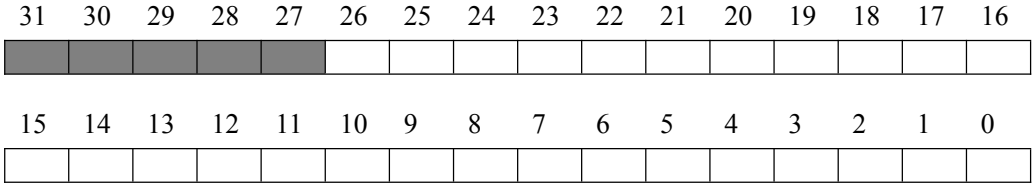
Offset_Address: 0x0000_01A4



Bit	Type	Reset	Description
31	R	0	**reserved**
30-16	R/W	0	**csi_de_h_fall**
15	R	0	**reserved**
14-0	R/W	0	**csi_de_h_rise**

7.1.91 MIPI_CSI_CFG4

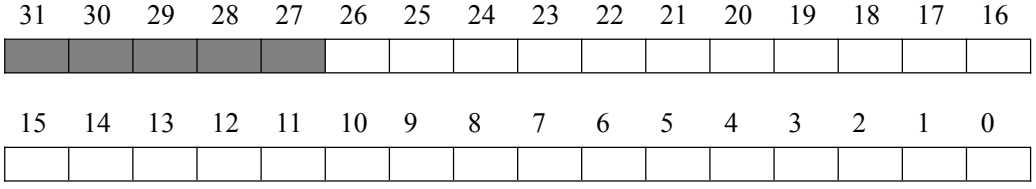
Offset_Address: 0x0000_01A8



Bit	Type	Reset	Description
31	R	0	**reserved**
30-16	R/W	0	**csi_hsync_h_fall**
15	R	0	**reserved**
14-0	R/W	0	**csi_hsync_h_rise**

7.1.92 GBUS_PRIORITY

Offset_Address: 0x0000_01AC

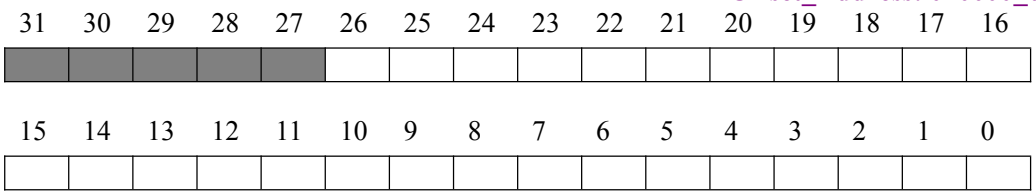


Bit	Type	Reset	Description
31-18	R	0	**reserved**
17-15	R/W	0	**mst_priority_m6**

14-12	R/W	0	**mst_priority_m5**
11-9	R/W	0	**mst_priority_m4**
8-6	R/W	0	**mst_priority_m3**
5-3	R/W	0	**mst_priority_m2**
2-0	R/W	0	**mst_priority_m1**

7.1.93 CHBUS_PRIORITY

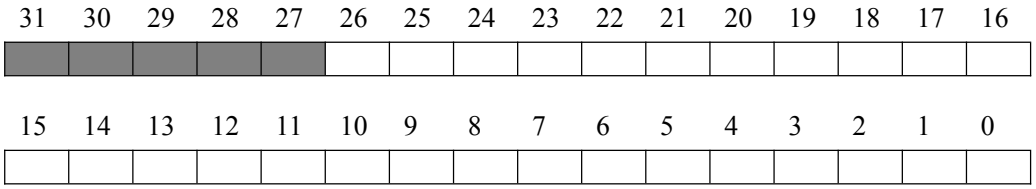
Offset_Address: 0x0000_01B0



Bit	Type	Reset	Description
31	R	0	**reserved**
30-28	R/W	0	**mst_priority_m5**
27-25	R/W	0	**mst_priority_m4**
24-22	R/W	0	**mst_priority_m3**
21-19	R/W	0	**mst_priority_m2**
18-16	R/W	0	**mst_priority_m1**
15-0	R	0	**reserved**

7.1.94 MIPI_TEST_CTL

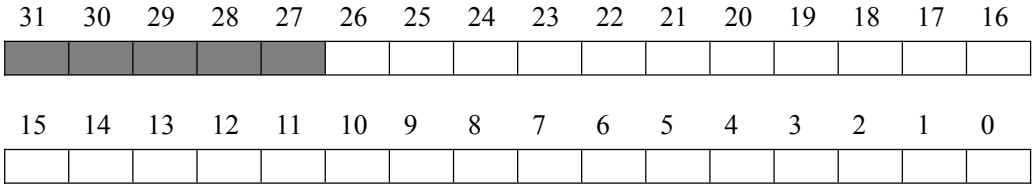
Offset_Address: 0x0000_01C0



Bit	Type	Reset	Description
31-11	R	0	**reserved**
10	R/W	0	**Mipi test en**
9	R/W	0	**Mipi_txrequestclkhs test**
8	R/W	0	**Mipi_txrequestdatahs test**
7-0	R/W	0	**Mipi txdatahs test**

7.1.95 MIPI_TEST_CTL1

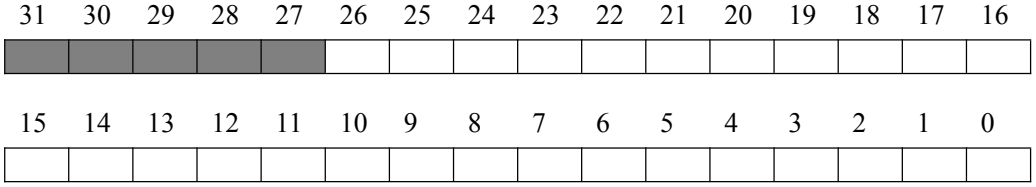
Offset_Address: 0x0000_01C4



Bit	Type	Reset	Description
31-0	R	0	**Mipi rxdatahs test**

7.1.96 WRAP_CFG

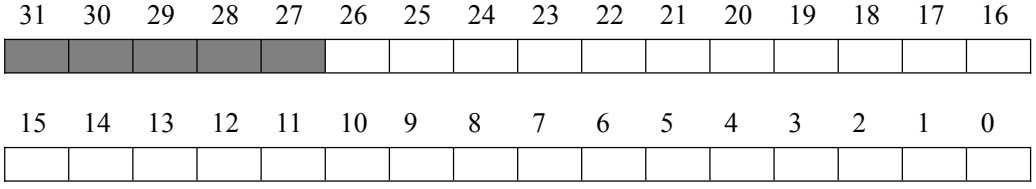
Offset_Address: 0x0000_01C8



Bit	Type	Reset	Description
31-1	R	0	**reserved**
0	R/W	0	**wrap_ram_sel** **0: CPU access;** **1: wrap access;**

7.1.96 RDC_CFG

Offset_Address: 0x0000_0200



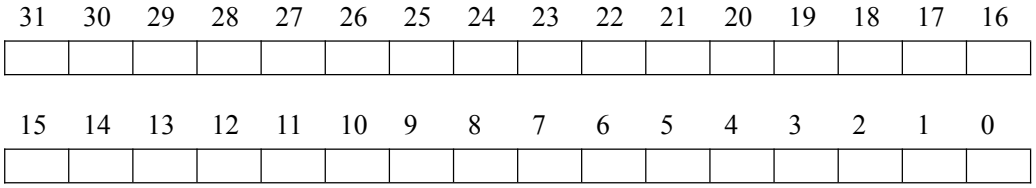
Bit	Type	Reset	Description
31-2	R	0	**reserved**
1	R/W	0	**sram_rdc_cfg_sec_resp**
0	R/W	0	**ddr_rdc_cfg_sec_resp**

7.2 MCU System Config Register Descriptions

MCU SYS BASE ADDR: 0x55000000

CPUrst_reg

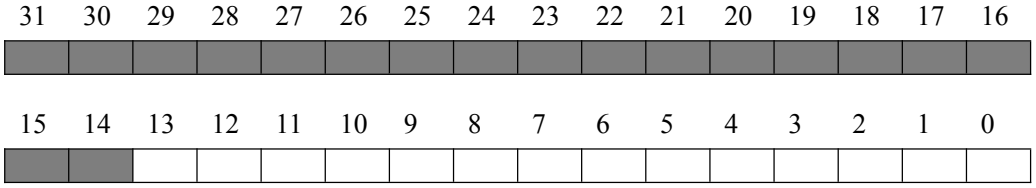
Offset_Address: 0x0000_0000



Bit	Type	Reset	Description
31-0	W	0	二次确认重置信号，初始值 32'h00000000 第一次输入 32'h12345678. 第二次输入 32'h55aabbcc 连续两次输入正确，softreset_CPU 信号拉低

CPU_ctl_cfg_reg

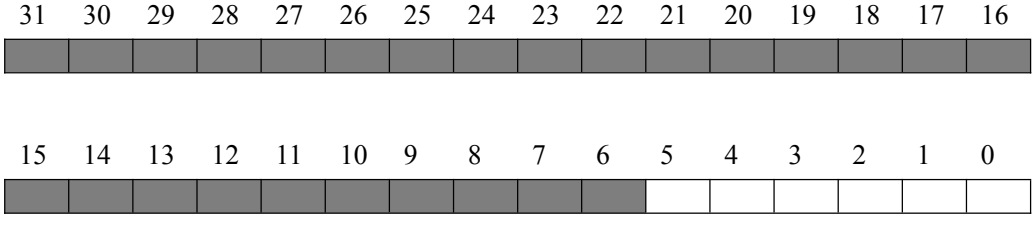
Offset_Address: 0x0000_0004



Bit	Type	Reset	Description
31-14	R/W	0	Reserved
13-8	R/W	0	CPU_ctrl
7-6	R/W	0	ss_sync_arl_me_bclk_div 输入 in 信号
5	R/W	0	重置信号 cbus_cresetn_sync(resetn)
4	R/W	0	重置信号 12resetn_sync(resetn)
3	R/W	1	在 csoc_dbgresetn[1]拉高的时候， CPU1_dresetn_sync(resetn)
2	R/W	1	在 csoc_dbgresetn[0]拉高的时候， CPU0_dresetn_sync(resetn)
1	R/W	0	重置信号 CPU1_cresetn_sync(resetn)
0	R/W	0	重置信号 CPU0_cresetn_sync(resetn)

CPUCPU_STAUS

Offset_Address: 0x0000_0008



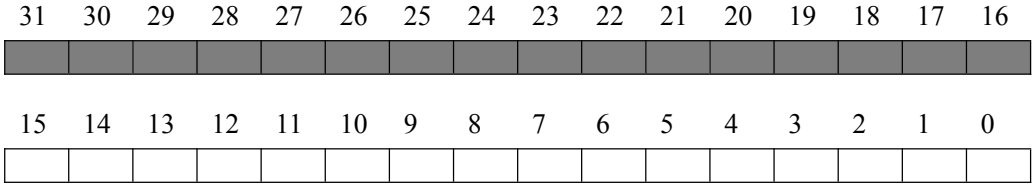
Bit	Type	Reset	Description

31-6	R	0	Reserved
5	R	0	wfc 模块 instr_type_wr_i 信号取值不等于 5'b01101 时拉高
4	R	0	wfc 模块 instr_type_wr_i 信号取值等于 5'b01101 时拉高
3	R	0	CPU_smp_en_1
2	R	0	1'b0
1	R	0	1'b0
0	R	0	1'b0

MCUbus_en_cfg_reg

初始值 32'hffff_ffff

Offset_Address: 0x0000_000c

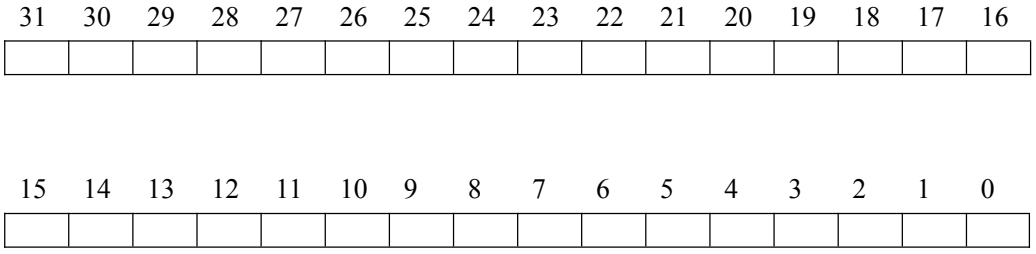


Bit	Type	Reset	Description
31-12	R/W	1	Reserved
11	R/W	1	MCU_pclk_group11_gate(clken)
10	R/W	1	MCU_pclk_group10_gate(clken)
9	R/W	1	MCU_pclk_group9_gate(clken)
8	R/W	1	MCU_pclk_group8_gate(clken)
7	R/W	1	MCU_pclk_group7_gate(clken)
6	R/W	1	MCU_pclk_group6_gate(clken)
5	R/W	1	MCU_pclk_group5_gate(clken)
4	R/W	1	MCU_pclk_group4_gate(clken)
3	R/W	1	MCU_pclk_group3_gate(clken)
2	R/W	1	MCU_pclk_group2_gate(clken)
1	R/W	1	MCU_pclk_group1_gate(clken)
0	R/W	1	MCU_pclk_group0_gate(clken)

MCUper_en_cfg_reg

初始值 32'hffff_ffff

Offset_Address: 0x0000_0010



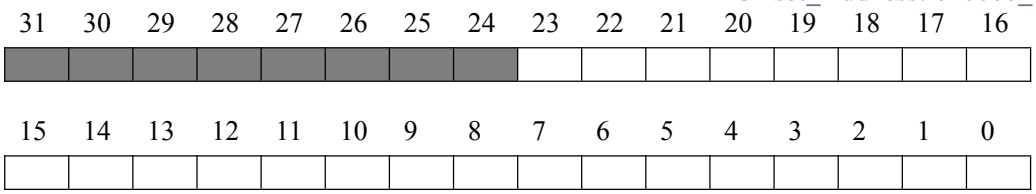
Bit	Type	Reset	Description
31-17	R/W	1	Reserved
16	R/W	1	MCU_pwm_clk_gate(clken)
15	R/W	1	MCU_uart0_clk_gate(clken)

14	R/W	1	MCU_dcic_osc_clk_gate(clken)
13	R/W	1	MCU_dcic_clk_gate(clken)
12	R/W	1	Inten_MCU_opfslv_clk
11	R/W	1	Inten_MCU_fuse_clk
10	R/W	1	Inten_MCU_spka_clk
9	R/W	1	ehse_clk_gate(clken)
8	R/W	1	Inten_MCU_trng_clk
7	R/W	1	Inten_MCU_wdt_clk
6	R/W	1	MCU_timer3_clk_gate(clken)
5	R/W	1	MCU_timer2_clk_gate(clken)
4	R/W	1	MCU_timer1_clk_gate(clken)
3	R/W	1	MCU_timer_clk_gate(clken)
2	R/W	1	Inten_MCU_gpio_clk
1	R/W	1	can1_clk_gate(clken)
0	R/W	1	can0_clk_gate(clken)

MCUrst_ctl_reg

初始值 32'hffff_ffff

Offset_Address: 0x0000_0014



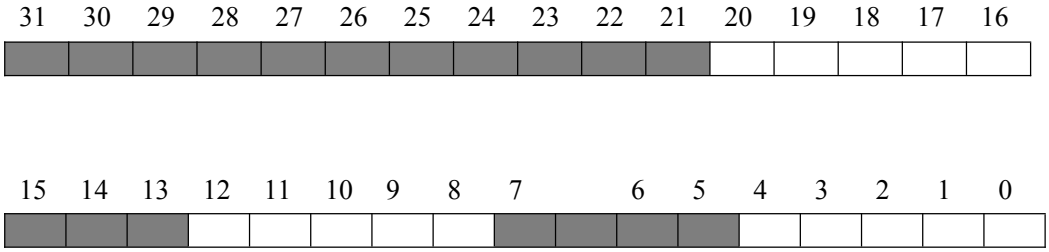
Bit	Type	Reset	Description
31	R/W	1	MCU_ecc_err_rst_en，信号iram_ecc_error取值判定条件
30-17	R/W	1	Reserved
16	R/W	1	控制信号MCU_softreset_pwm，控制 MCU_presetn_group[10] 0 信号取值1'b0 1 信号取值等同于hresetn_MCU
15	R/W	1	控制信号MCU_softreset_uart0，控制 MCU_presetn_group[9] 0 信号取值1'b0 1 信号取值等同于hresetn_MCU
14	R/W	1	MCU_softreset_dcic_sram
13	R/W	1	控制信号MCU_softreset_dcic，控制MCU_presetn_group[8] 0 信号取值1'b0 1 信号取值等同于hresetn_MCU
12	R/W	1	控制信号MCU_softreset_opfslv，控制 MCU_hresetn_group[4] 0 信号取值1'b0 1 信号取值等同于hresetn_MCU

11	R/W	1	控制信号MCU_softreset_fuse，控制MCU_presetn_group[7] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
10	R/W	1	控制信号MCU_softreset_spka，控制MCU_hresetn_group[3] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
9	R/W	1	控制信号MCU_softreset_ehse，控制MCU_hresetn_group[2] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
8	R/W	1	控制信号MCU_softreset_trng，控制MCU_presetn_group[6] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
7	R/W	1	控制信号MCU_softreset_wdt，控制MCU_presetn_group[0] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
6	R/W	1	控制信号MCU_softreset_timer3，控制 MCU_presetn_group[4] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
5	R/W	1	控制信号MCU_softreset_timer2，控制 MCU_presetn_group[3] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
4	R/W	1	控制信号MCU_softreset_timer1，控制 MCU_presetn_group[2] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
3	R/W	1	控制信号MCU_softreset_timer0，控制 MCU_presetn_group[1] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
2	R/W	1	控制信号MCU_softreset_gpio，控制MCU_presetn_group[5] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
1	R/W	1	控制信号MCU_softreset_can1，控制MCU_hresetn_group[1] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU
0	R/W	1	控制信号MCU_softreset_can0，控制MCU_hresetn_group[0] 0 信号取值1’b0 1 信号取值等同于hresetn_MCU

MCUtim_clk_cfg_reg

初始值 32’h0000_0000

Offset_Address: 0x0000_0018

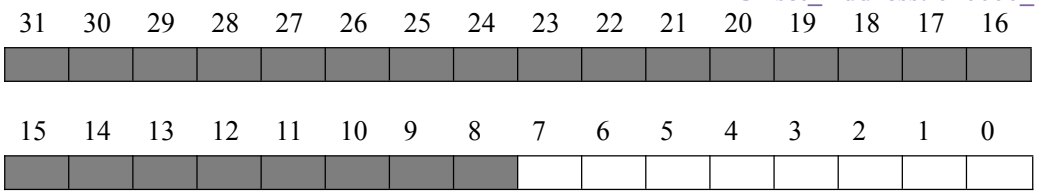


Bit	Type	Reset	Description
31-21	R/W	0	Reserved.
20	R/W	0	MCU_timer23src_sel_mux(sel)
20-16	R/W	0	MCU_timer23_clk_div(clk_div_couter)
15-13	R/W	0	Reserved.
12	R/W	0	MCU_timer1src_sel_mux(sel)
11-8	R/W	0	MCU_timer1_clk_div(clk_div_couter)
7-5	R/W	0	Reserved.
4	R/W	0	MCU_timersrc_sel_mux(sel)
3-0	R/W	0	MCU_timer_clk_div(clk_div_couter)

MCUehse_clk_cfg_reg

初始值 32'h0000_0000

Offset_Address: 0x0000_001c

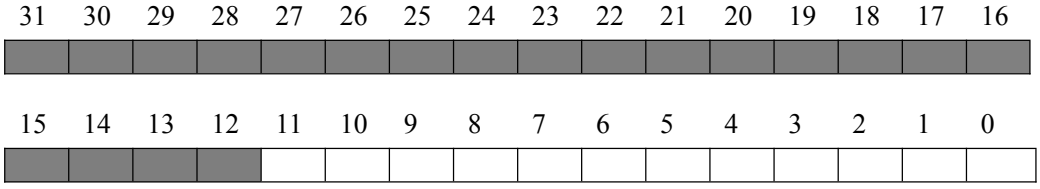


Bit	Type	Reset	Description
31-8	R/W	0	reserved
7-0	R/W	0	clk0_clk1_mux(sel)

MCUcan_clk_cfg_reg

初始值 32'h0000_0000

Offset_Address: 0x0000_0020

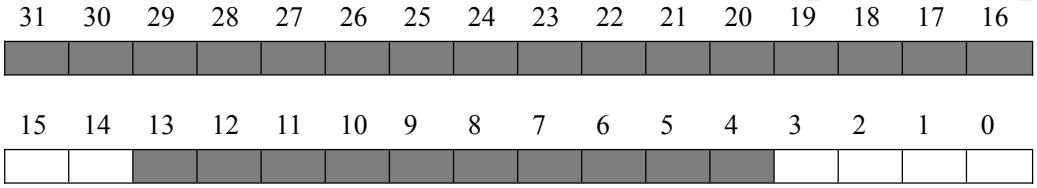


Bit	Type	Reset	Description
31-12	R/W	0	reserved
11	R/W	0	can1_clk_swich(sel)
10-6	R/W	0	can1_clk_div(clkout)
5	R/W	0	can0_clk_swich(sel)
4-0	R/W	0	can0_clk_div(clkout)

dbg_clt_cfg_reg

初始值 32'h0000_ffff

Offset_Address: 0x0000_0024

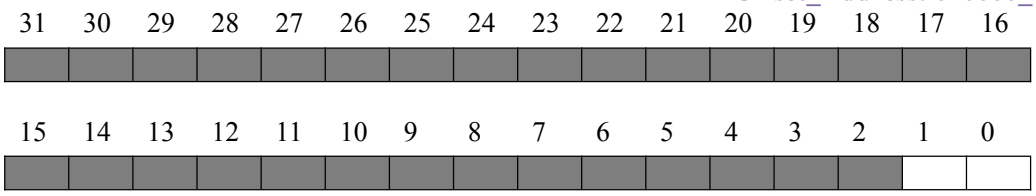


Bit	Type	Reset	Description
31-16	R/W	0	reserved
15	R/W	0	MCU_dapclken
14	R/W	0	dbg_deviceen
13-4	R/W	0	reserved
3	R/W	0	spniden_authen
2	R/W	0	spiden_authen
1	R/W	0	niden_authen
0	R/W	0	dbgen_authen

canfd_ctl_reg

初始值 32'h0000_0000

Offset_Address: 0x0000_0028

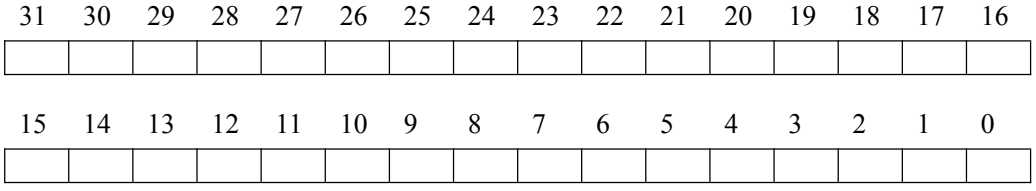


Bit	Type	Reset	Description
31-2	R/W	0	reserved
1	R/W	0	MCU_can1_fd_enable
0	R/W	0	MCU_can0_fd_enable

trng_statu

初始值 32'h0000_0000

Offset_Address: 0x0000_002c



Bit	Type	Reset	Description
31-8	R	0	trng_ctl_reg[32:8]
7	R	0	MCU_trng_secure
6	R	0	MCU_trng_reminder

5	R	0	MCU_trng_reseeding
4	R	0	MCU_trng_seeded
3	R	0	MCU_trng_rand_bit
2	R	0	MCU_trng_rand_vld
1-0	R	0	trng_ctl_reg[1:0]
31-17	W	0	reserved
16	W	0	MCU_spacc_endian
15-2	W	0	reserved
1	W	0	MCU_trng_reseed
0	W	0	MCU_trng_zeroize

SYS_PADCTL0

Offset_Address: 0x0000_00c0

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

Bit	Type	Reset	Description
31-30			决定模块选择输出信号 对 GPIO15 00 选择输出 MCU_gpio_dout[15] /输入 MCU_gpio_din[15] 01 选择输出 gpio_dout[15] /输入 gpio_din[15] 10 选择输出 ext1_clk 11 选择输出 1'b0
29-28			决定模块选择输出信号 对 GPIO14 00 选择输出 MCU_gpio_dout[14] /输入 MCU_gpio_din[14] 01 选择输出 gpio_dout[14] /输入 gpio_din[14] 10 选择输出 eth_rmii_mdo /输入 eth_rmii_mdi 11 选择输出 1'b0
27-26			决定模块选择输出信号 对 GPIO13 00 选择输出 MCU_gpio_dout[13] /输入 MCU_gpio_din[13] 01 选择输出 gpio_dout[13] /输入 gpio_din[13] 10 选择输出 eth_rmii_mdc 11 选择输出 1'b0
25-24			决定模块选择输出信号 对 GPIO12 00 选择输出 MCU_gpio_dout[12] /输入 MCU_gpio_din[12] 01 选择输出 gpio_dout[12] /输入 gpio_din[12]

			10 选择输入 pad_eth_rmii_clk 11 选择输出 1'b0
23-22			决定模块选择输出信号 对 GPIO11 00 选择输出 MCU_gpio_dout[11] /输入 MCU_gpio_din[11] 01 选择输出 gpio_dout[11] /输入 gpio_din[11] 10 选择输入 eth_phy_rxd[3] 11 选择输出 1'b0
21-20			决定模块选择输出信号 对 GPIO10 00 选择输出 MCU_gpio_dout[10] /输入 MCU_gpio_din[10] 01 选择输出 gpio_dout[10] /输入 gpio_din[10] 10 选择输入 eth_phy_rxd[2] 11 选择输出 1'b0
19-18			决定模块选择输出信号 对 GPIO9 00 选择输出 MCU_gpio_dout[9] /输入 MCU_gpio_din[9] 01 选择输出 gpio_dout[9] /输入 gpio_din[9] 10 选择输入 eth_phy_rxd[1] 11 选择输出 1'b0
17-16			决定模块选择输出信号 对 GPIO8 00 选择输出 MCU_gpio_dout[8] /输入 MCU_gpio_din[8] 01 选择输出 gpio_dout[8] /输入 gpio_din[8] 10 选择输入 eth_phy_rxd[0] 11 选择输入 MCU_fuse_data_in_cp
15-14			决定模块选择输出信号 对 GPIO7 00 选择输出 MCU_gpio_dout[7] /输入 MCU_gpio_din[7] 01 选择输出 gpio_dout[7] /输入 gpio_din[7] 10 选择输入 eth_phy_rxdv 11 选择输入 MCU_fuse_read_en_cp
13-12			决定模块选择输出信号 对 GPIO6 00 选择输出 MCU_gpio_dout[6] /输入 MCU_gpio_din[6] 01 选择输出 gpio_dout[6] /输入 gpio_din[6] 10 选择输入 pad_eth_rx_clk 11 选择输入 MCU_fuse_enable_cp
11-10			决定模块选择输出信号 对 GPIO5 00 选择输出 MCU_gpio_dout[5]/输入 MCU_gpio_din[5] 01 选择输出 gpio_dout[5] /输入 gpio_din[5] 10 选择输出 eth_phy_txd[3] 11 选择输出 MCU_fuse_no_access

9-8			决定模块选择输出信号 对 GPIO4 00 选择输出 MCU_gpio_dout[4] /输入 MCU_gpio_din[4] 01 选择输出 gpio_dout[4] /输入 gpio_din[4] 10 选择输出 eth_phy_txd[2] 11 选择输出 MCU_fuse_error
7-6			决定模块选择输出信号 对 GPIO3_a_mux 00 选择输出 MCU_gpio_dout[3] /输入 MCU_gpio_din[3] 01 选择输出 gpio_dout[3] /输入 gpio_din[3] 10 选择输出 eth_phy_txd[1] 11 选择输出 MCU_fuse_busy
5-4			决定模块选择输出信号 对 GPIO2 00 选择输出 MCU_gpio_dout[2] /输入 MCU_gpio_din[2] 01 选择输出 gpio_dout[2] /输入 gpio_din[2] 10 选择输出 eth_phy_txd[0] 11 选择输出 1’b0
3-2			决定模块选择输出信号 对 GPIO1 00 选择输出 MCU_gpio_dout[1] /输入 MCU_gpio_din[1] 01 选择输出 gpio_dout[1] /输入 gpio_din[1] 10 选择输出 eth_phy_txen 11 选择输出 1’b0
1-0			决定模块选择输出信号 对 GPIO0 00 选择输出 MCU_gpio_dout[0] /输入 MCU_gpio_din[0] 01 选择输出 gpio_dout[0]/输入 gpio_din[0] 10 选择输入 pad_eth_tx_clk 11 选择输出 eth_tx_clk_to_pad

SYS_PADCTL1

Offset_Address: 0x0000_00c4

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

Bit	Type	Reset	Description
31-30			决定模块选择输出信号 对 GPIO31 00 选择输出 MCU_gpio_dout[31] /输入 MCU_gpio_din[31] 01 选择输出 gpio_dout[31] /输入 gpio_din[31] 10 选择输出 1’b0

			11 选择输出 1’b0
29-28			决定模块选择输出信号 对 GPIO30 00 选择输出 MCU_gpio_dout[30] /输入 MCU_gpio_din[30] 01 选择输出 gpio_dout[30] /输入 gpio_din[30] 10 选择输出 1’b0 11 选择输出 1’b0
27-26			决定模块选择输出信号 对 GPIO29 00 选择输出 MCU_gpio_dout[29] /输入 MCU_gpio_din[29] 01 选择输出 gpio_dout[29] /输入 gpio_din[29] 10 选择输出 1’b0 11 选择输出 1’b0
25-24			决定模块选择输出信号 对 GPIO28 00 选择输出 MCU_gpio_dout[28] /输入 MCU_gpio_din[28] 01 选择输出 gpio_dout[28] /输入 gpio_din[28] 10 选择输出 1’b0 11 选择输出 1’b0
23-22			决定模块选择输出信号 对 GPIO27 00 选择输出 MCU_gpio_dout[27] /输入 MCU_gpio_din[27] 01 选择输出 gpio_dout[27] /输入 gpio_din[27] 10 选择输出 1’b0 11 选择输出 1’b0
21-20			决定模块选择输出信号 对 GPIO26 00 选择输出 MCU_gpio_dout[26] /输入 MCU_gpio_din[26] 01 选择输出 gpio_dout[26] /输入 gpio_din[26] 10 选择输出 1’b0 11 选择输出 1’b0
19-18			决定模块选择输出信号 对 GPIO25 00 选择输出 MCU_gpio_dout[25] /输入 MCU_gpio_din[25] 01 选择输出 gpio_dout[25] /输入 gpio_din[25] 10 选择输出 1’b0 11 选择输出 1’b0
17-16			决定模块选择输出信号 对 GPIO24 00 选择输出 MCU_gpio_dout[24] /输入 MCU_gpio_din[24] 01 选择输出 gpio_dout[24] /输入 gpio_din[24] 10 选择输出 1’b0 11 选择输出 1’b0
15-14			决定模块选择输出信号

			对 GPIO23 00 选择输出 MCU_gpio_dout[23] /输入 MCU_gpio_din[23] 01 选择输出 gpio_dout[23] /输入 gpio_din[23] 10 选择输出 ext2_clk 11 选择输出 1'b0
13-12			决定模块选择输出信号 对 GPIO22 00 选择输出 MCU_gpio_dout[22] /输入 MCU_gpio_din[22] 01 选择输出 gpio_dout[22] /输入 gpio_din[22] 10 选择输出 ext2_clk 11 选择输出 1'b0
11-10			决定模块选择输出信号 对 GPIO21 00 选择输出 MCU_gpio_dout[21] /输入 MCU_gpio_din[21] 01 选择输出 gpio_dout[21] /输入 gpio_din[21] 10 选择输出 ext2_clk 11 选择输出 1'b0
9-8			决定模块选择输出信号 对 GPIO20 00 选择输出 MCU_gpio_dout[20] /输入 MCU_gpio_din[20] 01 选择输出 gpio_dout[20] /输入 gpio_din[20] 10 选择输出 ext1_clk 11 选择输出 1'b0
7-6			决定模块选择输出信号 对 GPIO19 00 选择输出 MCU_gpio_dout[19] /输入 MCU_gpio_din[19] 01 选择输出 gpio_dout[19] /输入 gpio_din[19] 10 选择输出 ext1_clk 11 选择输出 1'b0
5-4			决定模块选择输出信号 对 GPIO18 00 选择输出 MCU_gpio_dout[18] /输入 MCU_gpio_din[18] 01 选择输出 gpio_dout[18] /输入 gpio_din[18] 10 选择输出 ext1_clk 11 选择输出 1'b0
3-2			决定模块选择输出信号 对 GPIO17 00 选择输出 MCU_gpio_dout[17] /输入 MCU_gpio_din[17] 01 选择输出 gpio_dout[17] /输入 gpio_din[17] 10 选择输出 ext1_clk 11 选择输出 1'b0
1-0			决定模块选择输出信号 对 GPIO16 00 选择输出 MCU_gpio_dout[16] /输入 MCU_gpio_din[16]

			01 选择输出 gpio_dout[16]/输入 gpio_din[16] 10 选择输出 ext1_clk 11 选择输出 1'b0

SYS_PADCTL2

Offset_Address: 0x0000_00c8

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

Bit	Type	Reset	Description
31-30			决定模块选择输出信号 对 scl1 00 选择输出 MCU_gpio_dout[65] /输入 MCU_gpio_din[65] 01 选择输出 gpio_dout[65] /输入 gpio_din[65] 10 选择输出 CPU_bist_fail /输入 i2c1_sck_in 11 选择输出 CPU_bist_fail
29-28			决定模块选择输出信号 对 sd1_d3 00 选择输出 MCU_gpio_dout[64] /输入 MCU_gpio_din[64] 01 选择输出 gpio_dout[64] /输入 gpio_din[64] 10 选择输出 sdmmc1_cdata_out[3] /输入 sdmmc1_cdata_in[13] 11 选择输出 sd1_d3_pad_a
27-26			决定模块选择输出信号 对 sd1_d2 00 选择输出 MCU_gpio_dout[63] /输入 MCU_gpio_din[63] 01 选择输出 gpio_dout[63] /输入 gpio_din[63] 10 选择输出 sdmmc1_cdata_out[2] /输入 sdmmc1_cdata_in[2] 11 选择输出 ble_bist_fail
25-24			决定模块选择输出信号 对 sd1_d1 00 选择输出 MCU_gpio_dout[62] /输入 MCU_gpio_din[62] 01 选择输出 gpio_dout[62] /输入 gpio_din[62] 10 选择输出 sdmmc1_cdata_out[1] /输入 sdmmc1_cdata_in[1] 11 选择输出 ble_bist_done
23-22			决定模块选择输出信号 对 sd1_d0 00 选择输出 MCU_gpio_dout[61] /输入 MCU_gpio_din[61] 01 选择输出 gpio_dout[61] /输入 gpio_din[61] 10 选择输出 sdmmc1_cdata_out[0] /输入

			sdmmc1_cdata_in[0] 11 选择输出 pxp_bist_fail
21-20			决定模块选择输出信号 对 sd1_cmd 00 选择输出 MCU_gpio_dout[60] /输入 MCU_gpio_din[60] 01 选择输出 gpio_dout[60] /输入 gpio_din[60] 10 选择输出 sdmmc1_ccmd_out /输入 sdmmc1_ccmd_in 11 选择输出 pxp_bist_done
19-18			决定模块选择输出信号 对 sd1_clk 00 选择输出 MCU_gpio_dout[59] /输入 MCU_gpio_din[59] 01 选择输出 gpio_dout[59] /输入 gpio_din[59] 10 选择输出 sdmmc1_cclk_out 11 选择输出 png_bist_fail
17-16			决定模块选择输出信号 对 sd1_cd 00 选择输出 MCU_gpio_dout[58] /输入 MCU_gpio_din[58] 01 选择输出 gpio_dout[58] /输入 gpio_din[58] 10 选择输入 sdmmc1_card_detect_n 11 选择输出 png_bist_done
15-14			决定模块选择输出信号 对 GPIO39 00 选择输出 MCU_gpio_dout[39] /输入 MCU_gpio_din[39] 01 选择输出 gpio_dout[39] /输入 gpio_din[39] 10 选择输出 MCU_uart0_txd 11 选择输出 mfc_bist_fail
13-12			决定模块选择输出信号 对 GPIO38 00 选择输出 MCU_gpio_dout[38] /输入 MCU_gpio_din[38] 01 选择输出 gpio_dout[38] /输入 gpio_din[38] 10 选择输入 MCU_uart0_rxd 11 选择输出 mfc_bist_done
11-10			决定模块选择输出信号 对 GPIO37 00 选择输出 MCU_gpio_dout[37] /输入 MCU_gpio_din[37] 01 选择输出 gpio_dout[37] /输入 gpio_din[37] 10 选择输出 1’b0 11 选择输出 gpu_bist_fail
9-8			决定模块选择输出信号 对 GPIO36 00 选择输出 MCU_gpio_dout[36] /输入 MCU_gpio_din[36] 01 选择输出 gpio_dout[36] /输入 gpio_din[36] 10 选择输出 1’b0 11 选择输出 gpu_bist_done

7-6			决定模块选择输出信号 对 GPIO35 00 选择输出 MCU_gpio_dout[35] /输入 MCU_gpio_din[35] 01 选择输出 gpio_dout[35] /输入 gpio_din[35] 10 选择输出 1'b0 11 选择输出 lcd_bist_fail
5-4			决定模块选择输出信号 对 GPIO34 00 选择输出 MCU_gpio_dout[34] /输入 MCU_gpio_din[34] 01 选择输出 gpio_dout[34] /输入 gpio_din[34] 10 选择输出 1'b0 11 选择输出 lcd_bist_done
3-2			决定模块选择输出信号 对 GPIO33 00 选择输出 MCU_gpio_dout[33] /输入 MCU_gpio_din[33] 01 选择输出 gpio_dout[33] /输入 gpio_din[33] 10 选择输出 1'b0 11 选择输出 ram_bist_fail
1-0			决定模块选择输出信号 对 GPIO32 00 选择输出 MCU_gpio_dout[32] /输入 MCU_gpio_din[32] 01 选择输出 gpio_dout[32]/输入 gpio_din[32] 10 选择输出 1'b0 11 选择输出 ram_bist_done

SYS_PADCTL3

Offset_Address: 0x0000_00cc

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

Bit	Type	Reset	Description
31-30			决定模块选择输出信号 对 GPIO31 00 悬空 01 选择输出 gpio_dout[70] /输入 gpio_din[70] 10 选择输入 uart2_rxd 11 悬空
29-28			决定模块选择输出信号 对 uart2_txd 00 悬空 01 选择输出 gpio_dout[69] /输入 gpio_din[69] 10 选择输出 uart2_txd

			11 悬空
27-26			决定模块选择输出信号 对 uart1_rxd 00 悬空 01 选择输出 gpio_dout[68] /输入 gpio_din[68] 10 选择输入 uart1_rxd 11 选择输出 MCU_iram_bist_fail
25-24			决定模块选择输出信号 对 uart1_txd 00 悬空 01 选择输出 gpio_dout[67] /输入 gpio_din[67] 10 选择输出 uart1_txd 11 选择输出 MCU_iram_bist_done
23-22			决定模块选择输出信号 对 uart0_rts 00 选择输出 MCU_gpio_dout[77] /输入 MCU_gpio_din[77] 01 悬空 10 选择输出 uart0_rts 11 选择输出 MCU_irom_bist_fail
21-20			决定模块选择输出信号 对 uart0_cts 00 选择输出 MCU_gpio_dout[76] /输入 MCU_gpio_din[76] 01 悬空 10 选择输入 uart0_cts 11 选择输出 MCU_irom_bist_done
19-18			决定模块选择输出信号 对 uart0_rxd 00 选择输出 MCU_gpio_dout[75] /输入 MCU_gpio_din[75] 01 悬空 10 选择输入 uart0_rxd 11 选择输出 can_bist_done
17-16			决定模块选择输出信号 对 uart0_txd 00 选择输出 MCU_gpio_dout[74] /输入 MCU_gpio_din[74] 01 悬空 10 选择输出 uart0_txd 11 选择输出 can_bist_fail
15-14			决定模块选择输出信号 对 emmc_d3 00 选择输出 MCU_gpio_dout[73] /输入 MCU_gpio_din[73] 01 悬空 10 选择输出 sdmmc_cdata_out[3] /输入 sdmmc_cdata_in[3] 11 选择输出 dsi_bist_fail
13-12			决定模块选择输出信号

			对 emmc_d2 00 选择输出 MCU_gpio_dout[72] /输入 MCU_gpio_din[72] 01 悬空 10 选择输出 sdmmc_cdata_out[2] /输入 sdmmc_cdata_in[2] 11 选择输出 dsi_bist_done
11-10			决定模块选择输出信号 对 emmc_d1 00 选择输出 MCU_gpio_dout[71] /输入 MCU_gpio_din[71] 01 悬空 10 选择输出 sdmmc_cdata_out[1] /输入 sdmmc_cdata_in[1] 11 选择输出 csi_bist_fail
9-8			决定模块选择输出信号 对 emmc_d0 00 选择输出 MCU_gpio_dout[70] /输入 MCU_gpio_din[70] 01 悬空 10 选择输出 sdmmc_cdata_out[0] /输入 sdmmc_cdata_in[0] 11 选择输出 csi_bist_done
7-6			决定模块选择输出信号 对 emmc_cmd 00 选择输出 MCU_gpio_dout[69] /输入 MCU_gpio_din[69] 01 悬空 10 选择输出 sdmmc_ccmd_out /输入 sdmmc_ccmd_in 11 选择输出 usb_bist_fail
5-4			决定模块选择输出信号 对 emmc_clk 00 选择输出 MCU_gpio_dout[68] /输入 MCU_gpio_din[68] 01 悬空 10 选择输出 sdmmc_cclk_out 11 选择输出 usb_bist_done
3-2			决定模块选择输出信号 对 emmc_cd 00 选择输出 MCU_gpio_dout[67] /输入 MCU_gpio_din[67] 01 悬空 10 选择输入 sdmmc_card_detect_n 11 选择输出 itu_bist_fail
1-0			决定模块选择输出信号 对 sda1 00 选择输出 MCU_gpio_dout[66] /输入 MCU_gpio_din[66] 01 选择输出 gpio_dout[66] /输入 gpio_din[66] 10 选择输出 ext1_clk /输入 i2c1_sda_in 11 选择输出 itu_bist_done

SYS_PADCTL4

Offset_Address: 0x0000_00d0

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

Bit	Type	Reset	Description
31-30			决定模块选择输出信号 对 ssp_clkout 00 选择输出 MCU_gpio_dout[84] /输入 MCU_gpio_din[84] 01 悬空 10 选择输出 ssp_clkout 11 悬空
29-28			决定模块选择输出信号 对 ssp_csn0 00 选择输出 MCU_gpio_dout[83] /输入 MCU_gpio_din[83] 01 选择输出 gpio_dout[127] /输入 gpio_din[127] 10 选择输出 ssp_csn0_out 11 悬空
27-26			决定模块选择输出信号 对 i2s1_sdin 00 悬空 01 选择输出 gpio_dout[79] /输入 gpio_din[79] 10 选择输出 i2s1_sadata_out /输入 i2s1_sadata_in_1 11 悬空
25-24			决定模块选择输出信号 对 i2s1_sdout 00 悬空 01 选择输出 gpio_dout[78] /输入 gpio_din[78] 10 选择输出 i2s1_sadata_out /输入 i2s1_sadata_in_0 11 悬空
23-22			决定模块选择输出信号 对 i2s1_bclk 00 悬空 01 选择输出 gpio_dout[77] /输入 gpio_din[77] 10 选择输出 i2s1_bitclk /输入 i2s1_codec_bclkin 11 悬空
21-20			决定模块选择输出信号 对 i2s1_mclk 00 悬空 01 选择输出 gpio_dout[76] /输入 gpio_din[76] 10 选择输出 i2s1_mclk 11 悬空
19-18			决定模块选择输出信号

			对 i2s1_sync 00 悬空 01 选择输出 gpio_dout[75] /输入 gpio_din[75] 10 选择输出 i2s1_syncout /输入 i2s1_syncin 11 悬空
17-16			决定模块选择输出信号 对 i2s_sdin 00 选择输出 MCU_gpio_dout[82] /输入 MCU_gpio_din[82] 01 悬空 10 选择输出 i2s_sadata_out /输入 i2s_sadata_in_1 11 悬空
15-14			决定模块选择输出信号 对 i2s_sdout 00 选择输出 MCU_gpio_dout[81] /输入 MCU_gpio_din[81] 01 悬空 10 选择输出 i2s_sadata_out /输入 i2s_sadata_in_0 11 悬空
13-12			决定模块选择输出信号 对 i2s_bclk 00 选择输出 MCU_gpio_dout[80] /输入 MCU_gpio_din[80] 01 悬空 10 选择输出 i2s_bitclk 11 悬空
11-10			决定模块选择输出信号 对 i2s_mclk 00 选择输出 MCU_gpio_dout[79] /输入 MCU_gpio_din[79] 01 悬空 10 选择输出 i2s_mclk 11 悬空
9-8			决定模块选择输出信号 对 i2s_sync 00 选择输出 MCU_gpio_dout[78] /输入 MCU_gpio_din[78] 01 悬空 10 选择输出 i2s_syncout /输入 i2s_syncin 11 选择输出 1’b0
7-6			决定模块选择输出信号 对 uart3_rts 00 悬空 01 选择输出 gpio_dout[74] /输入 gpio_din[74] 10 选择输出 uart3_rts 11 悬空
5-4			决定模块选择输出信号 对 uart3_cts 00 悬空

			01 选择输出 gpio_dout[73] /输入 gpio_din[73] 10 选择输出 uart3_cts 11 悬空
3-2			决定模块选择输出信号 对 uart3_rxd 00 悬空 01 选择输出 gpio_dout[72] /输入 gpio_din[72] 10 选择输入 uart3_rxd 11 悬空
1-0			决定模块选择输出信号 对 uart3_txd 00 悬空 01 选择输出 gpio_dout[71]/输入 gpio_din[71] 10 选择输出 uart3_txd 11 悬空

SYS_PADCTL5

Offset_Address: 0x0000_00d4

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

Bit	Type	Reset	Description
31-30			can0_tx 00 MCU_gpio_dout[90] / MCU_gpio_din[90] 01 10 mcan0_tx 11
29-28			can0_rx_pad 00 MCU_gpio_dout[89] / MCU_gpio_din[89] 01 10 mcan0_rx 11
27-26			ssp1_rxd_pad 00 01 gpio_dout[89] / gpio_din[89] 10 ssp1_rxd 11 ssp1_s_rxd_0
25-24			ssp1_txd_pad 00 ssp1_s_rxd_1 01 gpio_dout[88] / gpio_din[88] 10 ssp1_txd 11 ssp1_s_txd
23-22			ssp1_clk_pad 00 01 gpio_dout[87] / gpio_din[87]

			10 ssp1_clkout 11 ssp1_s_clkin
21-20			ssp1_csn_pad 00 MCU_gpio_din[125] / MCU_gpio_dout[125] 01 gpio_dout[86] / gpio_din[86] 10 ssp1_csn0_out 11 ssp1_s_cs
19-18			ssp2_d3_pad 00 01 gpio_din[85] / gpio_dout[85] 10 ssp2_rxd_tmp[3] / ssp2_txd[3] 11
17-16			ssp2_d2_pad 00 01 gpio_din[84] / gpio_dout[84] 10 ssp2_rxd_tmp[2] / ssp2_txd[2] 11
15-14			ssp2_d1_pad 00 01 gpio_din[83] / gpio_dout[83] 10 ssp2_rxd_tmp[1] / ssp2_txd[1] 11
13-12			ssp2_d0_pad 00 01 gpio_din[82] / gpio_dout[82] 10 ssp2_rxd_tmp[0] / ssp2_txd[0] 11
11-10			ssp2_clkout_pad 00 01 gpio_din[81] / gpio_dout[81] 10 ssp2_clkout 11
9-8			ssp2_csn0_out_pad 00 MCU_gpio_dout[124] / MCU_gpio_din[124] 01 gpio_din[80] / gpio_dout[80] 10 ssp2_csn0_out 11
7-6			ssp_d3_pad 00 MCU_gpio_din[88] / MCU_gpio_dout[88] 01 10 ssp_rxd_tmp[3] / ssp_txd[3] 11
5-4			ssp_d2_pad 00 MCU_gpio_din[87] / MCU_gpio_dout[87] 01 10 ssp_rxd_tmp[2] / ssp_txd[2] 11
3-2			ssp_d1_pad 00 MCU_gpio_din[86] / MCU_gpio_dout[86]

			01 10 ssp_rxd_tmp[1] / ssp_txd[1] 11
1-0			ssp_d0_pad 00 MCU_gpio_din[85] / MCU_gpio_dout[85] 01 10 ssp_rxd_tmp[0] / ssp_txd[0] 11 RC_12M

SYS_PADCTL6

Offset_Address: 0x0000_00d8

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

Bit	Type	Reset	Description
31-30			pwmin3_pad 00 01 gpio_din[95] / gpio_dout[95] 10 pwm_in[3] 11
29-28			pwmin2_pad 00 01 gpio_din[94] / gpio_dout[94] 10 pwm_in[2] 11
27-26			pwmin1_pad 00 01 gpio_din[93] / gpio_dout[93] 10 pwm_in[1] 11
25-24			pwmin0_pad 00 01 gpio_din[92] / gpio_dout[92] 10 pwm_in[0] 11
23-22			pwmout7_pad 00 01 gpio_din[91] / gpio_dout[91] 10 pwm_out[7] 11 uart3_re
21-20			pwmout6_pad 00 01 gpio_din[90] / gpio_dout[90] 10 pwm_out[6] 11 uart3_de
19-18			pwmout5_pad

			00 MCU_gpio_dout[100] / MCU_gpio_din[100] 01 10 pwm_out[5] 11 uart3_re
17-16			pwmout4_pad 00 MCU_gpio_dout[99] / MCU_gpio_din[99] 01 10 pwm_out[4] 11 uart3_de
15-14			pwmout3_pad 00 MCU_gpio_dout[98] / MCU_gpio_din[98] 01 10 pwm_out[3] 11
13-12			pwmout2_pad 00 MCU_gpio_dout[97] / MCU_gpio_din[97] 01 10 pwm_out[2] 11
11-10			pwmout1_pad 00 MCU_gpio_dout[96] / MCU_gpio_din[96] 01 10 pwm_out[1] 11
9-8			pwmout0_pad 00 MCU_gpio_dout[95] / MCU_gpio_din[95] 01 gpio_din[80] / gpio_dout[80] 10 ssp2_csn0_out 11
7-6			can1_stby_pad 00 MCU_gpio_din[94] / MCU_gpio_dout[94] 01 10 mcan1_stby 11 uart3_re
5-4			can1_tx_pad 00 MCU_gpio_din[93] / MCU_gpio_dout[93] 01 10 mcan1_tx 11
3-2			can1_rx_pad 00 MCU_gpio_din[92] / MCU_gpio_dout[92] 01 10 mcan1_rx 11
1-0			can0_stby_pad 00 MCU_gpio_din[91] / MCU_gpio_dout[91] 01 10 mcan0_stby 11 uart3_de

SYS_PADCTL7

Offset_Address: 0x0000_00dc

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

Bit	Type	Reset	Description
31-30			itu_d7_pad 00 01 gpio_din[106] / gpio_dout[106] 10 itu_vin[7] 11
29-28			itu_d6_pad 00 01 gpio_din[105] / gpio_dout[105] 10 itu_vin[6] 11 ext2_clk
27-26			itu_d5_pad 00 01 gpio_din[104] / gpio_dout[104] 10 itu_vin[5] 11 ext2_clk
25-24			itu_d4_pad 00 01 gpio_din[103] / gpio_dout[103] 10 itu_vin[4] 11 ext2_clk
23-22			itu_d3_pad 00 01 gpio_din[102] / gpio_dout[102] 10 itu_vin[3] 11 ext1_clk
21-20			itu_d2_pad 00 01 gpio_din[101] / gpio_dout[101] 10 itu_vin[2] 11 ext1_clk
19-18			itu_d1_pad 00 01 gpio_din[100] / gpio_dout[100] 10 itu_vin[1] 11 ext1_clk
17-16			itu_d0_pad 00 01 gpio_din[99] / gpio_dout[99] 10 itu_vin[0]

			11 ext1_clk
15-14			sda0_pad 00 01 gpio_din[98] / gpio_dout[98] 10 i2c_sda_in 11
13-12			scl0_pad 00 01 gpio_din[97] / gpio_dout[97] 10 i2c_sck_in 11
11-10			rcrt_in_pad 00 01 gpio_din[96] / gpio_dout[96] 10 rcrt_in 11
9-8			tck_pad 00 MCU_gpio_din[123] / MCU_gpio_dout[123] 01 10 TCK 11
7-6			tdo_pad 00 MCU_gpio_din[122] / MCU_gpio_dout[122] 01 10 TDO 11 uart2_rts
5-4			tdi_pad 00 MCU_gpio_din[121] / MCU_gpio_dout[121] 01 10 TDI 11 uart2_cts
3-2			tms_pad 00 MCU_gpio_din[120] / MCU_gpio_dout[120] 01 10 TMS 11 uart1_rts
1-0			ntrst_pad 00 MCU_gpio_din[119] / MCU_gpio_dout[119] 01 10 nTRST 11 uart1_cts

SYS_PADCTL8

Offset_Address: 0x0000_00e0

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

Bit	Type	Reset	Description
31-8			Reserved
7-6			mclk_out_pad 00 01 gpio_din[110] / gpio_dout[110] 10 ext2_clk 11
5-4			itu_vsync_pad 00 01 gpio_din[109] / gpio_dout[109] 10 itu_vsync 11
3-2			itu_hsync_pad 00 01 gpio_din[108] / gpio_dout[108] 10 itu_hsync 11
1-0			itu_clk_pad 00 01 gpio_din[107] / gpio_dout[107] 10 itu_clk 11

SYS_PADCTL9

Offset_Address: 0x0000_00e4

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

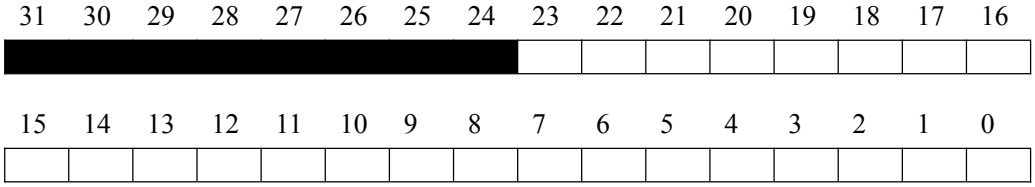
Bit	Type	Reset	Description
31-30			lcd_d15_pad 00 MCU_gpio_din[53] / MCU_gpio_dout[53] 01 gpio_din[53] / gpio_dout[53] 10 clcd_dout_new[15] 11
29-28			lcd_d14_pad 00 MCU_gpio_din[52] / MCU_gpio_dout[52] 01 gpio_din[52] / gpio_dout[52] 10 clcd_dout_new[14] 11
27-26			lcd_d13_pad 00 MCU_gpio_din[51] / MCU_gpio_dout[51] 01 gpio_din[51] / gpio_dout[51] 10 clcd_dout_new[13] 11

25-24			lcd_d12_pad 00 MCU_gpio_din[50] / MCU_gpio_dout[50] 01 gpio_din[50] / gpio_dout[50] 10 clcd_dout_new[12] 11
23-22			lcd_d11_pad 00 MCU_gpio_din[49] / MCU_gpio_dout[49] 01 gpio_din[49] / gpio_dout[49] 10 clcd_dout_new[11] 11
21-20			lcd_d10_pad 00 MCU_gpio_din[48] / MCU_gpio_dout[48] 01 gpio_din[48] / gpio_dout[48] 10 clcd_dout_new[10] 11
19-18			lcd_d9_pad 00 MCU_gpio_din[102] / MCU_gpio_dout[102] 01 10 clcd_dout_new[9] 11
17-16			lcd_d8_pad 00 MCU_gpio_din[101] / MCU_gpio_dout[101] 01 10 clcd_dout_new[8] 11
15-14			lcd_d7_pad 00 MCU_gpio_din[47] / MCU_gpio_dout[47] 01 gpio_din[47] / gpio_dout[47] 10 clcd_dout_new[7] 11
13-12			lcd_d6_pad 00 MCU_gpio_din[46] / MCU_gpio_dout[46] 01 gpio_din[46] / gpio_dout[46] 10 clcd_dout_new[6] 11
11-10			lcd_d5_pad 00 MCU_gpio_din[45] / MCU_gpio_dout[45] 01 gpio_din[45] / gpio_dout[45] 10 clcd_dout_new[5] 11
9-8			lcd_d4_pad 00 MCU_gpio_din[44] / MCU_gpio_dout[44] 01 gpio_din[44] / gpio_dout[44] 10 clcd_dout_new[4] 11
7-6			lcd_d3_pad 00 MCU_gpio_din[43] / MCU_gpio_dout[43] 01 gpio_din[43] / gpio_dout[43] 10 clcd_dout_new[3]

			11
5-4			lcd_d2_pad 00 MCU_gpio_din[42] / MCU_gpio_dout[42] 01 gpio_din[42] / gpio_dout[42] 10 clcd_dout_new[2] 11
3-2			lcd_d1_pad 00 MCU_gpio_din[41] / MCU_gpio_dout[41] 01 gpio_din[41] / gpio_dout[41] 10 clcd_dout_new[1] 11
1-0			lcd_d0_pad 00 MCU_gpio_din[40] / MCU_gpio_dout[40] 01 gpio_din[40] / gpio_dout[40] 10 clcd_dout_new[0] 11

SYS_PADCTL10

Offset_Address: 0x0000_00e8



Bit	Type	Reset	Description
31-24			Reserved
23-22			lcd_hsync_pad 00 MCU_gpio_din[57] / MCU_gpio_dout[57] 01 gpio_din[57] / gpio_dout[57]
21-20			lcd_vsync_pad 00 MCU_gpio_din[56] / MCU_gpio_dout[56] 01 gpio_din[56] / gpio_dout[56] 10 clcd_fp 11
19-18			lcd_clk_pad 00 MCU_gpio_din[55] / MCU_gpio_dout[55] 01 gpio_din[55] / gpio_dout[55] 10 clcd_cp 11
17-16			lcd_de_pad 00 MCU_gpio_din[54] / MCU_gpio_dout[54] 01 gpio_din[54] / gpio_dout[54] 10 clcd_ac 11
15-14			lcd_d23_pad 00 MCU_gpio_din[110] / MCU_gpio_dout[110] 01

			10 clcd_dout_new[23] 11
13-12			lcd_d22_pad 00 MCU_gpio_din[109] / MCU_gpio_dout[109] 01 10 clcd_dout_new[22] 11
11-10			lcd_d21_pad 00 MCU_gpio_din[108] / MCU_gpio_dout[108] 01 10 clcd_dout_new[21] 11
9-8			lcd_d20_pad 00 MCU_gpio_din[107] / MCU_gpio_dout[107] 01 10 clcd_dout_new[20] 11
7-6			lcd_d19_pad 00 MCU_gpio_din[106] / MCU_gpio_dout[106] 01 10 clcd_dout_new[19] 11
5-4			lcd_d18_pad 00 MCU_gpio_din[105] / MCU_gpio_dout[105] 01 10 clcd_dout_new[18] 11
3-2			lcd_d17_pad 00 MCU_gpio_din[104] / MCU_gpio_dout[104] 01 10 clcd_dout_new[17] 11
1-0			lcd_d16_pad 00 MCU_gpio_din[103] / MCU_gpio_dout[103] 01 10 clcd_dout_new[16] 11

SYS_PADCTL11

Offset_Address: 0x0000_00ec

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

Bit	Type	Reset	Description
31-30			sadc17_pad 00

			01 gpio_din[126] / gpio_dout[126] 10 11
29-28			sadc16_pad 00 01 gpio_din[125] / gpio_dout[125] 10 11
27-26			sadc15_pad 00 01 gpio_din[124] / gpio_dout[124] 10 11
25-24			sadc14_pad 00 01 gpio_din[123] / gpio_dout[123] 10 11
23-22			sadc13_pad 00 01 gpio_din[122] / gpio_dout[122] 10 11
21-20			sadc12_pad 00 01 gpio_din[121] / gpio_dout[121] 10 11
19-18			sadc11_pad 00 01 gpio_din[120] / gpio_dout[120] 10 11
17-16			sadc10_pad 00 01 gpio_din[119] / gpio_dout[119] 10 11
15-14			sadc07_pad 00 01 gpio_din[118] / gpio_dout[118] 10 11
13-12			sadc06_pad 00 01 gpio_din[117] / gpio_dout[117] 10 11
11-10			sadc05_pad

			00 01 gpio_din[116] / gpio_dout[116] 10 11
9-8			sadc04_pad 00 01 gpio_din[115] / gpio_dout[115] 10 11
7-6			sadc03_pad 00 01 gpio_din[114] / gpio_dout[114] 10 11
5-4			sadc02_pad 00 01 gpio_din[113] / gpio_dout[113] 10 11
3-2			sadc01_pad 00 01 gpio_din[112] / gpio_dout[112] 10 11
1-0			sadc00_pad 00 01 gpio_din[111] / gpio_dout[111] 10 11

SYS_PADCTL12

Offset_Address: 0x0000_00f0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

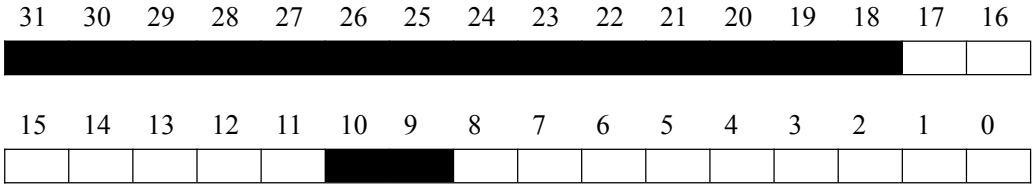
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-16			Reserved
15-14			sadc27_pad 00 MCU_gpio_din[118] / MCU_gpio_dout[118] 01 10 11
13-12			sadc26_pad 00 MCU_gpio_din[117] / MCU_gpio_dout[117] 01 10

			11
11-10			sadc25_pad 00 MCU_gpio_din[116] / MCU_gpio_dout[116] 01 10 11
9-8			sadc24_pad 00 MCU_gpio_din[115] / MCU_gpio_dout[115] 01 10 11
7-6			sadc23_pad 00 MCU_gpio_din[114] / MCU_gpio_dout[114] 01 10 11
5-4			sadc22_pad 00 MCU_gpio_din[113] / MCU_gpio_dout[113] 01 10 11
3-2			sadc21_pad 00 MCU_gpio_din[112] / MCU_gpio_dout[112] 01 10 11
1-0			sadc20_pad 00 MCU_gpio_din[111] / MCU_gpio_dout[111] 01 10 11

SYS_PADCTL13

Offset_Address: 0x0000_00f4



Bit	Type	Reset	Description
31-18			Reserved
17			SEL25 for PADCELL
16			SEL18 for PADCELL
15			can1_rx_fromRTC to can1_rx
14			can0_rx_fromRTC to can0_rx
13-8			reserved

7			i2s1_sdin_oe
6			i2s_sdin_oe
5			i2s1_sdout_oe
4			i2s_sdout_oe
3			ssp1_s_rxd sel
2			i2s1_sadata_in sel
1			i2s_sadata_in sel
0			clcd_dout_new arrange

SYS_PADCTL14

Offset_Address: 0x0000_00f8

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

Bit	Type	Reset	Description
31-30			GPIO143_pad 00 MCU_gpio_din[143] / MCU_gpio_dout[143] 01 gpio_din[143] / gpio_dout[143] 10 11
29-28			GPIO142_pad 00 MCU_gpio_din[142] / MCU_gpio_dout[142] 01 gpio_din[142] / gpio_dout[142] 10 11
27-26			GPIO141_pad 00 MCU_gpio_din[141] / MCU_gpio_dout[141] 01 gpio_din[141] / gpio_dout[141] 10 11
25-24			GPIO140_pad 00 MCU_gpio_din[140] / MCU_gpio_dout[140] 01 gpio_din[140] / gpio_dout[140] 10 11
23-22			GPIO139_pad 00 MCU_gpio_din[139] / MCU_gpio_dout[139] 01 gpio_din[139] / gpio_dout[139] 10 11
21-20			GPIO138_pad 00 MCU_gpio_din[138] / MCU_gpio_dout[138] 01 gpio_din[138] / gpio_dout[138] 10 11

19-18			GPIO137_pad 00 MCU_gpio_din[137] / MCU_gpio_dout[137] 01 gpio_din[137] / gpio_dout[137] 10 11
17-16			GPIO136_pad 00 MCU_gpio_din[136] / MCU_gpio_dout[136] 01 gpio_din[136] / gpio_dout[136] 10 11
15-14			GPIO135_pad 00 MCU_gpio_din[135] / MCU_gpio_dout[135] 01 gpio_din[135] / gpio_dout[135] 10 11
13-12			GPIO134_pad 00 MCU_gpio_din[134] / MCU_gpio_dout[134] 01 gpio_din[134] / gpio_dout[134] 10 11
11-10			GPIO133_pad 00 MCU_gpio_din[133] / MCU_gpio_dout[133] 01 gpio_din[133] / gpio_dout[133] 10 11
9-8			GPIO132_pad 00 MCU_gpio_din[132] / MCU_gpio_dout[132] 01 gpio_din[132] / gpio_dout[132] 10 11
7-6			GPIO131_pad 00 MCU_gpio_din[131] / MCU_gpio_dout[131] 01 gpio_din[131] / gpio_dout[131] 10 11
5-4			GPIO130_pad 00 MCU_gpio_din[130] / MCU_gpio_dout[130] 01 gpio_din[130] / gpio_dout[130] 10 11
3-2			GPIO129_pad 00 MCU_gpio_din[129] / MCU_gpio_dout[129] 01 gpio_din[129] / gpio_dout[129] 10 11
1-0			GPIO128_pad 00 MCU_gpio_din[128] / MCU_gpio_dout[128] 01 gpio_din[128] / gpio_dout[128] 10

			11

SYS_DRVCTL0 ~ SYS_DRVCTL15

Offset_Address: 0x0000_0100~0x0000_013c

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

Bit	Type	Reset	Description
31-0	R/W	0	MCU_pad_drv_ctl0 ~ MCU_pad_drv_ctl15 在对应sel_drv_ctl信号对应bit位低电平时传递给对应pad_drv_ctl信号。 0 pad_drv_ctl选择MCU核信号 1 pad_drv_ctl 选择 CPU 核信号

SYS_PUDCTL0 ~ SYS_PUDCTL14

Offset_Address: 0x0000_0140~0x0000_0178

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

Bit	Type	Reset	Description
31-0	R/W	0	MCU_pad_pud_ctl0 ~ MCU_pad_pud_ctl14 在对应sel_pud_ctl信号对应bit位低电平时传递给对应pad_pud_ctl信号。 0 pad_pud_ctl选择MCU核信号 1 pad_pud_ctl 选择 CPU 核信号

SEL_PADCTL0 ~ SEL_PADCTL14

Offset_Address: 0x0000_0200~0x0000_0238

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
						S									

Bit	Type	Reset	Description
31-0	R/W	0	sel_pad_ctl0 ~ sel_pad_ctl13 pad_ctl选择信号 00: pad_ctl选择MCU核信号 11: pad_ctl选择CPU核信号 Others: reserved

SEL_PADCTL15

Offset_Address: 0x0000_023C

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

Bit	Type	Reset	Description
31-16	R	0	reserved
15-0	R/W	0	sel_pwm [15]~[12]: select pwm in, 0: CPU; 1: 0; [11]~[8]: select pwm in, 0: MCU; 1: 0; [7]~[0]: select pwm out, 0: MCU; 1: CPU

SEL_DRVCTL0 ~ SEL_DRVCTL15

Offset_Address: 0x0000_0240~0x0000_027c

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

Bit	Type	Reset	Description
31-0	R/W	0	sel_drv_ctl0 ~ sel_drv_ctl15 pad_drv_ctl选择信号 0 pad_drv_ctl选择MCU核信号 1 pad_drv_ctl 选择 CPU 核信号

SEL_PUDCTL0 ~ SEL_PUDCTL14

Offset_Address: 0x0000_0280~0x0000_02b8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

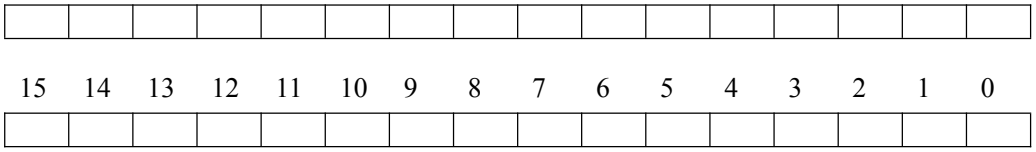
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-0	R/W	0	sel_pud_ctl0 ~ sel_pud_ctl14 pad_pud_ctl选择信号 0 pad_pud_ctl选择MCU核信号 1 pad_pud_ctl 选择 CPU 核信号

fuse_addr_reg

Offset_Address: 0x0000_01c0

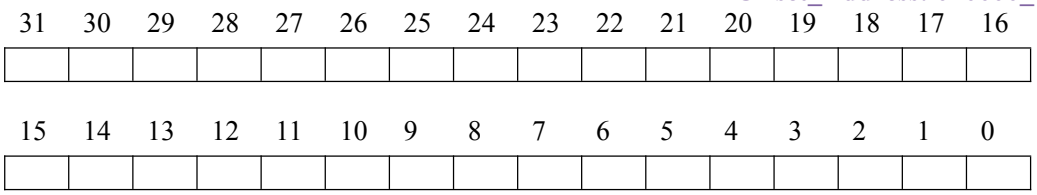
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16



Bit	Type	Reset	Description
31-0	R/W	0	CPU_addr_r

fuse_wdata_reg

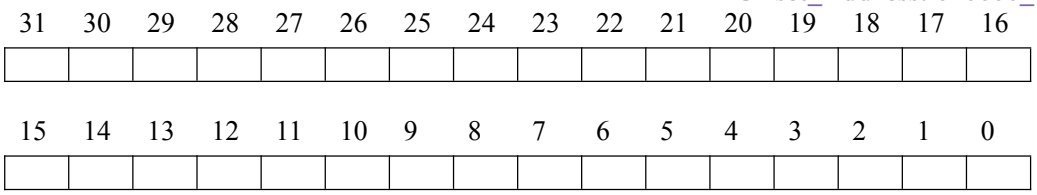
Offset_Address: 0x0000_01c4



Bit	Type	Reset	Description
31-0	R/W	0	MCU_fuse_wrdata

fuse_wmask_reg

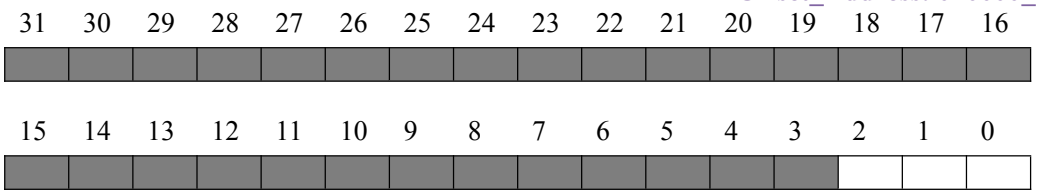
Offset_Address: 0x0000_01c8



Bit	Type	Reset	Description
31-0	R/W	0	MCU_fuse_wrmask

fuse_ctl_reg

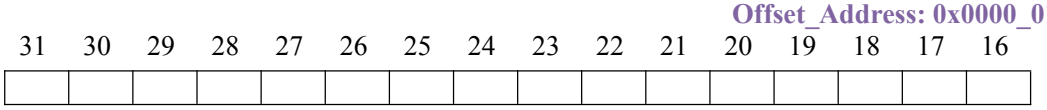
Offset_Address: 0x0000_01cc

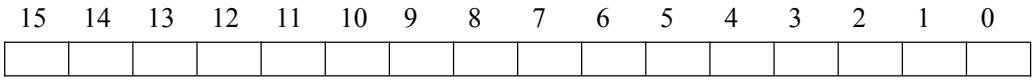


Bit	Type	Reset	Description
31-3	R/W	0	reserved
2	R/W	0	MCU_fuse_secrd_en
1	R/W	0	MCU_fuse_wren
0	R/W	0	MCU_fuse_op_start

fuse_cnt_num

Offset_Address: 0x0000_01d0

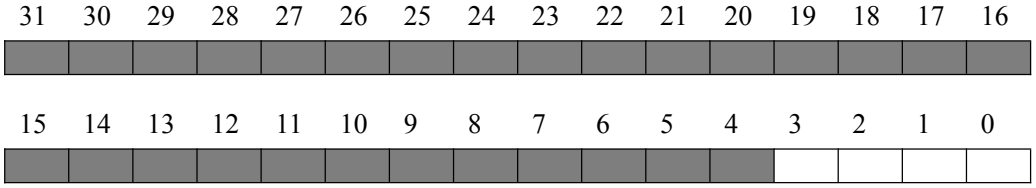




Bit	Type	Reset	Description
31-0	R/W	0	MCU_fuse_cnt_num

fuse_data5

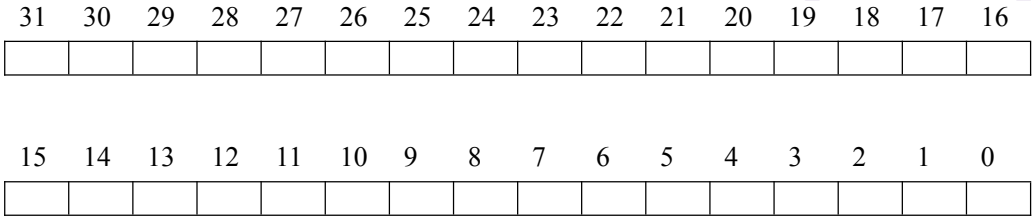
Offset_Address: 0x0000_01d4



Bit	Type	Reset	Description
31-4	R	0	Reserved
3	R	0	MCU_fuse_no_access
2	R	0	MCU_fuse_error
1	R	0	MCU_fuse_busy
0	R	0	MCU_fuse_op_done

MCU_fuse_rddata

Offset_Address: 0x0000_01d8



Bit	Type	Reset	Description
31-0	R/W	0	MCU_fuse_rddata

8.UART

8.1 UART Features

Support 2 channels 4-wire high-speed UART.
Support 4 channels normal UART for GPRS,Bluetooth, GPS etc.

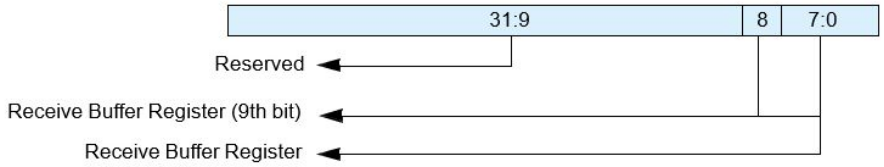
8.2 UART Register and Field Descriptions

UART0_REG_BASE: 0x5050_0000
UART1_REG_BASE: 0x5060_0000
UART2_REG_BASE: 0x5070_0000
UART3_REG_BASE: 0x5080_0000
UART0_REG_BASE(MCU): 0x5580_0000

8.2.1RBR

- **Name:** Receive Buffer Register
- **Size:** 32 bits
- **Address Offset: 0x00**
- **Read/write access:** read-only

This register can be accessed only when the DLAB bit (LCR[7]) is cleared.

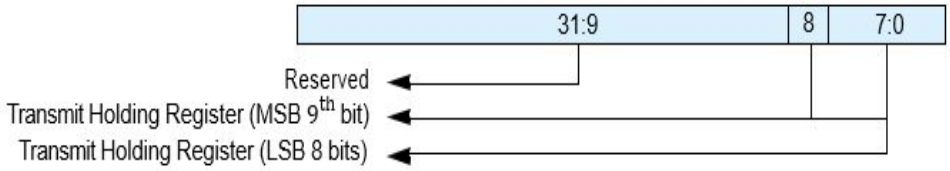


Bits	Name	R/W	Description
31:9	Reserved and read as 0		
8	Receive Buffer register (MSB 9th bit)	R	Data byte received on the serial input port (sin) in UART mode for the MSB 9th bit. It is applicable only when UART_9BIT_DATA_EN=1 **Reset Value**: 0x0
7:0	Receive Buffer Register (LSB 8 bits)	R	Data byte received on the serial input port (sin) in UART mode, or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line Status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE = NONE) or FIFOs are disabled (FCR[0] set to 0), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an over-run error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to 1), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO is preserved, but any incoming data are lost and an over-run error occurs. **Reset Value:** 0x0

8.2.2 THR

- **Name:** Transmit Holding Register
- **Size:** 32 bits
- **Address Offset: 0x00**
- **Read/write access:** write-only

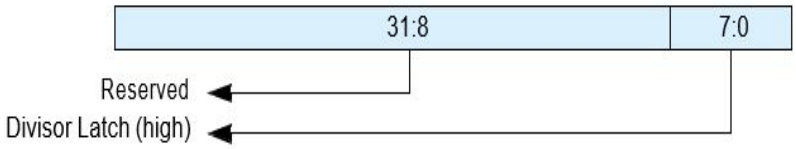
This register can be accessed only when the DLAB bit (LCR[7]) is cleared.



Bits	Name	R/W	Description
31:9	Reserved and read as 0		
8	Transmit Holding Register (MSB 9th bit)	W	Data to be transmitted on the serial output port (sout) in UART mode for the MSB 9th bit. It is applicable only when UART_9BIT_DATA_EN=1. **Reset Value**: 0x0
7:0	Transmit Holding Register (LSB 8 bits)	W	Data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] = 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] = 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost. **Reset Value:** 0x0

8.2.3DLH

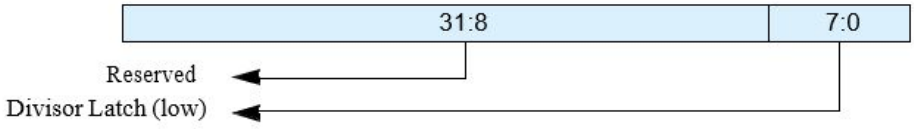
- **Name:** Divisor Latch High
- **Size:** 32 bits
- **Address Offset: 0x04**
- **Read/write access:** read/write



Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:0	Divisor Latch (High)	R/W	Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE = Enabled) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLH is set, at least 8 clock cycles of the slowest apb_uart clock should be allowed to pass before transmitting or receiving data. **Reset Value:** 0x0

8.2.4 DLL

- **Name:** Divisor Latch Low
- **Size:** 32 bits
- **Address Offset: 0x00**
- **Read/write access:** read/write

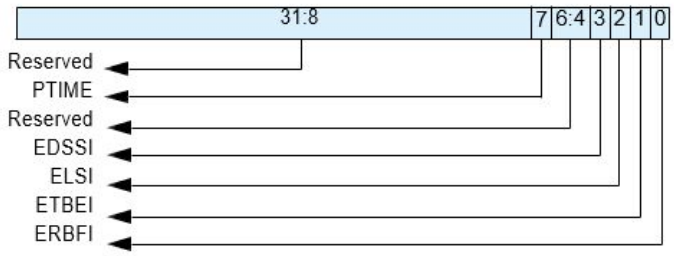


Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:0	Divisor Latch (Low)	R/W	Lower 8 bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART. The output baud rate is equal to the serial clock (pclk if one clock design, sclk if two clock design (CLOCK_MODE = Enabled) frequency divided by sixteen times the value of the baud rate divisor, as follows: baud rate = (serial clock freq) / (16 * divisor). Note that with the Divisor Latch Registers (DLL and DLH) set to 0, the baud clock is disabled and no serial communications occur. Also, once the DLL is set, at least 8 clock cycles of the slowest apb_uart clock should be allowed to pass before transmitting or receiving data. **Reset Value:** 0x0

8.2.5 IER

- **Name:** Interrupt Enable Register
- **Size:** 32 bits
- **Address Offset: 0x04**
- **Read/write access:** read/write

This register can be accessed only when the DLAB bit (LCR[7]) is cleared.

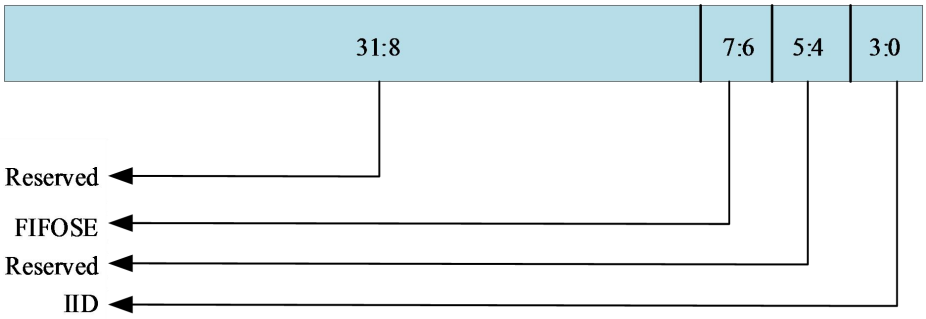


Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7	PTIME	R/W	Programmable THRE Interrupt Mode Enable that can be written to only when THRE_MODE_USER = Enabled, always readable. This is used to enable/disable the generation of THRE Interrupt. ▪ 0 – disabled ▪ 1 – enabled **Reset Value:** 0x0
6:4	Reserved and read as 0		
3	EDSSI	R/W	Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt. ▪ 0 – disabled ▪ 1 – enabled **Reset Value:** 0x0
2	ELSI	R/W	Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt. ▪ 0 – disabled ▪ 1 – enabled **Reset Value:** 0x0
1	ETBEI	R/W	Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt. ▪ 0 – disabled ▪ 1 – enabled **Reset Value:** 0x0
0	ERBFI	R/W	Enable Received Data Available Interrupt. This is used to enable/disable the generation of Received Data Available Interrupt and the Character Timeout Interrupt (if in FIFO mode and FIFOs enabled). These are the second highest priority interrupts. ▪ 0 – disabled ▪ 1 – enabled **Reset Value:** 0x0

8.2.6 IIR

- **Name:** Interrupt Identity Register
- **Size:** 32 bits
- **Address Offset: 0x08**

- **Read/write access:** read-only



Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:6	FIFOs Enabled (or FIFOSE)	R	FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. ▪ 00 – disabled ▪ 11 – enabled **Reset Value:** 0x00
5:4	Reserved	N/A	Reserved and read as 0
3:0	Interrupt ID (or IID)	R	Interrupt ID. This indicates the highest priority pending interrupt which can be one of the following types: ▪ 0000 – modem status ▪ 0001 – no interrupt pending ▪ 0010 – THR empty ▪ 0100 – received data available ▪ 0110 – receiver line status ▪ 0111 – busy detect ▪ 1100 – character timeout Bit 3 indicates an interrupt can only occur when the FIFOs are enabled and used to distinguish a Character Timeout condition interrupt. **Reset Value:** 0x01

Interrupt Control Functions

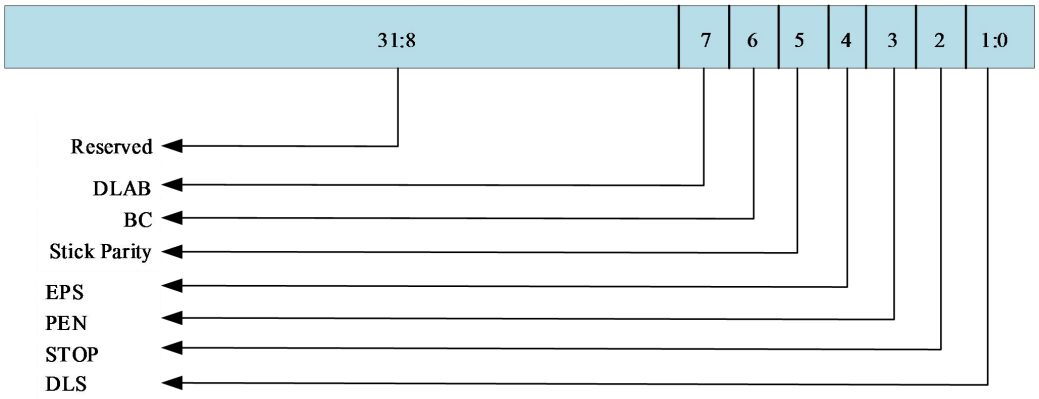
Interrupt ID				Interrupt Set and Reset Functions			
Bit 3	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	0	1	–	None	None	–
0	1	1	0	Highest	Receiver line status	Overrun/parity/ framing errors, break interrupt, or address received interrupt	Reading the line status register. In addition to LSR read, the Receiver line status is also cleared when RX_FIFO is read.
0	1	0	0	Second	Received data available	Receiver data available (non-FIFO mode or FIFOs disabled) or RCVR FIFO trigger level reached (FIFO mode and FIFOs enabled)	Reading the receiver buffer register (non-FIFO mode or FIFOs disabled) or the FIFO drops below the trigger level (FIFO mode and FIFOs enabled)
1	1	0	0	Second	Character timeout indication	No characters in or out of the RCVR FIFO during the last 4 character times and there is at least 1 character in it during this time	Reading the receiver buffer register

Interrupt Control Functions (Continued)

Interrupt ID				Interrupt Set and Reset Functions			
Bit 3	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Control
0	0	1	0	Third	Transmit holding register empty	Transmitter holding register empty (Prog. THRE Mode disabled) or XMIT FIFO at or below threshold (Prog. THRE Mode enabled)	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode not selected or disabled) or XMIT FIFO above threshold (FIFOs and THRE Mode selected and enabled).
0	0	0	0	Fourth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register
0	1	1	1	Fifth	Busy detect indication	UART_16550_COMPATIBLE = NO and master has tried to write to the Line Control Register while the apb_uart is busy (USR[0] is set to 1).	Reading the UART status register

8.2.7FCR

- **Name:** FIFO Control Register
- **Size:** 32 bits
- **Address Offset: 0x08**
- **Read/write access:** write-only



Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:6	RCVR Trigger (or RT)	W	RCVR Trigger. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. In auto flow control mode, this trigger is used to determine when the rts_n signal is de-asserted only when RTC_FCT is disabled. It also determines when the dma_rx_req_n signal is asserted in certain modes of operation. The following trigger levels are supported: ▪ 00 – 1 character in the FIFO ▪ 01 – FIFO ¼ full ▪ 10 – FIFO ½ full ▪ 11 – FIFO 2 less than full **Reset Value:** 0x0

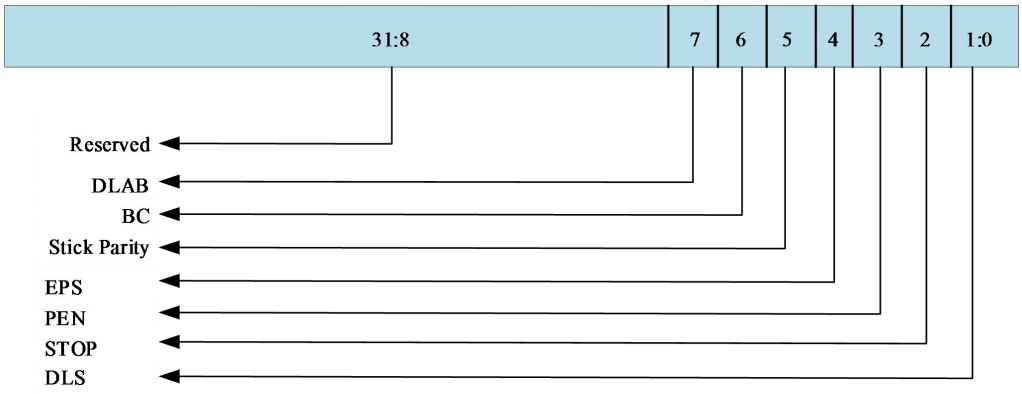
5:4	TX Empty Trigger (or TET)	W	TX Empty Trigger. Writes have no effect when THRE_MODE_USER = Disabled. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. It also determines when the dma_tx_req_n signal is asserted when in certain modes of operation. The following trigger levels are supported: ▪ 00 – FIFO empty ▪ 01 – 2 characters in the FIFO ▪ 10 – FIFO ¼ full ▪ 11 – FIFO ½ full **Reset Value:** 0x0

Bits	Name	R/W	Description
3	DMA Mode (or DMAM)	W	DMA Mode. This determines the DMA signaling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA = No). ▪ 0 – mode 0 ▪ 1 – mode 1 **Reset Value:** 0x0
2	XMIT FIFO Reset (or XFIFOR)	W	XMIT FIFO Reset. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA = YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit. **Reset Value:** 0x0
1	RCVR FIFO Reset (or RFIFOR)	W	RCVR FIFO Reset. This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA = YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit. **Reset Value:** 0x0
0	FIFO Enable (or FIFOE)	W	FIFO Enable. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. Whenever the value of this bit is changed both the XMIT and RCVR controller portion of FIFOs is reset. **Reset Value:** 0x0

8.2.8 LCR

- **Name: Line Control Register**

- **Size: 32 bits**
- **Address Offset: 0x0C**
- **Read/write access: read/write**



Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7	DLAB	R/W	Divisor Latch Access Bit. If UART_16550_COMPATIBLE = NO, then writeable only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH/LPDLL and LPDLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers. **Reset Value:** 0x0
6	Break (or BC)	R/W	Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to 1, the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE = Enabled and active (MCR[6] set to 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low. **Reset Value:** 0x0
5	Stick Parity	R/W	Stick Parity. If UART_16550_COMPATIBLE = NO, then writeable only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This bit is used to force parity value. When PEN, EPS, and Stick Parity are set to 1, the parity bit is transmitted and checked as logic 0. If PEN and Stick Parity are set to 1 and EPS is a logic 0, then parity bit is transmitted and checked as a logic 1. If this bit is set to 0, Stick Parity is disabled. **Reset Value:** 0x0

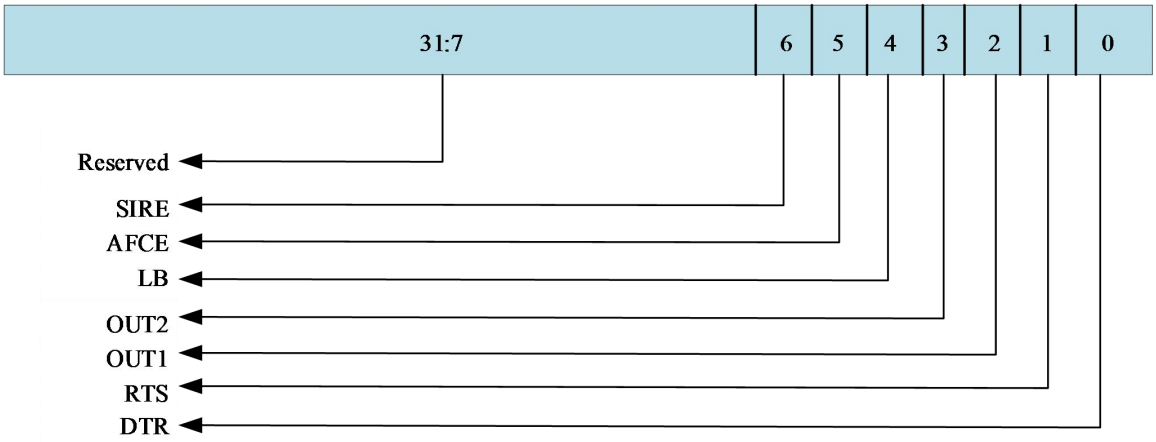
4	EPS	R/W	Even Parity Select. If UART_16550_COMPATIBLE = NO, then writeable only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to 1). If set to 1, an even number of logics 1s is transmitted or checked. If set to 0, an odd number of logics 1s is transmitted or checked. **Reset Value:** 0x0
3	PEN	R/W	Parity Enable. If UART_16550_COMPATIBLE = NO, then writeable only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. ▪ 0 – parity disabled ▪ 1 – parity enabled **Reset Value:** 0x0
2	STOP	R/W	Number of stop bits. If UART_16550_COMPATIBLE = NO, then writeable only when UART is not busy (USR[0] is 0); otherwise always writable, always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to 0, one stop bit is transmitted in the serial data. If set to 1 and the data bits are set to 5 (LCR[1:0] set to 0) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. ▪ 0 – 1 stop bit ▪ 1 – 1.5 stop bits when DLS (LCR[1:0]) is 0, else 2 stop bit NOTE: The STOP bit duration implemented by apb_uart may appear longer due to idle time inserted between characters for some configurations and baud clock divisor values in the transmit direction **Reset Value:** 0x0

Bits	Name	R/W	Description
1:0	DLS (or CLS, as used in legacy)	R/W	Data Length Select. If UART_16550_COMPATIBLE = NO, then writable only when UART is not busy (USR[0] is 0); otherwise always writable and readable. When DLS_E in LCR_EXT is set to 0, this register is used to select the number of data bits per character that the peripheral transmits and receives. The number of bits that may be selected are as follows: ▪ 00 – 5 bits ▪ 01 – 6 bits ▪ 10 – 7 bits ▪ 11 – 8 bits **Reset Value:** 0x0

8.2.9 MCR

- **Name: Modem Control Register**

- **Size: 32 bits**
- **Address Offset: 0x10**
- **Read/write access: read/write**



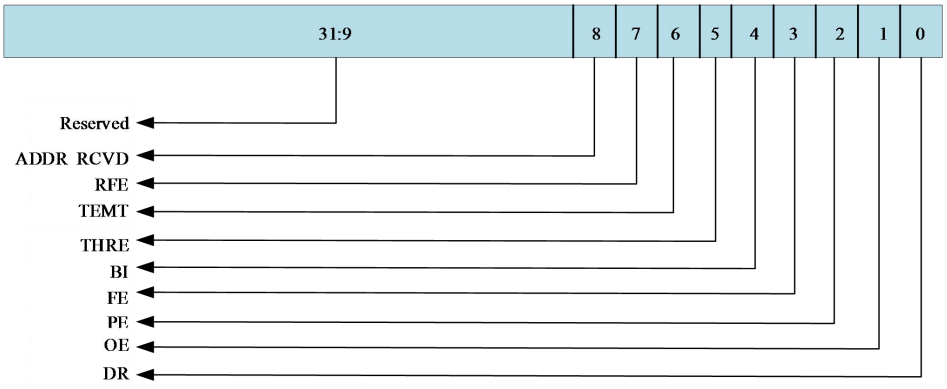
Bits	Name	R/W	Description
31:7	Reserved and read as 0		
6	SIRE	R/W	SIR Mode Enable. Writeable only when SIR_MODE = Enabled, always readable ▪ 0 – IrDA SIR Mode disabled ▪ 1 – IrDA SIR Mode enabled **Reset Value:** 0x0
5	AFCE	R/W	Auto Flow Control Enable. Writeable only when AFCE_MODE = Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set ▪ 0 – Auto Flow Control Mode disabled ▪ 1 – Auto Flow Control Mode enabled **Reset Value:** 0x0

Bits	Name	R/W	Description
4	LoopBack (or LB)	R/W	LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to 0), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE = Enabled AND active, MCR[6] set to 1), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line. **Reset Value:** 0x0

3	OUT2	R/W	OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: ▪ 0 – out2_n de-asserted (logic 1) ▪ 1 – out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to 1), the out2_n output is held inactive high while the value of this location is internally looped back to an input. **Reset Value:** 0x0
2	OUT1	R/W	OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: ▪ 0 – out1_n de-asserted (logic 1) ▪ 1 – out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to 1), the out1_n output is held inactive high while the value of this location is internally looped back to an input. **Reset Value:** 0x0
1	RTS	R/W	Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE = Enabled and active (MCR[5] set to 1) and FIFOs enable (FCR[0] set to 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold) only when the RTC Flow Trigger is disabled; otherwise it is gated by the receiver FIFO almost-full trigger, where "almost full" refers to two available slots in the FIFO (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to 1), the rts_n output is held inactive high while the value of this location is internally looped back to an input. **Reset Value:** 0x0
0	DTR	R/W	Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: ▪ 0 – dtr_n de-asserted (logic 1) ▪ 1 – dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to 1), the dtr_n output is held inactive high while the value of this location is internally looped back to an input. **Reset Value:** 0x0

8.2.10 LSR

- **Name:** Line Status Register
- **Size:** 32 bits
- **Address Offset: 0x14**
- **Read/write access:** read-only



Bits	Name	R/W	Description
31:9	Reserved and read as 0		
8	ADDR_RCVD	R/W	Address Received bit If 9-bit data mode (LCR_EXT[0]=1) is enabled, this bit is used to indicate that the 9th bit of the receive data is set to 1. This bit can also be used to indicate whether the incoming character is an address or data. ▪ 1 - Indicates that the character is an address. ▪ 0 - Indicates that the character is data. In the FIFO mode, since the 9th bit is associated with the received character, it is revealed when the character with the 9th bit set to 1 is at the top of the FIFO list. Reading the LSR clears the 9th bit. **NOTE**: You must ensure that an interrupt gets cleared (reading LSR register) before the next address byte arrives. If there is a delay in clearing the interrupt, then software will not be able to distinguish between multiple address related interrupt. **Reset Value**: 0x0
7	RFE	R	Receiver FIFO Error bit. This bit is only relevant when FIFO_MODE != NONE AND FIFOs are enabled (FCR[0] set to 1). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. ▪ 0 – no error in RX FIFO ▪ 1 – error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO. **Reset Value:** 0x0

Bits	Name	R/W	Description
6	TEMT	R	Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFOs enabled (FCR[0] set to 1), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty. **Reset Value:** 0x1
5	THRE	R	Transmit Holding Register Empty bit. If THRE_MODE_USER = Disabled or THRE mode is disabled (IER[7] set to 0) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If THRE_MODE_USER = Enabled AND FIFO_MODE != NONE and both modes are active (IER[7] set to 1 and FCR[0] set to 1 respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting. **Reset Value:** 0x1
4	BI	R	Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE = Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of *start time* + *data bits* + *parity* + *stop bits*. If in infrared mode (SIR_MODE = Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of *start time* + *data bits* + *parity* + *stop bits*. A break condition on serial input causes one and only one character, consisting of all 0s, to be received by the UART. In FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read. **NOTE:** If a FIFO is full when a break condition is received, a FIFO overrun occurs. The break condition and all the information associated with it—parity and framing errors—is discarded; any information that a break character was received is lost. **Reset Value:** 0x0
3	FE	R	Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data.

In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the apb_uart tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit; that is, data, and/or parity and stop.

It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). This happens because the break character implicitly generates a framing error by holding the sin input to logic 0 for longer than the duration of a character.

- 0 – no framing error
- 1 – framing error

Reading the LSR clears the FE bit.

Reset Value: 0x0

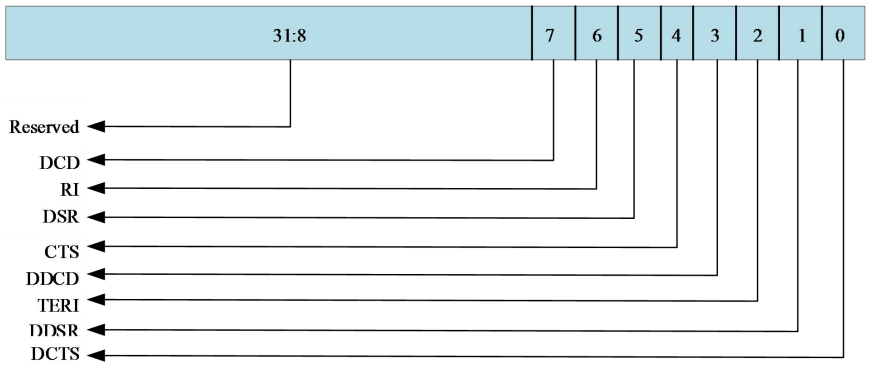
Bits	Name	R/W	Description
2	PE	R	Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) can be set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). In this situation, the Parity Error bit is set if parity generation and detection is enabled (LCR[3]=1) and the parity is set to odd (LCR[4]=0). - 0 – no parity error - 1 – parity error Reading the LSR clears the PE bit. **Reset Value:** 0x0
1	OE	R	Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. - 0 – no overrun error - 1 – overrun error Reading the LSR clears the OE bit. **Reset Value:** 0x0
0	DR	R	Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. - 0 – no data ready - 1 – data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode. **Reset Value:** 0x0

8.2.11 MSR

- **Name: Modem Status Register**
- **Size: 32 bits**
- **Address Offset: 0x18**
- **Read/write access: read-only**

Whenever bits 0, 1, 2 or 3 are set to logic 1, to indicate a change on the modem control inputs, a modem status interrupt is generated if enabled through the IER, regardless of when the change occurred. The bits of this register can be set after a reset—even though their respective modem signals are inactive—because the synchronized version of the modem signals have a reset value of 0 and change to value 1 after reset.

To prevent unwanted interrupts due to this change, a read of the MSR register can be performed after reset.



Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7	DCD	R	Data Carrier Detect. This is used to indicate the current state of the modem control line dcd_n. This bit is the complement of dcd_n. When the Data Carrier Detect input (dcd_n) is asserted it is an indication that the carrier has been detected by the modem or data set. ▪ 0 – dcd_n input is de-asserted (logic 1) ▪ 1 – dcd_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), DCD is the same as MCR[3] (Out2). **Reset Value:** 0x0

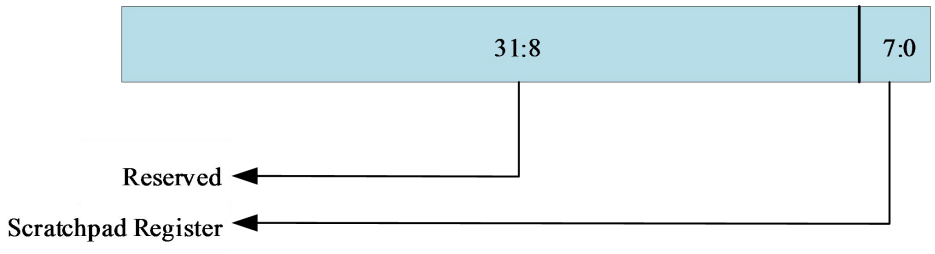
6	RI	R	Ring Indicator. This is used to indicate the current state of the modem control line ri_n. This bit is the complement of ri_n. When the Ring Indicator input (ri_n) is asserted, it is an indication that a telephone ringing signal has been received by the modem or data set. ▪ 0 – ri_n input is de-asserted (logic 1) ▪ 1 – ri_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), RI is the same as MCR[2] (Out1). **Reset Value:** 0x0
5	DSR	R	Data Set Ready. This is used to indicate the current state of the modem control line dsr_n. This bit is the complement of dsr_n. When the Data Set Ready input (dsr_n) is asserted it is an indication that the modem or data set is ready to establish communications with the apb_uart. ▪ 0 – dsr_n input is de-asserted (logic 1) ▪ 1 – dsr_n input is asserted (logic 0) In Loopback Mode (MCR[4] set to 1), DSR is the same as MCR[0] (DTR). **Reset Value:** 0x0
4	CTS	R	Clear to Send. This is used to indicate the current state of the modem control line cts_n. This bit is the complement of cts_n. When the Clear to Send input (cts_n) is asserted it is an indication that the modem or data set is ready to exchange data with the apb_uart. ▪ 0 – cts_n input is de-asserted (logic 1) ▪ 1 – cts_n input is asserted (logic 0) In Loopback Mode (MCR[4] = 1), CTS is the same as MCR[1] (RTS). **Reset Value:** 0x0

Bits	Name	R/W	Description

3	DDCD	R	Delta Data Carrier Detect. This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read. ▪ 0 – no change on dcd_n since last read of MSR ▪ 1 – change on dcd_n since last read of MSR Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] = 1), DDCD reflects changes on MCR[3] (Out2). Note, if the DDCD bit is not set and the dcd_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDCD bit is set when the reset is removed if the dcd_n signal remains asserted. **Reset Value:** 0x0
2	TERI	R	Trailing Edge of Ring Indicator. This is used to indicate that a change on the input ri_n (from an active-low to an inactive-high state) has occurred since the last time the MSR was read. ▪ 0 – no change on ri_n since last read of MSR ▪ 1 – change on ri_n since last read of MSR Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] = 1), TERI reflects when MCR[2] (Out1) has changed state from a high to a low. **Reset Value:** 0x0
1	DDSR	R	Delta Data Set Ready. This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read. ▪ 0 – no change on dsr_n since last read of MSR ▪ 1 – change on dsr_n since last read of MSR Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] = 1), DDSR reflects changes on MCR[0] (DTR). Note, if the DDSR bit is not set and the dsr_n signal is asserted (low) and a reset occurs (software or otherwise), then the DDSR bit is set when the reset is removed if the dsr_n signal remains asserted. **Reset Value:** 0x0
0	DCTS	R	Delta Clear to Send. This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read. ▪ 0 – no change on cts_n since last read of MSR ▪ 1 – change on cts_n since last read of MSR Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] = 1), DCTS reflects changes on MCR[1] (RTS). Note, if the DCTS bit is not set and the cts_n signal is asserted (low) and a reset occurs (software or otherwise), then the DCTS bit is set when the reset is removed if the cts_n signal remains asserted. **Reset Value:** 0x0

8.2.12 SCR

- **Name: Scratchpad Register**
- **Size: 32 bits**
- **Address Offset: 0x1C**
- **Read/write access: read/write**



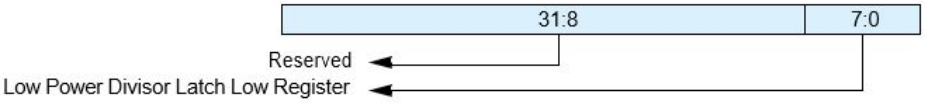
Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:0	Scratchpad Register	R/W	This register is for programmers to use as a temporary storage space. It has no defined purpose in the apb_uart. **Reset Value:** 0x0

8.2.13 LPDLL

- **Name: Low Power Divisor Latch Low Register**
- **Size: 32 bits**
- **Address Offset: 0x20**
- **Read/write access: read/write**

This register is only valid when the apb_uart is configured to have SIR low-power reception capabilities implemented (SIR_LP_RX = Yes). If SIR low-power reception capabilities are not implemented, this register does not exist and reading from this register address returns 0.

If UART_16550_COMPATIBLE = No, then this register can be accessed only when the DLAB bit (LCR[7]) is set and the UART is not busy—that is, USR[0] is 0; otherwise this register can be accessed only when the DLAB bit (LCR[7]) is set.



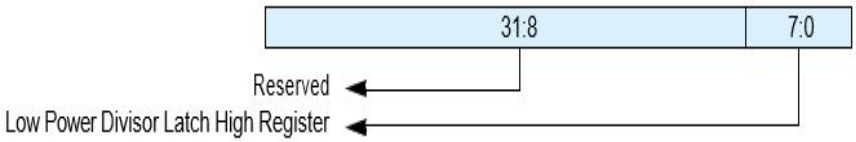
Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:0	LPDLL	R/W	This register makes up the lower 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock frequency)/(16* divisor) Therefore, a divisor must be selected to give a baud rate of 115.2K. **NOTE:** When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLL is set, at least eight clock cycles of the slowest apb_uart clock should be allowed to pass before transmitting or receiving data. **Reset Value:** 0x0

8.2.14 LPDLH

- **Name: Low Power Divisor Latch High Register**
- **Size: 32 bits**
- **Address Offset: 0x24**
- **Read/write access: read/write**

This register is valid only when the apb_uart is configured to have SIR low-power reception capabilities implemented (SIR_LP_RX = Yes). If SIR low-power reception capabilities are not implemented, this register does not exist and reading from this register address returns 0.

If UART_16550_COMPATIBLE = No, then this register can be accessed only when the DLAB bit (LCR[7]) is set and the UART is not busy—that is, USR[0] is 0; otherwise this register can be accessed only when the DLAB bit (LCR[7]) is set.



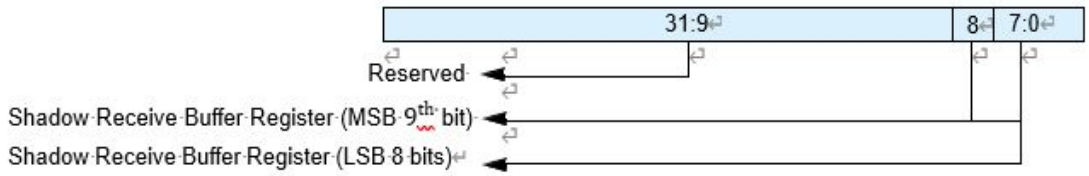
Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:0	LPDLH	R/W	This register makes up the upper 8-bits of a 16-bit, read/write, Low Power Divisor Latch register that contains the baud rate divisor for the UART, which must give a baud rate of 115.2K. This is required for SIR Low Power (minimum pulse width) detection at the receiver. The output low-power baud rate is equal to the serial clock (sclk) frequency divided by sixteen times the value of the baud rate divisor, as follows: Low power baud rate = (serial clock frequency)/(16* divisor) Therefore, a divisor must be selected to give a baud rate of 115.2K. **NOTE:** When the Low Power Divisor Latch registers (LPDLL and LPDLH) are set to 0, the low-power baud clock is disabled and no low-power pulse detection (or any pulse detection) occurs at the receiver. Also, once the LPDLH is set, at least eight clock cycles of the slowest apb_uart clock should be allowed to pass before transmitting or receiving data. **Reset Value:** 0x0

8.2.15 SRBR

- **Name: Shadow Receive Buffer Register**
- **Size: 32 bits**
- **Address Offset: 0x30 - 0x6C**
- **Read/write access: read-only**

This register is valid only when the apb_uart is configured to have additional shadow Register implemented (SHADOW = YES). If shadow Register are not implemented, this register does not exist and reading from this register address returns 0.

This register can be accessed only when the DLAB bit (LCR[7]) is cleared.



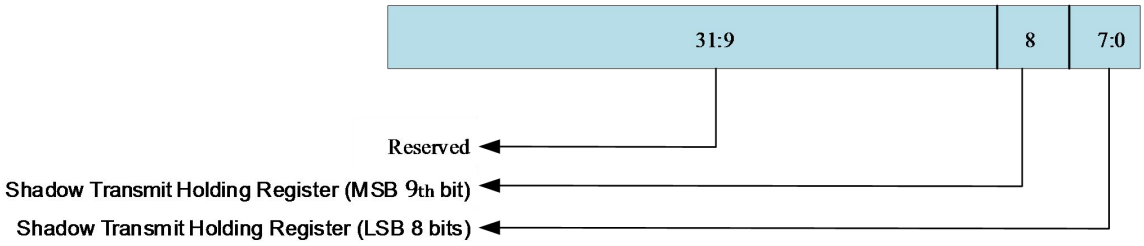
Bits	Name	R/W	Description
31:9	Reserved and read as 0		
8	Shadow Receive Buffer Register (MSB 9th bit)	R	This is a shadow register for the RBR[8] bit. It is applicable only when UART_9BIT_DATA_EN=1. **Reset Value**: 0x0
7:0	Shadow Receive Buffer Register (LSB 8 bits)	R	This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE = NONE) or FIFOs are disabled (FCR[0] set to 0), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to 1), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. **Reset Value:** 0x0

8.2.16 STHR

- **Name: Shadow Transmit Holding Register**
- **Size: 32 bits**
- **Address Offset: 0x30 - 0x6C**
- **Read/write access: write**

This register is valid only when the apb uart is configured to have additional shadow Register implemented (SHADOW = YES). If shadow Register are not implemented, this register does not exist, and reading from this register address returns 0.

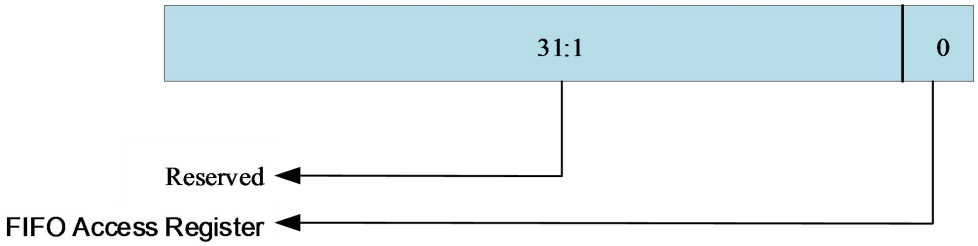
This register can be accessed only when the DLAB bit (LCR[7]) is cleared.



Bits	Name	R/W	Description
31:9	Reserved and read as 0		
8	Shadow Transmit Holding Register (MSB 9th bit)	W	This is a shadow register for the THR[8] bit. It is applicable only when UART_9BIT_DATA_EN=1 **Reset Value**: 0x0
7:0	Shadow Transmit Holding Register	W	This is a shadow register for the THR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains data to be transmitted on the serial output port (sout) in UART mode or the serial infrared output (sir_out_n) in infrared mode. Data should only be written to the THR when the THR Empty (THRE) bit (LSR[5]) is set. If in non-FIFO mode or FIFOs are disabled (FCR[0] set to 0) and THRE is set, writing a single character to the THR clears the THRE. Any additional writes to the THR before the THRE is set again causes the THR data to be overwritten. If in FIFO mode and FIFOs are enabled (FCR[0] set to 1) and THRE is set, x number of characters of data may be written to the THR before the FIFO is full. The number x (default=16) is determined by the value of FIFO Depth that you set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost. **Reset Value:** 0x0

8.2.17 FAR

- **Name: FIFO Access Register**
- **Size: 32 bits**
- **Address Offset: 0x70**
- **Read/write access: read/write**

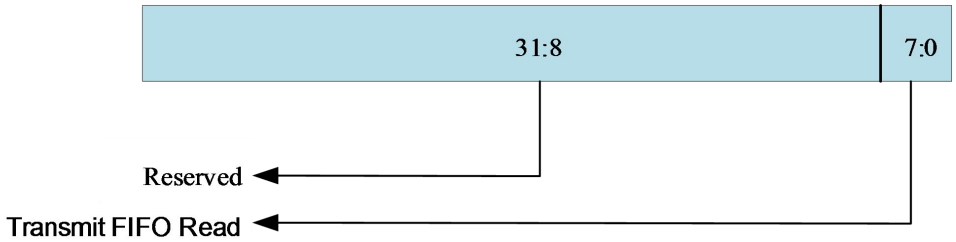


Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	FIFO Access Register	R/W	Writes have no effect when FIFO_ACCESS = No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. ▪ 0 – FIFO access mode disabled ▪ 1 – FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty. **Reset Value:** 0x0

8.2.18 TFR

- **Name: Transmit FIFO Read**
- **Size: 32 bits**
- **Address Offset: 0x74**
- **Read/write access: read-only**

This register is valid only when the apb_uart is configured to have the FIFO access test mode available (FIFO_ACCESS = YES). If not configured, this register does not exist and reading from this register address returns 0.

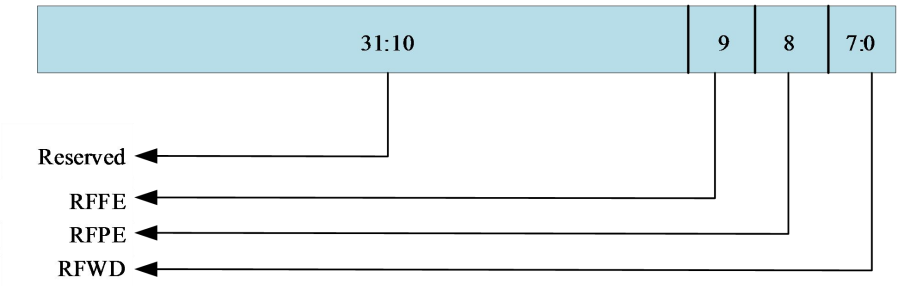


Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:0	Transmit FIFO Read	R	Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR. **Reset Value:** 0x0

8.2.19 RFW

- **Name: Receive FIFO Write**
- **Size: 32 bits**
- **Address Offset: 0x78**
- **Read/write access: write-only**

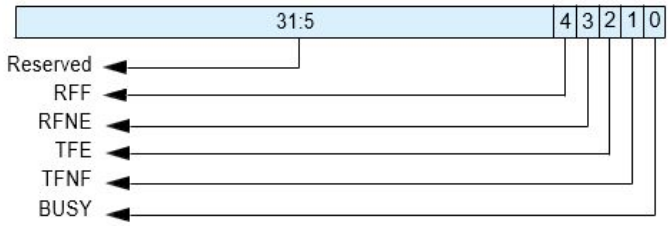
This register is valid only when the apb_uart is configured to have the FIFO access test mode available (FIFO_ACCESS = YES). If not configured, this register does not exist and reading from this register address returns 0.



Bits	Name	R/W	Description
31:10	Reserved and read as 0		
9	RFFE	W	Receive FIFO Framing Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write framing error detection information to the RBR. **Reset Value:** 0x0
8	RFPE	W	Receive FIFO Parity Error. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFOs are not implemented or not enabled, this bit is used to write parity error detection information to the RBR. **Reset Value:** 0x0
7:0	RFWD	W	Receive FIFO Write Data. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to 1). When FIFOs are implemented and enabled, the data that is written to the RFWD is pushed into the receive FIFO. Each consecutive write pushes the new data to the next write location in the receive FIFO. When FIFOs are not implemented or not enabled, the data that is written to the RFWD is pushed into the RBR. **Reset Value:** 0x0

8.2.20 USR

- **Name: UART Status Register**
- **Size: 32 bits**
- **Address Offset: 0x7C**
- **Read/write access: read-only**



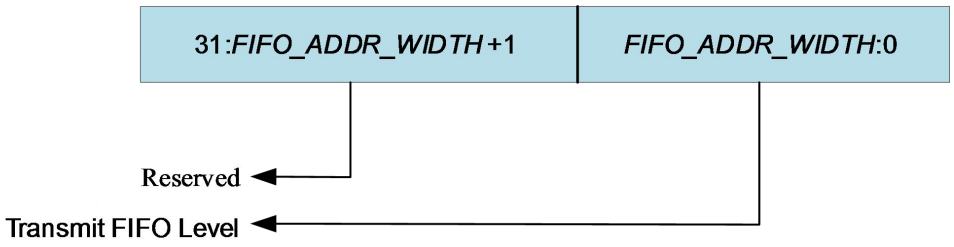
Bits	Name	R/W	Description
31:5	Reserved and read as 0		
4	RFF	R	Receive FIFO Full. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the receive FIFO is completely full. ▪ 0 – Receive FIFO not full ▪ 1 – Receive FIFO Full This bit is cleared when the RX FIFO is no longer full. **Reset Value:** 0x0
3	RFNE	R	Receive FIFO Not Empty. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the receive FIFO contains one or more entries. ▪ 0 – Receive FIFO is empty ▪ 1 – Receive FIFO is not empty This bit is cleared when the RX FIFO is empty. **Reset Value:** 0x0
2	TFE	R	Transmit FIFO Empty. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the transmit FIFO is completely empty. ▪ 0 – Transmit FIFO is not empty ▪ 1 – Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty. **Reset Value:** 0x1
1	TFNF	R	Transmit FIFO Not Full. This bit is only valid when FIFO_STAT = YES. This is used to indicate that the transmit FIFO in not full. ▪ 0 – Transmit FIFO is full ▪ 1 – Transmit FIFO is not full This bit is cleared when the TX FIFO is full. **Reset Value:** 0x1

Bits	Name	R/W	Description
0	BUSY	R	UART Busy. This bit is valid only when UART_16550_COMPATIBLE = NO and indicates that a serial transfer is in progress; when cleared, indicates that the apb_uart is idle or inactive. ▪ 0 – apb_uart is idle or inactive ▪ 1 – apb_uart is busy (actively transferring data) This bit will be set to 1 (busy) under any of the following conditions: 1. Transmission in progress on serial interface 2. Transmit data present in THR, when FIFO access mode is not being used (FAR = 0) and the baud divisor is non-zero ({DLH,DLL} does not equal 0) when the divisor latch access bit is 0 (LCR.DLAB = 0) 3. Reception in progress on the interface 4. Receive data present in RBR, when FIFO access mode is not being used (FAR = 0) **NOTE:** It is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the apb_uart has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE = Enabled), the assertion of this bit is also delayed by several cycles of the slower clock. **Reset Value:** 0x0

8.2.21 TFL

- **Name: Transmit FIFO Level**
- **Size: FIFO_ADDR_WIDTH + 1**
- **Address Offset: 0x80**
- **Read/write access: read-only**

This register is valid only when the apb_uart is configured to have additional FIFO status Register implemented (FIFO_STAT = YES). If status Register are not implemented, this register does not exist and reading from this register address returns 0.

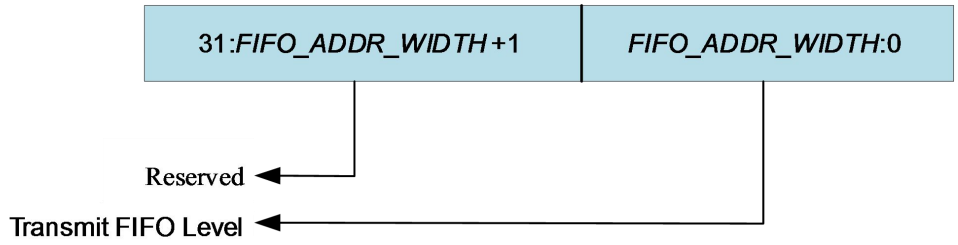


Bits	Name	R/W	Description
31:*FIFO_ADDR_WIDTH* + 1	Reserved and read as 0		
FIFO_ADDR_WIDTH:0	Transmit FIFO Level	R	Transmit FIFO Level. This indicates the number of data entries in the transmit FIFO. **Reset Value:** 0x0

8.2.22 RFL

- **Name: Receive FIFO Level**
- **Size: FIFO_ADDR_WIDTH + 1**
- **Address Offset: 0x84**
- **Read/write access: read-only**

This register is valid only when the apb_uart is configured to have additional FIFO status Register implemented (FIFO_STAT = YES). If status Register are not implemented, this register does not exist and reading from this register address returns 0.



Bits	Name	R/W	Description
31:*FIFO_ADDR_WIDTH* + 1	Reserved and read as 0		
FIFO_ADDR_WIDTH:0	Receive FIFO Level	R	Receive FIFO Level. This indicates the number of data entries in the receive FIFO. **Reset Value:** 0x0

8.2.23 SRR

- **Name: Software Reset Register**
- **Size: 32 bits**
- **Address Offset: 0x88**
- **Read/write access: write-only**

This register is valid only when the apb_uart is configured to have additional shadow Register implemented (SHADOW = YES). If shadow Register are not implemented, this register does

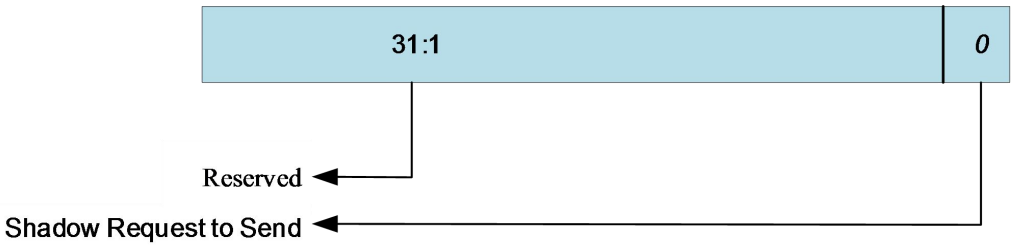
not exist and reading from this register address returns 0.

Bits	Name	R/W	Description
31:3	Reserved and read as 0		
2	XFR	W	XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA = YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit. **Reset Value:** 0x0 **Dependencies:** Writes have no effect when FIFO_MODE = None.
1	RFR	W	RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA = YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit. **Reset Value:** 0x0 **Dependencies:** Writes have no effect when FIFO_MODE = None.
0	UR	W	UART Reset. This asynchronously resets the apb_uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset. **Reset Value:** 0x0

8.2.24 SRTS

- **Name: Shadow Request to Send**
- **Size: 32 bits**
- **Address Offset: 0x8C**
- **Read/write access: read/write**

This register is valid only when the apb_uart is configured to have additional shadow Register implemented (SHADOW = YES). If shadow Register are not implemented, this register does not exist and reading from this register address returns 0.

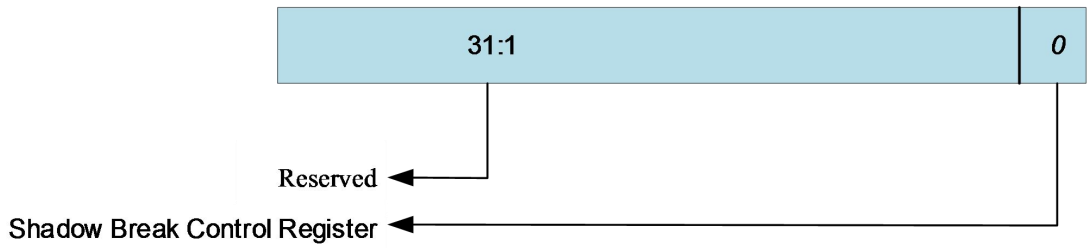


Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	Shadow Request to Send	R/W	Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the apb_uart is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE = Enabled and active (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold) only when RTC Flow Trigger is disabled; otherwise it is gated by the receiver FIFO almost-full trigger, where "almost full" refers to two available slots in the FIFO (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input. **Reset Value:** 0x0

8.2.25 SBCR

- **Name: Shadow Break Control Register**
- **Size: 32 bits**
- **Address Offset: 0x90**
- **Read/write access: read/write**

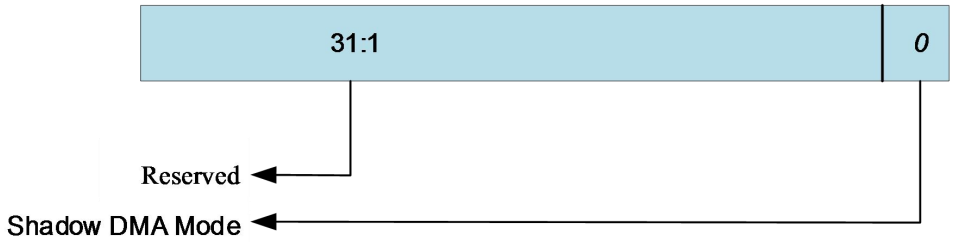
This register is valid only when the apb_uart is configured to have additional shadow Register implemented (SHADOW = YES). If shadow Register are not implemented, this register does not exist and reading from this register address returns 0.



Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	Shadow Break Control Register	R/W	Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device. If set to 1, the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE = Enabled and active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver. **Reset Value:** 0x0

8.2.26 SDMAM

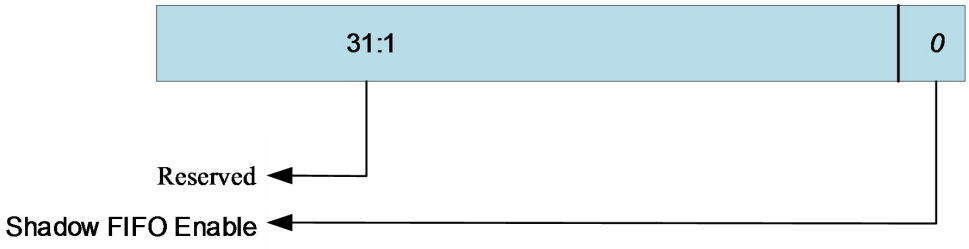
- **Name: Shadow DMA Mode**
- **Size: 32 bits**
- **Address Offset: 0x94**
- **Read/write access: read/write**



Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	Shadow DMA Mode	R/W	Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signaling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA = NO). ▪ 0 – mode 0 ▪ 1 – mode 1 **Reset Value:** 0x0

8.2.27 SFE

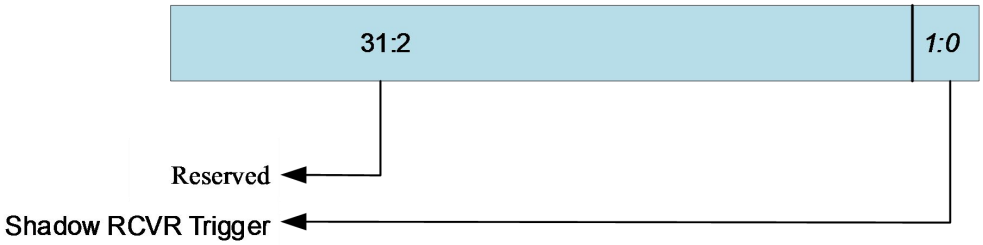
- **Name: Shadow FIFO Enable**
- **Size: 32 bits**
- **Address Offset: 0x98**
- **Read/write access: read/write**



Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	Shadow FIFO Enable	R/W	Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated. This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to 0 (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset. **Reset Value:** 0x0

8.2.28 SRT

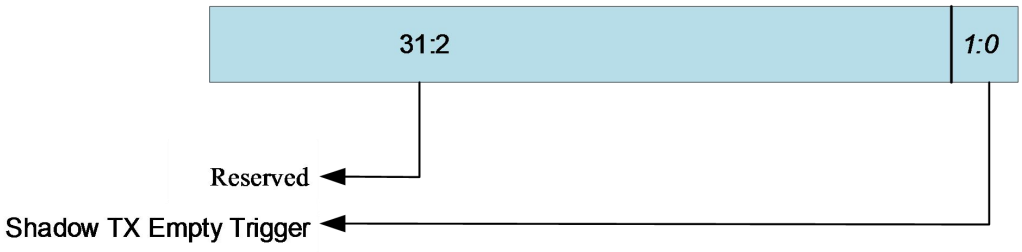
- **Name: Shadow RCVR Trigger**
- **Size: 32 bits**
- **Address Offset: 0x9C**
- **Read/write access: read/write**



Bits	Name	R/W	Description
31:2	Reserved and read as 0		
1:0	Shadow RCVR Trigger	R/W	Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. It also determines when the dma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported: ▪ 00 – 1 character in the FIFO ▪ 01 – FIFO ¼ full ▪ 10 – FIFO ½ full ▪ 11 – FIFO 2 less than full **Reset Value:** 0x0

8.2.29 STET

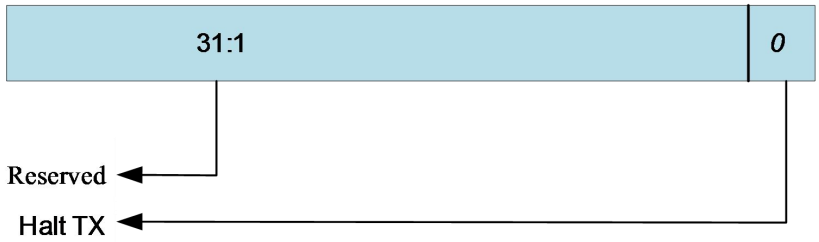
- **Name: Shadow TX Empty Trigger**
- **Size: 32 bits**
- **Address Offset: 0xA0**
- **Read/write access: read/write**



Bits	Name	R/W	Description
31:2	Reserved and read as 0		
1:0	Shadow TX Empty Trigger	R/W	Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: ▪ 00 – FIFO empty ▪ 01 – 2 characters in the FIFO ▪ 10 – FIFO ¼ full ▪ 11 – FIFO ½ full **Reset Value:** 0x0 **Dependencies:** Writes have no effect when THRE_MODE_USER = Disabled.

8.2.30 HTX

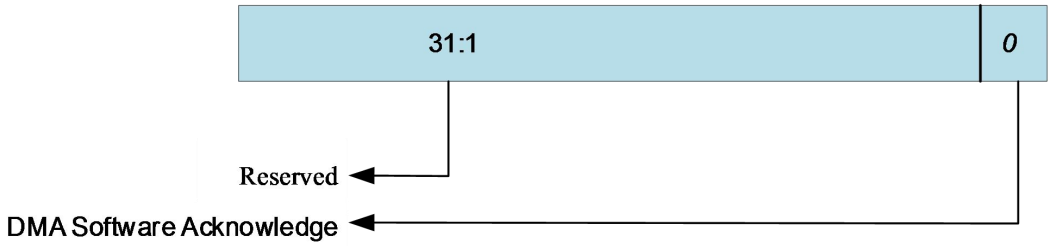
- **Name: Halt TX**
- **Size: 32 bits**
- **Address Offset: 0xA4**
- **Read/write access: read/write**



Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	Halt TX	R/W	This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. ▪ 0 – Halt TX disabled ▪ 1 – Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation. **Reset Value:** 0x0

8.2.31 DMASA

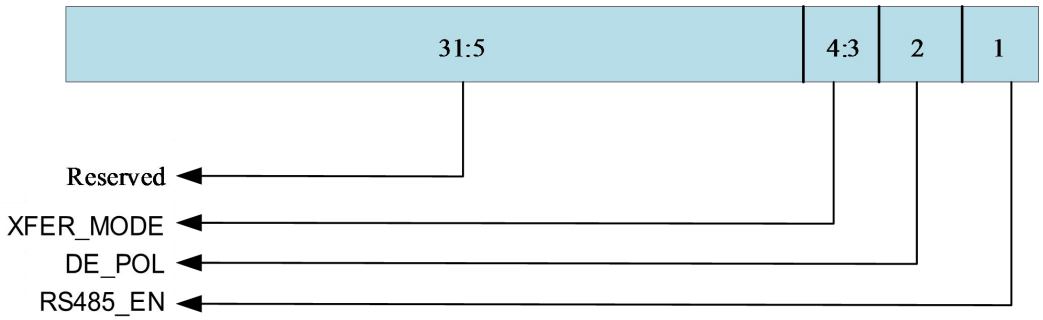
- **Name: DMA Software Acknowledge**
- **Size: 32 bits**
- **Address Offset: 0xA8**
- **Read/write access: write**



Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	DMA Software Acknowledge	W	This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the apb_uart should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit. **Reset Value:** 0x0 **Dependencies:** Writes have no effect when DMA_EXTRA = No.

8.2.32 TCR

- **Name: Transceiver Control Register (TCR)**
- **Size: 32 bits**
- **Address Offset: 0xAC**
- **Read/write access: read/write**



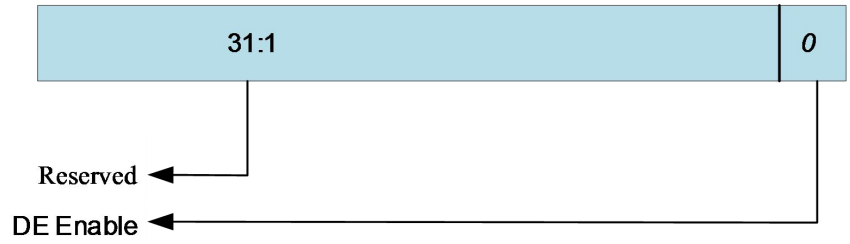
Bits	Name	R/W	Description
31:5	Reserved and read as 0		
4:3	XFER_MODE	R/W	Transfer Mode ■ 0 : In this mode, transmit and receive can happen simultaneously. You can enable DE_EN and RE_EN at any point of time. Turn around timing as programmed in the TAT register is not applicable in this mode. ■ 1 : In this mode, DE and RE are mutually exclusive. The hardware considers the turnaround timings that are programmed in the TAT register while switching from RE to DE or from DE to RE. Ensure that either DE or RE is expected to be enabled while programming. For transmission, hardware waits if it is in the midst of receiving any transfer, before it starts transmitting. ■ 2 : In this mode, DE and RE are mutually exclusive. Once DE_EN or RE_EN is programed, 're' is enabled by default and apb_uart controller will be ready to receive. If the user programs the TX FIFO with data, then apb_uart, after ensuring no receive is in progress, disables the 're' and enables the 'de' signal. Once the TX FIFO becomes empty, the 're' signal gets enabled and the 'de' signal will be disabled. In this mode of operation, the hardware considers the turnaround timings that are programmed in the TAT register while switching from RE to DE or from DE to RE. In this mode, 'de' and 're' signals are strictly complementary to each other. **Reset Value**: 0x0
2	DE_POL	R/W	Driver Enable Polarity ■ 1: DE signal is active high ■ 0: DE signal is active low **Reset Value**: UART_DE_POL
1	RE_POL	R/W	Receiver Enable Polarity ■ 1: RE signal is active high ■ 0: RE signal is active low **Reset Value**: UART_RE_POL

Bits	Name	R/W	Description
0	RS485_EN	R/W	RS485 Transfer Enable ■ 0 : In this mode, the transfers are still in the RS232 mode. All other fields in this register are reserved and registers DE_EN, RE_EN, DET and TAT are reserved. ■ 1: In this mode, the transfers will happen in RS485 mode. All other fields of this register are applicable. **Reset Value**: 0x0

8.2.33 DE_EN

- **Name: Driver Output Enable Register (DE_EN)**
- **Size: 32 bits**
- **Address Offset: 0xB0**
- **Read/write access: read/write**

The Driver Output Enable Register (DE_EN) is used to control the assertion and de-assertion of the DE signal.



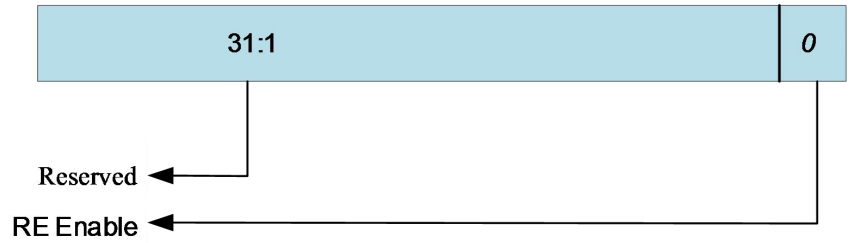
Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	DE Enable	R/W	DE Enable control The 'DE Enable' register bit is used to control assertion and de-assertion of 'de' signal. ■ 0: De-assert 'de' signal ■ 1: Assert 'de' signal **Reset Value**: 0x0

8.2.34 RE_EN

- **Name: Receiver Output Enable Register (RE_EN)**
- **Size: 32 bits**
- **Address Offset: 0xB4**

- **Read/write access: read/write**

The Receiver Output Enable Register (RE_EN) is used to control the assertion and de-assertion of the RE signal.

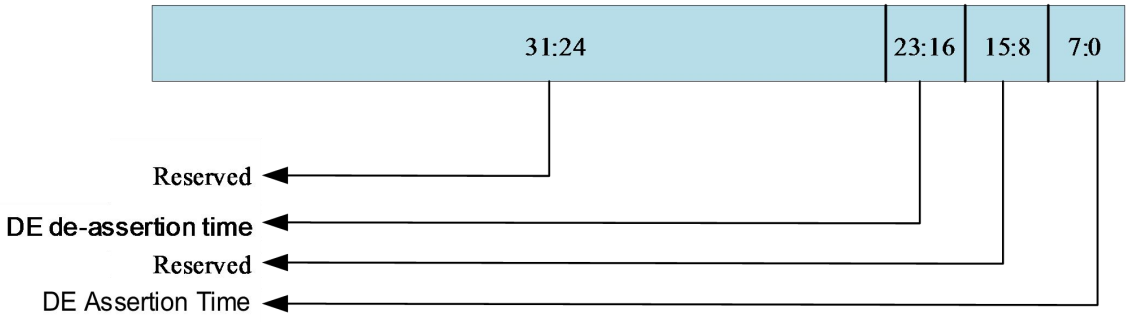


Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	RE Enable	R/W	RE Enable control The 'RE Enable' register bit is used to control assertion and de-assertion of 're' signal. ▪ 0: De-assert 're' signal ▪ 1: Assert 're' signal **Reset Value**: 0x0

8.2.35 DET

- **Name: Driver Output Enable Timing Register (DET)**
- **Size: 32 bits**
- **Address Offset: 0xB8**
- **Read/write access: read/write**

The Driver Output Enable Timing Register (DET) is used to control the DE assertion and de-assertion timings of 'de' signal.

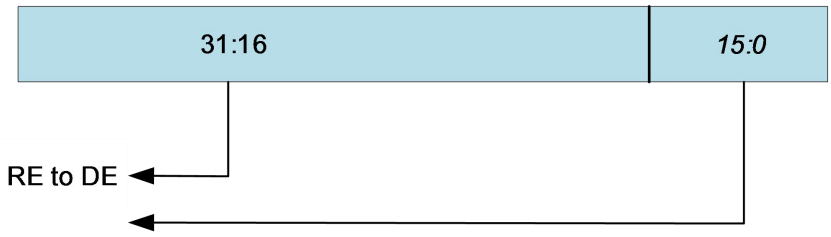


Bits	Name	R/W	Description
31:24	Reserved and read as 0		
23:16	DE de-assertion time	R/W	Driver enable de-assertion time. This field controls the amount of time (in terms of number of serial clock periods) between the end of stop bit on the serial output (sout) to the falling edge of Driver output enable signal. **Reset Value**: 0x0
15:8	Reserved and read as 0		
7:0	DE assertion time	R/W	Driver enable assertion time. This field controls the amount of time (in terms of number of serial clock periods) between the assertion of rising edge of Driver output enable signal to serial transmit enable. Any data in transmit buffer, will start on serial output (sout) after the transmit enable. **Reset Value**: 0x0

8.2.36 TAT

- **Name: TurnAround Timing Register (TAT)**
- **Size: 32 bits**
- **Address Offset: 0xBC**
- **Read/write access: read/write**

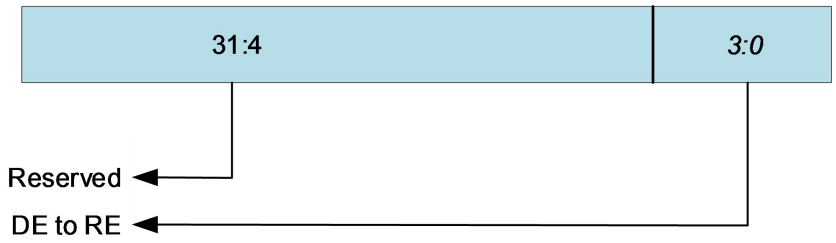
The TurnAround Timing Register (TAT) is used to hold the turnaround time between switching of 're' and 'de' signals.



Bits	Name	R/W	Description
31:16	RE to DE	R/W	Receiver Enable to Driver Enable TurnAround time. Turnaround time (in terms of serial clock) for RE de-assertion to DE assertion. **Note**: ▪ If the DE assertion time in the DET register is 0, then the actual value is the programmed value + 3. ▪ If the DE assertion time in the DET register is 1, then the actual value is the programmed value + 2. ▪ If the DE assertion time in the DET register is greater than 1, then the actual value is the programmed value + 1. **Reset Value**: 0x0
15:0	DE to RE	R/W	Driver Enable to Receiver Enable TurnAround time. Turnaround time (in terms of serial clock) for DE de-assertion to RE assertion. **Note**: The actual time is the programmed value + 1. **Reset Value**: 0x0

8.2.37 DLF

- **Name: Divisor Latch Fraction Register (DLF)**
- **Size: 32 bits**
- **Address Offset: 0xC0**
- **Read/write access: read/write**



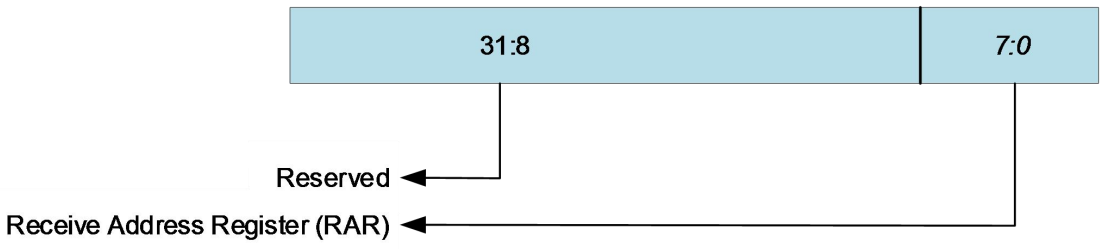
Bits	Name	R/W	Description
31:DLF_SIZE	Reserved and read as zero		
DLF_SIZE-1:0	DLF	R/W	Fractional part of divisor. The fractional value is added to integer value set by DLH, DLL. Fractional value is determined by (Divisor Fraction value)/(2^DLF_SIZE). **Reset Value**: 0x0

Divisor Latch Fractional Values

DLF Value	Fraction	Fractional Value
0000	0/16	0.0000
0001	1/16	0.0625
0010	2/16	0.125
0011	3/16	0.1875
0100	4/16	0.25
0101	5/16	0.3125
0110	6/16	0.375
0111	7/16	0.4375
1000	8/16	0.5
1001	9/16	0.5625
1010	10/16	0.625
1011	11/16	0.6875
1100	12/16	0.75
1101	13/16	0.8125
1110	14/16	0.875
1111	15/16	0.9375

8.2.38 RAR

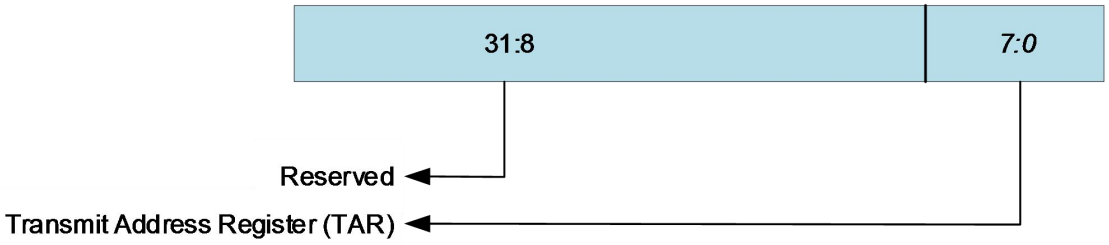
- **Name: Receive Address Register (RAR)**
- **Size: 32 bits**
- **Address Offset: 0xC4**
- **Read/write access: read/write**



Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:0	RAR	R/W	This is an address matching register during receive mode. If the 9-th bit is set in the incoming character then the remaining 8-bits will be checked against this register value. If the match happens then sub-sequent characters with 9-th bit set to 0 will be treated as data byte until the next address byte is received. **Note**: ▪ This register is applicable only when 'ADDR_MATCH' (LCR_EXT[1]) and 'DLS_E' (LCR_EXT[0]) bits are set to 1. ▪ If UART_16550_COMPATIBLE is configured to 0, then RAR should be programmed only when UART is not busy. ▪ If UART_16550_COMPATIBLE is configured to 0, then RAR can be programmed at any point of the time. However, user must not change this register value when any receive is in progress. **Reset Value**: 0x0

8.2.39 TAR

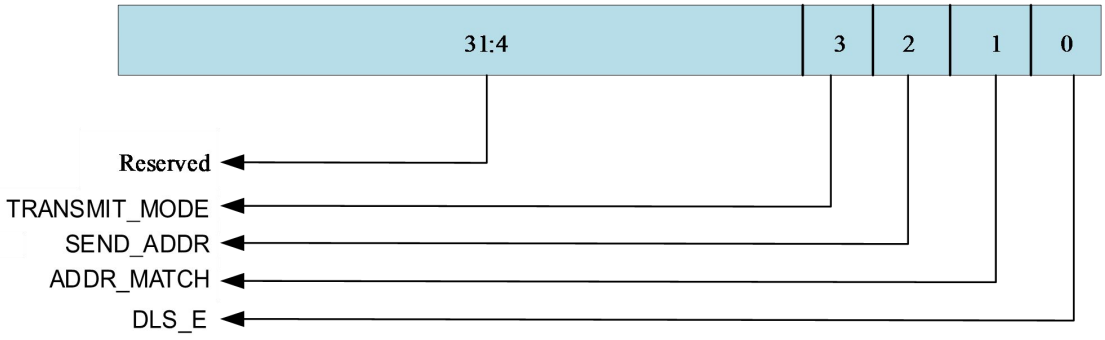
- **Name: Transmit Address Register (TAR)**
- **Size: 32 bits**
- **Address Offset: 0xC8**
- **Read/write access: read/write**



Bits	Name	R/W	Description
31:8	Reserved and read as 0		
7:0	TAR	R/W	This is an address matching register during transmit mode. If DLS_E (LCR_EXT[0]) bit is enabled, then apb_uart sends the 9-bit character with 9-th bit set to 1 and remaining 8-bit address will be sent from this register provided 'SEND_ADDR' (LCR_EXT[2]) bit is set to 1. **NOTE**: ▪ This register is used only to send the address. The normal data should be sent by programming THR register. ▪ Once the address is started to send on the apb_uart serial lane, then 'SEND_ADDR' bit will be auto-cleared by the hardware. **Reset Value**: 0x0

8.2.40 LCR_EXT

1. **Name:** Line Extended Control Register (LCR_EXT)
2. **Size:** 32 bits
3. **Address Offset: 0xCC**
4. **Read/write access:** read/write



Bits	Name	R/W	Description
31:4	Reserved and read as 0		
3	TRANSMIT_MODE	R/W	Transmit mode control bit. This bit is used to control the type of transmit mode during 9-bit data transfers. ▪ 1 : In this mode of operation, Transmit Holding Register (THR) and Shadow Transmit Holding Register (STHR) are 9-bit wide. You must ensure that the THR/STHR register is written correctly for address/data. Address: 9th bit is set to 1, Data: 9th bit is set to 0. **NOTE**: Transmit address register (TAR) is not applicable in this mode of operation. ▪ 0 : In this mode of operation, Transmit Holding Register (THR) and Shadow Transmit Holding register (STHR) are 8-bit wide. The user needs to program the address into Transmit Address Register (TAR) and data into the THR/STHR register. SEND_ADDR bit is used as a control knob to indicate the apb_uart on when to send the address. **Reset Value**: 0x0
2	SEND_ADDR	R/W	Send address control bit. This bit is used as a control knob for the user to determine when to send the address during transmit mode. ▪ 1 - 9-bit character will be transmitted with 9-th bit set to 1 and the remaining 8- bits will match to what is being programmed in "Transmit Address Register". ▪ 0 - 9-bit character will be transmitted with 9-th bit set to 0 and the remaining 8- bits will be taken from the TxFIFO which is programmed through 8-bit wide THR/STHR register. **NOTE**: 1. This bit is auto-cleared by the hardware, after sending out the address character. User is not expected to program this bit to 0. 2. This field is applicable only when DLS_E bit is set to 1 and TRANSMIT_MODE is set to 0. **Reset Value**: 0x0

Bits	Name	R/W	Description
1	ADDR_MATCH	R/W	Address Match Mode. This bit is used to enable the address match feature during receive. ▪ 1 - Address match mode; apb_uart will wait until the incoming character with 9-th bit set to 1. And, further checks to see if the address matches with what is programmed in "Receive Address Match Register". If match is found, then sub-sequent characters will be treated as valid data and apb_uart starts receiving data. ▪ 0 - Normal mode; apb_uart will start to receive the data and 9-bit character will be formed and written into the receive RxFIFO. User is responsible to read the data and differentiate b/n address and data. **NOTE**: This field is applicable only when DLS_E is set to 1.
0	DLS_E	R/W	Extension for DLS. This bit is used to enable 9-bit data for transmit and receive transfers. ▪ 1 = 9 bits per character ▪ 0 = Number of data bits selected by DLS **Reset Value**: 0x0

9.Watch Dog

9.1WDT Feature

16-bit down counter

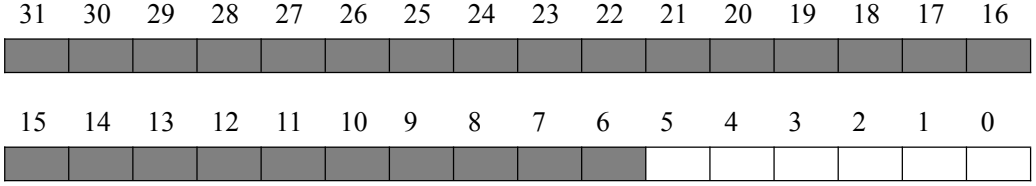
9.2WDT Register Descriptions

WDT_REG_BASE(CPU): 0x50C0_0000
WDT_REG_BASE(MCU): 0x5510_0000

9.2.1WDTCR: WDT Control Register

WDTCR is a control register and contains five fields that control the various function of WDT.

Address: WDT_REG_BASE + 0x00

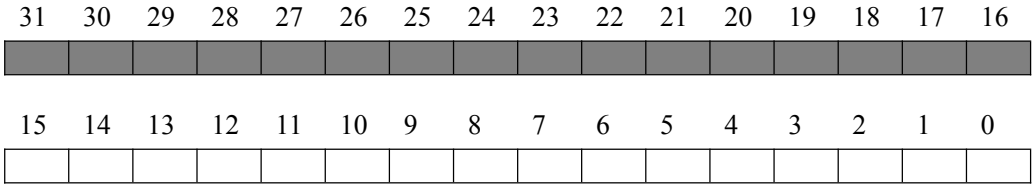


Bit	Type	Reset	Description
31-6	R	0	Not used.
5-4	R/W	2'b11	**DIV** Clock division factor selection: D[5] D[4] Functions 0 0 16 divison. 0 1 32 division. 1 0 64 division. 1 1 128 division.
3	R/W	0	**CLKSEL** This bit selects the down counting clock of WDT: 0: Prescaled and divided clock is used for down counting. 1: Prescaled clock is used for down counting.
2	R/W	0	**INTEN** Interrupt enable: This bit affects the generation of interrupt generation only when WDT is enabled (WDTEN bit is HIGH). 0: Time-out condition does not generate the **WDTINTR** interrupt. 1: Time-out condition generates the **WDTINTR** interrupt.
1	R/W	0	**RSTEN** Reset output enable. This bit controls the reset output signal nRESETOUT of WDT. This bit affects the generation of reset output only when WDT is enabled (WDTEN bit is HIGH). 0: Reset output function is disabled. 1: Reset output function is enabled.
0	R/W	0	**WDTEN** WDT enable: 0: Disable WDT (default). 1: Enable WDT.

9.2.2WDTPSR: WDT Prescale Register

WDTPSR is a prescale register and used to specify the prescale factor of 16-bit down counter in WDT.

Address: WDT_REG_BASE + 0x04



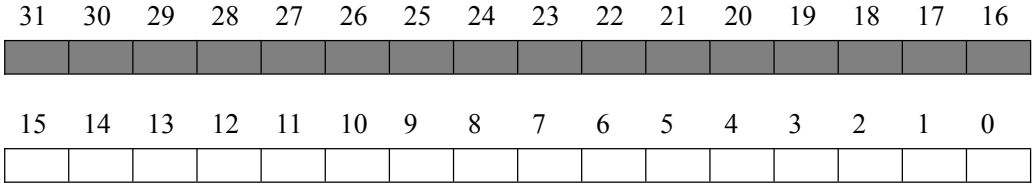
Bit	Type	Reset	Description
31-16	R	0	Not used.
15-0	R/W	0	**Prescaler value** Minimum value: 0x0000(PCLK clock is used for counting. Maximum value: 0xFFFF (Input PCLK clock prescaled by (2^16-1) is used for counting (default).

9.2.3WDTLDR: WDT Load Register

WDTLDR is a load register and

used to specify the time-out duration.

Address: WDT_REG_BASE + 0x08

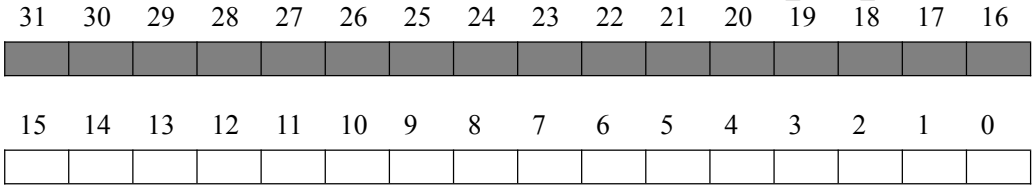


Bit	Type	Reset	Description
31-16	R	0	Not used.
15-0	R/W	0	**Load Value** The data register is a 16-bit read/write register that is programmed with the number of PCLK ticks.

9.2.4WDTVLR: WDT Value Register

WDTVLR is a read-only value register. This register is used to read the current value of 16-bit down counter in WDT.

Address: WDT_REG_BASE + 0x0c

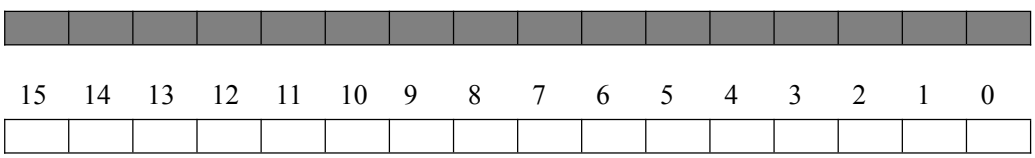


Bit	Type	Reset	Description
31-16	R	0	Not used.
15-0	R	0	**Read Value** The counter register contains the current value of WDT counter.

9.2.5WDTISR: WDT Interrupt Status Register

WDTISR is interrupt status register. It contains interrupt status information for the WDT.

Address: WDT_REG_BASE + 0x10



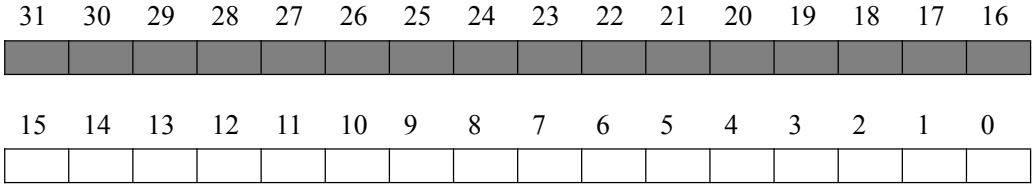
Bit	Type	Reset	Description
31-16	R	0	Not used.
15-0	R/W	0	Write operation:

			A write to this register clears the WDT interrupt status register, regardless of the value written. Read operation: This bit is set to 1 if the WDTINTR interrupt is asserted.(**Note**: Only WDTISR[0]).

9.2.6WDTRCR: WDT Reset Value Register

The reset cycle register of WDT.

Address: WDT_REG_BASE + 0x14

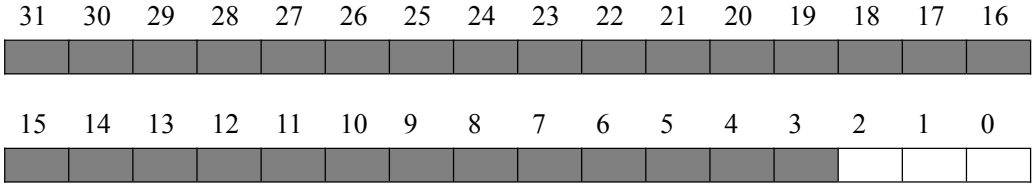


Bit	Type	Reset	Description
31-16	R	0	Not used.
15-0	R/W	0	**Reset Counter Value** The reset signal continues how may cycles.

9.2.7WDTTMR: WDT Test Mode Register

The test mode register of WDT. It contains three fields that control the operation in the test mode.

Address: WDT_REG_BASE + 0x18

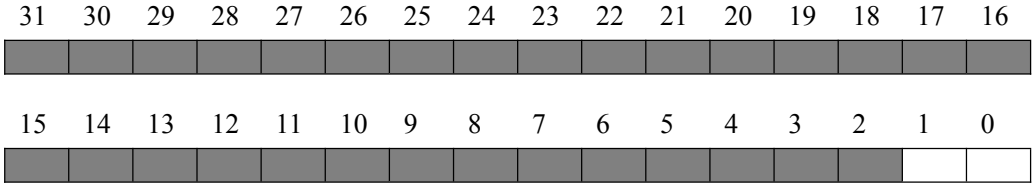


Bit	Type	Reset	Description
31-3	R	0	Not used.
2	R/W	0	**TRSET** Test reset. By default, this bit as cleared to 0 for normal operation when reset by presentn. Then this bit is set to 1, a reset is asserted throughout the module except for the test Register.
1	R/W	0	**TCKEN** Test clock enable. This bit selects the source of the test clock: 0: The internal clock enable is continuously HIGH. 1: The internal test clock enable is selected, so that the test clocks are enabled for only one period of the input clock per APB access.
0	R/W	0	**TM** Test mode select: 0: Normal operating mode is selected. 1: Test mode is selected.

9.2.8WDTTCR: WDT Capture Register

The capture Register of WDT. It captures the output signals (WDTINTR and nRESETOUT) during the test mode.

Address: WDT_REG_BASE + 0x1c



Bit	Type	Reset	Description
31-2	R	0	Not used.
1	R/W	0	**nRESETOUT** This field holds the current level of nRESETOUT output signal.
0	R/W	0	**WDTINTR** This field holds the current level of WDTINTR output signal.

10.CANFD

CANFD0_REG_BASE: 0x55600000
CANFD1_REG_BASE: 0x55700000

10.1 Features

- Free programmable data rate:
  - CAN 2.0B defines data rates up to 1Mbit/s
  - CAN FD is limited by the transceiver and the clock frequency of the can_clk.
- Programmable baud rate prescaler
- 16 acceptance filters
- Configurable interrupt sources

10.2 Register Descriptions

Register Map

	Bit position								Register
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**	
0x00 to 0x4f	Receive Buffer Register and Reception Time Stamp								RBUF (and RTS)

0x50 to 0x97	Transmit Buffer Register								TBUF
0x98 to 0x9	Transmission Time Stamp								TTS
0xa0	RESET	LBME	LBMI	TPSS	TSSS	RACTIV	TACTIVE	BUSOFF	CFG_STAT
0xa1	TBSEL	LOM	STBY	TPE	TP	TSONE	TSALL	TS	TCMD
0xa2	FD_ISO *)	TSNEXT	TSMOD	TTTBM	-		TSSTAT(1:0)		TCTRL
0xa3	SACK	ROM	ROV	RREL	RBALL	-	RSTAT(1:0)		RCTRL
0xa4	RI	ROIE	RFIE	RAFIE	TPIE	TSIE	EI	TSFF	RTIE
0xa5	RI	ROIF	RFIF	RAFIF	TPIF	TSIF	EIF	AI	RTIF
0xa6	EWARN	EPASS	EPIE	EPIF	ALIE	ALIF	BEIE	BEIF	ERRINT
0xC	AFWL(3:0)				EWL(3:0)				LIMIT
0xa8	S_Seg_1(7:0)								S_Seg_1 *)
0xa9	-	S_Seg							S_Seg_2 *)
0xaa	-	S_SJ							S_SJW *)
0xab	S_PRESC(7:0)								S_PRESC *)
0xac	-			F_Seg_1(4:0)					F_Seg_1 *)
0xad	-			F_Seg_2(					F_Seg_2 *)
0xae	-			F_SJW(3:					F_SJW *)
0xaf	F_PRESC(7:0)								F_PRESC *)
0xb0	KOER(2:0)			ALC(4:0)					EALCAP
0xb1	TDCEN	SSPOFF(6:0)							TDC *)
0xb2	RECNT								RECNT
0xb3	TECNT								TECNT
0xb4	-		SELMAS	-	ACFADR				ACFCTRL
0xb5	-						TIMEPOS	TIMEEN	TIMECFG
0xb6	AE_7	AE_6	AE_5	AE_4	AE_3	AE_2	AE_1	AE_0	ACF_EN_0
0xb7	AE_15	AE_14	AE_13	AE_12	AE_11	AE_10	AE_9	AE_8	ACF_EN_1
0xb8	ACODE_x or AMASK_x								ACF_0 *)
0xb9	ACODE_x or AMASK_x								ACF_1 *)
0xba	ACODE_x or AMASK_x								ACF_2 *)
0xbb	-	AIDEE	AIDE	ACODE_x or AMASK_x (28:24)					ACF_3 *)
0xbc	VERSION(7:0)								VER_0
0xbd	VERSION(15:8)								VER_1
0xbe	TB	TBF	TBPT						TBSLOT
0xbf	WTIE	WTIF	TEIF	TTIE	TTIF	T_PRESC(1:0)		TTEN	TTCFG
0xc0	REF_ID(7:0)								REF_MSG_
0xc1	REF_ID(15:8)								REF_MSG_
0xc2	REF_ID(23:16)								REF_MSG_
0xc3	REF_IDE	-	REF_ID(2						REF_MSG_

	Bit		Register name
0xc4	-	TTPTR(5:0)	TRIG_CFG_0

0xc5	TEW			-	TTYP			TRIG_CFG_1
0xc6	T							TT_TRIG_0
0xc7	T							TT_TRIG_1
0xc8	T							TT_WTRIG_0
0xc9	T							TT_WTRIG_1
0xca	-		MAEIF	MDEIF	MDWIF	MDWIE	MPEN	MEM_PROT
0xcb	-		HELOC(1:0)		T	T	ACFA	MEM_STAT
0xcc	MEAEE	ME1EE	M					MEM_ES_0
0xcd	-	ME2EE	M					MEM_ES_1
0xce	MEN		MEEEC(3:0)					MEM_ES_2
0xcf	-				MES	MEL(1:0)		MEM_ES_3
0xd0	-	FSTIM(2:0)		SWIF	SWIE	SEIF	XMRE	SCFG

A write access to an addressable location not shown in the register map results in no action and a read access will result in the value 0x00.

To reset an interrupt flag, the host controller needs to write a 1 to the flag. Writing a 0 has no effect. If a new interrupt event occurs while the write access is active then this event will set the flag and override the reset. This ensures that no interrupt event is lost.

Interrupt flags will only be set if the associated interrupt enable bit is set.

Configuration and Status Register CFG_STAT (0xa0)

Bits	Name	Access	Function
7	RESET	rw-1	RESET request bit 1 -The host controller performs a local reset of CAN-CTRL. 0 -no local reset of CAN-CTRL The some register (e.g for node configuration) can only be modified if RESET=1. Bit RESET forces several components to a reset state. RESET is automatically set if the node enters “bus off” state . Note that a CAN node will participate in CAN communication after RESET is switched to 0 after 11 CAN bit times. This delay is required by the CAN standard (bus idle time). If RESET is set to 1 and immediately set to 0, then it takes some time until RESET can be read as 0 and becomes inactive. The reason is clock domain crossing from host to CAN clock domain. RESET is held active as long as needed depending on the relation between host and CAN clock.
6	LBME	rw-0	Loop Back Mode, External 0 -Disabled 1 -Enabled LBME should not be enabled while a transmission is active.
5	LBMI	rw-0	Loop Back Mode, Internal 0 -Disabled 1 -Enabled LBMI should not be enabled while a transmission is active.

4	TPSS	rw-0	Transmission Primary Single Shot mode for PTB 0 -Disabled 1 -Enabled
3	TSSS	rw-0	Transmission Secondary Single Shot mode for STB 0 -Disabled 1 -Enabled
2	RACTIVE	r-0	Reception ACTIVE (Receive Status bit) 1 -The controller is currently receiving a frame. 0 -No receive activity.
1	TACTIVE	r-0	Transmission ACTIVE (Transmit Status bit) 1 -The controller is currently transmitting a frame. 0 -No transmit activity.
0	BUSOFF	rw-0	Bus Off (Bus Status bit) 1 -The controller status is “bus off”. 0 -The controller status is “bus on”. Writing a 1 to BUSOFF will reset TECNT and RECNT. This should be done only for debugging.

Command Register TCMD (0xa1)

Bits	Name	Access	Function
7	TBSEL	rw-0	Transmit Buffer Select Selects the transmit buffer to be loaded with a message. Use the TBUF Register for access. TBSEL needs to be stable all the time the TBUF Register are written and when TSNEXT is set. 0 -PTB (high- priority buffer) 1 -STB The bit will be reset to the hardware reset value if (TTEN=1 and TTTBM=1).
6	LOM	rw-0	Listen Only Mode 0 -Disabled 1 -Enabled LOM cannot be set if TPE, TSONE or TSALL is set. No transmission can be started if LOM is enabled and LBME is disabled. LOM=1 and LBME=0 disables all transmissions. LOM=1 and LBME=1 disables the ACK for received frames and error frames, but enables the transmission of own frames.
5	STBY	rw-0	Transceiver Standby Mode 0 -Disabled 1 -Enabled This register bit is connected to the output signal stby which can be used to control a standby mode of a transceiver.

Bits	Name	Access	Function

			STBY cannot be set to 1 if TPE=1, TSONE=1 or TSALL=1. If the host sets STBY to 0 then the host needs to wait for the time required by the transceiver to start up before the host requests a new transmission.
4	TPE	rw-0	Transmit Primary Enable 1 -Transmission enable for the message in the high-priority PTB 0 -No transmission for the PTB If TPE is set, the message from the PTB will be transmitted at the next possible transmit position. A started transmission from the STB will be completed before, but pending new messages are delayed until the PTB message has been transmitted. TPE stays set until the message has been transmitted successfully or it is aborted using TPA. The host controller can set TPE to 1 but can not reset it to 0. This would only be possible using TPA and aborting the message. The bit will be reset to the hardware reset value if RESET=1, STBY=1, (LOM=1 and LBME=0) or (TTEN=1 and TTTBM=1).
3	TPA	rw-0	Transmit Primary Abort 1 –Aborts a transmission from PTB which has been requested by TPE=1 but not started yet. (The data bytes of the message remains in the PTB.) 0 –no abort The bit has to be set by the host controller and will be reset by CAN-CTRL. Setting TPA automatically de-asserts TPE. The bit will be reset to the hardware reset value if RESET=1 or (TTEN=1 and TTTBM=1). TPA should not be set simultaneously with TPE.
2	TSONE	rw-0	Transmit Secondary ONE frame 1 –Transmission enable of one in the STB. In FIFO mode this is the oldest message and in priority mode this is the one with the highest priority. TSONE in priority mode is difficult to handle, because it is not always clear which message will be transmitted if new messages are written to the STB meanwhile. The controller starts the transmission as soon as the bus becomes vacant and no request of the PTB (bit TPE) is pending. 0 –No transmission for the STB. TSONE stays set until the message has been transmitted successfully or it is aborted using TSA. The host controller can set TSONE to 1 but can not reset it to 0. This would only be possible using TSA and aborting the message. The bit will be reset to the hardware reset value if RESET=1, STBY=1, (LOM=1 and LBME=0) or (TTEN=1 and TTTBM=1).

1	TSALL	rw-0	Transmit Secondary ALL frames 1 –Transmission enable of all messages in the STB. The controller starts the transmission as soon as the bus becomes vacant and no request of the PTB (bit TPE) is pending. 0 –No transmission for the STB. TSALL stays set until all messages have been transmitted successfully or they are aborted using TSA. The host controller can set TSALL to 1 but can not reset it to 0. This would only be possible using TSA and aborting the messages. The bit will be reset to the hardware reset value if RESET=1, STBY=1, (LOM=1 and LBME=0) or (TTEN=1 and TTTBM=1). If during a transmission the STB is loaded with a new frame then the new frame will be transmitted too. In other words: a transmission initiated by TSALL is finished when the STB becomes empty.
0	TSA	rw-0	Transmit Secondary Abort 1 –Aborts a transmission from STB which has been requested but not started yet. For a TSONE transmission, only one frame is aborted while for a TSALL Transmission, all frames are aborted. One or all message slots will be released which updates TSSTAT. All aborted messages are lost because they are not accessible any more. If in priority mode a TSONE transmission is aborted, then it is not clear which frame will be aborted if new frames are written to the STB meanwhile. 0 –no abort The bit has to be set by the host controller and will be reset by CAN-CTRL. Setting TSA,

Bits	Name	Access	Function
			automatically de-asserts TSONE or TSALL respectively. The host controller can set TSA to 1 but can not reset it to 0. The bit will be reset to the hardware reset value if RESET=1. TSA should not be set simultaneously with TSONE or TSALL.

Setting both TSONE and TSALL is meaningless. While TSALL is already set, it is impossible to set TSONE and vice versa. If both TSONE and TSALL are set simultaneously then TSALL wins and TSONE is cleared by the CAN-CTRL core.

Transmit Control Register TCTRL (0xa2)

Bits	Name	Access	Function

7	FD_ISO	rw-1	CAN FD ISO mode 0 -Bosch CAN FD (non-ISO) mode 1 -ISO CAN FD mode (ISO 11898-1:2015) ISO CAN FD mode has a different CRC initialization value and an additional stuff bit count. Both modes are incompatible and must not be mixed in one CAN network. This bit has no impact to CAN 2.0B. This bit is only writeable if RESET=1.
6	TSNEXT	rw-0	reserved
5	TSMODE	rw-0	Transmit buffer Secondary operation MODE 0 -FIFO mode 1 - priority decision mode In FIFO mode frames are transmitted in the order in that they are written into the STB. In priority decision mode the frame with the highest priority in the STB is automatically transmitted first. The ID of a frame is used for the priority decision. A lower ID means a higher priority of a frame. A frame in the PTB has always the highest priority regardless of the ID. TSMODE shall be switched only if the STB if empty.
4	TTTBM	rw-1	reserved

3:2	-	r-0	reserved
1:0	TSSTAT	r-0	Transmission Secondary STATus bits If TTEN=0 or TTTBM=0: 1 –STB is empty 2 –STB is less than or equal to half full 10 –STB is more than half full

Bits	Name	Access	Function
			11 –STB is full If the STB is disabled using STB_DISABLE, then TSSTAT=00. If TTEN=1 and TTTBM=1: 1 –PTB and STB are empty 2 –PTB and STB are not empty and not full 11 –PTB and STB are full

Receive Control Register RCTRL (0xa3)

Bits	Name	Access	Function
7	SACK	rw-0	Self-ACKnowledge 0 –no self-ACK 1 –self-ACK when LBME=1
6	ROM	rw-0	Receive buffer Overflow Mode In case of a full RBUF when a new message is received, then ROM selects the following: 1 –The new message will not be stored. 0 –The oldest message will be overwritten.
5	ROV	r-0	Receive buffer OVerflow 1 –Overflow. At least one message is lost. 0 –No Overflow. ROV is cleared by setting RREL=1.
4	RREL	rw-0	Receive buffer RELease The host controller has read the actual RB slot and releases it. Afterwards the CAN-CTRL core points to the next RB slot. RSTAT gets updated. 1 –Release: The host has read the RB. 0 –No release
3	RBALL	rw-0	Receive Buffer stores ALL data frames 0 –normal operation 1 –RB stores correct data frames as well as data frames with error
2	-	r-0	Reserved
1:0	RSTAT	r-0	Receiv e buffer STATus 00 -empty 01 -> empty and < almost full (AFWL) 10 -almost full (programmable threshold by AFWL) but not full and no overflow 11 -full (stays set in case of overflow – for overflow signaling see ROV)

Receive and Transmit Interrupt Enable Register RTIE (0xa4)

Bits	Name	Access	Function
7	RIE	rw-1	Receive Interrupt Enable 0 – Disabled, 1 – Enabled
6	ROIE	rw-1	RB Overrun Interrupt Enable 0 – Disabled, 1 – Enabled
5	RFIE	rw-1	RB Full Interrupt Enable 0 – Disabled, 1 – Enabled
4	RAFIE	rw-1	RB Almost Full Interrupt Enable 0 – Disabled, 1 – Enabled
3	TPIE	rw-1	Transmission Primary Interrupt Enable 0 – Disabled, 1 – Enabled
2	TSIE	rw-1	Transmission Secondary Interrupt Enable 0 – Disabled, 1 – Enabled
1	EIE	rw-1	Error Interrupt Enable 0 – Disabled, 1 – Enabled
0	TSFF	r-0	If TTEN=0 or TTTBM=0: Transmit Secondary buffer Full Flag 1 -The STB is filled with the maximal number of messages. 0 -The STB is not filled with the maximal number of messages If the STB is disabled using STB_DISABLE, then TSFF=0. If TTEN=1 and TTTBM=1: Transmit buffer Slot Full Flag 1 -The buffer slot selected by TBPTR is filled. 0 -The buffer slot selected by TBPTR is empty.

Receive and Transmit Interrupt Flag Register RTIF (0xa5)

Bits	Name	Access	Function
7	RIF	rw-0	Receive Interrupt Flag 1 -Data or a remote frame has been received and is available in the receive buffer. 0 -No frame has been received.

6	ROIF	rw-0	RB Overrun Interrupt Flag 1 -At least one received message has been overwritten in the RB. 0 -No RB overwritten. In case of an overrun both ROIF and RFIF will be set.
5	RFIF	rw-0	RB Full Interrupt Flag 1 -All RBs are full. If no RB will be released until the next valid message is received, the oldest message will be lost. 0 -The RB FIFO is not full.
4	RAFIF	rw-0	RB Almost Full Interrupt Flag 1 -number of filled RB slots AFWL_i 0-number of filled RB slots < AFWL_i
3	TPIF	rw-0	Transmission Primary Interrupt Flag 1 -The requested transmission of the PTB has been successfully completed. 0 -No transmission of the PTB has been completed.
2	TSIF	rw-0	Transmission Secondary Interrupt Flag 1 -The requested transmission of the STB has been successfully completed. 0 -No transmission of the STB has been completed successfully.
1	EIF	rw-0	Error Interrupt Flag 1 -The border of the error warning limit has been crossed in either direction, or the BUSOFF bit has been changed in either direction. 0 -There has been no change.
0	AIF	rw-0	Abort Interrupt Flag 1 -After setting TPA or TSA the appropriated message(s) have been aborted. It is recommended to not set both TPA and TSA simultaneously because both source AIF. 0 -No abort has been executed. The AIF does not have an associated enable register.

ERRor INTerrupt Enable and Flag Register ERRINT (0xa6)

Bits	Name	Access	Function
7	EWARN	r-0	Error WARNing limit reached 1 -One of the error counters RECNT or TECNT is equal or bigger than EWL 0 -The values in both counters are less than EWL.
6	EPASS	r-0	Error Passive mode active 0 -not active (node is error active) 1 -active (node is error passive)
5	EPIE	rw-0	Error Passive Interrupt Enable

4	EPIF	rw-0	Error Passive Interrupt Flag. EPIF will be activated if the error status changes from error active to error passive or vice versa and if this interrupt is enabled.
3	ALIE	rw-0	Arbitration Lost Interrupt Enable
2	ALIF	rw-0	Arbitration Lost Interrupt Flag
1	BEIE	rw-0	Bus Error Interrupt Enable
0	BEIF	rw-0	Bus Error Interrupt Flag

Warning Limits Register LIMIT (0xCPU)

Bits	Name	Access	Function
7:4	AFWL (3:0)	rw-0x1	receive buffer Almost Full Warning Limit AFWL defines the internal warning limit AFWL_i with n_{RB} being the number of available RB slots. AFWL_i is compared to the number of filled RB slots and triggers RAFIF if equal. The valid range of AFWL_i 1 n_{RB} . AFWL 0 ismeaninglessandautomaticallytreatedas0x1. (Note that AFWL is meant in this rule and not AFWL_i.) AFWL_i n_{RB} is meaningless and automatically treated as n_{RB} . AFWL_i n_{RB} is a valid value, but note that RFIF also exists.
3:0	EWL (3:0)	rw-0xB	Programmable Error Warning Limit = (EWL+1)*8. Possible Limit values: 8, 16, … 128. The value of EWL controls EIF. EWL needs to be transferred using CDC from host to CAN clock domain. During transfer EWL register bits are write-locked for the host for a few clocks until CDC is complete.

Bit Timing Register S_Seg_1 (0xa8)

Bits	Name	Access	Function
7:0	S_Seg_1(7:0)	rw-0x3	Bit Timing Segment 1 (slow speed) The sample point will be set to t_{Seg_1} Seg_1 2 TQ after start of bit time.

Bit Timing Register S_Seg_2 (0xa9)

Bits	Name	Access	Function
7	-	r-0	reserved
6:0	S_Seg_2(6:0)	rw-0x2	Bit Timing Segment 2 (slow speed) Time t_{Seg_2} Seg_2 1 TQ after the

Bit Timing Register S_SJW (0xaa)

Bits	Name	Access	Function
7	-	r-0	reserved
6:0	S_SJW(6:0)	rw-0x2	Synchronization Jump Width (slow speed) The Synchronization Jump Width t_{SJW} SJW 1 TQ is the maximum time for shortening or lengthening the Bit Time for resynchronization, where TQ is a time quanta.

Bit Timing Register F_Seg_1 (0xac)

Bits	Name	Access	Function
7:5	-	r-0	reserved
4:0	F_Seg_1(4:0)	rw-0x3	Bit Timing Segment 1 (fast speed) The sample point will be set to t_{Seg_1} Seg_1 2 TQ after start of bit time.

Bit Timing Register F_Seg_2 (0xad)

Bits	Name	Access	Function
7:4	-	r-0	reserved
3:0	F_Seg_2(3:0)	rw-0x2	Bit Timing Segment 2 (fast speed) Time t_{Seg_2} Seg_2 1 TQ after the sample point

Bit Timing Register F_SJW (0xae)

Bits	Name	Access	Function
7:4	-	r-0	reserved

3:0	F_SJW(3:0)	rw-02	Synchronization Jump Width (fast speed) The Synchronization Jump Width t_{SJW} SJW 1 TQ is the maximum time for shortening or lengthening the Bit Time for resynchronization, where TQ is a time quanta.

Prescaler Register S_PRESC (0xab) and F_PRESC (0xaf)

Bits	Name	Access	Function
7:0	S_PRESC F_PRESC	rw-0x01	Prescaler (slow and fast speed) The prescaler divides the system clock to get the time quanta clock tq_clk. Valid range PRESC=[0x00, 0xff] results in divider values 1 to 256.

Transmitter Delay Compensation Register TDC (0xb1)

Bits	Name	Access	Function
7	TDCEN	rw-0	Transmitter Delay Compensation ENable TDC will be activated during the data phase of a CAN FD frame if BRS is active if TDCEN=1.
6:0	SSPOFF	rw-0x00	Secondary Sample Point OFFset The transmitter delay plus SSPOFF defines the time of the secondary sample point for TDC. SSPOFF is given as a number of TQ.

Writing to S_Seg_1, S_Seg_2, S_SJW, S_PRESC, F_Seg_1, F_Seg_2, F_SJW, F_PRESC and TDC is only possible if RESET=1.

All timing parameters in are given for slow (prefix “S_”) and fast speed (prefix “F_”). Slow speed is used for CAN 2.0 and the CAN FD arbitration phase. Fast speed is used for the CAN FD data phase.

Error and Arbitration Lost Capture Register EALCAP (0xb0)

Bits	Name	Reset Value	Function

7:5	KOER(2:0)	r-0x0	Kind Of ERror (Error code) 000 -no error 001 -BIT ERROR 10 -FORM ERROR 11 -STUFF ERROR 100 -ACKNOWLEDGEMENT ERROR 101 -CRC ERROR 110 -OTHER ERROR (dominant bits after own error flag, received active Error Flag too long, dominant bit during Passive-Error-Flag after ACK error) 111 -not used KOER is updated with each new error. Therefore it stays untouched when frames are successfully transmitted or received.
4:0	ALC(4:0)	r-0x0	Arbitration Lost Capture (bit position in the frame where the arbitration has been lost)

Error Counter Register RECNT (0xb2) and TECNT (0xb3)

Bits	Name	Access	Function
7:0	RECNT	r-0x00	Receive Error CouNT (number of errors during reception) RECNT is incremented and decremented as defined in the CAN specification. RECNT does not overflow. If TXB=1, then the error counters are frozen.
7:0	TECNT	r-0x00	Transmit Error CouNT (number of errors during transmission) TECNT is incremented and decremented as defined in the CAN specification. In case of the “bus off state” TECNT may overflow. If TXB=1, then the error counters are frozen.

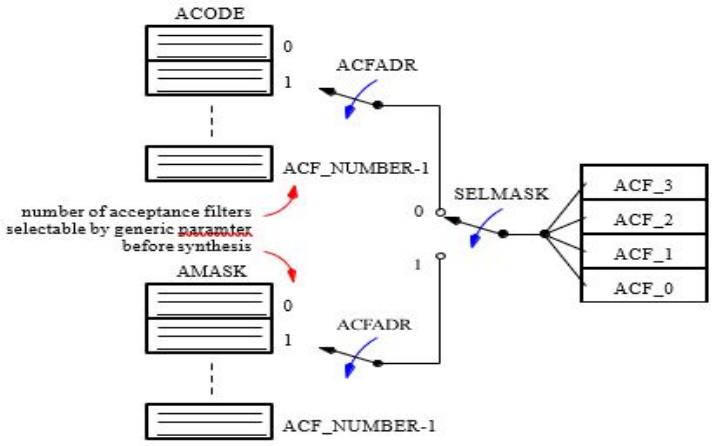
Acceptance Filter Control Register ACFCTRL (0xb4)

Bits	Name	Access	Function
7:6	-	r-0	reserved
5	SELMASK	rw-0	SELect acceptance MASK 0 -Register ACF_x point to acceptance code 1 -Register ACF_x point to acceptance mask. ACFADR selects one specific acceptance filter.
4	-	r-0	Reserved

3:0	ACFADR	rw-0	acceptance filter address ACFADR points to a specific acceptance filter. The selected filter is accessible using the Register ACF_x. Bit SELMASK selects between acceptance code and mask for the selected acceptance filter. A value of ACFADR>ACF_NUMBER-1 is meaningless and automatically treated as value ACF_NUMBER-1.

The acceptance filter Register ACF_x provide access to the acceptance filter codes ACODE_x and acceptance filter masks AMASK_x depending on the setting of SELMASK. Write access to ACF_x is only possible if RESET=1. If the pseudo dual-port memories are selected , then read access is also only possible if RESET=1.

The acceptance filters are build using a true 32 bit wide memory and therefore a write access needs to be performed as 32 bit write.



Access to the Acceptance Filters

Acceptance CODE ACODE_x (register ACF_x (0xb8 to 0xbb))

Bits	Name	Access	Function
7:0	ACODE_0 ACODE_x	rw-0x00 rw-u	Acceptance CODE 1 - ACC bit value to compare with ID bit of the received message 0 - ACC bit value to compare with ID bit of the received message ACODE_x(10:0) will be used for extended frames. ACODE_x(28:0) will be used for extended frames. Only filter 0 is affected by the power-on reset. All other filters stay uninitialized.

T Acceptance MASK AMASK_x (register ACF_x (0xb8 to 0xbb))

Bits	Name	Access	Function

7:0	AMASK_0 AMASK_x	rw-0xFF rw-u	Acceptance MASK 1 -acceptance check for these bits of receive identifier disabled 0 -acceptance check for these bits of receive identifier enable AMASK_x(10:0) will be used for extended frames. AMASK_x(28:0) will be used for extended frames. Disabled bits result in accepting the message. Therefore the default configuration after reset for filter 0 accepts all messages. Only filter 0 is affected by the power-on reset. All other filters stay uninitialized.

The AMASK_x includes additional bits in register ACF_3 which can be only accessed if SELMASK=1. These bits can be used to accept only either standard or extended frames with the selected ACODE / AMASK setting or to accept both frame types. Only acceptance filter 0 is affected by the power-on reset and it is configured to accept both frame types after power-up.

Bits in Register ACF_3, if SELMASK=1

Bits	Name	Access	Function
6	AIDEE	rw-0 rw-u	Acceptance mask IDE bit check enable 1 -acceptance filter accepts either standard or extended as defined by AIDE 0 -acceptance filter accepts both standard or extended frames Only filter 0 is affected by the power-on reset. All other filters stay uninitialized.
5	AIDE	rw-0 rw-u	Acceptance mask IDE bit value If AIDEE=1 then: 1 -acceptance filter accepts only extended frames 0 -acceptance filter accepts only standard frames Only filter 0 is affected by the power-on reset. All other filters stay uninitialized.

Acceptance Filter Enable ACF_EN_0 (0xb6)

Bits	Name	Access	Function

7:0	AE_x	rw-0x01	Acceptance filter Enable 1 -acceptance filter enabled 0 -acceptance filter disable Each acceptance filter (AMASK / ACODE) can be individually enabled or disabled. Only filter number 0 is enabled by default after hardware reset. Disabled filters reject a message. Only enabled filters can accept a message if the appropriate AMASK / ACODE configuration matches. To accept all messages one filter x has to be enabled by setting AE_x=1, AMASK_x=0xff and ACODE_x=0x00. This is the default configuration after hardware reset for filter x=0 while all other filters are disabled.

Acceptance Filter Enable ACF_EN_1 (0xb7)

Bits	Name	Access	Function
7:0	AE_x	rw-0x00	Acceptance filter Enable 1 -acceptance filter enabled 0 -acceptance filter disable Each acceptance filter (AMASK / ACODE) can be individually enabled or disabled. Disabled filters reject a message. Only enabled filters can accept a message if the appropriate AMASK / ACODE configuration matches.

MEM_PROT: Memory Protection (0xca)

Bits	Name	Access	Function
0	MPEN	rw-1	Memory Protection Enable 0 – disabled 1 – enabled MPEN is only writable if RESET=1. If synthesis parameter RAM_ECC=0 then this bit is always 0.
1	MDWIE	rw-1	Memory: Data Warning Interrupt Enable 0 – disabled 1 – enabled If synthesis parameter RAM_ECC=0 then this bit is always 0.
2	MDWIF	rw-0	Memory: Data Warning Interrupt Flag MDWIF will be set if MDWIE is set and memory protection has detected an error in the data which has been automatically corrected. The system continues
3	MDEIF	rw-0	Memory: Data Error Interrupt Flag MDEIF will be set if memory protection has detected an error in the data which was not correctable.
4	MAEIF	rw-0	Memory: Address Error Interrupt Flag MAEIF will be set if memory protection has detected an error in the address. The error was not corrected.
7:5	-	r-00	reserved

MEM_STAT: Memory Status (0xcb)

Bits	Name	Access	Function

0	ACFA	rw-0	Acceptance Filter Accept 0 – normal operation of ACF 1 – ACF disabled: all received frames are accepted If MDEIF or MAEIF is set because of an error in the address range of the ACF, ACFA is set. Then acceptance filtering is disabled and all frames will be accepted. ACFA can be reset similar to an interrupt flag by writing a 1 to it. But because ACFA will be set while the reception is still active, it is required to reset if later after the reception has been completed and e.g. RIF has been set.
1	TXS	r-0	Transmission Stop 0 – normal operation 1 – transmission stopped If MDEIF or MAEIF is set because of an error while the priority reordering machine accesses the memory, then any new transmission is stopped. If there is an active transmission, this will be completed before the stop, but if an error occurs during this transmission then no retransmission will be started. TXS is reset if RESET=1.
2	TXB	r-0	Transmission Block 0 – normal operation 1 – transmission blocked If MDEIF or MAEIF is set because of an error while the CAN protocol machine is reading data for transmission, then the transmission is immediately blocked. If SEIF is set, then the transmission is immediately blocked too. If TXB=1, then the error counters are
4:3	HELOC	r-00	Host side memory error location 1 – no error during access from host side 2 – error during access from host side in TBUF 10 – error during access from host side in RBUF 11 – error during access from host side in ACF HELOC will be updated with every new error during a read access from the host side. This is sufficient, because read errors during read accesses from the CAN side will be signaled by ACFA, TXS and TXB. HELOC will only be updated in case of an error, but not in case of a warning caused by a corrected single bit error.
7:5	-	r-0	reserved

MEM_ES_0: Memory Error Stimulation 0 (0xcc)

Bits	Name	Access	Function
5:0	MEBP1	rw-00	Memory Error: Bit Position 1 MEBP1 defines the position of one single bit error during memory read.
6	ME1EE	rw-0	Memory Error: 1st Error Enable 0 – disabled 1 – enabled If ME1EE E=1 then MEBP1 is used to do stimulation of a single bit error. To stimulate a dual bit error, MEBP2 and ME2EE needs to be used too. The error is only triggered for the access defined by MEEEC, MENEC, MEL and MES.

7	MEAEE	rw-0	Memory Error: Address Error Enable 0 – disabled 1 – enabled If MEAEE=1 then the address bit 2 is flipped. The result is an access to the neighbor 32 bit word. The error is only triggered for the access defined by MEEEC, MENEC, MEL and MES.

MEM_ES_1: Memory Error Stimulation 1 (0xcd)

Bits	Name	Access	Function
5:0	MEBP2	rw-00	Memory Error: Bit Position 2 MEBP2 defines the position of one single bit error during memory read. To stimulate a dual bit error, MEBP1 and ME1EE needs to be used too.
6	ME2EE	rw-0	Memory Error: 2nd Error Enable 0 – disabled 1 – enabled If ME2EE E=1 then MEBP2 is used to do stimulation of a single bit error.
7	-	r-0	reserved

MEM_ES_2: Memory Error Stimulation 2 (0xce)

Bits	Name	Access	Function
3:0	MEEEC	rw-00	Memory Error: Error Enable Counter Error stimulation is activated if MENEC=0 and MEEEC>0. MEEEC is automatically decremented after each event of an error stimulation (defined by MEL and MES) until it reaches the value 0x0.
7:4	MENEC	rw-00	Memory Error: No Error Counter Error stimulation is delayed if MENEC>0. MENEC is automatically decremented after each memory read access (defined by MEL and MES) until it reaches the value 0x0.

TMEM_ES_3: Memory Error Stimulation 3 (0xcf)

Bits	Name	Access	Function
1:0	MEL	rw-00	Memory Error Location 00 – ACF 01– TBUF 10– RBUF 11– ACF MEL defines the memory area which will be hit by the next memory error together with MES.
2	MES	rw-0	Memory Error Side 0 – Host side 1 – CAN side MES defines the side which will be hit by the next memory error together with MEL.
7:3	-	r-0x0	reserved

SCFG: Safety Configuration (0xd0)

Bits	Name	Access	Function

0	XMREN	rw-1	DMR / TMR enable 0 – disabled 1 – enabled It depends on the synthesis parameter SAFETY_ENABLE if DMR or TMR is available. XMREN can only be modified if RESET=1.
1	SEIF	rw-0	DMR / TMR: Safety Error Interrupt Flag 0 – no interrupt 1 – interrupt pending If XMREN=1 and if there is a not correctable mismatch between the redundant instances, then SEIF and TXB will be set.
2	SWIE	rw-1	DMR / TMR: Safety Warning Interrupt Enable 0 – SWIF disabled 1 – SWIF enable
3	SWIF	rw-0	DMR / TMR: Safety warning interrupt flag 0 – no interrupt 1 – interrupt pending If XMREN=1 and if there is a correctable mismatch between the redundant instances, then SWIF will be set if SWIE=1.
6:4	FSTIM	rw-000	Fault Stimulation Fault stimulation is only possible, if XMREN=1.
7	-	r-0	reserved

Receive Buffer Register RBUF – Standard Format (r-0)

Address	Bit position								Function
	7	6	5	4	3	2	1	0	
RBUF	I								Identifier
RBUF+1	-					I			Identifier
RBUF+2	-								Identifier
RBUF+3	ESI	-							Identifier
RBUF+4	IDE=0	RTR	FDF	BR	DLC(3:0)				Control
RBUF+5	KOER			T	-				Status
RBUF+8	d								Data byte 1
RBUF+9	d								Data byte 2
.	…								.
RBUF+71	d								Data byte 64

Receive Buffer Register RBUF – Extended Format (r-0)

Address	Bit								Function
	7	6	5	4	3	2	1	0	
RBUF	ID(7								Identifier
RBUF+1	ID(15								Identifier
RBUF+2	ID(2								Identifier
RBUF+3	ESI	-		ID(28:24)					Identifier
RBUF+4	IDE=1	RTR	FDF	BRS	D				Control
RBUF+5	KOER			TX	-				Status
RBUF+8	d								Data byte 1
RBUF+9	d								Data byte 2
.	…								.

RBUF+71	d64(7:0)	Data byte 64

The RBUF Register (0x00 to 0x4f) point the message slot with the oldest received message in the RB.All RBUF Register can be read in any order.

KOER in RBUF has the same meaning as the bits KOER in register EALCAP. KOER in RBUF becomes meaningful if RBALL=1.

Status bit TX in RBUF is set to 1 if the loop back mode is activated and the core has received its own transmitted frame. This can be useful if LBME=1 and other nodes in the network do also transmissions.

Transmit Buffer Register TBUF – Standard Format (rw-u)

Address	Bit								Function
	7	6	5	4	3	2	1	0	
TBUF	I								Identifier
TBUF+1	-					I			Identifier
TBUF+2	-								Identifier
TBUF+3	TTSEN	-							Identifier
TBUF+4	IDE=0	RTR	FDF	BRS	DLC(3:0)				Control
TBUF+8	d								Data byte 1
TBUF+9	d								Data byte 2
.	…								.
TBUF+71	d								Data byte 64

Transmit Buffer Register TBUF – Extended Format (rw-u)

Address	Bit position								Function
	7	6	5	4	3	2	1	0	
TBUF	I								Identifier
TBUF+1	I								Identifier
TBUF+2	I								Identifier
TBUF+3	TTSEN	-		ID(28:24)					Identifier
TBUF+4	IDE=1	RTR	FDF	BRS	DLC(3:0)				Control
TBUF+8	d								Data byte 1
TBUF+9	d								Data byte 2
.	…								.
TBUF+71	d								Data byte 64

The TBUF Register (0x50 to 0x97) point the next empty message slot in the STB if TBSEL=1 or to the PTB otherwise. All TBUF Register can be written in any order. For the STB it is necessary to set TSNEXT to mark a slot filled and to jump to the next message slot.

Please mind the gap inside the addressing range of TBUF from TBUF+5 to TBUF+7. This is for better address segment alignment. The memory cells in the gap can be read and written, but have no meaning for the CAN protocol.

TBUF is build using a true 32 bit wide memory and therefore a write access needs to be performed as 32 bit write.

Both RBUF and TBUF include some frame-individual control bits . For RBUF these bits signal the status of the appropriate CAN control field bits of the received CAN frame while for TBUF these bits select the appropriate CAN control field bit for the frame that has to be transmitted.

In contrast to RTS, which is stored for every received frame, TTS is stored only for the last transmitted frame if TTSEN=1. TTS is not related to the actual selected TBUF slot.

Control bits in RBUF and TBUF

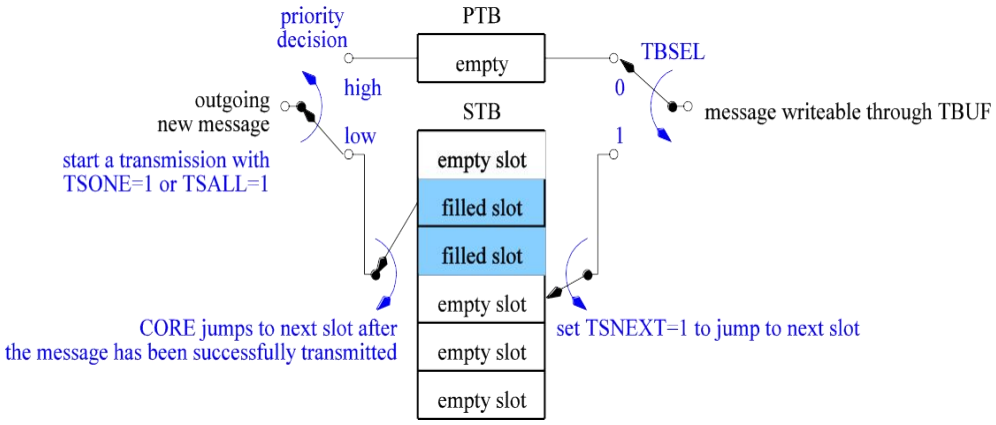
Bit	Description
IDE	IDentifier Extension 0 – Standard Format: ID(10:0) 1 – Extended Format: ID(28:0)
RTR	Remote Transmission Request 0 – data frame 1 – remote frame Only CAN 2.0 frames can be remote frames. There is no remote frame for CAN FD. Therefore RTR is forced to 0 if FDF =1 in TBUF and RBUF. If a CAN FD frame is received with bit RRS=1, then this is ignored, a data payload is expected for reception instead and RTR in RBUF is overridden but the CRC of the frame is calculated with RRS=1.
FDF	CAN FD frame 0 – CAN 2.0 frame (up to 8 bytes payload) 1 – CAN FD frame (up to 64 bytes payload)
BRS	Bit Rate Switch 0 – nominal / slow bit rate for the complete frame 1 – switch to data / fast bit rate for the data payload and the CRC Only CAN FD frames can switch the bitrate. Therefore BRS is forced to 0 if FDF =0.
ESI	Error State Indicator This is a read-only status bit for RBUF and is not available in TBUF. The protocol machine automatically embeds the correct value of ESI into transmitted frames. ESI is only included in CAN FD frames and does not exist in CAN 2.0 frames. 0 – CAN node is error active 1 – CAN node is error passive ESI in RBUF is always low for CAN 2.0 frames. The error state for transmission is shown with bit EPASS in register ERRINT.

The Data Length Code (DLC) in RBUF and TBUF defines the length of the payload – the number of payload bytes in a frame.

Remote frames (only for CAN 2.0 frames where FDF =0) are always transmitted with 0 payload bytes, but the content of the DLC is transmitted in the frame header. Therefore it is possible to code some information into the DLC bits for remote frames. But then care needs to be taken if different CAN nodes are allowed to transmit a remote frame with the same ID. In this case all transmitters need to use the same DLC because otherwise this would result in an unresolvable collision.

Definition of the DLC (according to the CAN 2.0 / FD specification)

DLC (binary)	Frame Type	Payload in Bytes
0000 to 1000	CAN 2.0 and CAN FD	0 to 8
1001 to 1111	CAN 2.0	8
1001	CAN FD	12
1010	CAN FD	16
1011	CAN FD	20
1100	CAN FD	24
1101	CAN FD	32
1110	CAN FD	48
1111	CAN FD	64



Schematic of PTB and STB in FIFO mode (empty PTB and 6 STB slots)

The TBUF Register and readable and writable. Therefore a host controller may use TBUF to successively prepare a message bit-by-bit if necessary.

11.GPIO

11.1GPIO Features

Programmable interface
General input

General output
Interrupt generation

11.2GPIO_A Register Descriptions

GPIO_A Register are described in the following.

GPIO_BASE(CPU): 0x5090_0000
GPIO_BASE(MCU): 0x5530_0000

11.2.1GPIO_SWPORTA_DR: Port A data register

Address: GPIO_BASE + 0x00

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

Bit	Type	Reset	Description
31-0	R/W	32'h0	Values written to this register are output on the I/O signals for Port A if the corresponding data direction bits for Port A are set to Output mode and the corresponding control bit for Port A is set to Software mode. The value read back is equal to the last value written to this register.

11.2.2GPIO_SWPORTA_DDR : Port A Data Direction Register

Address: GPIO_BASE + 0x04

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

Bit	Type	Reset	Description
31-0	R/W	0	Values written to this register independently control the direction of the corresponding data bit in Port A. **Values:** 0x0 (IN): Input Direction 0x1 (OUT): Output Direction

11.2.3GPIO_INTEN_A: Interrupt Enable Register

Address: GPIO_BASE + 0x30

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (DISABLED): Interrupt is disabled 0x1 (ENABLED): Interrupt is enabled

11.2.4GPIO_INTMASK_A: Interrupt mask register

Address: GPIO_BASE + 0x34

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

Bit	Type	Reset	Description
31-0	R/W	32'h0	0x0 (DISABLED): Interrupt bits are unmasked 0x1 (ENABLED): Mask interrupt

11.2.5GPIO_INTTYPE_LEVEL_A: Interrupt level

Address: GPIO_BASE + 0x38

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (LEVEL_SENSITIVE): Interrupt is level sensitive 0x1 (EDGE_SENSITIVE): Interrupt is edge sensitive

11.2.6GPIO_INT_POLARITY_A: Interrupt polarity

Address: GPIO_BASE + 0x3C

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (ACTIVE_LOW): Active Low polarity 0x1 (ACTIVE_HIGH): Active High polarity

11.2.7GPIO_INTSTATUS_A: Interrupt status

Address: GPIO_BASE + 0x40

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.2.8GPIO_RAW_INTSTATUS_A: Raw interrupt status

Address: GPIO_BASE + 0x44

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.2.9GPIO_DEBOUNCE_A: Debounce enable

Address: GPIO_BASE + 0x48

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (DISABLED): No debounce 0x1 (ENABLED): Enable debounce

11.2.10GPIO_PORTA_EOI : Port A clear interrupt register

Address: GPIO_BASE + 0x4C

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

Bit	Type	Reset	Description

31-0	W	0	Controls the clearing of edge type interrupts from Port A. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port A is not configured for interrupts. **Values:** 0x0 (DISABLED): No interrupt clear 0x1 (ENABLED): Clear Interrupt

11.2.11GPIO_EXT_PORTA : External port A register

Address: GPIO_BASE + 0x50

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

Bit	Type	Reset	Description
31-0	R	0	This register always reflects the signals value on the External Port A.

11.2.12GPIO_LS_SYNC_A : Synchronization level

Address: GPIO_BASE + 0x60

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

Bit	Type	Reset	Description
31-0	R/W	0	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. **Values:** 0x0 (DISABLED): No synchronization to pclk_intr (default) 0x1 (ENABLED): Synchronize to pclk_intr

11.2.13GPIO_INT_BOTHEDGE _A : Interrupt Both Edge type

Address: GPIO_BASE + 0x68

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

Bit	Type	Reset	Description
31-0	R/W	0	Controls the edge type of interrupt that can occur on Port A. -Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling

			edge of an external input signal corresponding to that bit on port A. -The values programmed in the Register gpio_intype_level and gpio_int_polarity for this particular bit are not considered when the corresponding bit of this register is set to 1. -Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio_inttype_level and gpio_int_polarity Register. **Values:** 0x0 (DISABLED): single edge sensitive 0x1 (ENABLED): both edge sensitive

11.3GPIO_B Register Descriptions

GPIO_B Register are described in the following.

11.3.1GPIO_SWPORTB_DR: Port B data register

Address: GPIO_BASE + 0x80 + 0x00

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

Bit	Type	Reset	Description
31-0	R/W	32'h0	Values written to this register are output on the I/O signals for Port B if the corresponding data direction bits for Port B are set to Output mode and the corresponding control bit for Port B is set to Software mode. The value read back is equal to the last value written to this register.

11.3.2GPIO_SWPORTB_DDR : Port B Data Direction Register

Address: GPIO_BASE + 0x80 + 0x04

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

Bit	Type	Reset	Description
31-0	R/W	0	Values written to this register independently control the direction of the corresponding data bit in Port B. **Values:** 0x0 (IN): Input Direction 0x1 (OUT): Output Direction

11.3.3GPIO_INTEN_B: Interrupt Enable Register

Address: GPIO_BASE + 0x80 + 0x30

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (DISABLED): Interrupt is disabled 0x1 (ENABLED): Interrupt is enabled

11.3.4GPIO_INTMASK_B: Interrupt mask register

Address: GPIO_BASE + 0x80 + 0x34

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

Bit	Type	Reset	Description
31-0	R/W	32'h0	0x0 (DISABLED): Interrupt bits are unmasked 0x1 (ENABLED): Mask interrupt

11.3.5GPIO_INTTYPE_LEVEL_B: Interrupt level

Address: GPIO_BASE + 0x80 + 0x38

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (LEVEL_SENSITIVE): Interrupt is level sensitive 0x1 (EDGE_SENSITIVE): Interrupt is edge sensitive

11.3.6GPIO_INT_POLARITY_B: Interrupt polarity

Address: GPIO_BASE + 0x80 + 0x3C

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (ACTIVE_LOW): Active Low polarity 0x1 (ACTIVE_HIGH): Active High polarity

11.3.7GPIO_INTSTATUS_B: Interrupt status

Address: GPIO_BASE + 0x80 + 0x40

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.3.8GPIO_RAW_INTSTATUS_B: Raw interrupt status

Address: GPIO_BASE + 0x80 + 0x44

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.3.9GPIO_DEBOUNCE_B: Debounce enable

Address: GPIO_BASE + 0x80 + 0x48

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (DISABLED): No debounce 0x1 (ENABLED): Enable debounce

11.3.10GPIO_PORTB_EOI : Port B clear interrupt register

Address: GPIO_BASE + 0x80 + 0x4C

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

Bit	Type	Reset	Description
31-0	W	0	Controls the clearing of edge type interrupts from Port B. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port B is not configured for interrupts. **Values:** 0x0 (DISABLED): No interrupt clear 0x1 (ENABLED): Clear Interrupt

11.3.11GPIO_EXT_PORTB : External port B register

Address: GPIO_BASE + 0x80 + 0x50

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

Bit	Type	Reset	Description
31-0	R	0	This register always reflects the signals value on the External Port B.

11.3.12GPIO_LS_SYNC_B : Synchronization level

Address: GPIO_BASE + 0x80 + 0x60

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

Bit	Type	Reset	Description
31-0	R/W	0	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. **Values:** 0x0 (DISABLED): No synchronization to pclk_intr (default) 0x1 (ENABLED): Synchronize to pclk_intr

11.3.13GPIO_INT_BOTHEDGE _B : Interrupt Both Edge type

Address: GPIO_BASE + 0x80 + 0x68

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

Bit	Type	Reset	Description
31-0	R/W	0	Controls the edge type of interrupt that can occur on Port B. -Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on port B. -The values programmed in the Register gpio_intype_level and gpio_int_polarity for this particular bit are not considered when the corresponding bit of this register is set to 1. -Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio_inttype_level and gpio_int_polarity Register. **Values:** 0x0 (DISABLED): single edge sensitive 0x1 (ENABLED): both edge sensitive

11.4 GPIO_C Register Descriptions

GPIO_C Register are described in the following.

11.4.1GPIO_SWPORTC_DR: Port C data register

Address: GPIO_BASE + 0x100 + 0x00

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

Bit	Type	Reset	Description
31-0	R/W	32'h0	Values written to this register are output on the I/O signals for Port C if the corresponding data direction bits for Port C are set to Output mode and the corresponding control bit for Port C is set to Software mode. The value read back is equal to the last value written to this register.

11.4.2GPIO_SWPORTC_DDR : Port C Data Direction Register

Address: GPIO_BASE + 0x100 + 0x04

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

Bit	Type	Reset	Description
31-0	R/W	0	Values written to this register independently control the direction of the corresponding data bit in Port C. **Values:** 0x0 (IN): Input Direction 0x1 (OUT): Output Direction

11.4.3GPIO_INTEN_C: Interrupt Enable Register

Address: GPIO_BASE + 0x100 + 0x30

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (DISABLED): Interrupt is disabled 0x1 (ENABLED): Interrupt is enabled

11.4.4GPIO_INTMASK_C: Interrupt mask register

Address: GPIO_BASE + 0x100 + 0x34

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

Bit	Type	Reset	Description
31-0	R/W	32'h0	0x0 (DISABLED): Interrupt bits are unmasked 0x1 (ENABLED): Mask interrupt

11.4.5GPIO_INTTYPE_LEVEL_C: Interrupt level

Address: GPIO_BASE + 0x100 + 0x38

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (LEVEL_SENSITIVE): Interrupt is level sensitive 0x1 (EDGE_SENSITIVE): Interrupt is edge sensitive

11.4.6GPIO_INT_POLARITY_C: Interrupt polarity

Address: GPIO_BASE + 0x100 + 0x3C

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (ACTIVE_LOW): Active Low polarity 0x1 (ACTIVE_HIGH): Active High polarity

11.4.7GPIO_INTSTATUS_C: Interrupt status

Address: GPIO_BASE + 0x100 + 0x40

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.4.8GPIO_RAW_INTSTATUS_C: Raw interrupt status

Address: GPIO_BASE + 0x100 + 0x44

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.4.9GPIO_DEBOUNCE_C: Debounce enable

Address: GPIO_BASE + 0x100 + 0x48

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (DISABLED): No debounce 0x1 (ENABLED): Enable debounce

11.4.10GPIO_PORTC_EOI : Port C clear interrupt register

Address: GPIO_BASE + 0x100 + 0x4C

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

Bit	Type	Reset	Description
31-0	W	0	Controls the clearing of edge type interrupts from Port C. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port C is not configured for interrupts. **Values:** 0x0 (DISABLED): No interrupt clear 0x1 (ENABLED): Clear Interrupt

11.4.11GPIO_EXT_PORTC : External port C register

Address: GPIO_BASE + 0x100 + 0x50

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

Bit	Type	Reset	Description
31-0	R	0	This register always reflects the signals value on the External

			Port C.

11.4.12GPIO_LS_SYNC_C : Synchronization level

Address: GPIO_BASE + 0x100 + 0x60

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

Bit	Type	Reset	Description
31-0	R/W	0	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. **Values:** 0x0 (DISABLED): No synchronization to pclk_intr (default) 0x1 (ENABLED): Synchronize to pclk_intr

11.4.13GPIO_INT_BOTHEDGE_C : Interrupt Both Edge type

Address: GPIO_BASE + 0x100 + 0x68

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

Bit	Type	Reset	Description
31-0	R/W	0	Controls the edge type of interrupt that can occur on Port C. -Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on port C. -The values programmed in the Register gpio_intype_level and gpio_int_polarity for this particular bit are not considered when the corresponding bit of this register is set to 1. -Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio_inttype_level and gpio_int_polarity Register. **Values:** 0x0 (DISABLED): single edge sensitive 0x1 (ENABLED): both edge sensitive

11.5 GPIO_D Register Descriptions

GPIO_D Register are described in the following.

11.5.1GPIO_SWPORTD_DR: Port D data register

Address: GPIO_BASE + 0x180 + 0x00

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

Bit	Type	Reset	Description
31-0	R/W	32'h0	Values written to this register are output on the I/O signals for Port D if the corresponding data direction bits for Port D are set to Output mode and the corresponding control bit for Port D is set to Software mode. The value read back is equal to the last value written to this register.

11.5.2GPIO_SWPORTD_DDR : Port D Data Direction Register

Address: GPIO_BASE + 0x180 + 0x04

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

Bit	Type	Reset	Description
31-0	R/W	0	Values written to this register independently control the direction of the corresponding data bit in Port D. **Values:** 0x0 (IN): Input Direction 0x1 (OUT): Output Direction

11.5.3GPIO_INTEN_D: Interrupt Enable Register

Address: GPIO_BASE + 0x180 + 0x30

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

Bit	Type	Reset	Description

31-0	R/W	0	0x0 (DISABLED): Interrupt is disabled 0x1 (ENABLED): Interrupt is enabled

11.5.4GPIO_INTMASK_D: Interrupt mask register

Address: GPIO_BASE + 0x180 + 0x34

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

Bit	Type	Reset	Description
31-0	R/W	32'h0	0x0 (DISABLED): Interrupt bits are unmasked 0x1 (ENABLED): Mask interrupt

11.5.5GPIO_INTTYPE_LEVEL_D: Interrupt level

Address: GPIO_BASE + 0x180 + 0x38

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (LEVEL_SENSITIVE): Interrupt is level sensitive 0x1 (EDGE_SENSITIVE): Interrupt is edge sensitive

11.5.6GPIO_INT_POLARITY_D: Interrupt polarity

Address: GPIO_BASE + 0x180 + 0x3C

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (ACTIVE_LOW): Active Low polarity 0x1 (ACTIVE_HIGH): Active High polarity

11.5.7GPIO_INTSTATUS_D: Interrupt status

Address: GPIO_BASE + 0x180 + 0x40

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.5.8GPIO_RAW_INTSTATUS_D: Raw interrupt status

Address: GPIO_BASE + 0x180 + 0x44

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.5.9GPIO_DEBOUNCE_D: Debounce enable

Address: GPIO_BASE + 0x180 + 0x48

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (DISABLED): No debounce 0x1 (ENABLED): Enable debounce

11.5.10GPIO_PORTD_EOI : Port D clear interrupt register

Address: GPIO_BASE + 0x180 + 0x4C

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

Bit	Type	Reset	Description

31-0	W	0	Controls the clearing of edge type interrupts from Port D. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port D is not configured for interrupts. **Values:** 0x0 (DISABLED): No interrupt clear 0x1 (ENABLED): Clear Interrupt

11.5.11GPIO_EXT_PORTD : External port D register

Address: GPIO_BASE + 0x180 + 0x50

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

Bit	Type	Reset	Description
31-0	R	0	This register always reflects the signals value on the External Port D.

11.5.12GPIO_LS_SYNC_D : Synchronization level

Address: GPIO_BASE + 0x180 + 0x60

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

Bit	Type	Reset	Description
31-0	R/W	0	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. **Values:** 0x0 (DISABLED): No synchronization to pclk_intr (default) 0x1 (ENABLED): Synchronize to pclk_intr

11.5.13GPIO_INT_BOTHEDGE_D : Interrupt Both Edge type

Address: GPIO_BASE + 0x180 + 0x68

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

Bit	Type	Reset	Description
31-0	R/W	0	Controls the edge type of interrupt that can occur on Port D. -Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on port D. -The values programmed in the Register gpio_intype_level and gpio_int_polarity for this particular bit are not considered when the corresponding bit of this register is set to 1. -Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio_inttype_level and gpio_int_polarity Register. **Values:** 0x0 (DISABLED): single edge sensitive 0x1 (ENABLED): both edge sensitive

11.6 GPIO_E Register Descriptions

GPIO_E Register are described in the following.

11.6.1GPIO_SWPORTE_DR: Port E data register

Address: GPIO_BASE + 0x200 + 0x00

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

Bit	Type	Reset	Description
31-0	R/W	32'h0	Values written to this register are output on the I/O signals for Port E if the corresponding data direction bits for Port E are set to Output mode and the corresponding control bit for Port E is set to Software mode. The value read back is equal to the last value written to this register.

11.6.2GPIO_SWPORTE_DDR : Port E Data Direction Register

Address: GPIO_BASE + 0x200 + 0x04

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

Bit	Type	Reset	Description
31-0	R/W	0	Values written to this register independently control the direction of the corresponding data bit in Port E.

			Values: 0x0 (IN): Input Direction 0x1 (OUT): Output Direction

11.6.3GPIO_INTEN_E: Interrupt Enable Register

Address: GPIO_BASE + 0x200 + 0x30

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (DISABLED): Interrupt is disabled 0x1 (ENABLED): Interrupt is enabled

11.6.4GPIO_INTMASK_E: Interrupt mask register

Address: GPIO_BASE + 0x200 + 0x34

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

Bit	Type	Reset	Description
31-0	R/W	32’h0	0x0 (DISABLED): Interrupt bits are unmasked 0x1 (ENABLED): Mask interrupt

11.6.5GPIO_INTTYPE_LEVEL_E: Interrupt level

Address: GPIO_BASE + 0x200 + 0x38

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (LEVEL_SENSITIVE): Interrupt is level sensitive 0x1 (EDGE_SENSITIVE): Interrupt is edge sensitive

11.6.6GPIO_INT_POLARITY_E: Interrupt polarity

Address: GPIO_BASE + 0x200 +

0x3C

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (ACTIVE_LOW): Active Low polarity 0x1 (ACTIVE_HIGH): Active High polarity

11.6.7GPIO_INTSTATUS_E: Interrupt status

Address: GPIO_BASE + 0x200 + 0x40

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.6.8GPIO_RAW_INTSTATUS_E: Raw interrupt status

Address: GPIO_BASE + 0x200 + 0x44

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

Bit	Type	Reset	Description
31-0	R	0	0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active

11.6.9GPIO_DEBOUNCE_E: Debounce enable

Address: GPIO_BASE + 0x200 + 0x48

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

Bit	Type	Reset	Description
31-0	R/W	0	0x0 (DISABLED): No debounce 0x1 (ENABLED): Enable debounce

11.6.10GPIO_PORTE_EOI : Port E clear interrupt register

Address: GPIO_BASE + 0x200 + 0x4C

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

Bit	Type	Reset	Description
31-0	W	0	Controls the clearing of edge type interrupts from Port E. When a 1 is written into a corresponding bit of this register, the interrupt is cleared. All interrupts are cleared when Port E is not configured for interrupts. **Values:** 0x0 (DISABLED): No interrupt clear 0x1 (ENABLED): Clear Interrupt

11.6.11GPIO_EXT_PORTE : External port E register

Address: GPIO_BASE + 0x200 + 0x50

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

Bit	Type	Reset	Description
31-0	R	0	This register always reflects the signals value on the External Port E.

11.6.12GPIO_LS_SYNC_E : Synchronization level

Address: GPIO_BASE + 0x200 + 0x60

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

Bit	Type	Reset	Description
31-0	R/W	0	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. **Values:** 0x0 (DISABLED): No synchronization to pclk_intr (default) 0x1 (ENABLED): Synchronize to pclk_intr

11.6.13GPIO_INT_BOTHEDGE_E : Interrupt Both Edge type

Address: GPIO_BASE + 0x200 + 0x68

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

Bit	Type	Reset	Description
31-0	R/W	0	Controls the edge type of interrupt that can occur on Port E. -Whenever a particular bit is programmed to 1, it enables the generation of interrupts on both the rising edge and the falling edge of an external input signal corresponding to that bit on port E. -The values programmed in the Register gpio_intype_level and gpio_int_polarity for this particular bit are not considered when the corresponding bit of this register is set to 1. -Whenever a particular bit is programmed to 0, the interrupt type depends on the value of the corresponding bits in the gpio_inttype_level and gpio_int_polarity Register. **Values:** 0x0 (DISABLED): single edge sensitive 0x1 (ENABLED): both edge sensitive

12.I2C

I2C_BASE: 0x5030_0000

12.1 Features

12.1.1 I2C Features

- Two-wire I^2C serial interface – consists of a serial data line (SDA) and a serial clock (SCL)
  - Three speeds:
    - Standard mode (0 to 100 Kb/s)
    - Fast mode (up to 400 Kb/s) or fast mode plus (up to 1Mb/s)
    - High-speed mode (up to 3.4 Mb/s)
  - Clock synchronization
  - Master OR slave I^2C operation
  - 7- or 10-bit addressing
  - 7- or 10-bit combined format transfers
  - Bulk transmit mode
  - Ignores CBUS addresses (an older ancestor of I^2C that used to share the I^2C bus)
  - Transmit and receive buffers
  - Interrupt or polled-mode operation
  - Handles Bit and Byte waiting at all bus speeds
  - Simple software interface consistent with APB peripherals
  - Component parameters for configurable software driver support
  - DMA handshaking interface compatible with the ahb dmac handshaking interface
  - Programmable SDA hold time (tHD;DAT)
  - Bus clear feature
  - Device ID feature
  - SMBus/PMBus support
  - SMBus Slave detects and responds to ARP commands.
  - Ultra-Fast mode support
  - Multiple SAR Feature
  - UDID feature support for all slaves (IC_SAR, IC_SAR2, IC_SAR3 and IC_SAR4)

12.2 Register Descriptions

12.2.1 IC_CON

- **Name:** I2C Control Register
- **Description:** I2C Control Register.

This register can be written only when the Apb_i2c is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.

Read/Write Access:

- If configuration parameter I2C_DYNAMIC_TAR_UPDATE=1, bit 4 is read only.
- If configuration parameter IC_RX_FULL_HLD_BUS_EN =0, bit 9 is read only.
- If configuration parameter IC_STOP_DET_IF_MASTER_ACTIVE =0, bit 10 is read only.
- If configuration parameter IC_BUS_CLEAR_FEATURE=0, bit 11 is read only
- If configuration parameter IC_OPTIONAL_SAR=0, bit 16 is read only
- If configuration parameter IC_SMBUS=0, bit 17 is read only
- If configuration parameter IC_SMBUS_ARP=0, bits 18 and 19 are read only.

- **Size:** 32 bits
- **Offset:** 0x0
- **Exists:** Always

Fields for Register: IC_CON

Bits	Name	Memory Access	Description
31:26	RSVD_IC_CON_2	R	IC_CON_2 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

25	IC_SAR4_SMBUS_ARP_EN	R/W	This bit controls whether Apb_i2c should enable Address Resolution Logic in SMBus Mode for IC_SAR4 Slave Address. The Slave mode will decode the Address Resolution Protocol commands and respond to it. The Apb_i2c slave also includes the generation/validity of PEC byte for Address Resolution Protocol commands. This bit is applicable only in Slave mode. **Values:** ▪ 0x1 (ENABLED): SMBus ARP control is enabled. ▪ 0x0 (DISABLED): SMBus ARP control is disabled. **Value After Reset:** 0x0
24	IC_SAR3_SMBUS_ARP_EN	R/W	This bit controls whether Apb_i2c should enable Address Resolution Logic in SMBus Mode for IC_SAR3 Slave Address. The Slave mode will decode the Address Resolution Protocol commands and respond to it. The Apb_i2c slave also includes the generation/validity of PEC byte for Address Resolution Protocol commands. This bit is applicable only in Slave mode. **Values:** ▪ 0x1 (ENABLED): SMBus ARP control is enabled. ▪ 0x0 (DISABLED): SMBus ARP control is disabled. **Value After Reset:** 0x0
Bits	**Name**	**Memory Access**	**Description**

23	IC_SAR2_SMBUS_ARP_EN	R/W	This bit controls whether Apb_i2c should enable Address Resolution Logic in SMBus Mode for IC_SAR2 Slave Address. The Slave mode will decode the Address Resolution Protocol commands and respond to it. The Apb_i2c slave also includes the generation/validity of PEC byte for Address Resolution Protocol commands. This bit is applicable only in Slave mode. **Values:** ▪ 0x1 (ENABLED): SMBus ARP control is enabled. ▪ 0x0 (DISABLED): SMBus ARP control is disabled. **Value After Reset:** 0x0
22	SMBUS_PERSISTENT_SLV_AD DR4_EN	R/W	The bit controls to enable Apb_i2c slave with Slave address 4 (IC_SAR4) as persistent or non persistent slave. If the slave is non-PSA then Apb_i2c slave device clears the Address valid flag for both General and Directed Reset ARP command else the address valid flag will always set to 1. This bit is applicable only in Slave mode. **Values:** ▪ 0x1 (ENABLED): SMBus Persistent Slave address (IC_SAR4) control is enabled. ▪ 0x0 (DISABLED): SMBus Persistent Slave address (IC_SAR4) control is disabled.
Bits	**Name**	**Memory Access**	**Description**
21	SMBUS_PERSISTENT_SLV_AD DR3_EN	R/W	The bit controls to enable Apb_i2c slave with Slave address 3 (IC_SAR3) as persistent or non persistent slave. If the slave is non-PSA then Apb_i2c slave device clears the Address valid flag for both General and Directed Reset ARP command else the address valid flag will always set to 1. This bit is applicable only in Slave mode. **Values:** ▪ 0x1 (ENABLED): SMBus Persistent Slave address (IC_SAR3) control is enabled. ▪ 0x0 (DISABLED): SMBus Persistent Slave address (IC_SAR3) control is disabled.

20	SMBUS_PERSISTENT_SLV_AD DR2_EN	R/W	The bit controls to enable Apb_i2c slave with Slave address 2 (IC_SAR2) as persistent or non persistent slave. If the slave is non-PSA then Apb_i2c slave device clears the Address valid flag for both General and Directed Reset ARP command else the address valid flag will always set to 1. This bit is applicable only in Slave mode. **Values:** ■ 0x1 (ENABLED): SMBus Persistent Slave address (IC_SAR2) control is enabled. ■ 0x0 (DISABLED): SMBus Persistent Slave address (IC_SAR2)control is disabled.

Bits	Name	Memory Access	Description
19	SMBUS_PERSISTENT_SLV_AD DR_EN	R/W	The bit controls to enable Apb_i2c slave as persistent or non persistent slave. If the slave is non-PSA then Apb_i2c slave device clears the Address valid flag for both General and Directed Reset ARP command else the address valid flag will always set to 1. This bit is applicable only in Slave mode. **Values:** ■ 0x1 (ENABLED): SMBus Persistent Slave address control is enabled. ■ 0x0 (DISABLED): SMBus Persistent Slave address control is disabled.
18	SMBUS_ARP_EN	R/W	This bit controls whether Apb_i2c should enable Address Resolution Logic in SMBus Mode. The Slave mode will decode the Address Resolution Protocol commands and respond to it. The Apb_i2c slave also includes the generation/validity of PEC byte for Address Resolution Protocol commands. This bit is applicable only in Slave mode. **Values:** ■ 0x1 (ENABLED): SMBus ARP control is enabled. ■ 0x0 (DISABLED): SMBus ARP control is disabled.

17	SMBUS_SLAVE_QUICK_EN	R/W	If this bit is set to 1, Apb_i2c slave only receives Quick commands in SMBus Mode. If this bit is set to 0, Apb_i2c slave receives all bus protocols but not Quick commands. This bit is applicable only in slave mode. **Values:** ▪ 0x1 (ENABLED): SMBus Slave is enabled to receive Quick command. ▪ 0x0 (DISABLED): SMBus Slave is disabled to receive Quick command.
Bits	**Name**	**Memory Access**	**Description**
16	OPTIONAL_SAR_CTRL	R/W	Enables the usage of IC_OPTIONAL_SAR register. If IC_OPTIONAL_SAR = 1, IC_OPTIONAL_SAR value is used as additional slave address. User must program a valid address in IC_OPTIONAL_SAR before writing 1 to this field. If IC_OPTIONAL_SAR = 0, IC_OPTIONAL_SAR value is not used as additional slave address. In this mode only one I2C slave address is used. **Values:** ▪ 0x1 (ENABLED): Optional SAR Address Register is enabled. ▪ 0x0 (DISABLED): Optional SAR Address Register is disabled.
15:12	RSVD_IC_CON_1	R	IC_CON_1 Reserved bits - Read Only

11	BUS_CLEAR_FEATURE_CTRL	R/W	This bit controls whether the Apb_i2c enables the bus clear feature. **Values:** ■ 0x1 (ENABLED): Bus Clear Feature is enabled. ■ 0x0 (DISABLED): Bus Clear Feature is disabled.
Bits	**Name**	**Memory Access**	**Description**
10	STOP_DET_IF_MASTER_ACTIV E	* Varies	In Master mode: ■ 1'b1: issues the STOP_DET interrupt only when master is active. ■ 1'b0: issues the STOP_DET irrespective of whether master is active or not. **Values:** ■ 0x1 (ENABLED): Master issues the STOP_DET interrupt only when master is active ■ 0x0 (DISABLED): Master issues the STOP_DET interrupt irrespective of whether master is active or not **Value After Reset:** 0x0 **Exists:** Always
9	RX_FIFO_FULL_HLD_CTRL	* Varies	This bit controls whether Apb_i2c should hold the bus when the Rx FIFO is physically full to its RX_BUFFER_DEPTH, as described in the IC_RX_FULL_HLD_BUS_EN parameter. **Values:** ■ 0x1 (ENABLED): Hold bus when RX_FIFO is full ■ 0x0 (DISABLED): Overflow when RX_FIFO is full **Value After Reset:** 0x0 **Exists:** Always

8	TX_EMPTY_CTRL	R/W	This bit controls the generation of the TX_EMPTY interrupt, as described in the IC_RAW_INTR_STAT register. **Values:** ▪ 0x1 (ENABLED): Controlled generation of TX_EMPTY interrupt ▪ 0x0 (DISABLED): Default behaviour of TX_EMPTY interrupt **Value After Reset:** 0x0 **Exists:** Always
Bits	**Name**	**Memory Access**	**Description**
7	STOP_DET_IFADDRESSED	R/W	In slave mode: ▪ 1'b1: issues the STOP_DET interrrupt only when it is addressed. ▪ 1'b0: issues the STOP_DET irrespective of whether it's addressed or not. **Note:** During a general call address, this slave does not issue the STOP_DET interrupt if STOP_DET_IF_ADDRESSED = 1'b1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (SAR). **Values:** ▪ 0x1 (ENABLED): slave issues STOP_DET intr only if addressed ▪ 0x0 (DISABLED): slave issues STOP_DET intr always

6	IC_SLAVE_DISABLE	R/W	This bit controls whether I2C has its slave disabled, which means once the presetn signal is applied, then this bit takes on the value of the configuration parameter IC_SLAVE_DISABLE. You have the choice of having the slave enabled or disabled after reset is applied, which means software does not have to configure the slave. By default, the slave is always enabled (in reset state as well). If you need to disable it after reset, set this bit to 1. If this bit is set (slave is disabled), Apb_i2c functions only as a master and does not perform any action that requires a slave. **Note:** Software should ensure that if this bit is written with 0, then bit 0 should also be written with a 0. **Values:** ▪ 0x1 (SLAVE_DISABLED): Slave mode is disabled ▪ 0x0 (SLAVE_ENABLED): Slave mode is enabled

Bits	Name	Memory Access	Description
5	IC_RESTART_EN	R/W	Determines whether RESTART conditions may be sent when acting as a master. Some older slaves do not support handling RESTART conditions; however, RESTART conditions are used in several Apb_i2c operations. When RESTART is disabled, the master is prohibited from performing the following functions: ▪ Sending a START BYTE ▪ Performing any high-speed mode operation ▪ High-speed mode operation ▪ Performing direction changes in combined format mode ▪ Performing a read operation with a 10-bit address By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple Apb_i2c transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. **Values:** ▪ 0x1 (ENABLED): Master restart enabled ▪ 0x0 (DISABLED): Master restart disabled

4	IC_10BITADDR_MASTER	R/W	IftheI2C_DYNAMIC_TAR_UPDATE configurationparameter is set to 'No' (0), this bit is named IC_10BITADDR_MASTER and controls whether the Apb_i2c starts its transfers in 7- or 10-bit addressing mode when acting as a master. If I2C_DYNAMIC_TAR_UPDATE is set to 'Yes' (1), the function of this bit is handled by bit 12 of IC_TAR register, and becomes a read-only copy called IC_10BITADDR_MASTER_rd_only. **Values:** ■ 0x1 (ADDR_10BITS): Master 10-bit addressing mode ■ 0x0 (ADDR_7BITS): Master 7-bit addressing mode
Bits	**Name**	**Memory Access**	**Description**
3	IC_10BITADDR_SLAVE	R/W	When acting as a slave, this bit controls whether the Apb_i2c responds to 7- or 10-bit addresses. ■ 0: 7-bit addressing. The Apb_i2c ignores transactions that involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared. ■ 1: 10-bit addressing. The Apb_i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register. **Values:** ■ 0x1 (ADDR_10BITS): Slave 10Bit addressing ■ 0x0 (ADDR_7BITS): Slave 7Bit addressing

2:1	SPEED	R/W	These bits control at which speed the Apb_i2c operates; its setting is relevant only if one is operating the Apb_i2c in master mode. Hardware protects against illegal values being programmed by software. These bits must be programmed appropriately for slave mode also, as it is used to capture correct value of spike filter as per the speed mode. This register should be programmed only with a value in the range of 1 to IC_MAX_SPEED_MODE; otherwise, hardware updates this register with the value of IC_MAX_SPEED_MODE. 1: standard mode (100 kbit/s) 2: fast mode (<=400 kbit/s) or fast mode plus (<=1000Kbit/s) 3: high speed mode (3.4 Mbit/s) **Note:** This field is not applicable when IC_ULTRA_FAST_MODE=1 **Values:** ■ 0x1 (STANDARD): Standard Speed mode of operation 0x2 (FAST): Fast or Fast Plus mode of operation ■ 0x3 (HIGH): High Speed mode of operation
Bits	**Name**	**Memory Access**	**Description**
0	MASTER_MODE	R/W	This bit controls whether the Apb_i2c master is enabled. **Note:** Software should ensure that if this bit is written with '1' then bit 6 should also be written with a '1'. **Values:** ■ 0x1 (ENABLED): Master mode is enabled ■ 0x0 (DISABLED): Master mode is disabled

12.2.2 IC_TAR

- **Name:** I2C Target Address Register
- **Description:** I2C Target Address Register

If the configuration parameter I2C_DYNAMIC_TAR_UPDATE is set to 'No' (0), this register

is 12 bits wide, and bits 31:12 are reserved. This register can be written to only when IC_ENABLE[0]

is set to 0.However, if I2C_DYNAMIC_TAR_UPDATE = 1, then the register becomes 13 bits wide.

In this case, writes to IC_TAR succeed when one of the following conditions are true:

- Apb_i2c is NOT enabled (IC_ENABLE[0] is set to 0); or
- Apb_i2c is enabled (IC_ENABLE[0]=1); AND Apb_i2c is NOT engaged in any Master (tx, rx) operation (IC_STATUS[5]=0); AND Apb_i2c is enabled to operate in Master mode (IC_CON[0]=1); AND there are NO entries in the TX FIFO (IC_STATUS[2]=1)

You can change the TAR address dynamically without losing the bus, only if the following conditions are met.

- Apb_i2c is enabled (IC_ENABLE[0]=1); AND IC_EMPTYFIFO_HOLD_MASTER_EN configuration parameter is set to 1; AND Apb_i2c is enabled to operate in Master mode (IC_CON[0]=1); AND there are NO entries in the Tx FIFO and the master is in HOLD state (IC_INTR_STAT[13]=1).

Note: If the software or application is aware that the Apb_i2c is not using the TAR address for the pending commands in the Tx FIFO, then it is possible to update the TAR address even while the Tx FIFO has entries (IC STATUS[2]= 0).

- It is not necessary to perform any write to this register if Apb_i2c is enabled as an I2C slave only.

- **Size:** 32 bits
- **Offset:** 0x4
- **Exists:** Always

Bits	Name	Memory Access	Description
31:17	RSVD_IC_TAR_2	R	IC_TAR_2 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

16	SMBUS_QUICK_CMD	R/W	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a Quick command is to be performed by the Apb_i2c. **Values:** ▪ 0x1 (ENABLED): Enables programming of QUICK-CMD transmission ▪ 0x0 (DISABLED): Disables programming of QUICK-CMD transmission **Value After Reset:** 0x0
15:14	RSVD_IC_TAR_1	R	IC_TAR_1 Reserved bits - Read Only **Value After Reset:** 0x0
13	DEVICE_ID	R/W	If bit 11 (SPECIAL) is set to 1, then this bit indicates whether a Device-ID of a particular slave mentioned in IC_TAR[9:0] is to be performed by the Apb_i2c Master. ▪ 0: Device-ID is not performed and checks ic_tar[10] to perform either general call or START byte command ▪ 1: Device-ID transfer is performed and bytes based on the number of read commands in the Tx-FIFO are received from the targeted slave and put in the Rx-FIFO. **Values:** ▪ 0x1 (ENABLED): Enables programming of DEVICE-ID transmission ▪ 0x0 (DISABLED): Disables programming of DEVICE-ID transmission **Value After Reset:** 0x0
12	IC_10BITADDR_MASTER	R/W	This bit controls whether the Apb_i2c starts its transfers in 7- or 10-bit addressing mode when acting as a master. **Values:** ▪ 0x1 (ADDR_10BITS): Address 10Bit transmission format ▪ 0x0 (ADDR_7BITS): Address 7Bit transmission format

Bits	Name	Memory Access	Description
11	SPECIAL	R/W	This bit indicates whether software performs a Device-ID or General Call or START BYTE command. ▪ 0: ignore bit 10 GC_OR_START and use IC_TAR normally ▪ 1: perform special I2C command as specified in Device_ID or GC_OR_START bit **Values:** ▪ 0x1 (ENABLED): Enables programming of GENERAL_CALL or START_BYTE transmission ▪ 0x0 (DISABLED): Disables programming of GENERAL_CALL or START_BYTE transmission **Value After Reset:** 0x0 **Exists:** Always
10	GC_OR_START	R/W	If bit 11 (SPECIAL) is set to 1 and bit 13(Device-ID) is set to 0, then this bit indicates whether a General Call or START byte command is to be performed by the Apb_i2c. ▪ 0: General Call Address - after issuing a General Call, only writes may be performed. Attempting to issue a read command results in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register. The Apb_i2c remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. ▪ 1: START BYTE **Values:** ▪ 0x1 (START_BYTE): START byte transmission ▪ 0x0 (GENERAL_CALL): GENERAL_CALL byte transmission **Value After Reset:** 0x0 **Exists:** Always

9:0	IC_TAR	R/W	This is the target address for any master transaction. When transmitting a General Call, these bits are ignored. To generate a START BYTE, the CPU needs to write only once into these bits. If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. A master cannot transmit to itself; it can transmit to only a slave.

12.2.3 IC_SAR

- **Name:** I2C Slave Address Register
- **Description:** I2C Slave Address Register
- **Size:** 32 bits
- **Offset:** 0x8
- **Exists:** Always

Bits	Name	Memory Access	Description
31:10	RSVD_IC_SAR	R	IC_SAR Reserved bits - Read Only **Value After Reset:**
9:0	IC_SAR	R/W	The IC_SAR holds the slave address when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. **Note:** The default values cannot be any of the reserved address locations: that is, 0x00 to 0x07, or 0x78 to 0x7f. The correct operation of the device is not guaranteed if you program the IC_SAR or IC_TAR to a reserved value. **Value After Reset:** IC_DEFAULT_SLAVE_ADDR **Exists:** Always **Volatile:** true

12.2.4 IC_HS_MADDR

- **Name:** I2C High Speed Master Mode Code Address Register

- **Description:** I2C High Speed Master Mode Code Address Register
- **Size:** 32 bits
- **Offset:** 0xc

Bits	Name	Memory Access	Description
31:3	RSVD_IC_HS_MAR	R	IC_HS_MAR Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always
2:0	IC_HS_MAR	R/W	This bit field holds the value of the I2C HS mode master code. HS-mode master codes are reserved 8-bit codes (00001xxx) that are not used for slave addressing or other purposes. Each master has its unique master code; up to eight high-speed mode masters can be present on the same I2C bus system. Valid values are from 0 to 7. This register goes away and becomes read-only returning 0's if the IC_MAX_SPEED_MODE configuration parameter is set to either Standard (1) or Fast (2). This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.

12.2.5 IC_DATA_CMD

- **Name:** I2C Rx/Tx Data Buffer and Command Register
- **Description:** I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes

  to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO.

The size of the register

changes as follows: Write:

- 11 bits when IC_EMPTYFIFO_HOLD_MASTER_EN=1
- 9 bits when

IC_EMPTYFIFO_HOLD_MA

STER_EN=0 Read:

- 12 bits when IC_FIRST_DATA_BYTE_STATUS = 1
- 8 bits when IC_FIRST_DATA_BYTE_STATUS = 0

Note: In order for the Apb_i2c to continue acknowledging reads, a read command should be written

for every byte that is to be received; otherwise the Apb_i2c will stop acknowledging.

- **Size:** 32 bits
- **Offset:** 0x10
- **Exists:** Always

Bits	Name	Memory Access	Description
31:12	RSVD_IC_DATA_CMD	R	IC_DATA_CMD Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always **Volatile:** true
11	FIRST_DATA_BYTE	R	Indicates the first data byte received after the address phase for receive transfer in Master receiver or Slave receiver mode. **Note:** In case of APB_DATA_WIDTH=8, ▪ The user has to perform two APB Reads to IC_DATA_CMD in order to get status on 11 bit. ▪ In order to read the 11 bit, the user has to perform the first data byte read [7:0] (offset 0x10) and then perform the second read[15:8](offset 0x11) in order to know the status of 11 bit (whether the data received in previous read is a first data byte or not). ▪ The 11th bit is an optional read field, user can ignore 2nd byte read [15:8] (offset 0x11) if not interested in FIRST_DATA_BYTE status. **Values:** ▪ 0x1 (ACTIVE): Non sequential data byte received ▪ 0x0 (INACTIVE): Sequential data byte received **Value After Reset:** 0x0

Bits	Name	Memory Access	Description
10	RESTART	W	This bit controls whether a RESTART is issued before the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. ■ 1: If IC_RESTART_EN is 1, a RESTART is issued before the data is sent/received (according to the value of CMD), regardless of whether or not the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. ■ 0: If IC_RESTART_EN is 1, a RESTART is issued only if the transfer direction is changing from the previous command; if IC_RESTART_EN is 0, a STOP followed by a START is issued instead. **Values:** ■ 0x1 (ENABLE): Issue RESTART before this command ■ 0x0 (DISABLE): Do not Issue RESTART before this command **Value After Reset:** 0x0

Bits	Name	Memory Access	Description
9	STOP	W	This bit controls whether a STOP is issued after the byte is sent or received. This bit is available only if IC_EMPTYFIFO_HOLD_MASTER_EN is configured to 1. ■ 1 - STOP is issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master immediately tries to start a new transfer by issuing a START and arbitrating for the bus. ■ 0 - STOP is not issued after this byte, regardless of whether or not the Tx FIFO is empty. If the Tx FIFO is not empty, the master continues the current transfer by sending/receiving data bytes according to the value of the CMD bit. If the Tx FIFO is empty, the master holds the SCL line low and stalls the bus until a new command is available in the Tx FIFO. **Values:** ■ 0x1 (ENABLE): Issue STOP after this command ■ 0x0 (DISABLE): Do not Issue STOP after this command **Value After Reset:** 0x0

Bits	Name	Memory Access	Description
8	CMD	W	This bit controls whether a read or a write is performed. This bit does not control the direction when the Apb_i2c acts as a slave. It controls only the direction when it acts as a master. When a command is entered in the TX FIFO, this bit distinguishes the write and read commands. In slave-receiver mode, this bit is a "don't care" because writes to this register are not required. In slave-transmitter mode, a "0" indicates that the data in IC_DATA_CMD is to be transmitted. When programming this bit, you should remember the following: ▪ Attempting to perform a read operation after a General Call command has been sent results in a TX_ABRT interrupt (bit 6 of the IC_RAW_INTR_STAT register), unless bit 11 (SPECIAL) in the IC_TAR register has been cleared. ▪ If a "1" is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs. **Values:** ▪ 0x1 (READ): Master Read Command ▪ 0x0 (WRITE): Master Write Command **Value After Reset:** 0x0 **Exists:** High-speed mode (3.4 Mb/s) Always **Volatile:** true

7:0	DAT	R/W	This register contains the data to be transmitted or received on the I2C bus. If you are writing to this register and want to perform a read, bits 7:0 (DAT) are ignored by the Apb_i2c. However, when you read this register, these bits return the value of data received on the Apb_i2c interface. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.6 IC_SS_SCL_HCNT

- **Name:** Standard Speed I2C Clock SCL High Count Register
- **Description:** Standard Speed I2C Clock SCL High Count Register
- **Size:** 32 bits
- **Offset:** 0x14

Bits	Name	Memory Access	Description
31:16	RSVD_IC_SS_SCL_HIGH_COU NT	R	IC_SS_SCL_HCNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	IC_SS_SCL_HCNT	* Varies	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for standard speed. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is derived through below formula: ((IC_ULTRA_FAST_MODE==1) ? 3 : ((IC_CLK_FREQ_OPTIMIZATION==1) ? 1 : 6)) Hardware prevents values less than this being written, and if attempted results in minimum valid value being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the Apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. **Note:** This register must not be programmed to a value higher than 65525, because Apb_i2c uses a 16-bit counter to flag an I2C bus idle condition when this counter reaches a value of IC_SS_SCL_HCNT + 10.

12.2.7 IC_UFM_SCL_HCNT

- **Name:** Ultra-Fast Speed I2C Clock SCL High Count Register
- **Description:** Ultra-Fast Speed I2C Clock SCL High Count Register
- **Size:** 32 bits
- **Offset:** 0x14

Bits	Name	Memory Access	Description
31:16	RSVD_IC_UFM_SCL_HCNT	R	IC_UFM_SCL_HCNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	IC_UFM_SCL_HCNT	* Varies	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for Ultra-Fast speed. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 3; hardware prevents values less than this being written, and if attempted results in 3 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the Apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed.

12.2.8 IC_SS_SCL_LCNT

- **Name:** Standard Speed I2C Clock SCL Low Count Register
- **Description:** Standard Speed I2C Clock SCL Low Count Register
- **Size:** 32 bits
- **Offset:** 0x18

Bits	Name	Memory Access	Description
31:16	RSVD_IC_SS_SCL_LOW_ COUN T	R	RSVD_IC_SS_SCL_LOW_COUNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	IC_SS_SCL_LCNT	* Varies	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for standard speed. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is derived through below formula: ((IC_ULTRA_FAST_MODE==1) ? 5 : ((IC_CLK_FREQ_OPTIMIZATION==1) ? 6 : 8)) Hardware prevents values less than this being written, and if attempted, results in minimum valid value being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of Apb_i2c. The lower byte must be programmed first, and then the upper byte is programmed.

12.2.9 IC_UFM_SCL_LCNT

- **Name:** Ultra-Fast Speed I2C Clock SCL Low Count Register
- **Description:** Ultra-Fast Speed I2C Clock SCL Low Count Register
- **Size:** 32 bits
- **Offset:** 0x18

Bits	Name	Memory Access	Description
31:16	RSVD_IC_UFM_SCL_LCNT	R	IC_UFM_SCL_LCNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	IC_UFM_SCL_LCNT	* Varies	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for Ultra-Fast speed. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 5; hardware prevents values less than this being written, and if attempted, results in 5 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of Apb_i2c. The lower byte must be programmed first, and then the upper byte is programmed.

12.2.10 IC_FS_SCL_HCNT

- **Name:** Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
- **Description:** Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
- **Size:** 32 bits
- **Offset:** 0x1c

Bits	Name	Memory Access	Description
31:16	RSVD_IC_FS_SCL_HCNT	R	IC_FS_SCL_HCNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	IC_FS_SCL_HCNT	* Varies	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high-period count for fast mode or fast mode plus. It is used in high-speed mode to send the Master Code and START BYTE or General CALL.. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is derived through below formula: ((IC_ULTRA_FAST_MODE==1) ? 3 : ((IC_CLK_FREQ_OPTIMIZATION==1) ? 1 : 6)) Hardware prevents values less than this being written, and if attempted results in minimum valid value being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the Apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed.

12.2.11 IC_UFM_TBUF_CNT

- **Name:** Ultra-Fast Speed mode TBuf Idle Count Register
- **Description:** Ultra-Fast Speed mode TBuf Idle Count Register
- **Size:** 32 bits
- **Offset:** 0x1c

Bits	Name	Memory Access	Description
31:16	RSVD_IC_UFM_TBUF_CNT	R	IC_UFM_TBUF_CNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	IC_UFM_TBUF_CNT	R/W	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the Bus-Free time between a STOP and STOP condition count for Ultra-Fast speed. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the Apb_i2c. The lower byte must be programmed first and then the upper byte is programmed. When the configuration parameter. **Note:** The Apb_i2c will add 9 ic_clks after tBuf time is expired to generate START on the Bus.

12.2.12 IC_FS_SCL_LCNT

- **Name:** Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
- **Description:** Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
- **Size:** 32 bits
- **Offset:** 0x20

Bits	Name	Memory Access	Description
31:16	RSVD_IC_FS_SCL_LCNT	R	IC_FS_SCL_LCNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	IC_FS_SCL_LCNT	* Varies	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. It is used in high- speed mode to send the Master Code and START BYTE or General CALL.. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is derived through below formula: ((IC_ULTRA_FAST_MODE==1) ? 5 : ((IC_CLK_FREQ_OPTIMIZATION==1) ? 6 : 8)) Hardware prevents values less than this being written, and if attempted results in minimum valid value being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the Apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than minimum valid value then the count value gets changed to minimum valid value.

12.2.13 IC_HS_SCL_HCNT

- **Name:** High Speed I2C Clock SCL High Count Register
- **Description:** High Speed I2C Clock SCL High Count Register
- **Size:** 32 bits
- **Offset:** 0x24

Bits	Name	Memory Access	Description
31:16	RSVD_IC_HS_SCL_HCNT	R	IC_HS_SCL_HCNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	IC_HS_SCL_HCNT	* Varies	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock high period count for high speed.refer to "IC_CLK Frequency Configuration". The SCL High time depends on the loading of the bus. For 100pF loading, the SCL High time is 60ns; for 400pF loading, the SCL High time is 120ns. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is derived through below formula: ((IC_ULTRA_FAST_MODE==1) ? 3 : ((IC_CLK_FREQ_OPTIMIZATION==1) ? 1 : 6)) Hardware prevents values less than this being written, and if attempted results in minimum valid value being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the Apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed.

12.2.14 IC_HS_SCL_LCNT

- **Name:** High Speed I2C Clock SCL Low Count Register
- **Description:** High Speed I2C Clock SCL Low Count Register
- **Size:** 32 bits
- **Offset:** 0x28

Bits	Name	Memory Access	Description
31:16	RSVD_IC_HS_SCL_LOW_CNT	R	IC_HS_SCL_LCNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	IC_HS_SCL_LCNT	* Varies	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for high speed.. The SCL low time depends on the loading of the bus. For 100pF loading, the SCL low time is 160ns; for 400pF loading, the SCL low time is 320ns. This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE != high. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is derived through below formula: ((IC_ULTRA_FAST_MODE==1) ? 5 : ((IC_CLK_FREQ_OPTIMIZATION==1) ? 6 : 8)) Hardware prevents values less than this being written, and if attempted results in minimum valid value being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the Apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than minimum valid value then the count value gets changed to minimum valid value.

12.2.15 IC_INTR_STAT

- **Name:** I2C Interrupt Status Register
- **Description:** I2C Interrupt Status Register.
  Each bit in this register has a corresponding mask bit in the IC_INTR_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the IC_RAW_INTR_STAT register.
- **Size:** 32 bits
- **Offset:** 0x2c
- **Exists:** Always

Bits	Name	Memory Access	Description
31:20	RSVD_IC_INTR_STAT	R	IC_INTR_STAT Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always
19	R_SLV_ADDR4_TAG	R	See IC_RAW_INTR_STAT for a detailed description of R_SLV_ADDR4_TAG bit. **Values:** ▪ 0x1 (DISABLED): SLV_ADDR4_TAG interrupt is active ▪ 0x0 (ENABLED): SLV_ADDR4_TAG interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
18	R_SLV_ADDR3_TAG	R	See IC_RAW_INTR_STAT for a detailed description of R_SLV_ADDR3_TAG bit. **Values:** ▪ 0x1 (DISABLED): SLV_ADDR3_TAG interrupt is active ▪ 0x0 (ENABLED): SLV_ADDR3_TAG interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
17	R_SLV_ADDR2_TAG	R	See IC_RAW_INTR_STAT for a detailed description of R_SLV_ADDR2_TAG bit. **Values:** ▪ 0x1 (DISABLED): SLV_ADDR2_TAG interrupt is active ▪ 0x0 (ENABLED): SLV_ADDR2_TAG interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

16	R_SLV_ADDR1_TAG	R	See IC_RAW_INTR_STAT for a detailed description of R_SLV_ADDR1_TAG bit. **Values:** ▪ 0x1 (DISABLED): SLV_ADDR1_TAG interrupt is active ▪ 0x0 (ENABLED): SLV_ADDR1_TAG interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
15	R_WR_REQ	R	See IC_RAW_INTR_STAT for a detailed description of R_WR_REQ bit. **Values:** ▪ 0x1 (ACTIVE): WR_REQ interrupt is active ▪ 0x0 (INACTIVE): WR_REQ interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
14	R_SCL_STUCK_AT_LOW	R	See IC_RAW_INTR_STAT for a detailed description of R_SCL_STUCK_AT_LOW bit. **Values:** ▪ 0x1 (ACTIVE): R_SCL_STUCK_AT_LOW interrupt is active ▪ 0x0 (INACTIVE): R_SCL_STUCK_AT_LOW interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
13	R_MASTER_ON_HOLD	R	See IC_RAW_INTR_STAT for a detailed description of R_MASTER_ON_HOLD bit. **Values:** ▪ 0x1 (ACTIVE): R_MASTER_ON_HOLD interrupt is active ▪ 0x0 (INACTIVE): R_MASTER_ON_HOLD interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always

12	R_RESTART_DET	R	See IC_RAW_INTR_STAT for a detailed description of R_RESTART_DET bit. **Values:** ▪ 0x1 (ACTIVE): R_RESTART_DET interrupt is active ▪ 0x0 (INACTIVE): R_RESTART_DET interrupt is inactive **Value After Reset:** 0x0
11	R_GEN_CALL	R	See IC_RAW_INTR_STAT for a detailed description of R_GEN_CALL bit. **Values:** ▪ 0x1 (ACTIVE): R_GEN_CALL interrupt is active ▪ 0x0 (INACTIVE): R_GEN_CALL interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
10	R_START_DET	R	See IC_RAW_INTR_STAT for a detailed description of R_START_DET bit. **Values:** ▪ 0x1 (ACTIVE): R_START_DET interrupt is active ▪ 0x0 (INACTIVE): R_START_DET interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
9	R_STOP_DET	R	See IC_RAW_INTR_STAT for a detailed description of R_STOP_DET bit. **Values:** ▪ 0x1 (ACTIVE): R_STOP_DET interrupt is active ▪ 0x0 (INACTIVE): R_STOP_DET interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

8	R_ACTIVITY	R	See IC_RAW_INTR_STAT for a detailed description of R_ACTIVITY bit. **Values:** ▪ 0x1 (ACTIVE): R_ACTIVITY interrupt is active ▪ 0x0 (INACTIVE): R_ACTIVITY interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
7	R_RX_DONE	R	See IC_RAW_INTR_STAT for a detailed description of R_RX_DONE bit. **Values:** ▪ 0x1 (ACTIVE): R_RX_DONE interrupt is active ▪ 0x0 (INACTIVE): R_RX_DONE interrupt is inactive **Value After Reset:** 0x0 **Exists:** IC_ULTRA_FAST_MODE==0 **Volatile:** true

Bits	Name	Memory Access	Description
6	R_TX_ABRT	R	See IC_RAW_INTR_STAT for a detailed description of R_TX_ABRT bit. **Values:** ▪ 0x1 (ACTIVE): R_TX_ABRT interrupt is active ▪ 0x0 (INACTIVE): R_TX_ABRT interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
5	R_RD_REQ	R	See IC_RAW_INTR_STAT for a detailed description of R_RD_REQ bit. **Values:** ▪ 0x1 (ACTIVE): R_RD_REQ interrupt is active ▪ 0x0 (INACTIVE): R_RD_REQ interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

4	R_TX_EMPTY	R	See IC_RAW_INTR_STAT for a detailed description of R_TX_EMPTY bit. **Values:** ▪ 0x1 (ACTIVE): R_TX_EMPTY interrupt is active ▪ 0x0 (INACTIVE): R_TX_EMPTY interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
3	R_TX_OVER	R	See IC_RAW_INTR_STAT for a detailed description of R_TX_OVER bit. **Values:** ▪ 0x1 (ACTIVE): R_TX_OVER interrupt is active ▪ 0x0 (INACTIVE): R_TX_OVER interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
2	R_RX_FULL	R	See IC_RAW_INTR_STAT for a detailed description of R_RX_FULL bit. **Values:** ▪ 0x1 (ACTIVE): R_RX_FULL interrupt is active ▪ 0x0 (INACTIVE): R_RX_FULL interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
1	R_RX_OVER	R	See IC_RAW_INTR_STAT for a detailed description of R_RX_OVER bit. **Values:** ▪ 0x1 (ACTIVE): R_RX_OVER interrupt is active ▪ 0x0 (INACTIVE): R_RX_OVER interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

0	R_RX_UNDER	R	See IC_RAW_INTR_STAT for a detailed description of R_RX_UNDER bit. **Values:** ▪ 0x1 (ACTIVE): RX_UNDER interrupt is active ▪ 0x0 (INACTIVE): RX_UNDER interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.16 IC_INTR_MASK

- **Name:** I2C Interrupt Mask Register
- **Description:** I2C Interrupt Mask Register.
  These bits mask their corresponding interrupt status bits. This register is active low; a value of 0 masks the interrupt, whereas a value of 1 unmasks the interrupt.
- **Size:** 32 bits
- **Offset:** 0x30
- **Exists:** Always

Bits	Name	Memory Access	Description
31:20	RSVD_IC_INTR_STAT	R	IC_INTR_STAT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always
19	M_SLV_ADDR4_TAG	R/W	This bit masks the R_SLV_ADDR4_TAG interrupt in IC_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): SLV_ADDR4_TAG interrupt is unmasked ▪ 0x0 (ENABLED): SLV_ADDR4_TAG interrupt is masked **Value After Reset:** 0x1

Bits	Name	Memory Access	Description
18	M_SLV_ADDR3_TAG	R/W	This bit masks the R_SLV_ADDR3_TAG interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): SLV_ADDR3_TAG interrupt is unmasked ■ 0x0 (ENABLED): SLV_ADDR3_TAG interrupt is masked **Value After Reset:** 0x1
17	M_SLV_ADDR2_TAG	R/W	This bit masks the R_SLV_ADDR2_TAG interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): SLV_ADDR2_TAG interrupt is unmasked ■ 0x0 (ENABLED): SLV_ADDR2_TAG interrupt is masked **Value After Reset:** 0x1
16	M_SLV_ADDR1_TAG	R/W	This bit masks the R_SLV_ADDR1_TAG interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): SLV_ADDR1_TAG interrupt is unmasked ■ 0x0 (ENABLED): SLV_ADDR1_TAG interrupt is masked **Value After Reset:** 0x1
15	M_WR_REQ	R/W	This bit masks the R_WR_REQ interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): WR_REQ interrupt is unmasked ■ 0x0 (ENABLED): WR_REQ interrupt is masked **Value After Reset:** 0x1

Bits	Name	Memory Access	Description
14	M_SCL_STUCK_AT_LOW	R/W	This bit masks the R_SCL_STUCK_AT_LOW interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): SCL_STUCK_AT_LOW interrupt is unmasked ■ 0x0 (ENABLED): SCL_STUCK_AT_LOW interrupt is masked
13	M_MASTER_ON_HOLD	R/W	This bit masks the R_MASTER_ON_HOLD interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): MASTER_ON_HOLD interrupt is unmasked ■ 0x0 (ENABLED): MASTER_ON_HOLD interrupt is masked **Value After Reset:** 0x0
12	M_RESTART_DET	R/W	This bit masks the R_RESTART_DET interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): RESTART_DET interrupt is unmasked ■ 0x0 (ENABLED): RESTART_DET interrupt is masked **Value After Reset:** 0x0
11	M_GEN_CALL	R/W	This bit masks the R_GEN_CALL interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): GEN_CALL interrupt is unmasked ■ 0x0 (ENABLED): GEN_CALL interrupt is masked **Value After Reset:** 0x1 **Exists:** Always

Bits	Name	Memory Access	Description
10	M_START_DET	R/W	This bit masks the R_START_DET interrupt in IC_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): START_DET interrupt is unmasked ▪ 0x0 (ENABLED): START_DET interrupt is masked **Value After Reset:** 0x0 **Exists:** Always
9	M_STOP_DET	R/W	This bit masks the R_STOP_DET interrupt in IC_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): STOP_DET interrupt is unmasked ▪ 0x0 (ENABLED): STOP_DET interrupt is masked **Value After Reset:** 0x0 **Exists:** Always
8	M_ACTIVITY	R/W	This bit masks the R_ACTIVITY interrupt in IC_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): ACTIVITY interrupt is unmasked ▪ 0x0 (ENABLED): ACTIVITY interrupt is masked **Value After Reset:** 0x0 **Exists:** Always
7	M_RX_DONE	R/W	This bit masks the R_RX_DONE interrupt in IC_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): RX_DONE interrupt is unmasked ▪ 0x0 (ENABLED): RX_DONE interrupt is masked **Value After Reset:** 0x1 **Exists:** IC_ULTRA_FAST_MODE == 0

6	M_TX_ABRT	R/W	This bit masks the R_TX_ABRT interrupt in IC_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): TX_ABORT interrupt is unmasked ▪ 0x0 (ENABLED): TX_ABORT interrupt is masked **Value After Reset:** 0x1 **Exists:** Always

Bits	Name	Memory Access	Description
5	M_RD_REQ	R/W	This bit masks the R_RD_REQ interrupt in IC_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): RD_REQ interrupt is unmasked ▪ 0x0 (ENABLED): RD_REQ interrupt is masked **Value After Reset:** 0x1
4	M_TX_EMPTY	R/W	This bit masks the R_TX_EMPTY interrupt in IC_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): TX_EMPTY interrupt is unmasked ▪ 0x0 (ENABLED): TX_EMPTY interrupt is masked **Value After Reset:** 0x1 **Exists:** Always
3	M_TX_OVER	R/W	This bit masks the R_TX_OVER interrupt in IC_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): TX_OVER interrupt is unmasked ▪ 0x0 (ENABLED): TX_OVER interrupt is masked **Value After Reset:** 0x1 **Exists:** Always

2	M_RX_FULL	R/W	This bit masks the R_RX_FULL interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): RX_FULL interrupt is unmasked ■ 0x0 (ENABLED): RX_FULL interrupt is masked **Value After Reset:** 0x1 **Exists:** Always
1	M_RX_OVER	R/W	This bit masks the R_RX_OVER interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): RX_OVER interrupt is unmasked ■ 0x0 (ENABLED): RX_OVER interrupt is masked **Value After Reset:** 0x1
Bits	**Name**	**Memory Access**	**Description**
0	M_RX_UNDER	R/W	This bit masks the R_RX_UNDER interrupt in IC_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): RX_UNDER interrupt is unmasked ■ 0x0 (ENABLED): RX_UNDER interrupt is masked **Value After Reset:** 0x1

12.2.17 IC_RAW_INTR_STAT

- **Name:** I2C Raw Interrupt Status Register
- **Description:** I2C Raw Interrupt Status Register.
  Unlike the IC_INTR_STAT register, these bits are not masked so they always show the true status of the Apb_i2c.
- **Size:** 32 bits
- **Offset:** 0x34
- **Exists:** Always

Bits	Name	Memory Access	Description
31:20	RSVD_IC_RAW_INTR_STAT	R	IC_RAW_INTR_STAT Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
19	SLV_ADDR4_TAG	R	This bit is set to 1 when Apb_i2c is acting as slave and another I2C master is attempting to write/read I2C transfers with the Slave address IC_SAR4. This can be used in conjunction with the WR_REQ/RD_REQto determine which slave address has been addressed. **Values:** ▪ 0x1 (ACTIVE): SLV_ADDR4_TAG interrupt is active ▪ 0x0 (INACTIVE): SLV_ADDR4_TAG interrupt is inactive. **Value After Reset:** 0x0 **Volatile:** true
18	SLV_ADDR3_TAG	R	This bit is set to 1 when Apb_i2c is acting as slave and another I2C master is attempting to write/read I2C transfers with the Slave address IC_SAR3. This can be used in conjunction with the WR_REQ/RD_REQto determine which slave address has been addressed. **Values:** ▪ 0x1 (ACTIVE): SLV_ADDR3_TAG interrupt is active ▪ 0x0 (INACTIVE): SLV_ADDR3_TAG interrupt is inactive. **Value After Reset:** 0x0 **Volatile:** true

17	SLV_ADDR2_TAG	R	This bit is set to 1 when Apb_i2c is acting as slave and another I2C master is attempting to write/read I2C transfers with the Slave address IC_SAR2. This can be used in conjunction with the WR_REQ/RD_REQto determine which slave address has been addressed. **Values:** ▪ 0x1 (ACTIVE): SLV_ADDR2_TAG interrupt is active ▪ 0x0 (INACTIVE): SLV_ADDR2_TAG interrupt is inactive. **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
16	SLV_ADDR1_TAG	R	This bit is set to 1 when Apb_i2c is acting as slave and another I2C master is attempting to write/read I2C transfers with the Slave address IC_SAR. This can be used in conjunction with the WR_REQ/RD_REQto determine which slave address has been addressed. **Values:** ▪ 0x1 (ACTIVE): SLV_ADDR1_TAG interrupt is active ▪ 0x0 (INACTIVE): SLV_ADDR1_TAG interrupt is inactive. **Value After Reset:** 0x0 **Volatile:** true
15	WR_REQ	R	This bit is set to 1 when Apb_i2c is acting as slave and another I2C master is attempting to werite data into Apb_i2c. This interrupt provides indication that, the write data from another I2C master is going to follow and this interrupt can be used in conjunction of SLV_ADDRx_TAG(where x = 1 ... IC_NUM_SARS) to determine for which Slave address the I2C transfer is received. **Values:** ▪ 0x1 (ACTIVE): WR_REQ interrupt is active ▪ 0x0 (INACTIVE): WR_REQ interrupt is inactive. **Value After Reset:** 0x0 **Volatile:** true

14	SCL_STUCK_AT_LOW	R	Indicates whether the SCL Line is stuck at low for the IC_SCL_STUCK_LOW_TIMEOUT number of ic_clk periods. Enabled only when IC_BUS_CLEAR_FEATURE=1 and IC_ULTRA_FAST_MODE=0. **Values:** ▪ 0x1 (ACTIVE): SCL_STUCK_AT_LOW interrupt is active ▪ 0x0 (INACTIVE): SCL_STUCK_AT_LOW interrupt is inactive. **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
13	MASTER_ON_HOLD	R	Indicates whether master is holding the bus and TX FIFO is empty. Enabled only when I2C_DYNAMIC_TAR_UPDATE=1 and IC_EMPTYFIFO_HOLD_MASTER_EN=1. **Values:** ▪ 0x1 (ACTIVE): MASTER_ON_HOLD interrupt is active ▪ 0x0 (INACTIVE): MASTER_ON_HOLD interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always
12	RESTART_DET	R	Indicates whether a RESTART condition has occurred on the I2C interface when Apb_i2c is operating in Slave mode and the slave is being addressed. Enabled only when IC_SLV_RESTART_DET_EN=1. **Note:** However, in high-speed mode or during a START BYTE transfer, the RESTART comes before the address field as per the I2C protocol. In this case, the slave is not the addressed slave when the RESTART is issued, therefore Apb_i2c does not generate the RESTART_DET interrupt. **Values:** ▪ 0x1 (ACTIVE): RESTART_DET interrupt is active ▪ 0x0 (INACTIVE): RESTART_DET interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always

11	GEN_CALL	R	Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling Apb_i2c or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. Apb_i2c stores the received data in the Rx buffer. **Values:** ▪ 0x1 (ACTIVE): GEN_CALL interrupt is active ▪ 0x0 (INACTIVE): GEN_CALL interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
10	START_DET	R	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether Apb_i2c is operating in slave or master mode. **Values:** ▪ 0x1 (ACTIVE): START_DET interrupt is active ▪ 0x0 (INACTIVE): START_DET interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always

9	STOP_DET	R	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether Apb_i2c is operating in slave or master mode. **In Slave Mode:** ▪ If IC_CON[7]=1'b1 (STOP_DET_IFADDRESSED), the STOP_DET interrupt will be issued only if slave is addressed. **Note:** During a general call address, this slave does not issue a STOP_DET interrupt if STOP_DET_IF_ADDRESSED=1'b1, even if the slave responds to the general call address by generating ACK. The STOP_DET interrupt is generated only when the transmitted address matches the slave address (SAR). ▪ If IC_CON[7]=1'b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is issued irrespective of whether it is being addressed. **In Master Mode:** ▪ If IC_CON[10]=1'b1 (STOP_DET_IF_MASTER_ACTIVE),the STOP_DET interrupt will be issued only if Master is active. ▪ If IC_CON[10]=1'b0 (STOP_DET_IFADDRESSED),the STOP_DET interrupt will be issued irrespective of whether master is active or not. **Values:** ▪ 0x1 (ACTIVE): STOP_DET interrupt is active ▪ 0x0 (INACTIVE): STOP_DET interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
8	ACTIVITY	R	This bit captures Apb_i2c activity and stays set until it is cleared. There are four ways to clear it: ▪ Disabling the Apb_i2c ▪ Reading the IC_CLR_ACTIVITY register ▪ Reading the IC_CLR_INTR register ▪ System reset Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the Apb_i2c module is idle, this bit remains set until cleared, indicating that there was activity on the bus. **Values:** ▪ 0x1 (ACTIVE): RAW_INTR_ACTIVITY interrupt is active ▪ 0x0 (INACTIVE): RAW_INTR_ACTIVITY interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always
7	RX_DONE	R	When the Apb_i2c is acting as a slave-transmitter, this bit is set to 1 if the master does not acknowledge a transmitted byte. This occurs on the last byte of the transmission, indicating that the transmission is done. **Values:** ▪ 0x1 (ACTIVE): RX_DONE interrupt is active ▪ 0x0 (INACTIVE): RX_DONE interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
6	TX_ABRT	R	This bit indicates if Apb_i2c, as an I2C transmitter, is unable to complete the intended actions on the contents of the transmit FIFO. This situation can occur both as an I2C master or an I2C slave, and is referred to as a 'transmit abort'. When this bit is set to 1, the IC_TX_ABRT_SOURCE register indicates the reason why the transmit abort takes places. **Note:** The Apb_i2c flushes/resets/empties only the TX_FIFO whenever there is a transmit abort caused by any of the events tracked by the IC_TX_ABRT_SOURCE register. The Tx FIFO remains in this flushed state until the register IC_CLR_TX_ABRT is read. Once this read is performed, the Tx FIFO is then ready to accept more data bytes from the APB interface. **Values:** ■ 0x1 (ACTIVE): TX_ABRT interrupt is active ■ 0x0 (INACTIVE): TX_ABRT interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
5	RD_REQ	R	This bit is set to 1 when Apb_i2c is acting as a slave and another I2C master is attempting to read data from Apb_i2c. The Apb_i2c holds the I2C bus in a wait state (SCL=0) until this interrupt is serviced, which means that the slave has been addressed by a remote master that is asking for data to be transferred. The processor must respond to this interrupt and then write the requested data to the IC_DATA_CMD register. This bit is set to 0 just after the processor reads the IC_CLR_RD_REQ register. **Values:** ■ 0x1 (ACTIVE): RD_REQ interrupt is active ■ 0x0 (INACTIVE): RD_REQ interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
4	TX_EMPTY	R	The behavior of the TX_EMPTY interrupt status differs based on the TX_EMPTY_CTRL selection in the IC_CON register. ▪ When TX_EMPTY_CTRL = 0: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register. ▪ When TX_EMPTY_CTRL = 1: This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register and the transmission of the address/data from the internal shift register for the most recently popped command is completed. It is automatically cleared by hardware when the buffer level goes above the threshold. When IC_ENABLE[0] is set to 0, the TX FIFO is flushed and held in reset. There the TX FIFO looks like it has no data within it, so this bit is set to 1, provided there is activity in the master or slave state machines. When there is no longer any activity, then with ic_en=0, this bit is set to 0. **Values:** ▪ 0x1 (ACTIVE): TX_EMPTY interrupt is active ▪ 0x0 (INACTIVE): TX_EMPTY interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always

3	TX_OVER	R	Set during transmit if the transmit buffer is filled to IC_TX_BUFFER_DEPTH and the processor attempts to issue another I2C command by writing to the IC_DATA_CMD register. When the module is disabled, this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared. **Values:** ▪ 0x1 (ACTIVE): TX_OVER interrupt is active ▪ 0x0 (INACTIVE): TX_OVER interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
2	RX_FULL	R	Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the IC_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues. **Values:** ▪ 0x1 (ACTIVE): RX_FULL interrupt is active ▪ 0x0 (INACTIVE): RX_FULL interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

1	RX_OVER	R	Set if the receive buffer is completely filled to IC_RX_BUFFER_DEPTH and an additional byte is received from an external I2C device. The Apb_i2c acknowledges this, but any data bytes received after the FIFO is full are lost. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared. **Note:** If the configuration parameter IC_RX_FULL_HLD_BUS_EN is enabled and bit 9 of the IC_CON register (RX_FIFO_FULL_HLD_CTRL) is programmed to HIGH, then the RX_OVER interrupt never occurs, because the Rx FIFO never overflows. **Values:** ■ 0x1 (ACTIVE): RX_OVER interrupt is active ■ 0x0 (INACTIVE): RX_OVER interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
0	RX_UNDER	R	Set if the processor attempts to read the receive buffer when it is empty by reading from the IC_DATA_CMD register. If the module is disabled (IC_ENABLE[0]=0), this bit keeps its level until the master or slave state machines go into idle, and when ic_en goes to 0, this interrupt is cleared. **Values:** ■ 0x1 (ACTIVE): RX_UNDER interrupt is active ■ 0x0 (INACTIVE): RX_UNDER interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always

12.2.18 IC_RX_TL

- **Name:** I2C Receive FIFO Threshold Register
- **Description:** I2C Receive FIFO Threshold Register

- **Size:** 32 bits
- **Offset:** 0x38
- **Exists:** Always

Bits	Name	Memory Access	Description
31:8	RSVD_IC_RX_TL	R	IC_RX_TL Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always
7:0	RX_TL	R/W	Receive FIFO Threshold Level. Controls the level of entries (or above) that triggers the RX_FULL interrupt (bit 2 in IC_RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that hardware does not allow this value to be set to a value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 1 entry, and a value of 255 sets the threshold for 256 entries. **Exists:** Always

12.2.19 IC_TX_TL

- **Name:** I2C Transmit FIFO Threshold Register
- **Description:** I2C Transmit FIFO Threshold Register
- **Size:** 32 bits
- **Offset:** 0x3c
- **Exists:** Always

Bits	Name	Memory Access	Description
31:8	RSVD_IC_TX_TL	R	IC_TX_TL Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

7:0	TX_TL	R/W	Transmit FIFO Threshold Level. Controls the level of entries (or below) that trigger the TX_EMPTY interrupt (bit 4 in IC_RAW_INTR_STAT register). The valid range is 0-255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries. **Exists:** Always

12.2.20 IC_CLR_INTR

- **Name:** Clear Combined and Individual Interrupt Register
- **Description:** Clear Combined and Individual Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x40
- **Exists:** Always

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_INTR	R	CLR_INTR Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always
0	CLR_INTR	R	Read this register to clear the combined interrupt, all individual interrupts, and the IC_TX_ABRT_SOURCE register. This bit does not clear hardware clearable interrupts but software clearable interrupts. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE. **Value After** **Reset:** 0x0 **Exists:** Always

12.2.21 IC_CLR_RX_UNDER

- **Name:** Clear RX_UNDER Interrupt Register
- **Description:** Clear RX_UNDER Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x44
- **Exists:** Always

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_RX_UNDER	R	IC_CLR_RX_UNDER Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always
0	CLR_RX_UNDER	R	Read this register to clear the RX_UNDER interrupt (bit 0) of the IC_RAW_INTR_STAT register. **Value After** **Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.22 IC_CLR_RX_OVER

- **Name:** Clear RX_OVER Interrupt Register
- **Description:** Clear RX_OVER Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x48
- **Exists:** Always

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_RX_OVER	R	IC_CLR_RX_OVER Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
0	CLR_RX_OVER	R	Read this register to clear the RX_OVER interrupt (bit 1) of the IC_RAW_INTR_STAT register. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.23 IC_CLR_TX_OVER

- **Name:** Clear TX_OVER Interrupt Register
- **Description:** Clear TX_OVER Interrupt Register

- **Size:** 32 bits
- **Offset:** 0x4c
- **Exists:** Always

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_TX_OVER	R	IC_CLR_TX_OVER Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always
0	CLR_TX_OVER	R	Read this register to clear the TX_OVER interrupt (bit 3) of the IC_RAW_INTR_STAT register. **Value After** **Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.24 IC_CLR_RD_REQ

- **Name:** Clear RD_REQ Interrupt Register
- **Description:** Clear RD_REQ Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x50

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_RD_REQ	R	IC_CLR_RD_REQ Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always
0	CLR_RD_REQ	R	Read this register to clear the RD_REQ interrupt (bit 5) of the IC_RAW_INTR_STAT register. **Value After** **Reset:** 0x0 **Exists:** Always

12.2.25 IC_CLR_TX_ABRT

- **Name:** Clear TX_ABRT Interrupt Register
- **Description:** Clear TX_ABRT Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x54

- **Exists:** Always

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_TX_ABRT	R	IC_CLR_TX_ABRT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:**
0	CLR_TX_ABRT	R	Read this register to clear the TX_ABRT interrupt (bit 6) of the IC_RAW_INTR_STAT register, and the IC_TX_ABRT_SOURCE register. This also releases the TX FIFO from the flushed/reset state, allowing more writes to the TX FIFO. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.26 IC_CLR_RX_DONE

- **Name:** Clear RX_DONE Interrupt Register
- **Description:** Clear RX_DONE Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x58

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_RX_DONE	R	IC_CLR_RX_DONE Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
0	CLR_RX_DONE	R	Read this register to clear the RX_DONE interrupt (bit 7) of the IC_RAW_INTR_STAT register. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.27 IC_CLR_ACTIVITY

- **Name:** Clear ACTIVITY Interrupt Register
- **Description:** Clear ACTIVITY Interrupt Register
- **Size:** 32 bits

- **Offset:** 0x5c
- **Exists:** Always

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_ACTIVITY	R	IC_CLR_ACTIVITY Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
0	CLR_ACTIVITY	R	Reading this register clears the ACTIVITY interrupt if the I2C is not active anymore. If the I2C module is still active on the bus, the ACTIVITY interrupt bit continues to be set. It is automatically cleared by hardware if the module is disabled and if there is no further activity on the bus. The value read from this register to get status of the ACTIVITY interrupt (bit 8) of the IC_RAW_INTR_STAT register. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.28 IC_CLR_STOP_DET

- **Name:** Clear STOP_DET Interrupt Register
- **Description:** Clear STOP_DET Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x60
- **Exists:** Always

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_STOP_DET	R	IC_CLR_STOP_DET Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
0	CLR_STOP_DET	R	Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.29 IC_CLR_START_DET

- **Name:** Clear START_DET Interrupt Register
- **Description:** Clear START_DET Interrupt Register

- **Size:** 32 bits
- **Offset:** 0x64
- **Exists:** Always

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_START_DET	R	IC_CLR_START_DET Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
0	CLR_START_DET	R	Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.30 IC_CLR_GEN_CALL

- **Name:** Clear GEN_CALL Interrupt Register
- **Description:** Clear GEN_CALL Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x68
- **Exists:** Always

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_GEN_CALL	R	IC_CLR_GEN_CALL Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always **Volatile:** true
0	CLR_GEN_CALL	R	Read this register to clear the GEN_CALL interrupt (bit 11) of IC_RAW_INTR_STAT register. **Value After** **Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.31 IC_ENABLE

- **Name:** I2C ENABLE Register

- **Description:** I2C Enable Register
- **Size:** 32 bits
- **Offset:** 0x6c
- **Exists:** Always

Bits	Name	Memory Access	Description
31:23	RSVD_IC_ENABLE_2	R	IC_ENABLE Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
22	IC_SAR4_EN	R/W	Controls whether the Apb_i2c SAR4 is enabled. If IC_SAR4_EN bit of the IC_ENABLE register is set to 1, then the Apb_i2c will acknowledge the address if it corresponds to IC_SAR4 and completes the I2C transfers as intended. Otherwise, the I2C transfer address will be Nacked. **Note:** This field needs to be updated only when IC_ENABLE[0] is 0. **Values:** ■ 0x1 (SLAVE_ENABLED): Slave address IC_SAR4 is enabled ■ 0x0 (SLAVE_DISABLED): Slave address IC_SAR4 is disabled **Value After Reset:** 0x0

21	IC_SAR3_EN	R/W	Controls whether the Apb_i2c SAR3 is enabled. If IC_SAR3_EN bit of the IC_ENABLE register is set to 1, then the Apb_i2c will acknowledge the address if it corresponds to IC_SAR3 and completes the I2C transfers as intended. Otherwise, the I2C transfer address will be Nacked. **Note:** This field needs to be updated only when IC_ENABLE[0] is 0. **Values:** ■ 0x1 (SLAVE_ENABLED): Slave address IC_SAR3 is enabled ■ 0x0 (SLAVE_DISABLED): Slave address IC_SAR3 is disabled **Value After Reset:** 0x0

Bits	Name	Memory Access	Description
20	IC_SAR2_EN	R/W	Controls whether the Apb_i2c SAR2 is enabled. If IC_SAR2_EN bit of the IC_ENABLE register is set to 1, then the Apb_i2c will acknowledge the address if it corresponds to IC_SAR2 and completes the I2C transfers as intended. Otherwise, the I2C transfer address will be Nacked. **Note:** This field needs to be updated only when IC_ENABLE[0] is 0. **Values:** ■ 0x1 (SLAVE_ENABLED): Slave address IC_SAR2 is enabled ■ 0x0 (SLAVE_DISABLED): Slave address IC_SAR2 is disabled **Value After Reset:** 0x0

19	IC_SAR_EN	R/W	Controls whether the Apb_i2c SAR is enabled. If IC_SAR_EN bit of the IC_ENABLE register is set to 1, then the Apb_i2c will acknowledge the address if it corresponds to IC_SAR and completes the I2C transfers as intended. Otherwise, the I2C transfer address will be Nacked. **Note:** This field needs to be updated only when IC_ENABLE[0] is 0. **Values:** ▪ 0x1 (SLAVE_ENABLED): Slave address IC_SAR is enabled ▪ 0x0 (SLAVE_DISABLED): Slave address IC_SAR is disabled **Value After Reset:** 0x1

Bits	Name	Memory Access	Description
18	SMBUS_ALERT_EN	R/W	The SMBUS_ALERT_CTRL register bit is used to control assertion of SMBALERT signal. - 1: Assert SMBALERT signal This register bit is auto-cleared after detection of Acknowledgement from master for Alert Response address. **Values:** ▪ 0x1 (ALERT_ENABLED): Slave initates the Alert signal to indicate SMBus Host ▪ 0x0 (SUSPEND_DISABLED): Slave will not initates the Alert signal to indicate SMBus Host. **Value After Reset:** 0x0
17	SMBUS_SUSPEND_EN	R/W	The SMBUS_SUSPEND_EN register bit is used to control assertion and de-assertion of SMBSUS signal. ▪ 0: De-assert SMBSUS signal ▪ 1: Assert SMBSUS signal **Values:** ▪ 0x1 (ENABLED): Host/Master initates the SMBUS system to enter Suspend Mode. ▪ 0x0 (DISABLED): Host/Master will not initates the SMBUS system to enter Suspend Mode. **Value After Reset:** 0x0

16	SMBUS_CLK_RESET	R/W	This bit is used in SMBus Host mode to initiate the SMBus Master Clock Reset. This bit should be enabled only when Master is in idle. Whenever this bit is enabled, the SMBCLK is held low for the IC_SCL_STUCK_TIMEOUT ic_clk cycles to reset the SMBus slave devices. **Values:** ▪ 0x1 (ENABLED): Master initates the SMBUS Clock Reset Mechanism. ▪ 0x0 (DISABLED): Master will not initates SMBUS Clock Reset Mechanism. **Value After Reset:** 0x0
15:4	RSVD_IC_ENABLE_1	R	RSVD_IC_ENABLE_1 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
3	SDA_STUCK_RECOVERY_ENABLE	R/W	If SDA is stuck at low indicated through the TX_ABORT interrupt (IC_TX_ABRT_SOURCE[17]), then this bit is used as a control knob to initiate the SDA Recovery Mechanism (that is, send at most 9 SCL clocks and STOP to release the SDA line) and then this bit gets auto clear **Values:** ▪ 0x1 (SDA_STUCK_RECOVERY_ENABLED): Master initiates the SDA stuck at low recovery mechanism. ▪ 0x0 (SDA_STUCK_RECOVERY_DISABLED): Master disabled the SDA stuck at low recovery mechanism. **Value After Reset:** 0x0 **Exists:** IC_BUS_CLEAR_FEATURE==1

2	TX_CMD_BLOCK	R/W	In Master mode: ▪ 1'b1: Blocks the transmission of data on I2C bus even if Tx FIFO has data to transmit. ▪ 1'b0: The transmission of data starts on I2C bus automatically, as soon as the first data is available in the Tx FIFO. **Note:** To block the execution of Master commands, set the TX_CMD_BLOCK bit only when Tx FIFO is empty (IC_STATUS[2]==1) and Master is in Idle state (IC_STATUS[5] == 0). Any further commands put in the Tx FIFO are not executed until TX_CMD_BLOCK bit is unset. **Values:** ▪ 0x1 (BLOCKED): Tx Command execution blocked ▪ 0x0 (NOT_BLOCKED): Tx Command execution not blocked

Bits	Name	Memory Access	Description
1	ABORT	R/W	When set, the controller initiates the transfer abort. ▪ 0: ABORT not initiated or ABORT done ▪ 1: ABORT operation in progress The software can abort the I2C transfer in master mode by setting this bit. The software can set this bit only when ENABLE is already set; otherwise, the controller ignores any write to ABORT bit. The software cannot clear the ABORT bit once set. In response to an ABORT, the controller issues a STOP and flushes the Tx FIFO after completing the current transfer, then sets the TX_ABORT interrupt after the abort operation. The ABORT bit is cleared automatically after the abort operation. For a detailed description on how to abort I2C transfers, refer to "Aborting I2C Transfers". **Values:** ▪ 0x1 (ENABLED): ABORT operation in progress ▪ 0x0 (DISABLE): ABORT operation not in progress **Value After Reset:** 0x0

Bits	Name	Memory Access	Description
0	ENABLE	R/W	Controls whether the Apb_i2c is enabled. ■ 0: Disables Apb_i2c (TX and RX FIFOs are held in an erased state) ■ 1: Enables Apb_i2c Software can disable Apb_i2c while it is active. However, it is important that care be taken to ensure that Apb_i2c is disabled properly. A recommended procedure is described in "Disabling Apb_i2c". When Apb_i2c is disabled, the following occurs: ■ The TX FIFO and RX FIFO get flushed. ■ Status bits in the IC_INTR_STAT register are still active until Apb_i2c goes into IDLE state. If the module is transmitting, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. If the module is receiving, the Apb_i2c stops the current transfer at the end of the current transfer and does not acknowledge the transfer. In systems with asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the Apb_i2c. For a detailed description on how to disable Apb_i2c, refer to "Disabling Apb_i2c" **Values:** ■ 0x1 (ENABLED): I2C is enabled ■ 0x0 (DISABLED): I2C is disabled **Value After Reset:** 0x0 **Exists:** Always

12.2.32 IC_STATUS

- **Name:** I2C STATUS Register
- **Description:** I2C Status Register.
  This is a read-only register used to indicate the current transfer status and FIFO status. The status

  register may be read at any time. None of the bits in this register request an interrupt.

  When the I2C is disabled by writing 0 in bit 0 of the IC_ENABLE register:

  - Bits 1 and 2 are set to 1
  - Bits 3 and 10 are set to 0

  When the master or slave state machines goes to idle and ic_en=0:

  - Bits 5 and 6 are set to 0

- **Size:** 32 bits
- **Offset:** 0x70
- **Exists:** Always

Bits	Name	Memory Access	Description
31:27	RSVD_IC_STATUS_2	R	IC_STATUS Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always
26	SMBUS_SLAVE_ADDR4_RESOL VED	R	This bit indicates whether the slave address 4 (IC_SAR4) is resolved by the ARP Master. **Values:** ▪ 0x1 (ACTIVE): SMBUS Slave Address 4 is Resolved. ▪ 0x0 (INACTIVE): SMBUS Slave Address 4 is not Resolved. **Value After Reset:** 0x0 **Volatile:** true
25	SMBUS_SLAVE_ADDR4_VALID	R	This bit indicates whether the slave address 4 (IC_SAR4) is valid or not. **Values:** ▪ 0x1 (ACTIVE): SMBUS Slave Address 4 is Valid. ▪ 0x0 (INACTIVE): SMBUS Slave Address 4 is not valid. **Volatile:** true
24	SMBUS_SLAVE_ADDR3_RESOL VED	R	This bit indicates whether the slave address 3 (IC_SAR3) is resolved by the ARP Master. **Values:** ▪ 0x1 (ACTIVE): SMBUS Slave Address 3 is Resolved. ▪ 0x0 (INACTIVE): SMBUS Slave Address 3 is not Resolved. **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
23	SMBUS_SLAVE_ADDR3_VALID	R	This bit indicates whether the slave address 3 (IC_SAR3) is valid or not. **Values:** ■ 0x1 (ACTIVE): SMBUS Slave Address 3 is Valid. ■ 0x0 (INACTIVE): SMBUS Slave Address 3 is not valid. **Volatile:** true
22	SMBUS_SLAVE_ADDR2_RESOL VED	R	This bit indicates whether the slave address 2 (IC_SAR2) is resolved by the ARP Master. **Values:** ■ 0x1 (ACTIVE): SMBUS Slave Address 2 is Resolved. ■ 0x0 (INACTIVE): SMBUS Slave Address 2 is not Resolved. **Value After Reset:** 0x0 **Volatile:** true
21	SMBUS_SLAVE_ADDR2_VALID	R	This bit indicates whether the slave address 2 (IC_SAR2) is valid or not. **Values:** ■ 0x1 (ACTIVE): SMBUS Slave Address 2 is Valid. ■ 0x0 (INACTIVE): SMBUS Slave Address 2 is not valid. **Volatile:** true

20	SMBUS_ALERT_STATUS	R	This bit indicates the status of the SMBus Alert signal (ic_smbalert_in_n). This signal is asserted when the SMBus Alert signal is asserted by the SMBus Device. **Values:** ▪ 0x1 (ACTIVE): SMBUS Alert is asserted. ▪ 0x0 (INACTIVE): SMBUS Alert is not asserted. **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
19	SMBUS_SUSPEND_STATUS	R	This bit indicates the status of the SMBus Suspend signal (ic_smbsus_in_n). This signal is asserted when the SMBus Suspend signal is asserted by the SMBus Host. **Values:** ▪ 0x1 (ACTIVE): SMBUS System is in Suspended mode. ▪ 0x0 (INACTIVE): SMBUS System is not in Suspended mode. **Value After Reset:** 0x0
18	SMBUS_SLAVE_ADDR_RESOLV ED	R	This bit indicates whether the slave address (ic_sar) is resolved by the ARP Master. **Values:** ▪ 0x1 (ACTIVE): SMBUS Slave Address is Resolved. ▪ 0x0 (INACTIVE): SMBUS Slave Address is not Resolved. **Value After Reset:** 0x0 **Volatile:** true
17	SMBUS_SLAVE_ADDR_VALID	R	This bit indicates whether the slave address (ic_sar) is valid or not. **Values:** ▪ 0x1 (ACTIVE): SMBUS Slave Address is Valid. ▪ 0x0 (INACTIVE): SMBUS Slave Address is not valid. **Volatile:** true

16	SMBUS_QUICK_CMD_BIT	R	This bit indicates the R/W bit of the Quick command received. This bit will be cleared after the user has read this bit. **Values:** ▪ 0x1 (ACTIVE): SMBUS QUICK CMD Read/write is set to 1. ▪ 0x0 (INACTIVE): SMBUS QUICK CMD Read/write is set to 0. **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
15:13	RSVD_IC_STATUS_1	R	RSVD_IC_STATUS_1 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
12	SLV_ISO_SAR_DATA_CLK_STR ETCH	R	This bit indicates the Bus hold (clock stretching) in Slave mode with Multiple SAR feature - for isolating the write transfer related data corresponding to one slave address with another slave address. **Values:** ▪ 0x1 (ACTIVE): Slave holds the bus to isolate current RX FIFO data from next transfer related to another slave address. ▪ 0x0 (INACTIVE): Slave is not holding the bus to isolate current RX FIFO data from next transfer related to another slave address. **Value After Reset:** 0x0 **Volatile:** true

11	SDA_STUCK_NOT_RECOVERE D	R	This bit indicates that SDA stuck at low is not recovered after the recovery mechanism. In Slave mode, this register bit is not applicable. **Values:** ▪ 0x1 (ACTIVE): SDA Stuck at low is not recovered after recovery mechanism. ▪ 0x0 (INACTIVE): SDA Stuck at low is recovered after recovery mechanism. **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
10	SLV_HOLD_RX_FIFO_FULL	R	This bit indicates the BUS Hold in Slave mode due to Rx FIFO is Full and an additional byte has been received (This kind of Bus hold is applicable if IC_RX_FULL_HLD_BUS_EN is set to 1 and bit 9 of the IC_CON register (RX_FIFO_FULL_HLD_CTRL) is programmed to HIGH). **Values:** ▪ 0x1 (ACTIVE): Slave holds the bus due to Rx FIFO is full ▪ 0x0 (INACTIVE): Slave is not holding the bus or Bus hold is not due to Rx FIFO is full **Value After Reset:** 0x0 **Volatile:** true
9	SLV_HOLD_TX_FIFO_EMPTY	R	This bit indicates the BUS Hold in Slave mode for the Read request when the Tx FIFO is empty. The Bus is in hold until the Tx FIFO has data to Transmit for the read request. **Values:** ▪ 0x1 (ACTIVE): Slave holds the bus due to Tx FIFO is empty ▪ 0x0 (INACTIVE): Slave is not holding the bus or Bus hold is not due to Tx FIFO is empty **Value After Reset:** 0x0 **Volatile:** true

8	MST_HOLD_RX_FIFO_FULL	R	This bit indicates the BUS Hold in Master mode due to Rx FIFO is Full and additional byte has been received (This kind of Bus hold is applicable if IC_RX_FULL_HLD_BUS_EN is set to 1 and bit 9 of the IC_CON register (RX_FIFO_FULL_HLD_CTRL) is programmed to HIGH). **Values:** ▪ 0x1 (ACTIVE): Master holds the bus due to Rx FIFO is full ▪ 0x0 (INACTIVE): Master is not holding the bus or Bus hold is not due to Rx FIFO is full **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
7	MST_HOLD_TX_FIFO_EMPTY	R	If the IC_EMPTYFIFO_HOLD_MASTER_EN parameter is set to 1, the Apb_i2c master stalls the write transfer when Tx FIFO is empty, and the the last byte does not have the Stop bit set. This bit indicates the BUS hold when the master holds the bus because of the Tx FIFO being empty, and the the previous transferred command does not have the Stop bit set. (This kind of Bus hold is applicable if IC_EMPTYFIFO_HOLD_MASTER_EN is set to 1). **Values:** ▪ 0x1 (ACTIVE): Master holds the bus due to Tx FIFO is empty ▪ 0x0 (INACTIVE): Master is not holding the bus or Bus hold is not due to Tx FIFO is empty **Value After Reset:** 0x0 **Volatile:** true
6	SLV_ACTIVITY	R	Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set. **Values:** ▪ 0x1 (ACTIVE): Slave not idle ▪ 0x0 (IDLE): Slave is idle **Value After Reset:** 0x0 **Exists:** Always

5	MST_ACTIVITY	R	Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set. **Note:** IC_STATUS[0]-that is, ACTIVITY bit-is the OR of SLV_ACTIVITY and MST_ACTIVITY bits. **Values:** ■ 0x1 (ACTIVE): Master not idle ■ 0x0 (IDLE): Master is idle **Value After** **Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
4	RFF	R	Receive FIFO Completely Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. **Values:** ■ 0x1 (FULL): Rx FIFO is full ■ 0x0 (NOT_FULL): Rx FIFO not full **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
3	RFNE	R	Receive FIFO Not Empty. This bit is set when the receive FIFO contains one or more entries; it is cleared when the receive FIFO is empty. **Values:** ■ 0x1 (NOT_EMPTY): Rx FIFO not empty ■ 0x0 (EMPTY): Rx FIFO is empty **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
2	TFE	R	Transmit FIFO Completely Empty. When the transmit FIFO is completely empty, this bit is set. When it contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. **Values:** ■ 0x1 (EMPTY): Tx FIFO is empty ■ 0x0 (NON_EMPTY): Tx FIFO not empty **Value After Reset:** 0x1 **Exists:** Always **Volatile:** true

1	TFNF	R	Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full. **Values:** ■ 0x1 (NOT_FULL): Tx FIFO not full ■ 0x0 (FULL): Tx FIFO is full **Value After Reset:** 0x1 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
0	ACTIVITY	R	I2C Activity Status. **Values:** ■ 0x1 (ACTIVE): I2C is active ■ 0x0 (INACTIVE): I2C is idle **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.33 IC_TXFLR

- **Name:** I2C Transmit FIFO Level Register
- **Description:** I2C Transmit FIFO Level Register.
  This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:
  - The I2C is disabled
  - There is a transmit abort - that is, TX_ABRT bit is set in the IC_RAW_INTR_STAT register
  - The slave bulk transmit mode is aborted

  The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.
- **Size:** 32 bits
- **Offset:** 0x74
- **Exists:** Always

Bits	Name	Memory Access	Description

31:y	RSVD_TXFLR	R	TXFLR Register field Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
x:0	TXFLR	R	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.34 IC_RXFLR

- **Name:** I2C Receive FIFO Level Register
- **Description:** I2C Receive FIFO Level Register.
  This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:
  - The I2C is disabled
  - Whenever there is a transmit abort caused by any of the events tracked in IC_TX_ABRT_SOURCE

  The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.
- **Size:** 32 bits
- **Offset:** 0x78
- **Exists:** Always

Bits	Name	Memory Access	Description
31:y	RSVD_RXFLR	R	RXFLR Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
x:0	RXFLR	R	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.35 IC_SDA_HOLD

- **Name:** I2C SDA Hold Time Length Register
- **Description:** I2C SDA Hold Time Length Register.
  The bits [15:0] of this register are used to control the hold time of SDA during transmit in both slave

  and master mode (after SCL goes from HIGH to LOW).

The bits [23:16] of this register are used to extend the SDA transition (if any) whenever SCL is HIGH in the

receiver in either master or slave mode.Writes to this register succeed only when IC_ENABLE[0]=0.

The values in this register are in units of ic_clk period. The value programmed in IC_SDA_TX_HOLD

must be greater than the minimum hold time in each mode one cycle in master mode, seven cycles in slave

mode for the value to be implemented.

The programmed SDA hold time during transmit (IC_SDA_TX_HOLD) cannot exceed at any time

the duration of the low part of scl. Therefore the programmed value cannot be larger than N_SCL_LOW-2,

where N_SCL_LOW is the duration of the low part of the scl period measured in ic_clk cycles.

- **Size:** 32 bits
- **Offset:** 0x7c
- **Exists:** Always

Bits	Name	Memory Access	Description
31:24	RSVD_IC_SDA_HOLD	R	IC_SDA_HOLD Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
23:16	IC_SDA_RX_HOLD	R/W	Sets the required SDA hold time in units of ic_clk period, when Apb_i2c acts as a receiver. **Exists:** Always
15:0	IC_SDA_TX_HOLD	R/W	Sets the required SDA hold time in units of ic_clk period, when Apb_i2c acts as a transmitter. **Exists:** Always

12.2.36 IC_TX_ABRT_SOURCE

- **Name:** I2C Transmit Abort Source Register
- **Description:** I2C Transmit Abort Source Register.
  This register has 32 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is

  cleared whenever the IC_CLR_TX_ABRT register or the IC_CLR_INTR register is read. To clear

  bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must

  be cleared (IC_TAR[10]).

Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same

manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before

attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

- **Size:** 32 bits
- **Offset:** 0x80
- **Exists:** Always

Bits	Name	Memory Access	Description
31:23	TX_FLUSH_CNT	R	This field indicates the number of Tx FIFO Data Commands which are flushed due to TX_ABRT interrupt. It is cleared whenever I2C is disabled. **Role of Apb_i2c:** Master-Transmitter or Slave- Transmitter **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
22:21	RSVD_IC_TX_ABRT_SOURCE	R	IC_TX_ABRT_SOURCE Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
20	ABRT_DEVICE_WRITE	R	This is a master-mode-only bit. Master is initiating the DEVICE_ID transfer and the Tx-FIFO consists of write commands. **Role of Apb_i2c:** Master **Values:** ▪ 0x1 (ACTIVE): This abort is generated because of NOACK for Slave address ▪ 0x0 (INACTIVE): This abort is not generated **Value After Reset:** 0x0 **Volatile:** true
19	ABRT_DEVICE_SLVADDR_NOA CK	R	This is a master-mode-only bit. Master is initiating the DEVICE_ID transfer and the slave address sent was not acknowledged by any slave. **Role of Apb_i2c:** Master **Values:** ▪ 0x1 (ACTIVE): This abort is generated because of NOACK for Slave address ▪ 0x0 (INACTIVE): This abort is not generated **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
18	ABRT_DEVICE_NOACK	R	This is a master-mode-only bit. Master is initiating the DEVICE_ID transfer and the device id sent was not acknowledged by any slave. **Role of Apb_i2c:** Master **Values:** ▪ 0x1 (ACTIVE): This abort is generated because of NOACK for DEVICE-ID ▪ 0x0 (INACTIVE): This abort is not generated **Value After Reset:** 0x0 **Volatile:** true
17	ABRT_SDA_STUCK_AT_LOW	R	This is a master-mode-only bit. Master detects the SDA Stuck at low for the IC_SDA_STUCK_AT_LOW_TIMEOUT value of ic_clks. **Role of Apb_i2c:** Master **Values:** ▪ 0x1 (ACTIVE): This abort is generated because of Sda stuck at low for IC_SDA_STUCK_AT_LOW_TIMEOUT value of ic_clks ▪ 0x0 (INACTIVE): This abort is not generated **Value After Reset:** 0x0 **Volatile:** true
16	ABRT_USER_ABRT	R	This is a master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1]) **Role of Apb_i2c:** Master-Transmitter **Values:** ▪ 0x1 (ABRT_USER_ABRT_GENERATED): Transfer abort detected by master ▪ 0x0 (ABRT_USER_ABRT_VOID): Transfer abort detected by master- scenario not present **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15	ABRT_SLVRD_INTX	R	When the processor side responds to a slave mode request for data to be transmitted to a remote master and user writes a 1 in CMD (bit 8) of IC_DATA_CMD register. **Role of Apb_i2c:** Slave-Transmitter **Values:** ▪ 0x1 (ABRT_SLVRD_INTX_GENERATED): Slave trying to transmit to remote master in read mode ▪ 0x0 (ABRT_SLVRD_INTX_VOID): Slave trying to transmit to remote master in read mode- scenario not present **Value After Reset:** 0x0 **Volatile:** true
14	ABRT_SLV_ARBLOST	R	This field indicates that a Slave has lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time. **Note:** Even though the slave never 'owns' the bus, something could go wrong on the bus. This is a fail safe check. For instance, during a data transmission at the low-to- high transition of SCL, if what is on the data bus is not what is supposed to be transmitted, then Apb_i2c no longer own the bus. **Role of Apb_i2c:** Slave-Transmitter **Values:** ▪ 0x1 (ABRT_SLV_ARBLOST_GENERATED): Slave lost arbitration to remote master ▪ 0x0 (ABRT_SLV_ARBLOST_VOID): Slave lost arbitration to remote master- scenario not present **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
13	ABRT_SLVFLUSH_TXFIFO	R	This field specifies that the Slave has received a read command and some data exists in the TX FIFO, so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO. **Role of Apb_i2c:** Slave-Transmitter **Values:** ▪ 0x1 (ABRT_SLVFLUSH_TXFIFO_GENERATED): Slave flushes existing data in TX-FIFO upon getting read command ▪ 0x0 (ABRT_SLVFLUSH_TXFIFO_VOID): Slave flushes existing data in TX-FIFO upon getting read command- scenario not present **Value After Reset:** 0x0 **Volatile:** true
12	ARB_LOST	R	This field specifies that the Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. **Role of Apb_i2c:** Master-Transmitter or Slave- Transmitter **Values:** ▪ 0x1 (ABRT_LOST_GENERATED): Master or Slave- Transmitter lost arbitration ▪ 0x0 (ABRT_LOST_VOID): Master or Slave-Transmitter lost arbitration- scenario not present **Value After Reset:** 0x0 **Volatile:** true
11	ABRT_MASTER_DIS	R	This field indicates that the User tries to initiate a Master operation with the Master mode disabled. **Role of Apb_i2c:** Master-Transmitter or Master- Receiver **Values:** ▪ 0x1(ABRT_MASTER_DIS_GENERATED):User initiating master operation when MASTER disabled ▪ 0x0 (ABRT_MASTER_DIS_VOID): User initiating master operation when MASTER disabled- scenario not present **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
10	ABRT_10B_RD_NORSTRT	R	This field indicates that the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the master sends a read command in 10-bit addressing mode. **Role of Apb_i2c:** Master-Receiver **Values:** ▪ 0x1 (ABRT_10B_RD_GENERATED): Master trying to read in 10Bit addressing mode when RESTART disabled ▪ 0x0 (ABRT_10B_RD_VOID): Master not trying to read in 10Bit addressing mode when RESTART disabled **Value After Reset:** 0x0
9	ABRT_SBYTE_NORSTRT	R	To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets reasserted. When this field is set to 1, the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to send a START Byte. **Role of Apb_i2c:** Master **Values:** ▪ 0x1 (ABRT_SBYTE_NORSTRT_GENERATED): User trying to send START byte when RESTART disabled ▪ 0x0 (ABRT_SBYTE_NORSTRT_VOID): User trying to send START byte when RESTART disabled-scenario not present **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
8	ABRT_HS_NORSTRT	R	This field indicates that the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to use the master to transfer data in High Speed mode. **Role of Apb_i2c:** Master-Transmitter or Master- Receiver **Values:** ▪ 0x1 (ABRT_HS_NORSTRT_GENERATED): User trying to switch Master to HS mode when RESTART disabled ▪ 0x0 (ABRT_HS_NORSTRT_VOID): User trying to switch Master to HS mode when RESTART disabled- scenario not present **Value After Reset:** 0x0 **Volatile:** true
7	ABRT_SBYTE_ACKDET	R	This field indicates that the Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). **Role of Apb_i2c:** Master **Values:** ▪ 0x1 (ABRT_SBYTE_ACKDET_GENERATED): ACK detected for START byte ▪ 0x0 (ABRT_SBYTE_ACKDET_VOID): ACK detected for START byte- scenario not present **Value After Reset:** 0x0 **Volatile:** true
6	ABRT_HS_ACKDET	R	This field indicates that the Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior). **Role of Apb_i2c:** Master **Values:** ▪ 0x1 (ABRT_HS_ACK_GENERATED): HS Master code ACKed in HS Mode ▪ 0x0 (ABRT_HS_ACK_VOID): HS Master code ACKed in HS Mode- scenario not present **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
5	ABRT_GCALL_READ	R	This field indicates that Apb_i2c in the master mode has sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1). **Role of Apb_i2c:** Master-Transmitter **Values:** ▪ 0x1 (ABRT_GCALL_READ_GENERATED): GCALL is followed by read from bus ▪ 0x0 (ABRT_GCALL_READ_VOID): GCALL is followed by read from bus-scenario not present **Value After Reset:** 0x0 **Volatile:** true
4	ABRT_GCALL_NOACK	R	This field indicates that Apb_i2c in master mode has sent a General Call and no slave on the bus acknowledged the General Call. **Role of Apb_i2c:** Master-Transmitter **Values:** ▪ 0x1 (ABRT_GCALL_NOACK_GENERATED): GCALLnot ACKed by any slave ▪ 0x0 (ABRT_GCALL_NOACK_VOID): GCALL not ACKed by any slave-scenario not present **Value After Reset:** 0x0 **Volatile:** true
3	ABRT_TXDATA_NOACK	R	This field indicates the master-mode only bit. When the master receives an acknowledgement for the address, but when it sends data byte(s) following the address, it did not receive an acknowledge from the remote slave(s). **Role of Apb_i2c:** Master-Transmitter **Values:** ▪ 0x1 (ABRT_TXDATA_NOACK_GENERATED): Transmitted data not ACKed by addressed slave ▪ 0x0 (ABRT_TXDATA_NOACK_VOID): Transmitted data non-ACKed by addressed slave-scenario not present **Value After Reset:** 0x0

Bits	Name	Memory Access	Description
2	ABRT_10ADDR2_NOACK	R	This field indicates that the Master is in 10-bit address mode and that the second address byte of the 10-bit address was not acknowledged by any slave. **Role of Apb_i2c:** Master-Transmitter or Master- Receiver **Values:** ■ 0x1 (ACTIVE): Byte 2 of 10Bit Address not ACKed by any slave ■ 0x0 (INACTIVE): This abort is not generated **Value After Reset:** 0x0 **Volatile:** true
1	ABRT_10ADDR1_NOACK	R	This field indicates that the Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. **Role of Apb_i2c:** Master-Transmitter or Master- Receiver **Values:** ■ 0x1 (ACTIVE): Byte 1 of 10Bit Address not ACKed by any slave ■ 0x0 (INACTIVE): This abort is not generated **Value After Reset:** 0x0 **Volatile:** true
0	ABRT_7B_ADDR_NOACK	R	This field indicates that the Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave. **Role of Apb_i2c:** Master-Transmitter or Master- Receiver **Values:** ■ 0x1 (ACTIVE): This abort is generated because of NOACK for 7-bit address ■ 0x0 (INACTIVE): This abort is not generated **Value After Reset:** 0x0 **Volatile:** true

12.2.37 IC_SLV_DATA_NACK_ONLY

- **Name:** Generate Slave Data NACK Register
- **Description:** Generate Slave Data NACK Register

The register is used to generate a NACK for the data part of a transfer when Apb_i2c is acting as a slave-receiver. This register only exists when the IC_SLV_DATA_NACK_ONLY parameter is set to

1. When this parameter disabled, this register does not exist and writing to the register's address has no effect.

A write can occur on this register if both of the following conditions are met:

- Apb_i2c is disabled (IC_ENABLE[0] = 0)
- Slave part is inactive (IC_STATUS[6] = 0)

Note: The IC_STATUS[6] is a register read-back location for the internal slv_activity signal; the user should poll this before writing the ic_slv_data_nack_only bit.

- **Size:** 32 bits
- **Offset:** 0x84

Bits	Name	Memory Access	Description
31:1	RSVD_IC_SLV_DATA_NACK_ON LY	R	IC_SLV_DATA_NACK_ONLY Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
0	NACK	R/W	Generate NACK. This NACK generation only occurs when Apb_i2c is a slave-receiver. If this register is set to a value of 1, it can only generate a NACK after a data byte is received; hence, the data transfer is aborted and the data received is not pushed to the receive buffer. When the register is set to a value of 0, it generates NACK/ACK, depending on normal criteria. ■ 1: generate NACK after data byte received ■ 0: generate NACK/ACK normally **Values:** ■ 0x1 (ENABLED): Slave receiver generates NACK upon data reception only ■ 0x0 (DISABLED): Slave receiver generates NACK normally **Value After Reset:** 0x0 **Exists:** Always

12.2.38 IC_DMA_CR

- **Name:** DMA Control Register
- **Description:** DMA Control Register.
  This register is only valid when Apb_i2c is configured with a set of DMA Controller interface signals (IC_HAS_DMA = 1). When Apb_i2c is not configured for DMA operation, this register does not exist and writing to the register's address has no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. There is a separate bit for transmit and receive. This can be programmed regardless of the state of IC_ENABLE.
- **Size:** 32 bits
- **Offset:** 0x88

Bits	Name	Memory Access	Description
31:2	RSVD_IC_DMA_CR_2_31	R	RSVD_IC_DMA_CR_2_31 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

1	TDMAE	R/W	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. **Values:** ▪ 0x1 (ENABLED): Transmit FIFO DMA channel enabled ▪ 0x0 (DISABLED): transmit FIFO DMA channel disabled **Value After Reset:** 0x0

Bits	Name	Memory Access	Description
0	RDMAE	R/W	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. **Values:** ▪ 0x1 (ENABLED): Receive FIFO DMA channel enabled ▪ 0x0 (DISABLED): Receive FIFO DMA channel disabled

12.2.39 IC_DMA_TDLR

- **Name:** DMA Transmit Data Level Register
- **Description:** DMA Transmit Data Level Register.
  This register is only valid when the Apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When Apb_i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.
- **Size:** 32 bits
- **Offset:** 0x8c

Bits	Name	Memory Access	Description
31:y	RSVD_DMA_TDLR	R	DMA_TDLR Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

x:0	DMATDL	R/W	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. **Value After Reset:** 0x0 **Exists:** Always

12.2.40 IC_DMA_RDLR

- **Name:** DMA Receive Data Level Register
- **Description:** I2C Receive Data Level Register.
  This register is only valid when Apb_i2c is configured with a set of DMA interface signals

  (IC_HAS_DMA = 1). When Apb_i2c is not configured for DMA operation, this register does

  not exist; writing to its address has no effect; reading from its address returns zero.
- **Size:** 32 bits
- **Offset:** 0x90

Bits	Name	Memory Access	Description
31:y	RSVD_DMA_RDLR	R	DMA_RDLR Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always
x:0	DMARDL	R/W	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO. **Value After Reset:** 0x0 **Exists:** Always

12.2.40 IC_SDA_SETUP

- **Name:** I2C SDA Setup Register
- **Description:** I2C SDA Setup Register.
  This register controls the amount of time delay (in terms of number of ic_clk clock periods)

  introduced in the rising edge of SCL - relative to SDA changing - when Apb_i2c services a

  read request in a slave-transmitter operation. The relevant I2C requirement is tSU:DAT as detailed

  in the I2C Bus Specification. This register must be programmed with a value greater than or equal

  to one ic_clock_period of time.

  Writes to this register succeed only when IC_ENABLE[0] = 0.

  Note: The length of setup time is calculated using below formula:

  - If IC_CLK_FREQ_OPTIMIZATION = 0 : [(IC_SDA_SETUP - 1) * (ic_clock_period)]
  - If IC_CLK_FREQ_OPTIMIZATION = 1 : [(IC_SDA_SETUP + 1) * (ic_clock_period)]

  The IC_SDA_SETUP register is only used by the Apb_i2c when operating as a slave transmitter.
- **Size:** 32 bits
- **Offset:** 0x94

Bits	Name	Memory Access	Description
31:8	RSVD_IC_SDA_SETUP	R	IC_SDA_SETUP Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description

7:0	SDA_SETUP	R/W	SDA Setup (tSU:DAT). It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz: - If IC_CLK_FREQ_OPTIMIZATION = 0 : IC_SDA_SETUP should be programmed to a value of 11. - If IC_CLK_FREQ_OPTIMIZATION = 1 : IC_SDA_SETUP should be programmed to a value of 9. **Exists:** Always

12.2.42 IC_ACK_GENERAL_CALL

- **Name:** I2C ACK General Call Register
- **Description:** I2C ACK General Call Register.
  The register controls whether Apb_i2c responds with a ACK or NACK when it receives an I2C General Call address.

  This register is applicable only when the Apb_i2c is in slave mode.
- **Size:** 32 bits
- **Offset:** 0x98

Bits	Name	Memory Access	Description
31:1	RSVD_IC_ACK_GEN_1_31	R	RSVD_IC_ACK_GEN_1_31 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always
0	ACK_GEN_CALL	R/W	ACK General Call. When set to 1, Apb_i2c responds with a ACK (by asserting ic_data_oe) when it receives a General Call. Otherwise, Apb_i2c responds with a NACK (by negating ic_data_oe). **Values:** ▪ 0x1 (ENABLED): Generate ACK for a General Call ▪ 0x0 (DISABLED): Generate NACK for General Call **Exists:** Always

12.2.43 IC_ENABLE_STATUS

- **Name:** I2C Enable Status Register
- **Description:** I2C Enable Status Register.

The register is used to report the Apb_i2c hardware status when the IC_ENABLE[0] register is set from

1 to 0; that is, when Apb_i2c is disabled.

If IC_ENABLE[0] has been set to 1, bits 2:1 are forced to 0, and bit 0 is forced to 1.

If IC_ENABLE[0] has been set to 0, bits 2:1 is only be valid as soon as bit 0 is read as '0'.

Note: When IC_ENABLE[0] has been set to 0, a delay occurs for bit 0 to be read as 0 because d

isabling the Apb_i2c depends on I2C bus activities.

- **Size:** 32 bits
- **Offset:** 0x9c
- **Exists:** Always

Bits	Name	Memory Access	Description
31:3	RSVD_IC_ENABLE_STATUS	R	IC_ENABLE_STATUS Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description

2	SLV_RX_DATA_LOST	R	Slave Received Data Lost. This bit indicates if a Slave- Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting bit 0 of IC_ENABLE from 1 to 0. When read as 1, Apb_i2c is deemed to have been actively engaged in an aborted I2C transfer (with matching address) and the data phase of the I2C transfer has been entered, even though a data byte has been responded with a NACK. **Note:** If the remote I2C master terminates the transfer with a STOP condition before the Apb_i2c has a chance to NACK a transfer, and IC_ENABLE[0] has been set to 0, then this bit is also set to 1. When read as 0, Apb_i2c is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer. **Note:** The CPU can safely read this bit when IC_EN (bit 0) is read as 0. **Values:** ▪ 0x1 (ACTIVE): Slave RX Data is lost ▪ 0x0 (INACTIVE): Slave RX Data is not lost **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description

1	SLV_DISABLED_WHILE_BUSY	R	Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting bit 0 of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the IC_ENABLE register while: (a) Apb_i2c is receiving the address byte of the Slave-Transmitter operation from a remote master; OR, (b) address and data bytes of the Slave-Receiver operation from a remote master. When read as 1, Apb_i2c is deemed to have forced a NACK during any part of an I2C transfer, irrespective of whether the I2C address matches the slave address set in Apb_i2c (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect. **Note:** If the remote I2C master terminates the transfer with a STOP condition before the Apb_i2c has a chance to NACK a transfer, and IC_ENABLE[0] has been set to 0, then this bit will also be set to 1. When read as 0, Apb_i2c is deemed to have been disabled when there is master activity, or when the I2C bus is idle. **Note:** The CPU can safely read this bit when IC_EN (bit 0) is read as 0. **Values:** ▪ 0x1 (ACTIVE): Slave is disabled when it is active ▪ 0x0 (INACTIVE): Slave is disabled when it is idle **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description

0	IC_EN	R	IC Enable Status. This bit always reflects the value driven on the output port ic_en. ▪ When read as 1, Apb_i2c is deemed to be in an enabled state. ▪ When read as 0, Apb_i2c is deemed completely inactive. **Note:** The CPU can safely read this bit anytime. When this bit is read as 0, the CPU can safely read SLV_RX_DATA_LOST (bit 2) and SLV_DISABLED_WHILE_BUSY (bit 1). **Values:** ▪ 0x1 (ENABLED): I2C enabled ▪ 0x0 (DISABLED): I2C disabled **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.44 IC_FS_SPKLEN

- **Name:** I2C SS, FS or FM+ spike suppression limit
- **Description:** I2C SS, FS or FM+ Spike Suppression Limit.
  This register is used to store the duration, measured in ic_clk cycles, of the longest spike that is filtered out by the spike suppression logic w hen the component is operating in SS, FS or FM+ modes. The relevant I2C requirement is tSP as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1.
- **Size:** 32 bits
- **Offset:** 0xa0

Bits	Name	Memory Access	Description
31:8	RSVD_IC_FS_SPKLEN	R	IC_FS_SPKLEN Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description

7:0	IC_FS_SPKLEN	R/W	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted results in 1 being set. **Exists:** Always

12.2.45 IC_UFM_SPKLEN

- **Name:** I2C Ultra-Fast mode spike suppression limit
- **Description:** I2C UFM spike suppression limit.
  This register is used to store the duration, measured in ic_clk cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in Ultra-Fast mode. The relevant I2C requirement is tSP as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1.
- **Size:** 32 bits
- **Offset:** 0xa0

Bits	Name	Memory Access	Description
31:8	RSVD_IC_UFM_SPKLEN	R	IC_UFM_SPKLEN Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description

7:0	IC_UFM_SPKLEN	R/W	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted results in 1 being set. **Exists:** Always

12.2.46 IC_HS_SPKLEN

- **Name:** I2C HS spike suppression limit register
- **Description:** I2C HS spike suppression limit register.
  This register is used to store the duration, measured in ic_clk cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in HS modes. The relevant I2C requirement is tSP as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1 and is implemented only if the component is configured to support HS mode; that is, if the IC_MAX_SPEED_MODE parameter is set to 3.
- **Size:** 32 bits
- **Offset:** 0xa4

Bits	Name	Memory Access	Description
31:8	RSVD_IC_HS_SPKLEN	R	IC_HS_SPKLEN Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description

7:0	IC_HS_SPKLEN	R/W	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic; This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted results in 1 being set. **Exists:** Always

12.2.47 IC_CLR_RESTART_DET

- **Name:** Clear RESTART_DET Interrupt Register
- **Description:** Clear RESTART_DET Interrupt Register
- **Size:** 32 bits
- **Offset:** 0xa8

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_RESTART_DET	R	IC_CLR_RESTART_DET Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
0	CLR_RESTART_DET	R	Read this register to clear the RESTART_DET interrupt (bit 12) of IC_RAW_INTR_STAT register. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.48 IC_SCL_STUCK_AT_LOW_TIMEOUT

- **Name:** I2C SCL Stuck at Low Timeout register
- **Description:** I2C SCL Stuck at Low Timeout.
  This register is used to store the duration, measured in ic_clk cycles, used to generate an Interrupt (ic_scl_stuck_at_low_intr/ic_scl_stuck_at_low_intr_n) if SCL is held low for the IC_SCL_STUCK_LOW_TIMEOUT duration.
- **Size:** 32 bits
- **Offset:** 0xac

Bits	Name	Memory Access	Description
31:0	IC_SCL_STUCK_LOW_TIMEOU T	R/W	Apb_i2c generate the interrupt to indicate SCL stuck at low (ic_scl_stuck_at_low_intr/ic_scl_stuck_at_low_intr_n) if it detects the SCL stuck at low for the IC_SCL_STUCK_LOW_TIMEOUT in units of ic_clk period. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. **Exists:** Always

12.2.49 IC_SDA_STUCK_AT_LOW_TIMEOUT

- **Name:** I2C SDA Stuck at Low Timeout register
- **Description:** I2C SDA Stuck at Low Timeout.
  This register is used to store the duration, measured in ic_clk cycles, used to Recover the Data (SDA) line through sending SCL pulses if SDA is held low for the mentioned duration.
- **Size:** 32 bits
- **Offset:** 0xb0

Bits	Name	Memory Access	Description
31:0	IC_SDA_STUCK_LOW_TIMEOU T	R/W	Apb_i2c initiates the recovery of SDA line through enabling the SDA_STUCK_RECOVERY_EN (IC_ENABLE[3]) register bit, if it detects the SDA stuck at low for the IC_SDA_STUCK_LOW_TIMEOUT in units of ic_clk period. **Exists:** Always

12.2.50 IC_CLR_SCL_STUCK_DET

- **Name:** Clear SCL Stuck at Low Detect interrupt Register
- **Description:** Clear SCL Stuck at Low Detect Interrupt Register
- **Size:** 32 bits
- **Offset:** 0xb4

Bits	Name	Memory Access	Description

31:1	RSVD_CLR_SCL_STUCK_DET	R	CLR_SCL_STUCK_DET Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always
0	CLR_SCL_STUCK_DET	R	Read this register to clear the SCL_STUCT_AT_LOW interrupt (bit 15) of the IC_RAW_INTR_STAT register. **Value After** **Reset:** 0x0 **Exists:** Always

12.2.51 IC_DEVICE_ID

- **Name:** I2C Device-Id register
- **Description:** I2C Device-ID Register.

This Register contains the Device-ID of the component which includes 12-bits of Manufacturer name. and 9-bits of part identification and 3 bits of die-version.

- **Size:** 32 bits
- **Offset:** 0xb8

Bits	Name	Memory Access	Description
31:24	RSVD_IC_DEVICE_ID	R	IC_DEVICE_ID Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always
23:0	DEVICE-ID	R	Contains the Device-ID of the component assigned through the configuration parameter 'IC_DEVICE_ID_VALUE' **Exists:** Always

12.2.52 IC_SMBUS_CLK_LOW_SEXT

- **Name:** SMBus Slave Clock Extend Timeout register
- **Description:** SMBus Slave Clock Extend Timeout Register.
  This Register contains the Timeout value used to determine the Slave Clock Extend Timeout in one transfer (from START to STOP).

  This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS is set to 1. This register is used to store the duration, measured in ic_clk cycles, used to detect the slave clock extend timeout if slave extends the clock (SCL) for the mentioned duration.
- **Size:** 32 bits
- **Offset:** 0xbc

Bits	Name	Memory Access	Description
31:0	SMBUS_CLK_LOW_SEXT_TIMEOUT	R/W	This field is used to detect the Slave Clock Extend timeout (tLOW:SEXT) in master mode extended by the slave device in one message from the initial START to the STOP. The values in this register are in units of ic_clk period. **Exists:** Always

12.2.53 IC_SMBUS_CLK_LOW_MEXT

- **Name:** SMBus Master Clock Extend Timeout register
- **Description:** SMBus Master Clock Extend Timeout Register.
  This Register contains the Timeout value used to determine the Master Clock Extend Timeout in one byte of transfer.

  This register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS is set to 1. This register is used to store the duration, measured in ic_clk cycles, used to detect the Master clock extend timeout if Master extends the clock (SCL) for the mentioned duration.
- **Size:** 32 bits
- **Offset:** 0xc0

Bits	Name	Memory Access	Description
31:0	SMBUS_CLK_LOW_MEXT_TIMEOUT	R/W	This field is used to detect the Master extend SMBus clock (SCLK) timeout defined from START-to-ACK, ACK-to-ACK, or ACK-to-STOP in Master mode. The values in this register are in units of ic_clk period. **Exists:** Always

12.2.54 IC_SMBUS_THIGH_MAX_IDLE_COUNT

- **Name:** SMBus Master THigh MAX Bus-idle count Register
- **Description:** SMBus Master THigh MAX Bus-idle count Register.
  This register programs the Bus-idle time period used when a master has been dynamically added to the bus or when a master has generated a clock reset on the bus. This register is used to store the duration, measured in ic_clk cycles, used to detect the Bus Idle condition if SCL and SDA are held high for the mentioned duration. This register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS is set to 1.
- **Size:** 32 bits

- **Offset:** 0xc4

Bits	Name	Memory Access	Description
31:16	RSVD_SMBUS_THIGH_MAX_BUS_IDLE_CNT	R	SMBUS_THIGH_MAX_BUS_IDLE_CNT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
15:0	SMBUS_THIGH_MAX_BUS_IDL E_CNT	R/W	This field is used to set the required Bus-Idle time period used when a master has been dynamically added to the bus and may not have detected a state transition on the SMBCLK or SMBDAT lines. In this case, the master must wait long enough to ensure that a transfer is not currently in progress The values in this register are in units of ic_clk period. **Exists:** Always

12.2.55 IC_SMBUS_INTR_STAT

- **Name:** SMBus Interrupt Status Register
- **Description:** SMBUS Interrupt Status Register.
  Each bit in this register has a corresponding mask bit in the IC_SMBUS_INTR_MASK register.

  These bits are cleared by writing the matching SMBus interrupt clear register (IC_CLR_SMBUS_INTR)

  bits.

  The unmasked raw versions of these bits are available in the IC_SMBUS_RAW_INTR_STAT register.
- **Size:** 32 bits
- **Offset:** 0xc8

Bits	Name	Memory Access	Description
31:11	RSVD_IC_SMBUS_INTR_STAT	R	IC_SMBUS_INTR_STAT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
10	R_SMBUS_ALERT_DET	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_SMBUS_ALERT_DET bit. **Values:** ■ 0x1 (ACTIVE): SMBUS_ALERT_DET interrupt is active ■ 0x0 (INACTIVE): SMBUS_ALERT_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
9	R_SMBUS_SUSPEND_DET	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_SMBUS_SUSPEND_DET bit. **Values:** ■ 0x1 (ACTIVE): SMBUS_SUSPEND_DET interrupt is active ■ 0x0 (INACTIVE): SMBUS_SUSPEND_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
8	R_SLV_RX_PEC_NACK	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_SLV_RX_PEC_NACK bit. **Values:** ■ 0x1 (ACTIVE): SLV_RX_PEC_NACK interrupt is active ■ 0x0 (INACTIVE): SLV_RX_PEC_NACK interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

7	R_ARP_ASSGN_ADDR_CMD_D ET	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_ARP_ASSGN_ADDR_CMD_DET bit. **Values:** ▪ 0x1 (ACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is active ▪ 0x0 (INACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
6	R_ARP_GET_UDID_CMD_DET	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_ARP_GET_UDID_CMD_DET bit. **Values:** ▪ 0x1 (ACTIVE): ARP_GET_UDID_CMD_DET interrupt is active ▪ 0x0 (INACTIVE): ARP_GET_UDID_CMD_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
5	R_ARP_RST_CMD_DET	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_ARP_RST_CMD_DET bit. **Values:** ▪ 0x1 (ACTIVE): ARP_RST_CMD_DET interrupt is active ▪ 0x0 (INACTIVE): ARP_RST_CMD_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

4	R_ARP_PREPARE_CMD_DET	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_ARP_PREPARE_CMD_DET bit. **Values:** ■ 0x1 (ACTIVE): ARP_PREPARE_CMD_DET interrupt is active ■ 0x0 (INACTIVE): ARP_PREPARE_CMD_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
3	R_HOST_NOTIFY_MST_DET	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_HOST_NOTIFY_MST_DET bit. **Values:** ■ 0x1 (ACTIVE): HOST_NOTIFY_MST_DET interrupt is active ■ 0x0 (INACTIVE): HOST_NOTIFY_MST_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
2	R_QUICK_CMD_DET	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_QUICK_CMD_DET bit. **Values:** ■ 0x1 (ACTIVE): QUICK_CMD_DET interrupt is active ■ 0x0 (INACTIVE): QUICK_CMD_DET interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always

1	R_MST_CLOCK_EXTND_TIMEO UT	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_MST_CLOCK_EXTND_TIMEOUT bit. **Values:** ■ 0x1 (ACTIVE): MST_CLOCK_EXTND_TIMEOUT interrupt is active ■ 0x0 (INACTIVE): MST_CLOCK_EXTND_TIMEOUT interrupt is inactive **Value After Reset:** 0x0
0	R_SLV_CLOCK_EXTND_TIMEO UT	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_SLV_CLOCK_EXTND_TIMEOUT bit. **Values:** ■ 0x1 (ACTIVE): SLV_CLOCK_EXTND_TIMEOUT interrupt is active ■ 0x0 (INACTIVE): SLV_CLOCK_EXTND_TIMEOUT interrupt is inactive **Value After Reset:** 0x0

12.2.56 IC_SMBUS_INTR_MASK

- **Name:** SMBus Interrupt Mask Register
- **Description:** SMBus Interrupt Mask Register
- **Size:** 32 bits
- **Offset:** 0xcc

Bits	Name	Memory Access	Description
31:11	RSVD_IC_SMBUS_INTR_MASK	R	IC_SMBUS_INTR_MASK Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always

10	M_SMBUS_ALERT_DET	R/W	This bit masks the R_SMBUS_ALERT_DET interrupt in IC_SMBUS_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): SMBUS_ALERT_DET interrupt is unmasked ▪ 0x0 (ENABLED): SMBUS_ALERT_DET interrupt is masked **Value After Reset:** 0x1

Bits	Name	Memory Access	Description
9	M_SMBUS_SUSPEND_DET	R/W	This bit masks the M_SMBUS_SUSPEND_DET interrupt in IC_SMBUS_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): SMBUS_SUSPEND_DET interrupt is unmasked ▪ 0x0 (ENABLED): SMBUS_SUSPEND_DET interrupt is masked **Value After Reset:** 0x1
8	M_SLV_RX_PEC_NACK	R/W	This bit masks the R_SLV_RX_PEC_NACK interrupt in IC_SMBUS_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): SLV_RX_PEC_NACK interrupt is unmasked ▪ 0x0 (ENABLED): SLV_RX_PEC_NACK interrupt is masked **Value After Reset:** 0x1
7	M_ARP_ASSGN_ADDR_CMD_DET	R/W	This bit masks the R_ARP_ASSGN_ADDR_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): ARP_ASSGN_ADDR_CMD_DET interrupt is unmasked ▪ 0x0 (ENABLED): ARP_ASSGN_ADDR_CMD_DET interrupt is masked **Value After Reset:** 0x1

6	M_ARP_GET_UDID_CMD_DET	R/W	This bit masks the R_ARP_GET_UDID_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): ARP_GET_UDID_CMD_DET interrupt is unmasked ■ 0x0 (ENABLED): ARP_GET_UDID_CMD_DET interrupt is masked **Value After Reset:** 0x1

Bits	Name	Memory Access	Description
5	M_ARP_RST_CMD_DET	R/W	This bit masks the R_ARP_RST_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): ARP_RST_CMD_DET interrupt is unmasked ■ 0x0 (ENABLED): ARP_RST_CMD_DET interrupt is masked **Value After Reset:** 0x1
4	M_ARP_PREPARE_CMD_DET	R/W	This bit masks the R_ARP_PREPARE_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): ARP_PREPARE_CMD_DET interrupt is unmasked ■ 0x0 (ENABLED): ARP_PREPARE_CMD_DET interrupt is masked **Value After Reset:** 0x1
3	M_HOST_NOTIFY_MST_DET	R/W	This bit masks the R_HOST_NOTIFY_MST_DET interrupt in IC_SMBUS_INTR_STAT register. **Values:** ■ 0x1 (DISABLED): HOST_NOTIFY_MST_DET interrupt is unmasked ■ 0x0 (ENABLED): HOST_NOTIFY_MST_DET interrupt is masked **Value After Reset:** 0x1

2	M_QUICK_CMD_DET	R/W	This bit masks the R_QUICK_CMD_DET interrupt in IC_SMBUS_INTR_STATregister. **Values:** ▪ 0x1 (DISABLED): QUICK_CMD_DET interrupt is unmasked ▪ 0x0 (ENABLED): QUICK_CMD_DET interrupt is masked **Value After Reset:** 0x1 **Exists:** Always

Bits	Name	Memory Access	Description
1	M_MST_CLOCK_EXTND_TIMEO UT	R/W	This bit masks the R_MST_CLOCK_EXTND_TIMEOUT interrupt in IC_SMBUS_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): MST_CLOCK_EXTND_TIMEOUT interrupt is unmasked ▪ 0x0 (ENABLED): MST_CLOCK_EXTND_TIMEOUT interrupt is masked **Value After Reset:** 0x1 **Exists:** Always
0	M_SLV_CLOCK_EXTND_TIMEO UT	R/W	This bit masks the R_SLV_CLOCK_EXTND_TIMEOUT interrupt in IC_SMBUS_INTR_STAT register. **Values:** ▪ 0x1 (DISABLED): SLV_CLOCK_EXTND_TIMEOUT interrupt is unmasked ▪ 0x0 (ENABLED): SLV_CLOCK_EXTND_TIMEOUT interrupt is masked **Value After Reset:** 0x1 **Exists:** Always

12.2.57 IC_SMBUS_RAW_INTR_STAT

- **Name:** SMBus Raw Interrupt Status Register
- **Description:** SMBus Raw Interrupt Status Register.
  Unlike the IC_SMBUS_INTR_STAT register, these bits are not masked so they always show the true status of the Apb_i2c.
- **Size:** 32 bits

- **Offset:** 0xd0

Bits	Name	Memory Access	Description
31:11	RSVD_IC_SMBUS_RAW_INTR_ STAT	R	IC_SMBUS_RAW_INTR_STAT Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
10	SMBUS_ALERT_ DET	R	Indicates whether a SMBALERT (ic_smbalert_in_n) signal is driven low by the slave. **Values:** ▪ 0x1 (ACTIVE): SMBUS Alert interrupt is active ▪ 0x0 (INACTIVE): SMBUS Alert interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
9	SMBUS_SUSPE ND_DET	R	Indicates whether a SMBSUS (ic_smbsus_in_n) signal is driven low by the Host. **Values:** ▪ 0x1 (ACTIVE): SMBUS System Suspended interrupt is active ▪ 0x0 (INACTIVE): SMBUS System Suspended interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
8	SLV_RX_PEC_N ACK	R	Indicates whether a NACK has been sent due to PEC mismatch while working as ARP slave. **Values:** ▪ 0x1 (ACTIVE): SLV_RX_PEC_NACK interrupt is active ▪ 0x0 (INACTIVE): SLV_RX_PEC_NACK interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

7	ARP_ASSGN_ADDR_CMD_DET	R	Indicates whether an Assign Address ARP command has been received. **Values:** ▪ 0x1 (ACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is active ▪ 0x0 (INACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
6	ARP_GET_UDID_CMD_DET	R	Indicates whether a Get UDID ARP command has been received. **Values:** ▪ 0x1 (ACTIVE): ARP_GET_UDID_CMD_DET interrupt is active ▪ 0x0 (INACTIVE): ARP_GET_UDID_CMD_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
5	ARP_RST_CMD_DET	R	Indicates whether a General or Directed Reset ARP command has been received. **Values:** ▪ 0x1 (ACTIVE): ARP_RST_CMD_DET interrupt is active ▪ 0x0 (INACTIVE): ARP_RST_CMD_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

4	ARP_PREPARE_CMD_DET	R	Indicates whether a prepare to ARP command has been received. **Values:** ▪ 0x1 (ACTIVE): ARP_PREPARE_CMD_DET interrupt is active ▪ 0x0 (INACTIVE): ARP_PREPARE_CMD_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true
3	HOST_NTFY_MST_DET	R	Indicates whether a Notify ARP Master ARP command has been received. **Values:** ▪ 0x1 (ACTIVE): HOST_NTFY_MST_DET interrupt is active ▪ 0x0 (INACTIVE): HOST_NTFY_MST_DET interrupt is inactive **Value After Reset:** 0x0 **Volatile:** true

Bits	Name	Memory Access	Description
2	QUICK_CMD_DET	R	Indicates whether a Quick command has been received on the SMBus interface regardless of whether Apb_i2c is operating in slave or master mode. Enabled only when IC_SMBUS=1 is set to 1. **Values:** ▪ 0x1 (ACTIVE): Quick Command interrupt is active ▪ 0x0 (INACTIVE): Quick Command interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

1	MST_CLOCK_EXTND_TIMEOUT	R	Indicates whether the Master device transaction (START-to- ACK, ACK-to-ACK, or ACK-to-STOP) from START to STOP exceeds IC_SMBUS_CLOCK_LOW_MEXT time with in each byte of message. This bit is enabled only when: ▪ IC_SMBUS=1 ▪ IC_CON[0]=1 ▪ IC_EMPTYFIFO_HOLD_MASTER_EN=1 or ▪ IC_RX_FULL_HLD_BUS_EN=1 **Values:** ▪ 0x1 (ACTIVE): Master Clock Extend Timeout interrupt is active ▪ 0x0 (INACTIVE): Master Clock Extend Timeout interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

Bits	Name	Memory Access	Description
0	SLV_CLOCK_EXTND_TIMEOUT	R	Indicates whether the transaction from Slave (i.e from START to STOP) exceeds IC_SMBUS_CLK_LOW_SEXT time. This bit is enabled only when: ▪ IC_SMBUS=1 ▪ IC_CON[0]=1 **Values:** ▪ 0x1 (ACTIVE): Slave Clock Extend Timeout interrupt is active ▪ 0x0 (INACTIVE): Slave Clock Extend Timeout interrupt is inactive **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.58 IC_CLR_SMBUS_INTR

- **Name:** Clear SMBus Interrupt Register
- **Description:** SMBus Clear Interrupt Register
- **Size:** 32 bits

- **Offset:** 0xd4

Bits	Name	Memory Access	Description
31:11	RSVD_IC_CLR_SMBUS_INTR	W	IC_CLR_SMBUS_INTR Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always
10	CLR_SMBUS_ALERT_DET	W	Write this register bit to clear the SMBUS_ALERT_DET interrupt (bit 10) of the IC_SMBUS_RAW_INTR_STAT register. **Value After Reset:** 0x0
Bits	**Name**	**Memory Access**	**Description**
9	CLR_SMBUS_SUSPEND_DET	W	Write this register bit to clear the SMBUS_SUSPEND_DET interrupt (bit 9) of the IC_SMBUS_RAW_INTR_STAT register. **Value After Reset:** 0x0
8	CLR_SLV_RX_PEC_NACK	W	Write this register bit to clear the SLV_RX_PEC_NACK interrupt (bit 8) of the IC_SMBUS_RAW_INTR_STAT register. **Value After Reset:** 0x0
7	CLR_ARP_ASSGN_ADDR_CMD_DET	W	Write this register bit to clear the ARP_ASSGN_ADDR_CMD_DET interrupt (bit 7) of the IC_SMBUS_RAW_INTR_STATregister. **Value After Reset:** 0x0
6	CLR_ARP_GET_UDID_CMD_DE T	W	Write this register bit to clear the ARP_GET_UDID_CMD_DET interrupt (bit 6) of the IC_SMBUS_RAW_INTR_STATregister. **Value After Reset:** 0x0
5	CLR_ARP_RST_CMD_DET	W	Write this register bit to clear the ARP_RST_CMD_DET interrupt (bit 5) of the IC_SMBUS_RAW_INTR_STAT register. **Value After Reset:** 0x0

4	CLR_ARP_PREPARE_CMD_DET	W	Write this register bit to clear the ARP_PREPARE_CMD_DET interrupt (bit 4) of the IC_SMBUS_RAW_INTR_STATregister. **Value After Reset:** 0x0
3	CLR_HOST_NOTIFY_MST_DET	W	Write this register bit to clear the HOST_NOTIFY_MST_DET interrupt (bit 3) of the IC_SMBUS_RAW_INTR_STAT register. **Value After Reset:** 0x0
2	CLR_QUICK_CMD_DET	W	Write this register bit to clear the QUICK_CMD_DET interrupt (bit 2) of the IC_SMBUS_RAW_INTR_STAT register. **Value After Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
1	CLR_MST_CLOCK_EXTND_TIM EOUT	W	Write this register bit to clear the MST_CLOCK_EXTND_TIMEOUT interrupt (bit 1) of the IC_SMBUS_RAW_INTR_STATregister. **Value After Reset:** 0x0 **Exists:** Always
0	CLR_SLV_CLOCK_EXTND_TIME OUT	W	Write this register bit to clear the SLV_CLOCK_EXTND_TIMEOUT interrupt (bit 0) of the IC_SMBUS_RAW_INTR_STATregister. **Value After Reset:** 0x0 **Exists:** Always

12.2.59 IC_OPTIONAL_SAR

- **Name:** I2C Optional Slave Address Register
- **Description:** I2C Optional Slave Address Register.
  Optional Slave address for I2C in SMBus Mode. A same restriction as IC_SAR applies on IC_OPTIONAL_SAR.
- **Size:** 32 bits
- **Offset:** 0xd8

Bits	Name	Memory Access	Description
31:7	RSVD_IC_OPTIONAL_SAR	R	IC_OPTIONAL_SAR Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always
6:0	OPTIONAL_SAR	R/W	Optional Slave address for Apb_i2c when operating as a slave in SMBus Mode. **Exists:** Always

12.2.60 IC_SMBUS_UDID_LSB

- **Name:** SMBUS ARP UDID LSB Register
- **Description:** SMBUS ARP UDID LSB Register.
  This Register can be written only when the Apb_i2c is disabled, which corresponds to IC

  ENABLE[0] being set to 0. This register is present only if configuration parameter IC

  SMBUS_UDID_HC is set to 1. This register is used to store the LSB 32 bit value of Slave

  UDID register used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0xdc

Bits	Name	Memory Access	Description
31:0	SMBUS_UDID_LSB	R/W	This field is used to store the LSB 32 bit value of slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.61 IC_SMBUS_UDID_WORD0

- **Name:** SMBUS ARP UDID WORD0 Register
- **Description:** SMBUS UDID WORD0 Register
  This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS_UDID_HC is set to 0. This register is used to store the Lower 32 bit value of Slave UDID register i.e. UDID[31:0] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0xdc

Bits	Name	Memory Access	Description
31:0	SMBUS_UDID_WORD0	R/W	This field is used to store the Lower 32 bit value of slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.62 IC_SMBUS_UDID_WORD1

- **Name:** SMBUS ARP UDID WORD1 Register
- **Description:** SMBUS UDID WORD1 Register
  This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS_UDID_HC is set to 0. This register is used to store the Middle-Lower 32 bit value of Slave UDID register i.e. UDID[63:32] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0xe0

Bits	Name	Memory Access	Description
31:0	SMBUS_UDID_ WORD1	R/W	This field is used to store the Middle-Lower 32 bit value of slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.63 IC_SMBUS_UDID_WORD2

- **Name:** SMBUS ARP UDID WORD2 Register
- **Description:** SMBUS UDID WORD2 Register
  This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_

  ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS_

  UDID_HC is set to 0. This register is used to store the Middle-Upper 32 bit value of Slave UDID

  register i.e. UDID[95:64] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0xe4

Bits	Name	Memory Access	Description
31:0	SMBUS_UDID_WORD2	R/W	This field is used to store the Middle-Upper 32 bit value of slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.64 IC_SMBUS_UDID_WORD3

- **Name:** SMBUS ARP UDID WORD3 Register
- **Description:** SMBUS UDID WORD3 Register
  This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_

  ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS

  _UDID_HC is set to 0. This register is used to store the Upper 32 bit value of Slave UDID

  register i.e. UDID[127:96] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0xe8

Bits	Name	Memory Access	Description
31:0	SMBUS_UDID_WORD3	R/W	This field is used to store the Upper 32 bit value of slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.65 REG_TIMEOUT_RST

- **Name:** Register timeout counter reset value
- **Description:** Name: Register timeout counter reset register Size: REG_TIMEOUT_WIDTH bits

  Address: 0xF0 Read/Write Access: Read/Write This register keeps the timeout value of register timer

  counter. The reset value of the register is REG_TIMEOUT_VALUE. The default reset value can

  be further modified if HC_REG_TIMEOUT_VALUE = 0. The final programmed value (or the

  default reset value if not programmed) determines from what value the register timeout counter starts

  counting down. A zero on this counter will break the waited transaction with

PSLVERR as high.

- **Size:** 32 bits
- **Offset:** 0xf0

Bits	Name	Memory Access	Description
31:y	RSVD_REG_TIMEOUT_RST	R	Reserved bits - Read Only **Value After** **Reset:** 0x0 **Exists:** Always

Bits	Name	Memory Access	Description
x:0	REG_TIMEOUT_RST_ro	R	This field holds reset value of REG_TIMEOUT counter register. **Volatile:** true
x:0	REG_TIMEOUT_RST_rw	R/W	This field holds reset value of REG_TIMEOUT counter register. **Volatile:** true

12.2.66 IC_COMP_PARAM_1

- **Name:** Component Parameter Register 1
- **Description:** Component Parameter Register 1
  Note: This is a constant read-only register that contains encoded information about the component's

  parameter settings.
- **Size:** 32 bits
- **Offset:** 0xf4

Bits	Name	Memory Access	Description
31:24	RSVD_IC_COMP_PARAM_1	R	IC_COMP_PARAM_1 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always
23:16	TX_BUFFER_DEPTH	R	▪ 0x00 = Reserved ▪ 0x01 = 2 ▪ 0x02 = 3 ▪ ... ▪ 0xFF = 256 **Exists:** Always

Bits	Name	Memory Access	Description
15:8	RX_BUFFER_DEPTH	R	▪ 0x00: Reserved ▪ 0x01: 2 ▪ 0x02: 3 ▪ ... ▪ 0xFF: 256 **Exists:** Always
7	ADD_ENCODED_PARAMS	R	Reading 1 in this bit means that the capability of reading these encoded parameters via software has been included. Otherwise, the entire register is 0 regardless of the setting of any other parameters that are encoded in the bits. **Values:** ▪ 0x1 (ENABLED): Enables capability of reading encoded parameters ▪ 0x0 (DISBALED): Disables capability of reading encoded parameters **Exists:** Always

6	HAS_DMA	R	**Values:** ▪ 0x1 (ENABLED): DMA handshaking signals are enabled ▪ 0x0 (DISABLED): DMA handshaking signals are disabled **Exists:** Always
5	INTR_IO	R	**Values:** ▪ 0x1 (COMBINED): COMBINED Interrupt outputs ▪ 0x0 (INDIVIDUAL): INDIVIDUAL Interrupt outputs **Exists:** Always

Bits	Name	Memory Access	Description
4	HC_COUNT_VALUES	R	**Values:** ▪ 0x1 (ENABLED): Hard code the count values for each mode. ▪ 0x0 (DISABLED): Programmable count values for each mode. **Exists:** Always
3:2	MAX_SPEED_MODE	R	▪ 0x0: Reserved ▪ 0x1: Standard ▪ 0x2: Fast ▪ 0x3: High **Values:** ▪ 0x1 (STANDARD): IC MAX SPEED is STANDARD MODE ▪ 0x2 (FAST): IC MAX SPEED is FAST MODE ▪ 0x3 (HIGH): IC MAX SPEED is HIGH MODE **Exists:** IC_ULTRA_FAST_MODE==0
1:0	APB_DATA_WIDTH	R	**Values:** ▪ 0x0 (APB_08BITS): APB data bus width is 08 bits ▪ 0x1 (APB_16BITS): APB data bus width is 16 bits ▪ 0x2 (APB_32BITS): APB data bus width is 32 bits ▪ 0x3 (RESERVED): Reserved bits **Value After Reset:** ENCODED_APB_DATA_WIDTH **Exists:** Always

12.2.67 IC_SAR2

- **Name:** I2C Slave Address Register 2
- **Description:** I2C Slave Address Register 2
- **Size:** 32 bits
- **Offset:** 0x100

Bits	Name	Memory Access	Description
31:10	RSVD_IC_SAR2	R	IC_SAR2 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
9:0	IC_SAR2	R/W	The IC_SAR2 holds the slave address 2 when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR2[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. **Note:** The default values cannot be any of the reserved address locations: that is, 0x00 to 0x07, or 0x78 to 0x7f. The correct operation of the device is not guaranteed if you program the IC_SAR2 or IC_TAR to a reserved value. **Exists:** Always **Volatile:** true

12.2.68 IC_SAR3

- **Name:** I2C Slave Address Register 3
- **Description:** I2C Slave Address Register 3
- **Size:** 32 bits
- **Offset:** 0x104

Bits	Name	Memory Access	Description
31:10	RSVD_IC_SAR3	R	IC_SAR3 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

9:0	IC_SAR3	R/W	The IC_SAR3 holds the slave address 3 when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR3[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. **Note:** The default values cannot be any of the reserved address locations: that is, 0x00 to 0x07, or 0x78 to 0x7f. The correct operation of the device is not guaranteed if you program the IC_SAR3 or IC_TAR to a reserved value. **Exists:** Always **Volatile:** true

12.2.69 IC_SAR4

- **Name:** I2C Slave Address Register 4
- **Description:** I2C Slave Address Register 4
- **Size:** 32 bits
- **Offset:** 0x108

Bits	Name	Memory Access	Description
31:10	RSVD_IC_SAR4	R	IC_SAR4 Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
9:0	IC_SAR4	R/W	The IC_SAR4 holds the slave address 4 when the I2C is operating as a slave. For 7-bit addressing, only IC_SAR4[6:0] is used. This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. **Note:** The default values cannot be any of the reserved address locations: that is, 0x00 to 0x07, or 0x78 to 0x7f. The correct operation of the device is not guaranteed if you program the IC_SAR4 or IC_TAR to a reserved value. **Exists:** Always **Volatile:** true

12.2.70 IC_CLR_WR_REQ

- **Name:** Clear WR_REQ Interrupt Register

- **Description:** Clear WR_REQ Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x11c

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_WR_REQ	R	IC_CLR_WR_REQ Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
0	CLR_WR_REQ	R	Read this register to clear the WR_REQ interrupt (bit 15) of the IC_RAW_INTR_STAT register. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.71 IC_CLR_SLV_ADDR_TAG

- **Name:** Clear CLR_SLV_ADDR_TAG Interrupt Register
- **Description:** Clear CLR_SLV_ADDR_TAG Interrupt Register
- **Size:** 32 bits
- **Offset:** 0x120

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_SLV_ADDR_TAG	R	IC_CLR_SLV_ADDR_TAG Reserved bits - Read Only **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true
0	CLR_SLV_ADDR_TAG	R	Read this register to clear all SLV_ADDRx_TAG (Where x is 1 through IC_NUM_SARS) interrupts (bit 16 ...(16+IC_NUM_SARS-1)) of the IC_RAW_INTR_STAT register. **Value After Reset:** 0x0 **Exists:** Always **Volatile:** true

12.2.72 IC_SAR2_SMBUS_UDID_WORD0

- **Name:** IC_SAR2 SMBUS ARP UDID WORD0 Register

- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR2_SMBUS_UDID_HC is set to 0. This register is used to store the Lower 32 bit value of IC_SAR2 Slave UDID register i.e. UDID[31:0] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0x124

Bits	Name	Memory Access	Description
31:0	SAR2_SMBUS_UDID_WORD0	R/W	This field is used to store the Lower 32 bit value of IC_SAR2 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.73 IC_SAR2_SMBUS_UDID_WORD1

- **Name:** IC_SAR2 SMBUS ARP UDID WORD1 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR2_SMBUS_UDID_HC is set to 0. This register is used to store the Middle-Lower 32 bit value of IC_SAR2 Slave UDID register i.e. UDID[63:32] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0x128

Bits	Name	Memory Access	Description
31:0	SAR2_SMBUS_UDID_WORD1	R/W	This field is used to store the Middle-Lower 32 bit value of IC_SAR2 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.74 IC_SAR2_SMBUS_UDID_WORD2

- **Name:** IC_SAR2 SMBUS ARP UDID WORD2 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR2_SMBUS_UDID_HC is set to 0. This register is used to store the Middle-Upper 32 bit value of IC_SAR2 Slave UDID register i.e. UDID[95:64] used in Address

Resolution Protocol of SMBus.

- **Size:** 32 bits
- **Offset:** 0x12c

Bits	Name	Memory Access	Description
31:0	SAR2_SMBUS_UDID_WORD2	R/W	This field is used to store the Middle-Upper 32 bit value of IC_SAR2 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.75 IC_SAR3_SMBUS_UDID_WORD0

- **Name:** IC_SAR3 SMBUS ARP UDID WORD0 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR3_SMBUS_UDID_HC is set to 0. This register is used to store the Lower 32 bit value of IC_SAR3 Slave UDID register i.e. UDID[31:0] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0x134

Bits	Name	Memory Access	Description
31:0	SAR3_SMBUS_UDID_WORD0	R/W	This field is used to store the Lower 32 bit value of IC_SAR3 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.76 IC_SAR3_SMBUS_UDID_WORD1

- **Name:** IC_SAR3 SMBUS ARP UDID WORD1 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR3_SMBUS_UDID_HC is set to 0. This register is used to store the Middle-Lower 32 bit value of IC_SAR3 Slave UDID register i.e. UDID[63:32] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits

- **Offset:** 0x138

Bits	Name	Memory Access	Description
31:0	SAR3_SMBUS_UDID_WORD1	R/W	This field is used to store the Middle-Lower 32 bit value of IC_SAR3 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.77 IC_SAR3_SMBUS_UDID_WORD2

- **Name:** IC_SAR3 SMBUS ARP UDID WORD2 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR3_SMBUS_UDID_HC is set to 0. This register is used to store the Middle-Upper 32 bit value of IC_SAR3 Slave UDID register i.e. UDID[95:64] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0x13c

Bits	Name	Memory Access	Description
31:0	SAR3_SMBUS_UDID_WORD2	R/W	This field is used to store the Middle-Upper 32 bit value of IC_SAR3 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.78 IC_SAR3_SMBUS_UDID_WORD3

- **Name:** IC_SAR3 SMBUS ARP UDID WORD3 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR3_SMBUS_UDID_HC is set to 0. This register is used to store the Upper 32 bit value of IC_SAR3 Slave UDID register i.e. UDID[127:96] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0x140

Bits	Name	Memory Access	Description
31:0	SAR3_SMBUS_UDID_WORD3	R/W	This field is used to store the Upper 32 bit value of IC_SAR3 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.79 IC_SAR4_SMBUS_UDID_WORD0

- **Name:** IC_SAR4 SMBUS ARP UDID WORD0 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR4_SMBUS_UDID_HC is set to 0. This register is used to store the Lower 32 bit value of IC_SAR4 Slave UDID register i.e. UDID[31:0] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0x144
-

Bits	Name	Memory Access	Description
31:0	SAR4_SMBUS_UDID_WORD0	R/W	This field is used to store the Lower 32 bit value of IC_SAR4 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.80 IC_SAR4_SMBUS_UDID_WORD1

- **Name:** IC_SAR4 SMBUS ARP UDID WORD1 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR4_SMBUS_UDID_HC is set to 0. This register is used to store the Middle-Lower 32 bit value of IC_SAR4 Slave UDID register i.e. UDID[63:32] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0x148

Bits	Name	Memory Access	Description

31:0	SAR4_SMBUS_UDID_WORD1	R/W	This field is used to store the Middle-Lower 32 bit value of IC_SAR4 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.81 IC_SAR4_SMBUS_UDID_WORD2

- **Name:** IC_SAR4 SMBUS ARP UDID WORD2 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR4_SMBUS_UDID_HC is set to 0. This register is used to store the Middle-Upper 32 bit value of IC_SAR4 Slave UDID register i.e. UDID[95:64] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0x14c

Bits	Name	Memory Access	Description
31:0	SAR4_SMBUS_UDID_WORD2	R/W	This field is used to store the Middle-Upper 32 bit value of IC_SAR4 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

12.2.82 IC_SAR4_SMBUS_UDID_WORD3

- **Name:** IC_SAR4 SMBUS ARP UDID WORD3 Register
- **Description:** This Register can be written only when the Apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SAR4_SMBUS_UDID_HC is set to 0. This register is used to store the Upper 32 bit value of IC_SAR4 Slave UDID register i.e. UDID[127:96] used in Address Resolution Protocol of SMBus.
- **Size:** 32 bits
- **Offset:** 0x150

Bits	Name	Memory Access	Description
31:0	SAR4_SMBUS_UDID_WORD3	R/W	This field is used to store the Upper 32 bit value of IC_SAR4 slave unique device identifier used in Address Resolution Protocol. **Exists:** Always

13.I2S

13.1 I2S Features

Inter-Integrated Circuit Sound (I2S) is a protocol for digital stereo audio. The I2S controller (I2SC) functional block for the SOC controls the I2S link (I2SLINK), which is a low-power four-pin serial interface for stereo audio.

The I2S controller consists of buffers, status and control Register, serializers, and counters for transferring digitized audio between the processor system memory and an external I2S CODEC.For playback of digitized audio or production of synthesized audio, the I2SC retrieves digitized audio samples from processor system memory and sends them to a CODEC through the I2SLINK. The external digital-to-analog converter in the CODEC then converts the audio samples into an analog audio waveform. For recording of digitiz- ed audio, the I2SC receives digitized audio samples from a CODEC (through the I2SLINK) and stores them in processor system memory. The I2S controller supports the normal-I2S formats.

I2S main features:

- A formatting or “Left/Right” control signal.
- Two serial audio pins, one input and one output.
- Data width is configurable.

13.2 I2S Register Descriptions

I2S0_REG_BASE: 0x50D0_0000
I2S1_REG_BASE: 0x50F0_0000

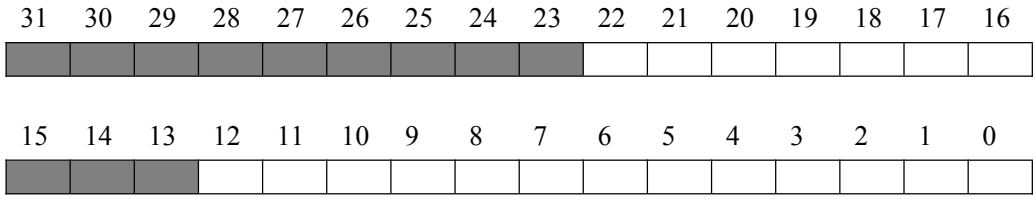
13.2.1 SACR0: Global Control Register

Note: when the Tx Fifo entries less than the threshold, I2S controller will send a dma request signal to DMA controller, then DMA controller start a burst transfer to I2S Tx FIFO. If the null entries less than the burst size, then some data will be discas by I2S controller for Tx FIFO full. So,

when we hire DMA transfer mode, please set correct threshold value as the following formula:

32 - threshold >= dma burst size.

Address: I2S_REG_BASE + 0x00

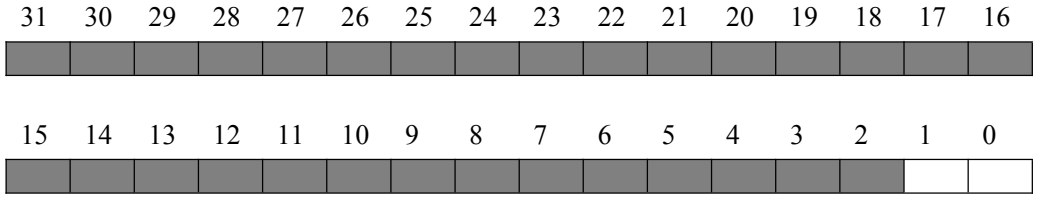


Bit	Type	Reset	Description
31-27	R	0	Not used.
26	R/W	1	**RFIFOFIRSTBIT** 0: right first 1: left first
25	R/W	1	**TFIFOFIRSTBIT** 0: right first 1: left first
24	R/W	0	**CHANLOCK** 0: no lock 1: lock channel(left first/right first)
23	R/W	0	**SCBIT** 0: normal 1: single channel
22	R/W	0	**I2SBIT** 1: 32 bit 0: 16 bit
21	R/W	0	**SYNCINV** 1: invert sync 0: not invert
20-16	R/W	F	**RFTH** Receive FIFO Interrupt: Set to value 0 – 31.This value must be set to the threshold value. Receive request asserted whenever the receive FIFO has >= RFTH entries.
15-13	R	0	Not used.
12-8	R/W	F	**TFTH** Set to value 0 – 31.This value must be set to the threshold value minus 1. Transmit request asserted whenever the receive FIFO has < (TFTH+1) entries.
7	R/W	0	DAC output clk edge select 0: data output at the positive edge of dac clk 1: data output at the middle of dac clk
6	R/W	0	**RDMAENA** Receive FIFO DMA transfer enable 0: disable. 1: enable.
5	R/W	0	**LOOPBACK:** 0 = Normal serial port operation enabled 1 = Output of transmit serial shifter is connected to input of receive serial shifter internally.
4	R/W	0	**RESET** Resets FIFOs logic and all Register, except this

			register(SACR0): 0: not reset. 1: reset is Active to Other Register.
3	R/W	0	**TDMAENA** Transmit FIFO DMA transfer enable: 0: disable. 1: enable.
2	R/W	0	**BCKD** BITCLK input/output: 0: input. 1: output.
1	R/W	0	**SYNCD** Word Select Clk input/output, if you hire inner dac, please clean this bit, else set this bit, ie (Using extend IIS Codec) 0: input. 1: output.
0	R/W	0	**ENB** Enable I²S Function: 0: I²S is disable 1: I²S is enable

13.2.2SACR1: Serial Audio I2S Control Register

Address: I2S_REG_BASE + 0x04



Bit	Type	Reset	Description
31-2	R	0	Not used.
1	R/W	1	**DRPL** Disable replaying function: 0: replaying Function is Enabled. 1: replaying Function is Disabled.
0	R/W	1	**DREC** Disable record function: 0: recording Function is Enabled. 1: recording Function is Disabled.

13.2.3SASR0: Serial Audio I2S and FIFO Status Register

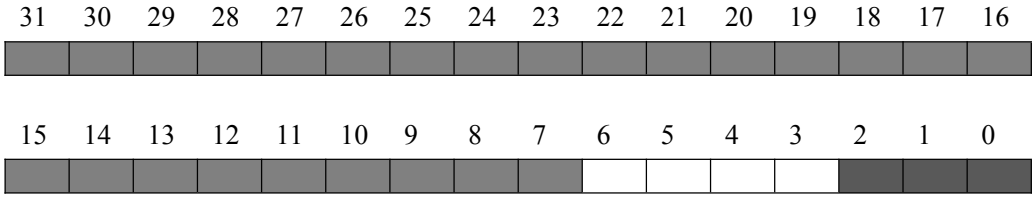
Address: I2S_REG_BASE + 0x0c

Bit	Type	Reset	Description
31-22	R	0	Not used.
21-16	R	/	**RFL**

			Receive FIFO level. Number of entries in receives FIFO.
15-14	R	/	Not used
13-8	R	/	**TFL** Transmit FIFO level. Number of entries in transmits FIFO.
7	R	0	Not used.
6	R	/	**ROR** Receive FIFO overrun: 0: receive FIFO has not experienced an overrun. 1: I^2S attempted data write to full receive FIFO. Can interrupt processor if bit6 of Serial Audio Interrupt Mask Register is set. Cleared by setting bit 6 of Serial Audio Interrupt Clear Register.
5	R	/	**TUR** Transmit FIFO under-run: 0: Transmit FIFO has not experienced an under-run. 1: I^2S attempted data read from an empty transmit FIFO. Can interrupt processor if bit5 of Serial Audio Interrupt Mask Register is set. Cleared by setting bit 5 of Serial Audio Interrupt Clear Register.
4	R	/	**RFS** Receive FIFO service request: 0: receive FIFO level below RFL threshold, or I^2S disabled. 1: receive FIFO level is at or above RFL threshold. Can interrupt processor if bit4 of Serial Audio Interrupt Mask Register is set. Cleared automatically when # of receive FIFO entries<(RFTH+1).
3	R	/	**TFS** Transmit FIFO service request: 0: transmit FIFO level exceeds TFL threshold, or I^2S disabled. 1: transmit FIFO level is at or below TFL threshold. Can interrupt processor if bit4 of Serial Audio Interrupt Mask Register is set. Cleared automatically when # of transmit FIFO entries>=(TFTH+1).
2	R	/	**BSY** I^2S busy: 0: I^2S is idle or disabled. 1: I^2S currently transmitting or receiving a frame.
1	R	/	**RNE** Receive FIFO not empty: 0: receive FIFO is not empty. 1: receive FIFO is empty.
0	R	/	**TNF** Transmit FIFO not full: 0: transmit FIFO is not full. 1: transmit FIFO is full.

13.2.4SAIMR: Serial Audio Interrupt Mask Register

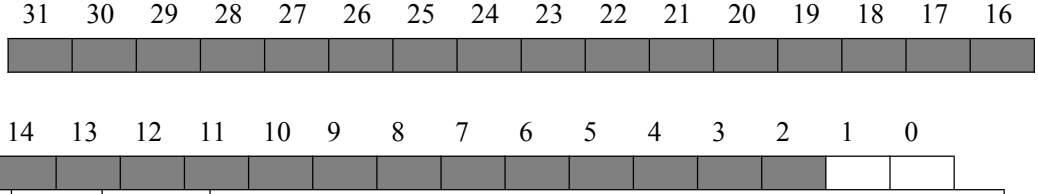
Address: I2S_REG_BASE + 0x14



Bit	Type	Reset	Description
31-7	R/W	0	Not used.
6	R/W	0	**ROR** Enable receive FIFO overrun condition based interrupt. 0: Disable 1: Enable
5	R/W	0	**TUR** Enable transmit FIFO under-run condition based interrupt. 0: Disable 1: Enable
4	R/W	0	**RFS** Enable receive FIFO service request based interrupt. 0: Disable 1: Enable
3	R/W	0	**TFS** Enable transmit FIFO service request based interrupt. 0: Disable 1: Enable
2-0	R	0	**Not use**

13.2.5SAICR: Serial Audio Interrupt Clear Register

Address: I2S_REG_BASE + 0x18



6	W	0	**ROR** Clear receive FIFO overrun interrupt and ROR status bit in SASR0. Writing bit 1 to this field, will clear ROR interrupt factor, this bit will not be auto clean, therefore, you need write bit 0 follow the bit 1 writing.
5	W	0	**TUR** Clear transmit FIFO under-run interrupt and TUR status bit in SASR0. Writing bit 1 to this field, will clear ROR interrupt factor, this bit will not be auto clean, therefore, you need write bit 0 follow the bit 1 writing.
4	W	0	**RXCR** **Clear** receive FIFO service request based interrupt. Writing bit 1 to this field, will clear ROR interrupt factor, this bit will not be auto clean, therefore, you need write bit 0 follow the bit 1 writing.
3	W	0	**TXCR** **Clear** ransmit FIFO service request based interrupt. Writing bit 1 to this field, will clear ROR interrupt factor, this bit will not be auto clean, therefore, you need write bit 0 follow the bit 1 writing.
2-0	R	0	**Not used.**

13.2.6SADR: Serial Audio Data Register (TX and RX FIFO Access Register)

Address: I2S_REG_BASE + 0x80

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

Bit	Type	Reset	Description
31-16	R/W	0	**DTH** Right data sample.
15-0	R/W	0	**DTL** Left data sample.

14.PWM

14.1 PWM Features

Support generate the clock with the user-defined frequency and duty cycle.

Support eight-channel and each can operate independently.

14.2 Register Descriptions (x=0, 1, 2, 3, 4, 5, 6, 7)

PWM_REG_BASE: 0x50B0_0000

The channel 0 Register started from **PWM_REG_BASE + 0x00;**

The channel 1 Register started from **PWM_REG_BASE + 0x10;**

The channel 2 Register started from **PWM_REG_BASE + 0x20;**

The channel 3 Register started from **PWM_REG_BASE + 0x30;**

The channel 4 Register started from **PWM_REG_BASE + 0x40;**

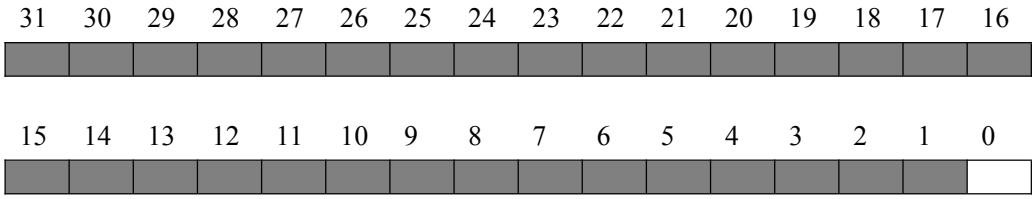
The channel 5 Register started from **PWM_REG_BASE + 0x50;**

The channel 6 Register started from **PWM_REG_BASE + 0x60;**

The channel 7 Register started from **PWM_REG_BASE + 0x70;**

14.2.1PWMENAx: PWM Channel x Enable

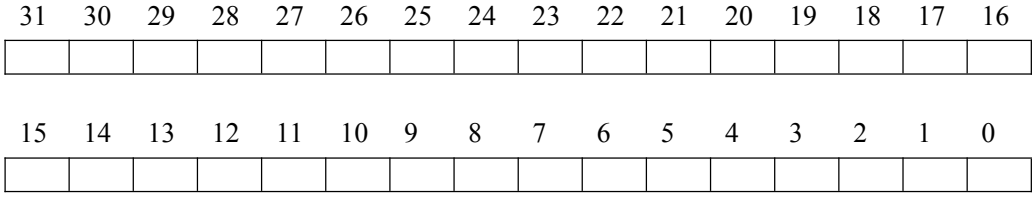
Address: PWM_REG_BASE + 0x00



Bit	Type	Reset	Description
31-1	R	0	Not used.
0	R/W	0x0	PWM Channel x Enable Control, when disabled PWMOUT[x] is low.

14.2.2PWMDUTYx: PWM Channel x Duty Counter

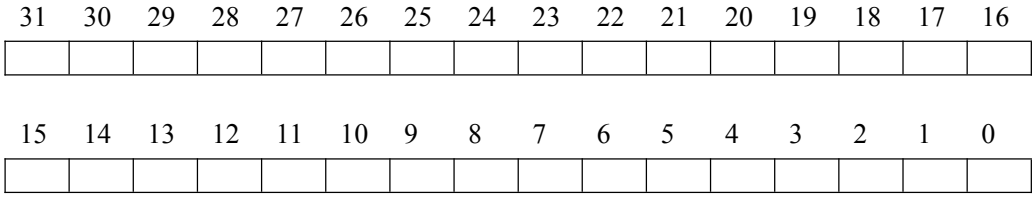
Address: PWM_REG_BASE + 0x04



Bit	Type	Reset	Description
31-0	R/W	0x0	PWM Channel x Duty register. It is the duty time in the cycle divided from input clock according to the value in PWMCNTRx

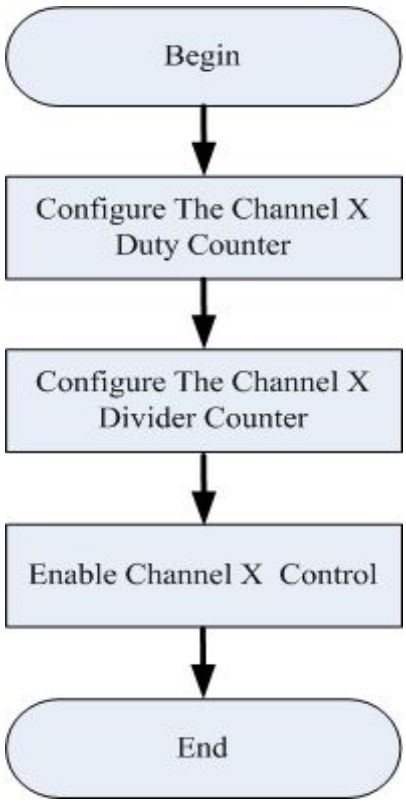
14.2.3PWMCNTRx: PWM Channel x Divider Counter

Address: PWM_REG_BASE + 0x08



Bit	Type	Reset	Description
31-0	R/W	0x1	PWM Channel x Divider register. Which is used to divide the input clock/PCLK.

14.3Software Flow



PWM-Channel configure flow

14.5 PWM Capture Register Description

PWM_CAP_REG_BASE: 0x50B0_0100

NAME	TYPE	OFFSET	DESCRIPTION
Sys_frq_1	R/W	0x00*4	Sys_frq: Support up to maximum 512M clock(at most support up to 500M for later mul logic) Cap_setting: [31]: cap_en [30]: cap_method. 0: cap_num; 1: cap_exist [29:28]: int_method. 0: no interrupt; 1: captured once to generate interrupt; 2'b10: cap once finished(trigger_finished) generate interrupt; 2'b11: finish all to generate interrupt [27:24]: glitch setting, glitch used to be glitch_setting + 1;(glitch setting should be less than 4'hf,between more than 1) [23:16]: cat_times,for cap_num method; support up to 255 cycle; 0, cap continuously; default continuously for cap exist,support once or twice(only support 1 or 2 setting,default once) [15]: default, no used! [14:12]: cap_based_unit, 3'h4: 1000ms(1s); 3'h2: 100ms; 3'h1: 10ms; 3'h0: 1ms(default); [11:8]: reserved [7:0]: cap_interval,based on cap_based_unit, support up to 127 * cap_based_unit, at least one, bit[7]: not used Cycle_cap: Current captured cycles, for continuously, for last captured cycles. Frq_cap: Current captured frequency, based on HZ; up to 16M Cap_int: When once trigger method(both cap_num and cap_exist method), sending interrupt to inform software
Sys_frq_1	R/W	0x01*4	
Cycle_cap_1	R/W	0x02*4	
Frq_cap_1	R/W	0x03*4	
Sys_frq_2	R/W	0x04*4	
Cap_setting_2	R/W	0x05*4	
Cycle_cap_2	R/W	0x06*4	
Frq_cap_2	R/W	0x07*4	
Sys_frq_3	R/W	0x08*4	
Cap_setting_3	R/W	0x09*4	
Cycle_cap_3	R/W	0x0a*4	
Frq_cap_3	R/W	0x0b*4	
Sys_frq_4	R/W	0x0c*4	
Cap_setting_4	R/W	0x0d*4	
Cycle_cap_4	R/W	0x0e*4	
Frq_cap_4	R/W	0x0f*4	
Cap_int_clear	R/W	0x10*4	
Cap_int_en	R/W	0x20*4	
Cap_int	R	0x21*4	

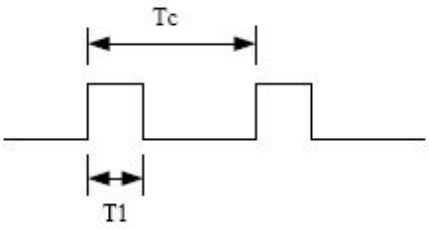
15. Remote Controller

15.1 Features

Support NEC mode and RC-5.

1) NEC with simple repeat code

A single pulse, modulated with 37.91KHZ signal at 455KHZ

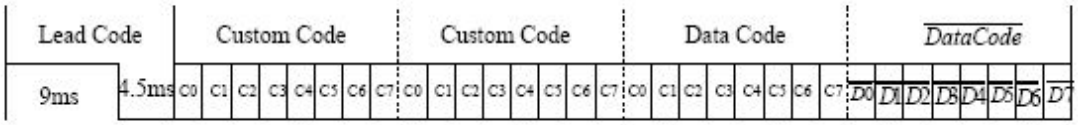


Carrier frequency

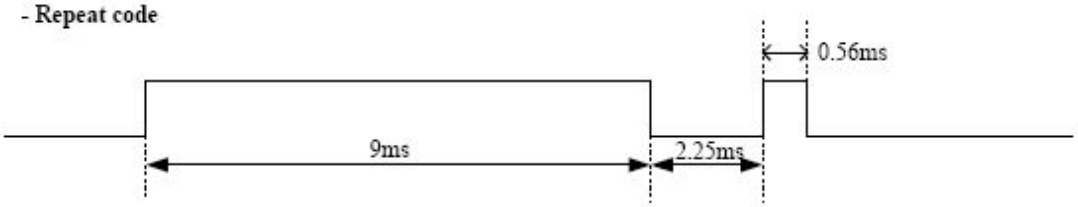
fcar =1/Tc =fosc/12
Duty ratio = T1/Tc = 1/3

- **Configuration of frame**

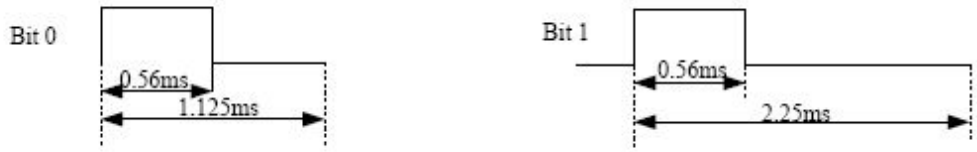
1st frame



- **Repeat code**

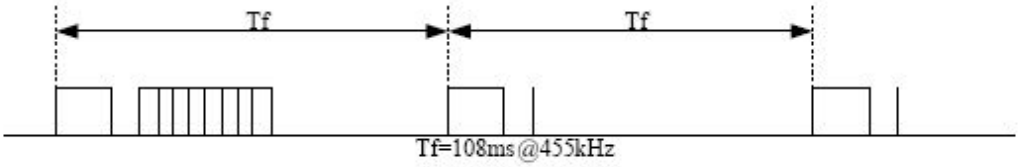


- **Bit Description**



- **Frame Interval : Tf**

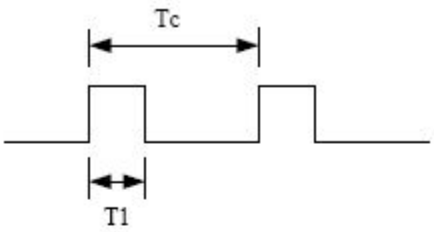
The transmitted waveform as long as a key is depressed



NEC mode timing

2) RC-5

A single pulse, modulated with 37.917KHZ signal at 455KHZ

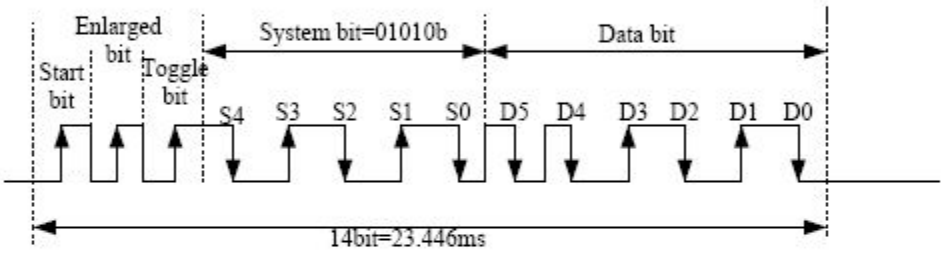


Carrier frequency

fcar =1/Tc =fosc/12
Duty ratio = T1/Tc = 1/3

- Configuration of frame

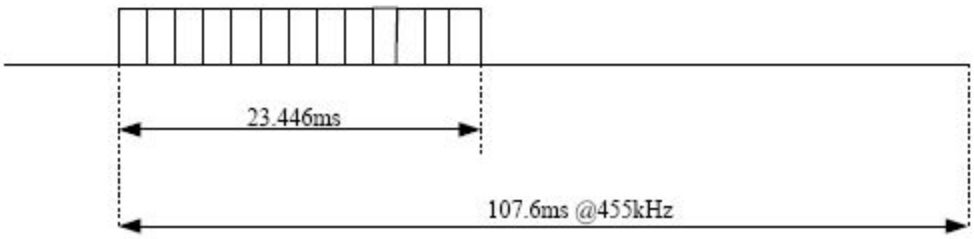
1st frame



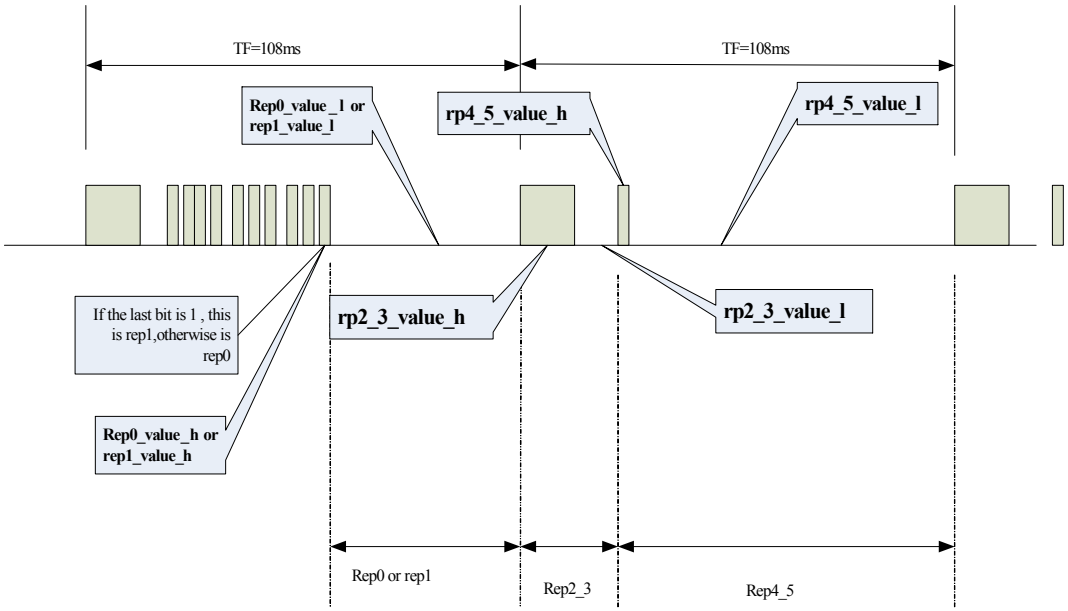
- Bit Descreption

- Frame Interval : Tf

The transmitted waveform as long as a key is depressed



RC-5 mode timing



repeat frame timing

15.2Remote Controller Register Descriptions

RC_BASE: 0x51C00000

15.2.1RCRT_DATA0

Address: RC_BASE + 0x00

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

Bit	Type	Reset	Description
D31-D24	R/W	0	R_FILTERTIME The filter width of input pulse.
D23-D15	R/W	0	R_PREDIV Divide the clock (24M) used for counter: 0: clock; 1: clock/4; 2: clock/6; .. n: clock/(n+1)×2.
D14-D7	R/W	0	R_DISPERSION The dispersion of the hardware examination pulse width.
D6-D4	R/W	0	R_VALID_RANGE Select the valid range of the data used by hardware.
D3	R/W	0	RC5_CS 0: NEC remote code. 1: RC5 remote code.
D2	R/W	0	DATA_POLARITY

			0: don't converse key value. 1: converse the key value.
D1	R/W	0	CUSTOM_POLARITY 0: don't converse the custom code. 1: converse the custom code.
D0	R/W	0	PULSE_POLARITY 0: don't converse the input pulse's polarity. 1: converse the input pulse's polarity.

15.2.2RCRT_DATA1

The negtive or postive plus width for all the start frame or rep stages calculated as the following forum:

$$plus_width = \frac{t_{plus_time} * f_{input}}{(R_PREDIV + 1) * 2} >> R_VALID_RANGE$$

Address: RC_BASE + 0x04

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

Bit	Type	Reset	Description
D31-D24	R/W	0	R_STARTL When RC5_CS=1, the value represents the negative pulse width of a RC5 code bit. Otherwise it represents the low pulse width of start code.
D23-D16	R/W	0	R_STARTH When RC5_CS=1, the value represents 1/2 of the positive pulse width of a RC5 code bit. Otherwise it represents the high pulse width of start code.
D15-D8	R/W	0	reserve
D7-D0	R/W	0	DISPERSION_RP_5L_DELTA The dispersion of the hardware examination pulse width, which is the delta only for repeat 5 low. (DISPERSION_RP_5L_DELTA+ R_DISPERSION used for repeat 5 low)

15.2.3RCRT_DATA2

Address: RC_BASE + 0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

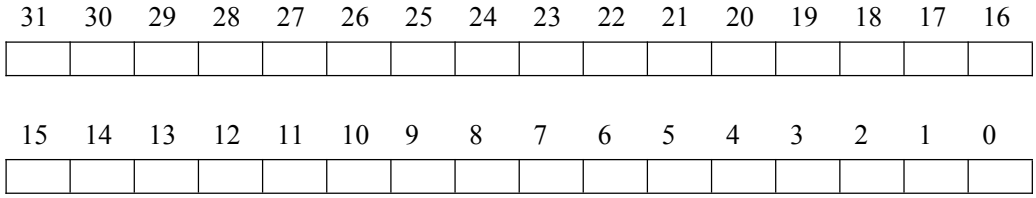
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
D31-D24	R/W	0	R_DATA0_NEG

			The width of negative pulse of “0” code.
D23-D16	R/W	0	R_DATA0_POS The width of positive pulse of “0” code.
D15-D8	R/W	0	R_DATA1_NEG The width of negative pulse of “1” code.
D7-D0	R/W	0	R_DATA1_POS The width of positive pulse of “1” code.

15.2.4RCRT_DATA3

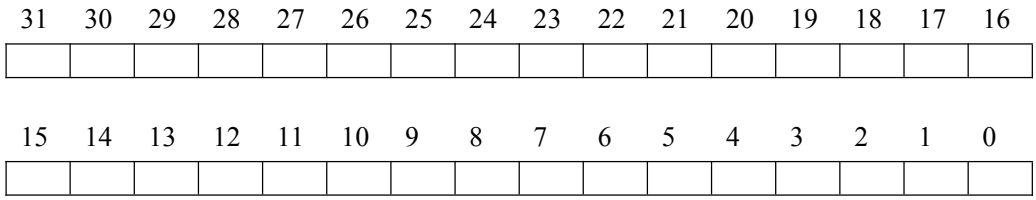
Address: RC_BASE + 0x0c



Bit	Type	Reset	Description
D31-D0	R/W	0	reserve

15.2.5RCRT_DATA4

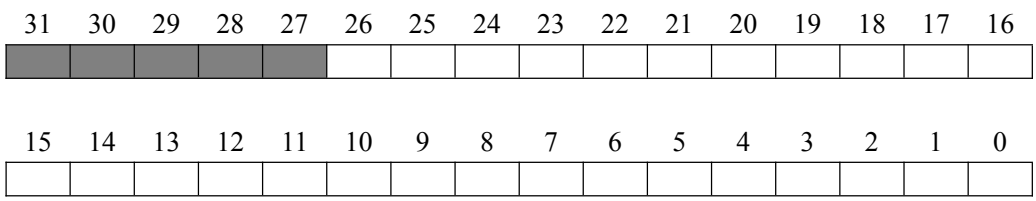
Address: RC_BASE + 0x10



Bit	Type	Reset	Description
D31-D24	R/W	0	R_RP45_NEG The width of negative pulse of “rp45” status, which follows the “rp23”.
D23-D16	R/W	0	R_RP45_POS The width of positive pulse of “rp45” status, which follows the “rp23”.
D15-D8	R/W	0	R_RP23_NEG The width of negative pulse of “rp23” status, which follows the “rp1”.
D7-D0	R/W	0	R_RP23_POS The width of positive pulse of “rp23” status, which follows the “rp1”.

15.2.6RCRT_DATA5

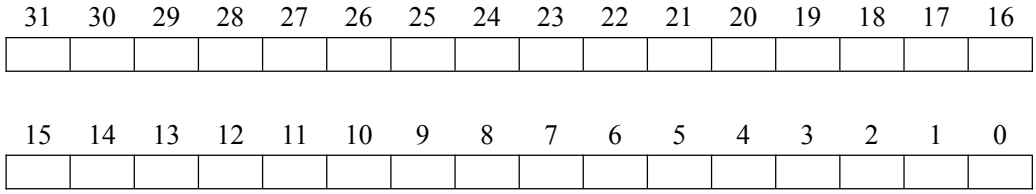
Address: RC_BASE + 0x14



Bit	Type	Reset	Description
D31-D27	-	-	Not used.
D26	R/W	0	Nec_release_int_en RC generates interrupt when release status is detected while the NEC remote code is used. 0: disable 1: enable
D25-D24	R/W	0	MODE_SEL 00: Not used. 01: Software decode mode. 10: Hardware & software decode mode. 11: Hardware decode mode
D23-D16	R/W	0	R_INT_NUM Set the interrupt position. (It's for software decoder)
D15-D0	R/W	0	R_RELEAE_TIME Set the key release time.

15.2.7RCRT_DATA6

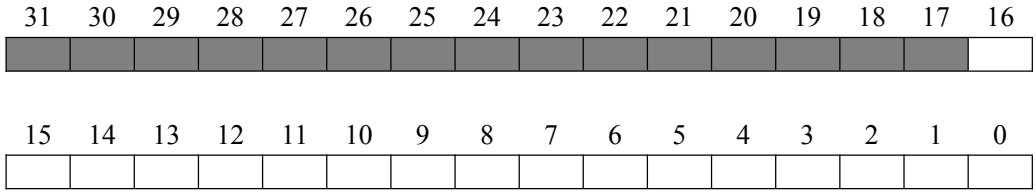
Address: RC_BASE + 0x18



Bit	Type	Reset	Description
D31-D24	R/W	0	custom_code2_h the up 8 bit of the 2nd custom code
D23-D16	R/W		custom_code2_l the low 8 bit of the 2nd custom code
D15-D8	R/W	0	custom_code1_h the up 8 bit of the first custom code
D7-D0	R/W	0	custom_code1_l the low 8 bit of the first custom code

15.2.8RCRT_DATCPU

Address: RC_BASE + 0x1c

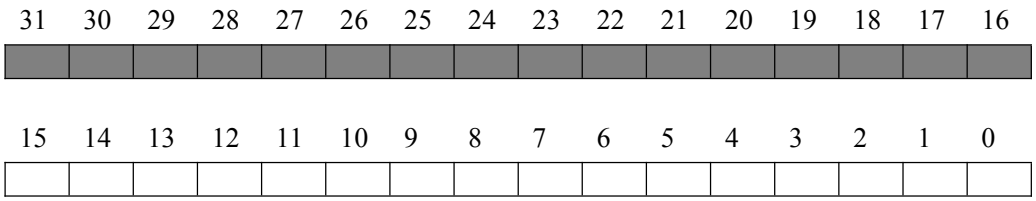


Bit	Type	Reset	Description
D31-D17	-	-	Not used.
D26	R/W	0	Data_not_jud_sel_n Data not code judgement select. 0: data not code must judge, when the data not code captured invert with the data code capture before, the remote is ok. 1: data not code need not judge, the remote is ok.

D25	R/W	0	Custom_not_jud_sel_n Custom code 2 judgement select. 0: custom code 2 must judge, when the custom code 2 captured same with the register value, the remote is ok. 1: custom code 2 need not judge, the remote is ok.
D24	R/W	0	Custom_jud_sel_n Custom code 1 judgement select. 0: custom code 1 must judge, when the custom code 1 captured same with the register value, the remote is ok. 1: custom code 1 need not judge, the remote is ok.
D23-D17	R/W	0	Reserve
D16	R/W	0	user_code_sel_reg 0: 8 bit user code 1: 13 bit user code
D15-D0	R/W	0	NEC_bit_2_pulse: pulse width of two pulse of NEC code

15.2.9CODE_VALUE_BUFF

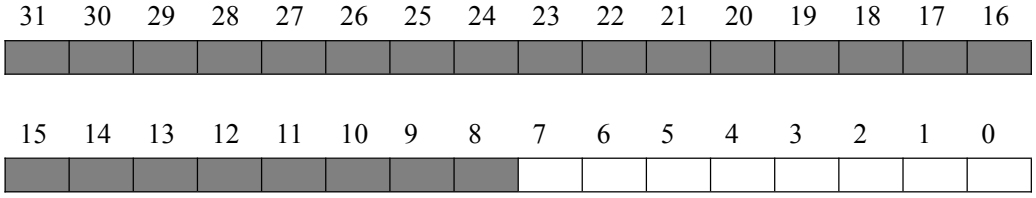
Address: RC_BASE + 0x20



Bit	Type	Reset	Description
D31-D16	-	-	Not used.
D15-D8	R		Pulse width of “1”
D7-D0	R	0	Pulse width of “0”

15.2.10CODE_VALUE

Address: RC_BASE + 0x24

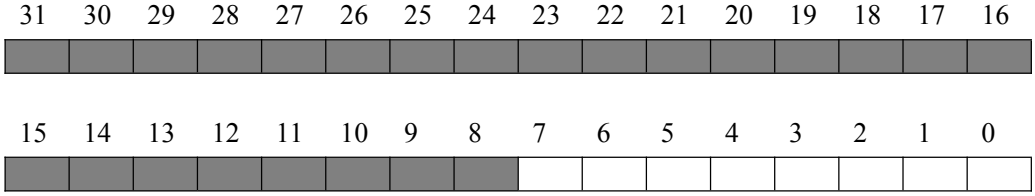


Bit	Type	Reset	Description
D31-D8	-	-	Not used.
D7-D0	R	0	CODE_VALUE This register reserves code status, can be used for hardware decoder or software decoder. The detail as follows: 0x00: “0” receive; 0x01: “1” receive; 0x02: start receive; 0x03: data end as 1; 0x04: data end as 0; 0x05: simple repeat start;

			0x06: simple repeat continue; 0xff: Default

15.2.11KEY_VALUE

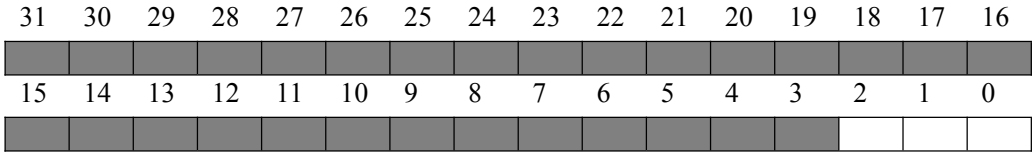
Address: RC_BASE + 0x28



Bit	Type	Reset	Description
D31-D8	-	-	Not used.
D7-D0	R/W	0	KEY_VALUE This register reserve the value decoded through the input pulse by the hardware.

15.2.12RCRT_STATE

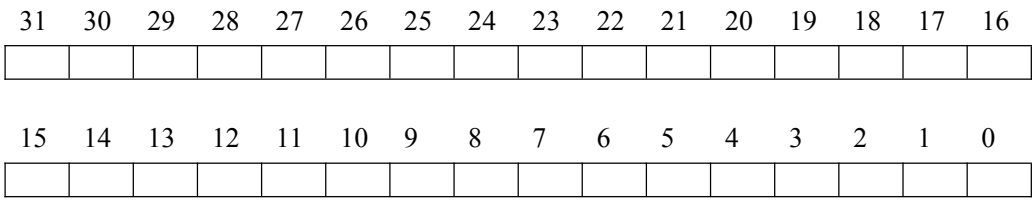
Address: RC_BASE + 0x2c



Bit	Type	Reset	Description
D31-D4	-	-	Not used.
D3	0	0	CODE_DETECT 0: None code is detected 1: Some code event is detected, but can not assure which one of it for key press, key repeat or key release.
D2	R	0	PULSE_POL 0: Negative pulse detected. 1: Positive pulse detected.
D1	R	0	REPEAT_FLAG 0: No repeat state detected. 1: Repeat state detected.
D0	R	0	RELEASE_FLAG 0: No release state detected. 1: Release state detected.

15.2.13RCRT_CUSTOM_CODE_3_4

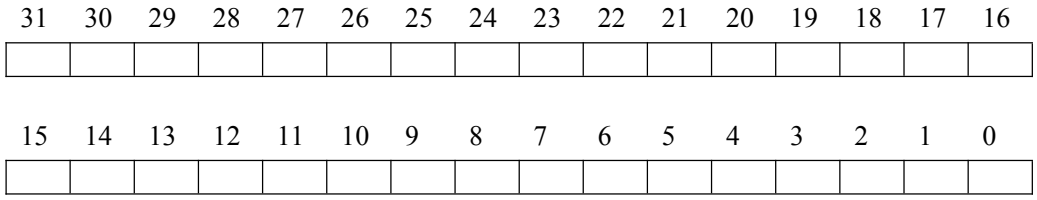
Address: RC_BASE + 0x30



Bit	Type	Reset	Description
D31-D24	R/W	00	High 8 bits for custom code 4(similar with custom code 2)
D23-D16	R/W	00	Low 8 bits for custom code 4(similar with custom code 2)
D15-D8	R/W	00	High 8 bits for custom code 3(similar with custom code 1)
D7-D0	R/W	00	Low 8 bits for Custom code 3(similar with custom code 1)

15.2.14RCRT_INTR_REQ_CLR

Address: RC_BASE + 0x34



Bit	Type	Reset	Description
D31-D0	R/W	00	Write the register to any value to clear the rcrt interrupt.

16. LCD Controller

16.1 Features

Support one video layers, one OSD layer.

Support overlay and per-pixel alpha blending between two channels, each channel can be stopped separately.

Support ARGB888/RGB666/RGB565/YUV420/YUV422 RGB Interface.

Support ITU656/601/1120/SRGB/CPU panel interface.

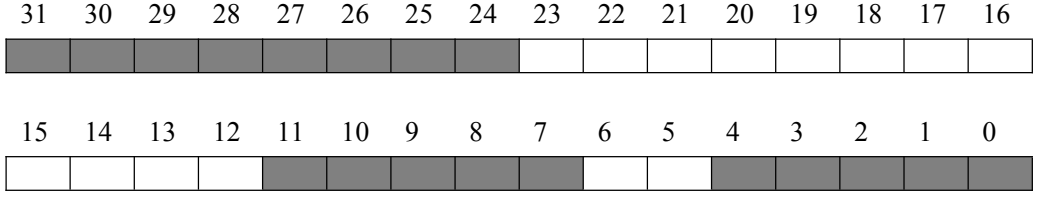
16.2 Register Descriptions

LCD_BASE: 0x40300000

LCD_PARAM0

LCD timing parameter

Offset_Address: 0x0000_0000



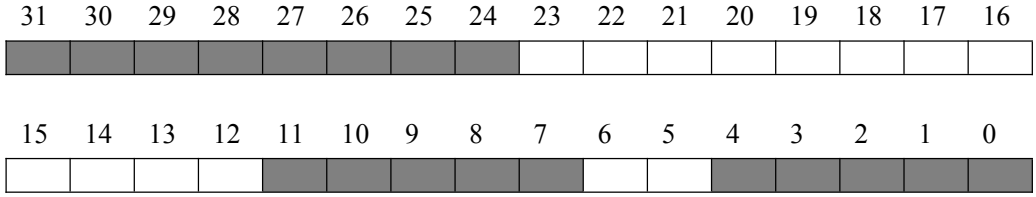
Bit	Type	Reset	Description

31	R/W	0	**Stop_lcd** **LCD停止，在lcd_done时，disable lcd_enable信号。**
30	R/W	0	V_syn_intr_enable 场同步中断使能，1有效。
29	R/W	0	Reserved
28	R/W	0	Reserved
27	R/W	0	Reserved
26	R/W	0	Lcd_done_intr_enable LCD done中断使能，1有效
25:22	R/W	0	Range_coeff_uv (Rersverd) U V乘上的参数
21:18	R/W	0	Range_coeff_y (Rersverd) Y乘上的参数
17	R/W	0	Reserved
16	R/W	0	Lcd_vsyn_pos_intr_enable LCD场方向任意位置中断使能，高有效。 此中断主要用来出场同步之前几行的中断，具体位置由寄存器lcd_vsyn_pos_num, lcd_vsyn_pos_num_v1决定。
15:13	R/W	0	Rgb_pad_mode 输出RGB顺序。 000：RGB 001：RBG 010：GRB 011：GBR 100：BRG 101：BGR 其它：RGB
12:1	R/W	0	Screen_width 显示屏宽，屏上显示数据的水平像素点数
0	R/W	0	Lcd_enable LCD使能信号，为1时LCD模块工作，为0时，LCD模块不工作。 1：使能； 0：无效。

LCD_PARAM1

LCD timing parameter

Offset_Address: 0x0000_0004



Bit	Type	Reset	Description

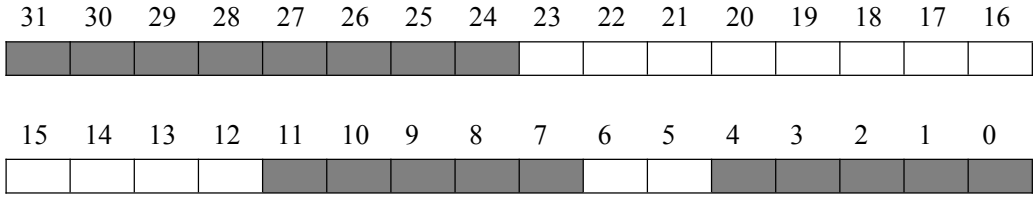
31:25	R/W	0	**Reserved**
24	R/W	0	Hv_manual_disable 行场等信号手动关闭使能 1：手动关闭，为1时不输出行场等信号； 0：为0时正常输出。
23	R/W	0	Hv_auto_disable 行场等信号自动关闭使能，当输出为CVBS，YPbPr时，自动将行场等信号关闭。 1：自动关闭； 0：需要手动关闭。
22	R/W	0	Itu656_out_sel ITU656输出模式选择，一般情况下选0。 0：普通ITU656； 1：逐行ITU656。
21	R/W	0	Rgb2yuv_mode_sel RGB到YUV转换时，Y的数据选择，一般情况下配1。 1：+16（Y最小值为16）； 0：+0。
20	R/W	0	Rgb2yuv_en RGB到YUV的转换使能，1有效。当输出为CVBS，YPbPr，ITU656等YUV信号输出是，该位置1，否则置0。
19	R/W	0	Lcd_ac_ivs 输出有效数据信号data_en是否取反选择，当为CPU屏时表示是否将CPU_screen_cs取反 1：取反； 0：不取反。
18	R/W	0	Vsync_ivs 输出场同步信号是否取反选择，当为CPU屏时表示是否将CPU_screen_rs取反 1：取反； 0：不取反。
17	R/W	0	Hsync_ivs 输出行同步信号是否取反选择，当为CPU屏时表示是否将CPU_screen_rd取反 1：取反； 0：不取反。
16	R/W	0	Field_ivs 输出field信号是否取反选择 1：取反； 0：不取反。
15	R/W	0	Reserved
14	R/W	0	Reserved
13	R/W	0	VSW1_enable 奇偶场时，用来表示另外一场的使能，因此，当隔行输出时，该位置1。
12：4	R/W	0	Reserved
3:1	R/W	0	Screen_type

			输出屏类型选择，当输出为CVBS，YPbPr，VGA时，选择并行数据屏类型。 001：并行数据屏； 010：串行Srgb屏； 011：ITU656输出； 101：并行数据屏； 110：串行CPU屏；
0	R/W	0	Lcd_interlace_flag LCD隔行取数模式，从memory中取数方式，一般当输出为隔行CVBS，隔行YPbPr，itu656，驱动隔行屏时，该位置1，否则置0。 1：隔行取数； 0：逐行取数。

LCD_PARAM2

LCD timing parameter –back_color

Offset_Address: 0x0000_0008

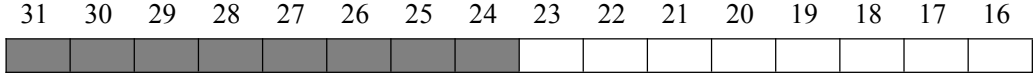


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
25	R/W		**Osd1clk_disable**
24	R/W	0	**Osd0_clk_disable**
23:16	R/W	0	Back_color_r 背景颜色，R分量
15:8	R/W	0	Back_color_g 背景颜色，G分量
7:0	R/W	0	Back_color_b 背景颜色，B分量

LCD_PARAM3

LCD timing parameter –DEN_h_position

Offset_Address: 0x0000_000C

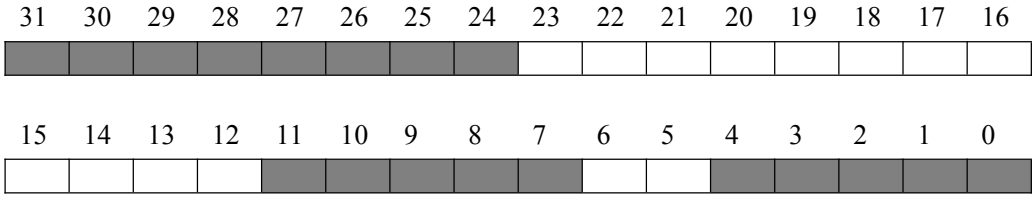


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	DEN_h_fall 数据使能信号data_en水平结束位置。
11:0	R/W	0	DEN_h_rise 数据使能信号data_en水平起始位置。

LCD_PARAMCU

LCD timing parameter –DEN0_v_position

Offset_Address: 0x0000_0010

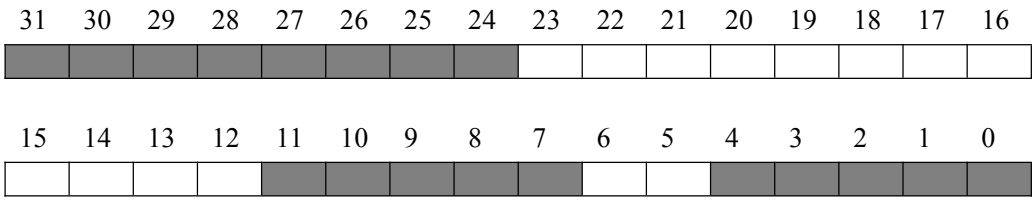


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	DEN0_v_fall 数据使能信号data_en垂直结束位置，逐行时序时，为每帧的垂直时序，隔行时序时，为第一场的时序。
11:0	R/W	0	DEN0_v_rise 数据使能信号data_en垂直起始位置，逐行时序时，为每帧的垂直时序，隔行时序时，为第一场的时序。

LCD_PARAM5

LCD timing parameter –DEN1_v_position

Offset_Address: 0x0000_0014



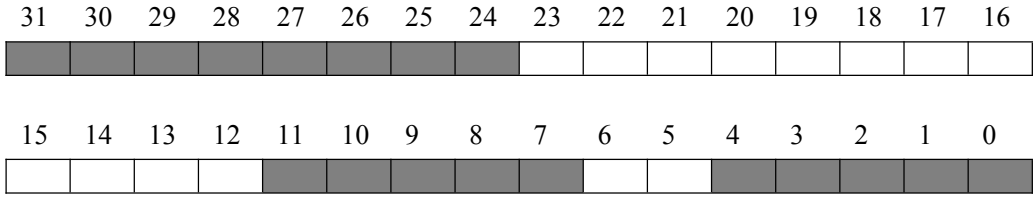
Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	DEN1_v_fall

			数据使能信号data_en垂直结束位置，当隔行输出时（VSW1_enable=1）有效，代表的是第二场的垂直结束位置，逐行输出时无效。
11:0	R/W	0	DEN1_v_rise 数据使能信号data_en垂直起始位置，当隔行输出时（VSW1_enable=1）有效，代表的是第二场的垂直起始位置，逐行输出时无效。

LCD_PARAM6

LCD timing parameter

Offset_Address: 0x0000_0018

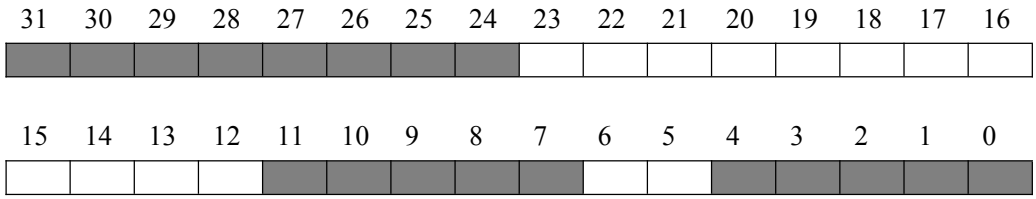


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	LPS-lines per scan frame 每帧的总行数，隔行时序时为奇偶场的总行数。
11:0	R/W	0	CPL – clocks per line 每行的总点数

LCD_PARAM7

LCD timing parameter –hsw_position

Offset_Address: 0x0000_001C

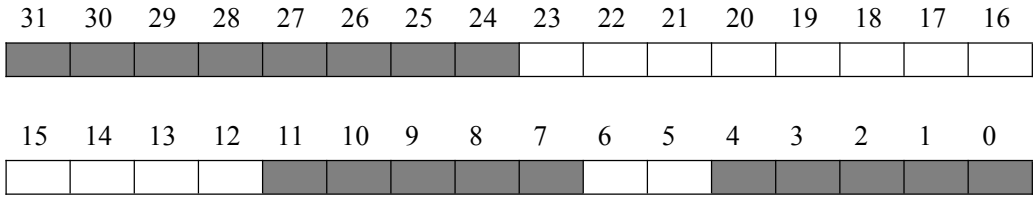


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	HSW_fall 行同步信号hsyn下降沿位置
11:0	R/W	0	HSW_rise 行同步信号hsyn上升沿位置

LCD_PARAM8

LCD timing parameter –vsw0_position

Offset_Address: 0x0000_0020

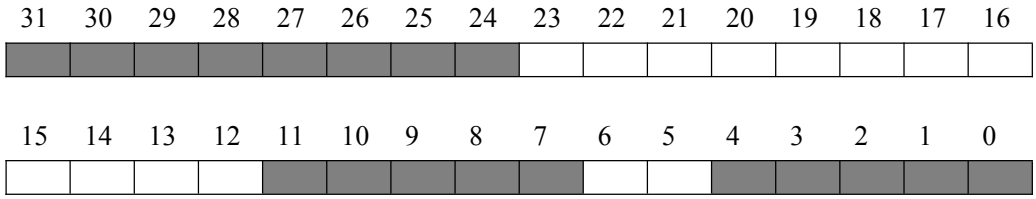


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	VSW0_v_fall 场同步信号vsyn0下降沿位置，逐行时序时为场同步，隔行时序时为第一场的场同步。
11:0	R/W	0	VSW0_v_rise 场同步信号vsyn0上升沿位置，逐行时序时为场同步，隔行时序时为第一场的场同步。

LCD_PARAM9

LCD timing parameter –vsw0_position

Offset_Address: 0x0000_0024

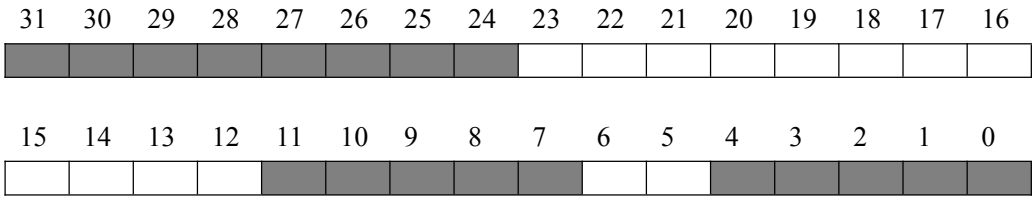


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	VSW0_h_fall 场同步信号vsyn0下降沿位置，逐行时序时为场同步，隔行时序时为第一场的场同步。通过VSW0_h_rise VSW0_h_fall可以调节场同步的具体位置，且奇偶场也可以出半行。
11:0	R/W	0	VSW0_h_rise 场同步信号vsyn0上升沿位置，逐行时序时为场同步，隔行时序时为第一场的场同步。通过VSW0_h_rise VSW0_h_fall可以调节场同步的具体位置，且奇偶场也可以出半行。

LCD_PARAM10

LCD timing parameter –vsw1_position

Offset_Address: 0x0000_0028

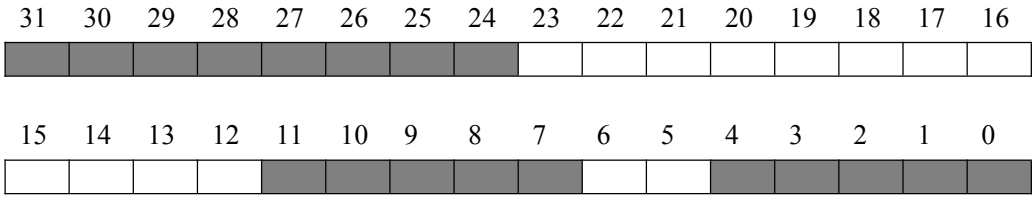


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	VSW1_v_fall 场同步信号vsyn1下降沿位置，逐行时序时无效(VSW1_enable=0)，隔行时序时为第二场的场同步（VSW1_enable=1）。
11:0	R/W	0	VSW1_v_rise 场同步信号vsyn1上升沿位置，逐行时序时无效(VSW1_enable=0)，隔行时序时为第二场的场同步（VSW1_enable=1）。

LCD_PARAM11

LCD timing parameter –vsw1_position

Offset_Address: 0x0000_002C



Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	VSW1_h_fall 场同步信号vsyn1下降沿位置，逐行时序（VSW1_enable=0）时无效，隔行时序(VSW1_enable=1)时为第一场的场同步。通过VSW1_h_rise VSW1_h_fall可以调节场同步的具体位置，且奇偶场也可以出半行。
11:0	R/W	0	VSW1_h_rise 场同步信号vsyn1上升沿位置，逐行时序（VSW1_enable=0）时无效，隔行时序(VSW1_enable=1)时为第一场的场同步。通过VSW1_h_rise VSW1_h_fall可以调节场同步的具体位置，且奇偶场也可以出半行。

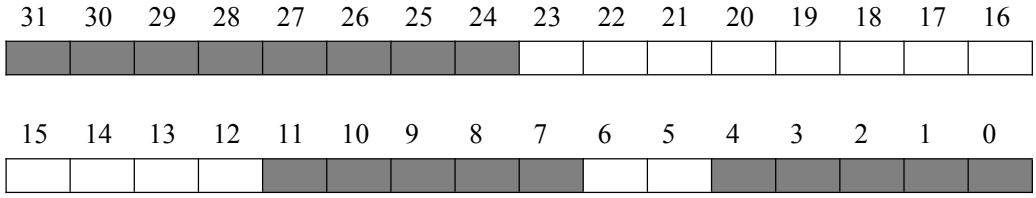
LCD_PARAM12

RGB to YCbCr coef0，计算 Y 时 R G B 的系数。

Y =（ R * RGB2YUV_coeff00 + G * RGB2YUV_coeff01 + B * RGB2YUV_coeff02）/256 +

Rgb2yuv_mode_sel * 16;

Offset_Address: 0x0000_0030



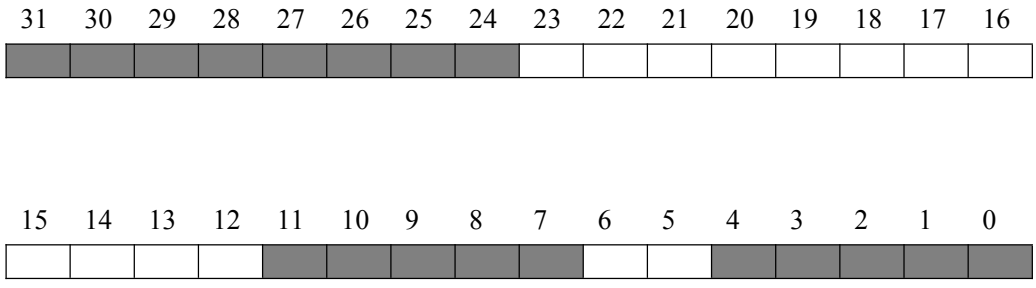
Bit	Type	Reset	Description
31:27	R/W	0	**Reserved**
26:18	R/W	1D	RGB2YUV_coeff02 RGB到YUV转换时，B的系数。
17:9	R/W	96	RGB2YUV_coeff01 RGB到YUV转换时，G的系数。
8:0	R/W	4D	RGB2YUV_coeff00 RGB到YUV转换时，R的系数。

LCD_PARAM13

RGB to YCbCr coef1，计算 Cb 时 R G B 的系数。

Cb =（ - R * RGB2YUV_coeff10 - G * RGB2YUV_coeff11 + B * RGB2YUV_coeff12）/256 + 128

Offset_Address: 0x0000_0034



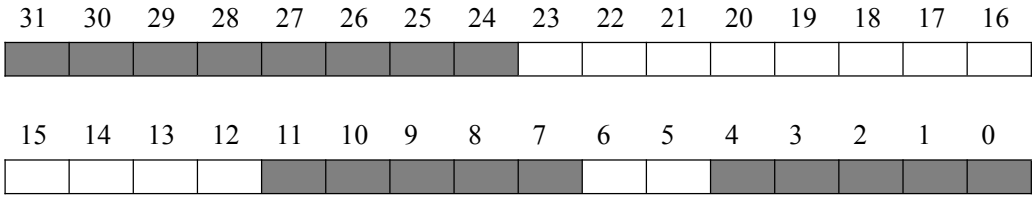
Bit	Type	Reset	Description
31:27	R/W	0	**Reserved**
26:18	R/W	83	RGB2YUV_coeff12 RGB到YUV转换时，B的系数。
17:9	R/W	57	RGB2YUV_coeff11 RGB到YUV转换时，G的系数。
8:0	R/W	2C	RGB2YUV_coeff10 RGB到YUV转换时，R的系数。

LCD_PARAM14

RGB to YCbCr coef2，计算 Cr 时 R G B 的系数。

Cr =（ R * RGB2YUV_coeff20 - G * RGB2YUV_coeff21 - B * RGB2YUV_coeff22）/256 + 128

Offset_Address: 0x0000_0038



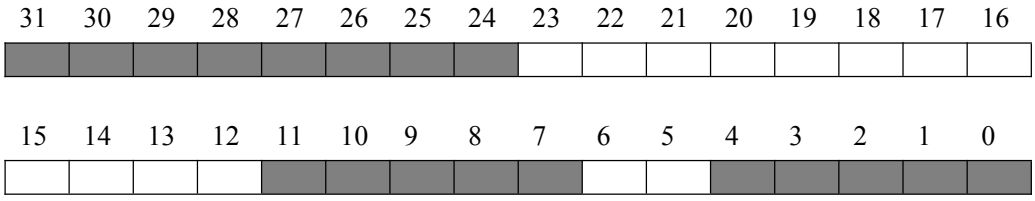
Bit	Type	Reset	Description
31:27	R/W	0	**Reserved**
26:18	R/W	15	RGB2YUV_coeff22 RGB到YUV转换时，B的系数。
17:9	R/W	6E	RGB2YUV_coeff21 RGB到YUV转换时，G的系数。
8:0	R/W	83	RGB2YUV_coeff20 RGB到YUV转换时，R的系数。

LCD_PARAM15

YCbCr to RGB coef0，计算 R 时 Y Cb Cr 的系数。

R =（ (Y –ysub_sel*16）* YUV2RGB_coeff00 +（Cr-128） * YUV2RGB_coeff02）/256

Offset_Address: 0x0000_003C



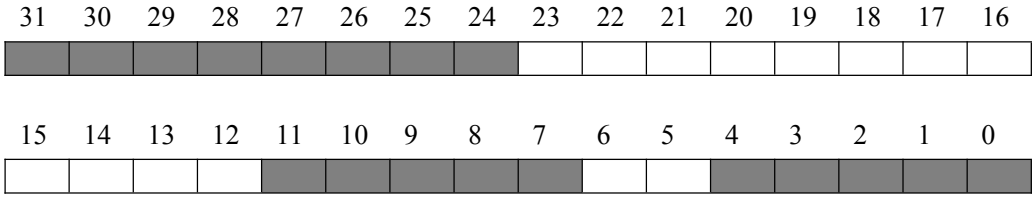
Bit	Type	Reset	Description
31:27	R/W	0	**Reserved**
26:18	R/W	15F	YUV2 RGB _coeff02 YUV到RGB转换时，Cr的系数。
17:9	R/W	0	Reserved
8:0	R/W	100	YUV2 RGB_coeff00 YUV到RGB转换时，Y的系数。

LCD_PARAM16

YCbCr to RGB coef1，计算 G 时 Y Cb Cr 的系数。

G =（(Y –ysub_sel*16）* YUV2RGB_coeff00 -（Cb-128）* YUV2RGB_coeff11 - （Cr-128） * YUV2RGB_coeff12）/256

Offset_Address: 0x0000_0040



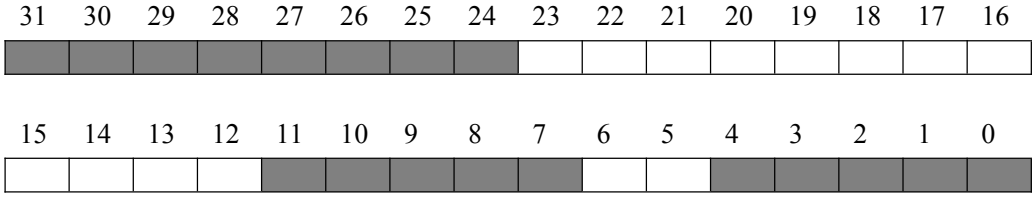
Bit	Type	Reset	Description
31:27	R/W	0	**Reserved**
26:18	R/W	B3	YUV2 RGB _coeff12 YUV到RGB转换时，Cr的系数。
17:9	R/W	56	YUV2 RGB_coeff11 YUV到RGB转换时，Cb的系数。
8:1	R/W	80	Reserved
0	R/W	0	Ysub_sel 计算R G B时Y上是否减16选择。 1：-16 0：-0

LCD_PARAM17

YCbCr to RGB coef2，计算 B 时 Y Cb Cr 的系数。

B =（(Y –ysub_sel*16）* YUV2RGB_coeff00 +（Cb-128）* YUV2RGB_coeff21）/256

Offset_Address: 0x0000_0044



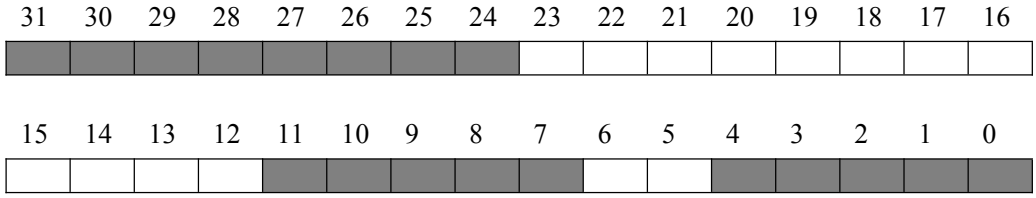
Bit	Type	Reset	Description
31:28	R/W	0	**Reserved**
27:19	R/W	0	Reserved
18:9	R/W	1BB	YUV2 RGB _coeff21

			YUV到RGB转换时，Cb的系数。
8:0	R/W	100	Reserved

LCD_PARAM18

ITU656时序生成寄存器

Offset_Address: 0x0000_0048

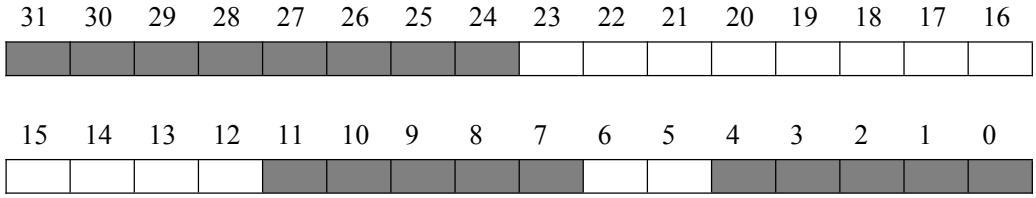


Bit	Type	Reset	Description
31:12	R/W	0	**Reserved**
11:0	R/W	0	Sav_pos ITU656有效数据起始位置

LCD_PARAM19

ITU656时序生成寄存器

Offset_Address: 0x0000_004C

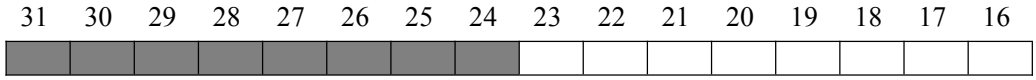


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	100	F1_vblk_end 第一场场消隐结束位置
11:0	R/W	0	F1_vblk_start 第一场场消隐开始位置

LCD_PARAM20

ITU656时序生成寄存器

Offset_Address: 0x0000_0050

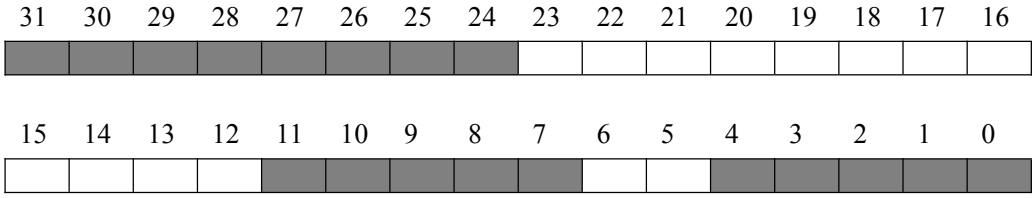


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	100	F2_vblk_end 第二场场消隐结束位置
11:0	R/W	0	F2_vblk_start 第二场场消隐开始位置

LCD_PARAM21

ITU656时序生成寄存器

Offset_Address: 0x0000_0054

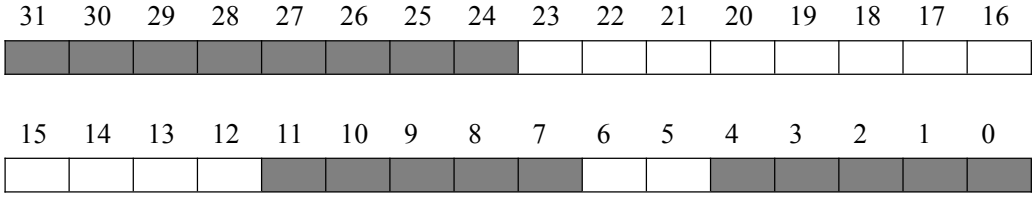


Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	100	F2_end 奇偶场标志（field=1）结束位置
11:0	R/W	0	F2_start 奇偶场标志（field=1）起始位置

LCD_PARAM22

LCD vsyn position intr parameter –lcd_vsyn_pos_num lcd_vsyn_pos_num_v1

Offset_Address: 0x0000_0058



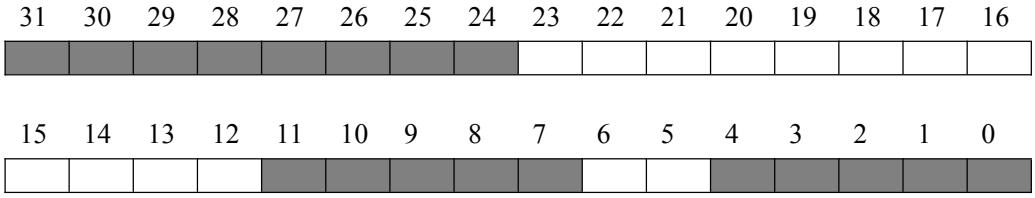
Bit	Type	Reset	Description
31:24	R/W	0	**Reserved**
23:12	R/W	0	lcd_vsyn_pos_num_v1 场方向中断位置寄存器，隔行输出时表示另一场的中断位

			置，逐行输出时需要与lcd_vsyn_pos_num配置成相同值。
11:0	R/W	0	lcd_vsyn_pos_num 场方向中断位置寄存器，在该位置出中断。

OSD_0_PARAM0

OSD 0 层参数

Offset_Address: 0x0000_005C

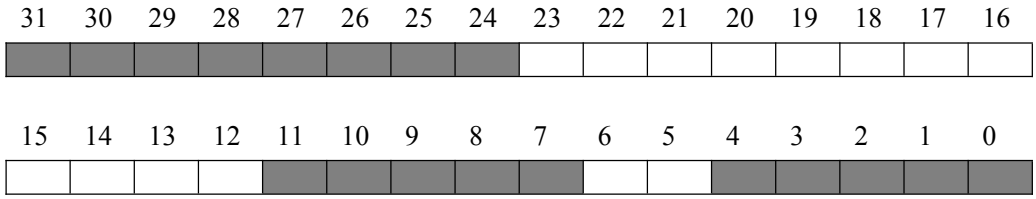


Bit	Type	Reset	Description
31	R/W	0	Osd_0_reverse 奇偶场取数翻转配置。当配置为隔行取数时，即lcd_interlace_flag=1时，该寄存器表示取数顺序。 0：第一场取1 3 5 … , 第二场取2 4 6… 1：第一场取2 4 6… , 第二场取1 3 5 …
30	R/W	0	Osd_0_vc1_range_map(Reserved) 对Y C数据进行乘积的使能 0：disable 1：enable
29:18	R/W	0	Osd_0_height 显示高度
17:6	R/W	0	Osd_0_width 显示宽度
5:3	R/W	0	Osd_0_format 数据源格式 0：YUV420 1：ARGB888 2：RGB565 3：RGB454
2	R/W	0	Reserved
1	R/W	0	Osd_0_enable 层使能，置1该层才能有数据显示。 1：enable 0：disable
0	R/W	0	Osd_0_global_coeff_enable 逐点blending的系数选择 0：ARGB88 8，数据源中的blending系数，可作逐点blending； 1：选择global coeff的blending系数。

OSD_0_PARAM1

OSD 0 层参数

Offset_Address: 0x0000_0060

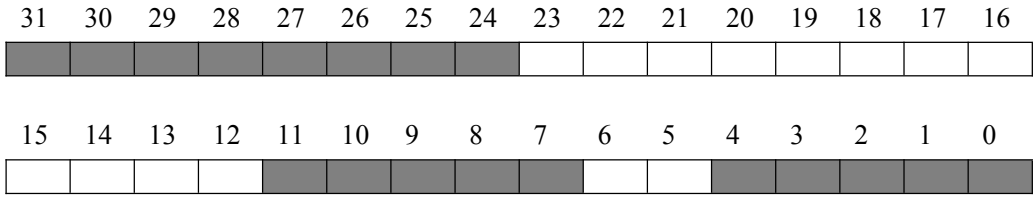


Bit	Type	Reset	Description
31	R/W	0	**Reserved**
30:24	R/W	0	Osd_0_global_coeff Blending系数
23:12	R/W	0	Osd_0_v_position 垂直方向显示位置设置
11:0	R/W	0	Osd_0_h_position 水平方向显示位置设置

OSD_0_PARAM2

OSD 0 层参数

Offset_Address: 0x0000_0064

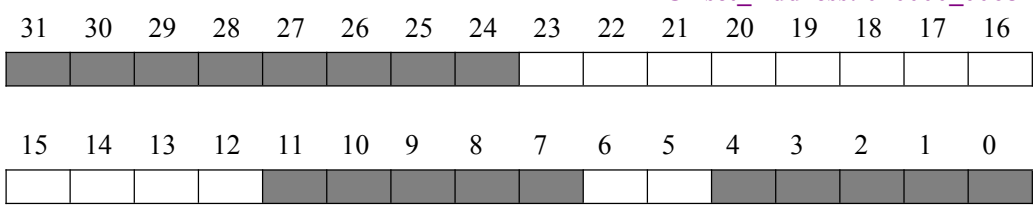


Bit	Type	Reset	Description
31:0	R/W	0	Osd_0_y_addr 读取数据起始位置，源为YUV420为Y地址，其它源为数据地址。

OSD_0_PARAM3

OSD 0 层参数

Offset_Address: 0x0000_0068



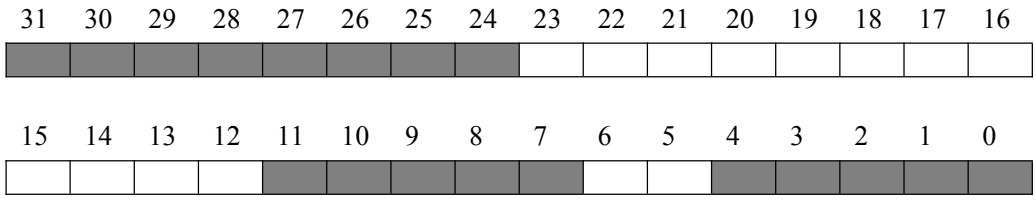
Bit	Type	Reset	Description

31:0	R/W	0	Osd_0_u_addr 读取数据起始位置，源为YUV420为U地址，其它源时无效。

OSD_0_PARAMCU

OSD 0 层参数

Offset_Address: 0x0000_006C

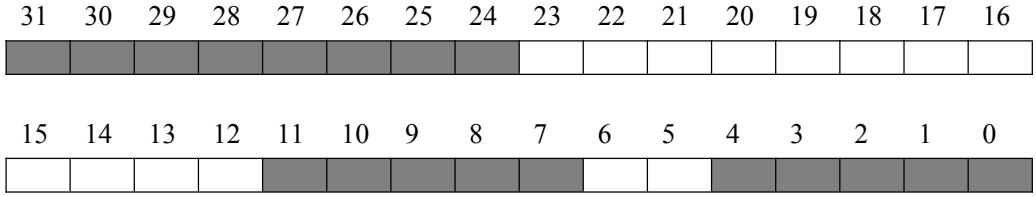


Bit	Type	Reset	Description
31:0	R/W	0	Osd_0_v_addr 读取数据起始位置，源为YUV420为V地址，其它源时无效。

OSD_0_PARAM5

OSD 0 层参数

Offset_Address: 0x0000_0070

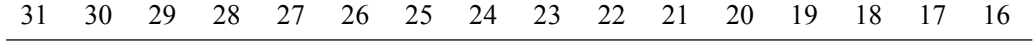


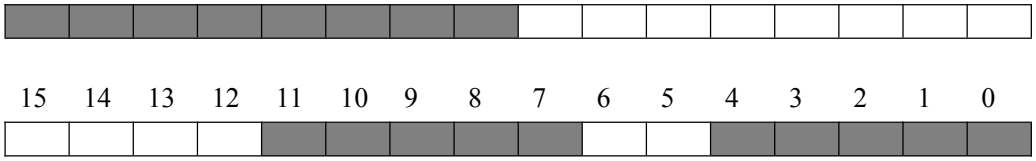
Bit	Type	Reset	Description
31	R/W	0	Reserved
30:24	R/W	0	Osd_0_mult_coef OSD 0层的数据乘积，可改变图象RGB大小。
23:12	R/W	0	Osd_0_h_offset 水平方向数据的偏移值，设置该寄存器可以改变显示的起始数据。
11:0	R/W	0	Osd_0_stride (行跳的间隔) 数据源图像的水平大小

OSD_1_PARAM0

OSD 1 层参数

Offset_Address: 0x0000_0074



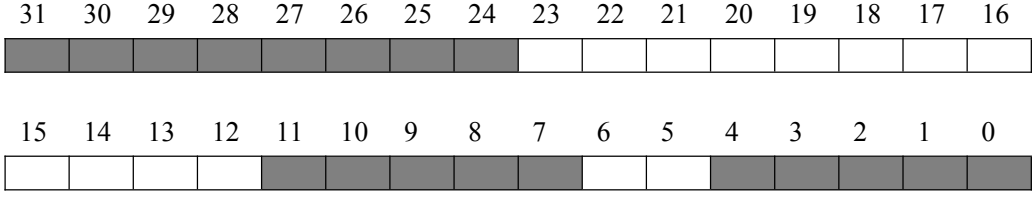


Bit	Type	Reset	Description
31	R/W	0	Osd_1_reverse 奇偶场取数翻转配置。当配置为隔行取数时，即lcd_interlace_flag=1时，该寄存器表示取数顺序。 0：第一场取1 3 5 … , 第二场取2 4 6… 1：第一场取2 4 6… , 第二场取1 3 5 …
30	R/W	0	Osd_1_vc1_range_map 对Y C数据进行乘积的使能 0：disable 1：enable
29:18	R/W	0	Osd_1_height 显示高度
17:6	R/W	0	Osd_1_width 显示宽度
5:3	R/W	0	Osd_1_format 数据源格式 0：YUV420 1：ARGB888 2：RGB565 3：RGB454
2	R/W	0	Reserved
1	R/W	0	Osd_1_enable 层使能，置1该层才能有数据显示。 1：enable 0：disable
0	R/W	0	Osd_1_global_coeff_enable 逐点blending的系数选择 0：ARGB888，数据源中的blending系数，可作逐点blending； 1：选择global coeff的blending系数。

OSD_1_PARAM1

OSD 1 层参数

Offset_Address: 0x0000_0078



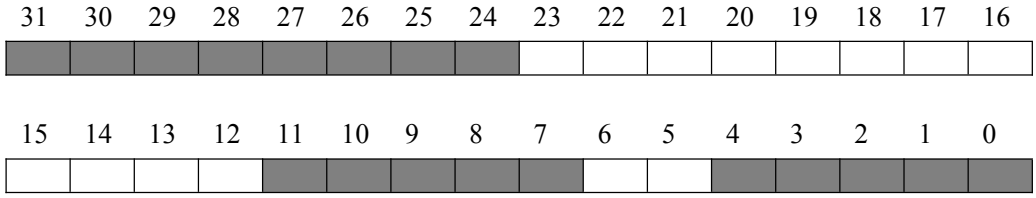
Bit	Type	Reset	Description

31	R/W	0	**Reserved**
30:24	R/W	0	Osd_1_global_coeff Blending系数
23:12	R/W	0	Osd_1_v_position 垂直方向显示位置设置
11:0	R/W	0	Osd_1_h_position 水平方向显示位置设置

OSD_1_PARAM2

OSD 1 层参数

Offset_Address: 0x0000_007C

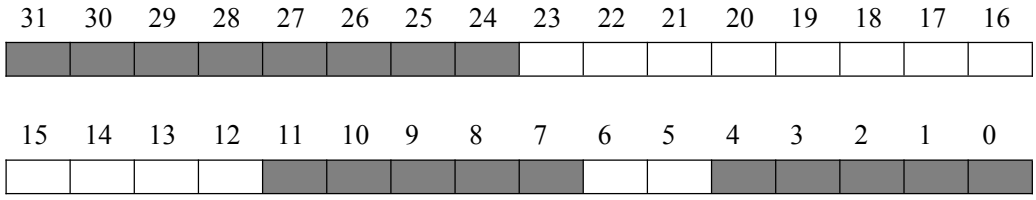


Bit	Type	Reset	Description
31:0	R/W	0	Osd_1_y_addr 读取数据起始位置，源为YUV420为Y地址，其它源为数据地址。

OSD_1_PARAM3

OSD 1 层参数

Offset_Address: 0x0000_0080

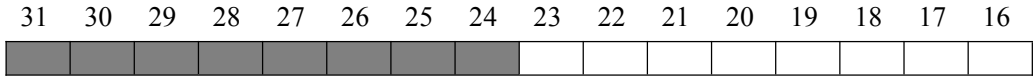


Bit	Type	Reset	Description
31:0	R/W	0	Osd_1_u_addr 读取数据起始位置，源为YUV420为U地址，其它源时无效。

OSD_1_PARAMCU

OSD 0 层参数

Offset_Address: 0x0000_0084

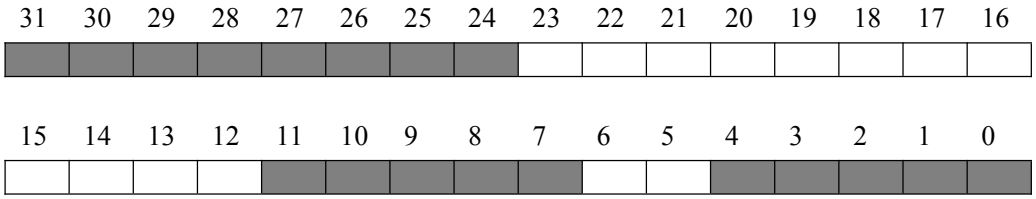


Bit	Type	Reset	Description
31:0	R/W	0	Osd_1_v_addr 读取数据起始位置，源为YUV420为V地址，其它源时无效。

OSD_1_PARAM5

OSD 1 层参数

Offset_Address: 0x0000_0088

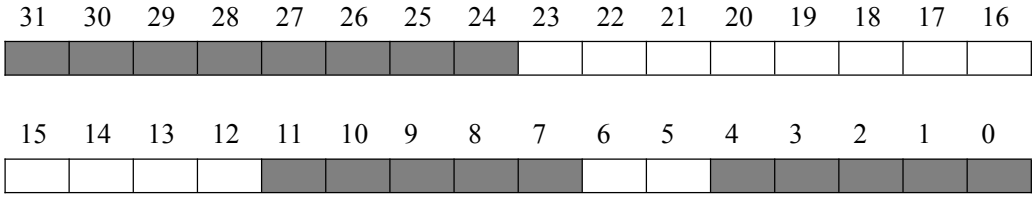


Bit	Type	Reset	Description
31	R/W	0	Reserved
30:24	R/W	0	Osd_1_mult_coef OSD 1层的数据乘积，可改变图象RGB大小。
23:12	R/W	0	Osd_1_h_offset 水平方向数据的便宜值，设置该寄存器可以改变显示的起始数据。
11:0	R/W	0	Osd_1_stride 数据源图像的水平大小

OSD_01_PARAM

OSD 0 1层参数

Offset_Address: 0x0000_008C



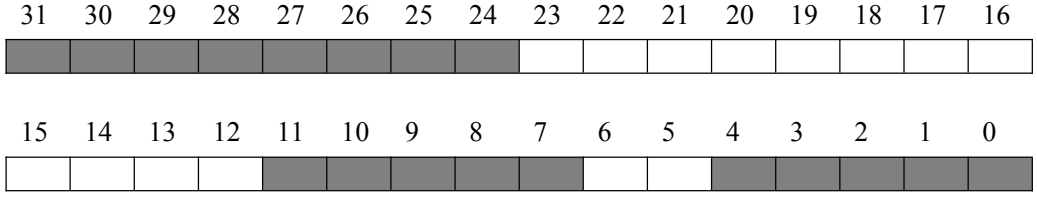
Bit	Type	Reset	Description
31:21	R/W	0	Reserved
20	R/W	0	OSD0 422 mode
19	R/W	0	OSD1 vflip_en
18	R/W	0	OSD1 hflip_en
17	R/W	0	OSD0 vflip_en

16	R/W	0	OSD0 hflip_en
15-14	R/W	0	Reserved
13	R/W	0	Osd_0_yuv420_v_first OSD0层源为y_uv 420模式有效，其他模式无效 0：VUVUVUVU，U存低位地址； 1：UVUVUVUV，U存高位地址。
12	R/W	0	Osd_0_yuv420_mode OSD0层源选择，当osd0_format=0时有效 0：源为y_u_v 420 模式，u v分块存储 1：源为y_uv420 模式，u v共块存储
11:8	R/W	0	
7:4	R/W	0	Osd_1_rgb_order OSD 1层RGB顺序的排列，在做blending之前 0：RGB 1：RBG 2：GRB 3：GBR 4：BRG 5：BGR 其它：RGB
3:0	R/W	0	Osd_0_rgb_order OSD 0层RGB顺序的排列，在做blending之前 0：RGB 1：RBG 2：GRB 3：GBR 4：BRG 5：BGR 其它：RGB

OSD_COEF_SYN

OSD 寄存器更新

Offset_Address: 0x0000_0090



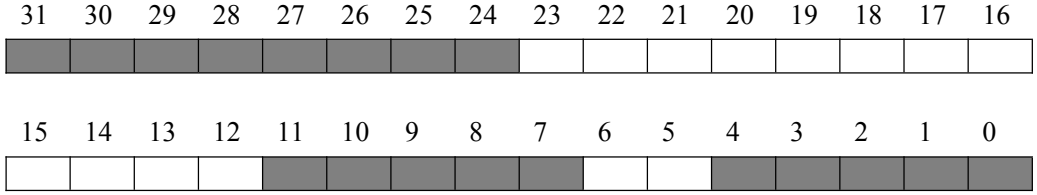
Bit	Type	Reset	Description
31-4	R	0	Reserved.
7-6	R		[3：0] Write value
3	R/W	0	global_ccm_coef_syn 对第 01 层 ccm 相关寄存器更新，只有对该位进行写 1 操作才能将 OSD 0 层相关寄存器更新。用以对所有寄存器进行同步。

			0：同步完成，已经同步且已经起作用。
2	R/W	0	Osd0_ccm_coef_syn 对第 0 层 osd ccm 相关寄存器更新，只有对该位进行写 1 操作才能将 OSD 0 层相关寄存器更新。用以对所有寄存器进行同步。 0：同步完成，已经同步且已经起作用。
1	R/W	0	Osd_1_coef_syn 对第 1 层 osd 相关寄存器更新，只有对该位进行写 1 操作才能将 OSD 1 层相关寄存器更新。用以对所有寄存器进行同步。相关寄存器有： OSD_1_PARAM0 OSD_1_PARAM1 OSD_1_PARAM2 OSD_1_PARAM3 OSD_1_PARAMCU OSD_1_PARAM5 OSD_012_PARAM[7：4] 读取该位寄存器表示： 是否配置的寄存器已经同步进硬件并开始作用。 1：同步进行中，已经同步但并没有起作用； 0：同步完成，已经同步且已经起作用。
0	R/W	0	Osd_0_coef_syn 对第 0 层 osd 相关寄存器更新，只有对该位进行写 1 操作才能将 OSD 0 层相关寄存器更新。用以对所有寄存器进行同步。相关寄存器有： OSD_0_PARAM0 OSD_0_PARAM1 OSD_0_PARAM2 OSD_0_PARAM3 OSD_0_PARAMCU OSD_0_PARAM5 OSD_012_PARAM[3：0] 读取该位寄存器表示： 是否配置的寄存器已经同步进硬件并开始作用。 1：同步进行中，已经同步但并没有起作用； 0：同步完成，已经同步且已经起作用。

OSD_1_PARAM8

OSD 1 层参数

Offset_Address: 0x0000_0094



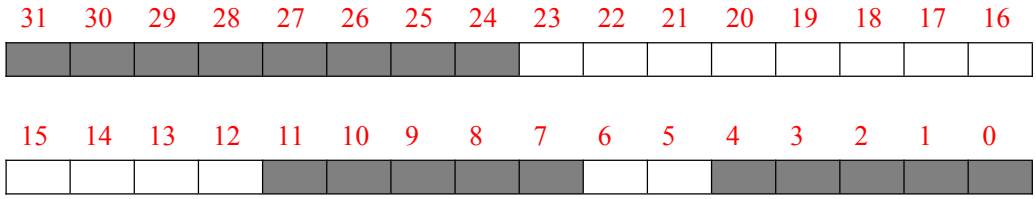
Bit	Type	Reset	Description
24	R/W	0	OSD1 color key en 0 disable 1 enable

23:0	R/W	0	OSD1_COLORKEY VALUE

OSD_1_PARAM9

OSD 1 层参数

Offset_Address: 0x0000_0098

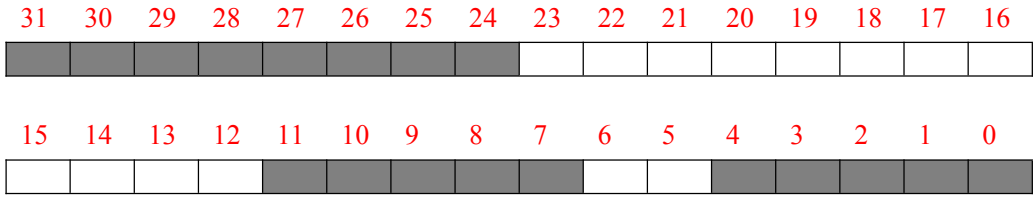


Bit	Type	Reset	Description
23:0	R/W	0	OSD1_COLORKY MASK_THLD

OSD_0_PARAM2_rd

OSD 0 层参数读取

Offset_Address: 0x0000_0280

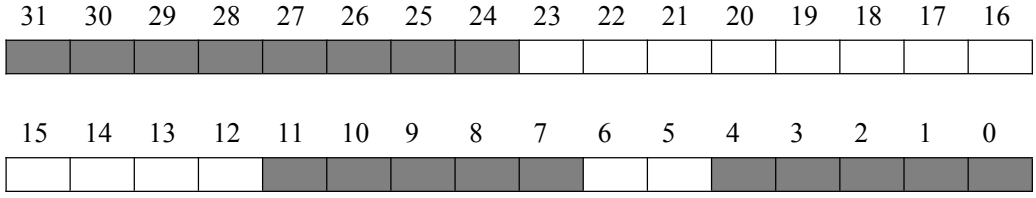


Bit	Type	Reset	Description
31:0	R/W	0	Osd_0_y_addr_rd 当前读取的Y地址。

OSD_0_PARAM3_rd

OSD 0 层参数读取

Offset_Address: 0x0000_0284

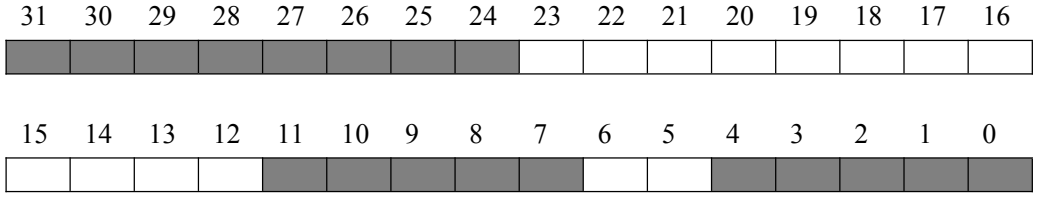


Bit	Type	Reset	Description
31:0	R/W	0	Osd_0_u_addr_rd 当前读取的U地址。

OSD_0_PARAMCU_rd

OSD 0 层参数读取

Offset_Address: 0x0000_0288

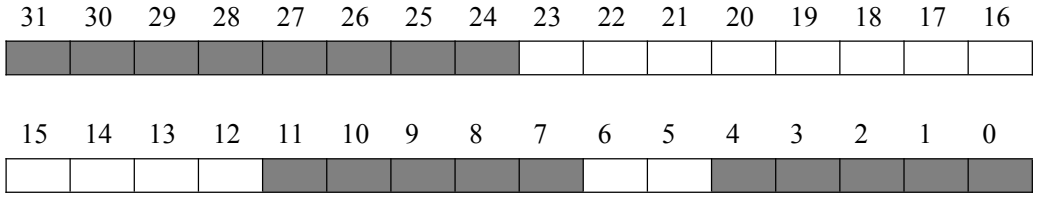


Bit	Type	Reset	Description
31:0	R/W	0	Osd_0_v_addr_rd 当前读取的V地址。

OSD_1_PARAM2_rd

OSD 1 层参数读取

Offset_Address: 0x0000_00B028c

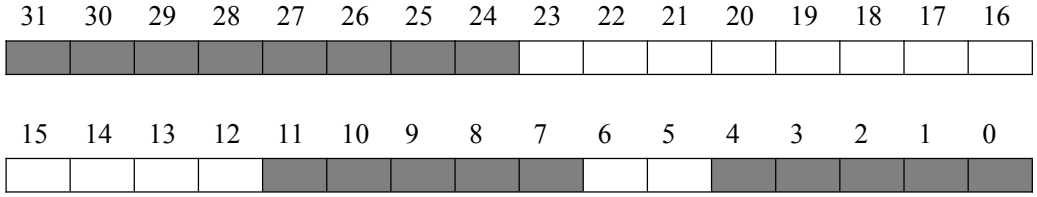


Bit	Type	Reset	Description
31:0	R/W	0	Osd_1_y_addr_rd 当前读取的Y地址。

OSD_1_PARAM3_rd

OSD 1 层参数读取

Offset_Address: 0x0000_0290

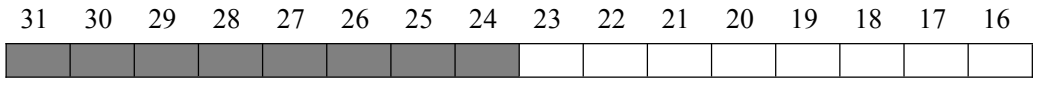


Bit	Type	Reset	Description
31:0	R/W	0	Osd_1_u_addr_rd 当前读取的U地址。

OSD_1_PARAMCU_rd

OSD 1 层参数读取

Offset_Address: 0x0000_0294

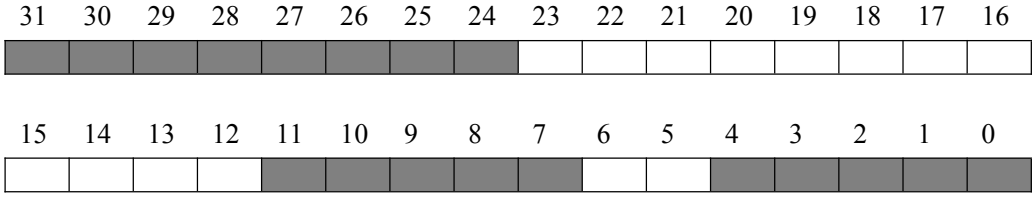


Bit	Type	Reset	Description
31:0	R/W	0	Osd_1_v_addr_rd 当前读取的V地址。

Dithering cfg 0

Dithering 0 配置参数

Offset_Address: 0x0000_00bc

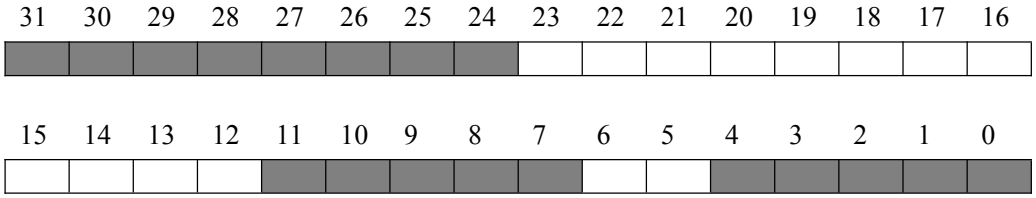


Bit	Type	Reset	Description
15:12	R/W	0	Dithering b output depth
11:8	R/W	0	Dithering g output depth
7:4	R/W	0	Dithering r output depth
3	R/W	0	Dithering testen
2	R/W	0	Dithering prefetch
1	R/W	0	Dithering softreset 0 reset 1 normal
0	R/W	0	Dithering enable 1 enable 0 disable

Dithering cfg 1

Dithering 1 配置参数

Offset_Address: 0x0000_00c0

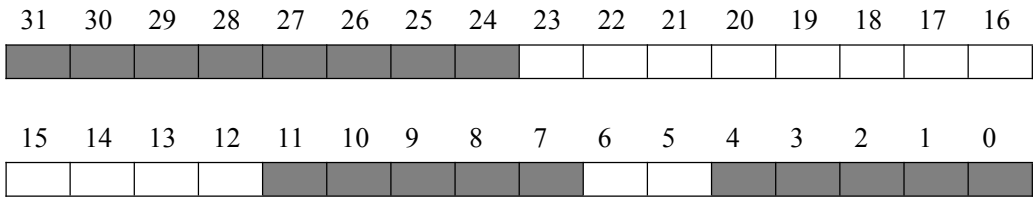


Bit	Type	Reset	Description
28:16	R/W	0	Dithering active vsize
12:0	R/W	0	Dithering active hsize

Dithering cfg 2

Dithering 2 配置参数

Offset_Address: 0x0000_00c4

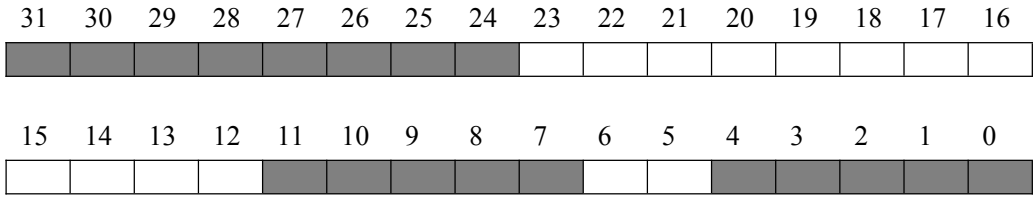


Bit	Type	Reset	Description
23:16	R/W	0	B test data
15:8	R/W	0	G test data
7:0	R/W	0	R test data

LCD_INTR_CLR

LCD 清中断寄存器

Offset_Address: 0x0000_00C8

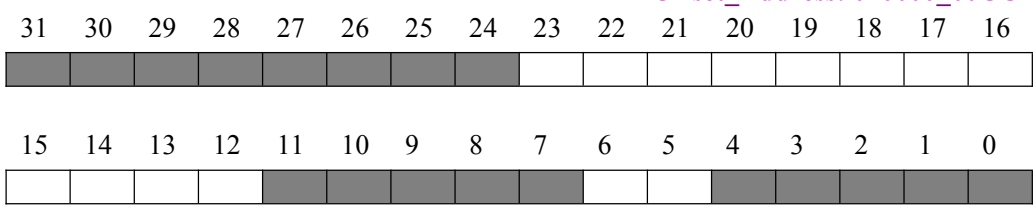


Bit	Type	Reset	Description
31:3	R/W	0	Reserved
2	R/W	0	Lcd_vsyn_pos_intr_clr LCD场方向任意位置中断清零位
1	R/W	0	V_syn_intr_clr LCD场同步中断清零位
0	R/W	0	Lcd_done_intr_clr LCD场结束中断清零位

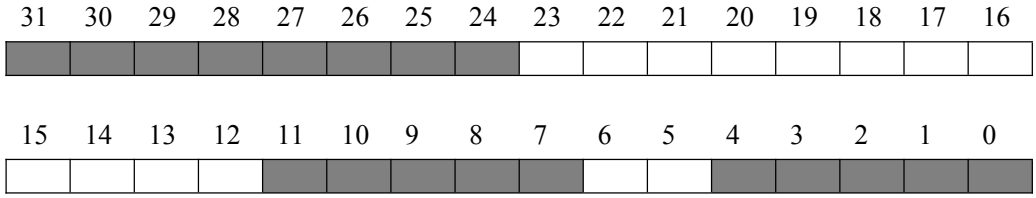
LCD_STATUS_REG

LCD 状态寄存器，只读寄存器

Offset_Address: 0x0000_00CC



Bit	Type	Reset	Description
31:7	R/W	0	Reserved
6	R	0	Dma_req_2 OSD2 层dma 请求状态位
5	R	0	Dma_req_1 OSD1 层dma 请求状态位
4	R	0	Dma_req_0 OSD0 层dma 请求状态位
3	R	0	Lcd_intr LCD中断状态，包括lcd_done_intr , v_syn_intr
2	R	0	Lcd_vsyn_pos_intr LCD场方向任意位置中断状态为
1	R	0	V_syn_intr LCD场同步中断状态位
0	R	0	Lcd_done_intr LCD场结束中断状态位

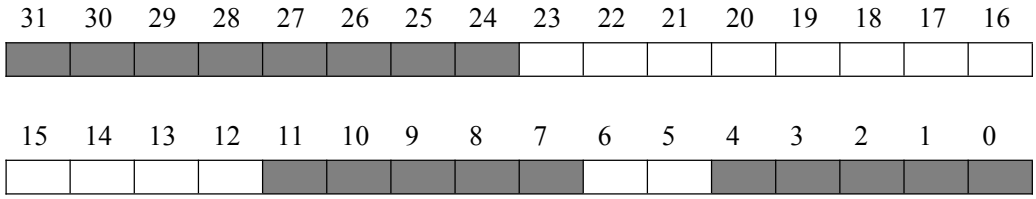


Bit	Type	Reset	Description
19:0	R/W	0	Adr clr data

R_BIT_ORDER

输出 R 的 bit 位进行灵活对调

Offset_Address: 0x0000_0130



Bit	Type	Reset	Description
31:30	R/W	0	Reserved
29:28	R/W	0	Clac_order_sel 0: clac, DE信号 1：cllp，行同步 2：clfp，场同步 其它：clac，DE信号
27:26	R/W	0	Clfp_order_sel 0: clfp, 场同步 1：cllp，行同步

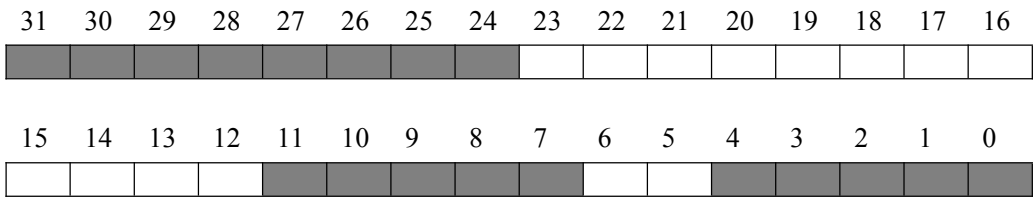
			2：clac，DE信号 其它：clfp，场同步
25:24	R/W	0	Cllp_order_sel 0: cllp, 行同步 1：clfp，场同步 2：clac，DE信号 其它：cllp，行同步
23:21	R/W	0	R7_bit_order 选择R7输出的信号 0：R0 1：R1 2: R2 3: R3 4: R4 5: R5 6: R6 7: R7
20:18	R/W	0	R6_bit_order 选择R6输出的信号 0：R0 1：R1 2: R2 3: R3 4: R4 5: R5 6: R6 7: R7
17:15	R/W	0	R5_bit_order 选择R5输出的信号 0：R0 1：R1 2: R2 3: R3 4: R4 5: R5 6: R6 7: R7
14:12	R/W	0	R4_bit_order 选择R4输出的信号 0：R0 1：R1 2: R2 3: R3 4: R4 5: R5 6: R6 7: R7
11:9	R/W	0	R3_bit_order 选择R3输出的信号 0：R0 1：R1 2: R2 3: R3

			4: R4 5: R5 6: R6 7: R7
8:6	R/W	0	R2_bit_order 选择R2输出的信号 0：R0 1：R1 2: R2 3: R3 4: R4 5: R5 6: R6 7: R7
5:3	R/W	0	R1_bit_order 选择R1输出的信号 0：R0 1：R1 2: R2 3: R3 4: R4 5: R5 6: R6 7: R7
2:0	R/W	0	R0_bit_order 选择R0输出的信号 0：R0 1：R1 2: R2 3: R3 4: R4 5: R5 6: R6 7: R7

G_BIT_ORDER

输出 G 的 bit 位进行灵活对调

Offset_Address: 0x0000_0134



Bit	Type	Reset	Description
31:24	R/W	0	Reserved
23:21	R/W	0	G7_bit_order 选择G7输出的信号 0：G0

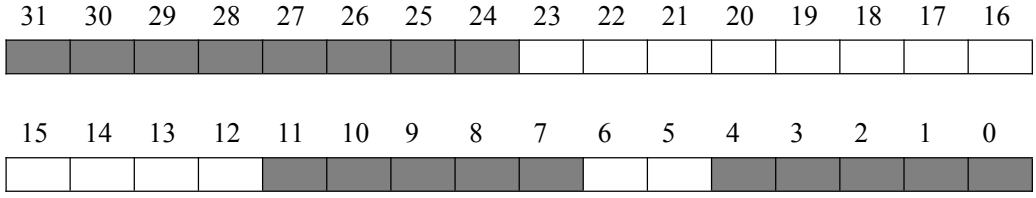
			1：G1 2: G2 3: G3 4: G4 5: G5 6: G6 7: G7
20:18	R/W	0	G6_bit_order 选择G6输出的信号 0：G0 1：G1 2: G2 3: G3 4: G4 5: G5 6: G6 7: G7
17:15	R/W	0	G5_bit_order 选择G5输出的信号 0：G0 1：G1 2: G2 3: G3 4: G4 5: G5 6: G6 7: G7
14:12	R/W	0	G4_bit_order 选择G4输出的信号 0：G0 1：G1 2: G2 3: G3 4: G4 5: G5 6: G6 7: G7
11:9	R/W	0	G3_bit_order 选择G3输出的信号 0：G0 1：G1 2: G2 3: G3 4: G4 5: G5 6: G6 7: G7
8:6	R/W	0	G2_bit_order 选择G2输出的信号 0：G0 1：G1 2: G2 3: G3 4: G4

			5: G5 6: G6 7: G7
5:3	R/W	0	G1_bit_order 选择G1输出的信号 0：G0 1：G1 2: G2 3: G3 4: G4 5: G5 6: G6 7: G7
2:0	R/W	0	G0_bit_order 选择G0输出的信号 0：G0 1：G1 2: G2 3: G3 4: G4 5: G5 6: G6 7: G7

B_BIT_ORDER

输出 B 的 bit 位进行灵活对调

Offset_Address: 0x0000_0138



Bit	Type	Reset	Description
31:24	R/W	0	Reserved
23:21	R/W	0	B7_bit_order 选择B7输出的信号 0：B0 1：B1 2: B2 3: B3 4: B4 5: B5 6: B6 7: B7
20:18	R/W	0	B6_bit_order 选择B6输出的信号 0：B0

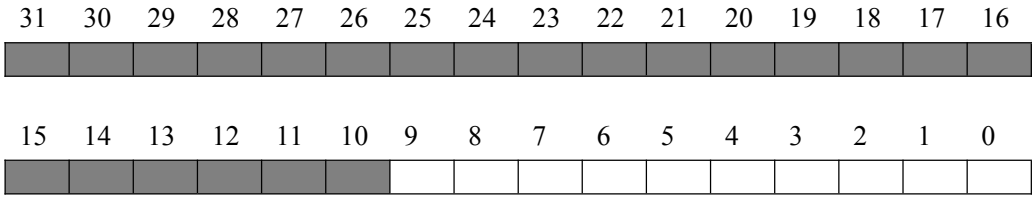
			1：B1 2: B2 3: B3 4: B4 5: B5 6: B6 7: B7
17:15	R/W	0	B5_bit_order 选择B5输出的信号 0：B0 1：B1 2: B2 3: B3 4: B4 5: B5 6: B6 7: B7
14:12	R/W	0	B4_bit_order 选择B4输出的信号 0：B0 1：B1 2: B2 3: B3 4: B4 5: B5 6: B6 7: B7
11:9	R/W	0	B3_bit_order 选择B3输出的信号 0：B0 1：B1 2: B2 3: B3 4: B4 5: B5 6: B6 7: B7
8:6	R/W	0	B2_bit_order 选择B2输出的信号 0：B0 1：B1 2: B2 3: B3 4: B4 5: B5 6: B6 7: B7
5:3	R/W	0	B1_bit_order 选择B1输出的信号 0：B0 1：B1 2: B2 3: B3 4: B4

			5: B5 6: B6 7: B7
2:0	R/W	0	B0_bit_order 选择B0输出的信号 0：B0 1：B1 2: B2 3: B3 4: B4 5: B5 6: B6 7: B7

AMMA_REG

GAMMA_REG_0

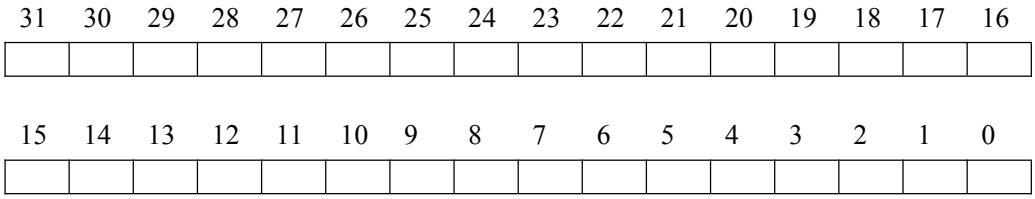
Offset_Address: 0x0000_013C



Bit	Type	Reset	Description
0	R/W	0	**CUSTOM_EN** Custom himself set gamma curve enable, active high
1	R/W	0	**GAMMA_EN** Gamma enable,active high
31-2	R	0	**Not used**

GAMMA_REG_1

Offset_Address: 0x0000_0140



Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val01** custom set gamma r curve coefficient 1
15-8	R/W	0	**custom_r_val02** custom set gamma r curve coefficient 2
23-16	R/W	0	**custom_r_val03** custom set gamma r curve coefficient 3
31-24	R/W	0	**custom_r_val04**

			custom set gamma r curve coefficient 4

GAMMA_REG_2

Offset_Address: 0x0000_0144

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val05** custom set gamma r curve coefficient 5
15-8	R/W	0	**custom_r_val06** custom set gamma r curve coefficient 6
23-16	R/W	0	**custom_r_val07** custom set gamma r curve coefficient 7
31-24	R/W	0	**custom_r_val08** custom set gamma r curve coefficient 8

GAMMA_REG_3

Offset_Address: 0x0000_0148

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val09** custom set gamma r curve coefficient 09
15-8	R/W	0	**custom_r_val10** custom set gamma r curve coefficient 10
23-16	R/W	0	**custom_r_val11** custom set gamma r curve coefficient 11
31-24	R/W	0	**custom_r_val12** custom set gamma r curve coefficient 12

GAMMA_REG_4

Offset_Address: 0x0000_014C

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val13** custom set gamma r curve coefficient 13
15-8	R/W	0	**custom_r_val14** custom set gamma r curve coefficient 14
23-16	R/W	0	**custom_r_val15** custom set gamma r curve coefficient 15
31-24	R/W	0	**custom_r_val16** custom set gamma r curve coefficient 16

GAMMA_REG_5

Offset_Address: 0x0000_0150

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val17** custom set gamma r curve coefficient 17
15-8	R/W	0	**custom_r_val18** custom set gamma r curve coefficient 18
23-16	R/W	0	**custom_r_val19** custom set gamma r curve coefficient 19
31-24	R/W	0	**custom_r_val20** custom set gamma r curve coefficient 20

GAMMA_REG_6

Offset_Address: 0x0000_0154

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val21** custom set gamma r curve coefficient 21
15-8	R/W	0	**custom_r_val22** custom set gamma r curve coefficient 22
23-16	R/W	0	**custom_r_val23** custom set gamma r curve coefficient 23
31-24	R/W	0	**custom_r_val24** custom set gamma r curve coefficient 24

GAMMA_REG_7

Offset_Address: 0x0000_0158

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val25** custom set gamma r curve coefficient 25
15-8	R/W	0	**custom_r_val26** custom set gamma r curve coefficient 26
23-16	R/W	0	**custom_r_val27** custom set gamma r curve coefficient 27
31-24	R/W	0	**custom_r_val28** custom set gamma r curve coefficient 28

GAMMA_REG_8

Offset_Address: 0x0000_015C

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val29** custom set gamma r curve coefficient 29
15-8	R/W	0	**custom_r_val30** custom set gamma r curve coefficient 30
23-16	R/W	0	**custom_r_val31** custom set gamma r curve coefficient 31
31-24	R/W	0	**custom_r_val32** custom set gamma r curve coefficient 32

GAMMA_REG_9

Offset_Address: 0x0000_0160

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val33** custom set gamma r curve coefficient 33
15-8	R/W	0	**custom_r_val34**

			custom set gamma r curve coefficient 34
23-16	R/W	0	**custom_r_val35** custom set gamma r curve coefficient 35
31-24	R/W	0	**custom_r_val36** custom set gamma r curve coefficient 36

GAMMA_REG_10

Offset_Address: 0x0000_0164

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val37** custom set gamma r curve coefficient 37
15-8	R/W	0	**custom_r_val38** custom set gamma r curve coefficient 38
23-16	R/W	0	**custom_r_val39** custom set gamma r curve coefficient 39
31-24	R/W	0	**custom_r_val40** custom set gamma r curve coefficient 40

GAMMA_REG_11

Offset_Address: 0x0000_0168

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val41** custom set gamma r curve coefficient 41
15-8	R/W	0	**custom_r_val42** custom set gamma r curve coefficient 42
23-16	R/W	0	**custom_r_val43** custom set gamma r curve coefficient 43
31-24	R/W	0	**custom_r_val44** custom set gamma r curve coefficient 44

GAMMA_REG_12

Offset_Address: 0x0000_016C

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val45** custom set gamma r curve coefficient 45
15-8	R/W	0	**custom_r_val46** custom set gamma r curve coefficient 46
23-16	R/W	0	**custom_r_val47** custom set gamma r curve coefficient 47
31-24	R/W	0	**custom_r_val48** custom set gamma r curve coefficient 48

GAMMA_REG_13

Offset_Address: 0x0000_0170

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val49** custom set gamma r curve coefficient 49
15-8	R/W	0	**custom_r_val50** custom set gamma r curve coefficient 50
23-16	R/W	0	**custom_r_val51** custom set gamma r curve coefficient 51
31-24	R/W	0	**custom_r_val52** custom set gamma r curve coefficient 52

GAMMA_REG_14

Offset_Address: 0x0000_0174

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val53** custom set gamma r curve coefficient 53
15-8	R/W	0	**custom_r_val54** custom set gamma r curve coefficient 54
23-16	R/W	0	**custom_r_val55** custom set gamma r curve coefficient 55

31-24	R/W	0	**custom_r_val56** custom set gamma r curve coefficient 56

GAMMA_REG_15

Offset_Address: 0x0000_0178

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val57** custom set gamma r curve coefficient 57
15-8	R/W	0	**custom_r_val58** custom set gamma r curve coefficient 58
23-16	R/W	0	**custom_r_val59** custom set gamma r curve coefficient 59
31-24	R/W	0	**custom_r_val60** custom set gamma r curve coefficient 60

GAMMA_REG_16

Offset_Address: 0x0000_017C

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_r_val61** custom set gamma r curve coefficient 61
15-8	R/W	0	**custom_r_val62** custom set gamma r curve coefficient 62
23-16	R/W	0	**custom_r_val63** custom set gamma r curve coefficient 63
31-24	R/W	0	**custom_r_val64** custom set gamma r curve coefficient 64

GAMMA_REG_17

Offset_Address: 0x0000_0180

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val01** custom set gamma g curve coefficient 1
15-8	R/W	0	**custom_g_val02** custom set gamma g curve coefficient 2
23-16	R/W	0	**custom_g_val03** custom set gamma g curve coefficient 3
31-24	R/W	0	**custom_g_val04** custom set gamma g curve coefficient 4

GAMMA_REG_18

Offset_Address: 0x0000_0184

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val05** custom set gamma g curve coefficient 5
15-8	R/W	0	**custom_g_val06** custom set gamma g curve coefficient 6
23-16	R/W	0	**custom_g_val07** custom set gamma g curve coefficient 7
31-24	R/W	0	**custom_g_val08** custom set gamma g curve coefficient 8

GAMMA_REG_19

Offset_Address: 0x0000_0188

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val09** custom set gamma g curve coefficient 9
15-8	R/W	0	**custom_g_val10** custom set gamma g curve coefficient 10
23-16	R/W	0	**custom_g_val11** custom set gamma g curve coefficient 11
31-24	R/W	0	**custom_g_val12** custom set gamma g curve coefficient 12

GAMMA_REG_20

Offset_Address: 0x0000_018C

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val13** custom set gamma g curve coefficient 13
15-8	R/W	0	**custom_g_val14** custom set gamma g curve coefficient 14
23-16	R/W	0	**custom_g_val15** custom set gamma g curve coefficient 15
31-24	R/W	0	**custom_g_val16** custom set gamma g curve coefficient 16

GAMMA_REG_21

Offset_Address: 0x0000_0190

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val17** custom set gamma g curve coefficient 17
15-8	R/W	0	**custom_g_val18** custom set gamma g curve coefficient 18
23-16	R/W	0	**custom_g_val19** custom set gamma g curve coefficient 19
31-24	R/W	0	**custom_g_val20** custom set gamma g curve coefficient 20

GAMMA_REG_22

Offset_Address: 0x0000_0194

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

Bit	Type	Reset	Description

7-0	R/W	0	**custom_g_val21** custom set gamma g curve coefficient 21
15-8	R/W	0	**custom_g_val22** custom set gamma g curve coefficient 22
23-16	R/W	0	**custom_g_val23** custom set gamma g curve coefficient 23
31-24	R/W	0	**custom_g_val24** custom set gamma g curve coefficient 24

GAMMA_REG_23

Offset_Address: 0x0000_0198

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val25** custom set gamma g curve coefficient 25
15-8	R/W	0	**custom_g_val26** custom set gamma g curve coefficient 26
23-16	R/W	0	**custom_g_val27** custom set gamma g curve coefficient 27
31-24	R/W	0	**custom_g_val28** custom set gamma g curve coefficient 28

GAMMA_REG_24

Offset_Address: 0x0000_019C

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val29** custom set gamma g curve coefficient 29
15-8	R/W	0	**custom_g_val30** custom set gamma g curve coefficient 30
23-16	R/W	0	**custom_g_val31** custom set gamma g curve coefficient 31
31-24	R/W	0	**custom_g_val32** custom set gamma g curve coefficient 32

GAMMA_REG_25

Offset_Address: 0x0000_01A0

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val33** custom set gamma g curve coefficient 33
15-8	R/W	0	**custom_g_val34** custom set gamma g curve coefficient 34
23-16	R/W	0	**custom_g_val35** custom set gamma g curve coefficient 35
31-24	R/W	0	**custom_g_val36** custom set gamma g curve coefficient 36

GAMMA_REG_26

Offset_Address: 0x0000_01A4

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val37** custom set gamma g curve coefficient 37
15-8	R/W	0	**custom_g_val38** custom set gamma g curve coefficient 38
23-16	R/W	0	**custom_g_val39** custom set gamma g curve coefficient 39
31-24	R/W	0	**custom_g_val40** custom set gamma g curve coefficient 40

GAMMA_REG_27

Offset_Address: 0x0000_01A8

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val41** custom set gamma g curve coefficient 41

15-8	R/W	0	**custom_g_val42** custom set gamma g curve coefficient 42
23-16	R/W	0	**custom_g_val43** custom set gamma g curve coefficient 43
31-24	R/W	0	**custom_g_val44** custom set gamma g curve coefficient 44

GAMMA_REG_28

Offset_Address: 0x0000_01AC

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val45** custom set gamma g curve coefficient 45
15-8	R/W	0	**custom_g_val46** custom set gamma g curve coefficient 46
23-16	R/W	0	**custom_g_val47** custom set gamma g curve coefficient 47
31-24	R/W	0	**custom_g_val48** custom set gamma g curve coefficient 48

GAMMA_REG_29

Offset_Address: 0x0000_01B0

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val49** custom set gamma g curve coefficient 49
15-8	R/W	0	**custom_g_val50** custom set gamma g curve coefficient 50
23-16	R/W	0	**custom_g_val51** custom set gamma g curve coefficient 51
31-24	R/W	0	**custom_g_val52** custom set gamma g curve coefficient 52

GAMMA_REG_30

Offset_Address: 0x0000_01B4

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val53** custom set gamma g curve coefficient 53
15-8	R/W	0	**custom_g_val54** custom set gamma g curve coefficient 54
23-16	R/W	0	**custom_g_val55** custom set gamma g curve coefficient 55
31-24	R/W	0	**custom_g_val56** custom set gamma g curve coefficient 56

GAMMA_REG_31

Offset_Address: 0x0000_01B8

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val57** custom set gamma g curve coefficient 57
15-8	R/W	0	**custom_g_val58** custom set gamma g curve coefficient 58
23-16	R/W	0	**custom_g_val59** custom set gamma g curve coefficient 59
31-24	R/W	0	**custom_g_val60** custom set gamma g curve coefficient 60

GAMMA_REG_32

Offset_Address: 0x0000_01BC

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_g_val61** custom set gamma g curve coefficient 61
15-8	R/W	0	**custom_g_val62** custom set gamma g curve coefficient 62
23-16	R/W	0	**custom_g_val63**

			custom set gamma g curve coefficient 63
31-24	R/W	0	**custom_g_val64** custom set gamma g curve coefficient 64

GAMMA_REG_33

Offset_Address: 0x0000_01C0

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val01** custom set gamma b curve coefficient 1
15-8	R/W	0	**custom_b_val02** custom set gamma b curve coefficient 2
23-16	R/W	0	**custom_b_val03** custom set gamma b curve coefficient 3
31-24	R/W	0	**custom_b_val04** custom set gamma b curve coefficient 4

GAMMA_REG_34

Offset_Address: 0x0000_01C4

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val05** custom set gamma b curve coefficient 5
15-8	R/W	0	**custom_b_val06** custom set gamma b curve coefficient 6
23-16	R/W	0	**custom_b_val07** custom set gamma b curve coefficient 7
31-24	R/W	0	**custom_b_val08** custom set gamma b curve coefficient 8

GAMMA_REG_35

Offset_Address: 0x0000_01C8

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val09** custom set gamma b curve coefficient 9
15-8	R/W	0	**custom_b_val10** custom set gamma b curve coefficient 10
23-16	R/W	0	**custom_b_val11** custom set gamma b curve coefficient 11
31-24	R/W	0	**custom_b_val12** custom set gamma b curve coefficient 12

GAMMA_REG_36

Offset_Address: 0x0000_01CC

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val13** custom set gamma b curve coefficient 13
15-8	R/W	0	**custom_b_val14** custom set gamma b curve coefficient 14
23-16	R/W	0	**custom_b_val15** custom set gamma b curve coefficient 15
31-24	R/W	0	**custom_b_val16** custom set gamma b curve coefficient 16

GAMMA_REG_37

Offset_Address: 0x0000_01D0

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val17** custom set gamma b curve coefficient 17
15-8	R/W	0	**custom_b_val18** custom set gamma b curve coefficient 18
23-16	R/W	0	**custom_b_val19** custom set gamma b curve coefficient 19

31-24	R/W	0	**custom_b_val20** custom set gamma b curve coefficient 20

GAMMA_REG_38

Offset_Address: 0x0000_01D4

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val21** custom set gamma b curve coefficient 21
15-8	R/W	0	**custom_b_val22** custom set gamma b curve coefficient 22
23-16	R/W	0	**custom_b_val23** custom set gamma b curve coefficient 23
31-24	R/W	0	**custom_b_val24** custom set gamma b curve coefficient 24

GAMMA_REG_39

Offset_Address: 0x0000_01D8

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val25** custom set gamma b curve coefficient 25
15-8	R/W	0	**custom_b_val26** custom set gamma b curve coefficient 26
23-16	R/W	0	**custom_b_val27** custom set gamma b curve coefficient 27
31-24	R/W	0	**custom_b_val28** custom set gamma b curve coefficient 28

GAMMA_REG_40

Offset_Address: 0x0000_01DC

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val29** custom set gamma b curve coefficient 29
15-8	R/W	0	**custom_b_val30** custom set gamma b curve coefficient 30
23-16	R/W	0	**custom_b_val31** custom set gamma b curve coefficient 31
31-24	R/W	0	**custom_b_val32** custom set gamma b curve coefficient 32

GAMMA_REG_41

Offset_Address: 0x0000_01E0

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val33** custom set gamma b curve coefficient 33
15-8	R/W	0	**custom_b_val34** custom set gamma b curve coefficient 34
23-16	R/W	0	**custom_b_val35** custom set gamma b curve coefficient 35
31-24	R/W	0	**custom_b_val36** custom set gamma b curve coefficient 36

GAMMA_REG_42

Offset_Address: 0x0000_01E4

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val37** custom set gamma b curve coefficient 37
15-8	R/W	0	**custom_b_val38** custom set gamma b curve coefficient 38
23-16	R/W	0	**custom_b_val39** custom set gamma b curve coefficient 39
31-24	R/W	0	**custom_b_val40** custom set gamma b curve coefficient 40

GAMMA_REG_43

Offset_Address: 0x0000_01E8

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val41** custom set gamma b curve coefficient 41
15-8	R/W	0	**custom_b_val42** custom set gamma b curve coefficient 42
23-16	R/W	0	**custom_b_val43** custom set gamma b curve coefficient 43
31-24	R/W	0	**custom_b_val44** custom set gamma b curve coefficient 44

GAMMA_REG_44

Offset_Address: 0x0000_01EC

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val45** custom set gamma b curve coefficient 45
15-8	R/W	0	**custom_b_val46** custom set gamma b curve coefficient 46
23-16	R/W	0	**custom_b_val47** custom set gamma b curve coefficient 47
31-24	R/W	0	**custom_b_val48** custom set gamma b curve coefficient 48

GAMMA_REG_45

Offset_Address: 0x0000_01F0

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val49** custom set gamma b curve coefficient 49
15-8	R/W	0	**custom_b_val50** custom set gamma b curve coefficient 50
23-16	R/W	0	**custom_b_val51** custom set gamma b curve coefficient 51
31-24	R/W	0	**custom_b_val52** custom set gamma b curve coefficient 52

GAMMA_REG_46

Offset_Address: 0x0000_01F4

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

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val53** custom set gamma b curve coefficient 53
15-8	R/W	0	**custom_b_val54** custom set gamma b curve coefficient 54
23-16	R/W	0	**custom_b_val55** custom set gamma b curve coefficient 55
31-24	R/W	0	**custom_b_val56** custom set gamma b curve coefficient 56

GAMMA_REG_47

Offset_Address: 0x0000_01F8

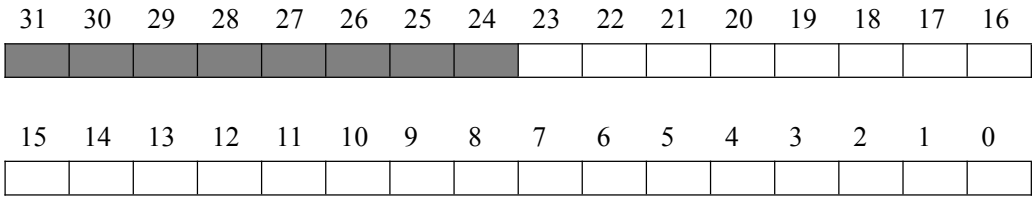
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val57** custom set gamma b curve coefficient 57
15-8	R/W	0	**custom_b_val58** custom set gamma b curve coefficient 58
23-16	R/W	0	**custom_b_val59** custom set gamma b curve coefficient 59
31-24	R/W	0	**custom_b_val60** custom set gamma b curve coefficient 60

GAMMA_REG_48

Offset_Address: 0x0000_01FC



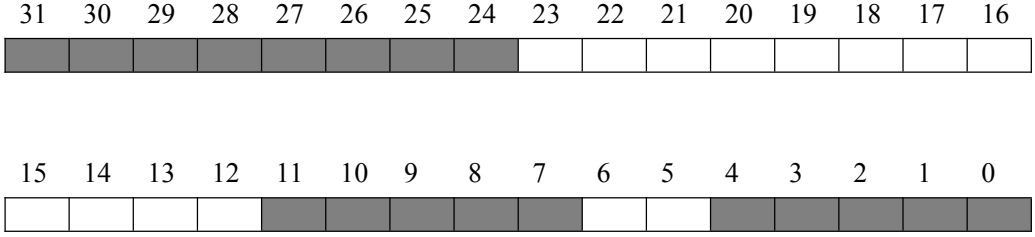
Bit	Type	Reset	Description
7-0	R/W	0	**custom_b_val61** custom set gamma b curve coefficient 61
15-8	R/W	0	**custom_b_val62** custom set gamma b curve coefficient 62
23-16	R/W	0	**custom_b_val63** custom set gamma b curve coefficient 63
31-24	R/W	0	**custom_b_val64** custom set gamma b curve coefficient 64

CPU screen & SRGB screen

SRGB_CFG

SRGB config Register

Offset_Address: 0x0000_0200



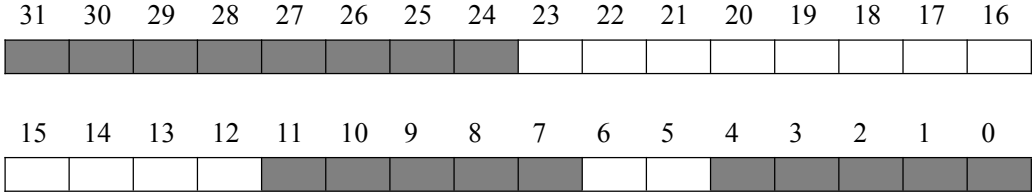
Bit	Type	Reset	Description
31-12	R	0	Reserved.
25	R/W	0	Srgb_yuv_y_sel Srgb 模式的 SYUV422 情况或 itu601 输出时，Y 与 Cb/Cr 位置选择 0： Cr Y Cb Y… … 1： Y Cr Y Cb… …
24	R/W	0	CPU_scr_v_reposition[0] CPU 屏在有 TE 功能时对 TE 信号重新进行场方向上的定位，该寄存器表示最低位。
23:12	R/W	0	CPU_scr_h_reposition[11:0] CPU 屏在有 TE 功能时对 TE 信号重新进行定位，该寄存器表示每行计数的点数，CPU_scr_h_reposition* CPU_scr_v_reposition 的点数表示输入 TE 信号后重新定位的位置。
11	R/W	0	**Srgb_disable** **1: srgb output disable, 0 is output**

			0: srgb output enable.
10	R/W	0	srgb_yuv_sel 1: yuv data is select; 0: rgb data is select.
9	R/W	0	CbCr filter mode 1: cb cr filter mode 1 0: cb cr filter mode 0
8	R/W	0	CbCr filter enable 1: cb cr filter(444 to 422) enable; 0: cb cr filter disable.
7:5	R/W	0	cf2 **odd line color setting: 000: RGB; 001:RBG; 010:BGR; 011:BRG; 100:GRB; 101:GBR; others:RGB**
4:2	R/W	0	**Cf1** **even line color setting: 000: RGB; 001:RBG; 010:BGR; 011:BRG; 100:GRB; 101:GBR; others:RGB**
1-0	R/W	0	Srgb_mode 00: through mode, R G B R G B… 01: sRGB dummy, R G B 0 R G B 0… 10: sYUV422, Cb Y Cr Y Cb Y Cr Y… 11: 0 is output

CPU_SCR_SOFT_REG

CPU Screen soft config Register

Offset_Address: 0x0000_00E0

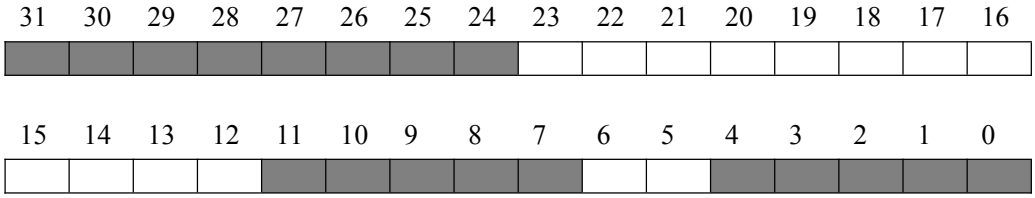


Bit	Type	Reset	Description
31-22	R	0	Reserved.
21	R/W	0	CPU_rs CPU rs signal from soft config
20	R/W	0	CPU_cs CPU cs signal from soft config
19	R/W	0	**CPU_rd** **CPU rd signal from soft config**
18	R/W	0	CPU_wr CPU wr signal from soft config.
17-0	R/W	0	Data_out 18 bits interface data output, to CPU screen for initialization.

CPU_SCR_CTRL_REG

CPU Screen control Register

Offset_Address: 0x0000_00E4



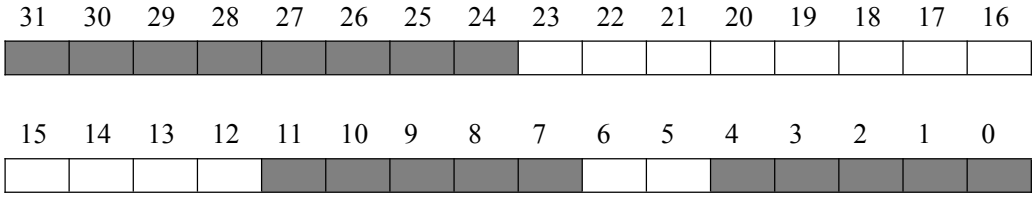
Bit	Type	Reset	Description
31-22	R/W	0	CPU_scr_v_reposition[10：1] CPU 屏在有 TE 功能时对 TE 信号重新进行场方向上的定位，该寄存器表示高 10 位。
21	R/W	0	CPU_scr_te_edge_sel FTE 信号上升沿或下降沿选择，在该沿处进行更新数据的提供。 1：上升沿； 0：下降沿。
20	R/W	0	CPU_scr_te_en CPU 屏 FTE 功能输入使能，使能时根据 CPU 屏输出的 FTE 信号提供写数据给 CPU 屏进行更新。 1：使能； 0：无效，忽略 FTE 信号输入。
19:18	R/W	0	Daten_mask 刷新的有效视频数据选择 00：正常数据； 01：延迟 2 cycle 并每行减少一个点； 10：延迟 1 cycle 并每行减少一个点； 11：延迟 2 个 cycle。
17:14	R/W	0	RAM_RD_NUM ADR_CLR_DATA_REG 更新的个数，最大为 16 个。
13:10	R/W	0	Interval_counter 更新 ADR_CLR_DATA_REG 每个的时间间隔（cycle 数）。
9	R/W	0	CPU_scr_rst CPU屏复位信号。
8	R/W	0	CPU_scr_comb 0: RGB 1: BGR
7：5	R/W	0	CPU_scr_mode 000: 18 bits 数据线，每个像素点18bits，（RGB666） 001: 18 bits 数据线，每个像素点16bits，此时bit9 bit0为0（RGB565） 010：9 bits数据线，每个像素点18bits，（RGB666） 011：8 bits数据线，每个像素点16bits，（RGB565） 100：8 bits数据线，每个像素点18bits，（RGB666）
4	R/W	0	Data_refresh_en: 数据往ＧＲＡＭ刷新使能。 １：enable, 有效视频数据每场刷新； ０：disable，有效视频数据不刷新。
3	R/W	0	Adr_clr_sel:　每场消隐更新选择 １：场同步上升沿开始更新； ０：场同步下降沿开始更新。
2	R/W	0	**Adr_clr_en: 每场消隐期间是否更新**

			ADR_CLR_DATA_REG中存放的命令。 1：enable, 每场消隐更新，更新个数由RAM_RD_NUM决定。 0: disable, 每场消隐不更新。
1	R/W	0	Cmd_mode 由 0 －＞ 1 上升沿触发命令模式，该模式下，硬件将ADR_CLR_DATA_REG中存放的命令写入ｃｐｕ屏，写入的个数由RAM_RD_NUM决定。
0	R/W	0	Soft_mode 1: soft mode enable, when set 1, CPU screen interface signal is config by CPU_SCR_SOFT_REG . 0: hard mode enable, when set 0, CPU screen interface signal is config by ADR_CLR_DATA_REG or active video signal.

ADR_CLR_DATA_REG0

CPU Screen update register 0

Offset_Address: 0x0000_00E8

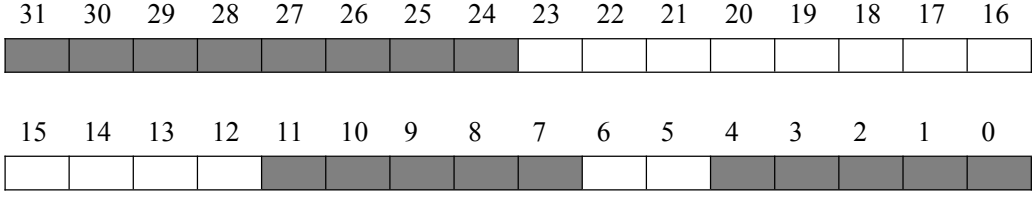


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_reg0 硬件 cmd 或每场消隐更新 CPU 屏的参数。 19bits：rs 18bits：wr 17：0： data

ADR_CLR_DATA_REG1

CPU Screen update register 1

Offset_Address: 0x0000_00EC

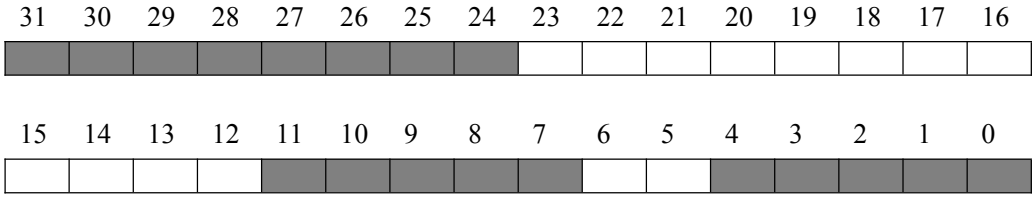


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_reg1 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REG2

CPU Screen update register 2

Offset_Address: 0x0000_00F0

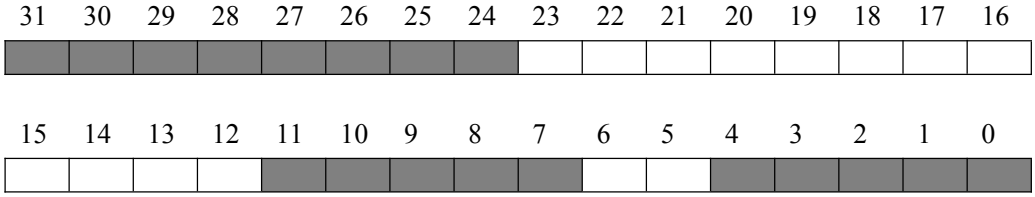


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_reg2 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REG3

CPU Screen update register 3

Offset_Address: 0x0000_00F4

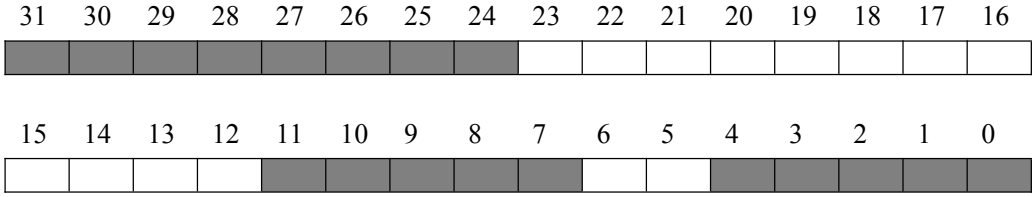


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_reg3 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REG4

CPU Screen update register 4

Offset_Address: 0x0000_00F8



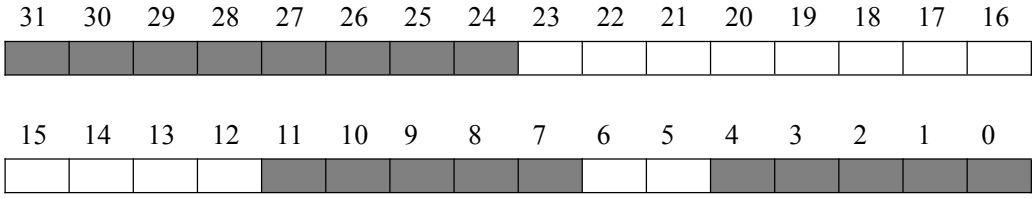
Bit	Type	Reset	Description
31-20	R	0	Reserved.

19:0	R/W	0	Adr_clr_data_reg4 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REG5

CPU Screen update register 5

Offset_Address: 0x0000_00FC

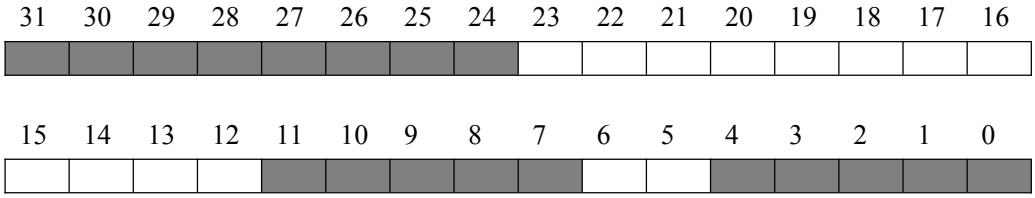


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_reg5 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REG6

CPU Screen update register 6

Offset_Address: 0x0000_0100

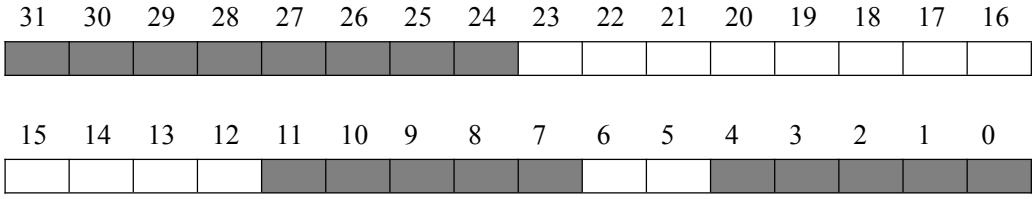


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_reg6 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REG7

CPU Screen update register 7

Offset_Address: 0x0000_0104

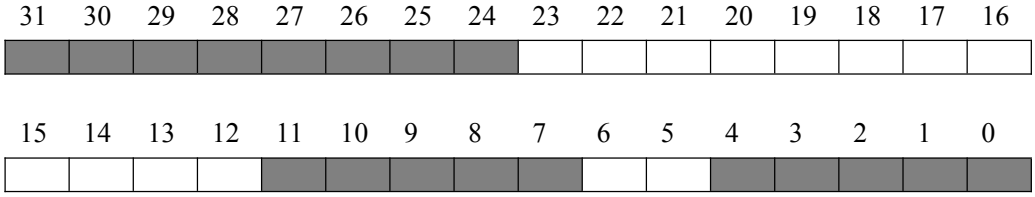


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_reg7 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REG8

CPU Screen update register 8

Offset_Address: 0x0000_0108

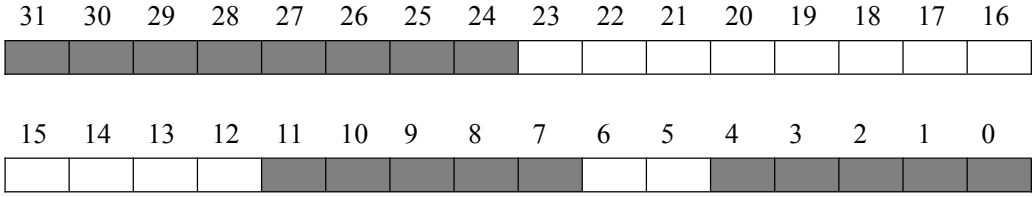


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_reg8 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REG9

CPU Screen update register 9

Offset_Address: 0x0000_010C

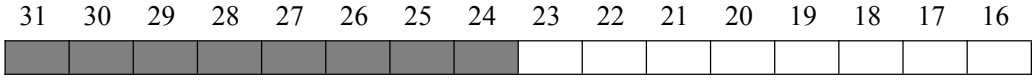


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_reg9 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REGA

CPU Screen update register A

Offset_Address: 0x0000_0110

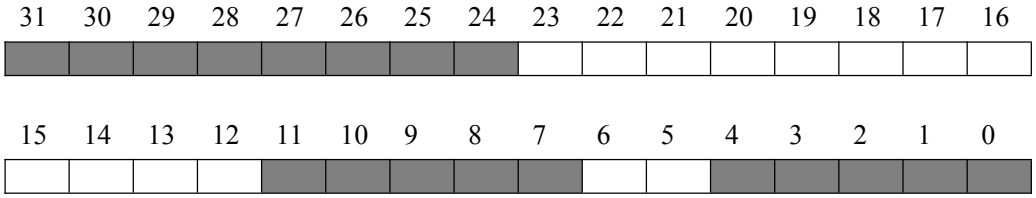


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_rega 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REGB

CPU Screen update register B

Offset_Address: 0x0000_0114

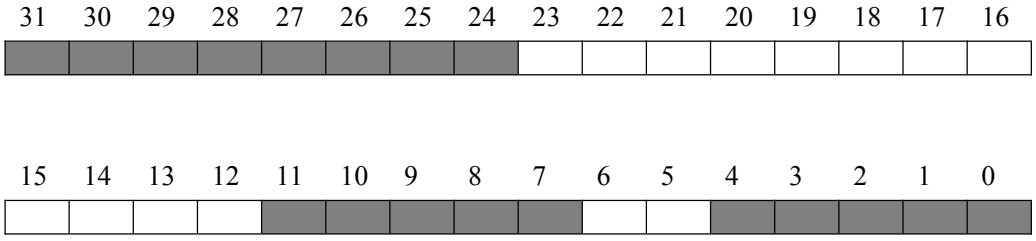


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_regb 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REGC

CPU Screen update register C

Offset_Address: 0x0000_0118

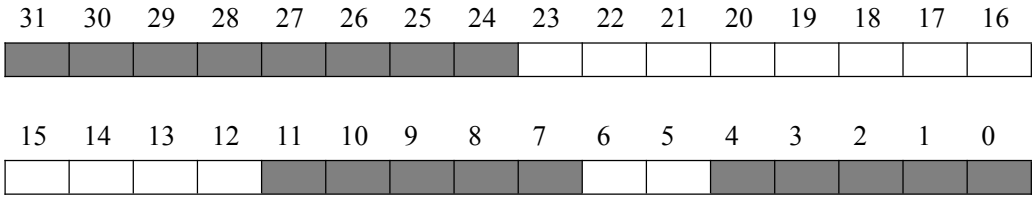


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_regc 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REGD

CPU Screen update register D

Offset_Address: 0x0000_011C

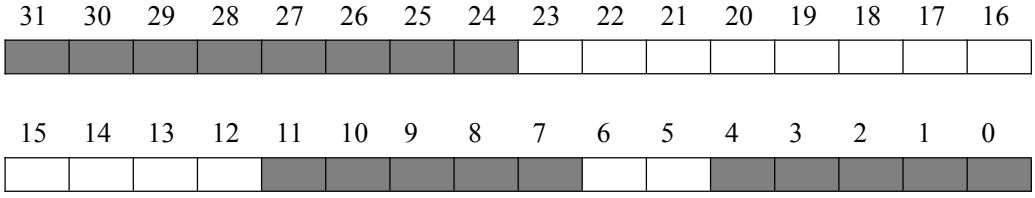


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_regd 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REGE

CPU Screen update register E

Offset_Address: 0x0000_0120

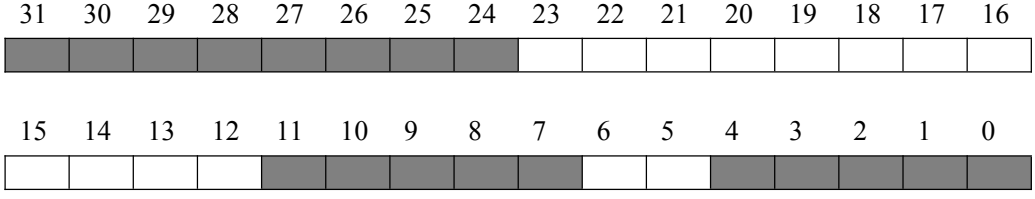


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_rege 硬件 cmd 或每场消隐更新 CPU 屏的参数。

ADR_CLR_DATA_REGF

CPU Screen update register F

Offset_Address: 0x0000_0124

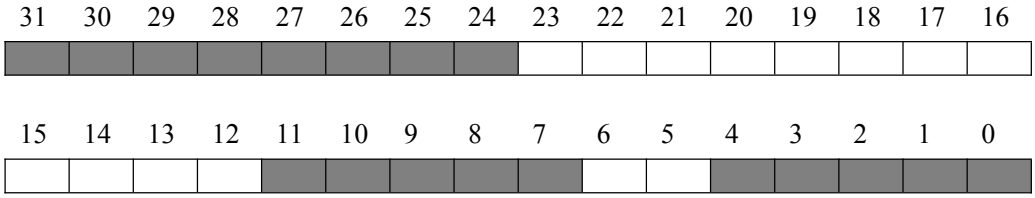


Bit	Type	Reset	Description
31-20	R	0	Reserved.
19:0	R/W	0	Adr_clr_data_regf 硬件 cmd 或每场消隐更新 CPU 屏的参数。

CPU_SCREEN_STATUS

CPU Screen status register

Offset_Address: 0x0000_0128

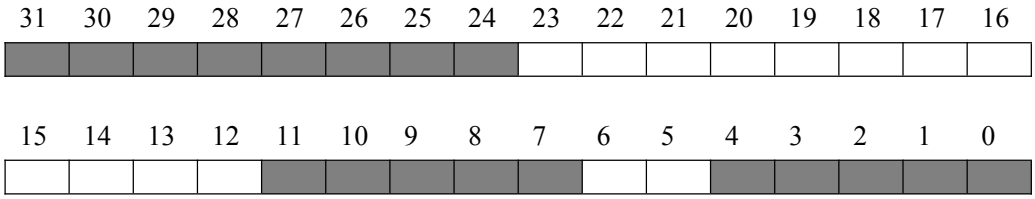


Bit	Type	Reset	Description
31-2	R	0	Reserved.
1	R	0	DATA_REFRESH_STATE CPU 屏在硬件模式，正常数据下，CS 的状态位。
0	R	0	ADR_CLR_STATE CPU 屏在硬件模式下每场更新的状态位。 0：场消隐处不更新； 1：场消隐处进行 CPU 屏初始化状态的更新。

OSD_CCM_REG0

osd0 ccm reg

Offset_Address: 0x0000_0208

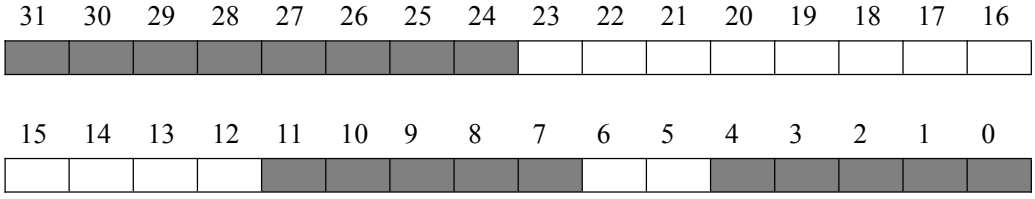


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	video layer ccm reg01
11-0	R/W	0	video layer ccm reg00

OSD_CCM_REG1

osd0 ccm reg

Offset_Address: 0x0000_020c

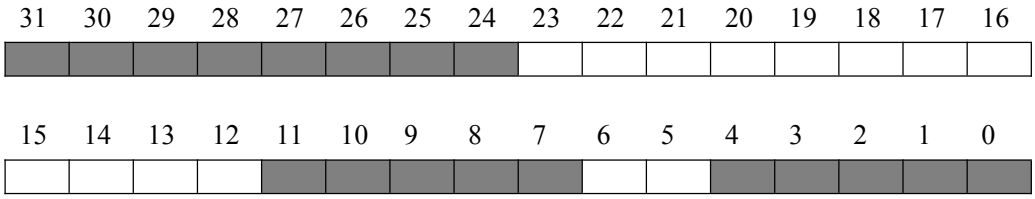


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	video layer ccm reg10
11-0	R/W	0	video layer ccm reg02

OSD_CCM_REG2

osd0 ccm reg

Offset_Address: 0x0000_0210

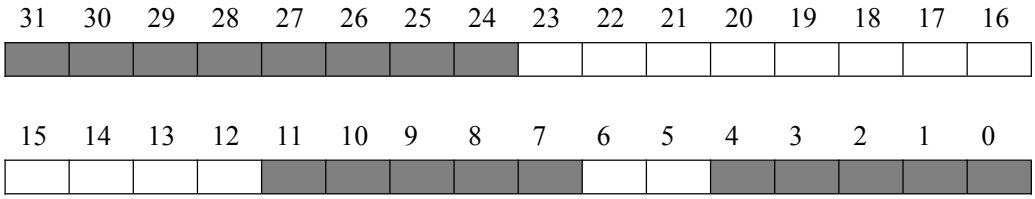


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	video layer ccm reg12
11-0	R/W	0	video layer ccm reg11

OSD_CCM_REG3

osd0 ccm reg

Offset_Address: 0x0000_0214

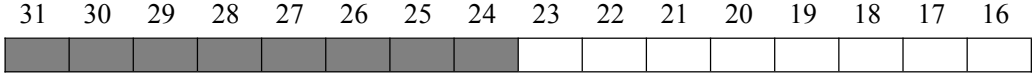


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	video layer ccm reg21
11-0	R/W	0	video layer ccm reg20

OSD_CCM_REG4

osd0 ccm reg

Offset_Address: 0x0000_0218

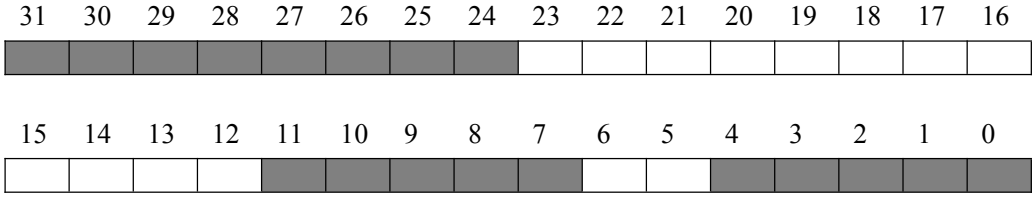


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	video layer ccm reg30
11-0	R/W	0	video layer ccm reg22

OSD_CCM_REG5

osd0 ccm reg

Offset_Address: 0x0000_021c

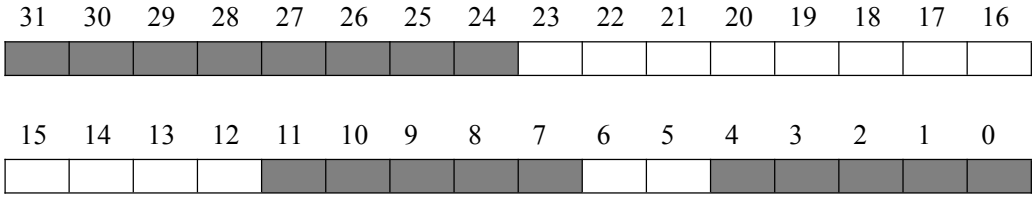


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	video layer ccm reg32
11-0	R/W	0	video layer ccm reg31

GLOBAL_CCM_REG0

Global ccm reg

Offset_Address: 0x0000_0238

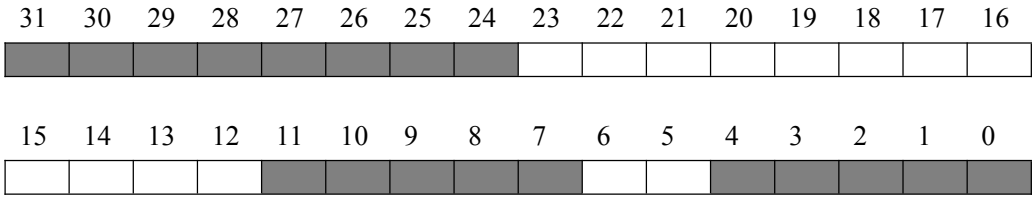


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	global ccm reg01
11-0	R/W	0	global ccm reg00

GLOBAL_CCM_REG0

Global ccm reg

Offset_Address: 0x0000_023c

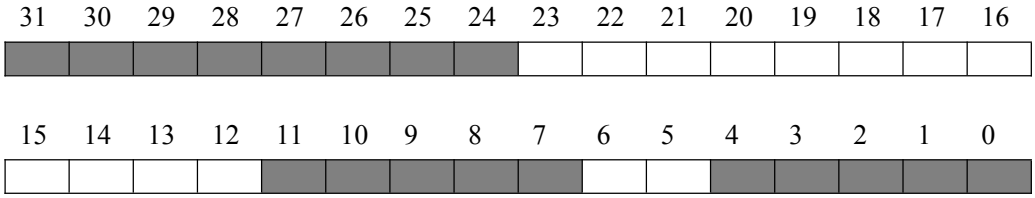


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	global ccm reg10
11-0	R/W	0	Global ccm reg02

GLOBAL_CCM_REG2

global ccm reg

Offset_Address: 0x0000_0240

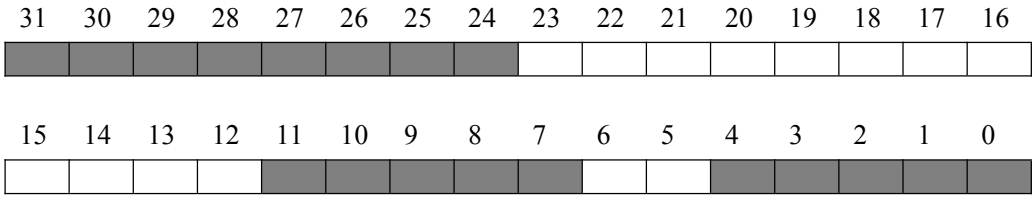


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	Global ccm reg12
11-0	R/W	0	Global ccm reg11

GLOBAL_CCM_REG3

global ccm reg

Offset_Address: 0x0000_0244



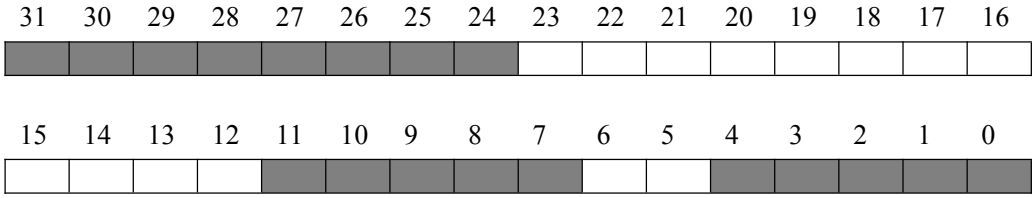
Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	Global ccm reg21

11-0	R/W	0	Global ccm reg20

GLOBAL_CCM_REG4

osd0 ccm reg

Offset_Address: 0x0000_0248

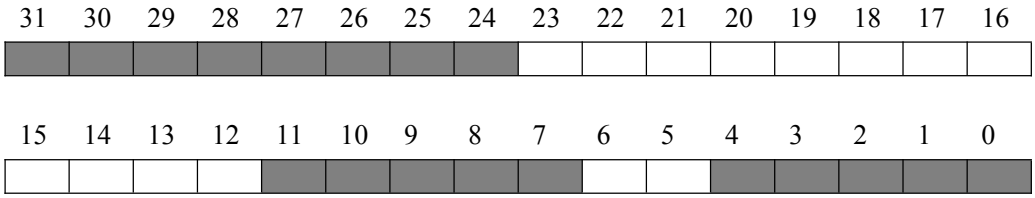


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	Global ccm reg30
11-0	R/W	0	global ccm reg22

GLOBAL_CCM_REG5

osd0 ccm reg

Offset_Address: 0x0000_024c

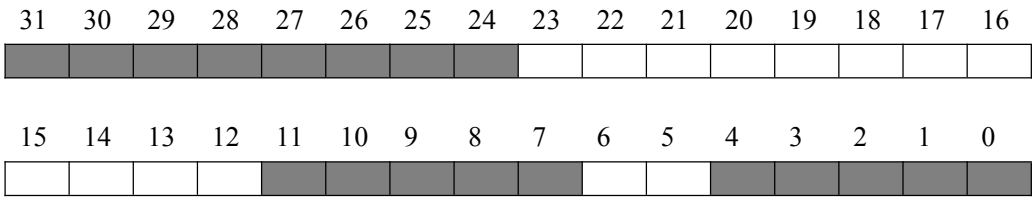


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	Global ccm reg32
11-0	R/W	0	Global ccm reg31

CCM_EN_REG

osd0 ccm reg

Offset_Address: 0x0000_0250

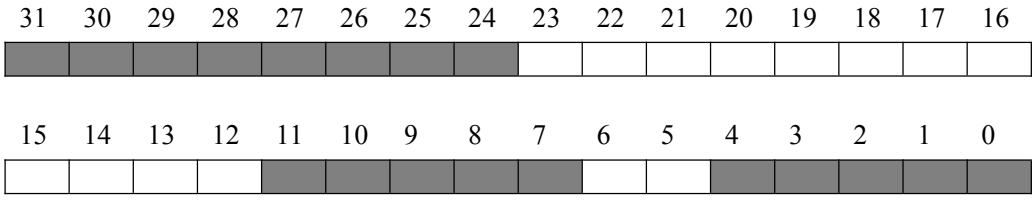


Bit	Type	Reset	Description
31-24	R	0	Reserved.
1	R/W	0	Global ccm en
0	R/W	0	Video layer ccm en

ITU_EAV_CODE _CFG

osd0 ccm reg

Offset_Address: 0x0000_0254

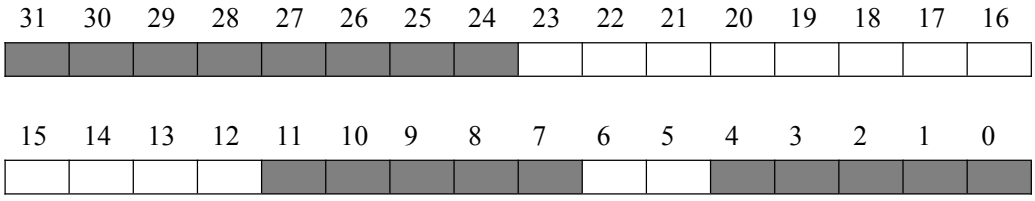


Bit	Type	Reset	Description
31-0	R	0	656 GEN EAV CODE CFG

ITU_SAV_CODE _CFG

osd0 ccm reg

Offset_Address: 0x0000_0258

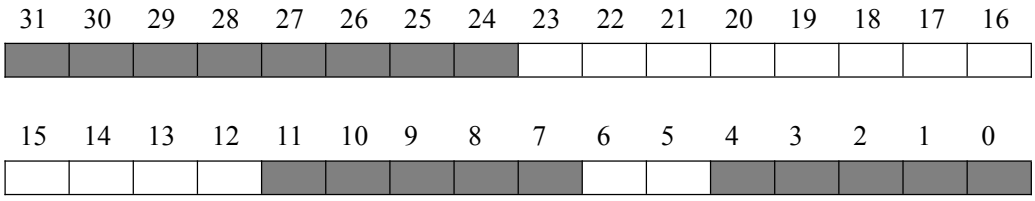


Bit	Type	Reset	Description
31-0	R	0	656 GEN SAV CODE CFG

HLOCK_CFG_RG

HLOCK

Offset_Address: 0x0000_025c



Bit	Type	Reset	Description
31-18	R/W	0	
17-16	R/W	0	axi_max_lenth, arlen 0: 16 ; 1: 32 ; 2: 64 ;
15-13	R	0	reserved
12:0	R/W	0	Hlock size

17. ITU

17.1 Features

Support ITU601/656, YUV422/420 input up to 1920x1080.
Auto-adapt to the format and the resolution of the input.
Support Stepless Scaler.
Support 90、180、270 rotate and flip.

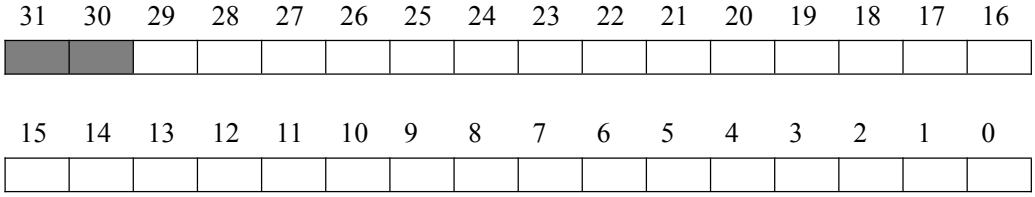
17.2 Itu Register Descriptions

BASE :0x4080_0000

Itu_ctl

Itu control register

Offset_Address: 0x0000_0000



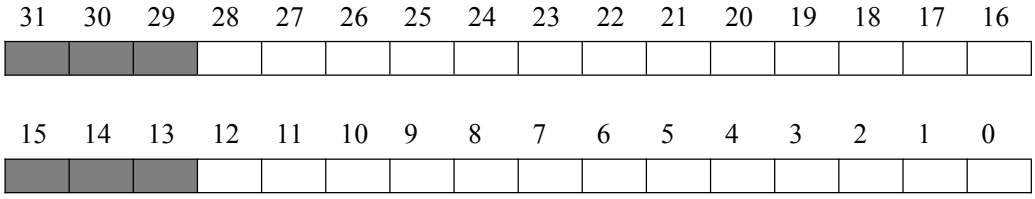
Bit	Type	Reset	Description
31-30	R	0	**reserved**
29	R/W	0	field_mode. 0:normal mode. 1:single field mode.
28	R/W	1	allow_mode. image write to frame buffer allow mode select. 0: software control completely. 1: software allow and hardware pause.
27	R	0	**reserved**
26	R/W	0	rgb_mode 0：y_uv422 or y_uv420 mode. 1: rgb mode.
25	R/W	0	Video_dat_sel1 Mipi_de select. 0:mipi_de 1:mipi hs&de

24	R/W	0	Video_dat_sel0 0:itu input 1:mipi input
23	R/W	0	video_in_inv_reg3 0:video_field normal 1:video_field_inv invert
22	R/W	0	video_in_inv_reg2 0:de normal 1:de invert
21	R/W	0	video_in_inv_reg1 0:video_vsyn normal 1:video_vsyn invert
20	R/W	0	video_in_inv_reg0 0:video_hsyn normal 1:video_hsyn invert
19	R/W	0	SCALE_BYPASS 0:scale enable 1:scale bypass
18	R/W	0	cr_sel 0:cr current 1:cr last
17-8	R/W	0	hmir_addr,0-1024 Horizontal mirror address,usually configured as last dot address.
7	R/W	0	yuv420 enable. 0: y_uv422 1:y_uv420.
6	R/W	0	mode_en Wirte image mode enable.
5	R/W	0	hmirror 0:normal 1:horizontal mirror
4	R/W	0	vflip 0:normal 1:vertical filp
3-2	R/W	0	fnum_sel 0: 2 frames mode 1: 3 frame mode 2: 4 frame mode
1	R/W	0	cbcr_inv 0:normal 1: cb cr invert
0	R	0	dma_en 0: dma not enable 1:dma enable.

image_width_reg

image_act_width and image_total_width register

Offset_Address: 0x0000_0004



Bit	Type	Reset	Description
31-29	R	0	**reserved**
28-16	R/W	0	image_total_width Image total width.
15-13	R	0	**reserved**
12-0	R/W	0	image_act_width Image active width.

yfram0_addr_start

yfram0_addr_start register

Offset_Address: 0x0000_0008

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

Bit	Type	Reset	Description
31-0	R/W	0	yfram0_addr_start. Y frame buffer0 start address in yuv mode or RGB frame buffer0 start address in rgb mode.

yfram1_addr_start

yfram1_addr_start register

Offset_Address: 0x0000_000c

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

Bit	Type	Reset	Description
31-0	R/W	0	yfram1_addr_start. Y frame buffer1 start address in yuv mode or RGB frame buffer1 start address in rgb mode.

yfram2_addr_start

yfram2_addr_start register

Offset_Address: 0x0000_0010

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

Bit	Type	Reset	Description
31-0	R/W	0	yfram2_addr_start. Y frame buffer2 start address in yuv mode or RGB frame buffer2 start address in rgb mode.

yfram3_addr_start

yfram3_addr_start register

Offset_Address: 0x0000_0014

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

Bit	Type	Reset	Description
31-0	R/W	0	yfram3_addr_start. Y frame buffer3 start address in yuv mode or RGB frame buffer3 start address in rgb mode.

yfram0_addr_flip

yfram0_addr_flip register

Offset_Address: 0x0000_0018

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

Bit	Type	Reset	Description
31-0	R/W	0	yfram0_addr_flip. Y frame buffer0 flip address in yuv mode or RGB frame buffer0 flip address in rgb mode.

yfram1_addr_flip

yfram1_addr_flip register

Offset_Address: 0x0000_001c

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

Bit	Type	Reset	Description
31-0	R/W	0	yfram1_addr_flip. Y frame buffer1 flip address in yuv mode or RGB frame buffer1 flip address in rgb mode.

yfram2_addr_flip

yfram2_addr_flip register

Offset_Address: 0x0000_0020

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

Bit	Type	Reset	Description

31-0	R/W	0	yfram2_addr_flip. Y frame buffer2 flip address in yuv mode or RGB frame buffer2 flip address in rgb mode.

yfram3_addr_flip

yfram3_addr_flip register

Offset_Address: 0x0000_0024

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

Bit	Type	Reset	Description
31-0	R/W	0	yfram3_addr_flip. Y frame buffer3 flip address in yuv mode or RGB frame buffer3 flip address in rgb mode.

cfram0_addr_start

cfram0_addr_start register

Offset_Address: 0x0000_0028

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

Bit	Type	Reset	Description
31-0	R/W	0	cfram0_addr_start. C frame buffer0 start address.

cfram1_addr_start

cfram1_addr_start register

Offset_Address: 0x0000_002c

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

Bit	Type	Reset	Description
31-0	R/W	0	cfram1_addr_start. C frame buffer1 start address.

cfram2_addr_start

cfram2_addr_start register

Offset_Address: 0x0000_00030

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

Bit	Type	Reset	Description
31-0	R/W	0	cfram2_addr_start. C frame buffer2 start address.

cfram3_addr_start

cfram3_addr_start register

Offset_Address: 0x0000_0034

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

Bit	Type	Reset	Description
31-0	R/W	0	cfram3_addr_start. C frame buffer3 start address.

cfram0_addr_flip

cfram0_addr_flip register

Offset_Address: 0x0000_0038

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

Bit	Type	Reset	Description
31-0	R/W	0	cfram0_addr_flip. C frame buffer0 flip address.

cfram1_addr_flip

cfram1_addr_flip register

Offset_Address: 0x0000_003c

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

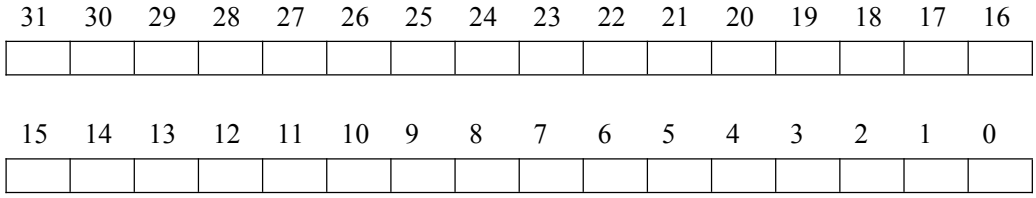
Bit	Type	Reset	Description

31-0	R/W	0	cfram1_addr_flip. C frame buffer1 flip address.

cfram2_addr_flip

cfram2_addr_flip register

Offset_Address: 0x0000_0040

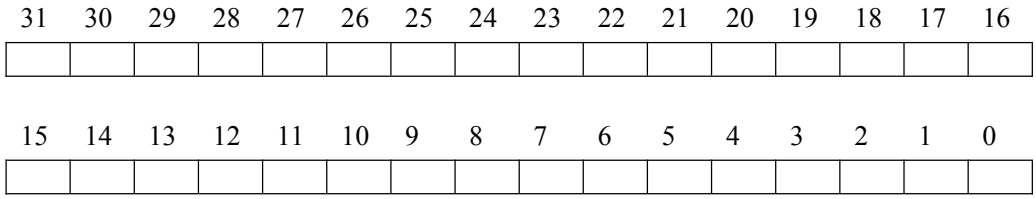


Bit	Type	Reset	Description
31-0	R/W	0	cfram2_addr_flip. C frame buffer2 flip address.

cfram3_addr_flip

cfram3_addr_flip register

Offset_Address: 0x0000_0044

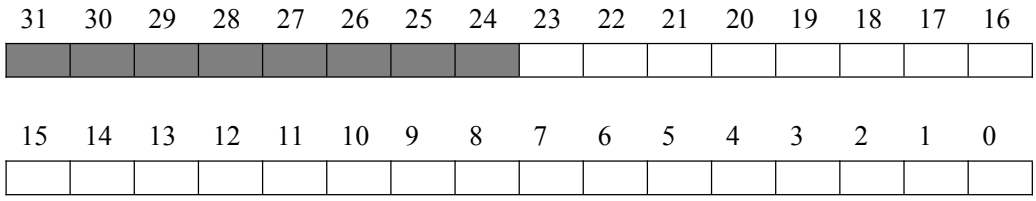


Bit	Type	Reset	Description
31-0	R/W	0	cfram3_addr_flip. C frame buffer3 flip address.

rec_hst/rec_hsp

rec_hst/rec_hsp

Offset_Address: 0x0000_0048

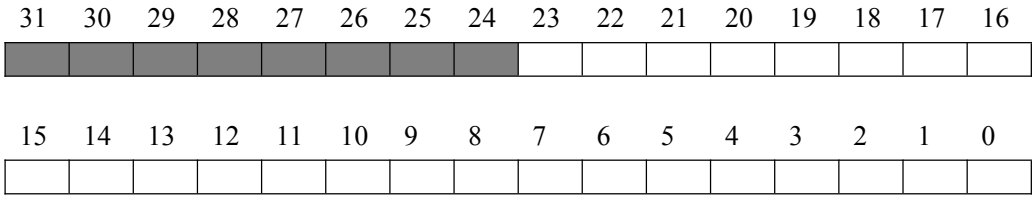


Bit	Type	Reset	Description
31-24	R	0	**reserved**
23-12	R/W	0	rec_hsp. Reference image horizontal stop.
11-0	R/W	0	rec_hst Reference image horizontal start.

rec_vst/rec_vsp

rec_vst/rec_vsp

Offset_Address: 0x0000_004c

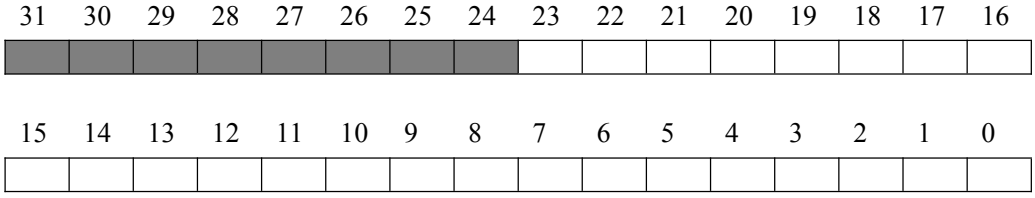


Bit	Type	Reset	Description
31-24	R	0	**reserved**
23-12	R/W	0	rec_vsp. Reference image vertical stop.
11-0	R/W	0	rec_vst. Reference image vertical start.

win_hst/win_hsp

win_hst/win_hsp

Offset_Address: 0x0000_0050

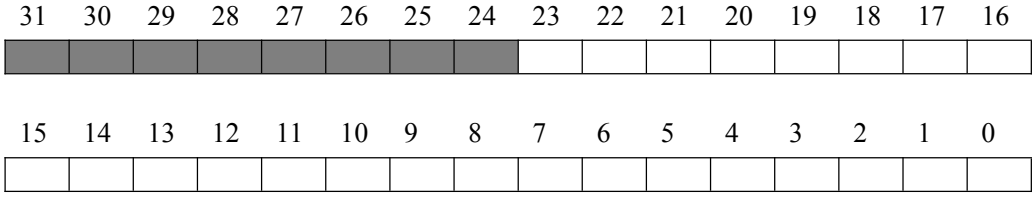


Bit	Type	Reset	Description
31-24	R	0	**reserved**
23-12	R/W	0	win_hsp. Image window horizontal stop position.
11-0	R/W	0	win_hst Image window horizontal start position.

win_vst/win_vsp

win_vst/win_vsp

Offset_Address: 0x0000_0054

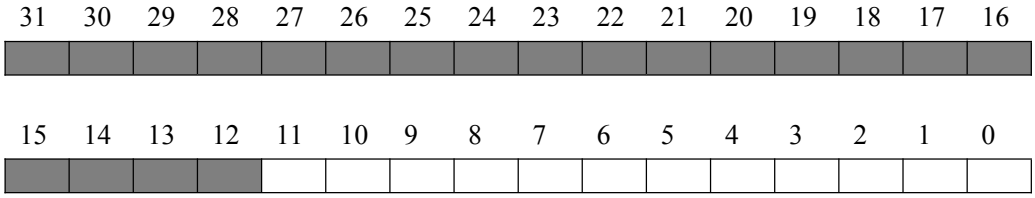


Bit	Type	Reset	Description
31-24	R	0	**reserved**
23-12	R/W	0	win_vsp. Image window vertical stop position.
11-0	R/W	0	win_vst. Image window vertical start position.

Itu_en_reg

Itu_en register

Offset_Address: 0x0000_0058

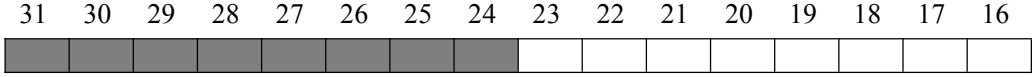


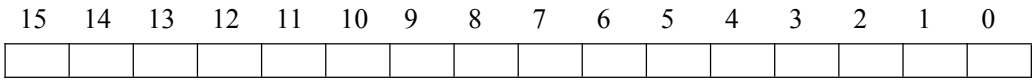
Bit	Type	Reset	Description
31-12	R	0	**reserved**
11	R/W	0	win_vedge_sel
10	R/W	0	win_hedge_sel
9	R/W	0	ituvsync_inv itu_vsync_inc 1:Use inv itu_vsync
8	R/W	0	ituhsync_inv itu_hsync_inc 1:Use inv itu_hsync
7-6	R/W	0	itu_data_mode
5-4	R/W	0	itu_mode 0: 656 8BIT 1:BIT1120 16BIT(not support in this version) 2:BIT 601 8BIT 3:BIT 601 16BIT (not support in this version)
3	R/W	0	re_mode 1:enable 656 use crc 0:disable 656 use crc
2	R/W	0	yuyv_mode 1: yuyv 0: uyvy
1	R/W	0	cbcr_inv_mode 1:Cb CR 0: CR CB
0	R/W	0	itu_en (itu module enable) 1:enable 0:disable

itu_hstart/itu_hstop

itu_hstart/itu_hstop register

Offset_Address: 0x0000_005c



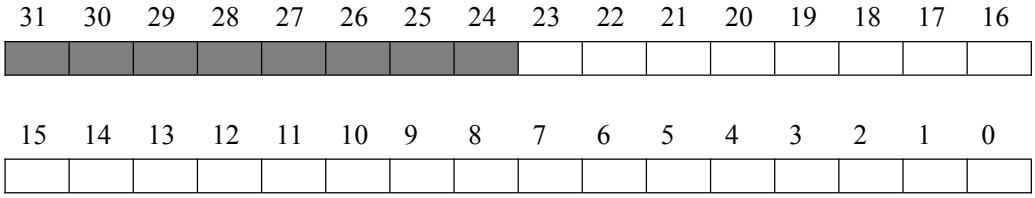


Bit	Type	Reset	Description
31-12	R	0	**reserved**
23-12	R/W	0	itu_hstop Itu horizontal stop position
11-0	R/W	0	itu_hstart Itu horizontal start position

itu_vstart/itu_vstop

itu_vstart/itu_vstop register

Offset_Address: 0x0000_0060

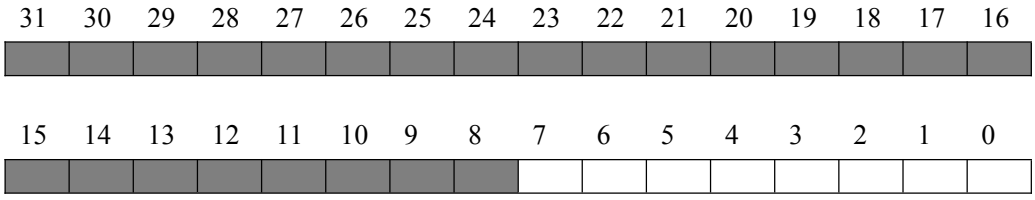


Bit	Type	Reset	Description
31-12	R	0	**reserved**
23-12	R/W	0	itu_vstop Itu vertical stop position
11-0	R/W	0	itu_vstart Itu vertical start position

global_alpha/test

Rgb global alpha register

Offset_Address: 0x0000_0064

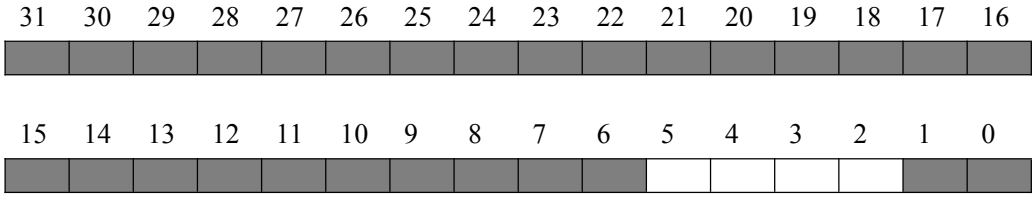


Bit	Type	Reset	Description
31-8	R	0	**reserved**
17	R/W	0	Test_y_acc enable
16	R/W	0	Test_y_reg enable
15-8	R/W	0	Test_y register,When Test_y_reg enable,this register is active.
7-0	R/W	0	global_alpha. Global alpha value, active when rgb mode is 1.

Itu_wr_stu

Itu write status register

Offset_Address: 0x0000_006c

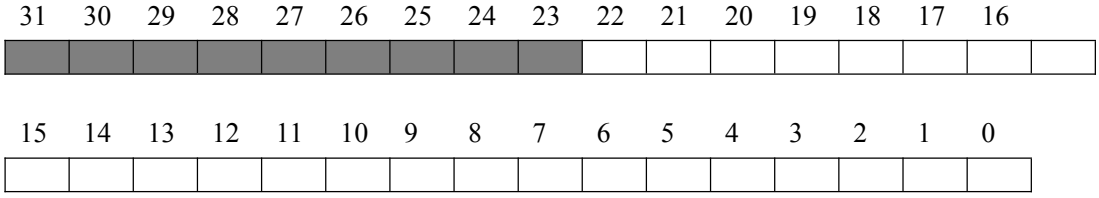


Bit	Type	Reset	Description
31-2	R	0	**reserved**
1-0	R	0	frame_wnum. Current write frame number.

Itu_scal_clk_rst

LUM_STEP_REG/offset_position register

Offset_Address: 0x0000_0070



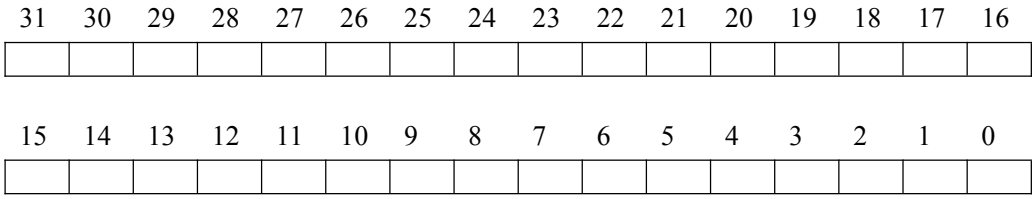
Bit	Type	Reset	Description
31-27	R	0	**reserved**
25-20	R/W	0	sca_clk_div_reg Scale_clk divison register scaler_clk_pre=clk_scale_pll/sca_clk_div_reg.
19	R/W	0	clk_itu_inv Itu clk in invert sel. 0:normal 1: invert
18-12	R/W	0	clk_itu_dly_reg Itu clk in delay sel.
11	R/W	0	video_clk_sel[1]. 0: clk_video_mipi= mipi_clk; 1: clk_video_mipi=clk_video_pre.
11-10	R/W	0	video_clk_sel[0]. 0: clk_video_pre=itu_svck_div; 1: clk_video_pre=itu_svck_div.
8-4	R/W	0x1f	softreset_reg[4:0]. bit0: itu_rst_n. bit1: resetn_clkvideo. bit2:resetn_clkscale. bit3:resetn_clkhscal. bit4:resetn_clkvidscal.
3-0	R/W	0xf	clk_gate_en[3:0] Bit0:osc_clk gate enable.

			Bit1:itu_clk gate enable. Bit2:clk_scale gate enable. Bit3:clk_hscale gate enable.

vgate_start_1/vgate_stop_1

vgate_start/vgate_stop register

Offset_Address: 0x0000_0080

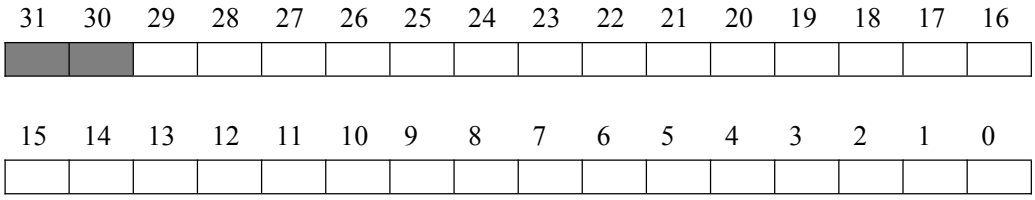


Bit	Type	Reset	Description
31-16	R/W	0	vgate_stop_1
15-0	R/W	0	vgate_start_1

SCALE_CTRL_REG

SCALE_CTRL_REG

Offset_Address: 0x0000_0084



Bit	Type	Reset	Description
31	R	0	reserved
30	R/W	0	scale_de_sel
29	R/W	0	cb_delay_sel 0: original data; 1: cb delay
28	R/W	0	cb_ahead_sel 0: original data; 1: cb ahead
27	R/W	0	vsyn_out_sel 0: vsync ; 1:vgate
26	R/W	0	hsyn_out_sel 0: hsync ; 1:hgate
25-24	R/W	0	h_flt_sel 水平方向滤波选择 0：直通，不做滤波； 1：2 阶滤波； 2：3 阶滤波； 3：4 阶滤波。
23	R/W	0	COS_EN 非线性 scaler 使能
22	R/W	0	reserved
21	R/W	0	HU_SCA_BYPA 0: not bypass hup scal; 1: bypass
20	R/W	0	HD_SCA_BYPA 0: not bypass hdown scale; 1: bypass

19	R/W	0	H_SCALE_BYPASS_SEL 0: auto bypass hscale; 1: load from bypass register
18	R/W	0	vsyn_del_field 场（奇场或偶场）起始位置的延迟选择使能打开时，该信号表示选择的是奇场还是偶场。 1: 奇场 0: 偶场
17	R/W	0	vsyn_del_en 场（奇场或偶场）起始位置的延迟选择使能，当使能打开时，可改变奇场或偶场的起始位置，以改变重建后输出场的位置。 1: enable; 0: disable.
16	R/W	0	DEN_BYPASS 0: not bypass daten; 1: bypass daten
15	R/W	0	FIELD_BYPASS 0: not bypass field; 1: bypass field
14	R/W	0	VSYN_BYPASS 0: not bypass vsync; 1: bypass vsync
13	R/W	0	HSYN_BYPASS 0: not bypass hsync; 1: bypass hsync
12	R/W	0	field_no_change Whether the odd field and even field is treated as the same, it is the same function to the deinterlacer. 1: the odd field and the even field is treated as the same, not do deintertlace 0: the odd field and the even field is treated as the different, do deinterlace
11	R/W	0	datena_inv 0: not invert daten; 1: invert daten
10	R/W	0	vsync_inv 0: not invert vsync; 1: invert vsync
9	R/W	0	hsync_inv 0: not invert hsync; 1: invert hsync
8	R/W	0	FIELDINV 0: not invert field; 1: invert field
7	R/W	0	CbCr_align Cb and Cr mode select if the filter enable of Cb and Cr in 444 to 422 transfer set 1(crcbfilter_en=1), if the crcbfilter_en is set to 0, this register is useless. 1: Y0Cb0 Y1Cr0 Y2Cb2 Y3Cb2 …… 0: Y0Cb0 Y1Cr1 Y2Cb2 Y3Cr3 ……
6	R/W	0	crcbfilter_en 0: filter disable; 1: filter enable
5	R/W	0	firstiscr 0: First is Cr; 1: First is Cb;
4	R/W	0	CR_CUT_LSB 0: Original data; 1: Cut CR LSB;
3	R/W	0	CB_CUT_LSB 0: Original data; 1: Cut CB LSB;
2	R/W	0	Y_CUT_LSB 0: Original data; 1: Cut Y LSB;
1	R/W	0	vsyn_full_sel
0	R/W	0	pn_sel_disable. PAL & NTSC use same register 0: PAL use PAL registers while NTSC use NTSC registers; 1: PAL and NTSC use the same registers (NTSC registers). Unsed in AHD

vsyn_del/HFZ

vsyn_del/HFZ

Offset_Address: 0x0000_0088

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

Bit	Type	Reset	Description
31-16	R/W	0	HFZ. Horizontal scale parameter
15-0	R/W	0	vsyn_del. 场开始（奇场或偶场）标志的延迟点数

VFZ_reg/Thlen_reg

VFZ_reg/Thlen_reg

Offset_Address: 0x0000_008c

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

Bit	Type	Reset	Description
31-16	R/W	0	Thlen. Total pixels per line
15-0	R/W	0	VFZ. Vertical scale parameter

hsyn_start/hsyn_stop

hsyn_start/hsyn_stop

Offset_Address: 0x0000_0090

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

Bit	Type	Reset	Description
31-16	R/W	0	hsyn_stop
15-0	R/W	0	hsyn_start.

hgate_start/hgate_stop

hgate_start/hgate_stop

Offset_Address: 0x0000_0094

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

Bit	Type	Reset	Description
31-16	R/W	0	hgate_stop.
15-0	R/W	0	hgate_start.

vsyn_start/vsyn_stop

vsyn_start/vsyn_stop

Offset_Address: 0x0000_0098

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

Bit	Type	Reset	Description
31-16	R/W	0	vsyn_stop
15-0	R/W	0	vsyn_start.

vgate_start/vgate_stop

vgate_start/vgate_stop

Offset_Address: 0x0000_009c

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

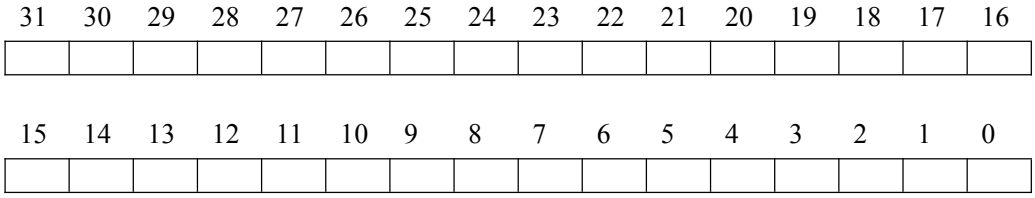
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-16	R/W	0	vgate_stop.
15-0	R/W	0	vgate_start.

img_cut_reg

Image cut register

Offset_Address: 0x0000_00a0

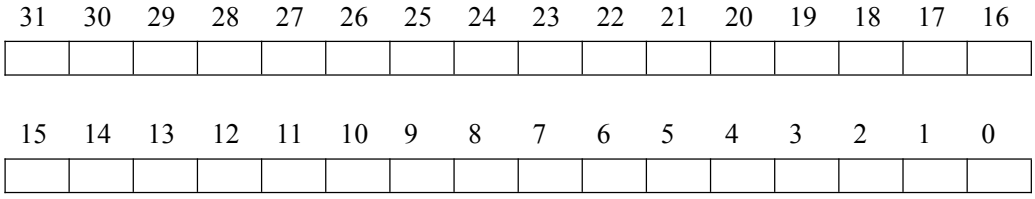


Bit	Type	Reset	Description
31-24	R/W	0	up_black_num. 垂直方向上方黑象素行数。
23-16	R/W	0	right_black_num_reg. 水平方向右边黑象素点数，另，在 4:3 模式时需要使用该功能。
15-8	R/W	0	left_black_num_reg. 水平方向左边黑象素点数，相当于将图像向右移动，另，在 4:3 模式时需要使用该功能。
7-0	R/W	0	left_cut_num 水平方向左边数据的切边，相当于将数据左移

down_black_num/hv_delay/COS_l

down_black_num/hv_delay/COS_reg register

Offset_Address: 0x0000_00a4

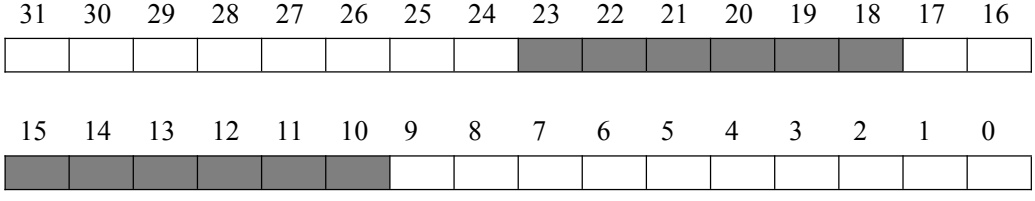


Bit	Type	Reset	Description
31-24	R/W	0	COS_LEVEL[7:0]. 水平方向非线性 scaler 的增加幅度，低 8 位。
23-16	R/W	0	COS_STEP. 水平方向非线性 scaler 的增加梯度值。
15-8	R/W	0	Delay hsync vsync out. 水平方向左边数据的切边，相当于将数据左移。
7-0	R/W	0	down_black_num. 垂直方向下方黑象素行数

COS_h/vxmod_oddf_ini/Tvlen_l

COS_h/vxmod_oddf_ini/Tvlen_l register

Offset_Address: 0x0000_00a8

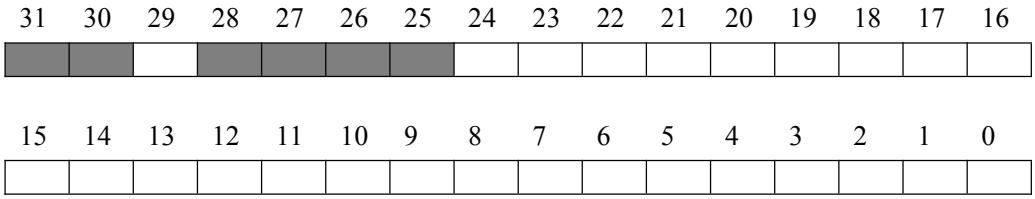


Bit	Type	Reset	Description
31-24	R/W	0	Tvlen[7:0]. Total lines per frame.
23-18	R	0	**reserved**
17-16	R/W	0	vxmod_oddf_ini[9:8]. 奇场或偶场做垂直 scaler 的起始步长，用来补偿奇偶场之间的隔行关系。
15-10	R	0	**reserved**
9-8	R/W	0	vxmod_oddf_ini[7:0]. 奇场或偶场做垂直 scaler 的起始步长，用来补偿奇偶场之间的隔行关系。
7-0	R/W	0	COS_LEVEL[9:8]. 水平方向非线性 scaler 的增加幅度，高 2 位。

Tvlen_h/scal_ctl

Tvlen_h/scal_ctl register

Offset_Address: 0x0000_00ac



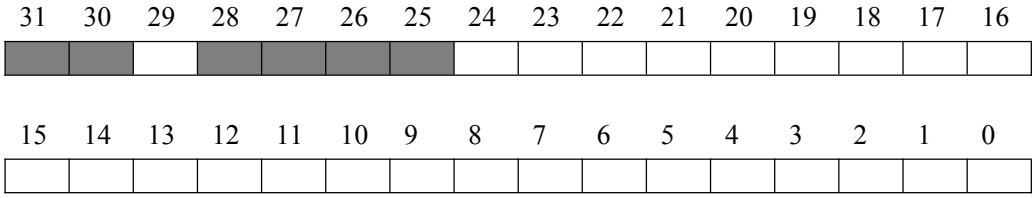
Bit	Type	Reset	Description
31-30	R	0	**reserved**
29	R/W	0	vgate_out_pos_sel.
28-25	R	0	**reserved**
24	R/W	0	pn_tvact_pos_sel
23	R/W	0	tv_active_position 配置寄存器时，TV_active 的同步位置。 1：场同步位置更新； 0：场中间位置更新。
22	R/W	0	hfz_lr_blank_en HFZ、左右显黑寄存器同步使能，此功能用来做 16:9 4:3 转换没有中间状态 0：不使能； 1：使能。
21	R/W	0	vblk_lbuf_dsn 场消隐标志来时是否继续写入行缓存空间。 0：场消隐时继续写入； 1：场消隐时不写。
20	R/W	0	hv_delay_man 0: auto delay hv; 1: load from hv_delay_reg
19-16	R/W	0	Thlen_VFZ_adj_step 当自动调整开启时，每场调节 Thlen 的步长
15	R/W	0	vfz_chg_en 垂直缩放比例 VFZ 变化时，Thlen 实际使用值是否从配置的值开始调整，高有效。

			0：VFZ 变化时，Thlen 以当前调整后的值为初始值开始调整； 1：VFZ 变化时，Thlen 以配置寄存器的值为初始值开始调整。
14	R/W	0	thlen_chg_en 寄存器配置的行总点数 Thlen 变化时，Thlen 实际使用值是否从配置的值开始调整，高有效。 0：Thlen 变化时，Thlen 实际使用值以当前调整后的值为初始值开始调整； 1：Thlen 变化时，Thlen 实际使用值以配置寄存器的值为初始值开始调整。
13	R/W	0	pn_pos_mod PN 变化时，PN 的值变化同步位置选择。 0：每场中间同步 PN 值，选 Tvlen/2 的位置； 1：每场开始同步 PN 值。
12	R/W	0	vsyn_mod_sel 场同步输出模式选择。 0：将每场信号补齐，补齐的那场行数较多； 1：每场行数最多为 Tvlen 行，不够的不补。
11	R/W	0	tv_active_disen Manual set tv_active
10	R/W	0	vsyn_out_pos_sel 0: hsync_start; 1:hsync_stop
9	R/W	0	vsyn_edge_sel 0: posedge; 1: negedge
8	R/W	0	Thlen_auto_adj_disen_reg 每行总点数（Thlen）自动调整 disable 0：自动调整开启； 1：自动调整关闭。
7-0	R/W	0	Tvlen[15:8]. Total lines per frame.

Scale_status

Tvlen_h/scal_ctl register

Offset_Address: 0x0000_00b0

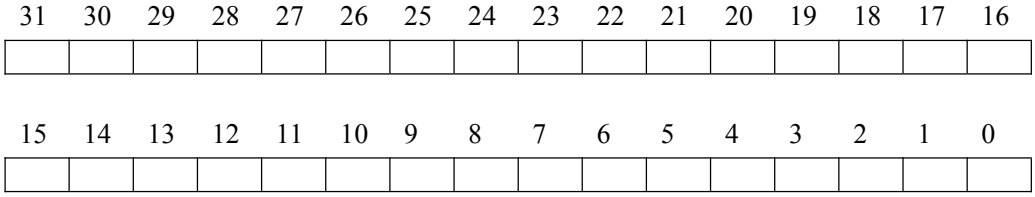


Bit	Type	Reset	Description
31-16	R	0	tvlen_read.
15-13	R	0	**reserved**
12	R	0	REGE7H[1]
11	R	0	thlen_adj_finish
10	R	0	REGE7H[0]
9	R	0	pn_sel
8	R	0	PN_detect
7-0	R	0	Thlen[15:8]

vfilt_coef0_3

Vfilt coefficient0-3

Offset_Address: 0x0000_00b4

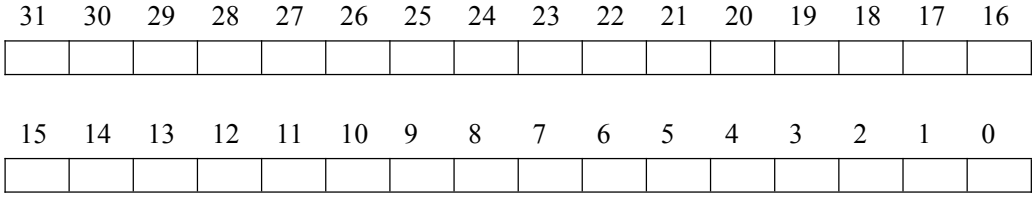


Bit	Type	Reset	Description
31-24	R/W	0	vfilt_coef3 Vfilter coefficient.
23-16	R/W	0	vfilt_coef2 Vfilter coefficient.
15-8	R/W	0	vfilt_coef1 Vfilter coefficient.
7-0	R/W	0	vfilt_coef0 Vfilter coefficient.

vfilt_coef4_6/vfilt_ctl

Vfilt coefficient4-6/vfilt_ctl

Offset_Address: 0x0000_00b8

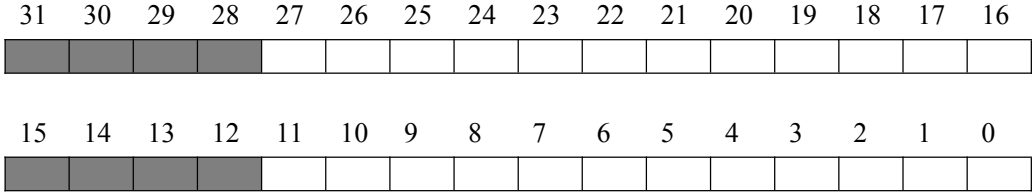


Bit	Type	Reset	Description
31-27	R/W	0	reserved
26	R/W	0	vfilt_bypass Bypass vfilter
25	R/W	0	vf_mode 00: 7 line filter; 01: 5 line filter; 10/11: 3 line filter
23-16	R/W	0	vfilt_coef6 Vfilter coefficient.
15-8	R/W	0	vfilt_coef5 Vfilter coefficient.
7-0	R/W	0	vfilt_coef4 Vfilter coefficient.

src_wide/src_high

src_wide/src_high register

Offset_Address: 0x0000_00bc

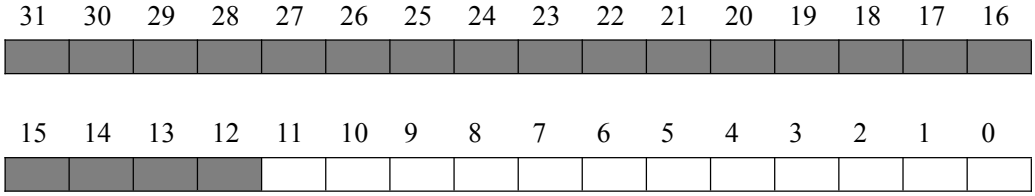


Bit	Type	Reset	Description
31-28	R	0	**reserved**
27-16	RW	0	src_high. 输入图像的高度
15-12	R	0	**reserved**
11-0	R/W	0	src_wide. 输入图像的宽度

src_wide_filter

src_wide_filter register

Offset_Address: 0x0000_00c0

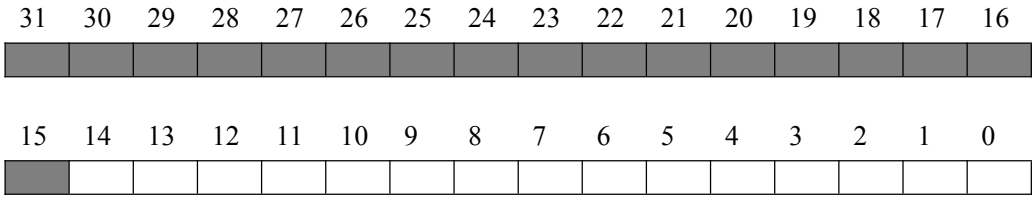


Bit	Type	Reset	Description
31-12	R	0	**reserved**
11-0	R/W	0	src_wide_filter. 输入图像的宽度

Det_reg0

Video format detect register0

Offset_Address: 0x0000_00c8

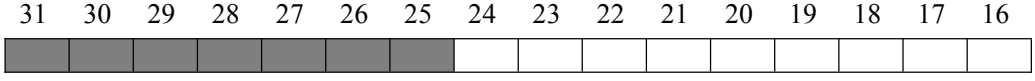


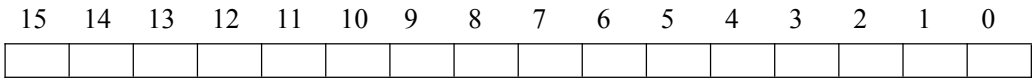
Bit	Type	Reset	Description
31-15	R	0	**reserved**
14-10	R/W	0	det_time
9-8	R/W	0	video_active_sel
7-0	R/W	0	alter_thr. Image alter threshold

Det_reg1

Video format detect register1

Offset_Address: 0x0000_00cc



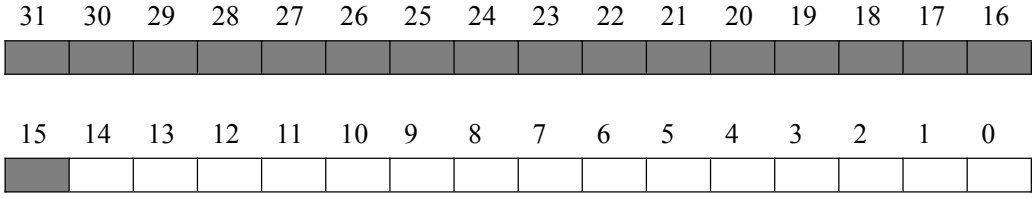


Bit	Type	Reset	Description
31-25	R	0	**reserved**
23	R/W	0	0:compare with width of pre frames. 1:compare with height_set and width_set.
22-12	R/W	0	height_set. Image height set for detect.
11-0	R/W	0	width_set. Image width set for detect.

Det_reg2

Video format detect register2

Offset_Address: 0x0000_00d0

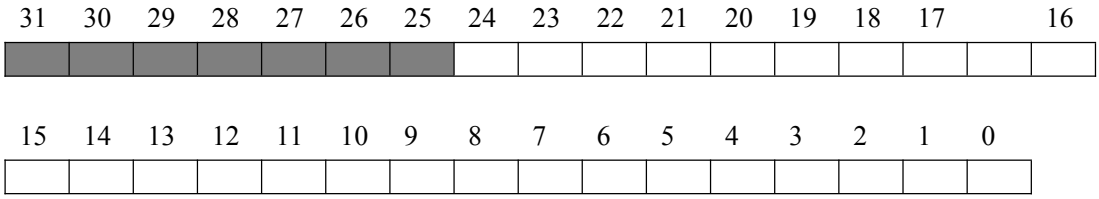


Bit	Type	Reset	Description
31-28	R	0	**reserved**
27	R	0	hv_active_int
26	R	0	hv_active
25	R	0	no_hsyn_in
24	R	0	no_vsyn_in
23-12	R	0	det_total_height
11-0	R	0	det_total_width

Det_reg3

Video format detect register3

Offset_Address: 0x0000_00d4

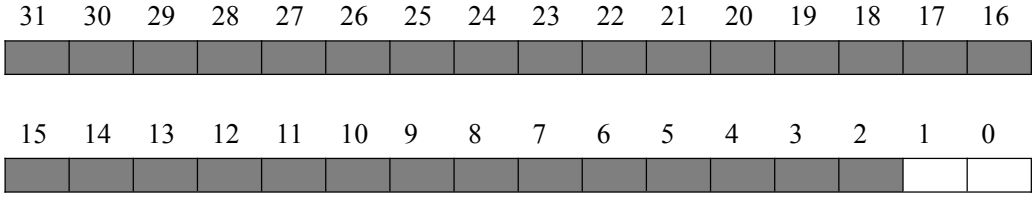


Bit	Type	Reset	Description
31-24	R	0	**reserved**
23-12	R	0	det_act_height
11-0	R	0	det_act_width

INT_EN

Itu interrupt enable control

Offset_Address: 0x0000_00d8

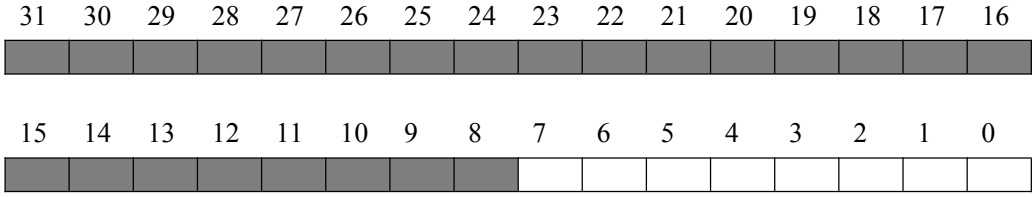


Bit	Type	Reset	Description
31-3	R	0	**reserved**
2	R/W	0	Frame write done int.
1	R/W	0	Vsyn_int_en. Vsyn interrupt enable.
0	R/W	0	hv_active_int_en . Video input active interrupt enable.

INT_STATUS

Itu status enable control

Offset_Address: 0x0000_00dc



Bit	Type	Reset	Description
31-8	R	0	**reserved**
7	R	0	itu_int status.
6	R/W	0	Frame write done int status.
5	R/W	0	Vsyn_int status. Vsyn interrupt status.
4	R/W	0	hv_active_int . Video input active interrupt status.
3	R/W	0	f3_allow. this bit may be set by hardware when frame buffer 3 has been written,while software write 0 can release the frame buffer 3 space to hardware.
2	R/W	0	f2_allow. this bit may be set by hardware when frame buffer 2 has been written,while software write 0 can release the frame buffer 2 space to hardware.
1	R/W	0	f1_allow. this bit may be set by hardware when frame buffer 1 has been written,while software write 0 can release the frame buffer 1 space to

			hardware.
0	R/W	0	f0_allow. this bit may be set by hardware when frame buffer 0 has been written,while software write 0 can release the frame buffer 0 space to hardware.

18. Timer Controller

18.1 Features

1)Up to eight programmable timers
2)Configurable timer width: 8 to 32 bits
3)Support for two operation modes: free-running and user-defined count
4)Support for independent clocking of timers

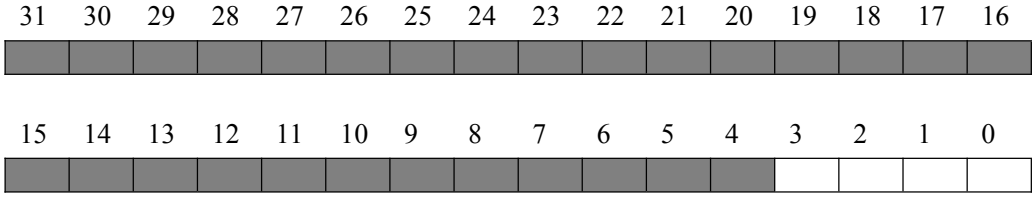
18.2 Register Descriptions

TIMER_REG_BASE(CPU): 0x50A0_0000
TIMER_REG_BASE:(MCU) 0x5520_0000

18.2.1Timer1 LoadCount : Timer1 Load Count Register

Value to be loaded into Timer1

Address: TIMER_REG_BASE + 0x00

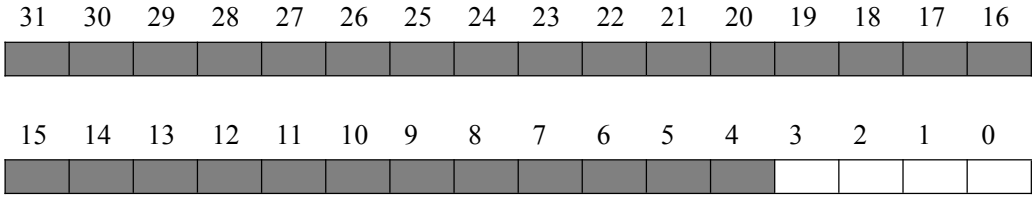


Bit	Type	Reset	Description
31-0	R/W	0	Value to be loaded into Timer 1. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.

18.2.2Timer1CurrentValue

Current value of Timer 1

Address: TIMER_REG_BASE + 0x04



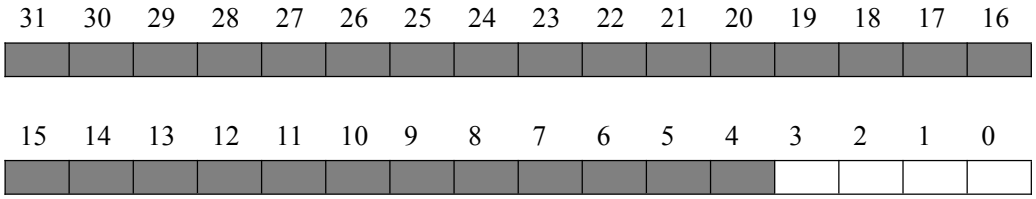
Bit	Type	Reset	Description
31-0	R	0	Current Value of Timer 1. When TIM_NEWMODE=0, This

			register is supported only when timer_1_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. When TIM_NEWMODE=1, no restrictions apply.

18.2.3Timer1ControlReg: Timer 1 Control Register

Control Register for Timer 1. This register controls enabling, operating mode (free-running or defined-count), and interrupt mask of Timer 1. You can program each Timer1ControlReg to enable or disable a specific timer and to control its mode of operation.

Address: TIMER_REG_BASE + 0x08

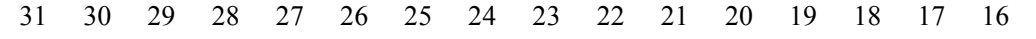


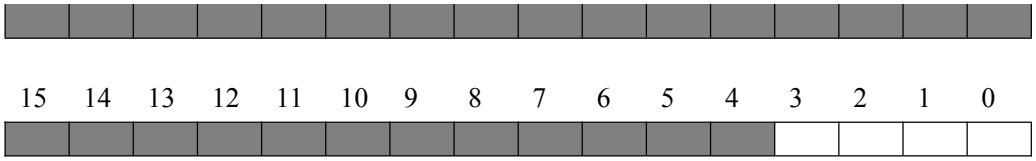
Bit	Type	Reset	Description
31-5	R	0	Not used.
4	R/W	0	**TIMER_0N100PWM_EN** Allows user to enable or disable the usage of Timer 0% and 100% mode feature. **Values:** 0x1 (ENABLED): Timer 0% and 100% PWM duty cycle mode is enabled 0x0 (DISABLE): Timer 0% and 100% PWM duty cycle mode is disabled
3	R/W	0	**TIMER_PWM** Pulse Width Modulation of timer_1_toggle output. **Values:** 0x1 (ENABLED): PWM for timer_1_toggle o/p is enabled 0x0 (DISABLE): PWM for timer_1_toggle o/p is disabled
2	R/W	0	**TIMER_INTERRUPT_MASK** Timer interrupt mask for Timer 1. **Values:** 0x1 (MASKED): Timer 1 interrupt is masked 0x0 (UNMASKED): Timer 1 interrupt is unmasked
1	R/W	0	**TIMER_MODE** Timer mode for Timer 1. **Values:** 0x1 (USER_DEFINED): User-Defined mode of operation 0x0 (FREE_RUNNING): Free Running mode of operation
0	R/W	0	**TIMER_ENABLE** Timer enable bit for Timer 1. **Values:** 0x1 (ENABLED): Timer 1 is enabled 0x0 (DISABLE): Timer 1 is disabled

18.2.4Timer1EOI : Timer 1 End-of-Interrupt Register

Clears the interrupt from Timer 1

Address: TIMER_REG_BASE + 0x0c



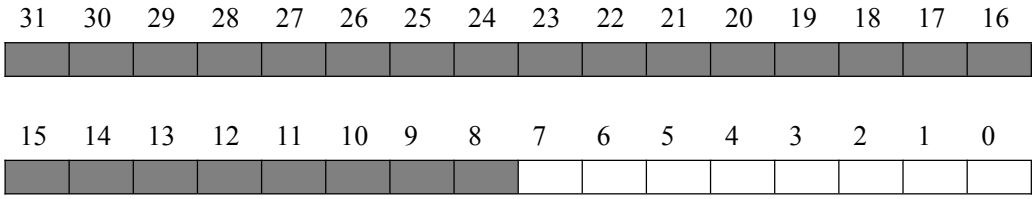


Bit	Type	Reset	Description
31-1	R	0	Not used.
0	R	0	Reading from this register returns all zeroes (0) and clears the interrupt from Timer 1.

18.2.5Timer1IntStatus : Timer 1 Interrupt Status Register

Contains the interrupt status for Timer 1

Address: TIMER_REG_BASE + 0x10

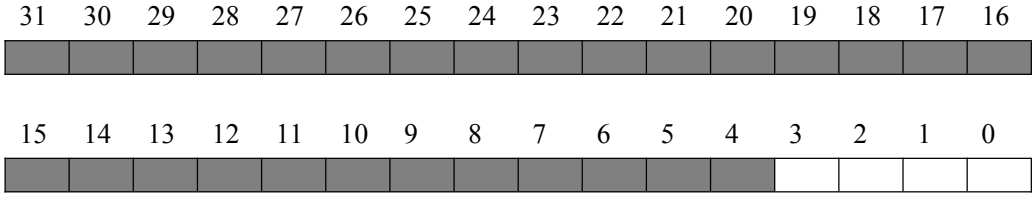


Bit	Type	Reset	Description
31-1	R	0	Not used.
0	R	0	Contains the interrupt status for Timer 1. **Values:** 0x1 (ACTIVE): Timer 1 Interrupt is active 0x0 (INACTIVE): Timer 1 Interrupt is inactive

18.2.6 Timer2LoadCount : Timer2 Load Count Register

Value to be loaded into Timer2

Address: TIMER_REG_BASE + 0x14

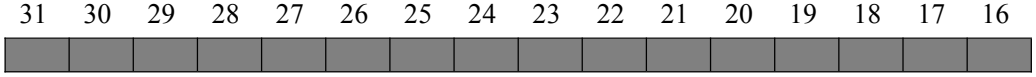


Bit	Type	Reset	Description
31-0	R/W	0	Value to be loaded into Timer 2. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.

18.2.7 Timer2CurrentValue

Current value of Timer 2

Address: TIMER_REG_BASE + 0x18

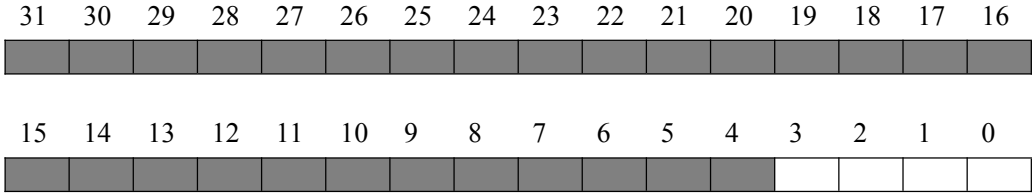


Bit	Type	Reset	Description
31-0	R	0	Current Value of Timer 2. When TIM_NEWMODE=0, This register is supported only when timer_2_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. When TIM_NEWMODE=1, no restrictions apply.

18.2.8Timer2ControlReg: Timer 2 Control Register

Control Register for Timer 2. This register controls enabling, operating mode (free-running or defined-count), and interrupt mask of Timer 2. You can program each Timer2ControlReg to enable or disable a specific timer and to control its mode of operation.

Address: TIMER_REG_BASE + 0x1C

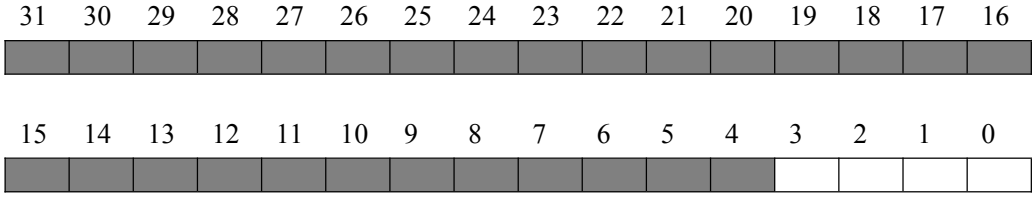


Bit	Type	Reset	Description
31-5	R	0	Not used.
4	R/W	0	**TIMER_0N100PWM_EN** Allows user to enable or disable the usage of Timer 0% and 100% mode feature. **Values:** 0x1 (ENABLED): Timer 0% and 100% PWM duty cycle mode is enabled 0x0 (DISABLE): Timer 0% and 100% PWM duty cycle mode is disabled
3	R/W	0	**TIMER_PWM** Pulse Width Modulation of timer_2_toggle output. **Values:** 0x1 (ENABLED): PWM for timer_2_toggle o/p is enabled 0x0 (DISABLE): PWM for timer_2_toggle o/p is disabled
2	R/W	0	**TIMER_INTERRUPT_MASK** Timer interrupt mask for Timer 2. **Values:** 0x1 (MASKED): Timer 2 interrupt is masked 0x0 (UNMASKED): Timer 2 interrupt is unmasked
1	R/W	0	**TIMER_MODE** Timer mode for Timer 2. **Values:** 0x1 (USER_DEFINED): User-Defined mode of operation 0x0 (FREE_RUNNING): Free Running mode of operation
0	R/W	0	**TIMER_ENABLE** Timer enable bit for Timer 2. **Values:** 0x1 (ENABLED): Timer 2 is enabled 0x0 (DISABLE): Timer 2 is disabled

18.2.9Timer2EOI : Timer 2 End-of-Interrupt Register

Clears the interrupt from Timer 2

Address: TIMER_REG_BASE + 0x20

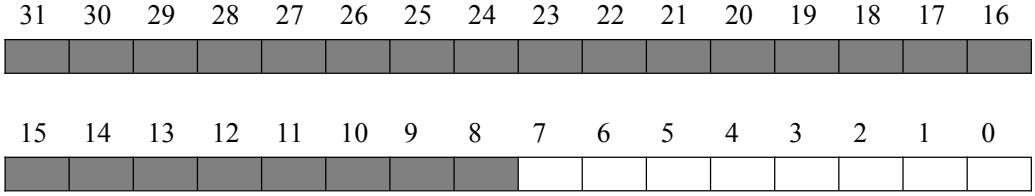


Bit	Type	Reset	Description
31-1	R	0	Not used.
0	R	0	Reading from this register returns all zeroes (0) and clears the interrupt from Timer 2.

18.2.10Timer2IntStatus : Timer 2 Interrupt Status Register

Contains the interrupt status for Timer 2

Address: TIMER_REG_BASE + 0x24

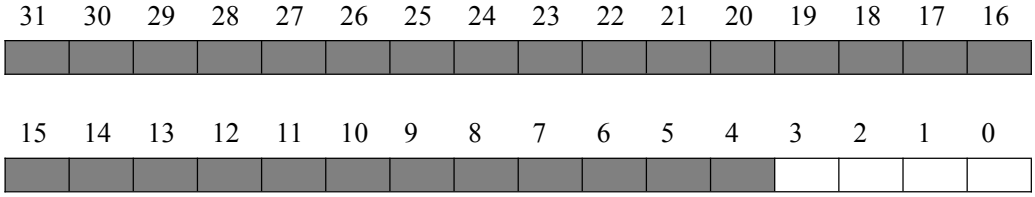


Bit	Type	Reset	Description
31-1	R	0	Not used.
0	R	0	Contains the interrupt status for Timer 2. **Values:** 0x1 (ACTIVE): Timer 2 Interrupt is active 0x0 (INACTIVE): Timer 2 Interrupt is inactive

18.2.11 Timer3LoadCount : Timer3 Load Count Register

Value to be loaded into Timer3

Address: TIMER_REG_BASE + 0x28

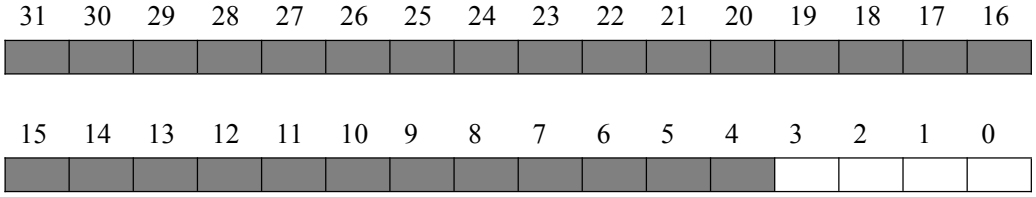


Bit	Type	Reset	Description
31-0	R/W	0	Value to be loaded into Timer 3. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.

18.2.12 Timer3CurrentValue

Current value of Timer 3

Address: TIMER_REG_BASE + 0x2C

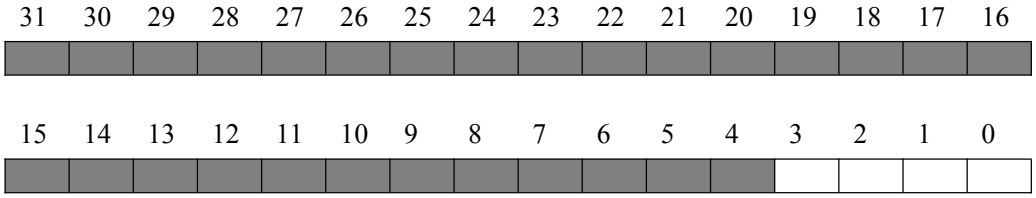


Bit	Type	Reset	Description
31-0	R	0	Current Value of Timer 3. When TIM_NEWMODE=0, This register is supported only when timer_3_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. When TIM_NEWMODE=1, no restrictions apply.

18.2.13Timer3ControlReg: Timer 3 Control Register

Control Register for Timer 3. This register controls enabling, operating mode (free-running or defined-count), and interrupt mask of Timer 3. You can program each Timer3ControlReg to enable or disable a specific timer and to control its mode of operation.

Address: TIMER_REG_BASE + 0x30



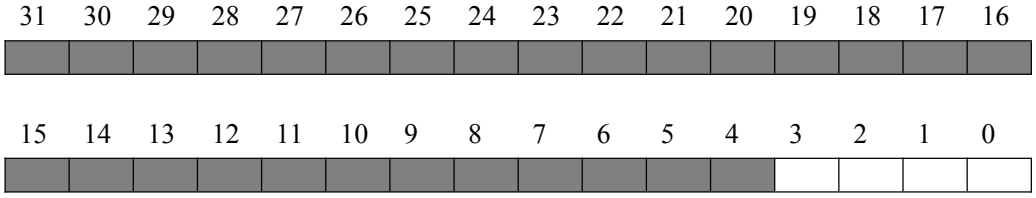
Bit	Type	Reset	Description
31-5	R	0	Not used.
4	R/W	0	**TIMER_0N100PWM_EN** Allows user to enable or disable the usage of Timer 0% and 100% mode feature. **Values:** 0x1 (ENABLED): Timer 0% and 100% PWM duty cycle mode is enabled 0x0 (DISABLE): Timer 0% and 100% PWM duty cycle mode is disabled
3	R/W	0	**TIMER_PWM** Pulse Width Modulation of timer_3_toggle output. **Values:** 0x1 (ENABLED): PWM for timer_3_toggle o/p is enabled 0x0 (DISABLE): PWM for timer_3_toggle o/p is disabled
2	R/W	0	**TIMER_INTERRUPT_MASK** Timer interrupt mask for Timer 3. **Values:** 0x1 (MASKED): Timer 3 interrupt is masked 0x0 (UNMASKED): Timer 3 interrupt is unmasked
1	R/W	0	**TIMER_MODE** Timer mode for Timer 3. **Values:** 0x1 (USER_DEFINED): User-Defined mode of operation 0x0 (FREE_RUNNING): Free Running mode of operation
0	R/W	0	**TIMER_ENABLE**

			Timer enable bit for Timer 3. **Values:** 0x1 (ENABLED): Timer 3 is enabled 0x0 (DISABLE): Timer 3 is disabled

18.2.14Timer3EOI : Timer 3 End-of-Interrupt Register

Clears the interrupt from Timer 3

Address: TIMER_REG_BASE + 0x34

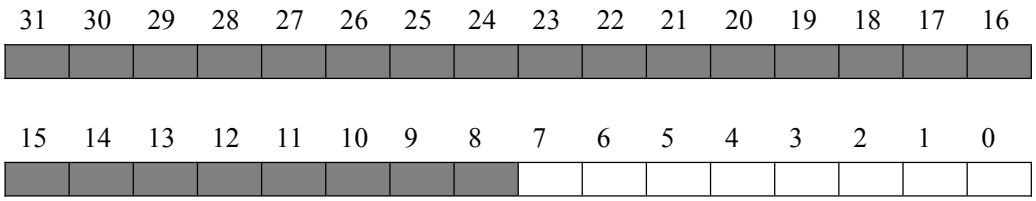


Bit	Type	Reset	Description
31-1	R	0	Not used.
0	R	0	Reading from this register returns all zeroes (0) and clears the interrupt from Timer 3.

18.2.15Timer3IntStatus : Timer 3 Interrupt Status Register

Contains the interrupt status for Timer 3

Address: TIMER_REG_BASE + 0x38

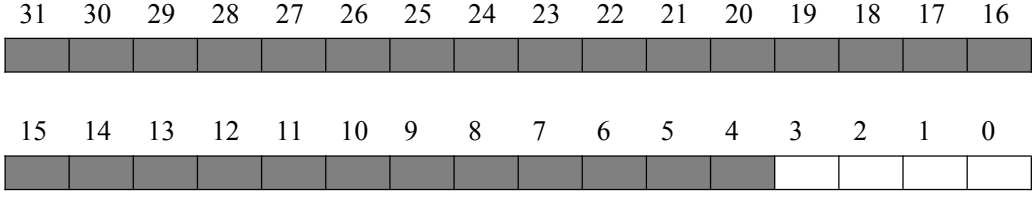


Bit	Type	Reset	Description
31-1	R	0	Not used.
0	R	0	Contains the interrupt status for Timer 3. **Values:** 0x1 (ACTIVE): Timer 3 Interrupt is active 0x0 (INACTIVE): Timer 3 Interrupt is inactive

18.2.16 Timer4LoadCount : Timer4 Load Count Register

Value to be loaded into Timer4

Address: TIMER_REG_BASE + 0x3C



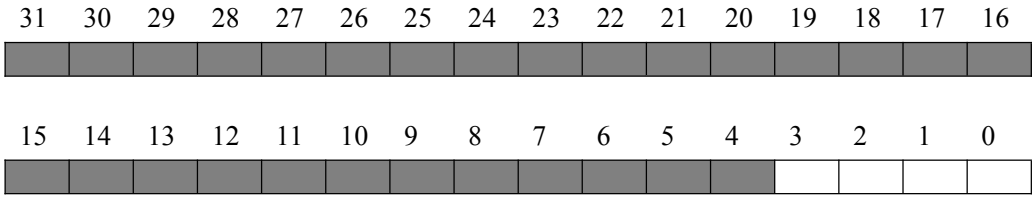
Bit	Type	Reset	Description

31-0	R/W	0	Value to be loaded into Timer 4. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.

18.2.17 Timer4CurrentValue

Current value of Timer 4

Address: TIMER_REG_BASE + 0x40

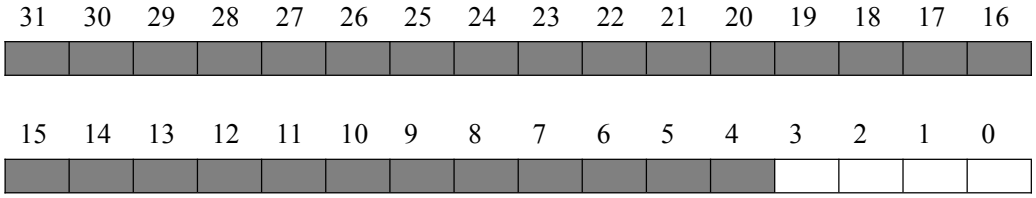


Bit	Type	Reset	Description
31-0	R	0	Current Value of Timer 4. When TIM_NEWMODE=0, This register is supported only when timer_4_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. When TIM_NEWMODE=1, no restrictions apply.

18.2.18 Timer4ControlReg: Timer 4 Control Register

Control Register for Timer 4. This register controls enabling, operating mode (free-running or defined-count), and interrupt mask of Timer 4. You can program each Timer3ControlReg to enable or disable a specific timer and to control its mode of operation.

Address: TIMER_REG_BASE + 0x44



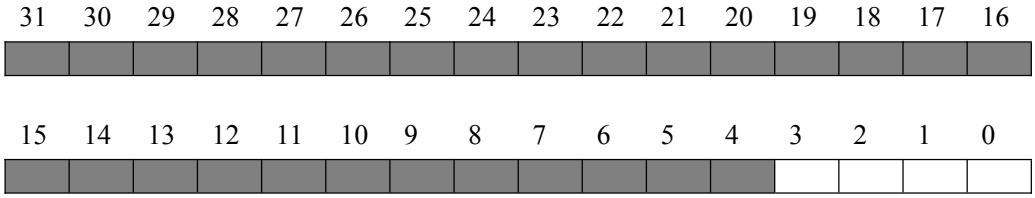
Bit	Type	Reset	Description
31-5	R	0	Not used.
4	R/W	0	**TIMER_0N100PWM_EN** Allows user to enable or disable the usage of Timer 0% and 100% mode feature. **Values:** 0x1 (ENABLED): Timer 0% and 100% PWM duty cycle mode is enabled 0x0 (DISABLE): Timer 0% and 100% PWM duty cycle mode is disabled
3	R/W	0	**TIMER_PWM** Pulse Width Modulation of timer_4_toggle output. **Values:** 0x1 (ENABLED): PWM for timer_4_toggle o/p is enabled 0x0 (DISABLE): PWM for timer_4_toggle o/p is disabled
2	R/W	0	**TIMER_INTERRUPT_MASK** Timer interrupt mask for Timer 4. **Values:** 0x1 (MASKED): Timer 4 interrupt is masked

			0x0 (UNMASKED): Timer 4 interrupt is unmasked
1	R/W	0	**TIMER_MODE** Timer mode for Timer 4. **Values:** 0x1 (USER_DEFINED): User-Defined mode of operation 0x0 (FREE_RUNNING): Free Running mode of operation
0	R/W	0	**TIMER_ENABLE** Timer enable bit for Timer 4. **Values:** 0x1 (ENABLED): Timer 4 is enabled 0x0 (DISABLE): Timer 4 is disabled

18.2.19Timer4EOI : Timer 4 End-of-Interrupt Register

Clears the interrupt from Timer 4

Address: TIMER_REG_BASE + 0x48

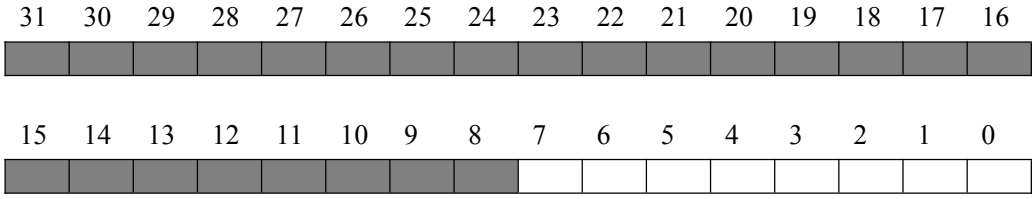


Bit	Type	Reset	Description
31-1	R	0	Not used.
0	R	0	Reading from this register returns all zeroes (0) and clears the interrupt from Timer 4.

18.2.20Timer4IntStatus : Timer 4 Interrupt Status Register

Contains the interrupt status for Timer 4

Address: TIMER_REG_BASE + 0x4C

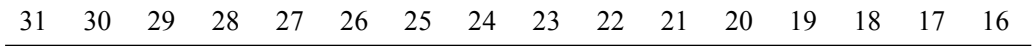


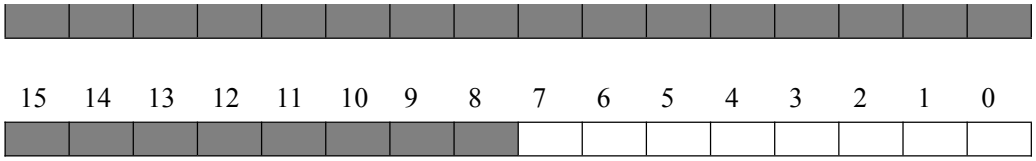
Bit	Type	Reset	Description
31-1	R	0	Not used.
0	R	0	Contains the interrupt status for Timer 4. **Values:** 0x1 (ACTIVE): Timer 4 Interrupt is active 0x0 (INACTIVE): Timer 4 Interrupt is inactive

18.2.21TimersIntStatus : Timers Interrupt Status Register

Contains the interrupt status of all timers in the component.

Address: TIMER_REG_BASE + 0xA0



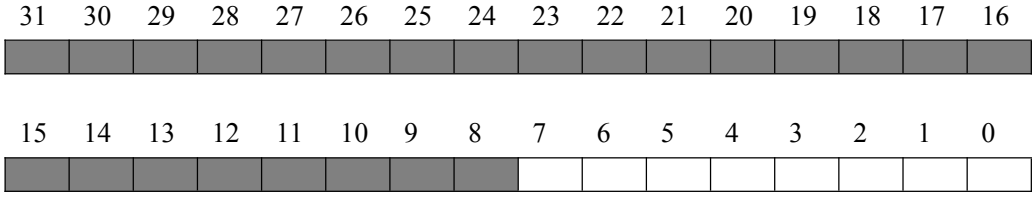


Bit	Type	Reset	Description
31-4	R	0	Not used.
3-0	R	0	Contains the interrupt status of all timers in the component. If a bit of this register is 0, then the corresponding timer interrupt is not active and the corresponding interrupt could be on either the timer_intr bus or the timer_intr_n bus, depending on the interrupt polarity you have chosen. Similarly, if a bit of this register is 1, then the corresponding interrupt bit has been set in the relevant interrupt bus. In both cases, the status reported is the status after the interrupt mask has been applied. Reading from this register does not clear any active interrupts. **Values:** 0x1 (ACTIVE): Timer_intr(_n) is active 0x0 (INACTIVE): Timer_intr(_n) is inactive

18.2.22TimersEOI : Timers End-of-Interrupt Register

Returns all zeroes (0) and clears all active interrupts

Address: TIMER_REG_BASE + 0xA4

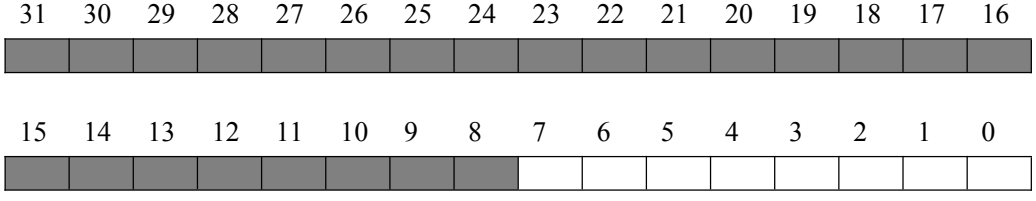


Bit	Type	Reset	Description
31-4	R	0	Not used.
3-0	R	0	Reading this register returns all zeroes (0) and clears all active interrupts.

18.2.23TimersRawIntStatus : Timers Raw Interrupt Status Register

Contains the unmasked interrupt status of all timers in the component.

Address: TIMER_REG_BASE + 0xA8



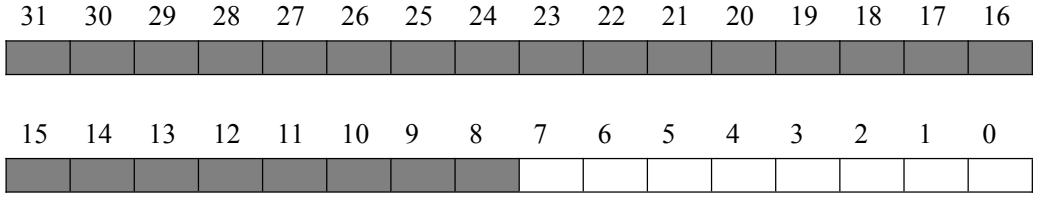
Bit	Type	Reset	Description
31-4	R	0	Not used.
3-0	R	0	The register contains the unmasked interrupt status of all timers in the component.

			Values: 0x1 (ACTIVE): Raw Timer_intr(_n) is active 0x0 (INACTIVE): Raw Timer_intr(_n) is inactive

18.2.24Timer1LoadCount2: Timer 1 Load Count2 Register

Value to be loaded into Timer 1 when toggle output changes from 0 to 1

Address: TIMER_REG_BASE + 0xB0

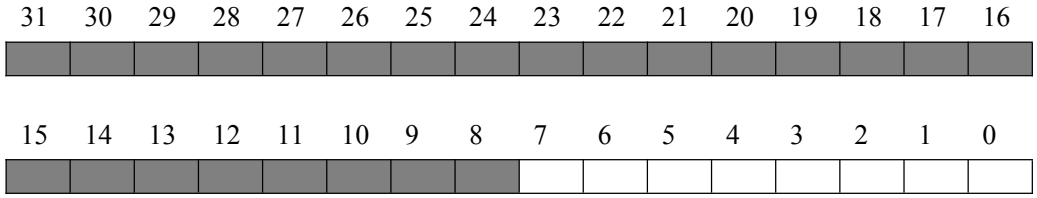


Bit	Type	Reset	Description
31-0	R/W	0	Value to be loaded into Timer 1 when timer_1_toggle output changes from 0 to 1. This value determines the width of the HIGH period of the timer_1_toggle output.

18.2.25Timer2LoadCount2: Timer 2 Load Count2 Register

Value to be loaded into Timer 2 when toggle output changes from 0 to 1

Address: TIMER_REG_BASE + 0xB4

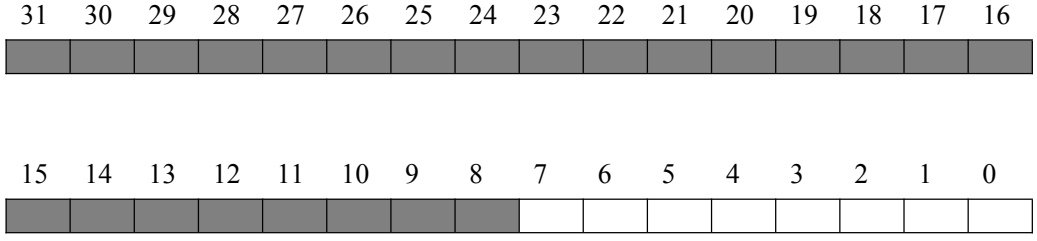


Bit	Type	Reset	Description
31-0	R/W	0	Value to be loaded into Timer 2 when timer_2_toggle output changes from 0 to 1. This value determines the width of the HIGH period of the timer_2_toggle output.

18.2.26Timer3LoadCount2: Timer 3 Load Count2 Register

Value to be loaded into Timer 3 when toggle output changes from 0 to 1

Address: TIMER_REG_BASE + 0xB8

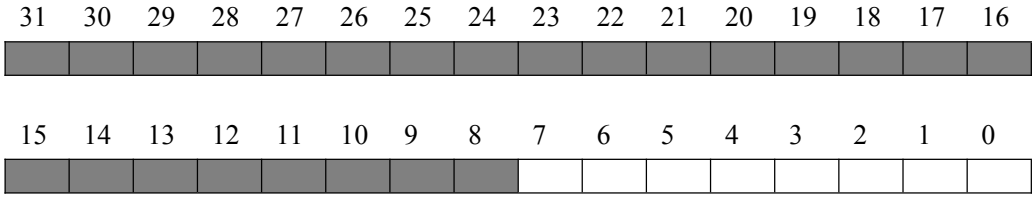


Bit	Type	Reset	Description
31-0	R/W	0	Value to be loaded into Timer 3 when timer_3_toggle output changes from 0 to 1. This value determines the width of the HIGH period of the timer_3_toggle output.

18.2.27 Timer4LoadCount2: Timer 4 Load Count2 Register

Value to be loaded into Timer 4 when toggle output changes from 0 to 1

Address: TIMER_REG_BASE + 0xBC



Bit	Type	Reset	Description
31-0	R/W	0	Value to be loaded into Timer 4 when timer_4_toggle output changes from 0 to 1. This value determines the width of the HIGH period of the timer_4_toggle output.

19. SDHC

19.1SDMMC Features

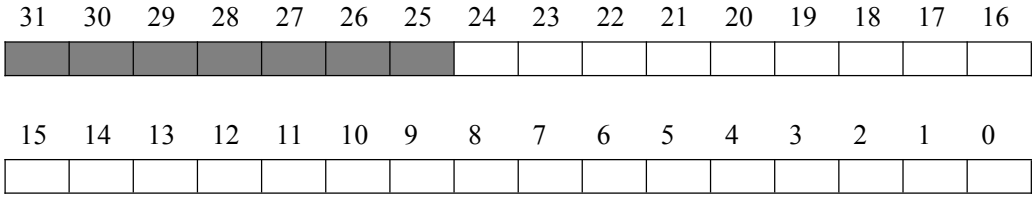
Fully compatible with the MMC version 5.0
Fully compatible with the SD version 3.0
Fully compatible with the SDIO version 3.0
Support 1/4-bit mode SD/SDIO
Support 1/4-bit mode MMC

19.2SDMMC Register Descriptions

SDHC0_BASE: 0x4010_0000
SDHC1_BASE: 0x4040_0000

19.2.1CTRL: Control Register

Address: SDHC_BASE + 0x00



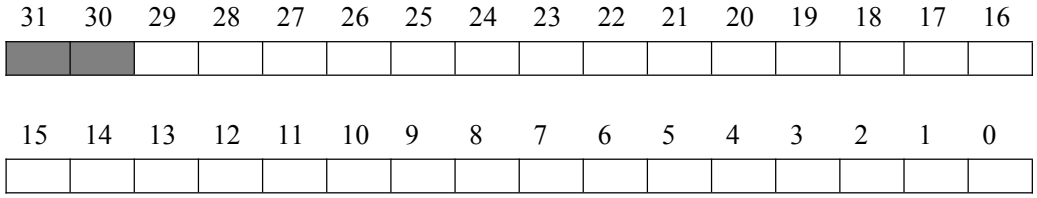
Bit	Type	Reset	Description
31-26	R	0	Not Used
25	R/W	0	**use_internal_dmac** Present only for the Internal DMAC configuration; else, it is reserved. 0 – The host performs data transfers through the slave interface 1 – Internal DMAC used for data transfer
24	R/W	1	**enable_OD_pullup** External open-drain pullup: 0 – Disable 1 – Enable Inverted value of this bit is output to ccmd_od_pullup_en_n port. When bit is set, command output always driven in open-drive mode; that is, SDMMC drives either 0 or high impedance, and does not drive hard 1. If you are using SD card, please disable open drain while you reset this controller before initial sd card. On the contrary, if the car is

			MMC, please disable this bit after initialize the card.
23-20	R/W	0	Reserved
19-16	R/W	0	Reserved
15-12	R/W	0	Not Used
11	R/W	0	**ceata_device_interrupt_status** 0 – Interrupts not enabled in CE-ATA device (nIEN = 1 in ATA control register) 1 –Interrupts are enabled in CE-ATA device (nIEN = 0 in ATA control register) Software should appropriately write to this bit after power-on reset or any other reset to CE-ATA device. After reset, usually CE-ATA device interrupt is disabled (nIEN = 1). If the host enables CE-ATA device interrupt, then software should set this bit.
10	R/W	0	**send_auto_stop_ccsd** 0 – Clear bit if SDMMC does not reset the bit. 1 –Send internally generated STOP after sending CCSD to CE-ATA device. When set, SDMMC automatically sends internally-generated STOP command (CMD12) to CE-ATA device. After sending internally-generated STOP command, Auto Command Done (ACD) bit in RINTSTS is set and generates interrupt to host if Auto Command Done interrupt is not masked. After sending the CCSD, SDMMC automatically clears send_auto_stop_ccsd bit.
9	R/W	0	**send_ccsd** 0 – Clear bit if SDMMC does not reset the bit. 1 –Send Command Completion Signal Disable (CCSD) to CE-ATA device When set, SDMMC sends CCSD to CE-ATA device. Software sets this bit only if current command is expecting CCS (that is, RW_BLK) and interrupts are enabled in CE-ATA device. Once the CCSD pattern is sent to device, SDMMC automatically clears send_ccsd bit. It also sets Command Done (CD) bit in RINTSTS register and generates interrupt to host if Command Done interrupt is not masked.
8	R/W	0	**abort_read_data** 0-- not changed 1 –After suspend command is issued during read-transfer, software polls card to find when suspend happened. Once suspend occurs, software sets bit to reset data state-machine, which is waiting for next block of data. Bit automatically clears once data state-machine resets to idle. Used in SDIO card suspends sequence.
7	R/W	0	**send_irq_response** 0 – No change 1 –Send auto IRQ response Bit automatically clears once response is sent. To wait for MMC card interrupts, host issues CMD40, and SDMMC waits for interrupt response from MMC card(s). In meantime, if host wants SDMMC to exit waiting for interrupt state, it can set this bit, at which time SDMMC command state-machine sends CMD40 response on bus and returns to idle state.
6	R/W	0	**read_wait** 0 – Clear read wait 1 – Assert read wait For sending read-wait to SDIO cards.
5	R/W	0	**dma_enable** 0 – Disable DMA transfer mode

			1 –Enable DMA transfer mode Even when DMA mode is enabled, host can still push/pop data into or from FIFO; this should not happen during the normal operation. If there is simultaneous FIFO access from host/DMA, the data coherency is lost. Also, there is no arbitration inside SDMMC to prioritize simultaneous host/DMA access.
4		0	**int_enable** Global interrupt enable/disable bit: 0 – Disable interrupts 1 – Enable interrupts The int port is 1 only when this bit is 1 and one or more unmasked interrupts are set.
3		0	Not Used
2		0	**dma_reset** 0 – No change 1 – Reset internal DMA interface control logic To reset DMA interface, firmware should set bit to 1. This bit is auto-cleared
1	R/W	0	**fifo_reset** 0 – No change 1 –Reset to data FIFO To reset FIFO pointers To reset FIFO, firmware should set bit to 1. This bit is auto-cleared after completion of reset operation.
0	R/W	0	**controller_reset** 0 – No change 1 –Reset SDMMC controller To reset controller, firmware should set bit to 1. This bit is auto-cleared.

19.2.2PWREN: Power Enable Register

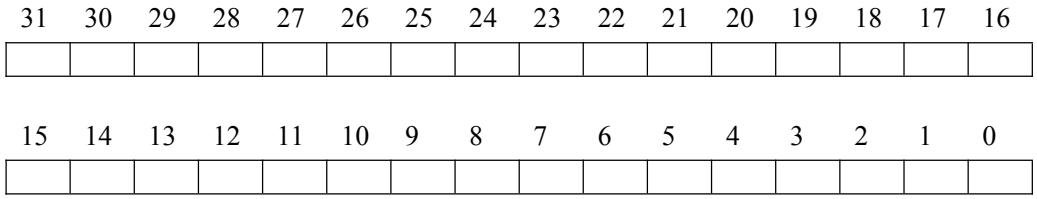
Address: SDHC_BASE + 0x04



Bit	Type	Reset	Description
31-30	R/W	0	Not Used
0	R/W	0	**power_enable** 0 – power off 1 –power on Power on/off switch for cards; for example, Once power is turned on, firmware should wait for regulator/switch ramp-up time before trying to initialize card.

19.2.3CLKDIV: Clock Divider Register

Address: SDHC_BASE + 0x08



Bit	Type	Reset	Description
31-24	R/W	0	**clk_divider3** Clock divider-3 value. Clock division is 2*n. For example, value of 0 means divide by 2*0 = 0 (no division, bypass), a value of 1 means divide by 2*1 = 2, a value of "ff" means divide by 2*255 = 510, and so on.
23-16	R/W	0	**clk_divider2** Clock divider-2 value. Clock division is 2*n. For example, value of 0 means divide by 2*0 = 0 (no division, bypass), value of 1 means divide by 2*1 = 2, value of "ff" means divide by 2*255 = 510, and so on.
15-8	R/W	0	**clk_divider1** Clock divider-1 value. Clock division is 2*n. For example, value of 0 means divide by 2*0 = 0 (no division, bypass), value of 1 means divide by 2*1 = 2, value of "ff" means divide by 2*255 = 510, and so on.
7-0	R/W	0	**clk_divider0** Clock divider-0 value. Clock division is 2*n. For example, value of 0 means divide by 2*0 = 0 (no division, bypass), value of 1 means divide by 2*1 = 2, value of "ff" means divide by 2*255 = 510, and so on.

19.2.4CLKSRC: SD Clock Source Register

Address: SDHC_BASE + 0x0c

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

Bit	Type	Reset	Description
31-2	R/W	0	Not used
1-0	R/W	0	**clk_source** Clock divider source .

19.2.5CLKENA: Clock Enable Register

Address: SDHC_BASE + 0x10

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

Bit	Type	Reset	Description
31-16	R/W	0	**cclk_low_power** Low-power control SD/MMC card clocks 0 – Non-low-power mode 1 – Low-power mode; stop clock when card in IDLE (should be normally set to only MMC and SD memory cards; for SDIO

			cards, if interrupts must be detected, clock should not be stopped).
15-0	R/W	0	**cclk_enable** Clock-enable control SD/MMC card clocks 0 – Clock disabled 1 – Clock enabled

19.2.6TMOUT: Timeout Register

Address: SDHC_BASE + 0x14

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

Bit	Type	Reset	Description
31-8	R/W	0xffffff	**data_timeout** Value for card Data Read Timeout; same value also used for Data Starvation by Host timeout. Value is in number of card output clocks – sdmmc clock out of selected card.
7-0	R/W	0x40	**response_timeout** Response timeout value. Value is in number of card output clocks – sdmmc clock out

19.2.7CTYPE: Card Type Register

Address: SDHC_BASE + 0x18

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

Bit	Type	Reset	Description
31-17	R/W	0	Reserved
16	R/W	0	**card_width** One bit per card indicates if card is 8-bit: 0 – Non 8-bit mode

			1 – 8-bit mode Bit[31] corresponds to card[15]; bit[16] corresponds to card[0].
15-1	R/W	0	
0	R/W	0	**card_width** One bit per card indicates if card is 1-bit or 4-bit: 0 – 1-bit mode 1 – 4-bit mode

19.2.8BLKSIZ: Block Size Register

Address: SDHC_BASE + 0x1c

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

Bit	Type	Reset	Description
31-16	R	0	Not Used
15-0	R/W	512	**block_size** Block size

19.2.9BYTCNT: Byte Count Register

Address: SDHC_BASE + 0x20

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

Bit	Type	Reset	Description
31-0	R/W	512	**byte_count** Number of bytes to be transferred; should be integer multiple of Block Size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is responsibility of host to explicitly send stop/abort command to terminate data transfer.

19.2.10INTMASK: Interrupt Mask Register

Address: SDHC_BASE + 0x24

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

Bit	Type	Reset	Description
31-16	R/W	0	**sdio_int_mask**

			Mask SDIO interrupts. Number of bytes to be transferred; should be integer multiple of Block Size for block transfers. For undefined number of byte transfers, byte count should be set to 0. When byte count is set to 0, it is responsibility of host to explicitly send stop/abort command to terminate data transfer.
15-0	R/W	0	**int_mask** Bits used to mask unwanted interrupts. Value of 0 masks interrupt; value of 1 enables interrupt. bit 15 – End-bit error (read)/Write no CRC (EBE) bit 14 – Auto command done (ACD) bit 13 – Start-bit error (SBE) bit 12 – Hardware locked write error (HLE) bit 11 – FIFO underrun/overrun error (FRUN) bit 10 – Data starvation-by-host timeout (HTO) bit 9 – Data read timeout (DRTO) bit 8 – Response timeout (RTO) bit 7 – Data CRC error (DCRC) bit 6 – Response CRC error (RCRC) bit 5 – Receive FIFO data request (RXDR) bit 4 – Transmit FIFO data request (TXDR) bit 3 – Data transfer over (DTO) bit 2 – Command done (CD) bit 1 – Response error (RE) bit 0 – Card detect (CD)

19.2.11CMDARG: Command Argument Register

Address: SDHC_BASE + 0x28

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

Bit	Type	Reset	Description
31-0	R/W	0	**cmd_arg** Value indicates command argument to be passed to card.

19.2.12CMD: Command Register

Address: SDHC_BASE + 0x2c

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

Bit	Type	Reset	Description
31	R/W	0	**start_cmd** Start command. It is auto-cleared. When bit is set, host should not attempt to write to any command Register. If write is attempted, hardware lock error is set in raw interrupt register. Once command is sent and response is received from SD_MMC/CE-ATA cards, Command Done bit is set in raw interrupt register.

30-24	R	0	Not Used
23	R/W	0	**ccs_expected** 0 – Interrupts are not enabled in CE-ATA device (nIEN = 1 in ATA control register), or command does not expect CCS from device 1 –Interrupts are enabled in CE-ATA device (nIEN = 0), and RW_BLK command expects command completion signal from CE-ATA device If the command expects Command Completion Signal (CCS) from the CE-ATA device, the software should set this control bit. SDMMC sets Data Transfer Over (DTO) bit in RINTSTS register and generates interrupt to host if Data Transfer Over interrupt is not masked.
22	R/W	0	**read_ceata_device** 0 – Host is not performing read access (RW_REG or RW_BLK) towards CE-ATA device 1 –Host is performing read access (RW_REG or RW_BLK) towards CE-ATA device Software should set this bit to indicate that CE-ATA device is being accessed for read transfer. This bit is used to disable read data timeout indication while performing CE-ATA read transfers. Maximum value of I/O transmission delay can be no less than 10 seconds. SDMMC should not indicate read data timeout while waiting for data from CE-ATA device.
21	R/W	0	**update_clock_Register_only** 0 – Normal command sequence 1 – Do not send commands, just update clock register value into card clock domain Following register values transferred into card clock domain: CLKDIV, CLRSRC, CLKENA. Changes card clocks (change frequency, truncate off or on, and set low-frequency mode); provided in order to change clock frequency or stop clock without having to send command to cards. During normal command sequence, when update_clock_Register_only = 0, following control Register are transferred from BIU to CIU: CMD, CMDARG, TMOUT, CTYPE, BLKSIZ, BYTCNT. CIU uses new register values for new command sequence to card(s). When bit is set, there are no Command Done interrupts because no command is sent to SD_MMC/CE-ATA cards.
20-16	R/W	0	**card_number** Card number in use. Represents physical slot number of card being accessed.
15	R/W	0	**send_initialization** 0 – Do not send initialization sequence (80 clocks of 1) before sending this command 1 – Send initialization sequence before sending this command After power on, 80 clocks must be sent to card for initialization before sending any commands to card. Bit should be set while sending first command to card so that controller will initialize clocks before sending command to card.
14	R/W	0	**stop_abort_cmd** 0 – Neither stop nor abort command to stop current data transfer in progress. If abort is sent to function-number currently selected or not in data-transfer mode, then bit should be set to 0. 1 –Stop or abort command intended to stop current data transfer in progress. When open-ended or predefined data transfer is in progress, and host issues stop or abort command to stop data transfer, bit should

			be set so that command/data state-machines of CIU can return correctly to idle state.
13	R/W	0	**wait_prvdata_complete** 0 – Send command at once, even if previous data transfer has not completed 1 –Wait for previous data transfer completion before sending command The wait_prvdata_complete = 0 option typically used to query status of card during data transfer or to stop current data transfer; card_number should be same as in previous command.
12	R/W	0	**send_auto_stop** 0 – No stop command sent at end of data transfer 1 – Send stop command at end of data transfer When set, SDMMC sends stop command to SD_MMC/CE-ATA cards at end of data transfer. when send_auto_stop bit should be set, since some data transfers do not need explicit stop commands open-ended transfers that software should explicitly send to stop command . Additionally, when “resume” is sent to resume – suspended memory access of SD-Combo card – bit should be set correctly if suspended data transfer needs send_auto_stop. Don’t care if no data expected from card.
11	R/W	0	**transfer_mode** 0 – Block data transfer command 1 – Stream data transfer command Don’t care if no data expected.
10	R/W	0	**read/write** 0 – Read from card 1 – Write to card Don’t care if no data expected from card.
9	R/W	0	**data_expected** 0 – No data transfer expected (read/write) 1 – Data transfer expected (read/write)
8	R/W	0	**check_response_crc** 0 – Do not check response CRC 1 – Check response CRC Some of command responses do not return valid CRC bits. Software should disable CRC checks for those commands in order to disable CRC checking by controller.
7	R/W	0	**response_ length** 0 – Short response expected from card 1 – Long response expected from card
6	R/W	0	**response_expect** 0 – No response expected from card 1 – Response expected from card
5-0	R/W	0	**cmd_index** Command index

19.2.13RESP0: Response Register 0

Address: SDHC_BASE + 0x30

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

Bit	Type	Reset	Description
31-0	R	0	**response 0** Bit[31:0] of response

19.2.14RESP1: Response Register 1

Address: SDHC_BASE + 0x34

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

Bit	Type	Reset	Description
31-0	R	0	**response 1** Bit[31:0] of response

19.2.15RESP2: Response Register 2

Address: SDHC_BASE + 0x38

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

Bit	Type	Reset	Description
31-0	R	0	**response 2** Bit[31:0] of response

19.2.16RESP3: Response Register 3

Address: SDHC_BASE + 0x3c

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

Bit	Type	Reset	Description
31-0	R	0	**response 3** Bit[31:0] of response

19.2.17MINTSTS: Masked Interrupt Status Register

Address: SDHC_BASE + 0x40

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

Bit	Type	Reset	Description
31-16	R	0	**sdio_interrupt** Interrupt from SDIO card; one bit for each card. Bit[31] corresponds to Card[15], and bit[16] is for Card[0]. SDIO interrupt for card enabled only if corresponding sdio_int_mask bit is set in Interrupt mask register (mask bit 1 enables interrupt; 0 masks interrupt). 0 – No SDIO interrupt from card 1 –SDIO interrupt from card
15-0	R	0	**int_status** Interrupt enabled only if corresponding bit in interrupt mask register is set. bit 15 – End-bit error (read)/write no CRC (EBE) bit 14 – Auto command done (ACD) bit 13 – Start-bit error (SBE) bit 12 – Hardware locked write error (HLE) bit 11 – FIFO underrun/overrun error (FRUN) bit 10 – Data starvation by host timeout (HTO) bit 9 – Data read timeout (DRTO) bit 8 – Response timeout (RTO) bit 7 – Data CRC error (DCRC) bit 6 – Response CRC error (RCRC) bit 5 – Receive FIFO data request (RXDR) bit 4 – Transmit FIFO data request (TXDR) bit 3 – Data transfer over (DTO) bit 2 – Command done (CD) bit 1 – Response error (RE) bit 0 – Card detect (CD)

19.2.18RINTSTS: Raw Interrupt Status Register

Address: SDHC_BASE + 0x44

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

Bit	Type	Reset	Description
31-16	R/W	0	**sdio_interrupt** Interrupt from SDIO card; one bit for each card. Bit[31] corresponds to Card[15], and bit[16] is for Card[0]. Writes to these bits clear them. Value of 1 clears bit and 0 leaves bit intact. 0 – No SDIO interrupt from card 1 –SDIO interrupt from card Bits are logged regardless of interrupt-mask status.
15-0	R/W	0	**int_status** Writes to bits clear status bit. Value of 1 clears status bit, and value of 0 leaves bit intact. Bits are logged regardless of interrupt mask status. bit 15 – End-bit error (read)/write no CRC (EBE)

			bit 14 – Auto command done (ACD) bit 13 – Start-bit error (SBE) bit 12 – Hardware locked write error (HLE) bit 11 – FIFO underrun/overrun error (FRUN) bit 10 – Data starvation-by-host timeout (HTO) bit 9 – Data read timeout (DRTO) bit 8 – Response timeout (RTO) bit 7 – Data CRC error (DCRC) bit 6 – Response CRC error (RCRC) bit 5 – Receive FIFO data request (RXDR) bit 4 – Transmit FIFO data request (TXDR) bit 3 – Data transfer over (DTO) bit 2 – Command done (CD) bit 1 – Response error (RE) bit 0 – Card detect (CD)

19.2.19STATUS: Status Register

Address: SDHC_BASE + 0x48

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

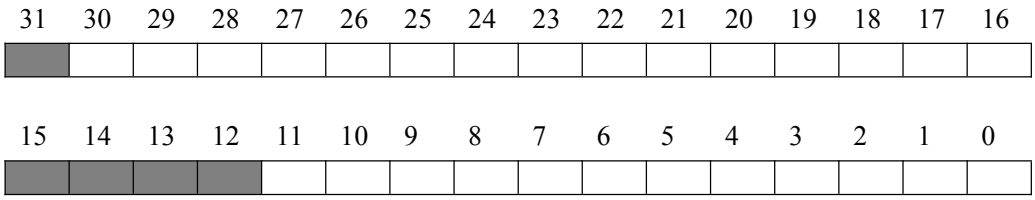
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31	R	0	**dma_req** DMA request signal state
30	R	0	**dma_ack** DMA acknowledge signal state
29-17	R	0	**fifo_count** FIFO count – Number of filled locations in FIFO
16-11	R	0	**response_index** Index of previous response, including any auto-stop sent by core
10	R	0	**data_state_mc_busy** Data transmit or receive state-machine is busy
9	R	1 or 0; depends on cdata_in	**data_busy** Inverted version of raw selected card_data[0] 0 – card data not busy 1 – card data busy
8	R	1 or 0; depends on cdata_in	**data_3_status** Raw selected card_data[3]; checks whether card is present 0 – card not present 1 – card present
7-4	R	0	**command fsm states** Command FSM states: 0 – Idle 1 – Send init sequence 2 – Tx cmd start bit 3 – Tx cmd tx bit

			4 – Tx cmd index + arg 5 – Tx cmd crc7 6 – Tx cmd end bit 7 – Rx resp start bit 8 – Rx resp IRQ response 9 – Rx resp tx bit 10 – Rx resp cmd idx 11 – Rx resp data 12 – Rx resp crc7 13 – Rx resp end bit 14 – Cmd path wait NCC 15 – Wait; CMD-to-response turnaround
3	R	0	**fifo_full** FIFO is full status
2	R	0	**fifo_empty** FIFO is empty status
1	R	0	**fifo_tx_watermark** FIFO reached Transmit watermark level; not qualified with data transfer.
0	R	0	**fifo_rx_watermark** FIFO reached Receive watermark level; not qualified with data transfer.

19.2.20FIFOTH: FIFO Threshold Watermark Register

Address: SDHC_BASE + 0x4c



Bit	Type	Reset	Description
31	R	0	Not Used
30-28	R/W	0	**DMA_Mutiple_Transaction_Size** Burst size of multiple transaction; should be programmed same as DMA controller multiple-transaction-size SRC/DEST_MSIZE. 000 – 1 transfers 001 – 4 010 – 8 011 – 16 100 – 32 101 – 64 110 – 128 111 – 256 Value should be sub-multiple of (RX_WMark + 1)* (F_DATA_WIDTH/H_DATA_WIDTH) and (FIFO_DEPTH - TX_WMark)* (F_DATA_WIDTH/ H_DATA_WIDTH) For example, if FIFO_DEPTH = 16, FDATA_WIDTH == H_DATA_WIDTH Allowed combinations for MSize and TX_WMark are: MSize = 1, TX_WMARK = 1-15 MSize = 4, TX_WMark = 8

			MSize = 4, TX_WMark = 4 MSize = 4, TX_WMark = 12 MSize = 8, TX_WMark = 8 MSize = 8, TX_WMark = 4 Allowed combinations for MSize and RX_WMark are: MSize = 1, RX_WMARK = 0-14 MSize = 4, RX_WMark = 3 MSize = 4, RX_WMark = 7 MSize = 4, RX_WMark = 11 MSize = 8, RX_WMark = 7 MSize = 8, RX_WMark = 11 Recommended: MSize = 8, TX_WMark = 8, RX_WMark = 7
27-16	R/W	FIFO_ DEPTH - 1	**RX_WMark** DEPTH - 1 FIFO threshold watermark level when receiving data to card. When FIFO data count reaches greater than this number, DMA/FIFO request is raised. During end of packet, request is generated regardless of threshold programming in order to complete any remaining data. In non-DMA mode, when receiver FIFO threshold (RXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, interrupt is not generated if threshold programming is larger than any remaining data. It is responsibility of host to read remaining bytes on seeing Data Transfer Done interrupt. In DMA mode, at end of packet, even if remaining bytes are less than threshold, DMA request does single transfers to flush out any remaining bytes before Data Transfer Done interrupt is set. 12 bits – 1 bit less than FIFO-count of status register, which is 13 bits. Limitation: RX_WMark <= FIFO_DEPTH-2 Recommended: (FIFO_DEPTH/2) - 1; (means greater than (FIFO_DEPTH/2) - 1)
15-12	R	0	Not Used
11-0	R/W	0	**TX_WMark** FIFO threshold watermark level when transmitting data to card. When FIFO data count is less than or equal to this number, DMA/FIFO request is raised. If Interrupt is enabled, then interrupt occurs. During end of packet, request or interrupt is generated, regardless of threshold programming. In non-DMA mode, when transmit FIFO threshold (TXDR) interrupt is enabled, then interrupt is generated instead of DMA request. During end of packet, on last interrupt, host is responsible for filling FIFO with only required remaining bytes (not before FIFO is full or after CIU completes data transfers, because FIFO may not be empty). In DMA mode, at end of packet, if last transfer is less than burst size, DMA controller does single cycles until required bytes are transferred. 12 bits – 1 bit less than FIFO-count of status register, which is 13 bits. Limitation: TX_WMark >= 1; Recommended: FIFO_DEPTH/2; (means less than or equal to

			FIFO_DEPTH/2)

19.2.21CDETECT: Card Detect Register

Address: SDHC_BASE + 0x50

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

Bit	Type	Reset	Description
31-30	R	0	Not Used
29-0	R	card_detect_n inputs	**card_detect_n** Value on card_detect_n input ports (1 bit per card); read-only bits. 0 represents presence of card. Only NUM_CARDS number of bits are implemented.

19.2.22WRTPRT: Write Protect Register

Address: SDHC_BASE + 0x54

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

Bit	Type	Reset	Description
31-30	R	0	Not Used
29-0	R	card_write_prt inputs	**write_protect** Value on card_write_prt input ports (1 bit per card). 1 represents write protection. Only NUM_CARDS number of bits are implemented.

19.2.23GPIO: General Purpose Input/Output Register

Address: SDHC_BASE + 0x58

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

Bit	Type	Reset	Description
31-0	R	0	Not Used

19.2.24TCBCNT: Transferred CIU Card Byte Count Register

Address: SDHC_BASE + 0x5c

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

Bit	Type	Reset	Description
31-0	R	0	**trans_card_byte_count** Number of bytes transferred by CIU unit to card. In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, internal 16-bit coherency register is implemented. User should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, higher 16 bits of counter are stored in temporary register. When higher 16 bits are read, data from temporary register is supplied. Both TCBCNT and TBBCNT share same coherency register. When AREA_OPTIMIZED parameter is 1, register should be read only after data transfer completes; during data transfer, register returns 0.

19.2.25TBBCNT: Transferred Host to BIU-FIFO Byte Count Register

Address: SDHC_BASE + 0x60

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

Bit	Type	Reset	Description
31-0	R	0	**trans_fifo_byte_count** Number of bytes transferred between Host/DMA memory and BIU FIFO. In 32-bit or 64-bit AMBA data-bus-width modes, register should be accessed in full to avoid read-coherency problems. In 16-bit AMBA data-bus-width mode, internal 16-bit coherency register is implemented. User should first read lower 16 bits and then higher 16 bits. When reading lower 16 bits, higher 16 bits of counter are stored in temporary register. When higher 16 bits are read, data from temporary register is supplied. Both TCBCNT and TBBCNT share same coherency register.

19.2.26DEBNCE: Debounce Count Register

Address: SDHC_BASE + 0x64

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

Bit	Type	Reset	Description

31-24	R	0	Not Used
23-0	R/W	24'hff_ffff	**debounce_count** Number of host clocks (clk) used by debounce filter logic; typical de-bounce time is 5-25 ms.

19.2.27USRID: User ID Register

Address: SDHC_BASE + 0x68

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

Bit	Type	Reset	Description
31-0	R/W	0	**USRID** User identification register

19.2.28RST_n: H/W Reset

Address: SDHC_BASE + 0x78

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

Bit	Type	Reset	Description
31-16	R	0	**reserved**
15-0	R/W	0	**CARD_RESET** Hardware reset. 1 – Active mode 0 – Reset These bits cause the cards to enter pre-idle state, which requires them to be re-initialized. CARD_RESET[0] should be set to 1'b0 to reset card number 0 CARD_RESET[15] should be set to 1'b0 to reset card number 15.

19.2.29BMOD: Bus Mode Register

Address: SDHC_BASE + 0x80

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

Bit	Type	Reset	Description
31-16	R	0	**reserved**
10-8	R	0	**PBL** Programmable Burst Length. These bits indicate the

			maximum number of beats to be performed in one IDMAC transaction. The IDMAC will always attempt to burst as specified in PBL each time it starts a Burst transfer on the host bus. The permissible values are 1, 4, 8, 16, 32, 64, 128 and 256. This value is the mirror of MSIZE of FIFOTH register. In order to change this value, write the required value to FIFOTH register. This is an encode value as follows. 000 – 1 transfers 001 – 4 transfers 010 – 8 transfers 011 – 16 transfers 100 – 32 transfers 101 – 64 transfers 110 – 128 transfers 111 – 256 transfers
7	R/W	0	**DE** IDMAC Enable. When set, the IDMAC is enabled.
6-2	R/W	0	**DSL** Descriptor Skip Length. Specifies the number of HWord/Word/Dword (depending on 16/32/64-bit bus) to skip between two unchained descriptors. This is applicable only for dual buffer structure.
1	R/W	0	**FB** Fixed Burst. Controls whether the AHB Master interface performs fixed burst transfers or not. When set, the AHB will use only SINGLE, INCR4, INCR8 or INCR16 during start of normal burst transfers. When reset, the AHB will use SINGLE and INCR burst transfer operations.
0	R/W	0	**SWR** Software Reset. When set, the DMA Controller resets all its internal Register.It is automatically cleared after 1 clock cycle.

19.2.30 CardThrCtl:Card Threshold Control Register

Address: SDHC_BASE + 0x100

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

Bit	Type	Reset	Description
N-16	R/W	0	**Card Threshold size** Card Threshold size; N depends on the FIFO size: N = 27, FIFO Size = 512 N = 26, FIFO Size = 256 N = 25, FIFO Size = 128 N = 24, FIFO Size = 64 N = 23, FIFO Size = 32 N = 22, FIFO Size = 16 N = 21, FIFO Size = 8
15-3	R	0	**reserved**
2	R/W	0	Card Write Threshold Enable.

			Applicable when HS400 mode is enabled. 0 - Card write Threshold disabled 1 - Card write Threshold enabled
1	R/W	0	Busy Clear Interrupt generation: 1'b0 - Busy Clear Interrupt disabled 1'b1 - Busy Clear Interrupt enabled
0	R/W	0	Card Read Threshold Enable 1'b0 - Card Read Threshold disabled 1'b1 - Card Read Threshold enabled.

19.2.31 Back_end_power: Back-end Power Register

Address: SDHC_BASE + 0x104

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

Bit	Type	Reset	Description
31-0	R/W	0	Back end power 1'b0– Off; Reset 1'b1– Back-end Power supplied to card application; one pin per card

19.2.32EMMC_DDR_REG: eMMC DDR Register

Address: SDHC_BASE + 0x10C

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

Bit	Type	Reset	Description
31	R/W	0	**hs400_mode** HS400 Mode enable 1 - Enable 0 - Disable
30-16	R	0	**reserved**
15-0	R/W	0	**HALF_START_BIT** Control for start bit detection mechanism inside SDMMC based on duration of start bit; each bit refers to one slot. For eMMC 4.5, start bit can be: Full cycle (HALF_START_BIT = 0) Less than one full cycle (HALF_START_BIT = 1) Set HALF_START_BIT=1 for eMMC 4.5 and above; set to 0 for SD applications. **Note:** This bit is not applicable for HS400 mode.

19.2.33ENABLE_SHIFT: Enable Phase Shift Register

Address: SDHC_BASE + 0x110

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

Bit	Type	Reset	Description
31-0	R/W	0	Control for the amount of phase shift provided on the default enables in the design. Two bits are assigned for each card/slot. For example, bits[1:0] control slot0 and indicate the following. 00 Default phase shift 01 Enables shifted to next immediate positive edge 10 Enables shifted to next immediate negative edge 11 Reserved

20. DDR Controller

20.1 DDR3 Controller Features

- Embedded 128MB DDR3 DRAM up to 1600M data rate
- Enter auto or self-refresh mode automatically
- Active and precharge power-down supported in the SDRAM
- Quality of Service (QoS) features for low latency transfers
- Multiple outstanding transactions
- Write data interleaving supported

21. USB Controller

21.1 USB main features

1. Compliant with the USB 2.0 specification for operation at high speed (480Mbit/sec) and full speed (12Mbit/sec).
2. USB Host/Peripheral Function controllers.

21.2USB Register Descriptions

USB_BASE: 0x4000_0000

21.2.1USB_FADDR (Peripheral Mode Only)

FADDR is an 8-bit register that should be written with the 7-bit address of the peripheral part of the transaction. When the USB is being used in Peripheral mode, this register should be written with the address received through a SET_ADDRESS command, which will then be used for decoding the function address in subsequent token packets.

Address: USB_BASE + 0x00

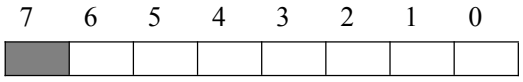
7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
7	-	-	Not used.
6-0	R/W	0	FADDR The function address.

21.2.2POWER

Power is an 8-bit register that is used for controlling Suspend and Resume signaling, and some basic operational aspects of the USB.

Address: USB_BASE + 0x01



Bit	Type	Reset	Description
7	-	-	Not used.
6	R/W	0	Soft Connect If Soft Connect/Disconnect feature is enabled, then the USB D+/D- lines are enabled when this bit is set by the CPU and tri-stated when this bit is cleared by the CPU. *Note: Only valid in Peripheral Mode.*
5	R/W	1	HS Enable When set by the CPU, the USB will negotiate for High-speed mode when the device is reset by the hub. If not set, the device will only operate in Full-speed mode.
4	R	0	HS Mode When set, this read-only bit indicates High-speed mode successfully negotiated during USB reset. In Peripheral Mode, becomes valid when USB reset completes (as indicated by USB reset interrupt). In Host Mode, becomes valid when Reset bit is cleared. Remains valid for the duration of the session. *Note: Allowance is made for Tiny-J signaling in determining the transfer speed to select.*
3	R	0	Reset This bit is set when Reset signaling is present on the bus. Note: This bit is Read/Write from the CPU in Host Mode but Read-Only in Peripheral Mode.
2	R/W	0	Resume Set by the CPU to generate Resume signaling when the function is in Suspend mode. The CPU should clear this bit after 10 ms (a maximum of 15 ms) to end Resume signaling. In Host mode, this bit is also automatically set when Resume signaling from the target is detected while the USB is suspended.
1	R	0	Suspend Mode In Host mode, this bit is set by the CPU to enter Suspend mode. In Peripheral mode, this bit is set on entry into Suspend mode. It is cleared when the CPU reads the interrupt register, or sets the Resume bit above.
0	R/W	0	Enable SuspendM Set by the CPU to enable the SUSPENDM output.

21.2.3INTRTX

IntrTx is a 16-bit read-only register that indicates which interrupts are currently active for

Endpoint 0 and the Tx Endpoints 1–3. *Note also that all active interrupts are cleared when this register is read.*

Address: USB_BASE + 0x02

Bit	Type	Reset	Description
15-4	-	-	Not used.
3	R	0	EP3 TxIntr Tx Endpoint 3 interrupt
2	R	0	EP2 TxIntr Tx Endpoint 2 interrupt
1	R	0	EP1 TxIntr Tx Endpoint 1 interrupt
0	R	0	EP0 Intr Endpoint 0 interrupt

21.2.4INTRRX

IntrRx is a 16-bit read-only register that indicates which of the interrupts for Rx Endpoints 1 – 15 are currently active. *Note also that all active interrupts are cleared when this register is read.*

Address: USB_BASE + 0x04

Bit	Type	Reset	Description
15-4	-	-	Not used.
3	R	0	EP3 RxIntr Rx Endpoint 3 interrupt
2	R	0	EP2 RxIntr Rx Endpoint 2 interrupt
1	R	0	EP1 RXIntr Rx Endpoint 1 interrupt
0	-	-	Not used.

21.2.5INTRTXE

IntrTxE is a 16-bit register that provides interrupt enable bits for the interrupts in IntrTx. On reset, the bits corresponding to Endpoint 0 and the Tx endpoints included in the design are set to 1, while the remaining bits are set to 0.

Address: USB_BASE + 0x06

Bit	Type	Reset	Description
15-4	-	-	Not used.
3	R/W	1	EP3 TxIntrE Tx Endpoint 3 interrupt enable.
2	R/W	1	EP2 TxIntrE

			Tx Endpoint 2 interrupt enable.
1	R/W	1	EP1 TxIntrE Tx Endpoint 1 interrupt enable.
0	R/W	1	EP0 IntrE Endpoint 0 interrupt enable.

21.2.6INTRRXE

IntrRxE is a 16-bit register that provides interrupt enable bits for the interrupts in IntrRx. On reset, the bits corresponding to the Rx endpoints included in the design are set to 1, while the remaining bits are set to 0.

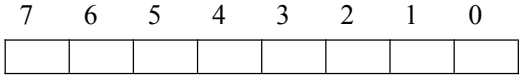
Address: USB_BASE + 0x08

Bit	Type	Reset	Description
15-4	-	-	Not used.
3	R/W	1	EP3 RxIntrE Rx Endpoint 3 interrupt enable.
2	R/W	1	EP2 RxIntrE Rx Endpoint 2 interrupt enable.
1	R/W	1	EP1 RxIntrE Rx Endpoint 1 interrupt enable.
0	-	-	Not used.

21.2.7INTRUSB

IntrUSB is an 8-bit read-only register that indicates which USB interrupts are currently active. All active interrupts will be cleared when this register is read.

Address: USB_BASE + 0x0A



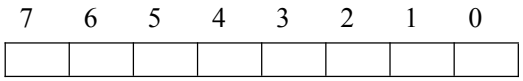
Bit	Type	Reset	Description
7	R	0	VBus Error Set when VBus drops below the VBus Valid threshold during a session. *Only valid when USB is 'A' device.*
6	R	0	Session Request Set when Session Request signaling has been detected. *Only valid when USB is 'A' device.*
5	R	0	Disconnect Set in Host mode when a device disconnects is detected. Set in Peripheral mode when a session ends. *Valid at all transaction speeds.*
4	R	0	Connect Set when a device connection is detected. *Only valid in Host mode. Valid at all transaction speeds.*
3	R	0	SOF Set when a new frame starts.
2	R	0	Reset Set in Peripheral mode when Reset signaling is detected on the e bus.

			Babble Set in Host mode when babble is detected.
1	R	0	Resume Set when Resume signaling is detected on the bus while the USB is in Suspend mode.
0	R	0	Suspend Set when Suspend signaling is detected on the bus. *Only valid in Peripheral mode.*

21.2.8INTRUSBE

IntrUSBE is an 8-bit register that provides interrupt enable bits for each of the interrupts in IntrUSB.

Address: USB_BASE + 0x0B

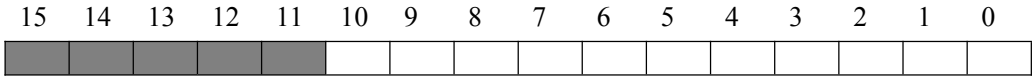


Bit	Type	Reset	Description
7	R/W	0	VBus Error interrupt enable.
6	R/W	0	Session Request interrupt enable.
5	R/W	0	Disconnect interrupt enable.
4	R/W	0	Connect interrupt enable.
3	R/W	0	SOF interrupt enable.
2	R/W	1	Reset/Babble interrupt enable.
1	R/W	1	Resume interrupt enable.
0	R/W	0	Suspend interrupt enable.

21.2.9FRAME

Frame is a 16-bit read-only register that holds the last received frame number.

Address: USB_BASE + 0x0C



Bit	Type	Reset	Description
15-11	-	-	Not used.
10-0	R	0	Frame number

21.2.10INDEX

Each Tx endpoint and each Rx endpoint have their own set of control/status Register. Index is a 4-bit register that determines which endpoint control/status Register are accessed. Before accessing an endpoint's control/status Register, the endpoint number should be written to the Index register to ensure that the correct control/status Register appear in the memory map.

Address: USB_BASE + 0x0E

Bit	Type	Reset	Description
7-4	-	-	Not used.

3-0	R/W	0	INDEX Selected endpoint.

21.2.11TESTMODE

Testmode is an 8-bit register that is primarily used to put the USB into one of the four test modes for High-speed operation described in the USB 2.0 specification – in response to a SET FEATURE: TESTMODE command. It is not used in normal operation. *Note: Only one of Bits 0 – 6 should be set at any time.*

Address: USB_BASE + 0x0F

7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
7	R/W	0	Force_Host The CPU sets this bit to instruct the core to enter Host mode when the Session bit is set, regardless of whether it is connected to any peripheral. The state of the CID input, HostDisconnect and LineState signals are ignored. The core will then remain in Host mode until the Session bit is cleared, even if a device is disconnected, and if the Force_Host bit remains set, will re-enter Host mode the next time the Session bit is set. While in this mode, the status of the HOSTDISCON signal from the PHY may be read from bit 7 of the DevCtl register. The operating speed is determined from the Force_HS and Force_FS bits as follows: Force_HS / Force_FS / Operating Speed 0 / 0 / Low Speed 0 / 1 / Full Speed 1 / 0 / High Speed 1 / 1 / Undefined
6	R/W	0	FIFO_Access The CPU sets this bit to transfer the packet in the Endpoint 0 Tx FIFO to the Endpoint 0 Rx FIFO. It is cleared automatically.
5	R/W	0	Force_FS The CPU sets this bit either in conjunction with bit 7 above or to force the USB into Full-speed mode when it receives a USB reset.
4	R/W	0	Force_HS The CPU sets this bit either in conjunction with bit 7 above or to force the USB into High-speed mode when it receives a USB reset.
3	R/W	0	Test_Packet *(High-speed mode)* The CPU sets this bit to enter the Test_Packet test mode. In this mode, the USB repetitively transmits on the bus a 53-byte test packet, the form of which is defined in the *Universal Serial Bus Specification* Revision 2.0. *Note: The test packet has a fixed format and must be loaded into the Endpoint 0 FIFO before the test mode is entered.*
2	R/W	0	Test_K *(High-speed mode)* The CPU sets this bit to enter the

			Test_K test mode. In this mode, the USB transmits a continuous K on the bus.
1	R/W	0	Test_J *(High-speed mode)* The CPU sets this bit to enter the Test_J test mode. In this mode, the USB transmits a continuous J on the bus.
0	R/W	0	Test_SE0_NAK *(High-speed mode)* The CPU sets this bit to enter the Test_SE0_NAK test mode. In this mode, the USB remains in High-speed mode but responds to any valid IN token with a NAK.

21.2.12DEVCTL

DevCtl is an 8-bit register that is used to select whether the USB is operating in Peripheral mode or in Host mode, and for controlling and monitoring the USB VBus line.

Address: USB_BASE + 0x60

7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
7	R	1	B-Device This Read-only bit indicates whether the USB is operating as the ‘A’ device or the ‘B’ device. 0-‘A’ device; 1-‘B’ device. *Only valid while a session is in progress. Note: If the core is in Force_Host mode (i.e. a session has been started with Testmode.D7 = 1), this bit will indicate the state of the HOSTDISCON input signal from the PHY.*
6	R	0	FSDev This Read-only bit is set when a full-speed or high-speed device has been detected being connected to the port. (High-speed devices are distinguished from full-speed by checking for high-speed chirps when the device is reset.) *Only valid in Host mode.*
5	R	0	LSDev This Read-only bit is set when a low-speed device has been detected being connected to the port. *Only valid in Host mode.*
4-3	R	0	VBus[1:0] These Read-only bits encode the current VBus level as follows: D4 / D3 / Meaning 0 / 0 / Below Session End 0 / 1 / Above Session End, below A Valid 1 / 0 / Above A Valid, below VBus Valid 1 / 1 / Above VBus Valid
2	R	0	Host Mode This Read-only bit is set when the USB is acting as a Host.
1	R/W	0	Host Request When set, the USB will initiate the Host Negotiation when Suspend mode is entered. It is cleared when Host Negotiation is completed. See Section 15. *(‘B’ device only)*
0	R/W	0	Session *When operating as an ‘A’ device,* this bit is set or cleared by the CPU to start or end a session.

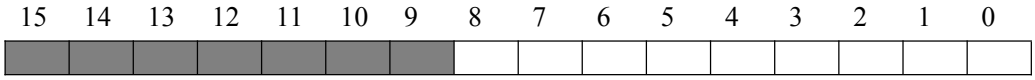
			When operating as a 'B' device, this bit is set/cleared by the USB when a session starts/ends. It is also set by the CPU to initiate the Session Request Protocol. When the USB is in Suspend mode, the bit may be cleared by the CPU to perform software disconnect.

21.2.13CSR0

CSR0 is a 16-bit register that provides control and status bits for Endpoint 0. *Note: The interpretation of the register depends on whether the USB is acting as a peripheral or as a host. Users should also be aware that the value returned when the register is read reflects the status attained e.g. as a result of writing to the register.*

21.2.13.1CSR0 in Peripheral Mode

Address: USB_BASE + 0x12

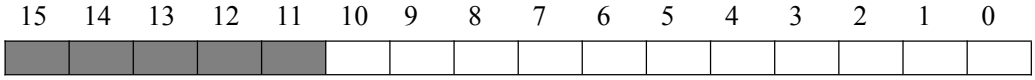


Bit	Type	Reset	Description
15-9	-	-	Not used.
8	SET	0	FlushFIFO The CPU writes a 1 to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset and the TxPktRdy/RxPktRdy bit (below) is cleared. *Note: FlushFIFO has no effect unless TxPktRdy/RxPktRdy is set.*
7	SET	0	ServicedSetupEnd The CPU writes a 1 to this bit to clear the SetupEnd bit. It is cleared automatically.
6	SET	0	ServicedRxPktRdy The CPU writes a 1 to this bit to clear the RxPktRdy bit. It is cleared automatically.
5	SET	0	SendStall The CPU writes a 1 to this bit to terminate the current transaction. The STALL handshake will be transmitted and then this bit will be cleared automatically.
4	R	0	SetupEnd This bit will be set when a control transaction ends before the DataEnd bit has been set. An interrupt will be generated and the FIFO flushed at this time. The bit is cleared by the CPU writing a 1 to the ServicedSetupEnd bit.
3	SET	0	DataEnd The CPU sets this bit: 1. When setting TxPktRdy for the last data packet. 2. When clearing RxPktRdy after unloading the last data packet. 3. When setting TxPktRdy for a zero length data packet. It is cleared automatically.
2	R/CLR	0	SentStall This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.
1	R/SET	0	TxPktRdy The CPU sets this bit after loading a data packet into the

			FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled).
0	R	0	RxPktRdy This bit is set when a data packet has been received. An interrupt is generated when this bit is set. The CPU clears this bit by setting the ServicedRxPktRdy bit.

21.2.13.2CSR0 in Host Mode

Address: USB_BASE + 0x12



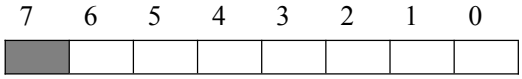
Bit	Type	Reset	Description
15-11	-	-	Not used.
10	SET	0	Data Toggle Write Enable The CPU writes a 1 to this bit to enable the current state of the Endpoint 0 data toggle to be written (see Data Toggle bit, below). This bit is automatically cleared once the new value is written.
9	R/W	0	Data Toggle When read, this bit indicates the current state of the Endpoint 0 data toggle. If D10 is high, this bit may be written with the required setting of the data toggle. If D10 is low, any value written to this bit is ignored.
8	SET	0	FlushFIFO The CPU writes a 1 to this bit to flush the next packet to be transmitted/read from the Endpoint 0 FIFO. The FIFO pointer is reset and the TxPktRdy/RxPktRdy bit (below) is cleared. *Note: FlushFIFO has no effect unless TxPktRdy/RxPktRdy is set.*
7	R/CLR	0	NAK Timeout This bit will be set when Endpoint 0 is halted following the receipt of NAK responses for longer than the time set by the NAKLimit0 register. The CPU should clear this bit to allow the endpoint to continue.
6	R/W	0	StatusPkt The CPU sets this bit at the same time as the TxPktRdy or ReqPkt bit is set, to perform a status stage transaction. Setting this bit ensures that the data toggle is set to 1 so that a DATA1 packet is used for the Status Stage transaction.
5	R/W	0	ReqPkt The CPU sets this bit to request an IN transaction. It is cleared when RxPktRdy is set.
4	R	0	Error This bit will be set when three attempts have been made to perform a transaction with no response from the peripheral. The CPU should clear this bit. An interrupt is generated when this bit is set.
3	R/W	0	SetupPkt The CPU sets this bit, at the same time as the TxPktRdy bit is set, to send a SETUP token instead of an OUT token for the transaction. *Note: Setting this bit also clears the Data Toggle.*
2	R/CLR	0	RxStall

			This bit is set when a STALL handshake is received. The CPU should clear this bit.
1	R/SET	0	TxPktRdy The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled).
0	R/CLR	0	RxPktRdy This bit is set when a data packet has been received. An interrupt is generated (if enabled) when this bit is set. The CPU should clear this bit when the packet has been read from the FIFO.

21.2.14COUNT0

Count0 is a 7-bit read-only register that indicates the number of received data bytes in the Endpoint 0 FIFO. The value returned changes as the contents of the FIFO change and is only valid while RxPktRdy (CSR0.0) is set.

Address: USB_BASE + 0x18

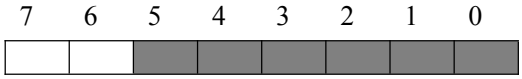


Bit	Type	Reset	Description
7	-	-	Not used.
6-0	R	0	Count0 Endpoint 0 Rx Count.

21.2.15TYPE0 (Host Mode Only)

Type0 is an 8-bit register of which only bits 6 and 7 are implemented. These bits should be written with the operating speed of the targeted device.

Address: USB_BASE + 0x1A



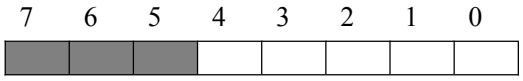
Bit	Type	Reset	Description
7-6	R/W	0	Speed Operating speed of the target device: 00: Not used *(Note: If selected, the target will be assumed to be using the same connection speed as the core.)* 01: High 10: Full 11: Low
5-0	-	-	Not used.

21.2.16NAKLIMIT0 (Host Mode Only)

NAKLimit0 is a 5-bit register that sets the number of frames/microframes (High-Speed transfers) after which Endpoint 0 should timeout on receiving a stream of NAK responses with the Index register set to 0. (Equivalent settings for other endpoints can be made through their TxInterval and RxInterval Register. The number of frames/microframes selected is $2^{(m-1)}$ (where *m* is the value set

in the register, valid values 2 – 16). If the host receives NAK responses from the target for more frames than the number represented by the Limit set in this register, the endpoint will be halted. *Note: A value of 0 or 1 disables the NAK timeout function.*

Address: USB_BASE + 0x1B



Bit	Type	Reset	Description
7-5	-	-	Not used.
4-0	R/W	0	NAKLimit0 Endpoint 0 NAK Limit (m)

21.2.17TXMAXP

The TxMaxP register defines the maximum amount of data that can be transferred through the selected Tx endpoint in a single operation. There is a TxMaxP register for each Tx endpoint (except Endpoint 0).

Bits 10:0 define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for Bulk and Interrupt transfers in Fullspeed and High-speed operations.

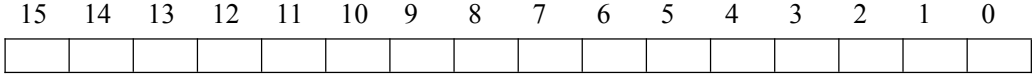
In the case of Bulk endpoints with the packet splitting option enabled, the multiplier *m* can be up to 32 and defines the maximum number of 'USB' packets (i.e. packets for transmission over the USB) of the specified payload into which a single data packet placed in the FIFO should be split, prior to transfer. (If the packet splitting option is not enabled, D15–D13 is not implemented and D12–D11 (if included) is ignored.) *Note: The data packet is required to be an exact multiple of the payload specified by bits 10:0, which is itself required to be either 8, 16, 32, 64 or (in the case of High Speed transfers) 512 bytes.*

The value written to bits 10:0 must match the value given in the wMaxPacketSize field of the Standard Endpoint Descriptor for the associated endpoint. A mismatch could cause unexpected results.

The total amount of data represented by the value written to this register (specified payload × *m*) must not exceed the FIFO size for the Tx endpoint, and should not exceed half the FIFO size if double-buffering is required.

If this register is changed after packets have been sent from the endpoint, the Tx endpoint FIFO should be completely flushed (using the FlushFIFO bit in TxCSR) after writing the new value to this register.

Address: USB_BASE + 0x10



Bit	Type	Reset	Description
15/12-11	R/W	0	m-1
10-0	R/W	0	Maximum Payload/transaction

21.2.18TXCSR

TxCSR is a 16-bit register that provides control and status bits for transfers through the currently selected Tx endpoint. There is a TxCSR register for each Tx endpoint (not including Endpoint 0). *Note: The interpretation of the register depends on whether the USB is acting as a peripheral or as a host. Users should also be aware that the value returned when the register is read reflects the status attained e.g. as a result of writing to the register.*

21.2.18.1In Peripheral Mode

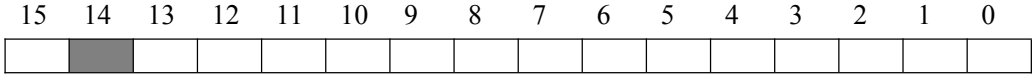
Address: USB_BASE + 0x12

Bit	Type	Reset	Description
15	R/W	0	AutoSet If the CPU sets this bit, TxPktRdy will be automatically set when data of the maximum packet size (value in TxMaxP) is loaded into the Tx FIFO. If a packet of less than the maximum packet size is loaded, then TxPktRdy will have to be set manually.
14	R/W	0	ISO The CPU clears it to enable the Tx endpoint for Bulk or Interrupt transfers. *Note: This bit only has any effect in Peripheral mode. In Host mode, it always returns zero.*
13	R/W	0	Mode The CPU sets this bit to enable the endpoint direction as Tx, and clears the bit to enable it as Rx. *Note: This bit only has any effect where the same endpoint FIFO is used for both Tx and Rx transactions.*
12	R/W	0	DMAReqEnab The CPU sets this bit to enable the DMA request for the Tx endpoint.
11	R/W	0	FrcDataTog The CPU sets this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received.
10	R/W	0	DMAReqMode The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0. *Note: This bit must not be cleared either before or in the same cycle as the above DMAReqEnab bit is cleared.*
9-8	-	-	Not used.
7	R/CLR	0	IncompTx When the endpoint is being used for high-bandwidth Interrupt transfers, this bit is set to indicate where a large packet has been split into 2 or 3 packets for transmission but insufficient IN tokens have been received to send all the parts. *Note: In anything other than a high-bandwidth transfer, this bit will always return 0.*
6	SET	0	ClrDataTog The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.
5	R/CLR	0	SentStall This bit is set when a STALL handshake is transmitted. The FIFO is flushed and the TxPktRdy bit is cleared (see below).

			The CPU should clear this bit.
4	R/W	0	SendStall The CPU writes a 1 to this bit to issue a STALL handshake to an IN token. The CPU clears this bit to terminate the stall condition.
3	SET	0	FlushFIFO The CPU writes a 1 to this bit to flush the latest packet from the endpoint Tx FIFO. The FIFO pointer is reset, the TxPktRdy bit (below) is cleared and an interrupt is generated. May be set simultaneously with TxPktRdy to abort the packet that is currently being loaded into the FIFO. *Note: FlushFIFO has no effect unless TxPktRdy is set. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set twice to completely clear the FIFO.*
2	R/CLR	0	UnderRun The USB sets this bit if an IN token is received when TxPktRdy is not set. The CPU should clear this bit.
1	R/CLR	0	FIFONotEmpty The USB sets this bit when there is at least 1 packet in the Tx FIFO.
0	R/SET	0	TxPktRdy The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled). TxPktRdy is also automatically cleared (but no interrupt is generated) prior to loading a second packet into a double-buffered FIFO.

21.2.18.2In Host Mode

Address: USB_BASE + 0x12



Bit	Type	Reset	Description
15	R/W	0	AutoSet If the CPU sets this bit, TxPktRdy will be automatically set when data of the maximum packet size (value in TxMaxP) is loaded into the Tx FIFO. If a packet of less than the maximum packet size isloaded, then TxPktRdy will have to be set manually.
14	-	-	Not used.
13	R/W	0	Mode The CPU sets this bit to enable the endpoint direction as Tx, and clears it to enable the endpoint direction as Rx. *Note: This bit only has any effect where the same endpoint FIFO is used for both Tx and Rx transactions.*
12	R/W	0	DMAReqEnab The CPU sets this bit to enable the DMA request for the Tx endpoint.
11	R/W	0	FrcDataTog The CPU sets this bit to force the endpoint data toggle to switch and the data packet to be cleared from the FIFO, regardless of whether an ACK was received.
10	R/W	0	DMAReqMode

			The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0. *Note: This bit must not be cleared either before or in the same cycle as the above DMAReqEnab bit is cleared.*
9	SET	0	Data Toggle Write Enable The CPU writes a 1 to this bit to enable the current state of the Tx Endpoint data toggle to be written (see Data Toggle bit, below). This bit is automatically cleared once the new value is written.
8	R/W	0	Data Toggle When read, this bit indicates the current state of the Tx Endpoint data toggle. If D9 is high, this bit may be written with the required setting of the data toggle. If 9 is low, any value written to this bit is ignored.
7	R/CLR	0	NAK Timeout This bit will be set when the Tx endpoint is halted following the receipt of NAK responses for longer than the time set as the NAK Limit by the TxInterval register. The CPU should clear this bit to allow the endpoint to continue. *Note: Valid only for Bulk endpoints.*
6	SET	0	ClrDataTog The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.
5	R/CLR	0	RxStall This bit is set when a STALL handshake is received. When this bit is set, any DMA request that is in progress is stopped, the FIFO is completely flushed and the TxPktRdy bit is cleared (see below). The CPU should clear this bit.
4	R/W	0	SetupPkt The CPU sets this bit, at the same time as the TxPktRdy bit is set, to send a SETUP token instead of an OUT token for the transaction. *Note: Setting this bit also clears the Data Toggle.*
3	SET	0	FlushFIFO The CPU writes a 1 to this bit to flush the latest packet from the endpoint Tx FIFO. The FIFO pointer is reset, the TxPktRdy bit (below) is cleared and an interrupt is generated. May be set simultaneously with TxPktRdy to abort the packet that is currently being loaded into the FIFO. *Note: FlushFIFO has no effect unless TxPktRdy is set. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set twice to completely clear the FIFO.*
2	R/CLR	0	Error The USB sets this bit when 3 attempts have been made to send a packet and no handshake packet has been received. When the bit is set, an interrupt is generated, TxPktRdy is cleared and the FIFO is completely flushed. The CPU should clear this bit. *Valid only when the endpoint is operating in Bulk or Interrupt mode.*
1	R/CLR	0	FIFONotEmpty The USB sets this bit when there is at least 1 packet in the Tx FIFO.
0	R/SET	0	TxPktRdy The CPU sets this bit after loading a data packet into the FIFO. It is cleared automatically when a data packet has been transmitted. An interrupt is also generated at this point (if enabled). TxPktRdy is also automatically cleared (but no

			interrupt is generated) prior to loading a second packet into a double-buffered FIFO.

21.2.19RXMAXP

The RxMaxP register defines the maximum amount of data that can be transferred through the selected Rx endpoint in a single operation. There is a RxMaxP register for each Rx endpoint (except Endpoint 0).

Bits 10:0 define (in bytes) the maximum payload transmitted in a single transaction. The value set can be up to 1024 bytes but is subject to the constraints placed by the USB Specification on packet sizes for Bulk and Interrupt transfers in Fullspeed and High-speed operations.

For Bulk endpoints with the packet splitting option enabled, the multiplier *m* can be up to 32 and defines the number of USB packets of the specified payload which are to be amalgamated into a single data packet within the FIFO. (If the packet splitting option is not enabled, D15–D13 is not implemented and D12–D11 (if included) is ignored.)

The value written to bits 10:0 must match the value given in the *wMaxPacketSize* field of the Standard Endpoint Descriptor for the associated endpoint. A mismatch could cause unexpected results.

The total amount of data represented by the value written to this register (specified payload × *m*) must not exceed the FIFO size for the RX endpoint, and should not exceed half the FIFO size if double-buffering is required.

Address: USB_BASE + 0x14

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
15/12-11	R/W	0	m-1
10-0	R/W	0	Maximum Payload/transaction

21.2.20RXCSR

RxCSR is a 16-bit register that provides control and status bits for transfers through the currently-selected Rx endpoint. There is a RxCSR register for each Rx endpoint (not including Endpoint 0).

Note: The interpretation of the register depends on whether the USB is acting as a peripheral or as a host. Users should also be aware that the value returned when the register is read reflects the status attained e.g. as a result of writing to the register.

21.2.20.1In Peripheral Mode

Address: USB_BASE + 0x16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
15	R/W	0	AutoClear If the CPU sets this bit then the RxPktRdy bit will be automatically cleared when a packet of RxMaxP bytes has been unloaded from the Rx FIFO. When packets of less than the maximum packet size are unloaded, RxPktRdy will have

			to be cleared manually.
14	R/W	0	ISO The CPU clears it to enable the Rx endpoint for Bulk/Interrupt transfers.
13	R/W	0	DMAReqEnab The CPU sets this bit to enable the DMA request for the Rx endpoint.
12	R/W	0	DisNyet The CPU sets this bit to disable the sending of NYET handshakes. When set, all successfully received Rx packets are ACK'd including at the point at which the FIFO becomes full. *Note: This bit only has any effect in High-speed mode, in which mode it should be set for all Interrupt endpoints.*
11	R/W	0	DMAReqMode The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0.
10-8	-	-	Not used.
7	SET	0	ClrDataTog The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.
6	R/CLR	0	SentStall This bit is set when a STALL handshake is transmitted. The CPU should clear this bit.
5	R/W	0	SendStall The CPU writes a 1 to this bit to issue a STALL handshake. The CPU clears this bit to terminate the stall condition.
4	SET	0	FlushFIFO The CPU writes a 1 to this bit to flush the next packet to be read from the endpoint Rx FIFO. The FIFO pointer is reset and the RxPktRdy bit (below) is cleared. *Note: FlushFIFO has no effect unless RxPktRdy is set. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set twice to completely clear the FIFO.*
3-2	-	-	Not used.
1	R	0	FIFOFull This bit is set when no more packets can be loaded into the Rx FIFO.
0	R/CLR	0	RxPktRdy This bit is set when a data packet has been received. The CPU should clear this bit when the packet has been unloaded from the Rx FIFO. An interrupt is generated when the bit is set.

21.2.20.2In Host Mode

Address: USB_BASE + 0x16

Bit	Type	Reset	Description
15	R/W	0	AutoClear If the CPU sets this bit then the RxPktRdy bit will be automatically cleared when a packet of RxMaxP bytes has been unloaded from the Rx FIFO. When packets of less than the maximum packet size are unloaded, RxPktRdy will

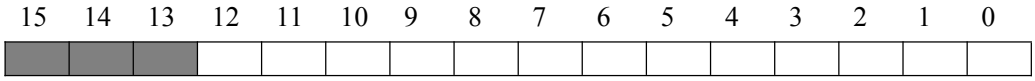
			have to be cleared manually.
14	R/W	0	AutoReq If the CPU sets this bit, the ReqPkt bit will be automatically set when the RxPktRdy bit is cleared.
13	R/W	0	DMAReqEnab The CPU sets this bit to enable the DMA request for the Rx endpoint.
12	R/W	0	DisNyet The CPU sets this bit to disable the sending of NYET handshakes. When set, all successfully received Rx packets are ACK'd including at the point at which the FIFO becomes full. *Note: This bit only has any effect in High-speed mode, in which mode it should be set for all Interrupt endpoints.*
11	R/W	0	DMAReqMode The CPU sets this bit to select DMA Request Mode 1 and clears it to select DMA Request Mode 0.
10	R	0	Data Toggle Write Enable The CPU writes a 1 to this bit to enable the current state of the Endpoint 0 data toggle to be written (see Data Toggle bit, below). This bit is automatically cleared once the new value is written.
9	R	0	Data Toggle When read, this bit indicates the current state of the Endpoint 0 data toggle. If D10 is high, this bit may be written with the required setting of the data toggle. If D10 is low, any value written to this bit is ignored.
8	-	-	Not used.
7	SET	0	ClrDataTog The CPU writes a 1 to this bit to reset the endpoint data toggle to 0.
6	R/CLR	0	RxStall When a STALL handshake is received, this bit is set and an interrupt is generated. The CPU should clear this bit.
5	R/W	0	ReqPkt The CPU writes a 1 to this bit to request an IN transaction. It is cleared when RxPktRdy is set.
4	SET	0	FlushFIFO The CPU writes a 1 to this bit to flush the next packet to be read from the endpoint Rx FIFO.The FIFO pointer is reset and the RxPktRdy bit (below) is cleared. *Note: FlushFIFO has no effect unless RxPktRdy is set. Also note that, if the FIFO is double-buffered, FlushFIFO may need to be set twice to completely clear the FIFO.*
3	R/CLR	0	DataError/NAK Timeout In Bulk mode, this bit will be set when the Rx endpoint is halted following the receipt of NAK responses for longer than the time set as the NAK Limit by the RxInterval register. The CPU should clear this bit to allow the endpoint to continue.
2	R/CLR	0	Error The USB sets this bit when 3 attempts have been made to receive a packet and no data packet has been received. The CPU should clear this bit. An interrupt is generated when the bit is set.
1	R	0	FIFOFull

			This bit is set when no more packets can be loaded into the Rx FIFO.
0	R/CLR	0	RxPktRdy This bit is set when a data packet has been received. The CPU should clear this bit when the packet has been unloaded from the Rx FIFO. An interrupt is generated when the bit is set.

21.2.21RXCOUNT

RxCount is a 13-bit read-only register that holds the number of received data bytes in the packet in the Rx FIFO. *Note: The value returned changes as the FIFO is unloaded and is only valid while RxPktRdy (RxCSR.0) is set.*

Address: USB_BASE + 0x18

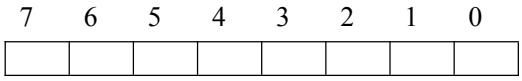


Bit	Type	Reset	Description
15-13	-	-	Not used.
12-0	R	0	Endpoint Rx Count

21.2.22TXTYPE (Host Mode Only)

TxType is an 8-bit register that should be written with the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected Tx endpoint, and its operating speed. There is a TxType register for each Tx endpoint (except Endpoint 0, which has its own Type register at 1Ah).

Address: USB_BASE + 0x1A



Bit	Type	Reset	Description
7-6	R/W	0	Speed Operating speed of the target device: 00: Not used. *(Note: If selected, the target will be assumed to be using the same connection speed as the core.)* 01: High 10: Full 11: Low
5-4	R/W	0	Protocol The CPU should set this to select the required protocol for the Tx endpoint: 00: Control 10: Bulk 11: Interrupt
3-0	R/W	0	Target Endpoint Number The CPU should set this value to the endpoint number contained in the Tx endpoint descriptor returned to the USB during device enumeration.

21.2.23TXINTERVAL (Host Mode Only)

TxInterval is an 8-bit register that, for Interrupt transfers, defines the polling interval for the currently-selected Tx endpoint. For Bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving stream of NAK responses.

Address: USB_BASE + 0x1B

7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
7-0	R/W	0	Tx Polling Interval/NAK Limit (m)

There is a TxInterval register for each Tx endpoint (except Endpoint 0). In each case the value that is set defines a number of frames/microframes (High Speed transfers), as follows:

Transfer Type	Speed	Valid values (m)	Interpretation
Bulk	FS/HS	2-16	NAK Limit is $2^{(m-1)}$ frames/microframes. *Note: A value of 0 or 1 disables the NAK timeout function.*
Interrupt	FS/LS	1-255	Polling interval is m frames.
	HS	1-16	Polling interval is $2^{(m-1)}$ microframes.

21.2.24RXTYPE (Host Mode Only)

RxType is an 8-bit register that should be written with the endpoint number to be targeted by the endpoint, the transaction protocol to use for the currently-selected Rx endpoint, and its operating speed. There is a RxType register for each Rx endpoint (except Endpoint 0, which has its own Type register at 1Ah).

Address: USB_BASE + 0x1C

7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
7-6	R/W	0	Speed Operating speed of the target device: 00: Not used. *(Note: If selected, the target will be assumed to be using the same connection speed as the core.)* 01: High 10: Full 11: Low
5-4	R/W	0	Protocol The CPU should set this to select the required protocol for the Rx endpoint: 00: Control 10: Bulk 11: Interrupt
3-0	R/W	0	Target Endpoint Number The CPU should set this value to the endpoint number contained in the Rx endpoint descriptor returned to the USB during device enumeration.

21.2.25RXINTERVAL (Host Mode Only)

RxInterval is an 8-bit register that, for Interrupt transfers, defines the polling interval for the currently-selected Rx endpoint. For Bulk endpoints, this register sets the number of frames/microframes after which the endpoint should timeout on receiving stream of NAK responses. There is a RxInterval register for each Rx endpoint (except Endpoint 0).

Address: USB_BASE + 0x1D

7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
7-0	R/W	0	Rx Polling Interval/NAK Limit (m)

In each case the value that is set defines a number of frames/microframes (High Speed transfers), as follows:

Transfer Type	Speed	Valid values (m)	Interpretation
Bulk	FS/HS	2-16	NAK Limit is $2^{(m-1)}$ frames/microframes. *Note: A value of 0 or 1 disables the NAK timeout function.*
Interrupt	FS/LS	1-255	Polling interval is m frames.
	HS	1-16	Polling interval is $2^{(m-1)}$ microframes.

21.2.26FIFOX (Address 20h ~2Ch)

This address range provides 4 addresses for CPU access to the FIFOs for each endpoint. Writing to these addresses loads data into the TxFIFO for the corresponding endpoint. Reading from these addresses unloads data from the RxFIFO for the corresponding endpoint.

The address range is 20h – 2Ch and the FIFOs are located on 32-bit double-word boundaries (Endpoint 0 at 20h, Endpoint 1 at 24h, Endpoint 2 at 28h, and Endpoint 3 at 2Ch).

Note: (i) Transfers to and from FIFOs may be 8-bit, 16-bit or 32-bit as required, and any combination of access is allowed provided the data accessed is contiguous. However, all the transfers associated with one packet must be of the same width so that the data is consistently byte-, word- or double-word-aligned. The last transfer may however contain fewer bytes than the previous transfers in order to complete an odd-byte or odd-word transfer.

(ii) Depending on the size of the FIFO and the expected maximum packet size, the FIFOs support either single-packet or doublepacket buffering. However, burst writing of multiple packets is not supported as flags need to be set after each packet is written.

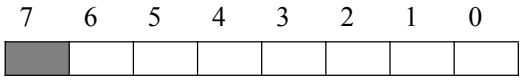
(iii) Following a STALL response or a Tx Strike Out error on Endpoint 1 – 3, the associated FIFO is completely flushed.

21.2.27TXFUNCADDR/RXFUNCADDR (Host Mode Only)

TxFuncAddr and RxFuncAddr are 7-bit read/write Register that record the address of the target function that is to be accessed through the associated endpoint (EP*n*). TxFuncAddr needs to be defined for each Tx endpoint that is used; RxFuncAddr needs to be defined for each Rx endpoint that is used.

Note: Both TxFuncAddr and RxFuncAddr need to be defined for Endpoint 0.

Address: USB_BASE + 0x80+8*n



Bit	Type	Reset	Description
7	-	-	Not used.
6-0	R/W	0	Address of Target Function

21.3DMA Controller Register Description

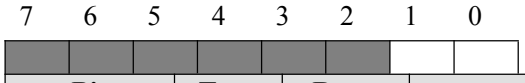
The USB include a 2-channel DMA controller.

The DMA controller has one interrupt register which indicates which channels have a pending interrupt, and a set of three control Register for each DMA channel.

21.3.1DMAINTR

DMAINTR is a 2-bit register that indicates which of the interrupts for DMA channels are currently active.

Address: USB_BASE + 0x200



Bit	Type	Reset	Description
7-2	-	-	Not used.
1	R	0	DMA channel 2 interrupts.
0	R	0	DMA channel 1 interrupt.

21.3.2DMACTL1

DMACTL1 is a 16-bit register that provides control and status bits for DMA channel 1.

Address: USB_BASE + 0x204

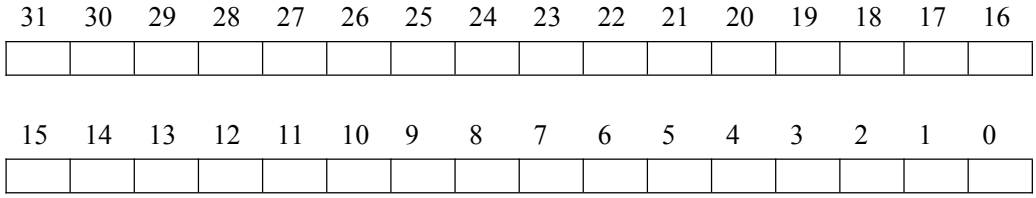
Bit	Type	Reset	Description
15	R/W	0	ADDRINC 0: During DMA, AHB memory addresses auto increase. 1: During DMA, AHB memory address don't increase.
14-11	-	-	**Not used.**
10-9	R/W	0	Burst Mode 00 - Burst Mode 0: Bursts of unspecified length 01 - Burst Mode 1: INCR4 or unspecified length 10 - Burst Mode 2: INCR8, INCR4 or unspecified length 11 - Burst Mode 3: INCR16, INCR8, INCR4 or unspecified length
8	R/W	0	BUS ERROR
7-6	-	-	Not used.
5-4	R/W	0	Endpoint Numbers
3	R/W	0	Interrupt Enable
2	R/W	0	DMA Mode
1	R/W	0	Direction 0: DMA Write (Rx endpoint).

			1: DMA Read (Tx endpoint).
0	R/W	0	Enable DMA

21.3.3DMAADDR1

DMAADDR1 is a 32-bit register that provides memory address for DMA channel 1.

Address: USB_BASE + 0x208

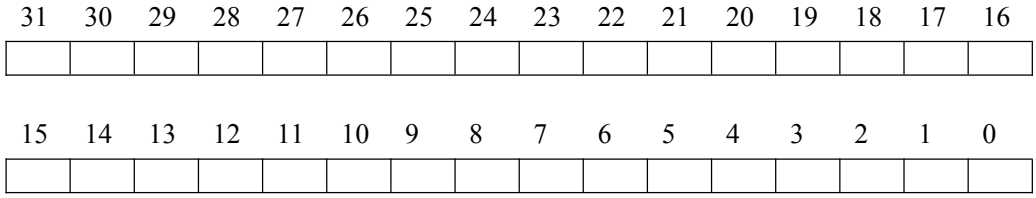


Bit	Type	Reset	Description
31-0	R/W	0	DMAADDR1 DMA Channel 1 AHB Memory Address.

21.3.4DMACOUNT1

DMACOUNT1 is a 32-bit register that provides number of the data should be translation thought DMA channel 1.

Address: USB_BASE + 0x20C



Bit	Type	Reset	Description
31-0	R/W	0	DMACOUNT1 DMA Channel 1 Byte Count.

21.3.5DMACTL2

DMACTL2 is a 16-bit register that provides control and status bits for DMA channel 2.

Address: USB_BASE + 0x214

Bit	Type	Reset	Description
15-11	-	-	Not used.
10-9	R/W	0	Burst Mode 00 = Burst Mode 0: Bursts of unspecified length 01 = Burst Mode 1: INCR4 or unspecified length 10 = Burst Mode 2: INCR8, INCR4 or unspecified length 11 = Burst Mode 3: INCR16, INCR8, INCR4 or unspecified length
8	R/W	0	BUS ERROR
7-6	-	-	Not used.
5-4	R/W	0	Endpoint Numbers
3	R/W	0	Interrupt Enable

2	R/W	0	DMA Mode
1	R/W	0	Direction 0: DMA Write (Rx endpoint). 1: DMA Read (Tx endpoint).
0	R/W	0	Enable DMA

21.3.6DMAADDR2

DMAADDR2 is a 32-bit register that provides memory address for DMA channel 2.

Address: USB_BASE + 0x218

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

Bit	Type	Reset	Description
31-0	R/W	0	DMAADDR2 DMA Channel 2 AHB Memory Address.

21.3.7DMACOUNT2

DMACOUNT2 is a 32-bit register that provides number of the data should be translation thought DMA channel 2.

Address: USB_BASE + 0x21C

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

Bit	Type	Reset	Description
31-0	R/W	0	DMACOUNT2 DMA Channel 2 Byte Count.

22. DMAC

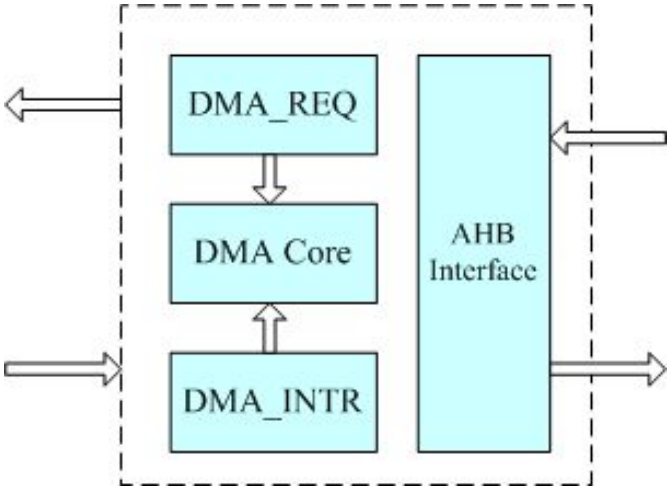
22.1 Features

- 16 DMA requests and response lines. The DMAC provides 16 peripheral DMA request lines.
- Programmable DMA burst size. The DMA burst size can be programmed to more efficiently transfer data. Usually the burst size is set to half the size of the FIFO in the peripheral. Support single and burst transfer.
- Support memory-to-memory, memory-to-peripheral, peripheral-to-memory, and

peripheral-to-peripheral transfers and support DMAC or destination peripheral or source peripheral as the flow control.

- Scatter or gather DMA is supported through the use of linked lists.
- Incrementing or non-incrementing addressing for source and destination.
- Separate and combined DMA error and DMA count interrupt requests. Interrupt masking. The DMA error and DMA terminal count interrupt requests can be masked. The DMA error and DMA count raw interrupt status can be read prior to masking.

22.2 DMAC Architecture Diagram



DMAC Architecture

22.3 DMAC Register Descriptions

DMACIntStatus: Interrupt Status Register

The DMACIntStatus Register is read-only and shows the status of the interrupts after masking. A HIGH bit indicates that a specific DMA channel interrupt request is active.

The request can be generated from either the error or terminal count interrupt requests.

Address: 0x4020_0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

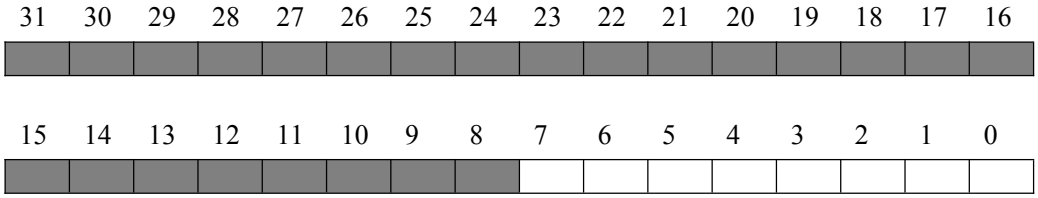
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits	Type	Reset	Description
31-8	R	0	Not used.
7-0	R	0	Status of the DMA interrupts after masking.

DMACIntTCStatus: Interrupt TC Status Register

The DMACIntTCStatus Register is read-only and indicates the status of the terminal count after masking. This register must be used in conjunction with the DMACIntStatus Register if the combined interrupt request, **DMACINTR**, is used to request interrupts. If the **DMACINTTC** interrupt request is used then you only have to read the DMACIntTCStatus Register to ascertain the source of the interrupt request.

Address: 0x4020_0004

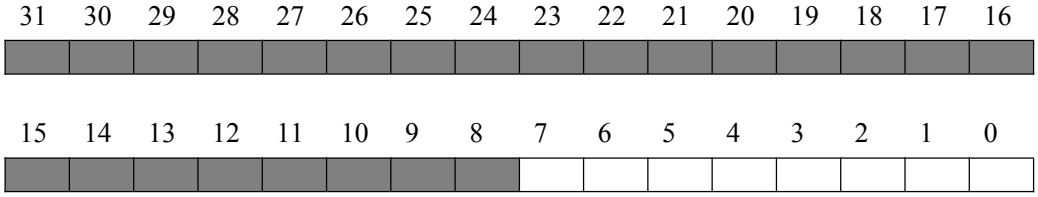


Bits	Type	Reset	Description
31-8	R	0	Not used.
7-0	R	0	Interrupt terminal count request status.

DMACIntTCClear: Interrupt TC Clear Register

The DMACIntTCClear Register is write-only and is used to clear a terminal count interrupt request. When writing to this register, each data bit that is set HIGH causes the corresponding bit in the Status Register to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.

Address: 0x4020_0008

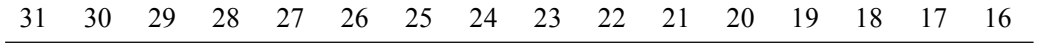


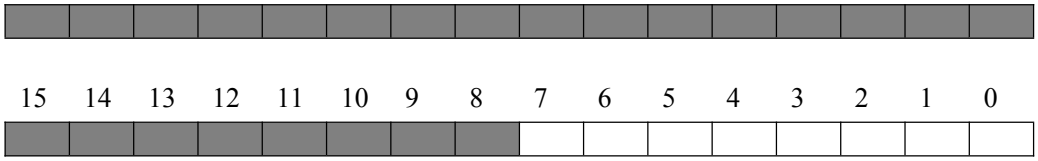
Bits	Type	Reset	Description
31-8	R	0	Not used.
7-0	W	0	Terminal count request clear.

DMACIntErrorStatus: Interrupt Error Status Register

The DMACIntErrorStatus Register is read-only and indicates the status of the error request after masking. You must use this register in conjunction with the DMACIntStatus Register if the combined interrupt request, **DMACINTR**, is used to request interrupts. If the **DMACINTERR** interrupt request is used then you only need to read the DMACIntErrorStatus Register.

Address: 0x4020_000c



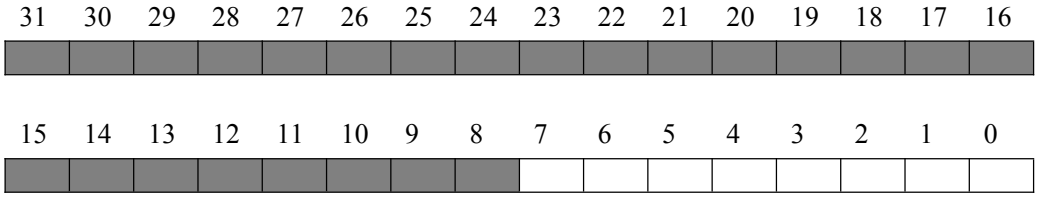


Bits	Type	Reset	Description
31-8	R	0	Not used.
7-0	R	0	Interrupt error status.

DMACIntErrClr: Interrupt Error Clear Register

The DMACIntErrClr Register is write-only and is used to clear the error interrupt requests. When writing to this register, each data bit that is HIGH causes the corresponding bit in the Status Register to be cleared. Data bits that are LOW have no effect on the corresponding bit in the register.

Address: 0x4020_0010

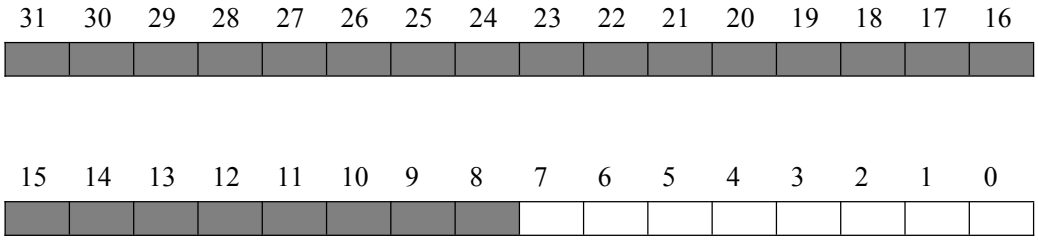


Bits	Type	Reset	Description
31-8	R	0	Not used.
7-0	W	0	Interrupt error clear.

DMACRawIntTCStatus: Raw Interrupt TC Status Register

The DMACRawIntTCStatus Register is read-only. It indicates DMA channels that are requesting a transfer complete (terminal count interrupt) prior to masking. A HIGH bit indicates that the terminal count interrupt request is active prior to masking.

Address: 0x4020_0014



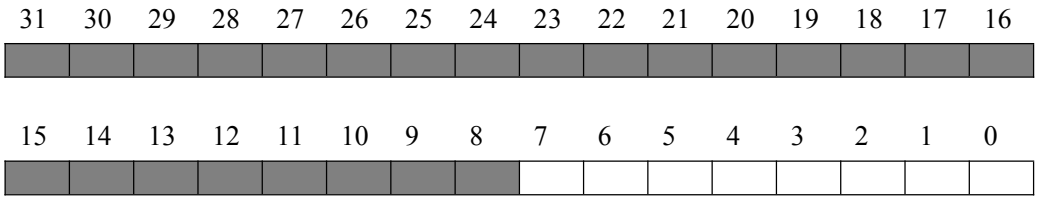
Bits	Type	Reset	Description
31-8	R	0	Not used.
7-0	R	0	Status of the terminal count interrupt prior to masking.

DMACRawIntErrorStatus: Raw Error Interrupt Status Register

The DMACRawIntErrorStatus Register is read-only. It indicates DMA channels that are

requesting an error interrupt prior to masking. A HIGH bit indicates that the error interrupt request is active prior to masking.

Address: 0x4020_0018

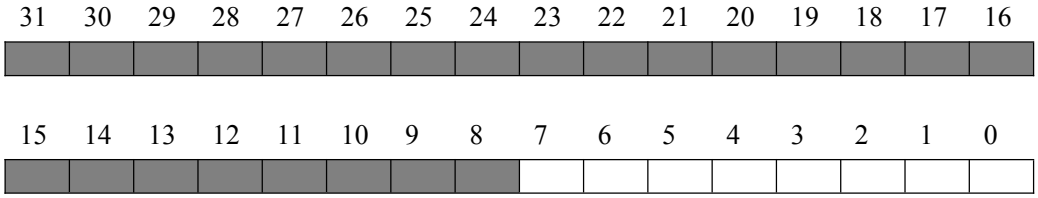


Bits	Type	Reset	Description
31-8	R	0	Not used.
7-0	R	0	Status of the error interrupt prior to masking.

DMACEnbldChns: Enabled Channel Register

The DMACEnbldChns Register is read-only and indicates which DMA channels are enabled, as indicated by the Enable bit in the DMACCxConfiguration Register. A HIGH bit indicates that a DMA channel is enabled. A bit is cleared on completion of the DMA transfer.

Address: 0x4020_001c

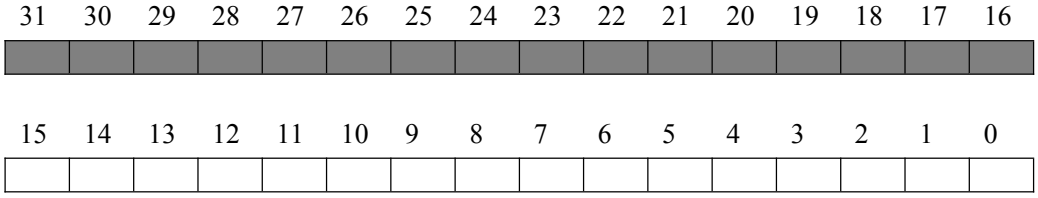


Bits	Type	Reset	Description
31-8	R	0	Not used.
7-0	R	0	Channel enable status.

DMACSoftBReq: Software Burst Request Register

The DMACSoftBReq Register is read/write and it enables DMA burst requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Writing 0 to this register has no effect. Reading the register indicates the sources that are requesting DMA burst transfers. A request can be generated from either a peripheral or the software request register.

Address: 0x4020_0020



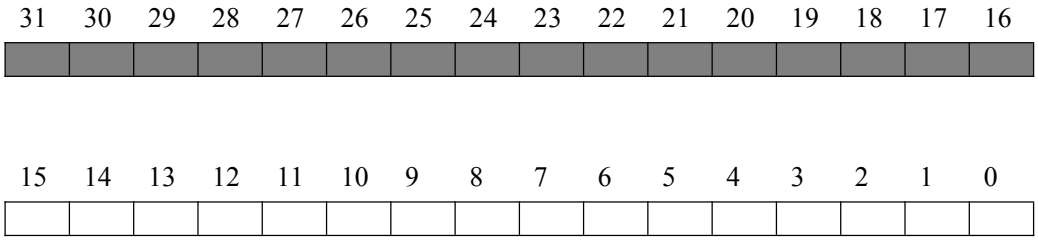
Bits	Type	Reset	Description
31-16	R	0	Not used.
15-0	R/W	0	Software burst request.

Note: It is recommended that software and hardware peripheral requests are not used at the same time.

DMACSoftSReq: Software Single Request Register

The DMACSoftSReq Register is read/write and enables DMA single requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Writing 0 to this register has no effect. Reading the register indicates the sources that are requesting single DMA transfers. A request can be generated from either a peripheral or the software request register.

Address: 0x4020_0024



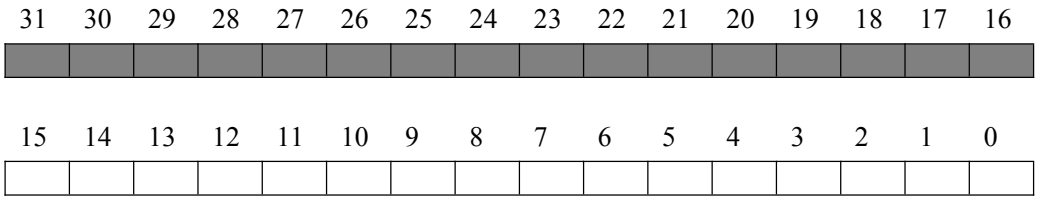
Bits	Type	Reset	Description
31-16	R	0	Not used.
15-0	R/W	0	Software single request.

Note: It is recommended that software and hardware peripheral requests are not used at the same time.

DMACSoftLBReq: Software LB Request Register

The DMACSoftLBReq Register is read/write and enables DMA last burst requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Writing 0 to this register has no effect. Reading the register indicates the sources that are requesting last burst DMA transfers. A request can be generated from either a peripheral or the software request register.

Address: 0x4020_0028

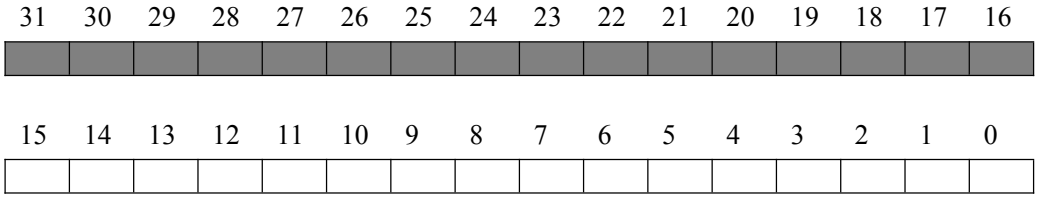


Bits	Type	Reset	Description
31-16	R	0	Not used.
15-0	R/W	0	Software last burst request.

DMACSoftLSReq: Software LS Request Register

The DMACSoftLSReq Register is read/write and enables DMA last single requests to be generated by software. A DMA request can be generated for each source by writing a 1 to the corresponding register bit. A register bit is cleared when the transaction has completed. Writing 0 to this register has no effect. Reading the register indicates the sources that are requesting last single DMA transfers. A request can be generated from either a peripheral or the software request register.

Address: 0x4020_002c



Bits	Type	Reset	Description
31-16	R	0	Not used.
15-0	R/W	0	Software last single request.

A very important Note for software request configuration:

You should make sure the following situations **not happened** when programming the software request Register:

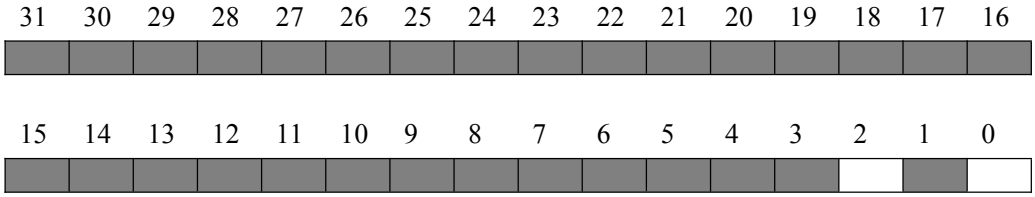
- Single / Last Single / Last Burst destination request asserted in memory to peripheral transfer with the DMAC as the flow controller.
- Last Single / Last Burst source request asserted in peripheral to memory transfer with the DMAC as the flow controller.
- Last Single / Last Burst source request asserted in peripheral to peripheral transfer with the DMAC as the flow controller.
- Single / Last Single / Last Burst destination request asserted in peripheral to peripheral transfer with the DMAC as the flow controller.
- Last Single / Last Burst source request asserted in peripheral to peripheral transfer with the destination peripheral as the flow controller.
- Single / Last Single / Last Burst destination request asserted in peripheral to peripheral transfer with the source peripheral as the flow controller.

DMACConfiguration: Configuration Register

The DMACConfiguration Register is read/write and configures the operation of the DMAC. The endianness of the individual AHB master interfaces can be altered by writing to the M1 and M2 bits of this register. The M1 bit enables the endianness of AHB master interface 1 to be altered. The M2 bit enables the endianness of AHB master interface 2 to be altered. The AHB master interfaces are set to little-endian mode on reset.

Note: The AHB master interfaces need not have the same endianness.

Address: 0x4020_0030



Bits	Type	Reset	Description
31-3	R	0	Not used.
2	R/W	0	**M** AHB Master endianness configuration: 0 : little-endian mode. 1 : big-endian mode. This bit is reset to 0.
1	R	0	Not used.
0	R/W	0	**E** DMAC enable: 0 : disabled. 1 : enabled. This bit is reset to 0. Disabling the DMAC reduces power consumption.

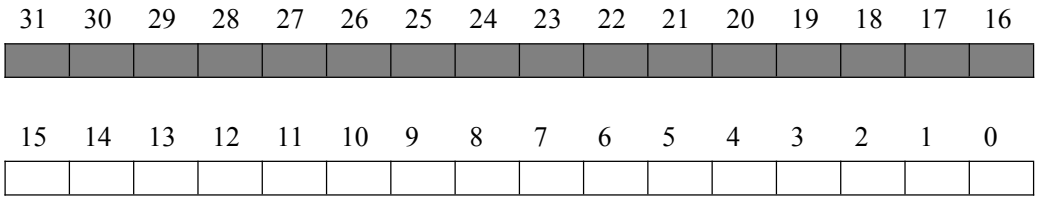
DMACSync: Synchronization Register

The DMACSync Register is read/write and is used to enable or disable synchronization logic for the DMA request signals. The DMA request signals consist of **DMACBREQ[15:0], DMACSREQ[15:0], DMACLBREQ[15:0], DMACLSREQ[15:0]** signals.

A bit set to 0 enables the synchronization logic for a particular group of DMA requests. A bit set to 1 disables the synchronization logic for a particular group of DMA requests. This register is reset to 0, synchronization logic enabled.

Note: You must use synchronization logic when the peripheral generating the DMA request runs on a different clock to the DMAC. For peripherals running on the same clock as the DMAC disabling the synchronization logic improves the DMA request response time. If necessary, the DMA response signals, **DMACLR** and **DMACTC**, must be synchronized in the peripheral.

Address: 0x4020_0034



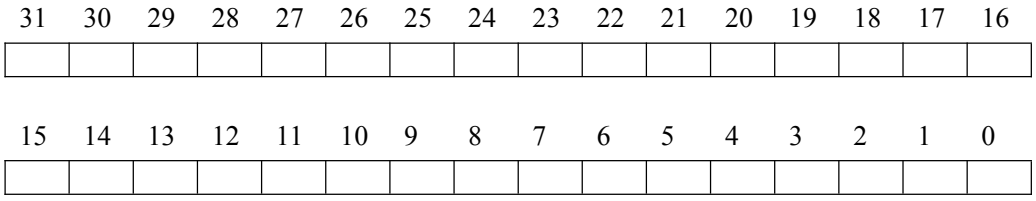
Bits	Type	Reset	Description
31-16	R	0	Not used.
15-0	R/W	0	DMA synchronization logic for DMA request signals enabled or disabled. A LOW bit indicates that the synchronization logic for the request signals is enabled. A HIGH bit indicates that the synchronization logic is disabled.

DMACC0SrcAddr: Channel Source Address Register

DMACC0SrcAddr Register contain the current source address (byte-aligned) of the data to be transferred. Each register is programmed directly by software before the appropriate channel is enabled. When the DMA channel is enabled this register is updated as the source address is incremented and by following the linked list when a complete packet of data has been transferred. Reading the register when the channel is active does not provide useful information. This is because by the time that software has processed the value read, the channel might have progressed. It is intended to be read only when the channel has stopped, in which case it shows the source address of the last item read.

Note: You must align source and destination addresses to the source and destination widths.

Address: 0x4020_0100



Bits	Type	Reset	Description
31-0	R/W	0	DMA source address.

DMACC0DestAddr: Channel Destination Address Register

DMACC0DestAddr Register contain the current destination address (byte-aligned) of the data to be transferred. Each register is programmed directly by software before the channel is enabled. When the DMA channel is enabled the register is updated as the destination address is incremented and by following the linked list when a complete packet of data has been transferred. Reading the register when the channel is active does not provide useful information. This is because by the time that software has processed the value read, the channel might have progressed. It is intended to be read only when a

channel has stopped, in which case it shows the destination address of the last item read.

Address: 0x4020_0104

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

Bits	Type	Reset	Description
31-0	R/W	0	DMA destination address.

DMACC0LLI: Channel Link List Item Register

DMACC0LLI Register contain a word-aligned address of the next LLI. If the LLI is 0, then the current LLI is the last in the chain, and the DMA channel is disabled after all DMA transfers associated with it are completed.

Note: Programming this register when the DMA channel is enabled has unpredictable side effects.

Address: 0x4020_0108

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

Bits	Type	Reset	Description
31-2	R/W	0	**LLI** Linked list item. Bits [31:2] of the address for the next LLI. Address bits [1:0] are 0.
1	R	0	Not used.
0	R/W	0	**This bit must be configured as value 1.**

Note: To make loading the LLIs more efficient for some systems, the LLI data structures can be made 4-word aligned.

DMACC0Control: Channel Control Register

The eight read/write DMACCxControl Register contain DMA channel control information such as the transfer size, burst size, and transfer width. Each register is programmed directly by software before the DMA channel is enabled. When the channel is enabled the register is updated by following the linked list when a complete packet of data has been transferred. Reading the register while the channel is active does not give

useful information. This is because by the time that software has processed the value read, the channel might have progressed. It is intended to be read only when a channel has stopped.

Address: 0x4020_010c

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

Bits	Type	Reset	Description
31	R/W	0	**I** Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt.
30-28	R/W	0	**Prot** Protection. Its details are described in the following(see protection and access information).
27	R/W	0	**DI** Destination increment. When set the destination address is incremented after each transfer.
26	R/W	0	**SI** Source increment. When set the source address is incremented after each transfer.
25	R/W	0	**D** **This bit must be configured as value 1.**
24	R/W	0	**S** **This bit must be configured as value 1.**
23-21	R/W	0	**DWidth** Destination transfer width. Transfers wider than the AHB master bus width are illegal. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data as required.
20-18	R/W	0	**SWidth** Source transfer width. Transfers wider than the AHB master bus width are illegal. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data as required.
17-15	R/W	0	**DBSize** Destination burst size. Indicates the number of transfers which make up a destination burst transfer request. This value must be set to the burst size of the destination peripheral, or if the destination is memory, to the memory boundary size. The burst size is the amount of data that is transferred when the DMACxBREQ signal goes active in the destination peripheral. The burst size is not related to the AHB HBURST signal.
14-12	R/W	0	**SBSize** Source burst size. Indicates the number of transfers which make up a source burst. This value must be set to the burst size of the source peripheral, or if the source is memory, to the memory boundary size. The burst size is the amount of

			data that is transferred when the DMACxBREQ signal goes active in the source peripheral. The burst size is not related to the AHB HBST signal.
11-0	R/W	0	**Transfer Size** Transfer size. A write to this field sets the size of the transfer when the DMAC is the flow controller. A read from this field indicates the number of transfers completed on the destination bus. Reading the register when the channel is active does not give useful information because by the time that the software has processed the value read, the channel might have progressed. It is intended to be used only when a channel is enabled and then disabled. The transfer size value is not used if the DMAC is not the flow controller.

Bit Value of SWidth or DWidth	Source or Destination Width
3'b000	Byte (8-bit)
3'b001	Halfword (16-bit)
3'b010	Word (32-bit)
3'b011	Reserved
3'b100	Reserved
3'b101	Reserved
3'b110	Reserved
3'b111	Reserved

Source or Destination Transfer Width(DWidth and SWidth)

Bit Value of DBSize or SBSize	Source or Destination Burst Transfer Request Size
3'b000	1
3'b001	4
3'b010	8
3'b011	16
3'b100	32
3'b101	64
3'b110	128
3'b111	256

Source or Destination Burst Size(DBSize and SBSize)

Protection and access information

AHB access information is provided to the source and destination peripherals when a transfer occurs. The transfer information is provided by programming the DMA channel (the Prot bit of the DMACC0Control Register, and the Lock bit of the DMACC0Configuration Register). These bits are programmed by software and peripherals can use this information if necessary. Three bits of information are provided.

Bits	Description	Purpose
0	Privileged or User	Indicates that the access is in user, or privileged mode: 0 : user mode. 1 : privileged mode.

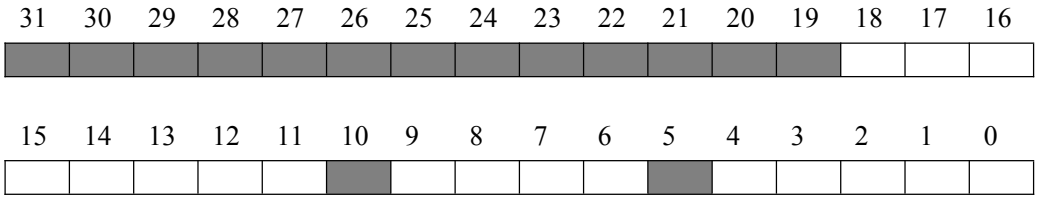
		This bit controls the AHB HPROT[1] signal.
1	Bufferable or nonbufferable	Indicates that the access is bufferable, or nonbufferable: 0： nonbufferable. 1： bufferable. This bit indicates whether or not the access is bufferable. This bit can, for example, be used to indicate to an AMBA bridge that the read can complete in zero wait states on the source bus without waiting for it to arbitrate for the destination bus and for the slave to accept the data. This bit controls the AHB HPROT[2] signal.
2	Cacheable or noncacheable	Indicates that the access is cacheable or noncacheable: 0： noncacheable. 1： cacheable. This indicates whether or not the access is cacheable. This bit can, for example, be used to indicate to an AMBA bridge that when it saw the first read of a burst of eight it can transfer the whole burst of eight reads on the destination bus, rather than pass the transactions through one at a time. This bit controls the AHB HPROT[3] signal.

Protection Bits

DMACC0Configuration: Channel Configure Register

DMACC0Configuration Register are read/write and are used to configure the DMA channel. The Register are not updated when a new LLI is requested.

Address: 0x4020_0110



Bits	Type	Reset	description
31-19	R	0	Not used.
18	R/W	0	**Halt** 0： enable DMA requests. 1： ignore further source DMA requests. The contents of the channels FIFO are drained. This value can be used with the Active and Channel Enable bits to cleanly disable a DMA channel.
17	R	0	**Active** 0： there is no data in the FIFO of the channel. 1： the FIFO of the channel has data. This value can be used with the Halt and Channel Enable bits to cleanly disable a DMA channel.
16	R/W	0	**Lock** When set this bit enables locked transfers.
15	R/W	0	**ITC** Terminal count interrupt mask. When cleared this bit masks out the terminal count interrupt of the relevant

			channel.
14	R/W	0	**IE** Interrupt error mask. When cleared this bit masks out the error interrupt of the relevant channel.
13-11	R/W	0	**FlowCntrl** Flow control and transfer type.This value indicates the flow controller and transfer type. The flow controller can be the DMAC, the source peripheral, or the destination peripheral. The transfer type can be memory to memory, memory to peripheral, peripheral to memory, or peripheral to peripheral.
10	R	0	Not used.
9-6	R/W	0	**Dest Peripheral** Destination peripheral. This value selects the DMA destination request peripheral. This field is ignored if the destination of the transfer is to memory.
5	R	0	Not used.
4-1	R/W	0	**Source Peripheral** Source peripheral. This value selects the DMA source request peripheral. This field is ignored if the source of the transfer is from memory.
0	R/W	0	**E** Channel enable. Reading this bit indicates whether a channel is currently enabled or disabled: 0 : channel disabled. 1 : channel enabled. The Channel Enable bit status can also be found by reading the DMACEnbldChns register. A channel is enabled by setting this bit. A channel can be disabled by clearing the Enable bit.This causes the current AHB transfer (if one is in progress) to complete and the channel is then disabled. Any data in the channel's FIFO is lost. Restarting the channel by simply setting the Channel Enable bit has unpredictable effects and the channel must be fully re-initialized. The channel is also disabled, and Channel Enable bit cleared, when the last LLI is reached or if a channel error is encountered. If a channel has to be disabled without losing data in a channel's FIFO the Halt bit must be set so that further DMA requests are ignored. The Active bit must then be polled until it reaches 0, indicating that there is no data left in the channels FIFO. Finally the Channel Enable bit can be cleared.

Bit Value	Transfer Type	Controller
000	Memory to memory	DMAC
001	Memory to peripheral	DMAC
010	Peripheral to memory	DMAC
011	Source peripheral to destination peripheral	DMAC
100	Source peripheral to destination peripheral	Destination peripheral
101	Memory to peripheral	Peripheral

110	Peripheral to memory	Peripheral
111	Source peripheral to destination peripheral	Source peripheral

Flow Control and Transfer Type Bits(FlowCntrl)

DMACC1SrcAddr: Channel Source Address Register

The same like register DMACC0SrcAddr

Address: 0x4020_0120

DMACC1DestAddr: Channel Destination Address Register

The same like register DMACC0DestAddr

Address: 0x4020_0124

DMACC1LLI: Channel Link List Item Register

The same like register DMACC0LLI

Address: 0x4020_0128

DMACC1Control: Channel Control Register

The same like register DMACC0Control

Address: 0x4020_012c

DMACC1Configuration: Channel Configure Register

The same like register DMACC0Configuration

Address: 0x4020_0130

DMACC2SrcAddr: Channel Source Address Register

The same like register DMACC0SrcAddr

Address: 0x4020_0140

DMACC2DestAddr: Channel Destination Address Register

The same like register DMACC0DestAddr

Address: 0x4020_0144

DMACC2LLI: Channel Link List Item Register

The same like register DMACC0LLI

Address: 0x4020_0148

DMACC2Control: Channel Control Register

The same like register DMACC0Control

Address: 0x4020_014c

DMACC2Configuration: Channel Configure Register

The same like register DMACC0Configuration

Address: 0x4020_0150

DMACC3SrcAddr: Channel Source Address Register

The same like register DMACC0SrcAddr

Address: 0x4020_0160

DMACC3DestAddr: Channel Destination Address Register

The same like register DMACC0DestAddr

Address: 0x4020_0164

DMACC3LLI: Channel Link List Item Register

The same like register DMACC0LLI

Address: 0x4020_0168

DMACC3Control: Channel Control Register

The same like register DMACC0Control

Address: 0x4020_016c

DMACC3Configuration: Channel Configure Register

The same like register DMACC0Configuration

Address: 0x4020_0170

DMACC4SrcAddr: Channel Source Address Register

The same like register DMACC0SrcAddr

Address: 0x4020_0180

DMACC4DestAddr: Channel Destination Address Register

The same like register DMACC0DestAddr

Address: 0x4020_0184

DMACC4LLI: Channel Link List Item Register

The same like register DMACC0LLI

Address: 0x4020_0188

DMACC4Control: Channel Control Register

The same like register DMACC0Control

Address: 0x4020_018c

DMACC4Configuration: Channel Configure Register

The same like register DMACC0Configuration

Address: 0x4020_0190

DMACC5SrcAddr: Channel Source Address Register

The same like register DMACC0SrcAddr

Address: 0x4020_01a0

DMACC5DestAddr: Channel Destination Address Register

The same like register DMACC0DestAddr

Address: 0x4020_01a4

DMACC5LLI: Channel Link List Item Register

The same like register DMACC0LLI

Address: 0x4020_01a8

DMACC5Control: Channel Control Register

The same like register DMACC0Control

Address: 0x4020_01ac

DMACC5Configuration: Channel Configure Register

The same like register DMACC0Configuration

Address: 0x4020_01b0

DMACC6SrcAddr: Channel Source Address Register

The same like register DMACC0SrcAddr

Address: 0x4020_01c0

DMACC6DestAddr: Channel Destination Address Register

The same like register DMACC0DestAddr

Address: 0x4020_01c4

DMACC6LLI: Channel Link List Item Register

The same like register DMACC0LLI

Address: 0x4020_01c8

DMACC6Control: Channel Control Register

The same like register DMACC0Control

Address: 0x4020_01cc

DMACC6Configuration: Channel Configure Register

The same like register DMACC0Configuration

Address: 0x4020_01d0

DMACC7SrcAddr: Channel Source Address Register

The same like register DMACC0SrcAddr

Address: 0x4020_01e0

DMACC7DestAddr: Channel Destination Address Register

The same like register DMACC0DestAddr

Address: 0x4020_01e4

DMACC7LLI: Channel Link List Item Register

The same like register DMACC0LLI

Address: 0x4020_01e8

DMACC7Control: Channel Control Register

The same like register DMACC0Control

Address: 0x4020_01ec

DMACC7Configuration: Channel Configure Register

The same like register DMACC0Configuration

Address: 0x4020_01f0

23. SSI Controller

23.1SSI Controller Features

1) Serial-master or serial-slave operation – Enables serial communication with serial-master or serial-slave peripheral devices.
2)Configurable and programmable Dual SPI and Quad SPI support in Master Mode for dual/quad SPI transfers.
3) Independent masking of interrupts.
4) Multi-master contention detection – Informs the processor of multiple serial-master accesses on the serial bus.
5)Programmable delay on the sample time of the received serial data bit (rxd), when configured in Master Mode; enables programmable control of routing delays resulting in higher serial data-bit rates.
6) Programmable features:
Serial interface operation – Choice of Motorola SPI, Texas Instruments Synchronous Serial

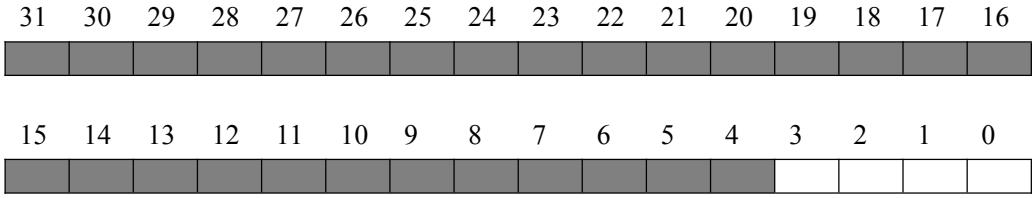
Protocol(SSP) or National Semiconductor Microwire.
Clock bit-rate – Dynamic control of the serial bit rate of the data transfer; used in only serial-master mode of operation.
Data Item size (4 to 32 bits) – Item size of each data transfer under the control of the programmer.

23.2 Register Descriptions

SSI0_BASE = 0x40D0_0000
SSI1_BASE = 0x40D8_0000

23.2.1 CTRLR0: Control Register 0

Address: SSI_BASE + 0x0



Bit	Type	Reset	Description
31-24	R	0	reserved
23-22	R/W	0	**SPI_FRF** SPI Frame Format: Selects data frame format for transmitting or receiving data. 00 – Standard SPI Format 01 – Dual SPI Format 10 – Quad SPI Format 11 – Reserved
21-20	R	0	**reserved**
19-16	R/W	0	**CFS** Control Frame Size. Selects the length of the control word for the Microwire frame format.
15	R	0	**Reserved**
14	R/W	0	**SSTE** Slave Select Toggle Enable. While operating in SPI mode with clock phase (SCPH) set to 0, this register controls the behavior of the slave select line (ss_n) between data frames. 0x1 (TOGGLE_EN): ss_n line toggles between consecutive data frames, with the serial clock (sclk) being held to its default value while ss_n is high 0x0 (TOGGLE_DISABLE): ss_n stays low and sclk
13	R/W	0	**SRL** Shift Register Loop. Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input. Can be used in both serial-slave and serial-master modes. 0 – Normal Mode Operation 1 – Test Mode Operation

			When the ssi is configured as a slave in loopback mode, the ss_in_n and ssi_clk signals must be provided by an external source. In this mode, the slave cannot generate these signals because there is nothing to which to loop back.
12	R/W	0	**SLV_OE** WSlave Output Enable. Relevant only when the ssi is configured as a serial-slave device. When configured as a serial master, this bit field has no functionality. This bit enables or disables the setting of the ssi_oe_n output from the ssi serial slave. When SLV_OE = 1, the ssi_oe_n output can never be active. When the ssi_oe_n output controls the tri-state buffer on the txd output from the slave, a high impedance state is always present on the slave txd output when SLV_OE = 1. This is useful when the master transmits in broadcast mode (master transmits data to all slave devices). Only one slave may respond with data on the master rxd line. This bit is enabled after reset and must be disabled by software (when broadcast mode is used), if you do not want this device to respond with data. 0 –- Slave txd is enabled 1 –- Slave txd is disabled
11-10	R/W	0	**TMOD** Transfer Mode. Selects the mode of transfer for serial communication. This field does not affect the transfer duplicity. Only indicates whether the receive or transmit data are valid. In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer. In receive-only mode, transmitted data are not valid. After the first write to the transmit FIFO, the same word is retransmitted for the duration of the transfer. In transmit-and-receive mode, both transmit and receive data are valid. The transfer continues until the transmit FIFO is empty. Data received from the external device are stored into the receive FIFO memory, where it can be accessed by the host processor. In eeprom-read mode, receive data is not valid while control data is being transmitted. When all control data is sent to the EEPROM, receive data becomes valid and transmit data becomes invalid. All data in the transmit FIFO is considered control data in this mode. This transfer mode is only valid when the ssi is configured as a master device. 00 –- Transmit & Receive 01 –- Transmit Only 10 –- Receive Only 11 –- EEPROM Read When SSI_SPI_MODE is set to "Dual" or "Quad" mode and SPI_FRF is not set to 2'b00. There are only two valid combinations: 10 –- Read 01 –- Write
9	R/W	0	**SCPOL**

			Serial Clock Polarity. Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the ssi master is not actively transferring data on the serial bus. 0 – Inactive state of serial clock is low 1 – Inactive state of serial clock is high
8	R/W	0	**SCPH** Serial Clock Phase. Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data are captured on the first edge of the serial clock. When SCPH = 1, the serial clock starts toggling one cycle after the slave select line is activated, and data are captured on the second edge of the serial clock. 0: Serial clock toggles in middle of first data bit 1: Serial clock toggles at start of first data bit
7-6	R/W	0	**FRF** Frame Format. Selects which serial protocol transfers the data. 00 –- Motorola SPI 01 –- Texas Instruments SSP 10 –- National Semiconductors Microwire 11 –- Reserved
5	R	0	**Reserved**
4-0	R/W	0x7	**DFS** Data Frame Size. This register field is only valid when SSI_MAX_XFER_SIZE is configured to 16. If SSI_MAX_XFER_SIZE is configured to 32, then writing to this field will not have any effect. Selects the data frame length. When the data frame size is programmed to be less than 16 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. **Note:** When SSI_SPI_MODE is set to "Dual" or "Quad" mode and SPI_FRF is not set to 2'b00. DFS value should be multiple of 2 if SPI_FRF = 01, and DFS value should be multiple of 4 if SPI_FRF = 10.

DFS Value Description

0000 Reserved – undefined operation
0001 Reserved – undefined operation
0010 Reserved – undefined operation
0011 4-bit serial data transfer
0100 5-bit serial data transfer
0101 6-bit serial data transfer
0110 7-bit serial data transfer
0111 8-bit serial data transfer

1000	9-bit serial data transfer
1001	10-bit serial data transfer
1010	11-bit serial data transfer
1011	12-bit serial data transfer
1100	13-bit serial data transfer
1101	14-bit serial data transfer
1110	15-bit serial data transfer
1111	16-bit serial data transfer

DFS_32 Value Description

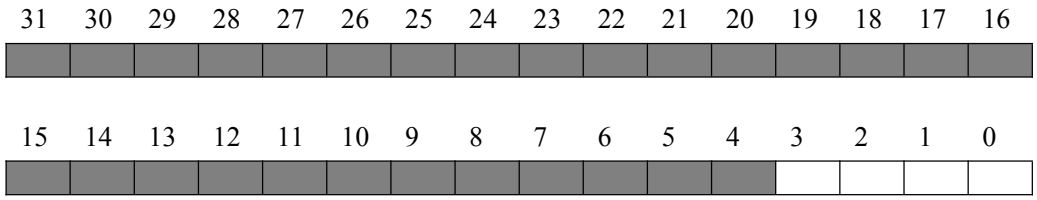
00000	Reserved
00001	Reserved
00010	Reserved
00011	4-bit serial data transfer
00100	5-bit serial data transfer
00101	6-bit serial data transfer
00110	7-bit serial data transfer
00111	8-bit serial data transfer
01000	9-bit serial data transfer
01001	10-bit serial data transfer
01010	11-bit serial data transfer
01011	12-bit serial data transfer
01100	13-bit serial data transfer
01101	14-bit serial data transfer
01110	15-bit serial data transfer
01111	16-bit serial data transfer
10000	17-bit serial data transfer
10001	18-bit serial data transfer
10010	19-bit serial data transfer
10011	20-bit serial data transfer
10100	21-bit serial data transfer
10101	22-bit serial data transfer
10110	23-bit serial data transfer
10111	24-bit serial data transfer
11000	25-bit serial data transfer
11001	26-bit serial data transfer
11010	27-bit serial data transfer
11011	28-bit serial data transfer
11100	29-bit serial data transfer
11101	30-bit serial data transfer
11110	31-bit serial data transfer
11111	32-bit serial data transfer

CFS Value Description

0000 1-bit control word
0001 2-bit control word
0010 3-bit control word
0011 4-bit control word
0100 5-bit control word
0101 6-bit control word
0110 7-bit control word
0111 8-bit control word
1000 9-bit control word
1001 10-bit control word
1010 11-bit control word
1011 12-bit control word
1100 13-bit control word
1101 14-bit control word
1110 15-bit control word
1111 16-bit control word

23.2.2 CTRLR1: Control Register 1

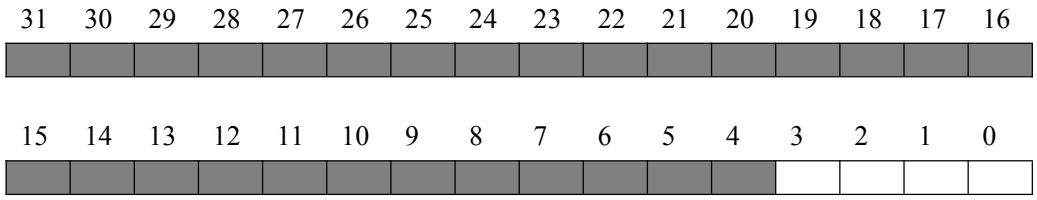
Address: SSI_BASE + 0x04



Bit	Type	Reset	Description
31-16	R	0	reserved
15-0	R/W	0	**NDF** Number of Data Frames. When TMOD = 10 or TMOD = 11, this register field sets the number of data frames to be continuously received by the ssi. The ssi continues to receive serial data until the number of data frames received is equal to this register value plus 1, which enables you to receive up to 64 KB of data in a continuous transfer. When the ssi is configured as a serial slave, the transfer continues for as long as the slave is selected. Therefore, this register serves no purpose and is not present when the ssi is configured as a serial slave.

23.2.3 SSIENR: SSI Enable Register

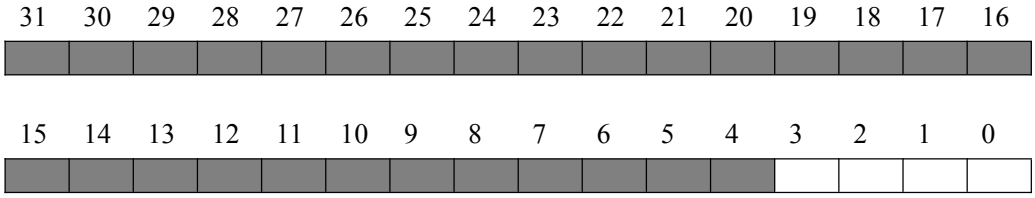
Address: SSI_BASE + 0x08



Bit	Type	Reset	Description
31-1	R	0	reserved
0	R/W	0	**SSI_EN** SSI Enable. Enables and disables all ssi operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is impossible to program some of the ssi control Register when enabled. When disabled, the ssi_sleep output is set (after delay) to inform the system that it is safe to remove the ssi_clk, thus saving power consumption in the system.

23.2.4 MWCR: Microwire Control Register

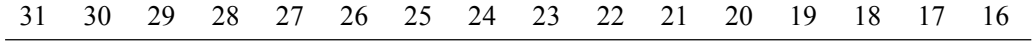
Address: SSI_BASE + 0x0C

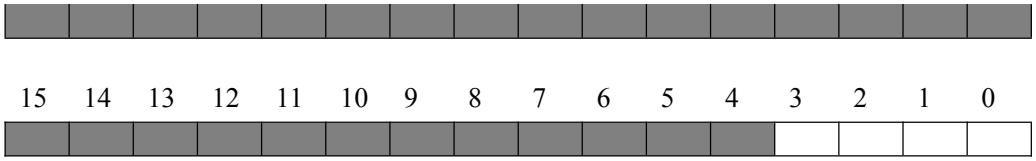


Bit	Type	Reset	Description
31-3	R	0	reserved
2	R/W	0	**MHS** Microwire Handshaking. Relevant only when the ssi is configured as a serial-master device. When configured as a serial slave, this bit field has no functionality. Used to enable and disable the “busy/ready” handshaking interface for the Microwire protocol. When enabled, the ssi checks for a ready status from the target slave, after the transfer of the last data/control bit, before clearing the BUSY status in the SR register. 0: handshaking interface is disabled 1: handshaking interface is enabled
1	R/W	0	**MDD** Microwire Control. Defines the direction of the data word when the Microwire serial protocol is used. When this bit is set to 0, the data word is received by the ssi MacroCell from the external serial device. When this bit is set to 1, the data word is transmitted from the ssi MacroCell to the external serial device.
0	R/W	0	**MWMOD** Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. 0 – non-sequential transfer 1 – sequential transfer

23.2.5 SER: Slave Enable Register

Address: SSI_BASE + 0x10

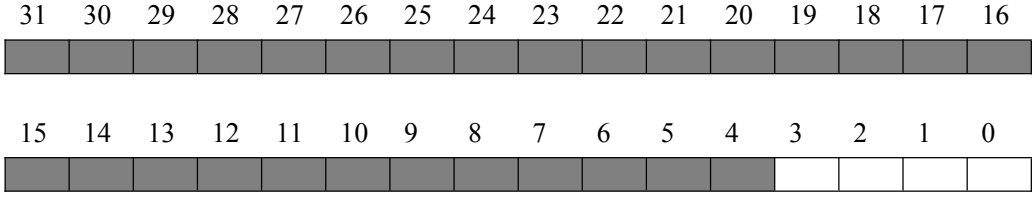




Bit	Type	Reset	Description
31-1	R	0	reserved
0	R/W	0	Slave Select Enable Flag. Each bit in this register corresponds to a slave select line (ss_x_n) from the ssi master. When a bit in this register is set (1), the corresponding slave select line from the master is activated when a serial transfer begins. It should be noted that setting or clearing bits in this register have no effect on the corresponding slave select outputs until a transfer is started. Before beginning a transfer, you should enable the bit in this register that corresponds to the slave device with which the master wants to communicate. When not operating in broadcast mode, only one bit in this field should be set. 1: Selected 0: Not Selected

23.2.6 BAUDR: Baud Rate Select

Address: SSI_BASE + 0x14

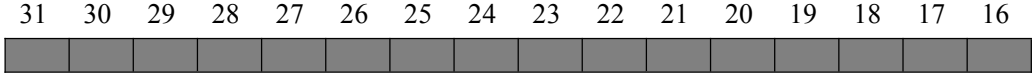


Bit	Type	Reset	Description
31-16	R	0	reserved
15-0	R/W	0	**SCKDV** SSI Clock Divider. The LSB for this field is always set to 0 and is unaffected by a write operation, which ensures an even value is held in this register. If the value is 0, the serial output clock (sclk_out) is disabled. The frequency of the sclk_out is derived from the following equation: **Fsclk_out = Fssi_clk/SCKDV** where SCKDV is any even value between 2 and 65534.2. **For example:** for Fssi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2 = 1.8432MHz

23.2.7 TXFTLR: Transmit FIFO Threshold Level

This register controls the threshold value for the transmit FIFO memory

Address: SSI_BASE + 0x18



Bit	Type	Reset	Description
31-8	R	0	reserved
7-0	R/W	0	**TFT** Transmit FIFO Threshold. Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256; this register is sized to the number of address bits needed to access the FIFO. If you attempt to set bits [7:0] of this register to a value greater than or equal to the depth of the FIFO, this field is not written and retains its current value. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.

TFT Value Description

0000_0000 ssi_txe_intr is asserted when 0 data entries are present in transmit FIFO

0000_0001 ssi_txe_intr is asserted when 1 or less data entry is present in transmit FIFO

0000_0010 ssi_txe_intr is asserted when 2 or less data entries are present in transmit FIFO

0000_0011 ssi_txe_intr is asserted when 3 or less data entries are present in transmit FIFO

::

::

1111_1100 ssi_txe_intr is asserted when 252 or less data entries are present in transmit FIFO

1111_1101 ssi_txe_intr is asserted when 253 or less data entries are present in transmit FIFO

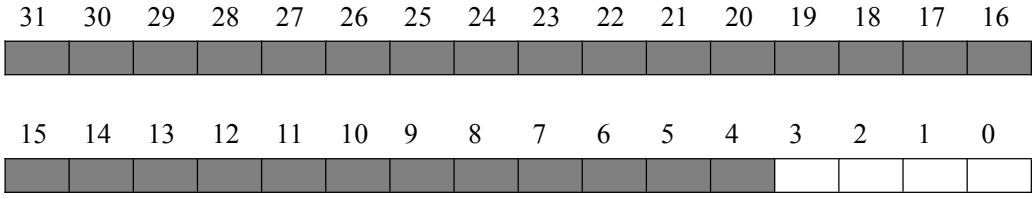
1111_1110 ssi_txe_intr is asserted when 254 or less data entries are present in transmit FIFO

1111_1111 ssi_txe_intr is asserted when 255 or less data entries are present in transmit FIFO

23.2.8 RXFTLR: Receive FIFO Threshold Level

This register controls the threshold value for the receive FIFO memory

Address: SSI_BASE + 0x1C



Bit	Type	Reset	Description
31-8	R	0	reserved
7-0	R/W	0	**RFT** Receive FIFO Threshold. Controls the level of entries (or above) at which the receive FIFO controller triggers an interrupt. The FIFO depth is configurable in the range 2-256. This register is sized to the number of address bits needed to access the FIFO. If you attempt to set this value greater than the depth of the FIFO, this field is not written and retains its current value.

			When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.

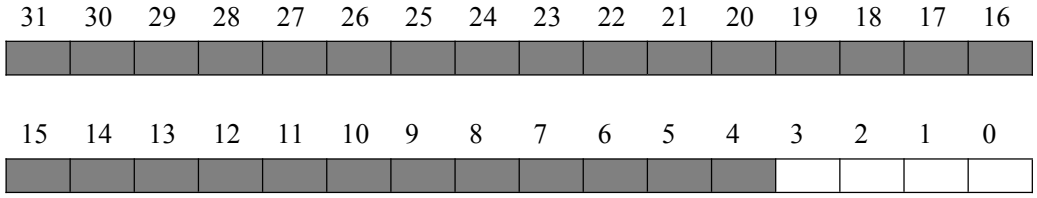
RFT Value Description

0000_0000 ssi_rxf_intr is asserted when 1 or more data entry is present in receive FIFO
0000_0001 ssi_rxf_intr is asserted when 2 or more data entries are present in receive FIFO
0000_0010 ssi_rxf_intr is asserted when 3 or more data entries are present in receive FIFO
0000_0011 ssi_rxf_intr is asserted when 4 or more data entries are present in receive FIFO
::
::
1111_1100 ssi_rxf_intr is asserted when 253 or more data entries are present in receive FIFO
1111_1101 ssi_rxf_intr is asserted when 254 or more data entries are present in receive FIFO
1111_1110 ssi_rxf_intr is asserted when 255 or more data entries are present in receive FIFO
1111_1111 ssi_rxf_intr is asserted when 256 data entries are present in receive FIFO

23.2.9 TXFLR: Transmit FIFO Level Register

This register contains the number of valid data entries in the transmit FIFO memory.

Address: SSI_BASE + 0x20

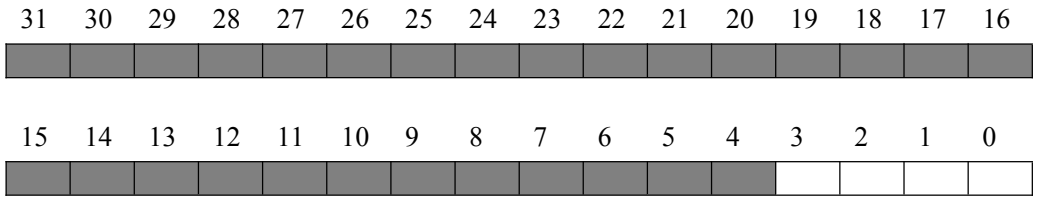


Bit	Type	Reset	Description
31-0	R	0	**TXTFL** Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO.

23.2.10 RXFLR: Receive FIFO Level Register

This register contains the number of valid data entries in the receive FIFO memory.

Address: SSI_BASE + 0x24

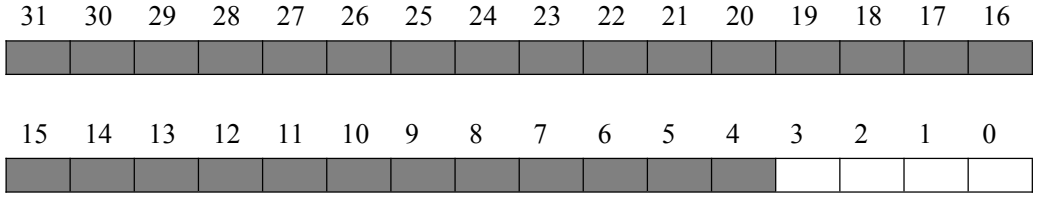


Bit	Type	Reset	Description
31-0	R	0	**RXTFL** Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.

23.2.11 SR: Status Register

This is a read-only register used to indicate the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The status register may be read at any time. None of
the bits in this register request an interrupt.

Address: SSI_BASE + 0x28

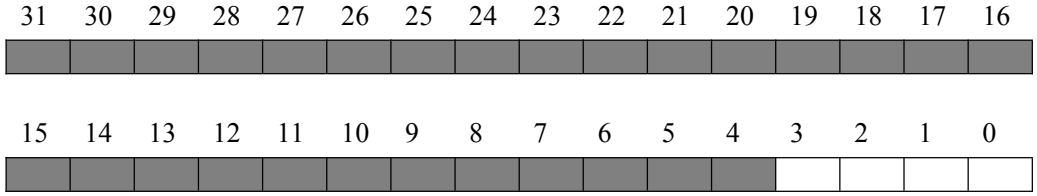


Bit	Type	Reset	Description
31-7	R	0	**reserved**
6	R	0	**DCOL** Data Collision Error. Relevant only when the ssi is configured as a master device. This bit is set if the ss_in_n input is asserted by another master,while the SSI master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read. 0 – No error 1 – Transmit data collision error
5	R	0	**TXE** Transmission Error. Set if the transmit FIFO is empty when a transfer is started. This bit can be set only when the ssi is configured as a slave device. Data from the previous transmission is resent on the txd line. This bit is cleared when read. 0 – No error 1 – Transmission error
4	R	0	**RFF** Receive FIFO Full. When the receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared. 0 – Receive FIFO is not full 1 – Receive FIFO is full
3	R	0	**RFNE** Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO. 0 – Receive FIFO is empty 1 – Receive FIFO is not empty
2	R	1	**TFE** Transmit FIFO Empty. When the transmit FIFO is completely empty, this bit is set. When the transmit FIFO contains one or more valid entries, this bit is cleared. This bit field does not request an interrupt. 0 – Transmit FIFO is not empty 1 – Transmit FIFO is empty
1	R	1	**TFNF** Transmit FIFO Not Full. Set when the transmit FIFO

			contains one or more empty locations, and is cleared when the FIFO is full. 0 – Transmit FIFO is full 1 – Transmit FIFO is not full
0	R	0	**BUSY** SSI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the SSI is idle or disabled. 0 – ssi is idle or disabled 1 – ssi is actively transferring data

23.2.12 IMR: Interrupt Mask Register

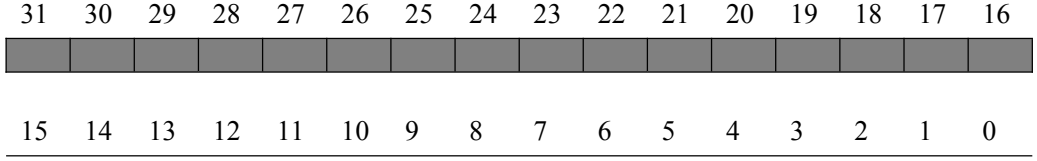
Address: SSI_BASE + 0x2C



Bit	Type	Reset	Description
31-7	R	0	**reserved**
5	R/W	1	**MSTIM** Multi-Master Contention Interrupt Mask. This bit field is not present if the ssi is configured as a serial-slave device. 0 – ssi_mst_intr interrupt is masked 1 – ssi_mst_intr interrupt is not masked
4	R/W	1	**RXFIM** Receive FIFO Full Interrupt Mask 0 – ssi_rxf_intr interrupt is masked 1 – ssi_rxf_intr interrupt is not masked
3	R/W	1	**RXOIM** Receive FIFO Overflow Interrupt Mask 0 – ssi_rxo_intr interrupt is masked 1 – ssi_rxo_intr interrupt is not masked
2	R/W	1	**RXUIM** Receive FIFO Underflow Interrupt Mask 0 – ssi_rxu_intr interrupt is masked 1 – ssi_rxu_intr interrupt is not masked
1	R/W	1	**TXOIM** Transmit FIFO Overflow Interrupt Mask 0 – ssi_txo_intr interrupt is masked 1 – ssi_txo_intr interrupt is not masked
0	R/W	1	**TXEIM** Transmit FIFO Empty Interrupt Mask 0 – ssi_txe_intr interrupt is masked 1 – ssi_txe_intr interrupt is not masked

23.2.13 ISR: Interrupt Status Register

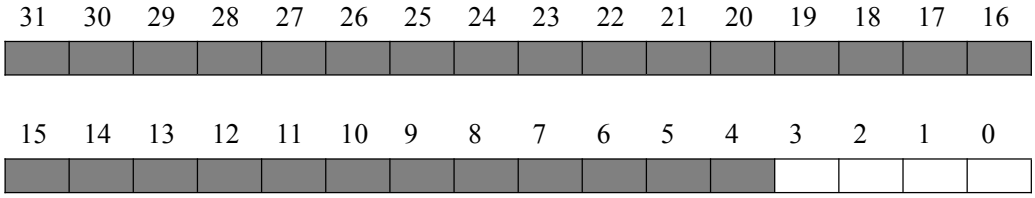
Address: SSI_BASE + 0x30



Bit	Type	Reset	Description
31-7	R	0	**reserved**
5	R	0	**MSTIS** Multi-Master Contention Interrupt Status. This bit field is not present if the ssi is configured as a serial-slave device. 0 = ssi_mst_intr interrupt not active after masking 1 = ssi_mst_intr interrupt is active after masking
4	R	0	**RXFIS** Receive FIFO Full Interrupt Status 0 = ssi_rxf_intr interrupt is not active after masking 1 = ssi_rxf_intr interrupt is full after masking
3	R	0	**RXOIS** Receive FIFO Overflow Interrupt Status 0 = ssi_rxo_intr interrupt is not active after masking 1 = ssi_rxo_intr interrupt is active after masking
2	R	0	**RXUIS** Receive FIFO Underflow Interrupt Status 0 = ssi_rxu_intr interrupt is not active after masking 1 = ssi_rxu_intr interrupt is active after masking
1	R	0	**TXOIS** Transmit FIFO Overflow Interrupt Status 0 = ssi_txo_intr interrupt is not active after masking 1 = ssi_txo_intr interrupt is active after masking
0	R	0	**TXEIS** Transmit FIFO Empty Interrupt Status 0 = ssi_txe_intr interrupt is not active after masking 1 = ssi_txe_intr interrupt is active after masking

23.2.14 RISR: Raw Interrupt Status Register

Address: SSI_BASE + 0x34

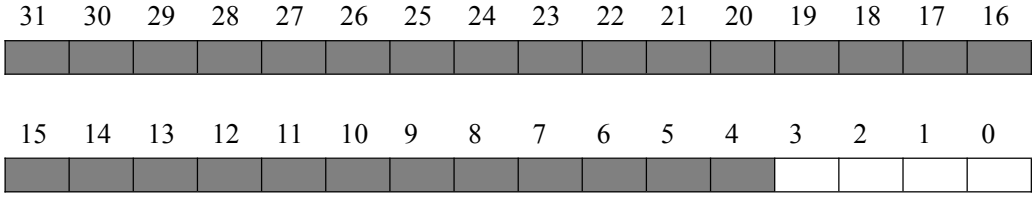


Bit	Type	Reset	Description
31-7	R	0	**reserved**
5	R	0	**MSTIR** Multi-Master Contention Raw Interrupt Status. This bit field is not present if the ssi is configured as a serial-slave device. 0 = ssi_mst_intr interrupt is not active prior to masking 1 = ssi_mst_intr interrupt is active prior masking
4	R	0	**RXFIR** Receive FIFO Full Raw Interrupt Status 0 = ssi_rxf_intr interrupt is not active prior to masking 1 = ssi_rxf_intr interrupt is active prior to masking
3	R	0	**RXOIR** Receive FIFO Overflow Raw Interrupt Status

			0 = ssi_rxo_intr interrupt is not active prior to masking 1 = ssi_rxo_intr interrupt is active prior masking
2	R	0	**RXUIR** Receive FIFO Underflow Raw Interrupt Status 0 = ssi_rxu_intr interrupt is not active prior to masking 1 = ssi_rxu_intr interrupt is active prior to masking
1	R	0	**TXOIR** Transmit FIFO Overflow Raw Interrupt Status 0 = ssi_txo_intr interrupt is not active prior to masking 1 = ssi_txo_intr interrupt is active prior masking
0	R	0	**TXEIR** Transmit FIFO Empty Raw Interrupt Status 0 = ssi_txe_intr interrupt is not active prior to masking 1 = ssi_txe_intr interrupt is active prior masking

23.2.15 TXOICR: Transmit FIFO Overflow Interrupt Clear Register

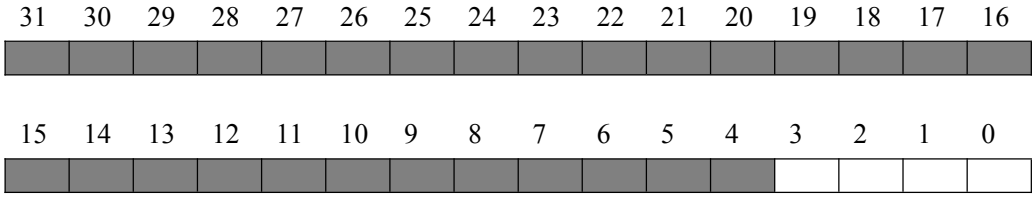
Address: SSI_BASE + 0x38



Bit	Type	Reset	Description
31-1	R	0	**reserved**
0	R	0	Clear Transmit FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect.

23.2.16 RXOICR: Receive FIFO Overflow Interrupt Clear Register

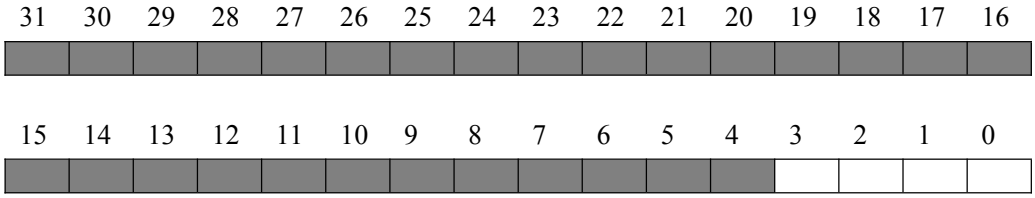
Address: SSI_BASE + 0x3C



Bit	Type	Reset	Description
31-1	R	0	**reserved**
0	R	0	Clear Receive FIFO Overflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.

23.2.17 RXUICR: Receive FIFO Underflow Interrupt Clear Register

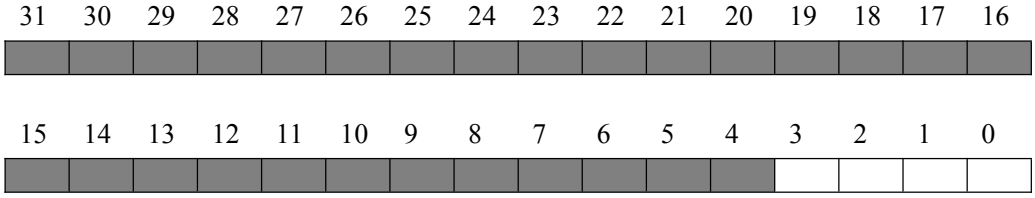
Address: SSI_BASE + 0x40



Bit	Type	Reset	Description
31-1	R	0	**reserved**
0	R	0	Clear Receive FIFO Underflow Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.

23.2.18 MSTICR: Multi-Master Interrupt Clear Register

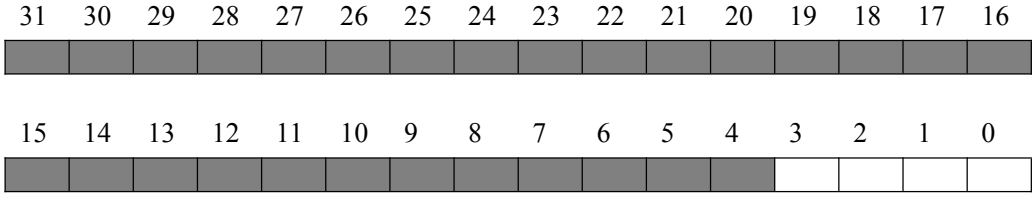
Address: SSI_BASE + 0x44



Bit	Type	Reset	Description
31-1	R	0	**reserved**
0	R	0	Clear Multi-Master Contention Interrupt. This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.

23.2.19 ICR: Interrupt Clear Register

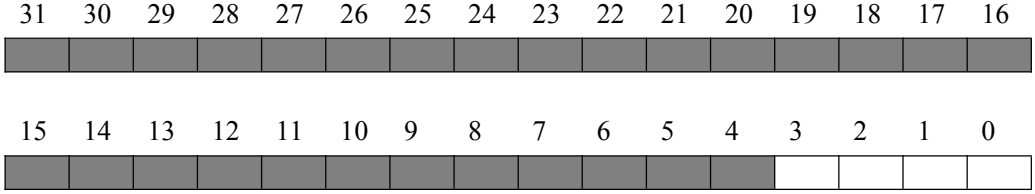
Address: SSI_BASE + 0x48



Bit	Type	Reset	Description
31-1	R	0	**reserved**
0	R	0	Clear Interrupts. This register is set if any of the interrupts below are active. A read clears the ssi_txo_intr, ssi_rxu_intr, ssi_rxo_intr, and the ssi_mst_intr interrupts. Writing to this register has no effect.

23.2.20 DMACR: DMA Control Register

Address: SSI_BASE + 0x4C

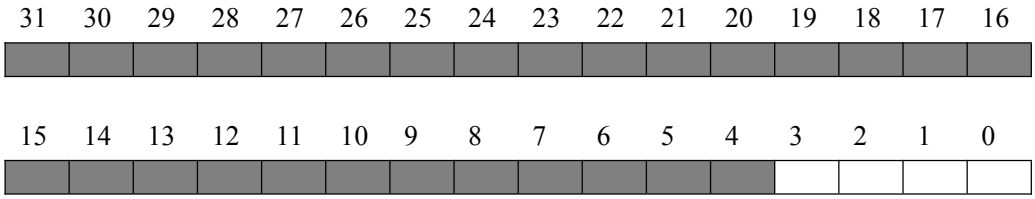


Bit	Type	Reset	Description
31-2	R	0	**reserved**
1	R/W	0	**TDMAE** Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled

			1 = Transmit DMA enabled
0	R/W	0	**RDMAE** Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel 0 = Receive DMA disabled 1 = Receive DMA enabled

23.2.21 DMATDLR: DMA Transmit Data Level

Address: SSI_BASE + 0x50



Bit	Type	Reset	Description
31-8	R	0	**reserved**
7-0	R/W	0	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1.

DMATDL Value Description

0000_0000 dma_tx_req is asserted when 0 data entries are present in the transmit FIFO

0000_0001 dma_tx_req is asserted when 1 or less data entry is present in the transmit FIFO

0000_0010 dma_tx_req is asserted when 2 or less data entries are present in the transmit FIFO

0000_0011 dma_tx_req is asserted when 3 or less data entries are present in the transmit FIFO

::

::

1111_1100 dma_tx_req is asserted when 252 or less data entries are present in the transmit FIFO

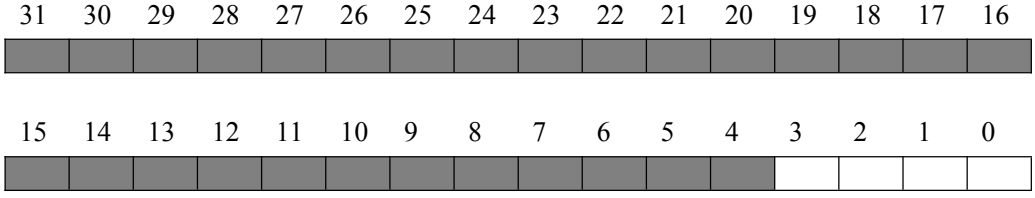
1111_1101 dma_tx_req is asserted when 253 or less data entries are present in the transmit FIFO

1111_1110 dma_tx_req is asserted when 254 or less data entries are present in the transmit FIFO

1111_1111 dma_tx_req is asserted when 255 or less data entries are present in the transmit FIFO

23.2.22 DMARDLR: DMA Receive Data Level

Address: SSI_BASE + 0x54



Bit	Type	Reset	Description
31-8	R	0	**reserved**

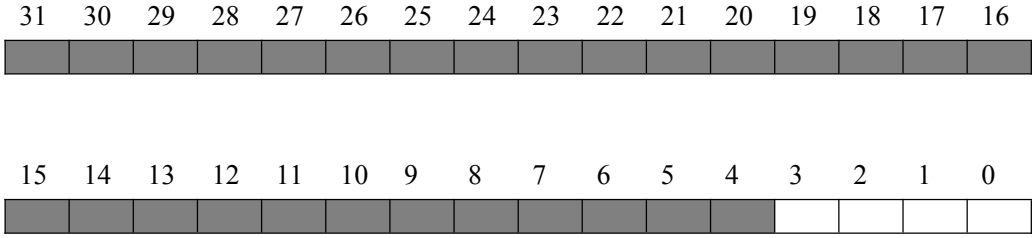
7-0	R/W	0	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or above this field value + 1, and RDMAE=1.

DMARDL Value Description

0000_0000 dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO
0000_0001 dma_rx_req is asserted when 2 or more data entries are present in the receive FIFO
0000_0010 dma_rx_req is asserted when 3 or more data entries are present in the receive FIFO
0000_0011 dma_rx_req is asserted when 4 or more data entries are present in the receive FIFO
::
::
1111_1100 dma_rx_req is asserted when 253 or more data entries are present in the receive FIFO
1111_1101 dma_rx_req is asserted when 254 or more data entries are present in the receive FIFO
1111_1110 dma_rx_req is asserted when 255 or more data entries are present in the receive FIFO
1111_1111 dma_rx_req is asserted when 256 data entries are present in the receive FIFO

23.2.23 DR: Data Register

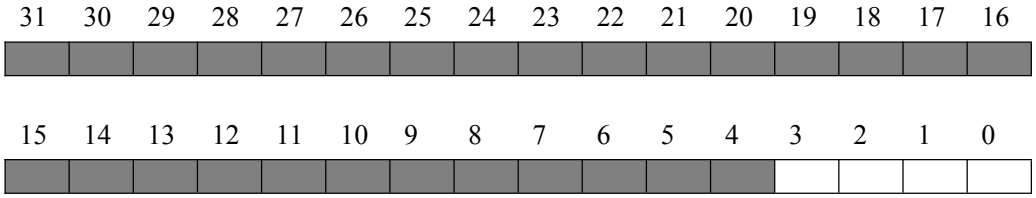
Address: SSI_BASE + 0x60



Bit	Type	Reset	Description
31-0	R/W	0	Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. If SSI_MAX_XFER_SIZE configuration parameter is set to 32, all 32 bits are valid. Otherwise, only 16 bits ([15:0]) of the register are valid. Read = Receive FIFO buffer Write = Transmit FIFO buffer

23.2.24 RX_SAMPLE_DLY: Rx Sample Delay Register

Address: SSI_BASE + 0xF0



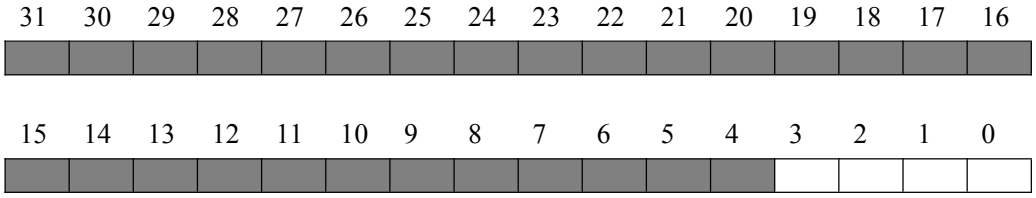
Bit	Type	Reset	Description
31-8	R	0	**Reserved**
7-0	R/W	0	Receive Data (rxd) Sample Delay. This register is used to delay the sample of the rxd input signal. Each value represents a single ssi_clk delay on the sample of the rxd

			signal. **NOTE:** If this register is programmed with a value that exceeds the depth of the internal shift Register (SSI_RX_DLY_SR_DEPTH), a zero (0) delay will be applied to the rxd sample.

23.2.25 SPI_CTRLR0: SPI Control Register

This register is used to control the serial data transfer in SPI mode of operation. The register is relevant only when SPI_FRF (in CTRLR0) is set to either 01 or 10. It is not possible to write to this register when the ssi is enabled (SSI_EN=1).

Address: SSI_BASE + 0xF4



Bit	Type	Reset	Description
31-16	R	0	**Reserved**
15-11	R/W	0	**WAIT_CYCLES** This bit defines the wait cycles in dual/quad mode between control frames transmit and data reception. Specified as number of SPI clock cycles
10	R	0	**Reserved**
9-8	R/W	2	**INST_L** Dual/Quad mode instruction length in bits. 00 - 0 bit (No instruction) 01 - 4 bits 10 - 8 bits 11 - 16 bits
7-6	R	0	**Reserved**
5-2	R/W	0	**ADDR_L** This bit defines length of address to be transmitted. 0000 0-bit Address Width 0001 4-bit Address Width 0010 8-bit Address Width 0011 12-bit Address Width 0100 16-bit Address Width 0101 20-bit Address Width 0110 24-bit Address Width 0111 28-bit Address Width 1000 32-bit Address Width 1001 36-bit Address Width 1010 40-bit Address Width 1011 44-bit Address Width 1100 48-bit Address Width 1101 52-bit Address Width 1110 56-bit Address Width 1111 60-bit Address Width
1-0	R/W	0	**TRANS_TYPE** Address and instruction transfer format. This bit selects whether ssi will transmit

			instruction/address either in Standard SPI mode or the SPI mode when the mode is specified in the CTRLR0.SPI_FRF field. 00 - Instruction and Address will be sent in Standard SPI Mode. 01 - Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by CTRLR0.SPI_FRF. 10 - Both Instruction and Address will be sent in the mode specified by CTRLR0.SPI_FRF. 11 - Reserved.

24. MAC

24.1 MAC Features

1) One MAC interface for connecting external Ethernet PHY
2) 10/100 Mbit/s Ethernet port with RMII interface
3) Compliant with IEEE 802.3-2002 standard
4) Supports full and half duplex operations

25. GPU

25.1 GPU Features

- Supports move operations, in positive and negative direction
- Supports 16 different ROP2 operations
- Porter-Duff composition with/without global alpha
- Integrated optional up/down bitmap scaling
- Integrated vertical and horizontal bitmap flipping
- Color-keyed transparency, source and destination
- Control of pixel alpha blending factors
- Anti-aliased 8-bit font expansion
- 2.5D rendering

- Supported color formats: YUV, YUV420 , RGB565, ARGB565, RGB888, and ARGB888

26.QOI DECODER

26.1 Features

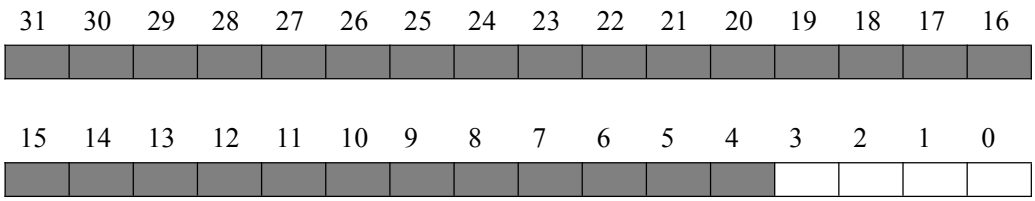
Support RGB888/ARGB888 format.
Support up to 1080p resolution.

26.1 Register Descriptions

AXI_QOI_DEC_BASE: 0x4090_0000

26.1.1 DEC_CTRL_REG

Offset_Address: 0x0000_0000

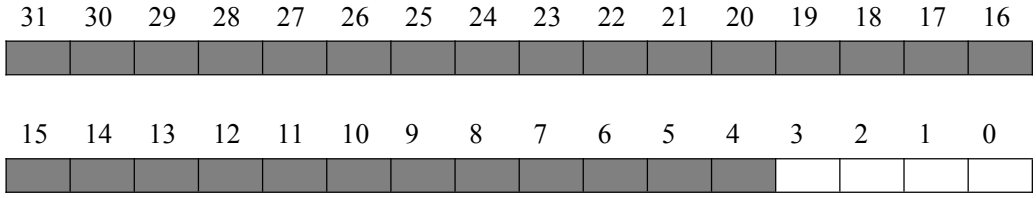


Bit	Type	Reset	Description
31-29	R	0	Reserved
28	R/W	0	wbbvalid_dis 0: 等待 bvalid 才开始下一次写操作 1：不等待 bvalid
27-24	R/W	0	awlen_bits 4'h4: awlen 4 位模式 4'h5: awlen 5 位模式 4'h6: awlen 6 位模式 4'h7: awlen 7 位模式 Other: awlen 8 位模式
23-20	R/W	0	dec_wb_axi_cmd_wid AXI 写 ID
19-16	R/W	0	dec_rd_axi_cmd_id AXI 读 ID
15-12	R/W	0	dec_rd_burst_lenbits 4'h4: arlen 4 位模式 4'h5: arlen 5 位模式 4'h6: arlen 6 位模式

			4'h7: arlen 7 位模式 Other: arlen 8 位模式
11-8	R/W	0	dec_rd_min_burst
7-4	R/W	0	dec_rd_max_burst
3	R/W	0	wb_fifo_state_sel 0: wb fifo AE 到 AF 之间开始回写 1: wb fifo AF 开始回写
2	R/W	0	wrfifo_finish_sel
1	R/W	0	dec_rd_burst_sel 0: unfixed burst 1: fixed burst
0	R/W	0	dec_en

26.1.2 DEC_SIZE_REG

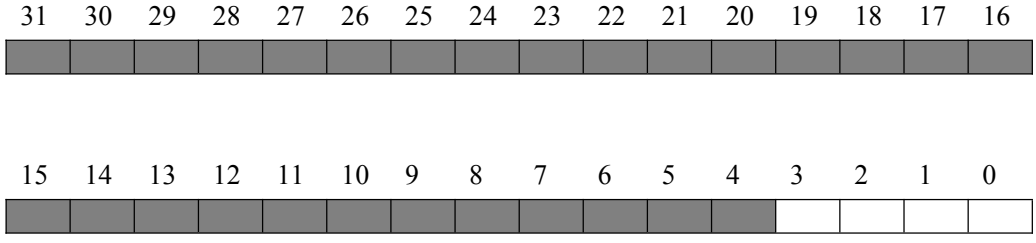
Offset_Address: 0x0000_0004



Bit	Type	Reset	Description
31-24	R	0	Reserved
23-12	R/W	0	dec_high 解码的高度
11-0	R/W	0	dec_wide 解码的宽度

26.1.3 DEC_RD_ADDR_REG

Offset_Address: 0x0000_0008



Bit	Type	Reset	Description
31-0	R/W	0	dec_rd_addr 码流地址

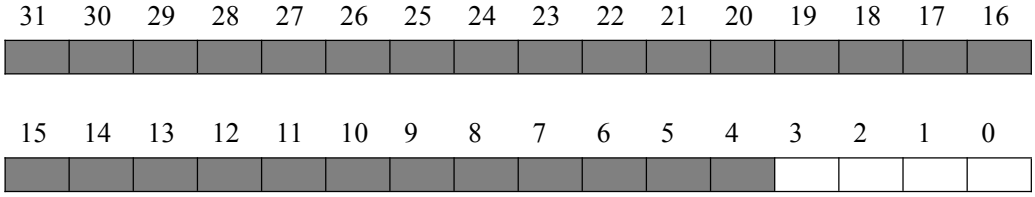
26.1.4 DEC_WB_ADDR_REG

Offset_Address: 0x0000_000C

Bit	Type	Reset	Description
31-0	R/W	0	dec_wb_addr 数据回写地址

26.1.5 DEC_WB_DATA_PER_FRAME_REG

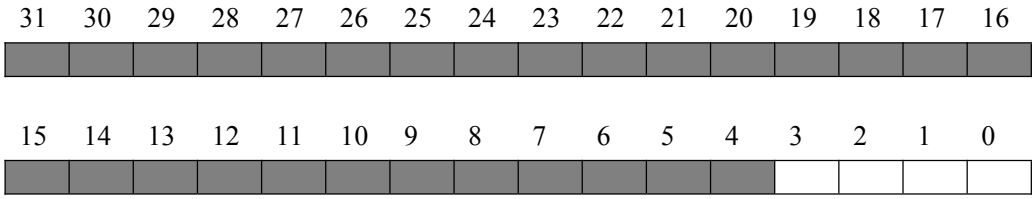
Offset_Address: 0x0000_0010



Bit	Type	Reset	Description
31-0	R/W	0	dec_wb_data_per_frame 数据回写大小

26.1.6 DEC_INT_EN_REG

Offset_Address: 0x0000_0014

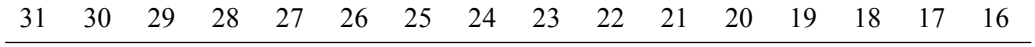


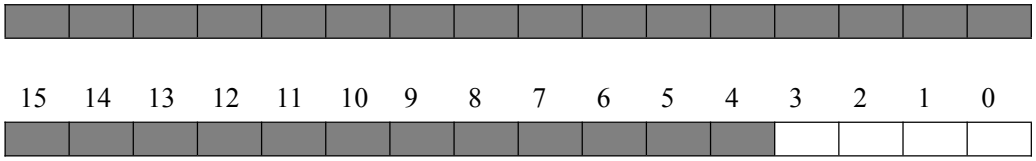
Bit	Type	Reset	Description
31-4	R	0	Reserved
3	R/W	0	dec_wb_frame_finish_int_enable 0: disable; 1: enable;
2	R/W	0	dec_bresp_error_int_enable 0: disable; 1: enable;
1	R/W	0	dec_finish_int_enable 0: disable; 1: enable;
0	R/W	0	dec_error_int_enable 0: disable; 1: enable;

26.1.7 DEC_INT_CLR_REG

Note: Write 1 and write 0 to clear the interrupt.

Offset_Address: 0x0000_0018

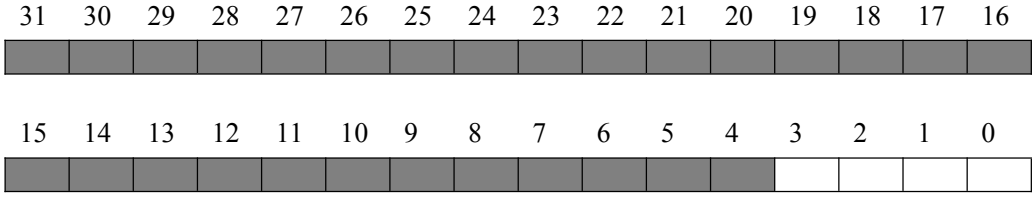




Bit	Type	Reset	Description
31-4	R	0	Reserved
3	R/W	0	dec_wb_frame_finish_int_clr
2	R/W	0	dec_bresp_error_int_clr
1	R/W	0	dec_finish_int_clr
0	R/W	0	dec_error_int_clr

26.1.8 DEC_STATUS_REG

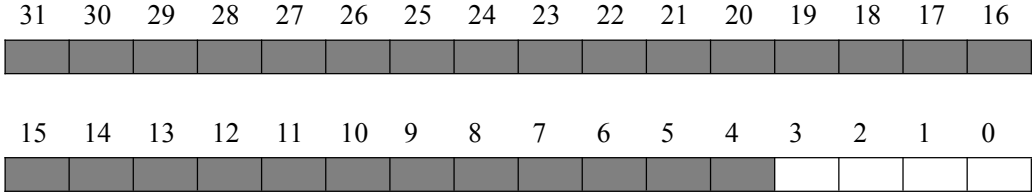
Offset_Address: 0x0000_001C



Bit	Type	Reset	Description
31-4	R	0	Reserved
3	R/W	0	dec_wb_frame_finish
2	R/W	0	dec_bresp_error
1	R/W	0	dec_finish
0	R/W	0	dec_error

26.1.9 DEC_WIDTH_RD_REG

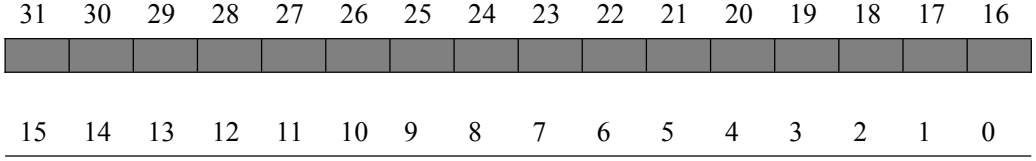
Offset_Address: 0x0000_0020



Bit	Type	Reset	Description
31-0	R	0	dec_width_rd 从码流头信息中解出的宽度

26.1.10 DEC_HEIGHT_RD_REG

Offset_Address: 0x0000_0024



Bit	Type	Reset	Description
31-0	R	0	dec_height_rd 从码流头信息中解出的高度

27. WRAP

27.1 Features

Support up to 1280x720.
Support RGB888 format.

27.2 Register Descriptions

AHB_WRAP_BASE: 0x40C0_0000

Sub Address	Default Value	Bits	Name	Description
0x000		[31:23]	-	reserved
		22	Pic_src_init_sel	原图模式选择 0：一个原图，大小由 Fisheye_w_init_s0, Fisheye_h_init_s0 决定； 1：每个区域 s 对应一个原图，分别由 Fisheye_w_init_s0, Fisheye_h_init_s0，Fisheye_w_init_s1, Fisheye_h_init_s1，... Fisheye_w_init_s5, Fisheye_h_init_s5 决定；
		21	rd_y_width_max_en	S 区域中，需要用到的数据中，最大的横坐标值监测使能，1 有效；
		20	wb_bus_bvalid_dsen	中断生成条件选择 0：回写数据发送完成且等待 bresp 反馈完成时，出中断； 1：回写数据发送完成了，立即出中断；
		19	wb_wait_en	总线回写时，是否等待 bresp 再进行下一次回写选择 0：不需要等待； 1：需要等待；
		18	-	
		17	fml_whc_sel	读取坐标时，数据在 DDR 中的

				存放方式 0：低位代表横坐标（第几列）； 1：高位代表横坐标（第几列）；
		16	fml_bit_sel	读取坐标时，数据在 DDR 中的存放方式 0：高低 32 位分别代表横坐标纵坐标; 1：低 24 位，次高 24 位，分别代表横坐标和纵坐标；
		[15:14]	-	
		13-8	Sections	分段数配置。 1 段：6'b000001 2 段：6'b000011 3 段：6'b000111 4 段：6'b001111 5 段：6'b011111 6 段：6'b111111
		7	-	
		6	-	
		5	bvalid_resp_error	回写总线返回错误标志
		4	Intr_read	只读寄存器，不带 mask 功能的中断状态位
		3	Work_done	Fisheye 完成标识。
		2	Intr_mask_read	只读寄存器，mask 功能的中断状态位
		1	Soft_reset	软复位。置 1 触发。 软件置 1，硬件自动清零。
		0	-	
0x001		31-12	-	
		0	Work_en	Fisheye 工作触发使能。置 1 触发。 软件置 1，硬件自动清零。
0x002		31-12	-	
		11-8	rd_id_ml	读畸变校正映射坐标数据时的 Read ID 配置值。
		7-4	rd_id_y	读畸变源数据 Y（rgb）通道时的 Read ID 配置值。
		3-0	wr_id_y	写畸变校正数据 y（rgb）通道时的 write ID 配置值。
0x003		31-1	-	Reserved
		0	Intr_clr	写 1 对中断进行清零。
0x004		31-1	-	Reserved
		0	Intr_mask	中断 mask 1：中断 mask，不输出中断；

				0：中断输出
0x005		31-29	-	Reserved
	R	28-16	rd_y_width_s1_max	S1 区域中，需要用到的数据中，横坐标最大的值，也即最大到第几列，只读；
		15-13	-	Reserved
	R	12-0	rd_y_width_s0_max	S0 区域中，需要用到的数据中，横坐标最大的值，也即最大到第几列，只读；
0x006		31-29	-	Reserved
	R	28-16	rd_y_width_s3_max	S3 区域中，需要用到的数据中，横坐标最大的值，也即最大到第几列，只读；
		15-13	-	Reserved
	R	12-0	rd_y_width_s2_max	S2 区域中，需要用到的数据中，横坐标最大的值，也即最大到第几列，只读；
0x007		31-29	-	Reserved
	R	28-16	rd_y_width_s5_max	S5 区域中，需要用到的数据中，横坐标最大的值，也即最大到第几列，只读；
		15-13	-	Reserved
	R	12-0	rd_y_width_s4_max	S4 区域中，需要用到的数据中，横坐标最大的值，也即最大到第几列，只读；
0x00b		31-29	-	Reserved
		28-16	Fisheye_h_init_s0	原图的高度。
		15-13	-	Reserved
		12-0	Fisheye_w_init_s0	原图的宽度。
0x00c		31-29	-	Reserved
		28-16	Fisheye_h_init_s1	原图的高度，对应区域 s1。
		15-13	-	Reserved
		12-0	Fisheye_w_init_s1	原图的宽度，对应区域 s1。
0x00d		31-29	-	Reserved
		28-16	Fisheye_h_init_s2	原图的高度，对应区域 s2。
		15-13	-	Reserved
		12-0	Fisheye_w_init_s2	原图的宽度，对应区域 s2。
0x00e		31-29	-	Reserved
		28-16	Fisheye_h_init_s3	原图的高度，对应区域 s3。
		15-13	-	Reserved
		12-0	Fisheye_w_init_s3	原图的宽度，对应区域 s3。
0x00f		31-29	-	Reserved
		28-16	Fisheye_h_init_s4	原图的高度，对应区域 s4。
		15-13	-	Reserved
		12-0	Fisheye_w_init_s4	原图的宽度，对应区域 s4。
0x010		31-29	-	Reserved
		28-16	Fisheye_h_init_s5	原图的高度，对应区域 s5。

		15-13	-	Reserved
		12-0	Fisheye_w_init_s5	原图的宽度，对应区域 s5。
0x011		31-0	rd_faddr_y_s0	源图，分段 1，Y(rgb)通道的起始地址配置。 必须为 64bit 对齐的。
0x012		31-0	rd_faddr_y_s1	源图，分段 2，Y(rgb)通道的起始地址配置。 必须为 64bit 对齐的。
0x013		31-0	rd_faddr_y_s2	源图，分段 3，Y(rgb)通道的起始地址配置。 必须为 64bit 对齐的。
0x014		31-0	rd_faddr_y_s3	源图，分段 4，Y(rgb)通道的起始地址配置。 必须为 64bit 对齐的。
0x015		31-0	rd_faddr_y_s4	源图，分段 5，Y(rgb)通道的起始地址配置。 必须为 64bit 对齐的。
0x016		31-0	rd_faddr_y_s5	源图，分段 6，Y(rgb)通道的起始地址配置。 必须为 64bit 对齐的。
0x017		31-29	-	Reserved
		28-16	rd_height_yuv_s0	源图分段 1 的高度。计量单位为行点。最大到 4095。
		15-12	-	Reserved
		11-0	rd_width_yuv_s0	源图分段 1 的宽度。配置值必须为 2 的整数倍。计量单位为像素点。最大到 512。
0x018		31-29	-	Reserved
		28-16	rd_height_yuv_s1	源图分段 2 的高度。计量单位为行点。最大到 4095。
		15-12	-	Reserved
		11-0	rd_width_yuv_s1	源图分段 2 的宽度。配置值必须为 2 的整数倍。计量单位为像素点。最大到 512。
0x019		31-29	-	Reserved
		28-16	rd_height_yuv_s2	源图分段 3 的高度。计量单位为行点。最大到 4095。
		15-12	-	Reserved
		11-0	rd_width_yuv_s2	源图分段 3 的宽度。配置值必须为 2 的整数倍。计量单位为像素点。最大到 512。
0x01a		31-29	-	Reserved

		28-16	rd_height_yuv_s3	源图分段 4 的高度。计量单位为行点。最大到 4095。
		15-12	-	Reserved
		11-0	rd_width_yuv_s3	源图分段 4 的宽度。配置值必须为 2 的整数倍。计量单位为像素点。最大到 512。
0x01b		31-29	-	Reserved
		28-16	rd_height_yuv_s4	源图分段 5 的高度。计量单位为行点。最大到 4095。
		15-12	-	Reserved
		11-0	rd_width_yuv_s4	源图分段 5 的宽度。配置值必须为 2 的整数倍。计量单位为像素点。最大到 512。
0x01c		31-29	-	Reserved
		28-16	rd_height_yuv_s5	源图分段 6 的高度。计量单位为行点。最大到 4095。
		15-12	-	Reserved
		11-0	rd_width_yuv_s5	源图分段 6 的宽度。配置值必须为 2 的整数倍。计量单位为像素点。最大到 512。
0x01d		31-29	-	Reserved
		28-16	rd_line_init_s0	分段 1 源图的起始点在原图中的垂直位置，0 表示为原图的第 1 行。
		15-13	-	Reserved
		12-0	rd_wide_init_s0	分段 1 源图的起始点在原图中的水平位置，0 表示为原图的第 1 列，值必须为 2 的倍数。
0x01e		31-29	-	Reserved
		28-16	rd_line_init_s1	分段 2 源图的起始点在原图中的垂直位置，0 表示为原图的第 1 行。
		15-13	-	Reserved
		12-0	rd_wide_init_s1	分段 2 源图的起始点在原图中的水平位置，0 表示为原图的第 1 列，值必须为 2 的倍数。
0x01f		31-29	-	Reserved
		28-16	rd_line_init_s2	分段 2 源图的起始点在原图中的垂直位置，0 表示为原图的

				第 1 行。
		15-13	-	Reserved
		12-0	rd_wide_init_s2	分段 2 源图的起始点在原图中的水平位置，0 表示为原图的第 1 列，值必须为 2 的倍数。
0x020		31-29	-	Reserved
		28-16	rd_line_init_s3	分段 4 源图的起始点在原图中的垂直位置，0 表示为原图的第 1 行。
		15-13	-	Reserved
		12-0	rd_wide_init_s3	分段 4 源图的起始点在原图中的水平位置，0 表示为原图的第 1 列，值必须为 2 的倍数。
0x021		31-29	-	Reserved
		28-16	rd_line_init_s4	分段 5 源图的起始点在原图中的垂直位置，0 表示为原图的第 1 行。
		15-13	-	Reserved
		12-0	rd_wide_init_s4	分段 5 源图的起始点在原图中的水平位置，0 表示为原图的第 1 列，值必须为 2 的倍数。
0x022		31-29	-	Reserved
		28-16	rd_line_init_s5	分段 6 源图的起始点在原图中的垂直位置，0 表示为原图的第 1 行。
		15-13	-	Reserved
		12-0	rd_wide_init_s5	分段 6 源图的起始点在原图中的水平位置，0 表示为原图的第 1 列，值必须为 2 的倍数。
0x028		31-0	rd_faddr_ml_s0	畸变校正图坐标，分段 1，映射坐标的起始地址配置。
0x029		31-0	rd_faddr_ml_s1	畸变校正图坐标，分段 2，映射坐标的起始地址配置。
0x02a		31-0	rd_faddr_ml_s2	畸变校正图坐标，分段 3，映射坐标的起始地址配置。
0x02b		31-0	rd_faddr_ml_s3	畸变校正图坐标，分段 4，映射坐标的起始地址配置。
0x02c		31-0	rd_faddr_ml_s4	畸变校正图坐标，分段 5，映射坐标的起始地址配置。
0x02d		31-0	rd_faddr_ml_s5	畸变校正图坐标，分段 6，映射坐标的起始地址配置。

0x02e		31-29	-	Reserved
		28-16	ml_height_s0	畸变校正图坐标，分段 1 的高度。
		15-12	-	Reserved
		11-0	ml_width_s0	畸变校正图坐标，分段 1 的宽度。
0x02f		31-29	-	Reserved
		28-16	ml_height_s1	畸变校正图坐标，分段 2 的高度。
		15-12	-	Reserved
		11-0	ml_width_s1	畸变校正图坐标，分段 2 的宽度。
0x030		31-29	-	Reserved
		28-16	ml_height_s2	畸变校正图坐标，分段 3 的高度。
		15-12	-	Reserved
		11-0	ml_width_s2	畸变校正图坐标，分段 3 的宽度。
0x031		31-29	-	Reserved
		28-16	ml_height_s3	畸变校正图坐标，分段 4 的高度。
		15-12	-	Reserved
		11-0	ml_width_s3	畸变校正图坐标，分段 4 的宽度。
0x032		31-29	-	Reserved
		28-16	ml_height_s4	畸变校正图坐标，分段 5 的高度。
		15-12	-	Reserved
		11-0	ml_width_s4	畸变校正图坐标，分段 5 的宽度。
0x033		31-29	-	Reserved
		28-16	ml_height_s5	畸变校正图坐标，分段 6 的高度。
		15-12	-	Reserved
		11-0	ml_width_s5	畸变校正图坐标，分段 6 的宽度。
0x034		31-29	-	Reserved
		28-16	img_r_s1	畸变校正图坐标读取,整个坐标图在 memory 的一行宽度，分段 2，一般为展开图坐标的宽度。
		15-13	-	Reserved
		12-0	img_r_s0	畸变校正图坐标读取,整个坐标图在 memory 的一行宽度，分段 1，一般为展开图坐标的宽度。
0x035		31-29	-	Reserved
		28-16	img_r_s3	畸变校正图坐标读取,整个坐标图在 memory 的一行宽度，分段

				4，一般为展开图坐标的宽度。
		15-13	-	Reserved
		12-0	img_r_s2	畸变校正图坐标读取,整个坐标图在 memory 的一行宽度，分段 3，一般为展开图坐标的宽度。
0x036		31-29	-	Reserved
		28-16	img_r_s5	畸变校正图坐标读取,整个坐标图在 memory 的一行宽度，分段 6，一般为展开图坐标的宽度。
		15-13	-	Reserved
		12-0	img_r_s4	畸变校正图坐标读取,整个坐标图在 memory 的一行宽度，分段 5，一般为展开图坐标的宽度。
0x037		31-29	-	Reserved
		28-16	img_w_mem_s1	畸变校正图回写 memory 时，memory 中一行的宽度，分段 2，也即 memory 中的行间距。
		15-13	-	Reserved
		12-0	img_w_mem_s0	畸变校正图回写 memory 时，memory 中一行的宽度，分段 1，也即 memory 中的行间距。
0x038		31-29	-	Reserved
		28-16	img_w_mem_s3	畸变校正图回写 memory 时，memory 中一行的宽度，分段 4，也即 memory 中的行间距。
		15-13	-	Reserved
		12-0	img_w_mem_s2	畸变校正图回写 memory 时，memory 中一行的宽度，分段 3，也即 memory 中的行间距。
0x039		31-29	-	Reserved
		28-16	img_w_mem_s5	畸变校正图回写 memory 时，memory 中一行的宽度，分段 6，也即 memory 中的行间距。
		15-13	-	Reserved
		12-0	img_w_mem_s4	畸变校正图回写 memory 时，memory 中一行的宽度，分段 5，也即 memory 中的行间距。
0x03a		31-0	wr_faddr_rgb_s0	畸变校正图，分段 1，数据的存储起始地址。必须为 64bit 对齐的。
0x03b		31-0	wr_faddr_rgb_s1	畸变校正图，分段 2，数据的存储起始地址。必须为 64bit 对齐的。
0x03c		31-0	wr_faddr_rgb_s2	畸变校正图，分段 3，数据的存储起始地址。必须为 64bit 对齐的。
0x03d		31-0	wr_faddr_rgb_s3	畸变校正图，分段 4，数据的存

				储起始地址。必须为 64bit 对齐的。
0x03e		31-0	wr_faddr_rgb_s4	畸变校正图，分段 5，数据的存储起始地址。必须为 64bit 对齐的。
0x03f		31-0	wr_faddr_rgb_s5	畸变校正图，分段 6，数据的存储起始地址。必须为 64bit 对齐的。

28. BLEND

28.1 Features

Support two layer overlay.
Support color key.
Support source data format ARGB888、RGB565、RGB888、RGB666，destination data format ARGB888、RGB888、RGB565.
Support up to 1280x720(ARGB888) resolution.

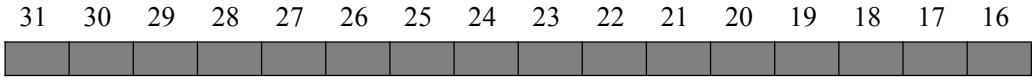
28.2 Blend Register Descriptions

BLEND_BASE: 0x40A0_0000

28.1.1 AXI_BLEND_ENABLE: AXI Blend Global Enable

This register globally enables the AXI Blend module，set 0 can directly disable.

Offset_Address: 0x0000_0000



Bit	Type	Reset	Description
31-1	R/W	0	Reserved.
0	R/W	0	1: enable AXI blend module 0: disable AXI blend module

28.1.2 AXI Blend start : AXI blend operation start Register

Offset_Address: 0x0000_0004

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-1	R/W	0	**Reserved**
0	R/W	0	**AXI blend module 开始操作** **该寄存器先置 1，然后置 0**

28.1.3 AXI_BLEND_CONTROL: AXI blend module Control Register

Control register of the AXI blend module.

Offset_Address: 0x0000_0008

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

Bit	Type	Reset	Description
31-2	R/W	0	**Reserved**
1	R/W	0	**osd_layer2_enable** osd 层 2 使能
0	R/W	0	**osd_layer1_enable** osd 层 1 使能

28.1.4 AXI_BLEND_MODE : AXI blend Control Register

Offset_Address: 0x0000_000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

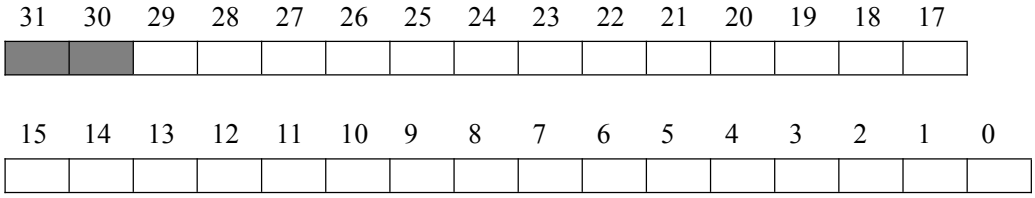
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-16	R	0	osd_req_wait_value Osd1 osd2 读请求时，读等待使能时，等待时间配置值。
15-12	R/W	0	**axi_cmd_id** **AXI read id number**
11-8	R/W	0	**bld_layer_mix_mode** 2 层 OSD 做 blend 混合时，2 层 OSD 混合的模式选择配置寄存器 0000: 2 层 OSD 做 blend 混合； 0001：OSD 层 1 数据输出； 0010：OSD 层 2 数据输出；

			0011：OSD 层 1 做 key color，key color 部分 overwrite，其余部分 transparent absolutely; 0100：OSD 层 1 做 key color，key color 部分 overwrite，其余部分 2 层 OSD 做 blend 混合; 0101：OSD 层 1 做 key color，key color 部分 transparent absolutely，其余部分 2 层 OSD 做 blend 混合; 0110：OSD 层 1 做 key color，key color 部分 transparent absolutely，其余部分 OSD 层 1 overwrite; 0111：OSD 层 1 做 key color，key color 部分 2 层 OSD 做 blend 混合，其余部分 OSD 层 1 transparent absolutely; 1000：OSD 层 1 做 key color，key color 部分 2 层 OSD 做 blend 混合，其余部分 OSD 层 1 overwrite; 1001：OSD 层 2 做 key color，key color 部分 transparent absolutely，其余部分 OSD 层 2 overwrite; 1010：OSD 层 2 做 key color，key color 部分 transparent absolutely，其余部分 2 层 OSD 做 blend 混合; 1011：OSD 层 2 做 key color，key color 部分 overwrite，其余部分 2 层 OSD 做 blend 混合; 1100：OSD 层 2 做 key color，key color 部分 overwrite，其余部分 transparent absolutely; 1101：OSD 层 2 做 key color，key color 部分 2 层 OSD 做 blend 混合，其余部分 OSD 层 2overwrite; 1110：OSD 层 2 做 key color，key color 部分 2 层 OSD 做 blend 混合，其余部分 transparent absolutely;
7	R/W	0	**empty_sel** Fifo (16x64) 空标志的计算方法
6	R/W	0	reserved
5	R/W	0	osd_req_wait_en Osd1 osd2 读请求时，读等待使能，1 有效。
4	R/W	0	**bld_layer1_alpha_mode** 2 层 OSD 做 blend 混合时，OSD 层 2 的 ALPHA 系数选择 0：选择 OSD 层 1 数据中的 ALPHA 值，作用到该层的值为（256-ALPHA）; 1：选择 OSD 层 2 数据中的 ALPHA 值，该 ALPHA 值可以是 OSD 层数据中固有的值，也可以选择寄存器配置的该层的 ALPHA 值。由寄存器 osd2_global_alpha_sel 选择。
3-1	R/W	0	RESERVED
0	R/W	0	**bld_layer0_alpha_mode** 2 层 OSD 做 blend 混合时，OSD 层 1 的 ALPHA 系数选择 0：选择 OSD 层 2 数据中的 ALPHA 值，作用到该层的值为（256-ALPHA）; 1：选择 OSD 层 1 数据中的 ALPHA 值，该 ALPHA 值可以是 OSD 层数据中固有的值，也可以选择寄存器配置的该层的 ALPHA 值。由寄存器 osd1_global_alpha_sel 选择。

28.1.5 AXI_BLEND_WRITE_WAIT : AXI blend Write wait Register

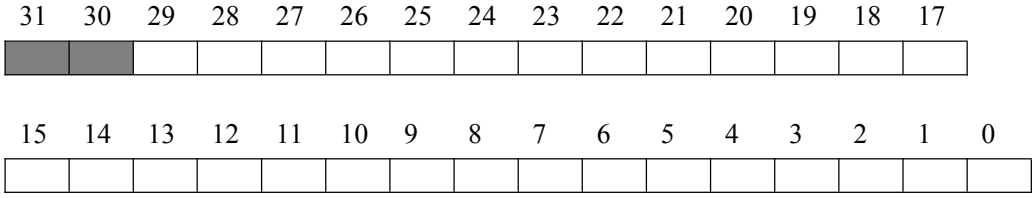
Offset_Address: 0x0000_0010



Bit	Type	Reset	Description
31-13			reserved
12	R/W	0	**trans_write_wait_en** AXI blend 回写时，写之间的等待模式设置，1 有效。
11-0	R/W	0	**trans_write_wait_value** AXI blend 回写时，写之间的等待时间值。

28.1.6 AXI_BLEND_ALPHA_THROLD : AXI blend alpha threshold register

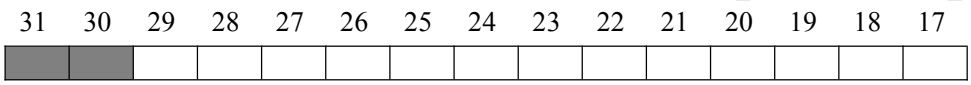
Offset_Address: 0x0000_0014

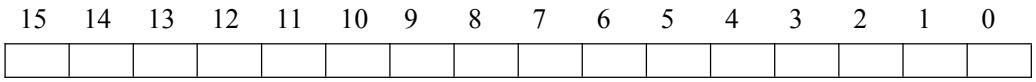


Bit	Type	Reset	Description
23-16	R/W	F4	**dest_alpha_a_big** AXI blend blending 时，目的层 alpha 阈值。
23-16	R/W	08	**dest_alpha_a_sma** AXI blend blending 时，目的层 alpha 阈值。 当源层数据中的 alpha 值在中间值，且当目的层数据中的 alpha 小于该阈值时，blending 后的结果选取源层数据及源层 alpha 值与 global alpha 值的积。
15-8	R/W	08	**source_alpha_a_small** AXI blend blending 时，源层 alpha 阈值。 当数据中的 alpha 小于该阈值时，blending 后的结果选取目的层数据及 alpha 值。
7-0	R/W	F8	**source_alpha_a_big** AXI blend blending 时，源层 alpha 阈值。 当数据中的 alpha 大于该阈值时，blending 后的结果选取源层数据及 alpha 值。

28.1.7 AXI_BLEND_ALPHA_THROLD_2 : AXI blend alpha threshold Register

Offset_Address: 0x0000_0018

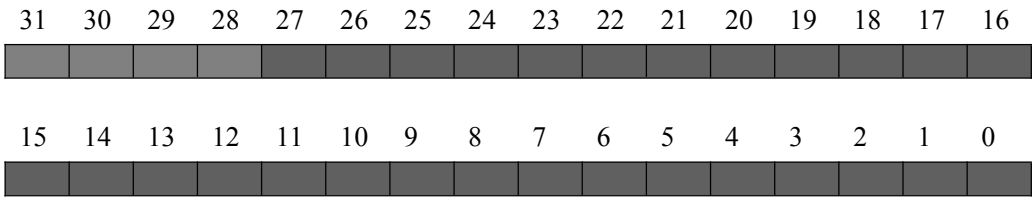




Bit	Type	Reset	Description
31-10			reserved
11	R/W	0	**Layer1_data_preproces_en** OSD 1 层，是否对数据做预处理使能，1 有效。 为 1 时，数据与自身的 alpha 做运算，r/g/b*alpha/256
10	R/W	0	**layer0_data_preproces_en** OSD 0 层，是否对数据做预处理使能，1 有效。 为 1 时，数据与自身的 alpha 做运算，r/g/b*alpha/256
9	R/W	0	**premul_mod_sel** AXI blend blending 时，blending 模式选择。 0：与程序算法一致的合成模式； 1：在预乘 alpha 通路中，目标层 alpha 小于 **dest_alpha_a_big** 时，采用与程序算法不一样的合成模式；
8	R/W	0	**out_a_sel** AXI blend 回写时，out_alpha 选择。 0：与程序算法一致的 alpha 输出； 1：两层做 blending 的 alpha 之和；
7-0	R/W	F8	**global_alpha_a_big** AXI blend blending 时，global alpha 的阈值。

28.1.8 AXI_BLEND_OSD1_BURST_CTL: OSD Layer1 Burst Control Register

Offset_Address: 0x0000_0020



Bit	Type	Reset	Description
31-17	R	0	Reserved.
23-16	R/W	0	osd1_global_alpha OSD层1数据中，寄存器配置的alpha系数 当osd1_global_alpha_sel = 1 时，该层的alpha系数选用 osd1_global_alpha
15	R/W	0	osd1_global_alpha_sel OSD层1数据中，alpha系数选择 1：选用寄存器配置的alpha作为该层的alpha系数； 0：选用数据中的alpha作为该层的alpha系数；
14	R/W	0	reserved
13	R/W	0	osd1_alpha_bigend_sel

			OSD层1数据中，alpha系数的位置 1：RGBA 0：ARGB
12	R/W	0	osd1_bigend_sel OSD层1数据，大端选择 1：大端 0：小端
11	R/W	0	Osd1_burst_bypass（**reserved**） 其它层是否考虑osd1层来参与仲裁的选择 1：不参与仲裁，此时，其它层将无视osd1层fifo中已取数的值，不与osd1层来仲裁，但osd1层还是需要来仲裁其它参与仲裁的层。该种情况下，人为地将osd1层的优先级降低。 0：参与仲裁。
10	R/W	0	Osd1_burst_sel OSD1层取数方式的选择 1：固定burst取数，此时根据osd1_min_burst确定burst个数来取数，与osd1_max_burst无关。 0：动态burst取数，此时根据osd1_min_burst、osd1_max_burst以及fifo中数的个数来取数，osd1_min_burst设定最小burst的值，osd1_max_burst设定最大burst的值，osd1_min_burst<=osd1_max_burst。当fifo中所剩空间大于等于osd1_max_burst，则以osd1_max_burst来取数，否则以与fifo中所剩空间相匹配的burst个数来取数（该burst数介于最小值与最大值之间）。
9-8	R/W	0	Osd1_dat_num_sel OSD1 layer中已取存放于fifo中数的倍数。乘上倍数之后的数值用来仲裁各层（video video2 osd1 osd2 osd3）对总线的占用，ram中数最少的层将优先得到取数权。根据源格式（422的显示点比444多一倍）以及显示时钟的大小来配置。原则是尽量平衡各层的资源。如osd1层为422源，video层为444源，则可将该位设为1，此时当osd1层ram中的数少于video层ram中的数一半时，osd1层才能够得到取数权。 0：x1 1：x2 2：x3 3：x4
7-4	R/W	0	Osd1_max_burst OSD1 layer取数的最大burst数，该值应设为2—12，对应的burst数如下所示： 2：最大burst数为2（这里一个burst为16x64），即取数最大以 32x64为单位。 3：最大burst数为3，即取数最大以 48x64为单位。 4：最大burst数为4，即取数最大以64x64为单位。 5：最大burst数为5，即取数最大以80x64为单位。 6：最大burst数为6，即取数最大以96x64为单位。 7：最大burst数为7，即取数最大以112x64为单位。 8：最大burst数为8，即取数最大以128x64为单位。

			9：最大burst数为9，即取数最大以144x64为单位。 10：最大burst数为10，即取数最大以160x64为单位。 10：最大burst数为11，即取数最大以176x64为单位。 >=12：最大burst数为12，即取数最大以192x64为单位。
3-0	R/W	0	Osd1_min_burst OSD1 layer取数的最小burst数，该值应设为2—12，对应的burst数如下所示： 2：最小burst数为2（这里一个burst为16x64），即取数小以 32x64为单位。 3：最小burst数为3，即取数最小以 48x64为单位。 4：最小burst数为4，即取数最小以64x64为单位。 5：最小burst数为5，即取数最小以80x64为单位。 6：最小burst数为6，即取数最小以96x64为单位。 7：最小burst数为7，即取数最小以112x64为单位。 8：最小burst数为8，即取数最小以128x64为单位。 9：最小burst数为9，即取数最小以144x64为单位。 10：最小burst数为10，即取数最小以160x64为单位。 10：最小burst数为11，即取数最小以176x64为单位。 >=12：最小burst数为12，即取数最小以192x64为单位。

28.1.9 AXI_BLEND_OSD1_CTL: OSD Layer1 Control Register

This layer owns a vde.

Offset_Address: 0x0000_0024

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

Bit	Type	Reset	Description
31-9	R/W	0	reserved
8	R/W	0	reserved
7	R/W	0	reserved
6-4	W/R	0	**Osd1_rgb_yuv_order_sel** (rgb565 only use 000 or 101) 000:RGB 001:RBG 010:GRB 011:GBR 100:BRG 101:BGR
3-0	R/W	0	**Osd layer1 format** **0-3:reserved** 4: rgb_i555(**reserved**) 5: rgb_565 6: rgb_a888 7: rgb_888

			8: (**reserved**) 9: rgb_a1888(**reserved**) 10: rgb_a4888(**reserved**) 11: rgb_666 12: rgb_a1666(**reserved**)

28.1.10 AXI_BLEND_OSD1_SIZE: OSD Layer1 Size

This register defines the size of an image displaying one this layer.

Offset_Address: 0x0000_0028

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

Bit	Type	Reset	Description
31-24	R/W	0	Saturation.
23-12	R/W	0	High of osd layer1
11-0	R/W	0	Wide of osd layer1

28.1.11 AXI_BLEND_OSD1_ADDR: OSD Layer1 Data Start Address Register

This register store the data buffer address group0 in memory for the image displaying on this layer

Offset_Address: 0x0000_002C

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

Bit	Type	Reset	Description
31-0	R/W	0	Start address.

28.1.12 AXI_BLEND_OSD2_BURST_CTL: OSD Layer2 Burst Control Register

Offset_Address: 0x0000_0030

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

Bit	Type	Reset	Description
31-12	R	0	Reserved.
23-16	R/W	0	osd1_global_alpha OSD层1数据中，寄存器配置的alpha系数 当osd1_global_alpha_sel = 1 时，该层的alpha系数选用osd1_global_alpha
15	R/W	0	osd1_global_alpha_sel OSD层1数据中，alpha系数选择 1：选用寄存器配置的alpha作为该层的alpha系数； 0：选用数据中的alpha作为该层的alpha系数；
14	R/W	0	reserved
13	R/W	0	osd1_alpha_bigend_sel OSD层1数据中，alpha系数的位置 1：RGBA 0：ARGB
12	R/W	0	osd1_bigend_sel OSD层1数据，大端选择 1：大端 0：小端
11	R/W	0	Osd2_burst_bypass（**reserved**） 其它层是否考虑osd2层来参与仲裁的选择 1：不参与仲裁，此时，其它层将无视osd2层fifo中已取数的值，不与osd2层来仲裁，但osd2层还是需要来仲裁其它参与仲裁的层。该种情况下，人为地将osd2层的优先级降低。 0：参与仲裁。
10	R/W	0	Osd2_burst_sel OSD2层取数方式的选择 1：固定burst取数，此时根据osd2_min_burst确定burst个数来取数，与osd2_max_burst无关。 0：动态burst取数，此时根据osd2_min_burst、osd2_max_burst以及fifo中数的个数来取数，osd2_min_burst设定最小burst的值，osd2_max_burst设定最大burst的值，osd2_min_burst<=osd2_max_burst。当fifo中所剩空间大于等于osd2_max_burst，则以osd2_max_burst来取数，否则以与fifo中所剩空间相匹配的burst个数来取数（该burst数介于最小值与最大值之间）。
9-8	R/W	0	Osd2_dat_num_sel OSD2 layer中已取存放于fifo中数的倍数。乘上倍数之后的数值用来仲裁各层（video video2 osd1 osd2 osd3）对总线的占用，ram中数最少的层将优先得到取数权。根据源格式（422的显示点比444多一倍）以及显示时钟的大小来配置。原则是尽量平衡各层的资源。如osd2层为422源，video层为444源，则可将该位设为1，此时当osd2层ram中的数少于video层ram中的数一半时，osd2层才能够得到取数权。 0：x1

			1：x2 2：x3 3：x4
7-4	R/W	0	Osd2_max_burst OSD2 layer取数的最大burst数，该值应设为2—12，对应的burst数如下所示： 2：最大burst数为2（这里一个burst为16x64），即取数最大以 32x64为单位。 3：最大burst数为3，即取数最大以 48x64为单位。 4：最大burst数为4，即取数最大以64x64为单位。 5：最大burst数为5，即取数最大以80x64为单位。 6：最大burst数为6，即取数最大以96x64为单位。 7：最大burst数为7，即取数最大以112x64为单位。 8：最大burst数为8，即取数最大以128x64为单位。 9：最大burst数为9，即取数最大以144x64为单位。 10：最大burst数为10，即取数最大以160x64为单位。 10：最大burst数为11，即取数最大以176x64为单位。 >=12：最大burst数为12，即取数最大以192x64为单位。
3-0	R/W	0	Osd2_min_burst OSD2 layer取数的最小burst数，该值应设为2—12，对应的burst数如下所示： 2：最小burst数为2（这里一个burst为16x64），即取数小以 32x64为单位。 3：最小burst数为3，即取数最小以 48x64为单位。 4：最小burst数为4，即取数最小以64x64为单位。 5：最小burst数为5，即取数最小以80x64为单位。 6：最小burst数为6，即取数最小以96x64为单位。 7：最小burst数为7，即取数最小以112x64为单位。 8：最小burst数为8，即取数最小以128x64为单位。 9：最小burst数为9，即取数最小以144x64为单位。 10：最小burst数为10，即取数最小以160x64为单位。 10：最小burst数为11，即取数最小以176x64为单位。 >=12：最小burst数为12，即取数最小以192x64为单位。

28.1.13 AXI_BLEND_OSD2_CTL: OSD Layer1 Control Register

Offset_Address: 0x0000_0034

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

Bit	Type	Reset	Description
31-9	R/W	0	reserved
8	R/W	0	reserved
7	R/W	0	reserved

6-4	W/R	0	**Osd2_rgb_yuv_order_sel** (rgb565 only use 000 or 101) 000:RGB 001:RBG 010:GRB 011:GBR 100:BRG 101:BGR
3-0	R/W	0	**Osd layer2 format** **0-3:reserved** 4: rgb_i555(**reserved**) 5: rgb_565 6: rgb_a888 7: rgb_888 8: (**reserved**) 9: rgb_a1888(**reserved**) 10: rgb_a4888(**reserved**) 11: rgb_666 12: rgb_a1666(**reserved**)

28.1.14 AXI_BLEND_OSD2_SIZE: OSD Layer2 Size

Offset_Address: 0x0000_0038

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

Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	**High of osd layer2**
11-0	R/W	0	**Wide of osd layer2**

28.1.15 AXI_BLEND_OSD2_ADDR: OSD Layer2 Video Start Address Register

This register store the data buffer address group0 in memory for the image displaying on this layer

Offset_Address: 0x0000_003C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

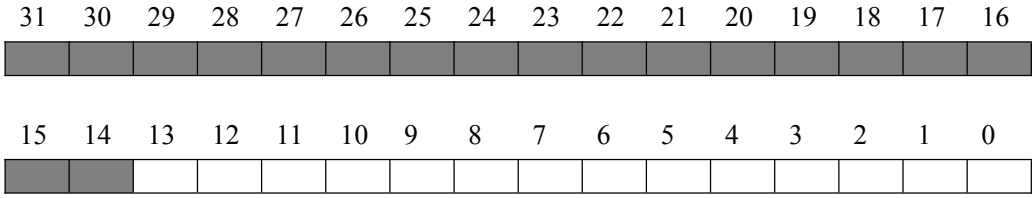
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-0	R/W	0	Start address of layer2 data.

28.1.16 AXI_BLEND_COLOR_KEY_MASK_THLD_OSD1

Color key mask threshold of osd1 layer.

Offset_Address: 0x0000_0040

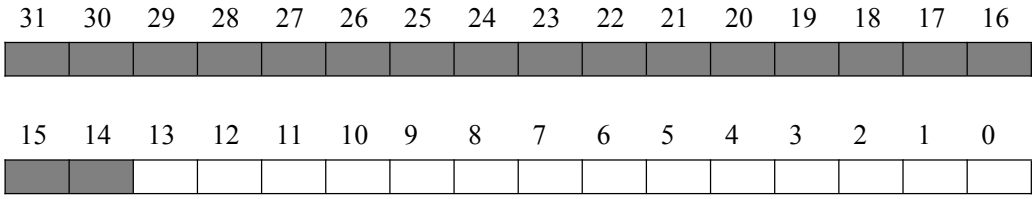


Bit	Type	Reset	Description
31-24	R/W	0	Reserved.
23-16	R/W	0	OSD1 layer color key mask threshold of R
15-8	R/W	0	OSD1 layer color key mask threshold of G
7-0	R/W	0	OSD1 layer color key mask threshold of B

28.1.17 AXI_BLEND_COLOR_KEY_MASK_THLD_OSD2

Color key mask threshold of osd2 layer.

Offset_Address: 0x0000_0044

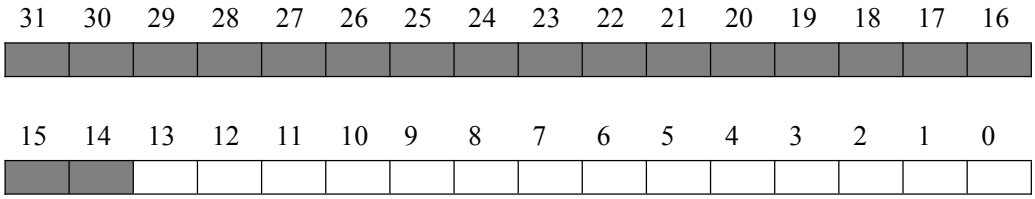


Bit	Type	Reset	Description
31-24	R/W	0	Reserved.
23-16	R/W	0	OSD2 layer color key mask threshold of Y
15-8	R/W	0	OSD2 layer color key mask threshold of Cb
7-0	R/W	0	OSD2 layer color key mask threshold of Cr

28.1.18 AXI_BLEND_COLOR_KEY_MASK_VALUE_OSD1

Color key mask value of OSD1 layer.

Offset_Address: 0x0000_0048

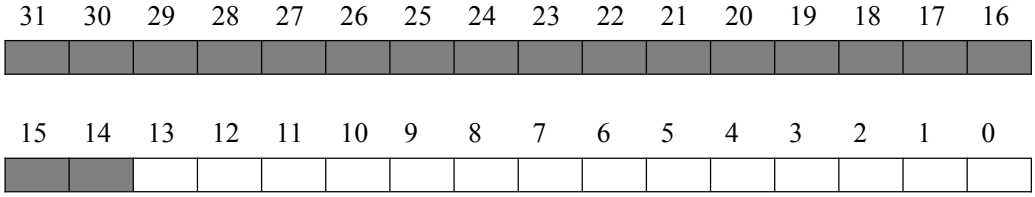


Bit	Type	Reset	Description
31-26	R/W	0	Reserved.
25	R/W	0	OSD1_COLRKEY_BACK_EN 1：OSD 层 1 与背景层先做 color key，key color 部分选择背景色 osd1_back_color； 0：OSD 层 1 不与背景层做 color key；
24	R/W	0	Osd1 layer color key mask enable 1: enable; 0: disable.
23-16	R/W	0	OSD1 layer color key mask value of R
15-8	R/W	0	OSD1 layer color key mask value of G
7-0	R/W	0	OSD1 layer color key mask value of B

28.1.19 AXI_BLEND_COLOR_KEY_MASK_VALUE_OSD2

Color key mask value of OSD2 layer.

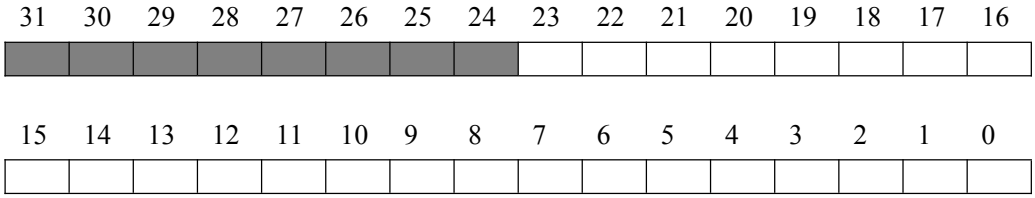
Offset_Address: 0x0000_004C



Bit	Type	Reset	Description
31-26	R/W	0	Reserved.
25	R/W	0	OSD2_COLRKEY_BACK_EN 1：OSD 层 2 与背景层先做 color key，key color 部分选择背景色 osd2_back_color； 0：OSD 层 2 不与背景层做 color key；
24	R/W	0	Osd2 layer color key mask enable 1: enable; 0: disable.
23-16	R/W	0	OSD2 layer color key mask value of R
15-8	R/W	0	OSD2 layer color key mask value of G
7-0	R/W	0	OSD2 layer color key mask value of B

28.1.20 AXI_BLEND_OSD1_WINDOW_POINT: Start Point of OSD1 Layer Image

Offset_Address: 0x0000_0050

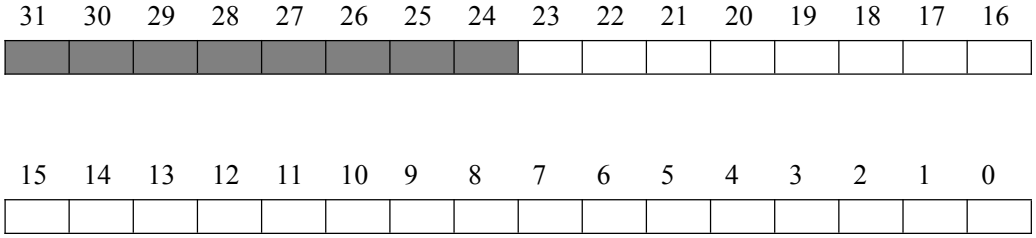


Bit	Type	Reset	Description
31-24	R/W	0	Reserved

23-12	R/W	0	**Y value of the start point**
11-0	R/W	0	**X value of the start point**

28.1.21 AXI_BLEND_OSD2_WINDOW_POINT: Start Point of OSD2 Layer Image

Offset_Address: 0x0000_0054

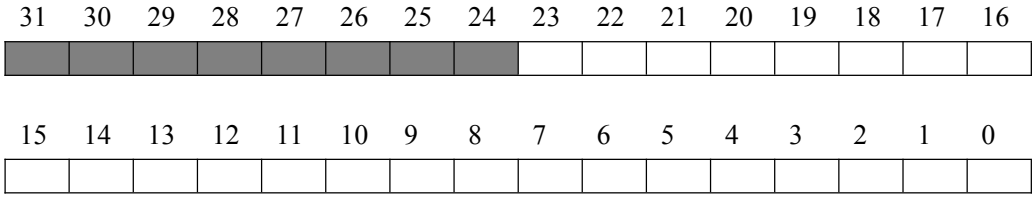


Bit	Type	Reset	Description
31-24	R/W	0	Reserved
23-12	R/W	0	**Y value of the start point**
11-0	R/W	0	**X value of the start point**

28.1.22 AXI_BLEND_OSD1_SOURCE_SIZE: OSD1 Source Size Configure Register

Config the image original size in pixel, which located in memory and waiting to be displayed

Offset_Address: 0x0000_0058

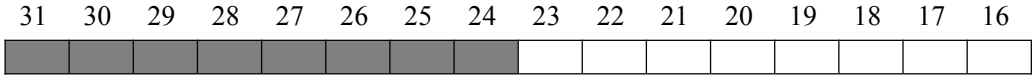


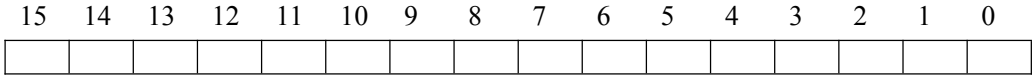
Bit	Type	Reset	Description
31-24	R/W	0	Reserved
23-12	R/W	0	**High of Source OSD1**
11-0	R/W	0	**Wide of Source OSD1**

28.1.23 AXI_BLEND_OSD2_SOURCE_SIZE: OSD2 Source Size Configure Register

Config the image original size in pixel, which located in memory and waiting to be displayed

Offset_Address: 0x0000_005C



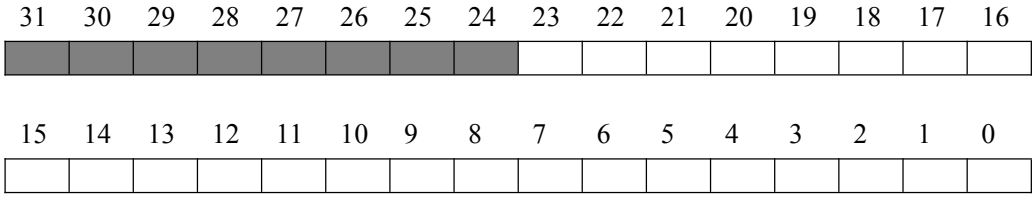


Bit	Type	Reset	Description
31-24	R/W	0	Reserved
23-12	R/W	0	**High of Source OSD2**
11-0	R/W	0	**Wide of Source OSD2**

28.1.24 AXI_BLEND_WRITE_BACK_SOURCE_SIZE: AXI BLEND WRITE BACK Source Size Configure Register

Config the image destination size in pixel, which located in memory and waiting to be displayed

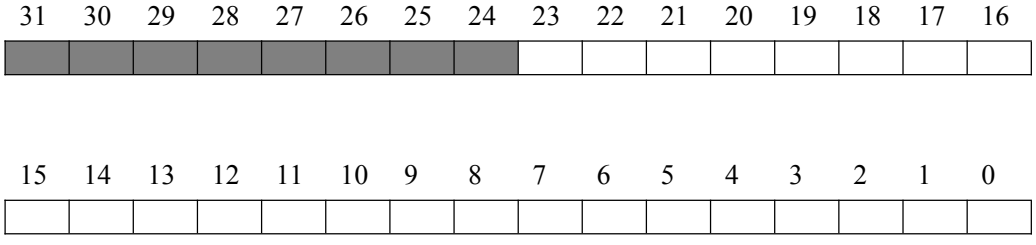
Offset_Address: 0x0000_0060



Bit	Type	Reset	Description
31-24	R/W	0	Reserved
23-12	R/W	0	**High of Source blend write back**
11-0	R/W	0	**Wide of Source blend write back**

28.1.25 AXI_BLEND_BLD_SIZE: OSD BLEND Size

Offset_Address: 0x0000_0064

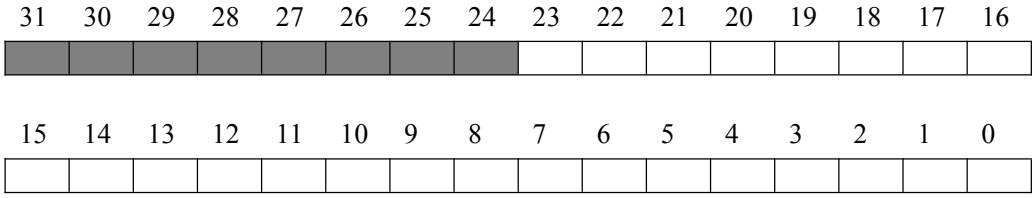


Bit	Type	Reset	Description
31-24	R	0	Reserved.
23-12	R/W	0	**High of osd BLEND**
11-0	R/W	0	**Wide of osd BLEND**

28.1.26 AXI_BLEND_BLD_WINDOW_POINT: Start Point of BLEND Layer

Image

Offset_Address: 0x0000_0068

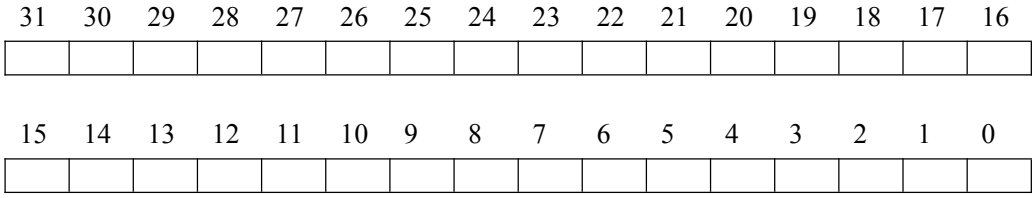


Bit	Type	Reset	Description
31-24	R/W	0	Reserved
23-12	R/W	0	**Y value of the start point**
11-0	R/W	0	**X value of the start point**

28.1.27 AXI_BLEND_BLD_ADDR: OSD BLD Data Start Address Register

This register store the data buffer address in memory for the blending destination.

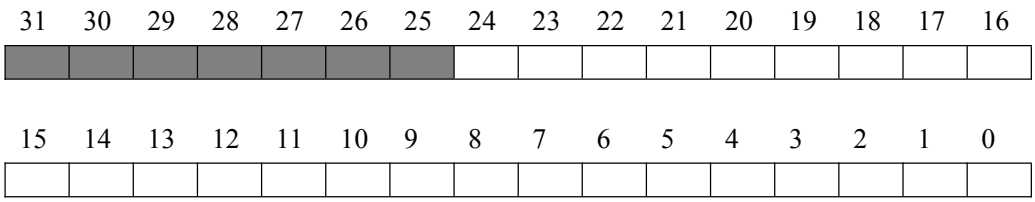
Offset_Address: 0x0000_006C



Bit	Type	Reset	Description
31-0	R/W	0	Start address.

28.1.28 AXI_BLEND_BLD_CTL: OSD Layer BLENDING Control Register

Offset_Address: 0x0000_0070



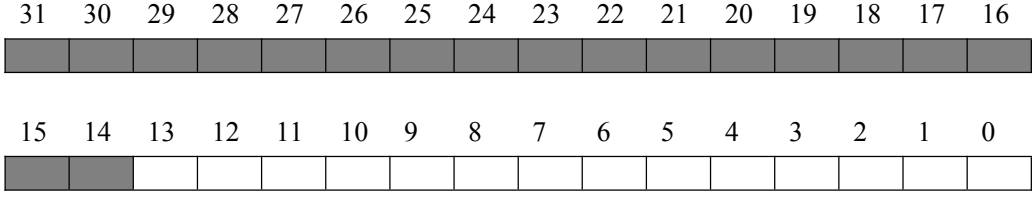
Bit	Type	Reset	Description
31-24	R/W	0	**bld_alpha_wb** Blending 混合后的 alpha 值的寄存器配置值
23	R/W	0	reserved
22	R/W	0	osd_bld_alpha_out_mode Blending 之后输出的逐点 alpha 值模式选择 0：以前的模式，可选择单层、global 的值或其乘积；

			1：源层、目标层、global alpha 的 alpha blending；
21	R/W	0	pre_mul_alpha_en Global alpha 是否参加 blending 选择，当 alpha_blend_mode =1 时有效。 0：global alpha 不参加 blending 运算； 1：global alpha 参加源层、目标层的 blending；
20	R/W	0	alpha_blend_mode Blending 混合的模式选择 0：之前的模式，用单层的 alpha 或 global alpha； 1：采用源层、目标层的 alpha，以及 global alpha 来做 blending
19-17	R/W	0	reserved
16-8	R/W	0	BLD_ADAT_MUL Blending 混合后的 alpha 值的乘系数， 0 -256
7-5	R/W	0	reserved
6-5	R/W	0	**BLD_ALPHA_SEL** **Blending 混合后的数据，数据格式为 ARGB 时，回写的 alpha 值选择** **00: 选择 osd 层 1 中的 alpha 值** **01: 选择 osd 层 2 中的 alpha 值** **10: 选择寄存器配置的 bld_alpha_wb 的值** **11: 选择寄存器配置的 bld_alpha_wb 的值**
4	W/R	0	**BLD_ALPHA_BIG_SEL** **Blending 结果回写时，alpha 放置在数据中的位置** **0：RGBA** **1：ARGB**
3-0	R/W	0	**Osd BLEND format** **0-3:reserved** 4: rgb_i555(**reserved**) 5: rgb_565 6: rgb_a888 7: rgb_888 8: (**reserved**) 9: rgb_a1888(**reserved**) 10: rgb_a4888(**reserved**) 11: rgb_666(**reserved**) 12: rgb_a1666(**reserved**)

28.1.29 AXI_BLEND_OSD1_BACK_COLOR

BACK COLOR OF OSD1 IN KEY COLOR.

Offset_Address: 0x0000_0074



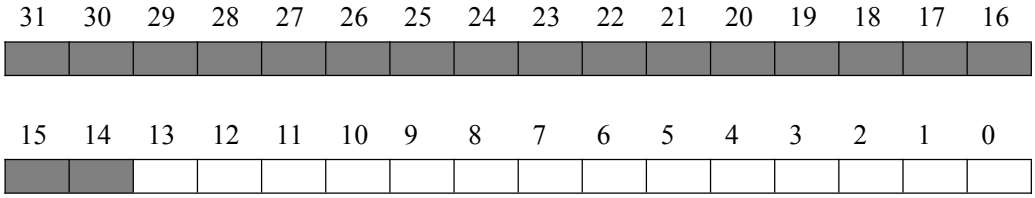
Bit	Type	Reset	Description
31-24	R/W	0	Reserved.
23-16	R/W	0	OSD1 layer color key back color value of R

15-8	R/W	0	OSD1 layer color key back color value of G
7-0	R/W	0	OSD1 layer color key back color value of B

28.1.30 AXI_BLEND_OSD2_BACK_COLOR

BACK COLOR OF OSD2 IN KEY COLOR.

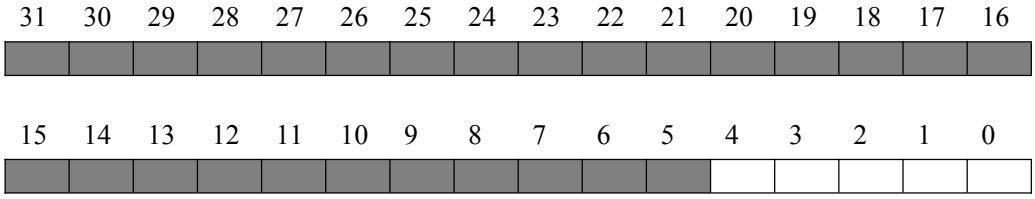
Offset_Address: 0x0000_0078



Bit	Type	Reset	Description
31-24	R/W	0	Reserved.
23-16	R/W	0	OSD2 layer color key back color value of R
15-8	R/W	0	OSD2 layer color key back color value of G
7-0	R/W	0	OSD2 layer color key back color value of B

28.1.31 AXI_BLEND_INT_CTL: AXI BLEND Interrupt Control Register

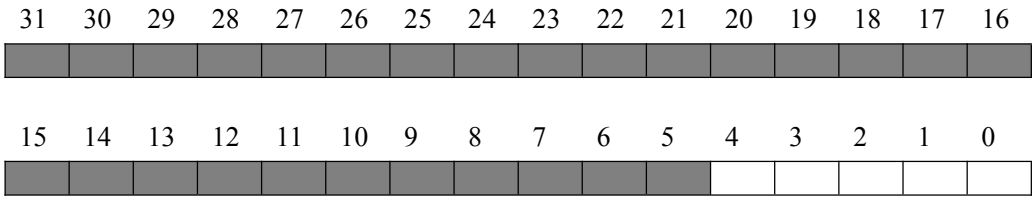
Offset_Address: 0x0000_0080



Bit	Type	Reset	Description
31-2	R/W	0	Reserved
1	R/W	0	**AXI Blend write back bus error interrupt mask** 0: mask 1: enable
0	R/W	0	**AXI Blend write back finish interrupt mask** 0: mask 1: enable

28.1.32 AXI_BLEND_INT_CLR: AXI BLEND Interrupt Clear Register

Offset_Address: 0x0000_0084

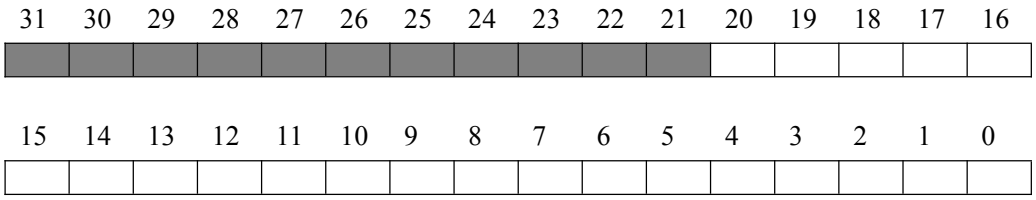


Bit	Type	Reset	Description
31-2	R/W	0	Reserved

1	R/W	0	**AXI Blend write back bus error interrupt clear** Write 0 to clear interrupt
0	R/W	0	**AXI Blend write back finish interrupt clear** Write 0 to clear interrupt

28.1.33 AXI_BLEND_INT_STA: AXI Blend Interrupt Status Register

Offset_Address: 0x0000_0088



Bit	Type	Reset	Description
31-10	R/W	0	Reserved
9	R/W	0	**Osd layer2 fifo push error**
8	R/W	0	**Osd layer2 fifo2 pop error**
7	R/W	0	**Osd layer2 fifo push error**
6	R/W	0	**Osd layer2 fifo pop error**
5	R/W	0	**Osd layer1 fifo push error**
4	R/W	0	**Osd layer1 fifo2 pop error**
3	R/W	0	**Osd layer1 fifo push error**
2	R/W	0	**Osd layer1 fifo pop error**
1	R/W	0	**AXI Blend write back bus error interrupt**
0	R/W	0	**AXI Blend write back finish interrupt**

29. DCIC

29.1 Features

Image Detect:

Support CRC (Cyclic Redundancy Check) Function

Support BDP (Blind Data Packet) Function

Support Y/R/G/B five section histogram

Support IMC (Image Comparison) Function ,compare with character OSD

OSD:

Support character display function.

The character size and position is selectable by register setting

The character size H x V Dot (H,V: Every Character from 30 pixels to 510 pixels)

The blocker window up to 64 ,OSD RAM up to 512KByte, support 8 bit color

Palette：

Support OSD window replace

Support two set color palette

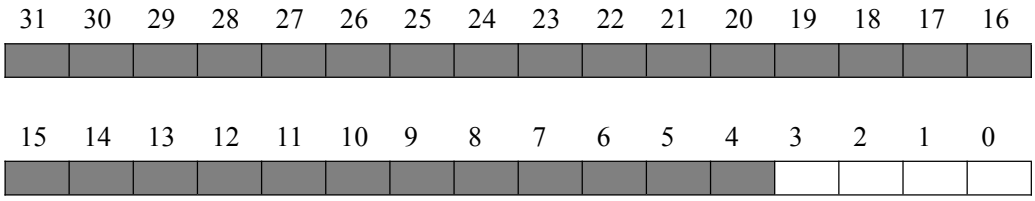
Support window blink function

29.2 Register Descriptions

BASE_ADDR: 0x55500000

DCIC_CFG0

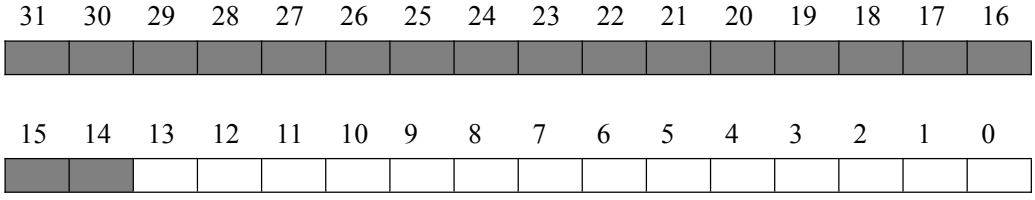
Offset_Address: 0x5550_0000



Bit	Type	Reset	Description
31-24	R/W	0	vsync_detect_th0 DCIC 行/场同步信号检测成功阈值
23-16	R/W	0	line_num_pf_delta DCIC delta of line number between the two near fields. It is used to judge if the line number in field is changed.
15-8	R/W	0	pix_num_delta DCIC delta of pix number between the two near lines. It is used to judge if the pix number in line is changed.
7	R/W	0	hv_det_en DCIC 输入行场检测使能 0: disable; 1: enable
6-4	R/W	0	hv_det_polarity[2:0] DCIC 输入行场极性 [0]: 0: hsyn not inv; 1: hsyn inv; [1]: 0: vsyn not inv; 1: vsyn inv; [2]: 0: de not inv; 1: de inv;
3-0	R	0	reserved

DCIC_CFG3

Offset_Address: 0x5550_000C

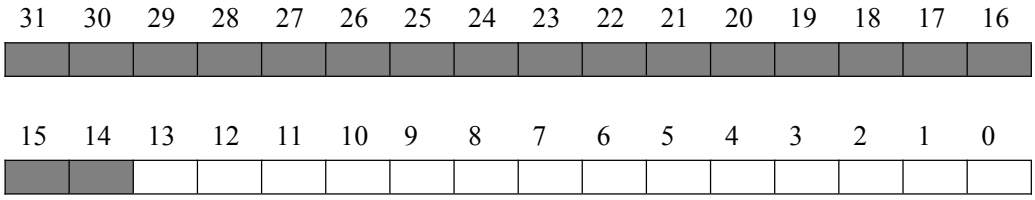


Bit	Type	Reset	Description
31-26	R	0	Reserved.

24	R/W	0	hv_status_sel 0: DCIC 检测场同步信号是否丢失 1：DCIC 检测行同步信号是否丢失
23-0	R/W	0	vsync_detect_th1 DCIC 行/场同步信号检测丢失阈值

DCIC_CFG4

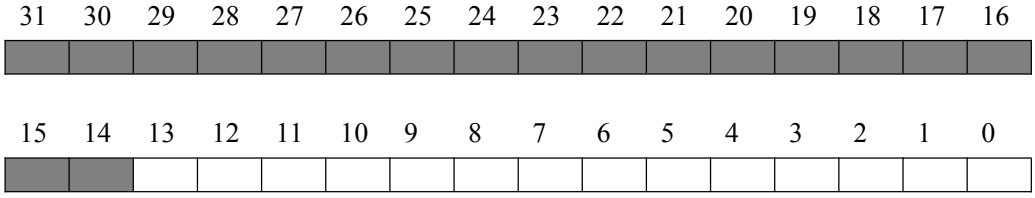
Offset_Address: 0x5550_0010



Bit	Type	Reset	Description
31-24	R/W	0	vsync_debnc_num DCIC debounce field num when line num or pixel num change
23	R	0	reserved
22-0	R/W	0	total_pix_out DCIC valid pix per frame

DCIC_CFG5

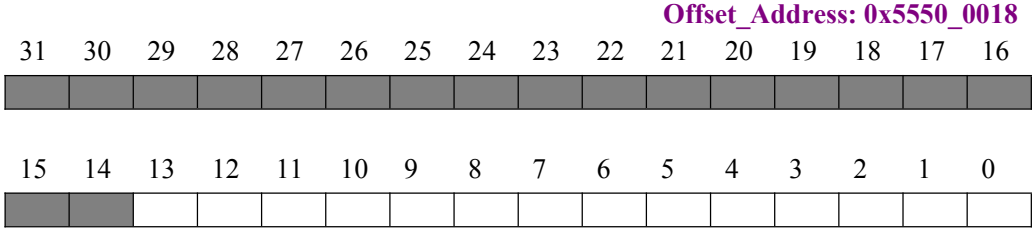
Offset_Address: 0x5550_0014



Bit	Type	Reset	Description
31-29	R	0	reserved
28-16	R/W	0	window_h_stop0 DCIC CRC 检测窗口 0 的行结束点
15-13	R	0	reserved
12-0	R/W	0	window_h_start0 DCIC CRC 检测窗口 0 的行起始点

DCIC_CFG6

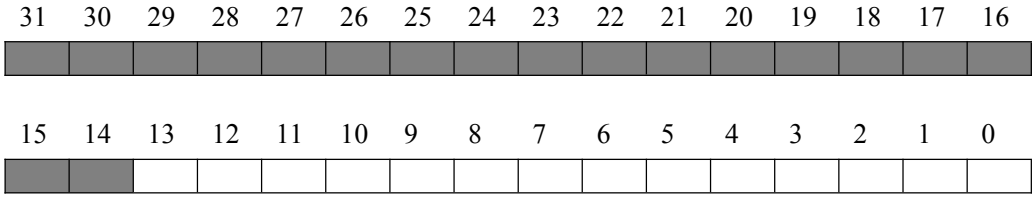
Offset_Address: 0x5550_0018



Bit	Type	Reset	Description
31-27	R	0	reserved
26-16	R/W	0	window_v_stop0 DCIC CRC 检测窗口 0 的场结束点
15-10	R	0	reserved
11-0	R/W	0	window_v_start0 DCIC CRC 检测窗口 0 的场起始点

DCIC_CFG7

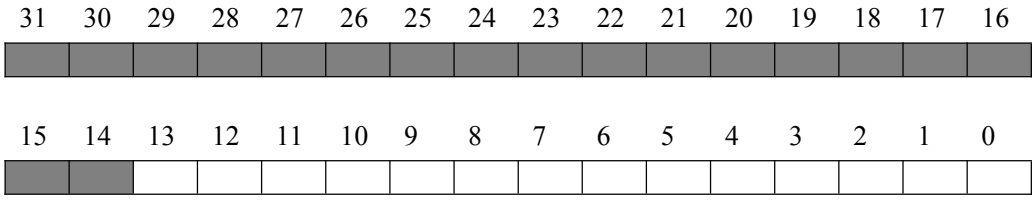
Offset_Address: 0x5550_001C



Bit	Type	Reset	Description
31-16	R/W	0	crc_g_reg0 DCIC CRC 检测窗口 0 当前帧的 Green CRC 比较值
15-0	R/W	0	crc_r_reg0 DCIC CRC 检测窗口 0 当前帧的 Red CRC 比较值

DCIC_CFG8

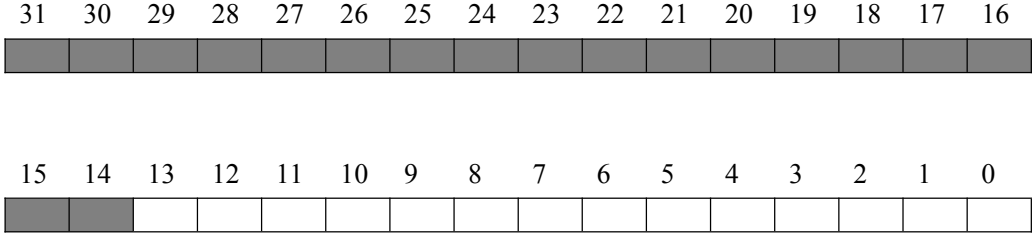
Offset_Address: 0x5550_0020



Bit	Type	Reset	Description
31-17	R	0	Reserved
16	R/W	0	error_det_en0 DCIC CRC 检测窗口 0 的使能位 0: disable; 1: enable;
15-0	R/W	0	crc_b_reg0 DCIC CRC 检测窗口 0 的 Blue CRC 比较值

DCIC_CFG9

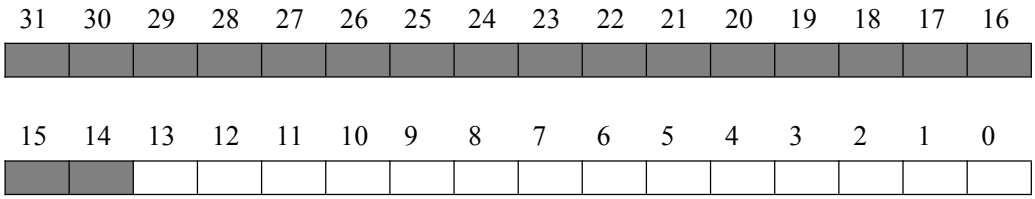
Offset_Address: 0x5550_0024



Bit	Type	Reset	Description
31-29	R	0	reserved
28-16	R/W	0	window_h_stop1 DCIC CRC 检测窗口 1 的行结束点
15-13	R	0	reserved
12-0	R/W	0	window_h_start1 DCIC CRC 检测窗口 1 的行起始点

DCIC_CFG10

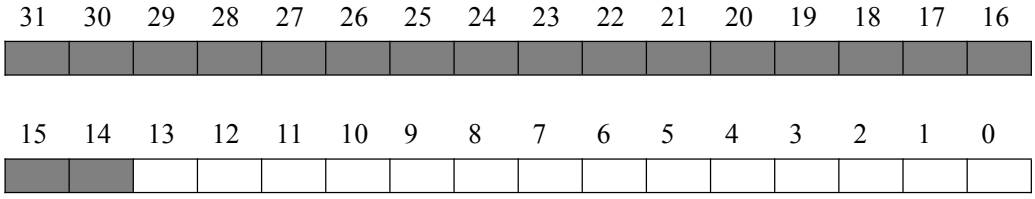
Offset_Address: 0x5550_0028



Bit	Type	Reset	Description
31-27	R	0	reserved
26-16	R/W	0	window_v_stop1 DCIC CRC 检测窗口 1 的场结束点
15-12	R	0	reserved
11-0	R/W	0	window_v_start1 DCIC CRC 检测窗口 1 的场起始点

DCIC_CFG11

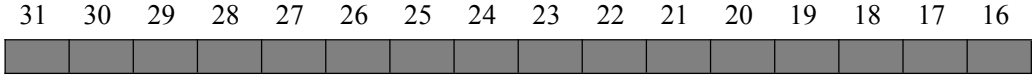
Offset_Address: 0x5550_002C

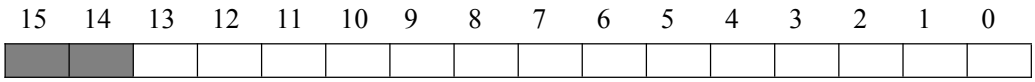


Bit	Type	Reset	Description
31-16	R/W	0	crc_g_reg1 DCIC CRC 检测窗口 1 当前帧的 Green CRC 比较值
15-0	R/W	0	crc_r_reg1 DCIC CRC 检测窗口 1 当前帧的 Red CRC 比较值

DCIC_CFG12

Offset_Address: 0x5550_0030

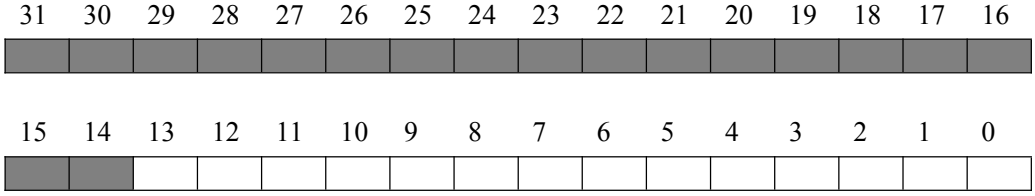




Bit	Type	Reset	Description
31-17	R	0	Reserved
16	R/W	0	error_det_en1 DCIC CRC 检测窗口 1 的使能位 0: disable; 1: enable;
15-0	R/W	0	crc_b_reg1 DCIC CRC 检测窗口 1 的 Blue CRC 比较值

DCIC_CFG13

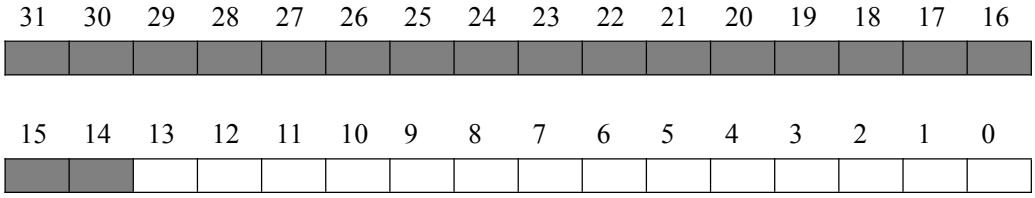
Offset_Address: 0x5550_0034



Bit	Type	Reset	Description
31-29	R	0	reserved
28-16	R/W	0	window_h_stop2 DCIC CRC 检测窗口 2 的行结束点
15-13	R	0	reserved
12-0	R/W	0	window_h_start2 DCIC CRC 检测窗口 2 的行起始点

DCIC_CFG14

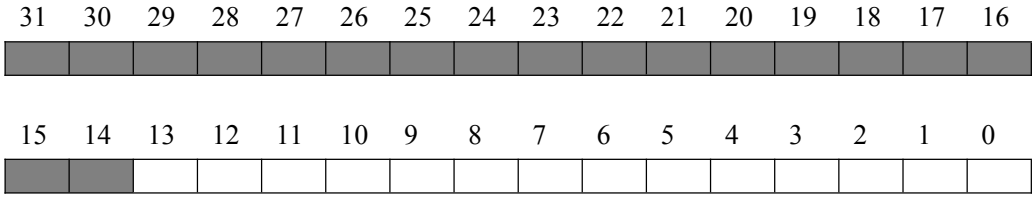
Offset_Address: 0x5550_0038



Bit	Type	Reset	Description
31-27	R	0	reserved
26-16	R/W	0	window_v_stop2 DCIC CRC 检测窗口 2 的场结束点
15-12	R	0	reserved
11-0	R/W	0	window_v_start2 DCIC CRC 检测窗口 2 的场起始点

DCIC_CFG15

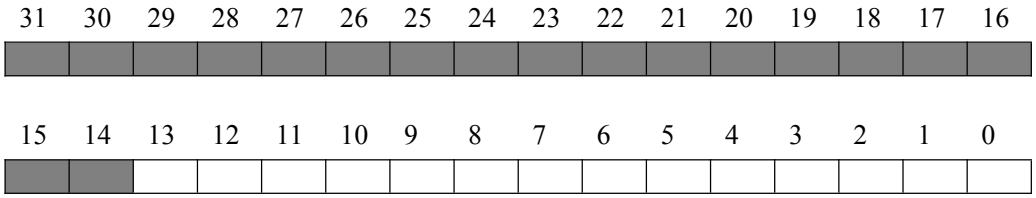
Offset_Address: 0x5550_003C



Bit	Type	Reset	Description
31-16	R/W	0	crc_g_reg2 DCIC CRC 检测窗口 2 当前帧的 Green CRC 比较值
15-0	R/W	0	crc_r_reg2 DCIC CRC 检测窗口 2 当前帧的 Red CRC 比较值

DCIC_CFG16

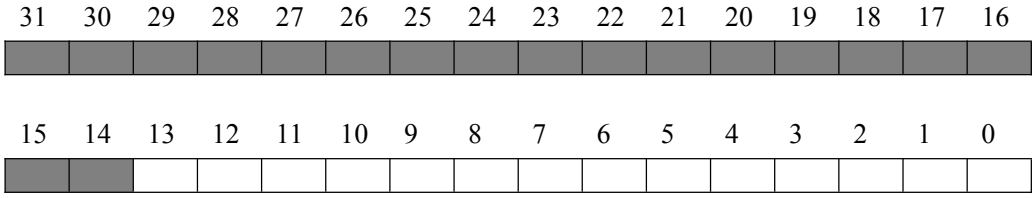
Offset_Address: 0x5550_0040



Bit	Type	Reset	Description
31-17	R	0	Reserved
16	R/W	0	error_det_en2 DCIC CRC 检测窗口 2 的使能位 0: disable; 1: enable;
15-0	R/W	0	crc_b_reg2 DCIC CRC 检测窗口 2 的 Blue CRC 比较值

DCIC_CFG17

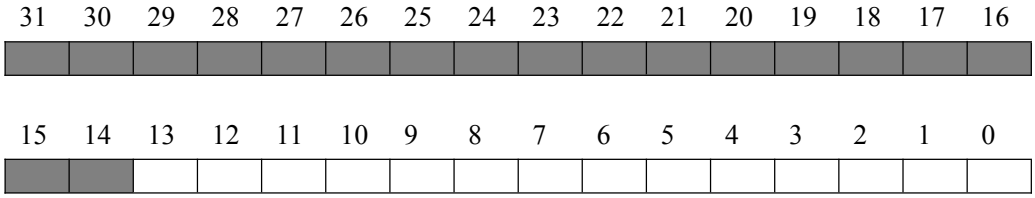
Offset_Address: 0x5550_0044



Bit	Type	Reset	Description
31-29	R	0	reserved
28-16	R/W	0	window_h_stop3 DCIC CRC 检测窗口 3 的行结束点
15-13	R	0	reserved
12-0	R/W	0	window_h_start3 DCIC CRC 检测窗口 3 的行起始点

DCIC_CFG18

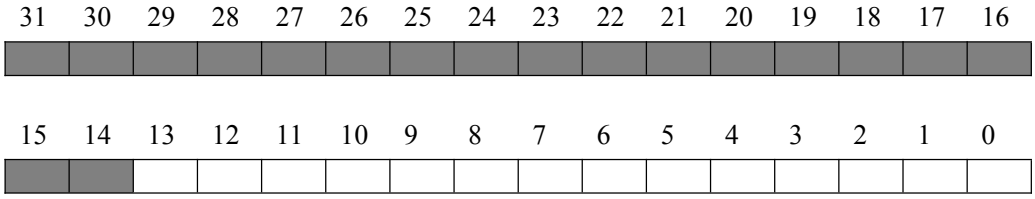
Offset_Address: 0x5550_0048



Bit	Type	Reset	Description
31-27	R	0	reserved
26-16	R/W	0	window_v_stop3 DCIC CRC 检测窗口 3 的场结束点
15-12	R	0	reserved
11-0	R/W	0	window_v_start3 DCIC CRC 检测窗口 3 的场起始点

DCIC_CFG19

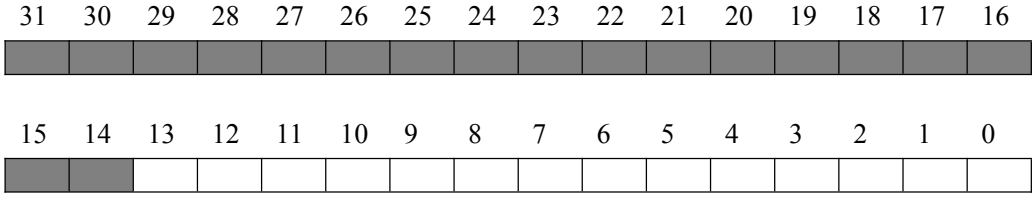
Offset_Address: 0x5550_004C



Bit	Type	Reset	Description
31-16	R/W	0	crc_g_reg3 DCIC CRC 检测窗口 3 当前帧的 Green CRC 比较值
15-0	R/W	0	crc_r_reg3 DCIC CRC 检测窗口 3 当前帧的 Red CRC 比较值

DCIC_CFG20

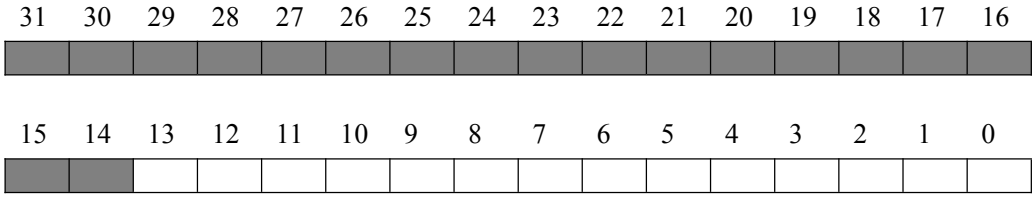
Offset_Address: 0x5550_0050



Bit	Type	Reset	Description
31-17	R	0	Reserved
16	R/W	0	error_det_en3 DCIC CRC 检测窗口 3 的使能位 0: disable; 1: enable;
15-0	R/W	0	crc_b_reg3 DCIC CRC 检测窗口 3 的 Blue CRC 比较值

DCIC_CFG21

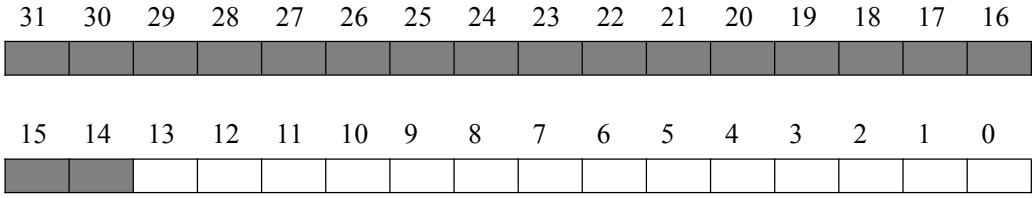
Offset_Address: 0x5550_0054



Bit	Type	Reset	Description
31-29	R	0	reserved
28-16	R/W	0	window_h_stop4 DCIC CRC 检测窗口 4 的行结束点
15-13	R	0	reserved
12-0	R/W	0	window_h_start4 DCIC CRC 检测窗口 4 的行起始点

DCIC_CFG22

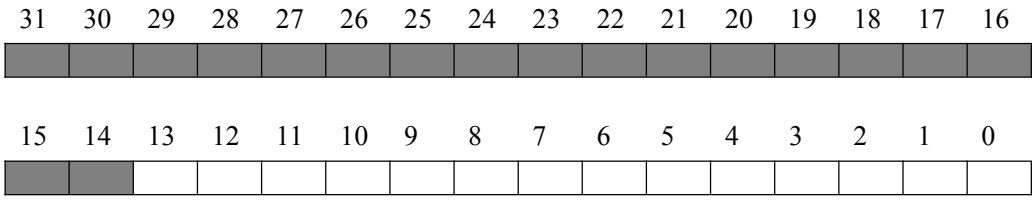
Offset_Address: 0x5550_0058



Bit	Type	Reset	Description
31-27	R	0	reserved
26-16	R/W	0	window_v_stop4 DCIC CRC 检测窗口 4 的场结束点
15-12	R	0	reserved
11-0	R/W	0	window_v_start4 DCIC CRC 检测窗口 4 的场起始点

DCIC_CFG23

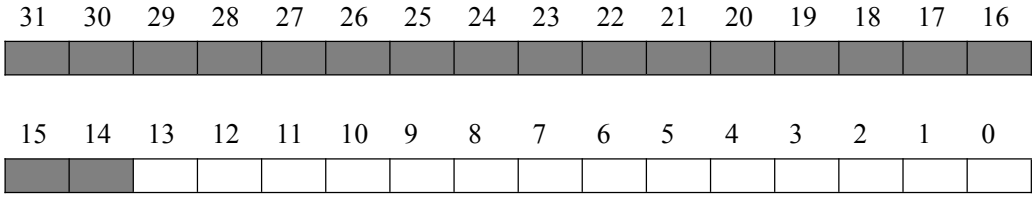
Offset_Address: 0x5550_005C



Bit	Type	Reset	Description
31-16	R/W	0	crc_g_reg4 DCIC CRC 检测窗口 4 当前帧的 Green CRC 比较值
15-0	R/W	0	crc_r_reg4 DCIC CRC 检测窗口 4 当前帧的 Red CRC 比较值

DCIC_CFG24

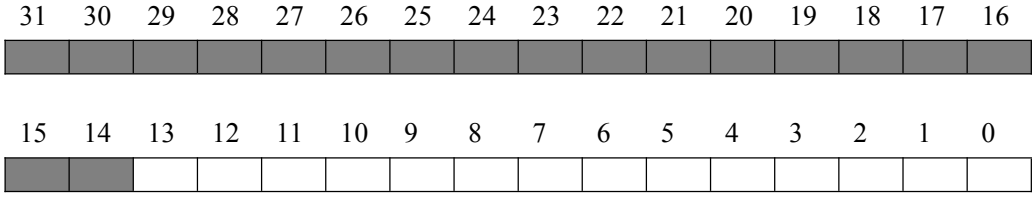
Offset_Address: 0x5550_0060



Bit	Type	Reset	Description
31-17	R	0	Reserved
16	R/W	0	error_det_en4 DCIC CRC 检测窗口 4 的使能位 0: disable; 1: enable;
15-0	R/W	0	crc_b_reg4 DCIC CRC 检测窗口 4 的 Blue CRC 比较值

DCIC_CFG25

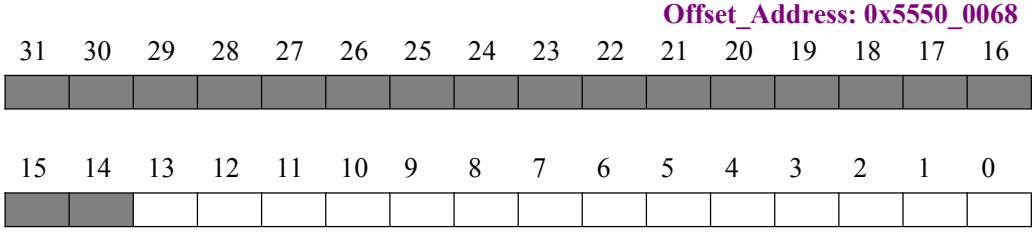
Offset_Address: 0x5550_0064



Bit	Type	Reset	Description
31-29	R	0	reserved
28-16	R/W	0	window_h_stop5 DCIC CRC 检测窗口 5 的行结束点
15-13	R	0	reserved
12-0	R/W	0	window_h_start5 DCIC CRC 检测窗口 5 的行起始点

DCIC_CFG26

Offset_Address: 0x5550_0068

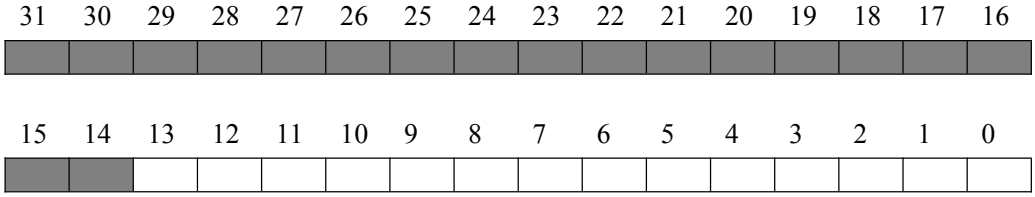


Bit	Type	Reset	Description
31-27	R	0	reserved
26-16	R/W	0	window_v_stop5 DCIC CRC 检测窗口 5 的场结束点
15-12	R	0	reserved

11-0	R/W	0	window_v_start5 DCIC CRC 检测窗口 5 的场起始点

DCIC_CFG27

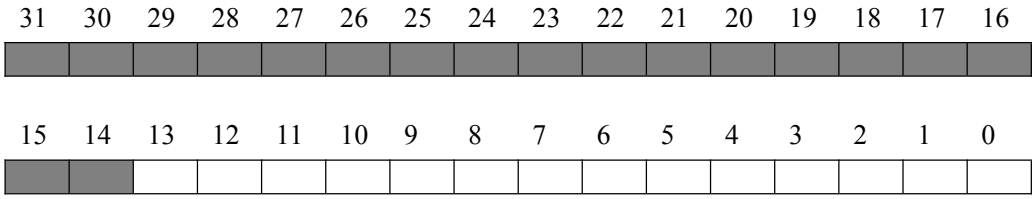
Offset_Address: 0x5550_006C



Bit	Type	Reset	Description
31-16	R/W	0	crc_g_reg5 DCIC CRC 检测窗口 5 当前帧的 Green CRC 比较值
15-0	R/W	0	crc_r_reg5 DCIC CRC 检测窗口 5 当前帧的 Red CRC 比较值

DCIC_CFG28

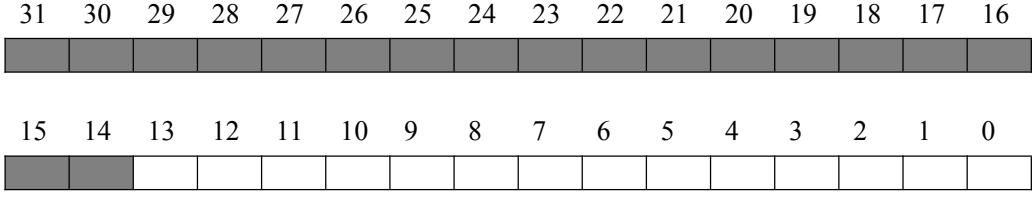
Offset_Address: 0x5550_0070



Bit	Type	Reset	Description
31-17	R	0	Reserved
16	R/W	0	error_det_en5 DCIC CRC 检测窗口 5 的使能位 0: disable; 1: enable;
15-0	R/W	0	crc_b_reg5 DCIC CRC 检测窗口 5 的 Blue CRC 比较值

DCIC_CFG29

Offset_Address: 0x5550_0074

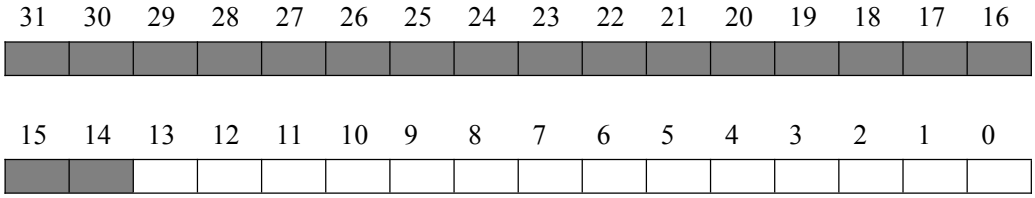


Bit	Type	Reset	Description
31-29	R	0	reserved
28-16	R/W	0	window_h_stop6 DCIC CRC 检测窗口 6 的行结束点

15-13	R	0	reserved
12-0	R/W	0	window_h_start6 DCIC CRC 检测窗口 6 的行起始点

DCIC_CFG30

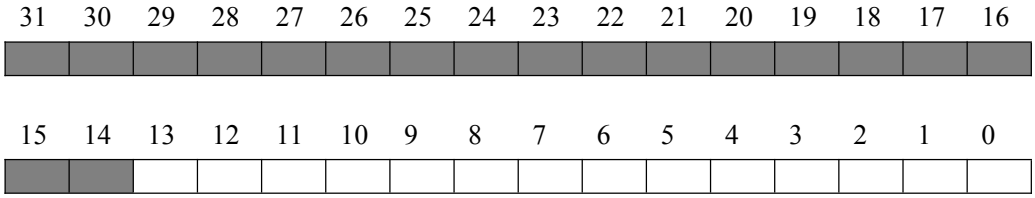
Offset_Address: 0x5550_0078



Bit	Type	Reset	Description
31-27	R	0	reserved
26-16	R/W	0	window_v_stop6 DCIC CRC 检测窗口 6 的场结束点
15-12	R	0	reserved
11-0	R/W	0	window_v_start6 DCIC CRC 检测窗口 6 的场起始点

DCIC_CFG31

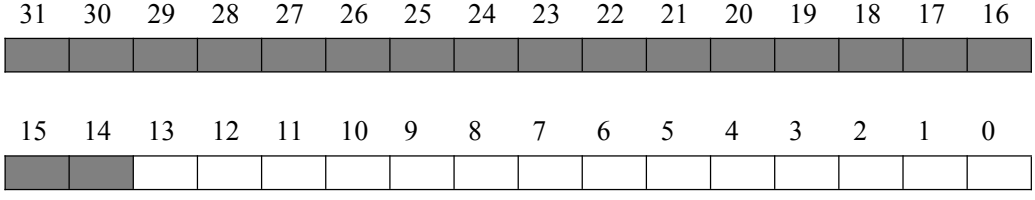
Offset_Address: 0x5550_007C



Bit	Type	Reset	Description
31-16	R/W	0	crc_g_reg6 DCIC CRC 检测窗口 6 当前帧的 Green CRC 比较值
15-0	R/W	0	crc_r_reg6 DCIC CRC 检测窗口 6 当前帧的 Red CRC 比较值

DCIC_CFG32

Offset_Address: 0x5550_0080

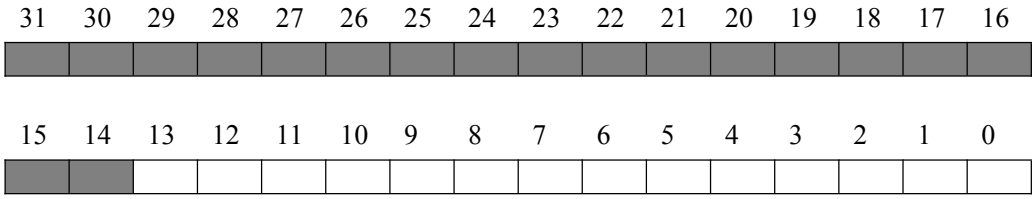


Bit	Type	Reset	Description
31-17	R	0	Reserved
16	R/W	0	error_det_en6

			DCIC CRC 检测窗口 6 的使能位 0: disable; 1: enable;
15-0	R/W	0	crc_b_reg6 DCIC CRC 检测窗口 6 的 Blue CRC 比较值

DCIC_CFG33

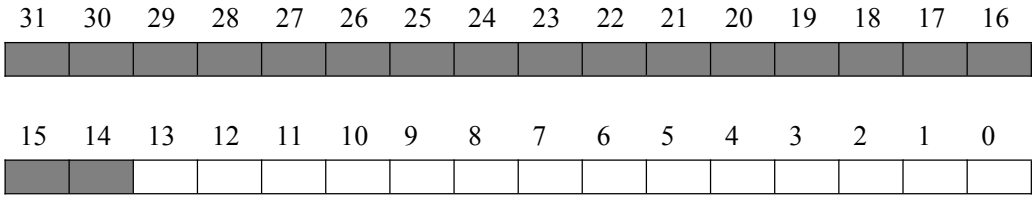
Offset_Address: 0x5550_0084



Bit	Type	Reset	Description
31-29	R	0	reserved
28-16	R/W	0	window_h_stop7 DCIC CRC 检测窗口 7 的行结束点
15-13	R	0	reserved
12-0	R/W	0	window_h_start7 DCIC CRC 检测窗口 7 的行起始点

DCIC_CFG34

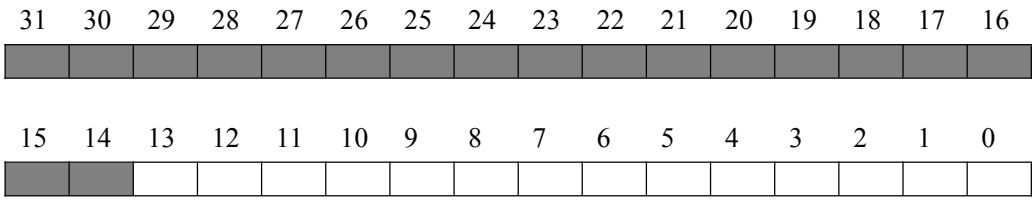
Offset_Address: 0x5550_0088



Bit	Type	Reset	Description
31-27	R	0	reserved
26-16	R/W	0	window_v_stop7 DCIC CRC 检测窗口 7 的场结束点
15-12	R	0	reserved
11-0	R/W	0	window_v_start7 DCIC CRC 检测窗口 7 的场起始点

DCIC_CFG35

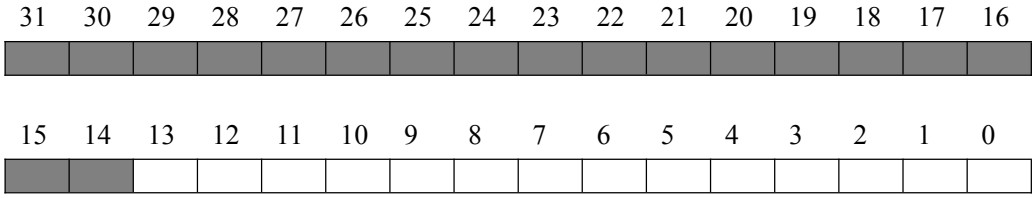
Offset_Address: 0x5550_008C



Bit	Type	Reset	Description
31-16	R/W	0	crc_g_reg7 DCIC CRC 检测窗口 7 当前帧的 Green CRC 比较值
15-0	R/W	0	crc_r_reg7 DCIC CRC 检测窗口 7 当前帧的 Red CRC 比较值

DCIC_CFG36

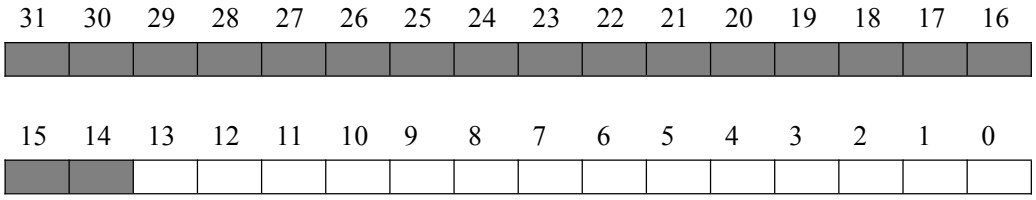
Offset_Address: 0x5550_0090



Bit	Type	Reset	Description
31-17	R	0	Reserved
16	R/W	0	error_det_en7 DCIC CRC 检测窗口 7 的使能位 0: disable; 1: enable;
15-0	R/W	0	crc_b_reg7 DCIC CRC 检测窗口 7 的 Blue CRC 比较值

DCIC_CFG37

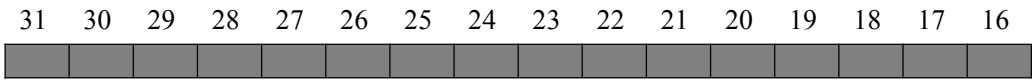
Offset_Address: 0x5550_0094

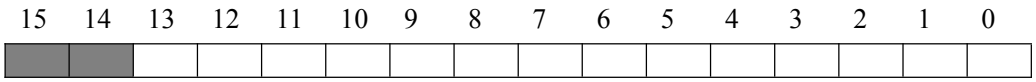


Bit	Type	Reset	Description
31-29	R	0	reserved
28-16	R/W	0	freeze_window_h_stop DCIC 冻结检测窗口的行结束点
15-13	R	0	reserved
12-0	R/W	0	freeze_window_h_start DCIC 冻结检测窗口的行起始点

DCIC_CFG38

Offset_Address: 0x5550_0098

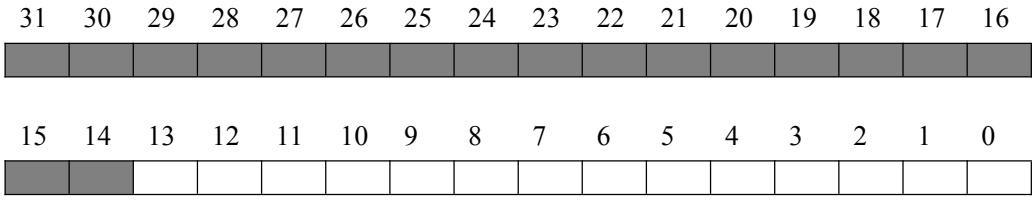




Bit	Type	Reset	Description
31-27	R	0	reserved
26-16	R/W	0	freeze_window_v_stop DCIC 冻结检测窗口的场结束点
15-12	R	0	reserved
11-0	R/W	0	freeze_window_v_start DCIC 冻结检测窗口的场起始点

DCIC_CFG39

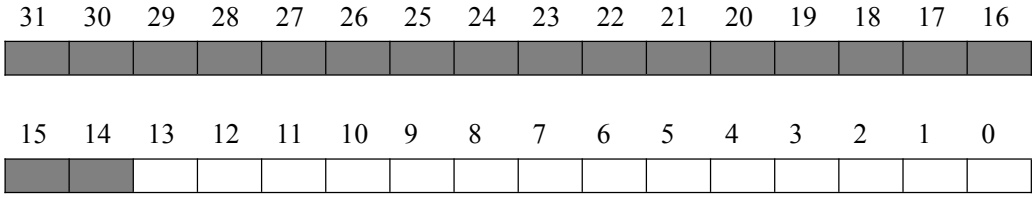
Offset_Address: 0x5550_009C



Bit	Type	Reset	Description
31-24	R/W	0	defreeze_det_th DCIC 冻结恢复检测阈值(帧数)
20	R/W	0	freeze_det_en DCIC 冻结检测使能 0: disable; 1: enable;
19-12	R/W	0	defreeze_th DCIC 冻结检测恢复阈值(帧数)
11-0	R/W	0	freeze_th DCIC 冻结检测阈值(帧数)

DCIC_CFG40

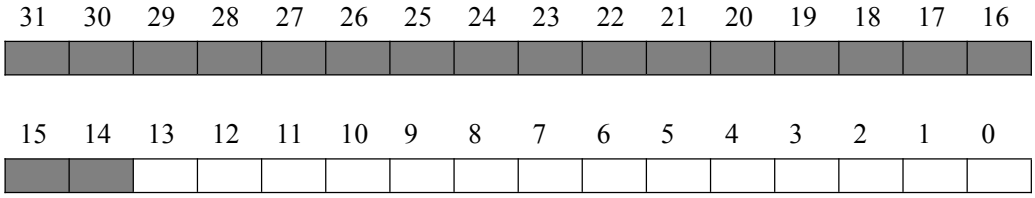
Offset_Address: 0x5550_00A0



Bit	Type	Reset	Description
31-28	R/W	0	histh34Eris_R DCIC 输入数据直方图统计 R 区间 3~4 划分值
27-16	R/W	0	histh23Eris_R DCIC 输入数据直方图统计 R 区间 2~3 划分值
15-8	R/W	0	histh12Eris_R DCIC 输入数据直方图统计 R 区间 1~2 划分值
7-0	R/W	0	histh01Eris_R DCIC 输入数据直方图统计 R 区间 0~1 划分值

DCIC_CFG41

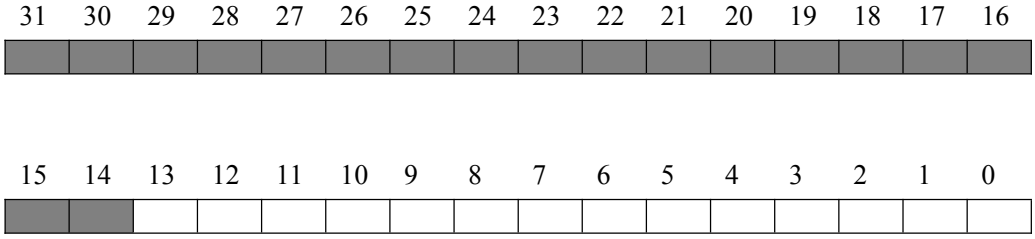
Offset_Address: 0x5550_00A4



Bit	Type	Reset	Description
31	R/W	0	histo_en_R DCIC 输入数据直方图统计 R 使能 0: disable; 1: enable;
30-22	R	0	Reserved
21-0	R/W	0	hisEris0_th_R DCIC 输入数据直方图统计 R 区间 0 阈值

DCIC_CFG42

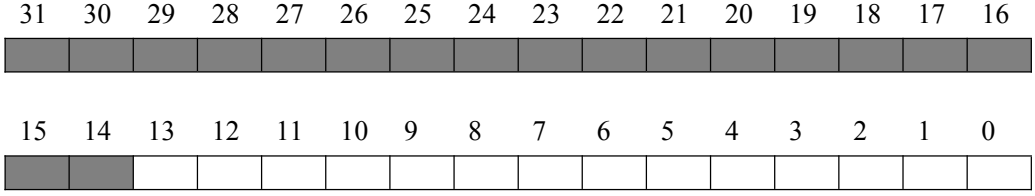
Offset_Address: 0x5550_00A8



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris1_th_R DCIC 输入数据直方图统计 R 区间 1 阈值

DCIC_CFG43

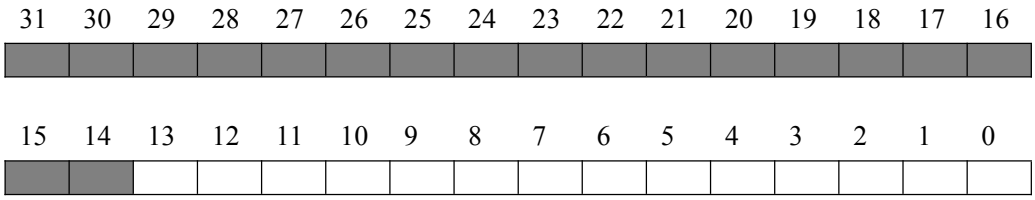
Offset_Address: 0x5550_00AC



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris2_th_R DCIC 输入数据直方图统计 R 区间 2 阈值

DCIC_CFG44

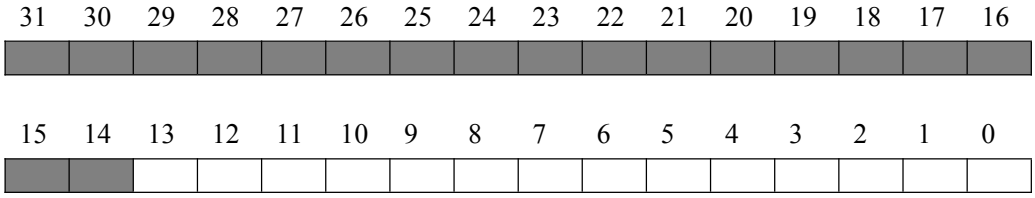
Offset_Address: 0x5550_00B0



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris3_th_R DCIC 输入数据直方图统计 R 区间 3 阈值

DCIC_CFG45

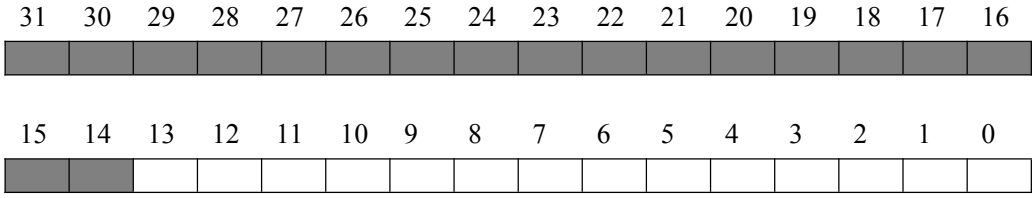
Offset_Address: 0x5550_00B4



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris4_th_R DCIC 输入数据直方图统计 R 区间 4 阈值

DCIC_CFG46

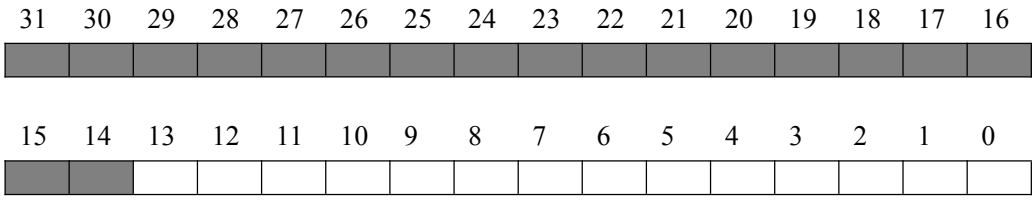
Offset_Address: 0x5550_00B8



Bit	Type	Reset	Description
31-28	R/W	0	histh34Eris_G DCIC 输入数据直方图统计 G 区间 3~4 划分值
27-16	R/W	0	histh23Eris_G DCIC 输入数据直方图统计 G 区间 2~3 划分值
15-8	R/W	0	histh12Eris_G DCIC 输入数据直方图统计 G 区间 1~2 划分值
7-0	R/W	0	histh01Eris_G DCIC 输入数据直方图统计 G 区间 0~1 划分值

DCIC_CFG47

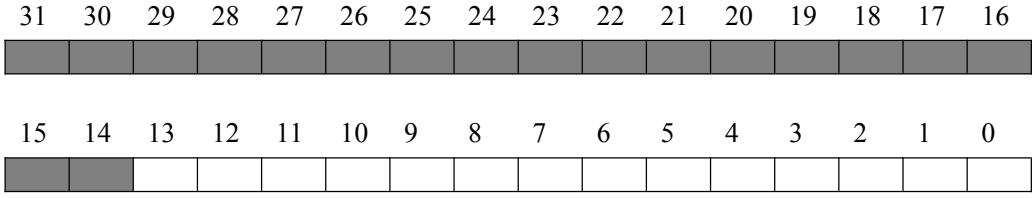
Offset_Address: 0x5550_00BC



Bit	Type	Reset	Description
31	R/W	0	histo_en_G DCIC 输入数据直方图统计 G 使能 0: disable; 1: enable;
30-22	R	0	Reserved
21-0	R/W	0	hisEris0_th_G DCIC 输入数据直方图统计 G 区间 0 阈值

DCIC_CFG48

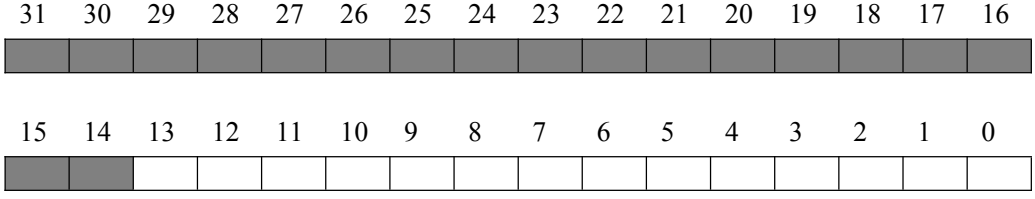
Offset_Address: 0x5550_00C0



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris1_th_G DCIC 输入数据直方图统计 G 区间 1 阈值

DCIC_CFG49

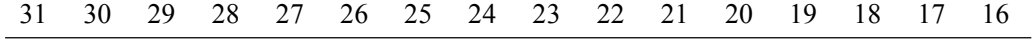
Offset_Address: 0x5550_00C4

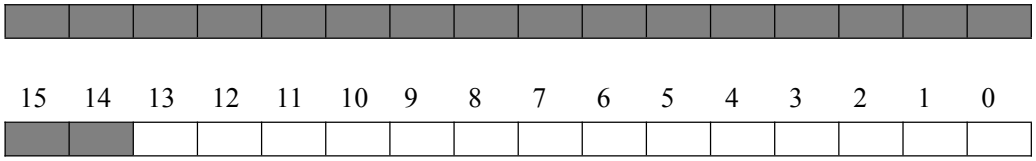


Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris2_th_G DCIC 输入数据直方图统计 G 区间 2 阈值

DCIC_CFG50

Offset_Address: 0x5550_00C8

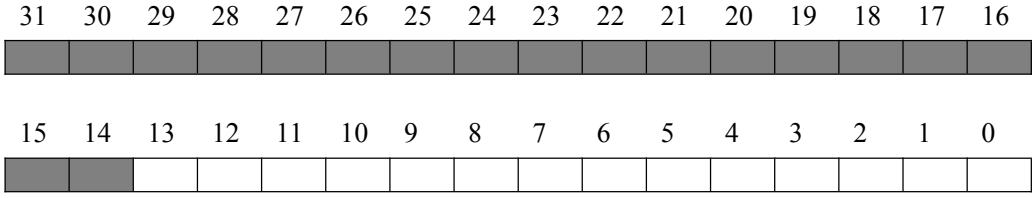




Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris3_th_G DCIC 输入数据直方图统计 G 区间 3 阈值

DCIC_CFG51

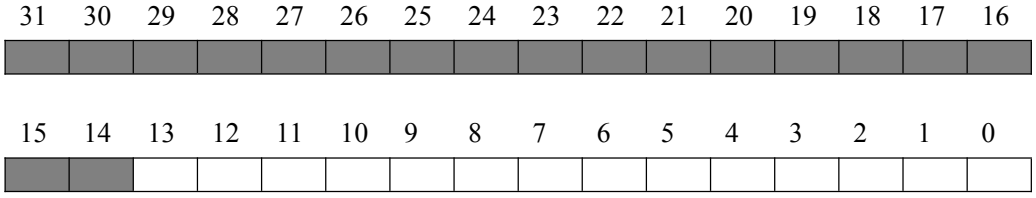
Offset_Address: 0x5550_00CC



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris4_th_G DCIC 输入数据直方图统计 G 区间 4 阈值

DCIC_CFG52

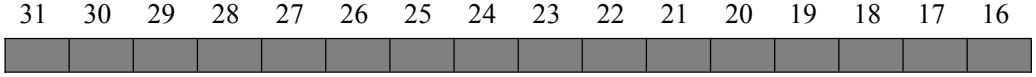
Offset_Address: 0x5550_00D0

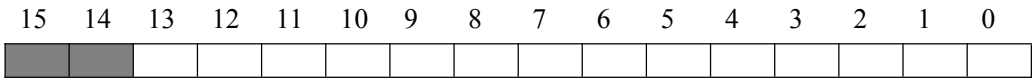


Bit	Type	Reset	Description
31-28	R/W	0	histh34Eris_B DCIC 输入数据直方图统计 B 区间 3~4 划分值
27-16	R/W	0	histh23Eris_B DCIC 输入数据直方图统计 B 区间 2~3 划分值
15-8	R/W	0	histh12Eris_B DCIC 输入数据直方图统计 B 区间 1~2 划分值
7-0	R/W	0	histh01Eris_B DCIC 输入数据直方图统计 B 区间 0~1 划分值

DCIC_CFG53

Offset_Address: 0x5550_00D4

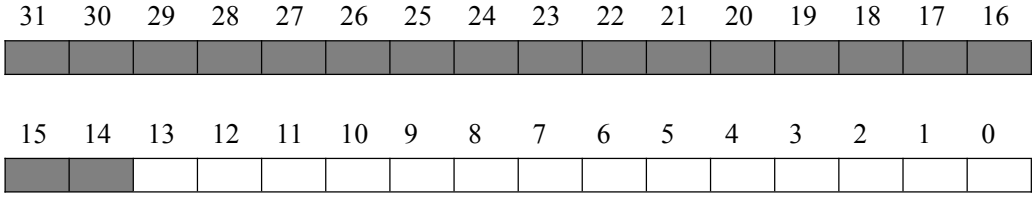




Bit	Type	Reset	Description
31	R/W	0	histo_en_B DCIC 输入数据直方图统计 B 使能 0: disable; 1: enable;
30-22	R	0	Reserved
21-0	R/W	0	hisEris0_th_B DCIC 输入数据直方图统计 B 区间 0 阈值

DCIC_CFG54

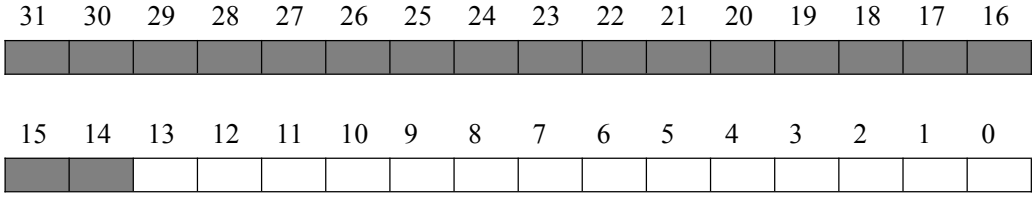
Offset_Address: 0x5550_00D8



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris1_th_B DCIC 输入数据直方图统计 B 区间 1 阈值

DCIC_CFG55

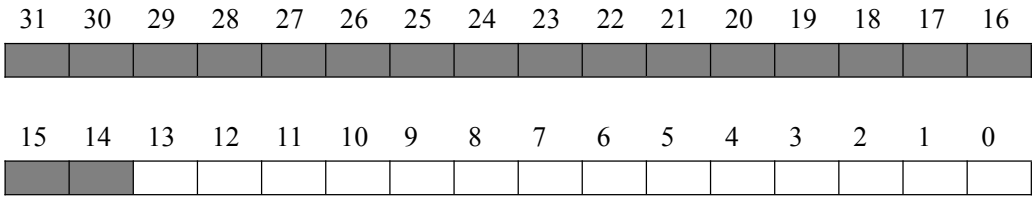
Offset_Address: 0x5550_00DC



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris2_th_B DCIC 输入数据直方图统计 B 区间 2 阈值

DCIC_CFG56

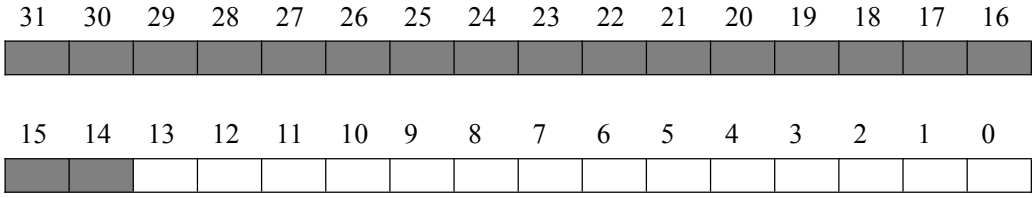
Offset_Address: 0x5550_00E0



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris3_th_B DCIC 输入数据直方图统计 B 区间 3 阈值

DCIC_CFG57

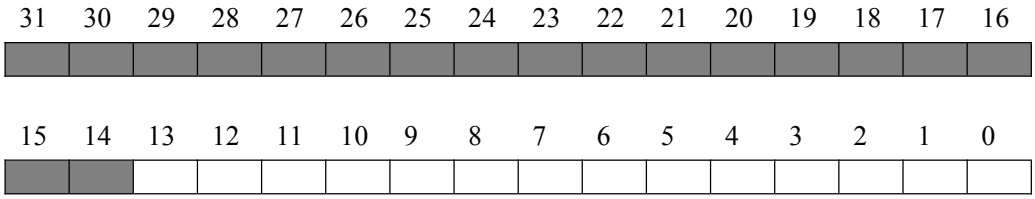
Offset_Address: 0x5550_00E4



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris4_th_B DCIC 输入数据直方图统计 B 区间 4 阈值

DCIC_CFG58

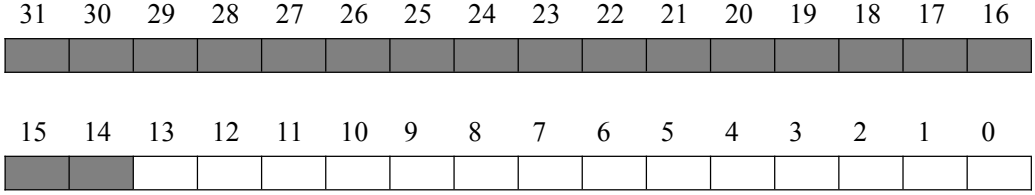
Offset_Address: 0x5550_00E8



Bit	Type	Reset	Description
31-28	R/W	0	histh34Eris_Y DCIC 输入数据直方图统计 Y 区间 3~4 划分值
27-16	R/W	0	histh23Eris_Y DCIC 输入数据直方图统计 Y 区间 2~3 划分值
15-8	R/W	0	histh12Eris_Y DCIC 输入数据直方图统计 Y 区间 1~2 划分值
7-0	R/W	0	histh01Eris_Y DCIC 输入数据直方图统计 Y 区间 0~1 划分值

DCIC_CFG59

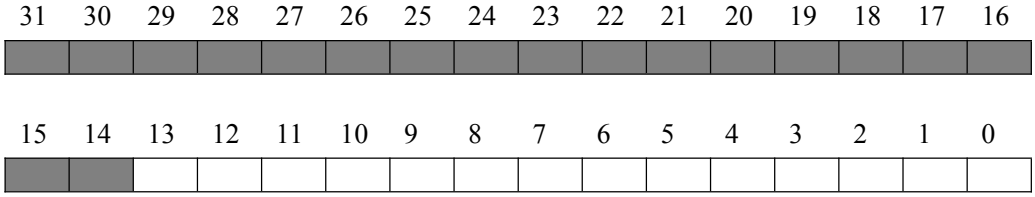
Offset_Address: 0x5550_00EC



Bit	Type	Reset	Description
31	R/W	0	histo_en_Y DCIC 输入数据直方图统计 Y 使能 0: disable; 1: enable;
30-22	R	0	Reserved
21-0	R/W	0	hisEris0_th_Y DCIC 输入数据直方图统计 Y 区间 0 阈值

DCIC_CFG60

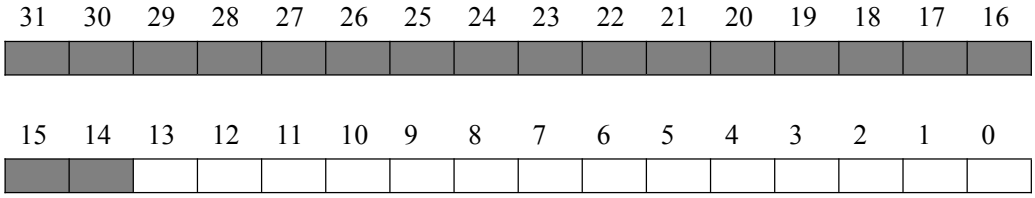
Offset_Address: 0x5550_00F0



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris1_th_Y DCIC 输入数据直方图统计 Y 区间 1 阈值

DCIC_CFG61

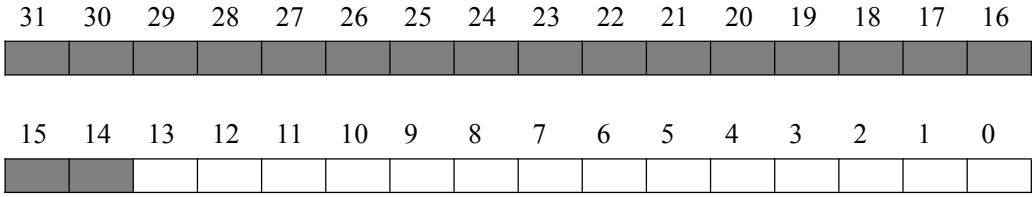
Offset_Address: 0x5550_00F4



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris2_th_Y DCIC 输入数据直方图统计 Y 区间 2 阈值

DCIC_CFG62

Offset_Address: 0x5550_00F8



Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris3_th_Y

			DCIC 输入数据直方图统计 Y 区间 3 阈值

DCIC_CFG63

Offset_Address: 0x5550_00FC

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

Bit	Type	Reset	Description
31-22	R	0	Reserved
21-0	R/W	0	hisEris4_th_Y DCIC 输入数据直方图统计 Y 区间 4 阈值

DCIC_CFG64

Offset_Address: 0x5550_0100

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

Bit	Type	Reset	Description
31-28	R	0	reserved
27-16	R/W	0	hv_regen_DEN_h_fall DCIC 重建 DE 信号的结束点数
15-12	R	0	reserved
11-0	R/W	0	hv_regen_DEN_h_rise DCIC 重建 DE 信号的起始点数

DCIC_CFG65

Offset_Address: 0x5550_0104

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

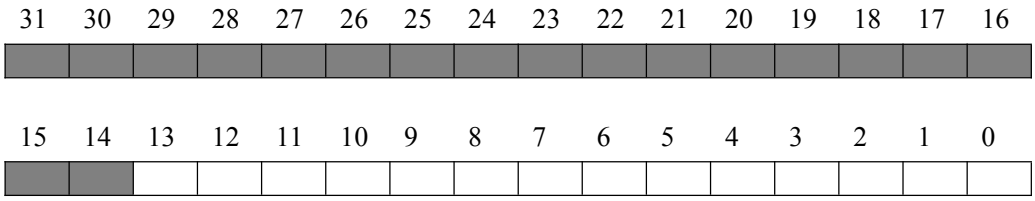
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-28	R	0	reserved
27-16	R/W	0	hv_regen_DEN_v_fall DCIC 重建 DE 信号的结束行数
15-12	R	0	reserved

11-0	R/W	0	hv_regen_DEN_v_rise DCIC 重建 DE 信号的起始行数

DCIC_CFG66

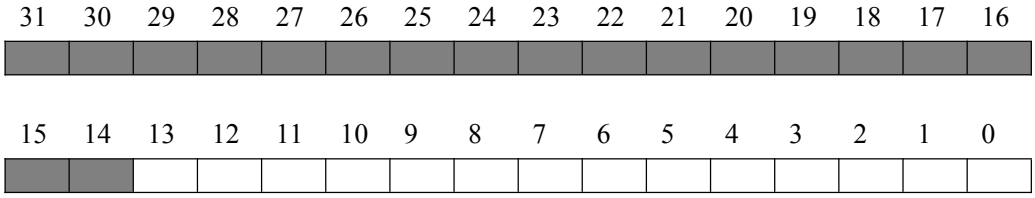
Offset_Address: 0x5550_0108



Bit	Type	Reset	Description
31-28	R	0	reserved
27-16	R/W	0	hv_regen_LPS DCIC 重建行场总行数
15-12	R	0	reserved
11-0	R/W	0	hv_regen_CPL DCIC 重建行场总点数

DCIC_CFG67

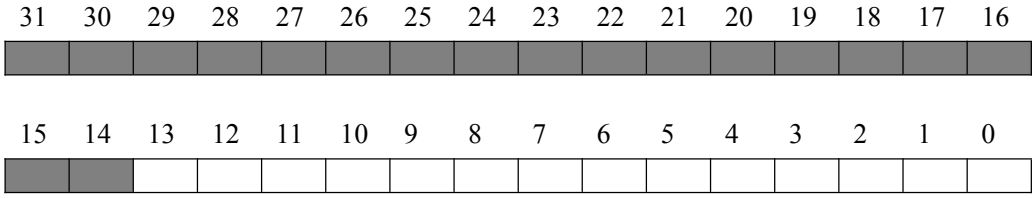
Offset_Address: 0x5550_010C



Bit	Type	Reset	Description
31-28	R	0	reserved
27-16	R/W	0	hv_regen_HSW_fall DCIC 重建行同步信号的结束点数
15-12	R	0	reserved
11-0	R/W	0	hv_regen_HSW_rise DCIC 重建行同步信号的起始点数

DCIC_CFG68

Offset_Address: 0x5550_0110

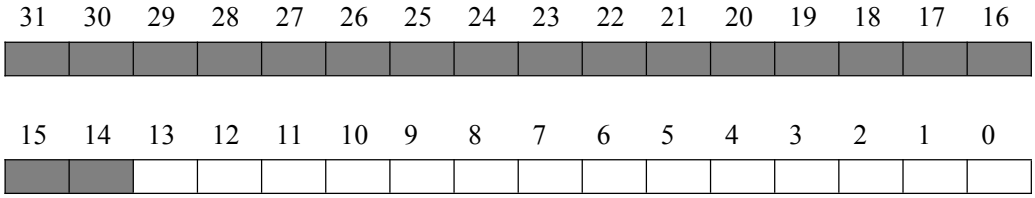


Bit	Type	Reset	Description

31-28	R	0	reserved
27-16	R/W	0	hv_regen_VSW_v_fall DCIC 重建场同步信号的结束行数
15-12	R	0	reserved
11-0	R/W	0	hv_regen_VSW_v_rise DCIC 重建场同步信号的起始行数

DCIC_CFG69

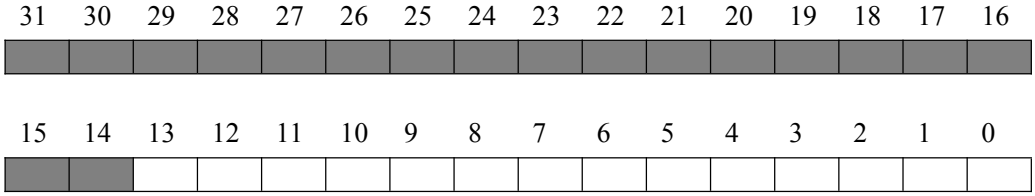
Offset_Address: 0x5550_0114



Bit	Type	Reset	Description
31-28	R	0	reserved
27-16	R/W	0	hv_regen_VSW_h_fall DCIC 重建场同步信号的结束点数
15-12	R	0	reserved
11-0	R/W	0	hv_regen_VSW_h_rise DCIC 重建场同步信号的起始点数

DCIC_CFG70

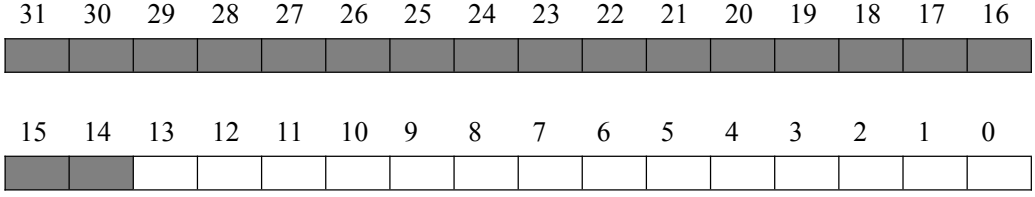
Offset_Address: 0x5550_0118



Bit	Type	Reset	Description
31-0	R/W	0	v_cnt_pre_th_l DCIC 检测总点数阈值范围最小值

DCIC_CFG71

Offset_Address: 0x5550_011C

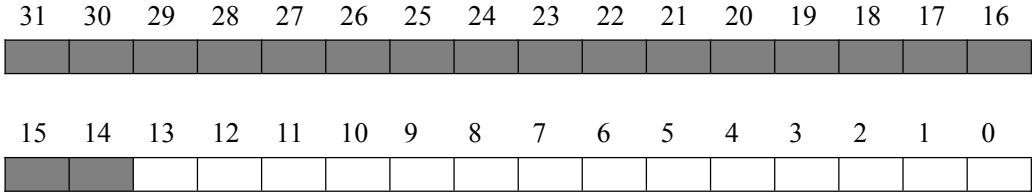


Bit	Type	Reset	Description
31-0	R/W	0	v_cnt_pre_th_h

			DCIC 检测总点数阈值范围最大值

DCIC_CFG72

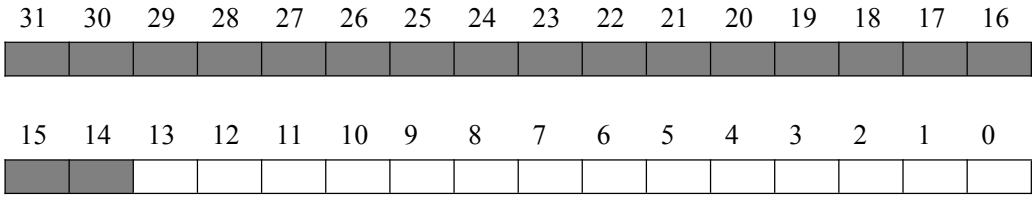
Offset_Address: 0x5550_0120



Bit	Type	Reset	Description
31-11	R	0	Reserved
10-8	R/W	0	de_mask_th DCIC DE 信号 MASK 阈值(0 表示 MASK 1 个点)
7	R	0	Reserved
6	R/W	0	de_mask_en DCIC DE 信号 MASK 使能位
5	R/W	0	hv_regen_load_timing_cfg DCIC 重建行场时序配置更新使能位
4	R/W	0	hv_regen_timing_init DCIC 重建行场时序配置初始化使能位
3	R	0	reserved
2	R/W	0	hv_active_enable_1 DCIC 重建行场检测 1 行/场有无信号 0: disable; 1: enable;
1	R/W	0	hv_active_enable_0 DCIC 重建行场检测 0 行/场有无信号 0: disable; 1: enable;
0	R/W	0	hv_regen_enable DCIC 重建行场使能位 0: disable; 1: enable;

DCIC_CFG73

Offset_Address: 0x5550_0124

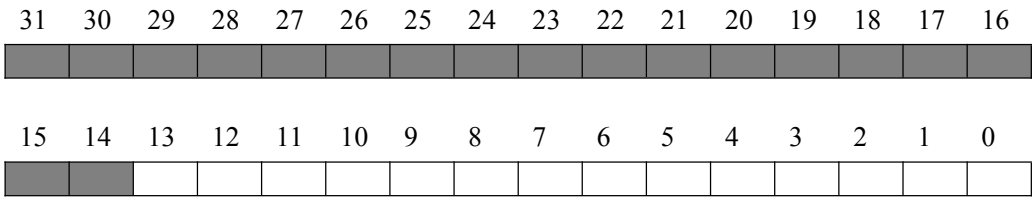


Bit	Type	Reset	Description
31	R/W	0	DCIC bypass 0: output from DCIC; 1: output from LCD;
30-29	R	0	reserved
28	R/W	0	DCIC_mux_bitorder0 0: normal order; 1: bitorder inv;

27-25	R/W	0	DCIC_mux_rgb_mode1 000: RGB; 001: RBG; 010: GRB; 011: GBR; 100: BRG; 101: BGR;
24-22	R/W	0	DCIC_mux_rgb_mode0 000: RGB; 001: RBG; 010: GRB; 011: GBR; 100: BRG; 101: BGR;
21	R/W	0	DCIC_mux_de_inv 0: not inv; 1: de inv;
20	R/W	0	DCIC_mux_vsyn_inv 0: not inv; 1: vsyn inv;
19	R/W	0	DCIC_mux_hsyn_inv 0: not inv; 1: hsyn inv;
18	R/W	0	DCIC_mux_sel_reg DCIC 输出数据源选择 0: DCIC; 1: OSD;
16-0	R	0	Reserved

DCIC_CFG74

Offset_Address: 0x5550_0128

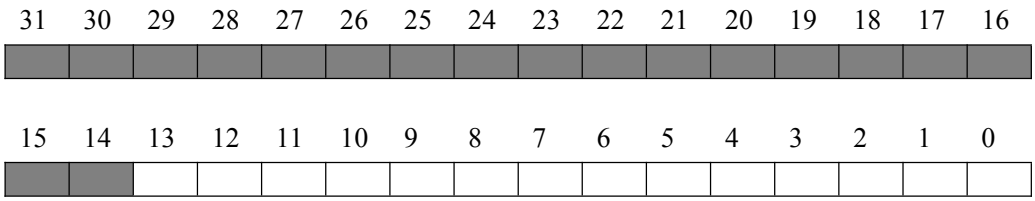


Bit	Type	Reset	Description
31	R/W	0	hv_mode DCIC 输入信号模式选择 0: DE 模式； 1: HV 模式；
30	R/W	0	hv_det_de_sel DCIC 输入信号 DE 选择 0: DE； 1: HSYN；
39-28	R	0	Reserved
27-16	R/W	0	hv_mode_active_pix DCIC 输入信号 HV 模式有效点数

15-12	R	0	Reserved
11-0	R/W	0	hv_mode_h_start DCIC 输入信号 HV 模式有效点起始值(HBP)

DCIC_CFG75

Offset_Address: 0x5550_012C

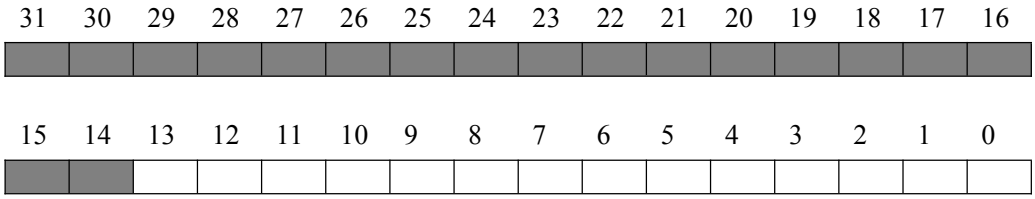


Bit	Type	Reset	Description
31	R/W	0	font_ram_sel 0: CPU/MCU access; 1: dcic access;
30-28	R	0	Reserved
27-16	R/W	0	hv_mode_active_line DCIC 输入信号 HV 模式有效行数
15-12	R	0	Reserved
11-0	R/W	0	hv_mode_v_start DCIC 输入信号 HV 模式有效行起始值(VBP)

DCIC_INT_EN0

DCIC_fault_intr

Offset_Address: 0x5550_01C0



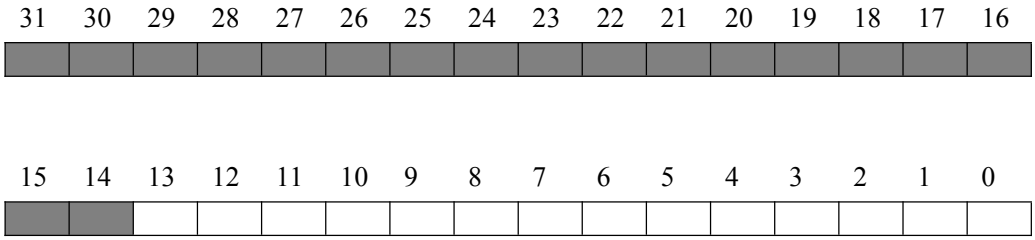
Bit	Type	Reset	Description
31-9	R	0	Reserved
8	R/W	0	hv_regen_vsyn_out_int DCIC 场同步信号中断使能(fault_int_4) 0: disable; 1: enable;
7	R/W	0	defreeze DCIC 冻结恢复中断使能(fault_int_2) 0: disable; 1: enable;
6	R/W	0	freeze DCIC 冻结中断使能(fault_int_2) 0: disable; 1: enable;
5	R/W	0	hv_regen_vsyn_error DCIC LCD 场同步信号异常中断使能(fault_int_0) 0: disable; 1: enable;
4	R/W	0	hv_status DCIC 行/场信号有无检测中断使能(fault_int_0) 0: disable; 1: enable;

3	R/W	0	tol_line_num_pf_chag DCIC 总行数变化中断使能(fault_int_0) 0: disable; 1: enable;
2	R/W	0	act_line_num_pf_chag DCIC 有效行数变化中断使能(fault_int_0) 0: disable; 1: enable;
1	R/W	0	tol_pix_num_pl_chag DCIC 总点数变化中断使能(fault_int_0) 0: disable; 1: enable;
0	R/W	0	act_pix_num_pl_chag DCIC 有效点数变化中断使能(fault_int_0) 0: disable; 1: enable;

DCIC_INT_EN1

DCIC_fault_intr

Offset_Address: 0x5550_01C4



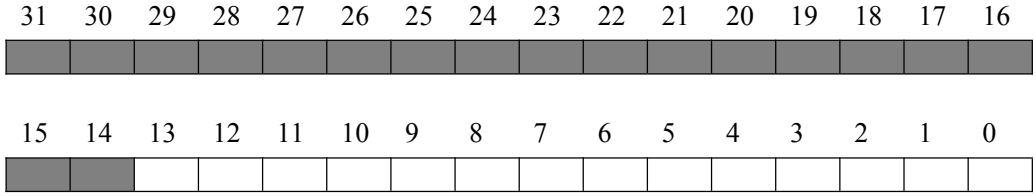
Bit	Type	Reset	Description
31-17	R	0	reserved
16	R/W	0	osd_error_int OSD 错误中断使能(fault_int_1) 0: disable; 1: enable;
15	R/W	0	crc_rgb_error7 DCIC 窗口 7 CRC 错误中断使能(fault_int_1) 0: disable; 1: enable;
14	R/W	0	crc_finish7 DCIC 窗口 7 CRC 计算完成中断使能(fault_int_1) 0: disable; 1: enable;
13	R/W	0	crc_rgb_error6 DCIC 窗口 6 CRC 错误中断使能(fault_int_1) 0: disable; 1: enable;
12	R/W	0	crc_finish6 DCIC 窗口 6 CRC 计算完成中断使能(fault_int_1) 0: disable; 1: enable;
11	R/W	0	crc_rgb_error5 DCIC 窗口 5 CRC 错误中断使能(fault_int_1) 0: disable; 1: enable;
10	R/W	0	crc_finish5 DCIC 窗口 5 CRC 计算完成中断使能(fault_int_1) 0: disable; 1: enable;
9	R/W	0	crc_rgb_error4 DCIC 窗口 4 CRC 错误中断使能(fault_int_1) 0: disable; 1: enable;
8	R/W	0	crc_finish4 DCIC 窗口 4 CRC 计算完成中断使能(fault_int_1)

			0: disable; 1: enable;
7	R/W	0	crc_rgb_error3 DCIC 窗口 3 CRC 错误中断使能(fault_int_1) 0: disable; 1: enable;
6	R/W	0	crc_finish3 DCIC 窗口 3 CRC 计算完成中断使能(fault_int_1) 0: disable; 1: enable;
5	R/W	0	crc_rgb_error2 DCIC 窗口 2 CRC 错误中断使能(fault_int_1) 0: disable; 1: enable;
4	R/W	0	crc_finish2 DCIC 窗口 2 CRC 计算完成中断使能(fault_int_1) 0: disable; 1: enable;
3	R/W	0	crc_rgb_error1 DCIC 窗口 1 CRC 错误中断使能(fault_int_1) 0: disable; 1: enable;
2	R/W	0	crc_finish1 DCIC 窗口 1 CRC 计算完成中断使能(fault_int_1) 0: disable; 1: enable;
1	R/W	0	crc_rgb_error0 DCIC 窗口 0 CRC 错误中断使能(fault_int_1) 0: disable; 1: enable;
0	R/W	0	crc_finish0 DCIC 窗口 0 CRC 计算完成中断使能(fault_int_1) 0: disable; 1: enable;

DCIC_INT_EN2

DCIC_fault_intr

Offset_Address: 0x5550_01C8



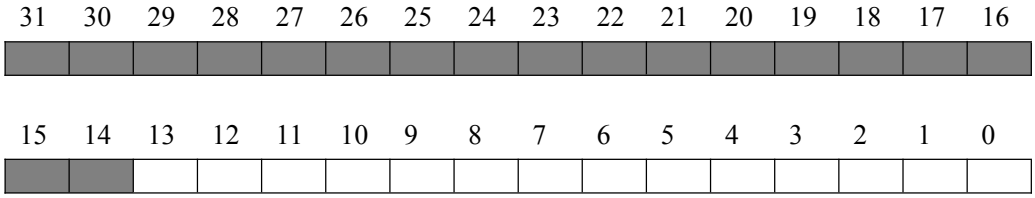
Bit	Type	Reset	Description
31-20	R	0	Reserved
19	R/W	0	hisEris4_comp_err_Y DCIC 直方图 Y 区间 4 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
18	R/W	0	hisEris3_comp_err_Y DCIC 直方图 Y 区间 3 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
17	R/W	0	hisEris2_comp_err_Y DCIC 直方图 Y 区间 2 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
16	R/W	0	hisEris1_comp_err_Y DCIC 直方图 Y 区间 1 比较错误中断使能(fault_int_3) 0: disable; 1: enable;

15	R/W	0	hisEris0_comp_err_Y DCIC 直方图 Y 区间 0 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
14	R/W	0	hisEris4_comp_err_B DCIC 直方图 B 区间 4 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
13	R/W	0	hisEris3_comp_err_B DCIC 直方图 B 区间 3 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
12	R/W	0	hisEris2_comp_err_B DCIC 直方图 B 区间 2 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
11	R/W	0	hisEris1_comp_err_B DCIC 直方图 B 区间 1 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
10	R/W	0	hisEris0_comp_err_B DCIC 直方图 B 区间 0 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
9	R/W	0	hisEris4_comp_err_G DCIC 直方图 G 区间 4 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
8	R/W	0	hisEris3_comp_err_G DCIC 直方图 G 区间 3 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
7	R/W	0	hisEris2_comp_err_G DCIC 直方图 G 区间 2 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
6	R/W	0	hisEris1_comp_err_G DCIC 直方图 G 区间 1 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
5	R/W	0	hisEris0_comp_err_G DCIC 直方图 G 区间 0 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
4	R/W	0	hisEris4_comp_err_R DCIC 直方图 R 区间 4 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
3	R/W	0	hisEris3_comp_err_R DCIC 直方图 R 区间 3 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
2	R/W	0	hisEris2_comp_err_R DCIC 直方图 R 区间 2 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
1	R/W	0	hisEris1_comp_err_R DCIC 直方图 R 区间 1 比较错误中断使能(fault_int_3) 0: disable; 1: enable;
0	R/W	0	hisEris0_comp_err_R DCIC 直方图 R 区间 0 比较错误中断使能(fault_int_3) 0: disable; 1: enable;

DCIC_INT_CLR0

Interrupt clear register, write 1,then write 0 to clear the interrupt.

Offset_Address: 0x5550_01E0



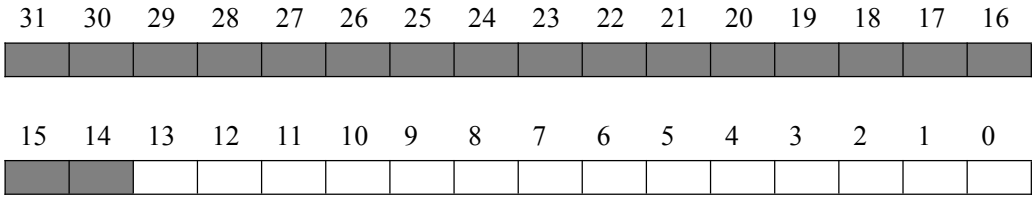
Bit	Type	Reset	Description
31	R	0	Reserved
30	R/W	0	hv_regen_vsyn_error_int_clr DCIC 行/场异常中断清除位
29	R/W	0	hv_regen_vsyn_out_int_clr DCIC 场中断清除位
28	R	0	Reserved
27	R/W	0	hisEris4_Y_int_clr DCIC 直方图 Y 区间 4 比较错误中断清除位
26	R/W	0	hisEris3_Y_int_clr DCIC 直方图 Y 区间 3 比较错误中断清除位
25	R/W	0	hisEris2_Y_int_clr DCIC 直方图 Y 区间 2 比较错误中断清除位
24	R/W	0	hisEris1_Y_int_clr DCIC 直方图 Y 区间 1 比较错误中断清除位
23	R/W	0	hisEris0_Y_int_clr DCIC 直方图 Y 区间 0 比较错误中断清除位
22	R/W	0	hisEris4_B_int_clr DCIC 直方图 B 区间 4 比较错误中断清除位
21	R/W	0	hisEris3_B_int_clr DCIC 直方图 B 区间 3 比较错误中断清除位
20	R/W	0	hisEris2_B_int_clr DCIC 直方图 B 区间 2 比较错误中断清除位
19	R/W	0	hisEris1_B_int_clr DCIC 直方图 B 区间 1 比较错误中断清除位
18	R/W	0	hisEris0_B_int_clr DCIC 直方图 B 区间 0 比较错误中断清除位
17	R/W	0	hisEris4_G_int_clr DCIC 直方图 G 区间 4 比较错误中断清除位
16	R/W	0	hisEris3_G_int_clr DCIC 直方图 G 区间 3 比较错误中断清除位
15	R/W	0	hisEris2_G_int_clr DCIC 直方图 G 区间 2 比较错误中断清除位
14	R/W	0	hisEris1_G_int_clr DCIC 直方图 G 区间 1 比较错误中断清除位
13	R/W	0	hisEris0_G_int_clr DCIC 直方图 G 区间 0 比较错误中断清除位
12	R/W	0	hisEris4_R_int_clr DCIC 直方图 R 区间 4 比较错误中断清除位
11	R/W	0	hisEris3_R_int_clr DCIC 直方图 R 区间 3 比较错误中断清除位
10	R/W	0	hisEris2_R_int_clr DCIC 直方图 R 区间 2 比较错误中断清除位
9	R/W	0	hisEris1_R_int_clr

			DCIC 直方图 R 区间 1 比较错误中断清除位
8	R/W	0	hisEris0_R_int_clr DCIC 直方图 R 区间 0 比较错误中断清除位
7	R/W	0	defreeze_int_clr DCIC 冻结恢复中断清除位
6	R/W	0	freeze_int_clr DCIC 冻结中断清除位
5	R/W	0	hv_status_int_clr DCIC 行/场信号有无检测中断清除位
4	R/W	0	tol_line_num_pf_chag_int_clr DCIC 总行数变化中断清除位
3	R/W	0	act_line_num_pf_chag_int_clr DCIC 有效行数变化中断清除位
2	R/W	0	tol_pix_num_pl_chag_int_clr DCIC 总点数变化中断清除位
1	R/W	0	act_pix_num_pl_chag_int_clr DCIC 有效点数变化中断清除位
0	R	0	reserved

DCIC_INT_CLR1

Interrupt clear register, write 1,then write 0 to clear the interrupt.

Offset_Address: 0x5550_01E4



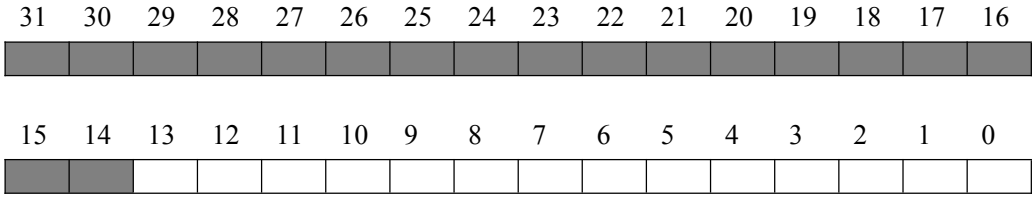
Bit	Type	Reset	Description
31	R/W	0	crc_b_error_clr7 DCIC 窗口 7 Blue CRC 错误中断清除位
30	R/W	0	crc_g_error_clr7 DCIC 窗口 7 Green CRC 错误中断清除位
29	R/W	0	crc_r_error_clr7 DCIC 窗口 7 Red CRC 错误中断清除位
28	R/W	0	crc_finish_int_clr7 DCIC 窗口 7 CRC 计算完成中断清除位
27	R/W	0	crc_b_error_clr6 DCIC 窗口 6 Blue CRC 错误中断清除位
26	R/W	0	crc_g_error_clr6 DCIC 窗口 6 Green CRC 错误中断清除位
25	R/W	0	crc_r_error_clr6 DCIC 窗口 6 Red CRC 错误中断清除位
24	R/W	0	crc_finish_int_clr6 DCIC 窗口 6 CRC 计算完成中断清除位
23	R/W	0	crc_b_error_clr5 DCIC 窗口 5 Blue CRC 错误中断清除位
22	R/W	0	crc_g_error_clr5 DCIC 窗口 5 Green CRC 错误中断清除位
21	R/W	0	crc_r_error_clr5 DCIC 窗口 5 Red CRC 错误中断清除位
20	R/W	0	crc_finish_int_clr5 DCIC 窗口 5 CRC 计算完成中断清除位
19	R/W	0	crc_b_error_clr4 DCIC 窗口 4 Blue CRC 错误中断清除位

18	R/W	0	crc_g_error_clr4 DCIC 窗口 4 Green CRC 错误中断清除位
17	R/W	0	crc_r_error_clr4 DCIC 窗口 4 Red CRC 错误中断清除位
16	R/W	0	crc_finish_int_clr4 DCIC 窗口 4 CRC 计算完成中断清除位
15	R/W	0	crc_b_error_clr3 DCIC 窗口 3 Blue CRC 错误中断清除位
14	R/W	0	crc_g_error_clr3 DCIC 窗口 3 Green CRC 错误中断清除位
13	R/W	0	crc_r_error_clr3 DCIC 窗口 3 Red CRC 错误中断清除位
12	R/W	0	crc_finish_int_clr3 DCIC 窗口 3 CRC 计算完成中断清除位
11	R/W	0	crc_b_error_clr2 DCIC 窗口 2 Blue CRC 错误中断清除位
10	R/W	0	crc_g_error_clr2 DCIC 窗口 2 Green CRC 错误中断清除位
9	R/W	0	crc_r_error_clr2 DCIC 窗口 2 Red CRC 错误中断清除位
8	R/W	0	crc_finish_int_clr2 DCIC 窗口 2 CRC 计算完成中断清除位
7	R/W	0	crc_b_error_clr1 DCIC 窗口 1 Blue CRC 错误中断清除位
6	R/W	0	crc_g_error_clr1 DCIC 窗口 1 Green CRC 错误中断清除位
5	R/W	0	crc_r_error_clr1 DCIC 窗口 1 Red CRC 错误中断清除位
4	R/W	0	crc_finish_int_clr1 DCIC 窗口 1 CRC 计算完成中断清除位
3	R/W	0	crc_b_error_clr0 DCIC 窗口 0 Blue CRC 错误中断清除位
2	R/W	0	crc_g_error_clr0 DCIC 窗口 0 Green CRC 错误中断清除位
1	R/W	0	crc_r_error_clr0 DCIC 窗口 0 Red CRC 错误中断清除位
0	R/W	0	crc_finish_int_clr0 DCIC 窗口 0 CRC 计算完成中断清除位

DCIC_STATUS0

DCIC status register

Offset_Address: 0x5550_0200



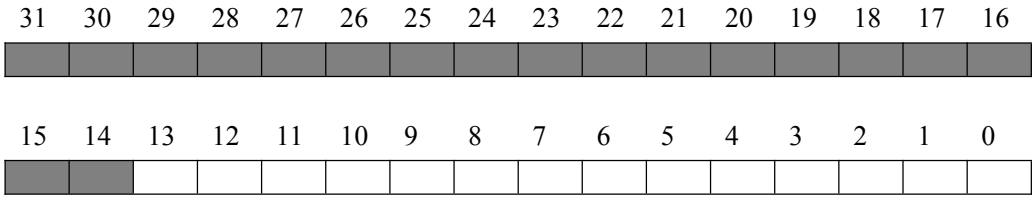
Bit	Type	Reset	Description
31	R	0	Reserved
30	R	0	hv_regen_vsyn_error DCIC 行/场异常中断状态位
29	R	0	hv_regen_vsyn_out_int DCIC 场中断状态位
28	R	0	Reserved
27	R	0	hisEris4_comp_err_Y DCIC 直方图 Y 区间 4 比较错误中断状态位
26	R	0	hisEris3_comp_err_Y DCIC 直方图 Y 区间 3 比较错误中断状态位

25	R	0	hisEris2_comp_err_Y DCIC 直方图 Y 区间 2 比较错误中断状态位
24	R	0	hisEris1_comp_err_Y DCIC 直方图 Y 区间 1 比较错误中断状态位
23	R	0	hisEris0_comp_err_Y DCIC 直方图 Y 区间 0 比较错误中断状态位
22	R	0	hisEris4_comp_err_B DCIC 直方图 B 区间 4 比较错误中断状态位
21	R	0	hisEris3_comp_err_B DCIC 直方图 B 区间 3 比较错误中断状态位
20	R	0	hisEris2_comp_err_B DCIC 直方图 B 区间 2 比较错误中断状态位
19	R	0	hisEris1_comp_err_B DCIC 直方图 B 区间 1 比较错误中断状态位
18	R	0	hisEris0_comp_err_B DCIC 直方图 B 区间 0 比较错误中断状态位
17	R	0	hisEris4_comp_err_G DCIC 直方图 G 区间 4 比较错误中断状态位
16	R	0	hisEris3_comp_err_G DCIC 直方图 G 区间 3 比较错误中断状态位
15	R	0	hisEris2_comp_err_G DCIC 直方图 G 区间 2 比较错误中断状态位
14	R	0	hisEris1_comp_err_G DCIC 直方图 G 区间 1 比较错误中断状态位
13	R	0	hisEris0_comp_err_G DCIC 直方图 G 区间 0 比较错误中断状态位
12	R	0	hisEris4_comp_err_R DCIC 直方图 R 区间 4 比较错误中断状态位
11	R	0	hisEris3_comp_err_R DCIC 直方图 R 区间 3 比较错误中断状态位
10	R	0	hisEris2_comp_err_R DCIC 直方图 R 区间 2 比较错误中断状态位
9	R	0	hisEris1_comp_err_R DCIC 直方图 R 区间 1 比较错误中断状态位
8	R	0	hisEris0_comp_err_R DCIC 直方图 R 区间 0 比较错误中断状态位
7	R	0	defreeze DCIC 冻结恢复中断状态位
6	R	0	freeze DCIC 冻结中断状态位
5	R	0	hv_status DCIC 行/场信号有无检测中断状态位
4	R	0	tol_line_num_pf_chag DCIC 总行数变化中断状态位
3	R	0	act_line_num_pf_chag DCIC 有效行数变化中断状态位
2	R	0	tol_pix_num_pl_chag DCIC 总点数变化中断状态位
1	R	0	act_pix_num_pl_chag DCIC 有效点数变化中断状态位
0	R	0	reserved

DCIC_STATUS1

DCIC status register

Offset_Address: 0x5550_0204



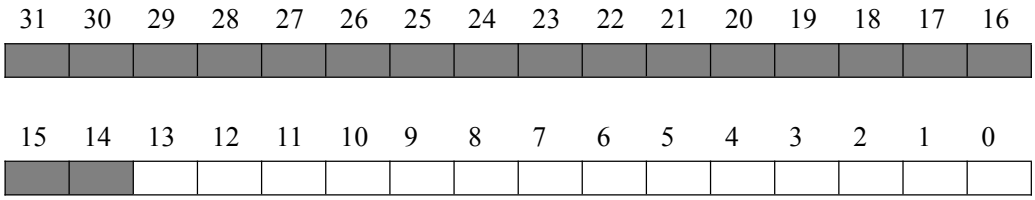
Bit	Type	Reset	Description
31	R	0	crc_b_error7 DCIC 窗口 7 Blue CRC 错误中断状态位
30	R	0	crc_g_error7 DCIC 窗口 7 Green CRC 错误中断状态位
29	R	0	crc_r_error7 DCIC 窗口 7 Red CRC 错误中断状态位
28	R	0	crc_finish7 DCIC 窗口 7 CRC 计算完成中断状态位
27	R	0	crc_b_error6 DCIC 窗口 6 Blue CRC 错误中断状态位
26	R	0	crc_g_error6 DCIC 窗口 6 Green CRC 错误中断状态位
25	R	0	crc_r_error6 DCIC 窗口 6 Red CRC 错误中断状态位
24	R	0	crc_finish6 DCIC 窗口 6 CRC 计算完成中断状态位
23	R	0	crc_b_error5 DCIC 窗口 5 Blue CRC 错误中断状态位
22	R	0	crc_g_error5 DCIC 窗口 5 Green CRC 错误中断状态位
21	R	0	crc_r_error5 DCIC 窗口 5 Red CRC 错误中断状态位
20	R	0	crc_finish5 DCIC 窗口 5 CRC 计算完成中断状态位
19	R	0	crc_b_error4 DCIC 窗口 4 Blue CRC 错误中断状态位
18	R	0	crc_g_error4 DCIC 窗口 4 Green CRC 错误中断状态位
17	R	0	crc_r_error4 DCIC 窗口 4 Red CRC 错误中断状态位
16	R	0	crc_finish4 DCIC 窗口 4 CRC 计算完成中断状态位
15	R	0	crc_b_error3 DCIC 窗口 3 Blue CRC 错误中断状态位
14	R	0	crc_g_error3 DCIC 窗口 3 Green CRC 错误中断状态位
13	R	0	crc_r_error3 DCIC 窗口 3 Red CRC 错误中断状态位
12	R	0	crc_finish3 DCIC 窗口 3 CRC 计算完成中断状态位
11	R	0	crc_b_error2 DCIC 窗口 2 Blue CRC 错误中断状态位
10	R	0	crc_g_error2 DCIC 窗口 2 Green CRC 错误中断状态位
9	R	0	crc_r_error2 DCIC 窗口 2 Red CRC 错误中断状态位
8	R	0	crc_finish2 DCIC 窗口 2 CRC 计算完成中断状态位
7	R	0	crc_b_error1 DCIC 窗口 1 Blue CRC 错误中断状态位
6	R	0	crc_g_error1 DCIC 窗口 1 Green CRC 错误中断状态位
5	R	0	crc_r_error1 DCIC 窗口 1 Red CRC 错误中断状态位

4	R	0	crc_finish1 DCIC 窗口 1 CRC 计算完成中断状态位
3	R	0	crc_b_error0 DCIC 窗口 0 Blue CRC 错误中断状态位
2	R	0	crc_g_error0 DCIC 窗口 0 Green CRC 错误中断状态位
1	R	0	crc_r_error0 DCIC 窗口 0 Red CRC 错误中断状态位
0	R	0	crc_finish0 DCIC 窗口 0 CRC 计算完成中断状态位

DCIC_STATUS2

DCIC status register

Offset_Address: 0x5550_0208

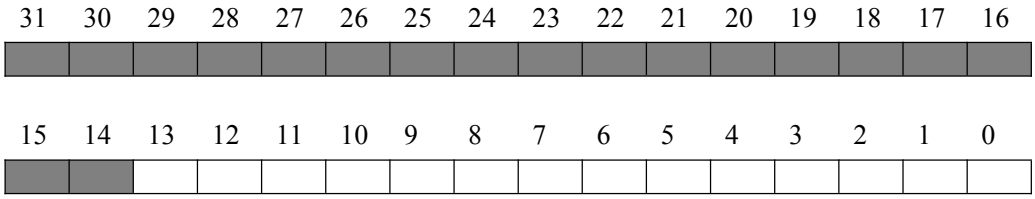


Bit	Type	Reset	Description
31-28	R	0	Reserved
27-16	R	0	tol_pix_num_pl_out DCIC 当前帧的总点数
15-13	R	0	Reserved
12-0	R	0	act_pix_num_pl_out DCIC 当前帧的有效点数

DCIC_STATUS3

DCIC status register

Offset_Address: 0x5550_020C

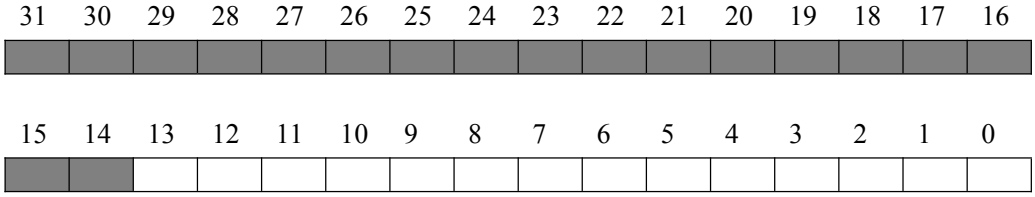


Bit	Type	Reset	Description
31-28	R	0	Reserved
27-16	R	0	tol_line_num_pf_out DCIC 当前帧的总行数
15-12	R	0	Reserved
11-0	R	0	act_line_num_pf_out DCIC 当前帧的有效行数

DCIC_STATUS4

DCIC status register

Offset_Address: 0x5550_0210



Bit	Type	Reset	Description
31-2	R	0	Reserved
1-0	R	0	freeze_state DCIC 冻结检测状态机当前状态 00: IDLE; 01: FREEZE; 10: DEFREEZE;

29.3 OSD Normal Register(Address:0x55500400 + addr*4)

地址	复位值	读写类型	Bit 位	名称	说明
0x00	00h	RW	[31]	OSDC31_EN	osd block 31 使能。0 无效，1 有效
			[30]	OSDC30_EN	osd block 30 使能。0 无效，1 有效
			[29]	OSDC29_EN	osd block 29 使能。0 无效，1 有效
			[28]	OSDC28_EN	osd block 28 使能。0 无效，1 有效
			[27]	OSDC27_EN	osd block 27 使能。0 无效，1 有效
			[26]	OSDC26_EN	osd block 26 使能。0 无效，1 有效
			[25]	OSDC25_EN	osd block 25 使能。0 无效，1 有效
			[24]	OSDC24_EN	osd block 24 使能。0 无效，1 有效
			[23]	OSDC23_EN	osd block 23 使能。0 无效，1 有效
			[22]	OSDC22_EN	osd block 22 使能。0 无效，1 有效
			[21]	OSDC21_EN	osd block 21 使能。0 无效，1 有效
			[20]	OSDC20_EN	osd block 20 使能。0 无效，1 有效
			[19]	OSDC19_EN	osd block 19 使能。0 无效，1 有效

			[18]	OSDC18_EN	osd block 18 使能。0 无效，1 有效
			[17]	OSDC17_EN	osd block 17 使能。0 无效，1 有效
			[16]	OSDC16_EN	osd block 16 使能。0 无效，1 有效
			[15]	OSDC15_EN	osd block 15 使能。0 无效，1 有效
			[14]	OSDC14_EN	osd block 14 使能。0 无效，1 有效
			[13]	OSDC13_EN	osd block 13 使能。0 无效，1 有效
			[12]	OSDC12EN	osd block 12 使能。0 无效，1 有效
			[11]	OSDC11_EN	osd block 11 使能。0 无效，1 有效
			[10]	OSDC10_EN	osd block 10 使能。0 无效，1 有效
			[9]	OSDC9_EN	osd block 9 使能。0 无效，1 有效
			[8]	OSDC8_EN	osd block 8 使能。0 无效，1 有效
			[7]	OSDC7_EN	osd block 7 使能。0 无效，1 有效
			[6]	OSDC6_EN	osd block 6 使能。0 无效，1 有效
			[5]	OSDC5_EN	osd block 5 使能。0 无效，1 有效
			[4]	OSDC4_EN	osd block 4 使能。0 无效，1 有效
			[3]	OSDC3_EN	osd block 3 使能。0 无效，1 有效
			[2]	OSDC2_EN	osd block 2 使能。0 无效，1 有效
			[1]	OSDC1_EN	osd block 1 使能。0 无效，1 有效
			[0]	OSDC0_EN	osd block 0 使能。0 无效，1 有效
0x01	00h	RW	[31]	OSDC63_EN	osd block 63 使能。0 无效，1 有效
			[30]	OSDC62_EN	osd block 62 使能。0 无效，1 有效
			[29]	OSDC61_EN	osd block 61 使能。0 无效，1 有效
			[28]	OSDC60_EN	osd block 60 使能。0 无效，1 有效

			[27]	OSDC59_EN	osd block 59 使能。0 无效，1 有效
			[26]	OSDC58_EN	osd block 58 使能。0 无效，1 有效
			[25]	OSDC57_EN	osd block 57 使能。0 无效，1 有效
			[24]	OSDC56_EN	osd block 56 使能。0 无效，1 有效
			[23]	OSDC55_EN	osd block 55 使能。0 无效，1 有效
			[22]	OSDC54_EN	osd block 54 使能。0 无效，1 有效
			[21]	OSDC53_EN	osd block 53 使能。0 无效，1 有效
			[20]	OSDC52_EN	osd block 52 使能。0 无效，1 有效
			[19]	OSDC51_EN	osd block 51 使能。0 无效，1 有效
			[18]	OSDC50_EN	osd block 50 使能。0 无效，1 有效
			[17]	OSDC49_EN	osd block 49 使能。0 无效，1 有效
			[16]	OSDC48_EN	osd block 48 使能。0 无效，1 有效
			[15]	OSDC47_EN	osd block 47 使能。0 无效，1 有效
			[14]	OSDC46_EN	osd block 46 使能。0 无效，1 有效
			[13]	OSDC45_EN	osd block 45 使能。0 无效，1 有效
			[12]	OSDC44_EN	osd block 44 使能。0 无效，1 有效
			[11]	OSDC43_EN	osd block 43 使能。0 无效，1 有效
			[10]	OSDC42_EN	osd block 42 使能。0 无效，1 有效
			[9]	OSDC41_EN	osd block 41 使能。0 无效，1 有效
			[8]	OSDC40_EN	osd block 40 使能。0 无效，1 有效
			[7]	OSDC39_EN	osd block 39 使能。0 无效，1 有效
			[6]	OSDC38_EN	osd block 38 使能。0 无效，1 有效
			[5]	OSDC37_EN	osd block 37 使能。0 无效，1 有效

			[4]	OSDC36_EN	osd block 36 使能。0 无效，1 有效
			[3]	OSDC35_EN	osd block 35 使能。0 无效，1 有效
			[2]	OSDC34_EN	osd block 34 使能。0 无效，1 有效
			[1]	OSDC33_EN	osd block 33 使能。0 无效，1 有效
			[0]	OSDC32_EN	osd block 32 使能。0 无效，1 有效
0x02	00h	RW	[31]	OSD_BL_31_EN	osd block 31 Blink 使能。0 无效，1 有效
			[30]	OSD_BL_30_EN	osd block 30 Blink 使能。0 无效，1 有效
			[29]	OSD_BL_29_EN	osd block 29 Blink 使能。0 无效，1 有效
			[28]	OSD_BL_28_EN	osd block 28 Blink 使能。0 无效，1 有效
			[27]	OSD_BL_27_EN	osd block 27 Blink 使能。0 无效，1 有效
			[26]	OSD_BL_26_EN	osd block 26 Blink 使能。0 无效，1 有效
			[25]	OSD_BL_25_EN	osd block 25 Blink 使能。0 无效，1 有效
			[24]	OSD_BL_24_EN	osd block 24 Blink 使能。0 无效，1 有效
			[23]	OSD_BL_23_EN	osd block 23 Blink 使能。0 无效，1 有效
			[22]	OSD_BL_22_EN	osd block 22 Blink 使能。0 无效，1 有效
			[21]	OSD_BL_21_EN	osd block 21 Blink 使能。0 无效，1 有效
			[20]	OSD_BL_20_EN	osd block 20 Blink 使能。0 无效，1 有效
			[19]	OSD_BL_19_EN	osd block 19 Blink 使能。0 无效，1 有效
			[18]	OSD_BL_18_EN	osd block 18 Blink 使能。0 无效，1 有效
			[17]	OSD_BL_17_EN	osd block 17 Blink 使能。0 无效，1 有效
			[16]	OSD_BL_16_EN	osd block 16 Blink 使能。0 无效，1 有效
			[15]	OSD_BL_15_EN	osd block 15 Blink 使能。0 无效，1 有效
			[14]	OSD_BL_14_EN	osd block 14 Blink 使能。0 无效，1 有效

			[13]	OSD_BL_13_EN	osd block 13 Blink 使能。0 无效，1 有效
			[12]	OSD_BL_12EN	osd block 12 Blink 使能。0 无效，1 有效
			[11]	OSD_BL_11_EN	osd block 11 Blink 使能。0 无效，1 有效
			[10]	OSD_BL_10_EN	osd block 10 Blink 使能。0 无效，1 有效
			[9]	OSD_BL_9_EN	osd block 9 Blink 使能。0 无效，1 有效
			[8]	OSD_BL_8_EN	osd block 8 Blink 使能。0 无效，1 有效
			[7]	OSD_BL_7_EN	osd block 7 Blink 使能。0 无效，1 有效
			[6]	OSD_BL_6_EN	osd block 6 Blink 使能。0 无效，1 有效
			[5]	OSD_BL_5_EN	osd block 5 Blink 使能。0 无效，1 有效
			[4]	OSD_BL_4_EN	osd block 4 Blink 使能。0 无效，1 有效
			[3]	OSD_BL_3_EN	osd block 3 Blink 使能。0 无效，1 有效
			[2]	OSD_BL_2_EN	osd block 2 Blink 使能。0 无效，1 有效
			[1]	OSD_BL_1_EN	osd block 1 Blink 使能。0 无效，1 有效
			[0]	OSD_BL_0_EN	osd block 0 Blink 使能。0 无效，1 有效
0x03	00h	RW	[31]	OSD_BL_63_EN	osd block 63 Blink 使能。0 无效，1 有效
			[30]	OSD_BL_62_EN	osd block 62 Blink 使能。0 无效，1 有效
			[29]	OSD_BL_61_EN	osd block 61 Blink 使能。0 无效，1 有效
			[28]	OSD_BL_60_EN	osd block 60 Blink 使能。0 无效，1 有效
			[27]	OSD_BL_59_EN	osd block 59 Blink 使能。0 无效，1 有效
			[26]	OSD_BL_58_EN	osd block 58 Blink 使能。0 无效，1 有效
			[25]	OSD_BL_57_EN	osd block 57 Blink 使能。0 无效，1 有效
			[24]	OSD_BL_56_EN	osd block 56 Blink 使能。0 无效，1 有效
			[23]	OSD_BL_55_EN	osd block 55 Blink 使能。0 无效，1 有效

			[22]	OSD_BL_54_EN	osd block 54 Blink 使能。 0 无效，1 有效
			[21]	OSD_BL_53_EN	osd block 53 Blink 使能。 0 无效，1 有效
			[20]	OSD_BL_52_EN	osd block 52 Blink 使能。 0 无效，1 有效
			[19]	OSD_BL_51_EN	osd block 51 Blink 使能。 0 无效，1 有效
			[18]	OSD_BL_50_EN	osd block 50 Blink 使能。 0 无效，1 有效
			[17]	OSD_BL_49_EN	osd block 49 Blink 使能。 0 无效，1 有效
			[16]	OSD_BL_48_EN	osd block 48 Blink 使能。 0 无效，1 有效
			[15]	OSD_BL_47_EN	osd block 47 Blink 使能。 0 无效，1 有效
			[14]	OSD_BL_46_EN	osd block 46 Blink 使能。 0 无效，1 有效
			[13]	OSD_BL_45_EN	osd block 45 Blink 使能。 0 无效，1 有效
			[12]	OSD_BL_44_EN	osd block 44 Blink 使能。 0 无效，1 有效
			[11]	OSD_BL_43_EN	osd block 43 Blink 使能。 0 无效，1 有效
			[10]	OSD_BL_42_EN	osd block 42 Blink 使能。 0 无效，1 有效
			[9]	OSD_BL_41_EN	osd block 41 Blink 使能。 0 无效，1 有效
			[8]	OSD_BL_40_EN	osd block 40 Blink 使能。 0 无效，1 有效
			[7]	OSD_BL_39_EN	osd block 39 Blink 使能。 0 无效，1 有效
			[6]	OSD_BL_38_EN	osd block 38 Blink 使能。 0 无效，1 有效
			[5]	OSD_BL_37_EN	osd block 37 Blink 使能。 0 无效，1 有效
			[4]	OSD_BL_36_EN	osd block 36 Blink 使能。 0 无效，1 有效
			[3]	OSD_BL_35_EN	osd block 35 Blink 使能。 0 无效，1 有效
			[2]	OSD_BL_34_EN	osd block 34 Blink 使能。 0 无效，1 有效
			[1]	OSD_BL_33_EN	osd block 33 Blink 使能。 0 无效，1 有效
			[0]	OSD_BL_32_EN	osd block 32 Blink 使能。 0 无效，1 有效

0x04	00h	RW	[7]	osd_comp_err_clr	0: not clear compare error statu 1: clear compare error statu
		RW	[6]	osd_replace_mode	0: OSD 全覆盖 1：OSD 不覆盖底色
		RW	[5]	osd_comp_sel	0: 选择一个调色盘颜色比较 1：选择除透明色之外的颜色比较
		RW	[4]	vsyn_src_sel	输入场同步源选择，用来定位 osd 位置 1：选择场有效信号 vblank； 0：选择场同步信号 vsyn；
		RW	[3]	hsyn_src_sel	输入行同步源选择，用来定位 osd 位置 1：选择行有效信号 daten； 0：选择行同步信号 hsyn；
		RW	[2]	Vsyn_inv_sel	
		RW	[1]	Hsyn_inv_sel	
		RW	[0]	OSD_ODD_EVEN_MODE	OSD 输出奇偶模式选择，设置为 1 时表示双通道奇偶输出
0x05	00h	RW			
		RW	[15:8]	OSD_BRIGHT1	OSD 底色亮度调整 R = （R * OSD_BRIGHT1）/ 128 G = （G * OSD_BRIGHT1）/ 128 B = （B * OSD_BRIGHT1）/ 128
		RW	[7:0]	OSD_BRIGHT	OSD 亮度调整 R = （R * OSD_BRIGHT）/ 128 G = （G * OSD_BRIGHT）/ 128 B = （B * OSD_BRIGHT）/ 128
0x06	00h				
		RW	[31:0]	compare_en	每一个 bit 对应一个窗口 0~31 0: disable; 1: enable;
0x07	00h	RW			
		RW	[31:0]	compare_en1	每一个 bit 对应一个窗口 32~63 0: disable; 1: enable;

0x08	00h	RW	[0]	osd_coef_syn	
0x09	00h	RW	[23:16]	imc_bth	Image compare B 阈值
		RW	[15:8]	imc_gth	Image compare G 阈值
		RW	[7:0]	imc_rth	Image compare R 阈值
0x0A ~0x0 B	00h	R	[31:0]	compare_statu	OSD 0~63 窗口数据比较状态位 0：数据比较正确 1：数据比较错误
0x10 ~0x1 F	00h	RW	[31:24]	compare03_value	窗口 3 主颜色比较值
			[23:16]	compare02_value	窗口 2 主颜色比较值
			[15:8]	compare01_value	窗口 1 主颜色比较值
			[7:0]	compare00_value	窗口 0 主颜色比较值
0x70	00h	RW	[31:6]	---	reserved
			[8]	test_pattern_en	测试图形生成使能控制，1 有效
			[7:6]	---	reserved

			[5:0]	Test_pattern_sel	测试图形输出选择 0：White； 1：Black； 2：Red； 3：Green； 4：Blue; 5：Cyan； 6：Magenta； 7：Yellow； 8：V Gray； 9：V Gray Red； 10：V Gray Green； 11：V Gray Blue； 12：Gray； 13：H Gray； 14：H Gray Red; 15：H Gray Green； 16：H Gray Blue； 17：V Gray 2； 18：V Gray 2 Red； 19：V Gray 2 Green; 20：V Gray 2 Blue； 21：H Gray 2； 22：H Gray 2 Red; 23：H Gray 2 Green； 24：H Gray 2 Blue； 25：Chess 26：Sub-pixel Check； 27 ： 2line Sub-pixel Check； 28 ： 2line+1 Sub-pixel Check； 29：Pixel Check； 30：V Color Bar； 31：V Color Bar 2； 32 ： V Gray V Color Bar； 33： V Gray V Color Bar 2； 34：H Color Bar 35：H Gray H Color Bar 36：Outer Frame； 37：Outer Frame 2； 38：Crosstalk 1； 39：Crosstalk 2； 40 ： H Linear Gradient Grey； 41：Orientation Test； 42：V Flicker Test；

					43：H Flicker Test； 44：Pure color
0x71	00h	RW	[31:20]	---	reserved
			[19:12]	vgray_step_cfg	垂直灰阶测试图形梯度
			[11:0]	vgray_lnum_cfg	垂直灰阶测试图形宽度
0x72	00h	RW	[31:20]	---	reserved
			[19:12]	hgray_step_cfg	水平灰阶测试图形梯度
			[11:0]	hgray_lnum_cfg	水平灰阶测试图形宽度
0x73	00h	RW	[31:29]	---	reserved
			[28:16]	hgray_rgb_lum	水平渐变测试图形渐变步长， 计算：256÷图像有效宽度×2048
			[15:13]	---	reserved
			[12:0]	vgray_rgb_lum	垂直渐变测试图形渐变步长， 计算：256÷图像有效高度×2048
0x74	00h	RW	[31:24]	---	reserved
			[23:12]	chess_lnum	棋格（chess）测试图形高度
			[11:0]	chess_pnum	棋格（chess）测试图形宽度
0x75	00h	RW	[31:24]	---	reserved
			[23:12]	subpix_lunm	颜色格子（sub pix）测试图形高度
			[11:0]	subpix_pnum	颜色格子（sub pix）测试图形宽度
0x76	00h	RW	[31:24]	---	reserved
			[23:12]	hcolbar_step	水平彩条测试图形宽度，一般配置为：图像有效高度÷8
			[11:0]	vcolbar_step	垂直彩条测试图形宽度，一般配置为：图像有效宽度÷8
0x77	00h	RW	[31:24]	---	reserved
			[23:12]	crostalk_xstop	crosstalk 测试图形水平结束位置
			[11:0]	crostalk_xstart	crosstalk 测试图形水平起始位置
0x78	00h	RW	[31:24]	---	reserved
			[23:12]	crostalk_ystop	crosstalk 测试图形垂直结束位置
			[11:0]	crostalk_ystart	crosstalk 测试图形垂直起始位置

0x79	00h	RW	[31:24]	---	reserved
			[23:12]	orient_ystop	orientation 测试图形垂直结束位置
			[11:0]	orient_xstart	orientation 测试图形水平起始位置
0x7A	00h	RW	[31:24]	---	reserved
			[23:12]	hflicker_step	H flicker 测试图形宽度
			[11:0]	vflicker_step	V flicker 测试图形宽度
0x7B	00h	RW	[31:29]	---	reserved
			[28:16]	hgray_hline_lum	Rgb 色彩水平渐变色的水平渐变梯度，计算：256÷宽度×2048
			[15:8]	outerf_lnum	Outer Frame 测试图形垂直高度
			[7:0]	outerf_pnum	Outer Frame 测试图形水平宽度
0x7C	00h	RW	[31:24]	---	reserved
			[23:16]	pure_b	纯色测试图形蓝色 B 分量
			[15:8]	pure_g	纯色测试图形绿色 G 分量
			[7:0]	pure_r	纯色测试图形红色 R 分量
0x7D	00h	RW	[31:28]	---	reserved
			[27:16]	highth	每场有效行数
			[15:12]	---	reserved
			[11:0]	width	每行有效宽度
OSD1 Register(Address:0x55500800 + addr*4)					
0x00	00h	RW	[31:25]	---	reserved
			[24:16]	Char0_size_y	Osd block0 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char0_size_x	Osd block0 字符大小，每个 icon 的水平像素点数
0x01	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc0_posy	osd block 0 起始点像素的垂直位置
			[11:0]	Osdc0_posx	osd block 0 起始点像素的水平位置
0x02	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc0_addr	确定 osd block 0 对应的 ram 起始地址

0x03	00h	RW	[7:6]	Osdc0_pal_src	Osd block 0 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc0_blink	osd block 0 闪烁区域闪烁频率调节
0x04	00h	RW	[31:25]	---	reserved
			[24:16]	Char1_size_y	Osd block1 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char1_size_x	Osd block1 字符大小，每个 icon 的水平像素点数
0x05	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc1_posy	osd block 1 起始点像素的垂直位置
			[11:0]	Osdc1_posx	osd block 1 起始点像素的水平位置
0x06	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc1_addr	确定 osd block 1 对应的 ram 起始地址
0x07	00h	RW	[7:6]	Osdc1_pal_src	Osd block 1 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc1_blink	osd block 1 闪烁区域闪烁频率调节
0x08	00h	RW	[31:25]	---	reserved
			[24:16]	Char2_size_y	Osd block2 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char2_size_x	Osd block2 字符大小，每个 icon 的水平像素点数
0x09	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc2_posy	osd block 2 起始点像素的垂直位置
			[11:0]	Osdc2_posx	osd block 2 起始点像素的水平位置
0x0A	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc2_addr	确定 osd block 2 对应的 ram 起始地址

0x0B	00h	RW	[7:6]	Osdc2_pal_src	Osd block 2 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc2_blink	osd block 2 闪烁区域闪烁频率调节
0x0C	00h	RW	[31:25]	---	reserved
			[24:16]	Char3_size_y	Osd block3 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char3_size_x	Osd block3 字符大小，每个 icon 的水平像素点数
0x0D	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc3_posy	osd block 3 起始点像素的垂直位置
			[11:0]	Osdc3_posx	osd block 3 起始点像素的水平位置
0x0E	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc3_addr	确定 osd block 3 对应的 ram 起始地址
0x0F	00h	RW	[7:6]	Osdc3_pal_src	Osd block 3 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc3_blink	osd block 3 闪烁区域闪烁频率调节
0x10	00h	RW	[31:25]	---	reserved
			[24:16]	Char4_size_y	Osd block4 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char4_size_x	Osd block4 字符大小，每个 icon 的水平像素点数
0x11	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc4_posy	osd block 4 起始点像素的垂直位置
			[11:0]	Osdc4_posx	osd block 4 起始点像素的水平位置
0x12	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc4_addr	确定 osd block 4 对应的 ram 起始地址

0x13	00h	RW	[7:6]	Osdc4_pal_src	Osd block 4 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc4_blink	osd block 4 闪烁区域闪烁频率调节
0x14	00h	RW	[31:25]	---	reserved
			[24:16]	Char5_size_y	Osd block5 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char5_size_x	Osd block5 字符大小，每个 icon 的水平像素点数
0x15	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc5_posy	osd block 5 起始点像素的垂直位置
			[11:0]	Osdc5_posx	osd block 5 起始点像素的水平位置
0x16	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc5_addr	确定 osd block 5 对应的 ram 起始地址
0x17	00h	RW	[7:6]	Osdc5_pal_src	Osd block 5 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc5_blink	osd block 5 闪烁区域闪烁频率调节
0x18	00h	RW	[31:25]	---	reserved
			[24:16]	Char6_size_y	Osd block6 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char6_size_x	Osd block6 字符大小，每个 icon 的水平像素点数
0x19	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc6_posy	osd block 6 起始点像素的垂直位置
			[11:0]	Osdc6_posx	osd block 6 起始点像素的水平位置
0x1A	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc6_addr	确定 osd block 6 对应的 ram 起始地址

0x1B	00h	RW	[7:6]	Osdc6_pal_src	Osd block 6 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc6_blink	osd block 6 闪烁区域闪烁频率调节
0x1C	00h	RW	[31:25]	---	reserved
			[24:16]	Char7_size_y	Osd block7 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char7_size_x	Osd block7 字符大小，每个 icon 的水平像素点数
0x1D	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc7_posy	osd block 7 起始点像素的垂直位置
			[11:0]	Osdc7_posx	osd block 7 起始点像素的水平位置
0x1E	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc7_addr	确定 osd block 7 对应的 ram 起始地址
0x1F	00h	RW	[7:6]	Osdc7_pal_src	Osd block 7 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc7_blink	osd block 7 闪烁区域闪烁频率调节
0x20	00h	RW	[31:25]	---	reserved
			[24:16]	Char8_size_y	Osd block8 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char8_size_x	Osd block8 字符大小，每个 icon 的水平像素点数
0x21	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc8_posy	osd block 8 起始点像素的垂直位置
			[11:0]	Osdc8_posx	osd block 8 起始点像素的水平位置
0x22	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc8_addr	确定 osd block 8 对应的 ram 起始地址

0x23	00h	RW	[7:6]	Osdc8_pal_src	Osd block 8 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc8_blink	osd block 8 闪烁区域闪烁频率调节
0x24	00h	RW	[31:25]	---	reserved
			[24:16]	Char9_size_y	Osd block9 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char9_size_x	Osd block9 字符大小，每个 icon 的水平像素点数
0x25	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc9_posy	osd block 9 起始点像素的垂直位置
			[11:0]	Osdc9_posx	osd block 9 起始点像素的水平位置
0x26	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc9_addr	确定 osd block 9 对应的 ram 起始地址
0x27	00h	RW	[7:6]	Osdc9_pal_src	Osd block 9 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc9_blink	osd block 9 闪烁区域闪烁频率调节
0x28	00h	RW	[31:25]	---	reserved
			[24:16]	Char10_size_y	Osd block10 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char10_size_x	Osd block10 字符大小，每个 icon 的水平像素点数
0x29	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc10_posy	osd block 10 起始点像素的垂直位置
			[11:0]	Osdc10_posx	osd block 10 起始点像素的水平位置
0x2A	00h	RW	[31:17]	---	reserved

			[16:0]	Osdc10_addr	确定 osd block 10 对应的 ram 起始地址
0x2B	00h	RW	[7:6]	Osdc10_pal_src	Osd block 10 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc10_blink	osd block 10 闪烁区域闪烁频率调节
0x2C	00h	RW	[31:25]	---	reserved
			[24:16]	Char11_size_y	Osd block11 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char11_size_x	Osd block11 字符大小，每个 icon 的水平像素点数
0x2D	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc11_posy	osd block 11 起始点像素的垂直位置
			[11:0]	Osdc11_posx	osd block 11 起始点像素的水平位置
0x2E	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc11_addr	确定 osd block 11 对应的 ram 起始地址
0x2F	00h	RW	[7:6]	Osdc11_pal_src	Osd block 11 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc11_blink	osd block 11 闪烁区域闪烁频率调节
0x30	00h	RW	[31:25]	---	reserved
			[24:16]	Char12_size_y	Osd block12 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char12_size_x	Osd block12 字符大小，每个 icon 的水平像素点数
0x31	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc12_posy	osd block 12 起始点像素的垂直位置

			[11:0]	Osdc12_posx	osd block 12 起始点像素的水平位置
0x32	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc12_addr	确定 osd block 12 对应的 ram 起始地址
0x33	00h	RW	[7:6]	Osd12_pal_src	Osd block 12 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc12_blink	osd block 12 闪烁区域闪烁频率调节
0x34	00h	RW	[31:25]	---	reserved
			[24:16]	Char13_size_y	Osd block13 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char13_size_x	Osd block13 字符大小，每个 icon 的水平像素点数
0x35	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc13_posy	osd block 13 起始点像素的垂直位置
			[11:0]	Osdc13_posx	osd block 13 起始点像素的水平位置
0x36	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc13_addr	确定 osd block 13 对应的 ram 起始地址
0x37	00h	RW	[7:6]	Osdc13_pal_src	Osd block 13 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc13_blink	osd block 13 闪烁区域闪烁频率调节
0x38	00h	RW	[31:25]	---	reserved
			[24:16]	Char14_size_y	Osd block14 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char14_size_x	Osd block14 字符大小，每个 icon 的水平像素点数
0x39	00h	RW	[31:24]	---	reserved

			[23:12]	Osdc14_posy	osd block 14 起始点像素的垂直位置
			[11:0]	Osdc14_posx	osd block 14 起始点像素的水平位置
0x3A	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc14_addr	确定 osd block 14 对应的 ram 起始地址
0x3B	00h	RW	[7:6]	Osdc14_pal_src	Osd block 14 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc14_blink	osd block 14 闪烁区域闪烁频率调节
0x3C	00h	RW	[31:25]	---	reserved
			[24:16]	Char15_size_y	Osd block15 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char15_size_x	Osd block15 字符大小，每个 icon 的水平像素点数
0x3D	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc15_posy	osd block 15 起始点像素的垂直位置
			[11:0]	Osdc15_posx	osd block 15 起始点像素的水平位置
0x3E	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc15_addr	确定 osd block 15 对应的 ram 起始地址
0x3F	00h	RW	[7:6]	Osdc15_pal_src	Osd block 15 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc15_blink	osd block 15 闪烁区域闪烁频率调节
0x40	00h	RW	[31:25]	---	reserved
			[24:16]	Char16_size_y	Osd block16 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved

			[8:0]	Char16_size_x	Osd block16 字符大小，每个 icon 的水平像素点数
0x41	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc16_posy	osd block 16 起始点像素的垂直位置
			[11:0]	Osdc16_posx	osd block 16 起始点像素的水平位置
0x42	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc16_addr	确定 osd block 16 对应的 ram 起始地址
0x43	00h	RW	[7:6]	Osdc16_pal_src	Osd block 16 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc16_blink	osd block 16 闪烁区域闪烁频率调节
0x44	00h	RW	[31:25]	---	reserved
			[24:16]	Char17_size_y	Osd block17 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char17_size_x	Osd block17 字符大小，每个 icon 的水平像素点数
0x45	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc17_posy	osd block 17 起始点像素的垂直位置
			[11:0]	Osdc17_posx	osd block 17 起始点像素的水平位置
0x46	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc17_addr	确定 osd block 17 对应的 ram 起始地址
0x47	00h	RW	[7:6]	Osdc17_pal_src	Osd block 17 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc17_blink	osd block 17 闪烁区域闪烁频率调节
0x48	00h	RW	[31:25]	---	reserved

			[24:16]	Char18_size_y	Osd block18 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char18_size_x	Osd block18 字符大小，每个 icon 的水平像素点数
0x49	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc18_posy	osd block 18 起始点像素的垂直位置
			[11:0]	Osdc18_posx	osd block 18 起始点像素的水平位置
0x4A	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc18_addr	确定 osd block 18 对应的 ram 起始地址
0x4B	00h	RW	[7:6]	Osdc18_pal_src	Osd block 18 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc18_blink	osd block 18 闪烁区域闪烁频率调节
0x4C	00h	RW	[31:25]	---	reserved
			[24:16]	Char19_size_y	Osd block19 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char19_size_x	Osd block19 字符大小，每个 icon 的水平像素点数
0x4D	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc19_posy	osd block 19 起始点像素的垂直位置
			[11:0]	Osdc19_posx	osd block 19 起始点像素的水平位置
0x4E	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc19_addr	确定 osd block 19 对应的 ram 起始地址
0x4F	00h	RW	[7:6]	Osdc19_pal_src	Osd block 19 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像

		RW	[5:0]	Osdc19_blink	osd block 19 闪烁区域闪烁频率调节
0x50	00h	RW	[31:25]	---	reserved
			[24:16]	Char20_size_y	Osd block20 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char20_size_x	Osd block20 字符大小，每个 icon 的水平像素点数
0x51	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc20_posy	osd block 20 起始点像素的垂直位置
			[11:0]	Osdc20_posx	osd block 20 起始点像素的水平位置
0x52	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc20_addr	确定 osd block 20 对应的 ram 起始地址
0x53	00h	RW	[7:6]	Osdc20_pal_src	Osd block 20 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc20_blink	osd block 20 闪烁区域闪烁频率调节
0x54	00h	RW	[31:25]	---	reserved
			[24:16]	Char21_size_y	Osd block21 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char21_size_x	Osd block21 字符大小，每个 icon 的水平像素点数
0x55	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc21_posy	osd block 21 起始点像素的垂直位置
			[11:0]	Osdc21_posx	osd block 21 起始点像素的水平位置
0x56	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc21_addr	确定 osd block 21 对应的 ram 起始地址

0x57	00h	RW	[7:6]	Osdc21_pal_src	Osd block 21 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc21_blink	osd block 21 闪烁区域闪烁频率调节
0x58	00h	RW	[31:25]	---	reserved
			[24:16]	Char22_size_y	Osd block22 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char22_size_x	Osd block22 字符大小，每个 icon 的水平像素点数
0x59	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc22_posy	osd block 22 起始点像素的垂直位置
			[11:0]	Osdc22_posx	osd block 22 起始点像素的水平位置
0x5A	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc22_addr	确定 osd block 22 对应的 ram 起始地址
0x5B	00h	RW	[7:6]	Osdc22_pal_src	Osd block 22 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc22_blink	osd block 22 闪烁区域闪烁频率调节
0x5C	00h	RW	[31:25]	---	reserved
			[24:16]	Char23_size_y	Osd block23 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char23_size_x	Osd block23 字符大小，每个 icon 的水平像素点数
0x5D	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc23_posy	osd block 23 起始点像素的垂直位置
			[11:0]	Osdc23_posx	osd block 23 起始点像素的水平位置
0x5E	00h	RW	[31:17]	---	reserved

			[16:0]	Osdc23_addr	确定 osd block 23 对应的 ram 起始地址
0x5F	00h	RW	[7:6]	Osdc23_pal_src	Osd block 23 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc23_blink	osd block 23 闪烁区域闪烁频率调节
0x60	00h	RW	[31:25]	---	reserved
			[24:16]	Char24_size_y	Osd block24 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char24_size_x	Osd block24 字符大小，每个 icon 的水平像素点数
0x61	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc24_posy	osd block 24 起始点像素的垂直位置
			[11:0]	Osdc24_posx	osd block 24 起始点像素的水平位置
0x62	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc24_addr	确定 osd block 24 对应的 ram 起始地址
0x63	00h	RW	[7:6]	Osdc24_pal_src	Osd block 24 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc24_blink	osd block 24 闪烁区域闪烁频率调节
0x64	00h	RW	[31:25]	---	reserved
			[24:16]	Char25_size_y	Osd block25 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char25_size_x	Osd block25 字符大小，每个 icon 的水平像素点数
0x65	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc25_posy	osd block 25 起始点像素的垂直位置

			[11:0]	Osdc25_posx	osd block 25 起始点像素的水平位置
0x66	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc25_addr	确定 osd block 25 对应的 ram 起始地址
0x67	00h	RW	[7:6]	Osdc25_pal_src	Osd block 25 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc25_blink	osd block 25 闪烁区域闪烁频率调节
0x68	00h	RW	[31:25]	---	reserved
			[24:16]	Char26_size_y	Osd block26 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char26_size_x	Osd block26 字符大小，每个 icon 的水平像素点数
0x69	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc26_posy	osd block 26 起始点像素的垂直位置
			[11:0]	Osdc26_posx	osd block 26 起始点像素的水平位置
0x6A	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc26_addr	确定 osd block 26 对应的 ram 起始地址
0x6B	00h	RW	[7:6]	Osdc26_pal_src	Osd block 26 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc26_blink	osd block 26 闪烁区域闪烁频率调节
0x6C	00h	RW	[31:25]	---	reserved
			[24:16]	Char27_size_y	Osd block27 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char27_size_x	Osd block27 字符大小，每个 icon 的水平像素点数
0x6D	00h	RW	[31:24]	---	reserved

			[23:12]	Osdc27_posy	osd block 27 起始点像素的垂直位置
			[11:0]	Osdc27_posx	osd block 27 起始点像素的水平位置
0x6E	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc27_addr	确定 osd block 27 对应的 ram 起始地址
0x6F	00h	RW	[7:6]	Osdc27_pal_src	Osd block 27 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc27_blink	osd block 27 闪烁区域闪烁频率调节
0x70	00h	RW	[31:25]	---	reserved
			[24:16]	Char28_size_y	Osd block28 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char28_size_x	Osd block28 字符大小，每个 icon 的水平像素点数
0x71	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc28_posy	osd block 28 起始点像素的垂直位置
			[11:0]	Osdc28_posx	osd block 28 起始点像素的水平位置
0x72	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc28_addr	确定 osd block 28 对应的 ram 起始地址
0x73	00h	RW	[7:6]	Osdc28_pal_src	Osd block 28 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc28_blink	osd block 28 闪烁区域闪烁频率调节
0x74	00h	RW	[31:25]	---	reserved
			[24:16]	Char29_size_y	Osd block29 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved

			[8:0]	Char29_size_x	Osd block29 字符大小，每个 icon 的水平像素点数
0x75	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc29_posy	osd block 29 起始点像素的垂直位置
			[11:0]	Osdc29_posx	osd block 29 起始点像素的水平位置
0x76	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc29_addr	确定 osd block 29 对应的 ram 起始地址
0x77	00h	RW	[7:6]	Osdc29_pal_src	Osd block 29 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc29_blink	osd block 29 闪烁区域闪烁频率调节
0x78	00h	RW	[31:25]	---	reserved
			[24:16]	Char30_size_y	Osd block30 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char30_size_x	Osd block30 字符大小，每个 icon 的水平像素点数
0x79	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc30_posy	osd block 30 起始点像素的垂直位置
			[11:0]	Osdc30_posx	osd block 30 起始点像素的水平位置
0x7A	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc30_addr	确定 osd block 30 对应的 ram 起始地址
0x7B	00h	RW	[7:6]	Osdc30_pal_src	Osd block 30 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc30_blink	osd block 30 闪烁区域闪烁频率调节
0x7C	00h	RW	[31:25]	---	reserved

			[24:16]	Char31_size_y	Osd block31 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char31_size_x	Osd block31 字符大小，每个 icon 的水平像素点数
0x7D	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc31_posy	osd block 31 起始点像素的垂直位置
			[11:0]	Osdc31_posx	osd block 31 起始点像素的水平位置
0x7E	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc31_addr	确定 osd block 31 对应的 ram 起始地址
0x7F	00h	RW	[7:6]	Osdc31_pal_src	Osd block 31 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc31_blink	osd block 31 闪烁区域闪烁频率调节
0x80	00h	RW	[31:25]	---	reserved
			[24:16]	Char32_size_y	Osd block32 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char32_size_x	Osd block32 字符大小，每个 icon 的水平像素点数
0x81	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc32_posy	osd block 32 起始点像素的垂直位置
			[11:0]	Osdc32_posx	osd block 32 起始点像素的水平位置
0x82	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc32_addr	确定 osd block 32 对应的 ram 起始地址
0x83	00h	RW	[7:6]	Osdc32_pal_src	Osd block 32 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像

		RW	[5:0]	Osdc32_blink	osd block 32 闪烁区域闪烁频率调节
0x84	00h	RW	[31:25]	---	reserved
			[24:16]	Char33_size_y	Osd block33 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char33_size_x	Osd block33 字符大小，每个 icon 的水平像素点数
0x85	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc33_posy	osd block 33 起始点像素的垂直位置
			[11:0]	Osdc33_posx	osd block 33 起始点像素的水平位置
0x86	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc33_addr	确定 osd block 33 对应的 ram 起始地址
0x87	00h	RW	[7:6]	Osdc33_pal_src	Osd block 33 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc33_blink	osd block 33 闪烁区域闪烁频率调节
0x88	00h	RW	[31:25]	---	reserved
			[24:16]	Char34_size_y	Osd block34 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char34_size_x	Osd block34 字符大小，每个 icon 的水平像素点数
0x89	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc34_posy	osd block 34 起始点像素的垂直位置
			[11:0]	Osdc34_posx	osd block 34 起始点像素的水平位置
0x8A	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc34_addr	确定 osd block 34 对应的 ram 起始地址

0x8B	00h	RW	[7:6]	Osdc34_pal_src	Osd block 34 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc34_blink	osd block 34 闪烁区域闪烁频率调节
0x8C	00h	RW	[31:25]	---	reserved
			[24:16]	Char35_size_y	Osd block35 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char35_size_x	Osd block35 字符大小，每个 icon 的水平像素点数
0x8D	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc35_posy	osd block 35 起始点像素的垂直位置
			[11:0]	Osdc35_posx	osd block 35 起始点像素的水平位置
0x8E	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc35_addr	确定 osd block 35 对应的 ram 起始地址
0x8F	00h	RW	[7:6]	Osdc35_pal_src	Osd block 35 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc35_blink	osd block 35 闪烁区域闪烁频率调节
0x90	00h	RW	[31:25]	---	reserved
			[24:16]	Char36_size_y	Osd block36 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char36_size_x	Osd block36 字符大小，每个 icon 的水平像素点数
0x91	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc36posy	osd block 36 起始点像素的垂直位置
			[11:0]	Osdc36_posx	osd block 36 起始点像素的水平位置
0x92	00h	RW	[31:17]	---	reserved

			[16:0]	Osdc36_addr	确定 osd block 36 对应的 ram 起始地址
0x93	00h	RW	[7:6]	Osdc36_pal_src	Osd block 36 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc36_blink	osd block 36 闪烁区域闪烁频率调节
0x94	00h	RW	[31:25]	---	reserved
			[24:16]	Char37_size_y	Osd block37 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char37_size_x	Osd block37 字符大小，每个 icon 的水平像素点数
0x95	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc37_posy	osd block 37 起始点像素的垂直位置
			[11:0]	Osdc37_posx	osd block 37 起始点像素的水平位置
0x96	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc37_addr	确定 osd block 37 对应的 ram 起始地址
0x97	00h	RW	[7:6]	Osdc37_pal_src	Osd block 37 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc37_blink	osd block 37 闪烁区域闪烁频率调节
0x98	00h	RW	[31:25]	---	reserved
			[24:16]	Char38_size_y	Osd block38 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char38_size_x	Osd block38 字符大小，每个 icon 的水平像素点数
0x99	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc38_posy	osd block 38 起始点像素的垂直位置

			[11:0]	Osdc38_posx	osd block 38 起始点像素的水平位置
0x9A	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc38_addr	确定 osd block 38 对应的 ram 起始地址
0x9B	00h	RW	[7:6]	Osdc38_pal_src	Osd block 38 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc38_blink	osd block 38 闪烁区域闪烁频率调节
0x9C	00h	RW	[31:25]	---	reserved
			[24:16]	Char39_size_y	Osd block39 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char39_size_x	Osd block39 字符大小，每个 icon 的水平像素点数
0x9D	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc39_posy	osd block 39 起始点像素的垂直位置
			[11:0]	Osdc39_posx	osd block 39 起始点像素的水平位置
0x9E	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc39_addr	确定 osd block 39 对应的 ram 起始地址
0x9F	00h	RW	[7:6]	Osdc39_pal_src	Osd block 39 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc39_blink	osd block 39 闪烁区域闪烁频率调节
0xA0	00h	RW	[31:25]	---	reserved
			[24:16]	Char40_size_y	Osd block40 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char40_size_x	Osd block40 字符大小，每个 icon 的水平像素点数
0xA1	00h	RW	[31:24]	---	reserved

			[23:12]	Osdc40_posy	osd block 40 起始点像素的垂直位置
			[11:0]	Osdc40_posx	osd block 40 起始点像素的水平位置
0xA2	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc40_addr	确定 osd block 40 对应的 ram 起始地址
0xA3	00h	RW	[7:6]	Osdc40_pal_src	Osd block 40 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc40_blink	osd block 40 闪烁区域闪烁频率调节
0xA4	00h	RW	[31:25]	---	reserved
			[24:16]	Char41_size_y	Osd block41 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char41_size_x	Osd block41 字符大小，每个 icon 的水平像素点数
0xA5	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc41_posy	osd block 41 起始点像素的垂直位置
			[11:0]	Osdc41_posx	osd block 41 起始点像素的水平位置
0xA6	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc41_addr	确定 osd block 41 对应的 ram 起始地址
0xCPU	00h	RW	[7:6]	Osdc41_pal_src	Osd block 41 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc41_blink	osd block 41 闪烁区域闪烁频率调节
0xA8	00h	RW	[31:25]	---	reserved
			[24:16]	Char42_size_y	Osd block42 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved

			[8:0]	Char42_size_x	Osd block42 字符大小，每个 icon 的水平像素点数
0xA9	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc42_posy	osd block 42 起始点像素的垂直位置
			[11:0]	Osdc42_posx	osd block 42 起始点像素的水平位置
0xAA	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc42_addr	确定 osd block 42 对应的 ram 起始地址
0xAB	00h	RW	[7:6]	Osdc42_pal_src	Osd block 42 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc42_blink	osd block 42 闪烁区域闪烁频率调节
0xAC	00h	RW	[31:25]	---	reserved
			[24:16]	Char43_size_y	Osd block43 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char43_size_x	Osd block43 字符大小，每个 icon 的水平像素点数
0xAD	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc43_posy	osd block 43 起始点像素的垂直位置
			[11:0]	Osdc43_posx	osd block 43 起始点像素的水平位置
0xAE	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc43_addr	确定 osd block 43 对应的 ram 起始地址
0xAF	00h	RW	[7:6]	Osdc43_pal_src	Osd block 43 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc43_blink	osd block 43 闪烁区域闪烁频率调节
0xB0	00h	RW	[31:25]	---	reserved

			[24:16]	Char44_size_y	Osd block44 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char44_size_x	Osd block44 字符大小，每个 icon 的水平像素点数
0xB1	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc44_posy	osd block 44 起始点像素的垂直位置
			[11:0]	Osdc44_posx	osd block 44 起始点像素的水平位置
0xB2	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc44_addr	确定 osd block 44 对应的 ram 起始地址
0xB3	00h	RW	[7:6]	Osdc44_pal_src	Osd block 44 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc44_blink	osd block 44 闪烁区域闪烁频率调节
0xB4	00h	RW	[31:25]	---	reserved
			[24:16]	Char45_size_y	Osd block45 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char45_size_x	Osd block45 字符大小，每个 icon 的水平像素点数
0xB5	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc45_posy	osd block 45 起始点像素的垂直位置
			[11:0]	Osdc45_posx	osd block 45 起始点像素的水平位置
0xB6	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc45_addr	确定 osd block 45 对应的 ram 起始地址
0xB7	00h	RW	[7:6]	Osdc45_pal_src	Osd block 45 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像

		RW	[5:0]	Osdc45_blink	osd block 45 闪烁区域闪烁频率调节
0xB8	00h	RW	[31:25]	---	reserved
			[24:16]	Char46_size_y	Osd block46 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char46_size_x	Osd block46 字符大小，每个 icon 的水平像素点数
0xB9	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc46_posy	osd block 46 起始点像素的垂直位置
			[11:0]	Osdc46_posx	osd block46 起始点像素的水平位置
0xBA	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc46_addr	确定 osd block 46 对应的 ram 起始地址
0xBB	00h	RW	[7:6]	Osdc46_pal_src	Osd block 46 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc46_blink	osd block 46 闪烁区域闪烁频率调节
0xBC	00h	RW	[31:25]	---	reserved
			[24:16]	Char47_size_y	Osd block47 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char47_size_x	Osd block47 字符大小，每个 icon 的水平像素点数
0xBD	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc47_posy	osd block 47 起始点像素的垂直位置
			[11:0]	Osdc47_posx	osd block 47 起始点像素的水平位置
0xBE	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc47_addr	确定 osd block 47 对应的 ram 起始地址

0xBF	00h	RW	[7:6]	Osdc47_pal_src	Osd block 47 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc47_blink	osd block 47 闪烁区域闪烁频率调节
0xC0	00h	RW	[31:25]	---	reserved
			[24:16]	Char48_size_y	Osd block48 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char48_size_x	Osd block48 字符大小，每个 icon 的水平像素点数
0xC1	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc48_posy	osd block 48 起始点像素的垂直位置
			[11:0]	Osdc48_posx	osd block 48 起始点像素的水平位置
0xC2	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc48_addr	确定 osd block 48 对应的 ram 起始地址
0xC3	00h	RW	[7:6]	Osdc48_pal_src	Osd block 48 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc48_blink	osd block 48 闪烁区域闪烁频率调节
0xC4	00h	RW	[31:25]	---	reserved
			[24:16]	Char49_size_y	Osd block49 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char49_size_x	Osd block49 字符大小，每个 icon 的水平像素点数
0xC5	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc49_posy	osd block 49 起始点像素的垂直位置
			[11:0]	Osdc49_posx	osd block 49 起始点像素的水平位置
0xC6	00h	RW	[31:17]	---	reserved

			[16:0]	Osdc49_addr	确定 osd block 49 对应的 ram 起始地址
0xC7	00h	RW	[7:6]	Osdc49_pal_src	Osd block 49 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc49_blink	osd block 49 闪烁区域闪烁频率调节
0xC8	00h	RW	[31:25]	---	reserved
			[24:16]	Char50_size_y	Osd block50 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char50_size_x	Osd block50 字符大小，每个 icon 的水平像素点数
0xC9	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc50_posy	osd block 50 起始点像素的垂直位置
			[11:0]	Osdc50_posx	osd block50 起始点像素的水平位置
0xCA	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc50_addr	确定 osd block 50 对应的 ram 起始地址
0xCB	00h	RW	[7:6]	Osdc50_pal_src	Osd block 50 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc50_blink	osd block 50 闪烁区域闪烁频率调节
0xCC	00h	RW	[31:25]	---	reserved
			[24:16]	Char51_size_y	Osd block15 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char51_size_x	Osd block51 字符大小，每个 icon 的水平像素点数
0xCD	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc51_posy	osd block 51 起始点像素的垂直位置

			[11:0]	Osdc51_posx	osd block 51 起始点像素的水平位置
0xCE	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc51_addr	确定 osd block 51 对应的 ram 起始地址
0xCF	00h	RW	[7:6]	Osdc51_pal_src	Osd block 51 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc51_blink	osd block 51 闪烁区域闪烁频率调节
0xD0	00h	RW	[31:25]	---	reserved
			[24:16]	Char52_size_y	Osd block52 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char52_size_x	Osd block52 字符大小，每个 icon 的水平像素点数
0xD1	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc52_posy	osd block 52 起始点像素的垂直位置
			[11:0]	Osdc52_posx	osd block 52 起始点像素的水平位置
0xD2	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc52_addr	确定 osd block 52 对应的 ram 起始地址
0xD3	00h	RW	[7:6]	Osdc52_pal_src	Osd block 52 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc52_blink	osd block 52 闪烁区域闪烁频率调节
0xD4	00h	RW	[31:25]	---	reserved
			[24:16]	Char53_size_y	Osd block53 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char53_size_x	Osd block53 字符大小，每个 icon 的水平像素点数
0xD5	00h	RW	[31:24]	---	reserved

			[23:12]	Osdc53_posy	osd block 53 起始点像素的垂直位置
			[11:0]	Osdc53_posx	osd block 53 起始点像素的水平位置
0xD6	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc53_addr	确定 osd block 53 对应的 ram 起始地址
0xD7	00h	RW	[7:6]	Osdc53_pal_src	Osd block 53 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc53_blink	osd block 53 闪烁区域闪烁频率调节
0xD8	00h	RW	[31:25]	---	reserved
			[24:16]	Char54_size_y	Osd block54 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char54_size_x	Osd block54 字符大小，每个 icon 的水平像素点数
0xD9	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc54_posy	osd block 54 起始点像素的垂直位置
			[11:0]	Osdc54_posx	osd block54 起始点像素的水平位置
0xDA	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc54_addr	确定 osd block 54 对应的 ram 起始地址
0xDB	00h	RW	[7:6]	Osdc54_pal_src	Osd block 54 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc54_blink	osd block 54 闪烁区域闪烁频率调节
0xDC	00h	RW	[31:25]	---	reserved
			[24:16]	Char55_size_y	Osd block55 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved

			[8:0]	Char55_size_x	Osd block55 字符大小，每个 icon 的水平像素点数
0xDD	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc55_posy	osd block 55 起始点像素的垂直位置
			[11:0]	Osdc55_posx	osd block 55 起始点像素的水平位置
0xDE	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc55_addr	确定 osd block 55 对应的 ram 起始地址
0xDF	00h	RW	[7:6]	Osdc55_pal_src	Osd block 55 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc55_blink	osd block 55 闪烁区域闪烁频率调节
0xE0	00h	RW	[31:25]	---	reserved
			[24:16]	Char56_size_y	Osd block56 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char56_size_x	Osd block56 字符大小，每个 icon 的水平像素点数
0xE1	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc56_posy	osd block 56 起始点像素的垂直位置
			[11:0]	Osdc56_posx	osd block 56 起始点像素的水平位置
0xE2	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc56_addr	确定 osd block 56 对应的 ram 起始地址
0xE3	00h	RW	[7:6]	Osdc56_pal_src	Osd block 56 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc56_blink	osd block 56 闪烁区域闪烁频率调节
0xE4	00h	RW	[31:25]	---	reserved

			[24:16]	Char57_size_y	Osd block57 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char57_size_x	Osd block57 字符大小，每个 icon 的水平像素点数
0xE5	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc57_posy	osd block 57 起始点像素的垂直位置
			[11:0]	Osdc57_posx	osd block 57 起始点像素的水平位置
0xE6	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc57_addr	确定 osd block 57 对应的 ram 起始地址
0xE7	00h	RW	[7:6]	Osdc57_pal_src	Osd block 57 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc57_blink	osd block 57 闪烁区域闪烁频率调节
0xE8	00h	RW	[31:25]	---	reserved
			[24:16]	Char58_size_y	Osd block58 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char58_size_x	Osd block58 字符大小，每个 icon 的水平像素点数
0xE9	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc58_posy	osd block 58 起始点像素的垂直位置
			[11:0]	Osdc58_posx	osd block58 起始点像素的水平位置
0xEA	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc58_addr	确定 osd block 58 对应的 ram 起始地址
0xEB	00h	RW	[7:6]	Osdc58_pal_src	Osd block 58 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像

		RW	[5:0]	Osdc58_blink	osd block 58 闪烁区域闪烁频率调节
0xEC	00h	RW	[31:25]	---	reserved
			[24:16]	Char59_size_y	Osd block59 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char59_size_x	Osd block59 字符大小，每个 icon 的水平像素点数
0xED	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc59_posy	osd block 59 起始点像素的垂直位置
			[11:0]	Osdc59_posx	osd block 59 起始点像素的水平位置
0xEE	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc59_addr	确定 osd block 59 对应的 ram 起始地址
0xEF	00h	RW	[7:6]	Osdc59_pal_src	Osd block 59 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc59_blink	osd block 59 闪烁区域闪烁频率调节
0xF0	00h	RW	[31:25]	---	reserved
			[24:16]	Char60_size_y	Osd block60 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char60_size_x	Osd block60 字符大小，每个 icon 的水平像素点数
0xF1	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc60_posy	osd block 60 起始点像素的垂直位置
			[11:0]	Osdc60_posx	osd block 60 起始点像素的水平位置
0xF2	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc60_addr	确定 osd block 60 对应的 ram 起始地址

0xF3	00h	RW	[7:6]	Osdc60_pal_src	Osd block 60 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc60_blink	osd block 60 闪烁区域闪烁频率调节
0xF4	00h	RW	[31:25]	---	reserved
			[24:16]	Char61_size_y	Osd block61 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char61_size_x	Osd block61 字符大小，每个 icon 的水平像素点数
0xF5	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc61_posy	osd block 61 起始点像素的垂直位置
			[11:0]	Osdc61_posx	osd block 61 起始点像素的水平位置
0xF6	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc61_addr	确定 osd block 61 对应的 ram 起始地址
0xF7	00h	RW	[7:6]	Osdc61_pal_src	Osd block 61 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc61_blink	osd block 61 闪烁区域闪烁频率调节
0xF8	00h	RW	[31:25]	---	reserved
			[24:16]	Char62_size_y	Osd block62 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char62_size_x	Osd block62 字符大小，每个 icon 的水平像素点数
0xF9	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc62_posy	osd block 62 起始点像素的垂直位置
			[11:0]	Osdc62_posx	osd block62 起始点像素的水平位置
0xFA	00h	RW	[31:17]	---	reserved

			[16:0]	Osdc62_addr	确定 osd block 62 对应的 ram 起始地址
0xFB	00h	RW	[7:6]	Osdc62_pal_src	Osd block 62 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc62_blink	osd block 62 闪烁区域闪烁频率调节
0xFC	00h	RW	[31:25]	---	reserved
			[24:16]	Char63_size_y	Osd block63 字符大小，每个 icon 的垂直行数
			[15:9]	---	reserved
			[8:0]	Char63_size_x	Osd block63 字符大小，每个 icon 的水平像素点数
0xFD	00h	RW	[31:24]	---	reserved
			[23:12]	Osdc63_posy	osd block 63 起始点像素的垂直位置
			[11:0]	Osdc63_posx	osd block 63 起始点像素的水平位置
0xFE	00h	RW	[31:17]	---	reserved
			[16:0]	Osdc63_addr	确定 osd block 63 对应的 ram 起始地址
0xFF	00h	RW	[7:6]	Osdc63_pal_src	Osd block 63 调色盘选择 [6]: 0: 0 号调色盘 1: 1 号调色盘 [7]: 0: OSD 覆盖前端图像 1: OSD 不覆盖前端图像
		RW	[5:0]	Osdc63_blink	osd block 63 闪烁区域闪烁频率调节
OSD2 Register(Address:0x55500C00 + addr*4)					
0x00 ~0xF F	00h	RW	[23:0]	Osd_palette0_data	Osd 0 号调色盘 0~255 数据
OSD3 Register(Address:0x55501000 + addr*4)					
0x00 ~0xF F	00h	RW	[23:0]	Osd_palette1_data	Osd 1 号调色盘 0~255 数据

30. CRC

30.1 Feature

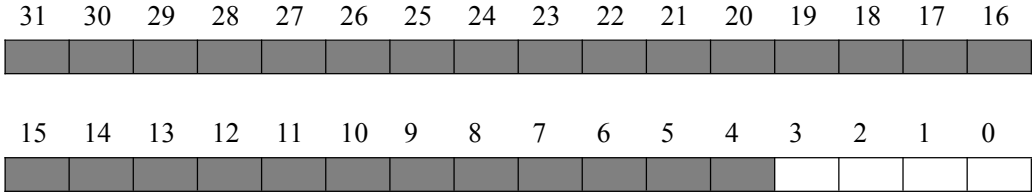
Support CRC-8,CRC-16,CRC-32.

30.2 Register descriptions

AHB_CRC_BASE: 0x40E0_0000

30.2.1 CRC_EN_REG

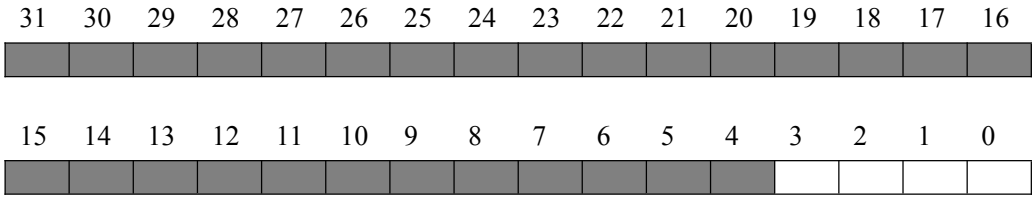
Offset_Address: 0x0000_0000



Bit	Type	Reset	Description
31-1	R	0	Reserved
0	R/W	0	CRC_en

30.2.2 CRC_CTRL_REG

Offset_Address: 0x0000_0004

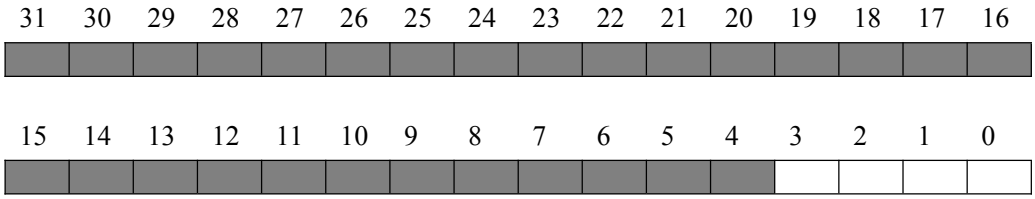


Bit	Type	Reset	Description
31-11	R	0	Reserved
10	R/W	0	CRC_d32_start CRC 数据有效位 32 位使能 0：disable；1：enable；
9	R/W	0	CRC_d16_start CRC 数据有效位 16 位使能 0：disable；1：enable；
8	R/W	0	CRC_d8_start CRC 数据有效位 8 位使能 0：disable；1：enable；
7	R	0	reserved
6	R/W	0	Seed_init CRC 初始值使能 0：Normal data; 1: CRC Initial data;
5-4	R	0	reserved

3-2	R/W	0	poly_vect CRC 计算多项式选择 00: CRC32； 01: CRC16; 10/11: CRC8;
1	R/W	0	swap_vect CRC 结果高低位交换 0: 不交换； 1：交换；
0	R/W	0	Inv_vect CRC 结果按位取反 0: 不取反； 1：取反；

30.2.3 CRC_DATA_REG

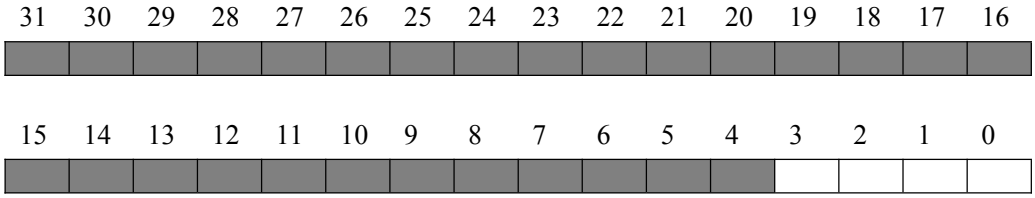
Offset_Address: 0x0000_0008



Bit	Type	Reset	Description
31-0	R/W	0	CRC_DATA

30.2.4 CRC_INIT_REG

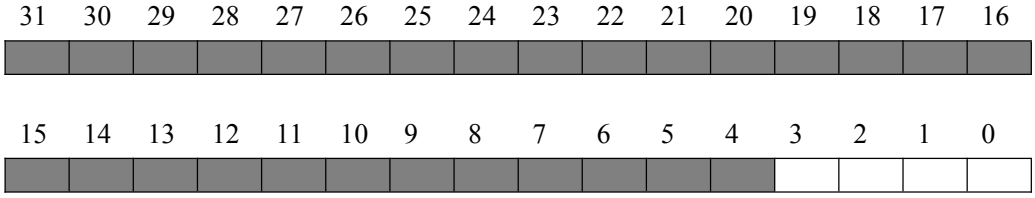
Offset_Address: 0x0000_000C



Bit	Type	Reset	Description
31-0	R/W	0	CRC_INIT_DATA

30.2.5 CRC_RESULT_REG

Offset_Address: 0x0000_0010



Bit	Type	Reset	Description
31-0	R	0	CRC_RESULT

31. SEMA4

31.1 Feature

Support 64 gates.

31.2 Register Descriptions

SEMA4_BASE: 0x40F00000

31.2.1 GATE

The GATE register can lock by cp0 or cp1. Only processor which locks the door can unlock it.

Semaphore Gate n Register (0~63)

Offset_Address: 0x40f0_0000~0x40f0_00FC

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

Bit	Type	Reset	Description
7-2	R/W	0	**Reserved**
1-0	R	0	**GTFSM** **00 The gate is unlocked (free).** **01 The gate has been locked by processor 0.** **10 The gate has been locked by processor 1.** **11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.**

31.2.2 CPnINE

中断通知使能寄存器。

Semaphore Processor 0 IRQ Notification Enable (0-31)

Semaphore Processor 0 IRQ Notification Enable (32-63)

Semaphore Processor 1 IRQ Notification Enable (0-31)

Semaphore Processor 1 IRQ Notification Enable (32-63)

Offset_Address: 0x40f0_0100-0x40f0_0110

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32

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

Bit	Type	Reset	Description
63-0	R/W	0	每比特位对应一个门寄存器 0 禁用通知中断产生 1 启用通知中断产生

31.2.3 CPnNTF

中断请求寄存器，表示尝试锁定门寄存器失败时发出的中断请求通知。

Semaphore Processor 0 IRQ Notification

Semaphore Processor 1 IRQ Notification

Offset_Address: 0x40f0_0120-0x40f0_0130

63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48

47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32

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

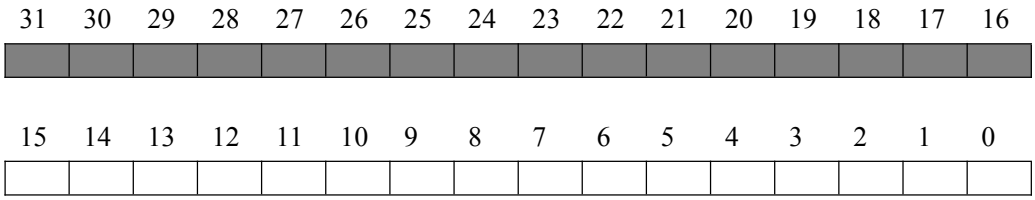
Bit	Type	Reset	Description
63-0	R	0	每比特位表示尝试锁定对应门失败时发出的中断请求通知。

31.2.4 RSTGT

通过遵循特定的双写入允许访问模式来初始化所有的硬件门。

Semaphore (Secure) RESET Gate n

Offset_Address: 0x40f0_0140



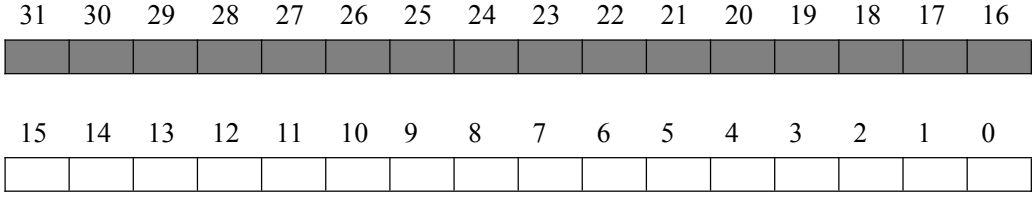
Bit	Type	Reset	Description
15-8	R/W	0	重置门编号。这个 8bit 字段制定要复位的特定硬件门。该字段在第二次写入时更新。 该字段包含门编号的十六进制值。如果 RSTGTN<64,则重置 RSTGTN 定义的单个门，否则复位所有门。相应的 IRQ 通知状态机器也被重置。
7-0	W	0	**RSTGDP** 重置门数据模式。在连续的两次写入中，使用特殊数据模式访问只写字段，从而启用门重启机制。第一次写入，RSTGDP = 0xe2，第二次写入 RSTGDP = 0x1d。
7-6	R	0	保留，通常读数为 0。
5-4	R	0	**RSTGSM** 复位门有限状态机。读取 SEMA4_RSTGT 寄存器返回编码状态机值。复位状态机是一个 2 位三态 implementation，定义如下： 00 空闲。等待第一个数据模式写入 01 等待第二个数据模式写入 10 2-写序列已经完成。生成制定的门重置。在复位被执行后，本机器返回到初始状态（等待第一数据写入）模式。注意 RSTGSM =0b10 状态仅对单一机器周期有效，使用读操作不可能返回该值。 11 这种状态编码永远不会被使用并因此被保留。
3	R	0	保留，通常读数为 0。
2-0	R	0	**RSTGMS** 重置门总线主机。这个 3bit 只读字段记录了执行门复位函数的总线主机的逻辑编号。复位函数要求该寄存器的两次连续写入必须由同一总线主机发起才能成功。每次对该寄存器执行写操作的时候，该字段都会更新。

31.2.5 RSTNTF

通过遵循特定的双写访问流程来初始化 IRQ 通知。

Semaphore (Secure) RESET IRQ Notification

Offset_Address: 0x40f0_0150



Bit	Type	Reset	Description
15-8	R/W	0	重置通知编号。这个 8bit 字段指定要重置的特定 IRQ

			通知状态机。该字段在第二次写入时更新。 该字段包含门编号的十六进制值。如果 RSTGTN<64, 则重置 RSTGTN 定义的单个 IRQ 通知，否则复位所有通知。相应的 IRQ 通知状态机器也被重置。
7-0	W	0	**RSTNDP** 重置通知数据模式。在连续的两次写入中，使用指定数据模式访问这个只写字段，从而启用通知重启机制。第一次写入，RSTNDP= 0x47，第二次写入 RSTNDP= 0xb8。
7-6	R	0	保留，通常读数为 0。
5-4	R	0	**RSTNSM** 复位通知有限状态机。读取 SEMA4_RSTNTF 寄存器返回编码状态机值。复位状态机是一个 2 位三态 implementation，定义如下： 00 空闲。等待第一个数据模式写入 01 等待第二个数据模式写入 10 2-写序列已经完成。生成指定的通知重置。在复位被执行后，本机器返回到初始状态（等待第一数据写入）模式。注意 RSTNSM =10 的状态仅对单一机器周期有效，使用读操作不可能返回该值。 11 这种状态编码永远不会被使用并因此被保留。
3	R	0	保留，通常读数为 0。
2-0	R	0	**RSTNMS** 重置通知总线主机。这个 3bit 只读字段记录了执行通知测试功能的总线主机的逻辑编号。复位函数要求该寄存器的两次连续写入必须由同一总线主机发起才能成功。每次对该寄存器执行写操作的时候，该字段都会更新。

31.2.6 CPnINT

中断寄存器

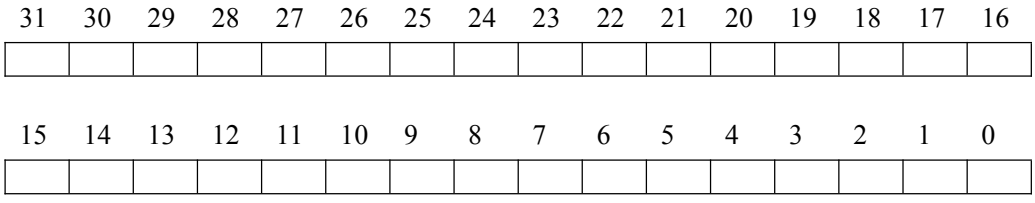
Semaphore Processor 0 Notification (0-31)

Semaphore Processor 0 Notification (32-63)

Semaphore Processor 1 Notification (0-31)

Semaphore Processor 1 Notification (32-63)

Offset_Address: 0x40f0_0160-0x40f0_0178



Bit	Type	Reset	Description
31-0	R	0	每比特位对应一个门寄存器 0 通知中断产生 1 通知中断产生

31.2.7 Periph prot IRQ MCU CLEAR

Peripheral protect IRQ MCU CLEAR

Offset_Address: 0x40f0_0180

Bit	Type	Reset	Description
1	R/W	0	CP0 的 IRQ 清除信号

31.2.8 Periph prot IRQ CPU CLEAR

Peripheral protect IRQ CPU CLEAR

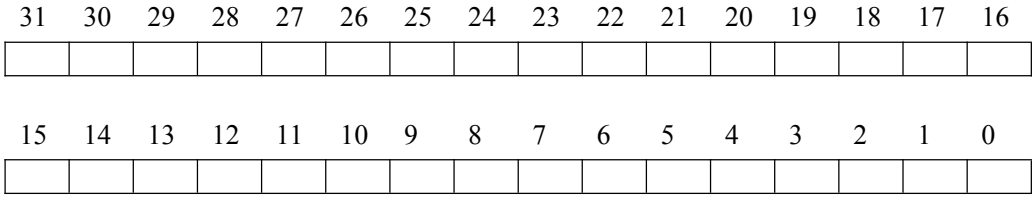
Offset_Address: 0x40f0_0184

Bit	Type	Reset	Description
1	R/W	0	CP1 的 IRQ 清除信号

31.2.9 Periph prot IRQ STATUS MCU

Peripheral protect IRQ STATUS MCU (0-31)

Offset_Address: 0x40f0_0188



Bit	Type	Reset	Description
31	R	0	ADC0 register access violation.
30	R	0	RTC register access violation.
29	R	0	I2S1 register access violation.
28	R	0	I2S0 register access violation.

27	R	0	WDT register access violation.
26	R	0	PWM register access violation.
25	R	0	TIMER register access violation.
24	R	0	GPIO register access violation.
23	R	0	UART3 register access violation.
22	R	0	UART2 register access violation.
21	R	0	UART1 register access violation.
20	R	0	UART0 register access violation.
19	R	0	I2C1 register access violation.
18	R	0	I2C0 register access violation.
17	R	0	ECSPI register access violation.
16	R	0	SYS register access violation.
15	R	0	CRC register access violation.
14	R	0	SPI1 register access violation.
13	R	0	SPI0 register access violation.
12	R	0	WRAP register access violation.
11	R	0	ETH register access violation.
10	R	0	BLENDING register access violation.
9	R	0	QOI register access violation.
8	R	0	ITU register access violation.
7	R	0	MFC register access violation.
6	R	0	PXP register access violation.
5	R	0	GPU register access violation.
4	R	0	SDMMC1 register access violation.
3	R	0	LCD register access violation.
2	R	0	DMA register access violation.
1	R	0	SDMMC0 register access violation.
0	R	0	USB register access violation.

31.2.10 Periph prot IRQ STATUS MCU

Peripheral protect IRQ STATUS MCU (32-63)

Offset_Address: 0x40f0_0190

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

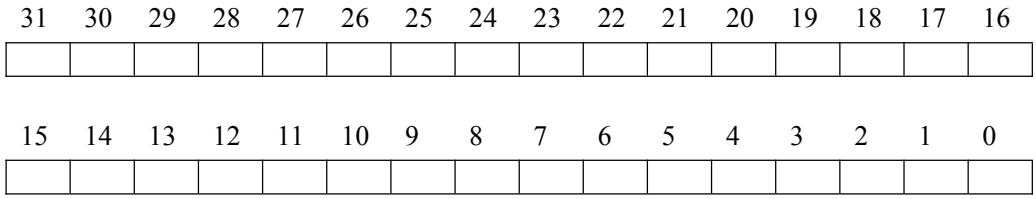
Bit	Type	Reset	Description
31	R	0	Read register access violation.
30	R	0	Write register access violation
29-13	R	0	RESERVED.
12	R	0	SRAMRDC register access violation.

11	R	0	DDRRDC register access violation.
10	R	0	RCRT register access violation.
9	R	0	CSI register access violation.
8	R	0	DSI register access violation.
7	R	0	CABUS register access violation.
6	R	0	CHBUS register access violation.
5	R	0	GBUS register access violation.
4	R	0	VBUS register access violation.
3	R	0	DDRPHY register access violation.
2	R	0	DDRCTL register access violation.
1	R	0	ADC2 register access violation.
0	R	0	ADC1 register access violation.

31.2.11 Periph prot IRQ STATUS CPU

Peripheral protect IRQ STATUS CPU (0-31)

Offset_Address: 0x40f0_0198



Bit	Type	Reset	Description
31	R	0	ADC0 register access violation.
30	R	0	RTC register access violation.
29	R	0	I2S1 register access violation.
28	R	0	I2S0 register access violation.
27	R	0	WDT register access violation.
26	R	0	PWM register access violation.
25	R	0	TIMER register access violation.
24	R	0	GPIO register access violation.
23	R	0	UART3 register access violation.
22	R	0	UART2 register access violation.
21	R	0	UART1 register access violation.
20	R	0	UART0 register access violation.
19	R	0	I2C1 register access violation.
18	R	0	I2C0 register access violation.
17	R	0	ECSPI register access violation.
16	R	0	SYS register access violation.
15	R	0	CRC register access violation.
14	R	0	SPI1 register access violation.
13	R	0	SPI0 register access violation.
12	R	0	WRAP register access violation.

11	R	0	ETH register access violation.
10	R	0	BLENDING register access violation.
9	R	0	QOI register access violation.
8	R	0	ITU register access violation.
7	R	0	MFC register access violation.
6	R	0	PXP register access violation.
5	R	0	GPU register access violation.
4	R	0	SDMMC1 register access violation.
3	R	0	LCD register access violation.
2	R	0	DMA register access violation.
1	R	0	SDMMC0 register access violation.
0	R	0	USB register access violation.

31.2.12 Periph prot IRQ STATUS CPU

Peripheral protect IRQ STATUS CPU (32-63)

Offset_Address: 0x40f0_01a0

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

Bit	Type	Reset	Description
31	R	0	Read register access violation.
30	R	0	Write register access violation
29-13	R	0	RESERVED.
12	R	0	SRAMRDC register access violation.
11	R	0	DDRRDC register access violation.
10	R	0	RCRT register access violation.
9	R	0	CSI register access violation.
8	R	0	DSI register access violation.
7	R	0	CABUS register access violation.
6	R	0	CHBUS register access violation.
5	R	0	GBUS register access violation.
4	R	0	VBUS register access violation.
3	R	0	DDRPHY register access violation.
2	R	0	DDRCTL register access violation.
1	R	0	ADC2 register access violation.
0	R	0	ADC1 register access violation.

32. Pixel Pipeline

32.1 PXP Features

Support input data format:

- 24-bit unpacked RGB(32bpp)
- 24-bit packed RGB(24bpp)
- 16-bit RGB555 or RGB565
- 3-plane YUV/YCbCr in 4:2:0 or 4:2:2
- 2-plane YUV/YCbCr in 4:2:0 or 4:2:2
- 2-plane YUV/YCbCr in 4:2:2
- 32-bit RGB
- 16-bit RGB555、RGB565 or 1555

Support output data format:

- 32-bit RGB
- 24-bit packed RGB(24bpp)
- 16-bit RGB555、RGB565 or 1555
- YUV 4:4:4/4:2:2 1-plane
- YUV 4:2:2/4:2:0 2-plane

32.2 PXP Register Descriptions

PXP_BASE：0x40600000

PXP Control Register

Offset_Address: 0x0000_0000

Bits	Name	RW	Reset	Description
31	**SFTRST**	RW	0x1	Set this bit to zero to enable normal PXP operation. Set this bit to one (default) to disable clocking with the PXP and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the PXP block to its default state.
30	**CLKGATE**	RW	0x1	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29:28	**RSVD2**	RO	0x0	Reserved, always set to zero.
27:26	**INTERLACED_OUTPUT**	RW	0x0	Determines how the PXP writes it's output data. Output interlacing should not be used in conjunction with input interlacing. Splitting frames into fields is most efficient using output interlacing. 2-plane output formats AND interlaced output is NOT supported. PROGRESSIVE = 0x0 All data written in progressive format to the OUTBUF Pointer. FIELD0 = 0x1 Interlaced output: only data for field 0 is written to the OUTBUF Pointer. FIELD1 = 0x2 Interlaced output: only data for field 1 is written to the OUTBUF2 Pointer. INTERLACED = 0x3 Interlaced output: data for field 0 is written to OUTBUF and data for field 1 is written to OUTBUF2.

25:24	**INTERLACED_INPUT**	RW	0x0	When set, causes the fetch side of the PXP to fetch every other line from the source buffers. This effectively produces one field of interlaced output data. Scaling should NOT be enabled for interlaced operation and only overlays with boundaries on 8x16 multiples are supported. PROGRESSIVE = 0x0 All data will be read and processed in progressive format. FIELD0 = 0x2 Interlaced, Field 0: only data for field 0 (even lines) is read/processed. FIELD1 = 0x3 Interlaced, Field 1: only data for field 1 (odd lines) is read/processed.
23	**BLOCK_SIZE**	RW	0x0	Select the block size to process. 8X8 = 0x0 Process 8x8 pixel blocks. 16X16 = 0x1 Process 16x16 pixel blocks.
22	**ALPHA_OUTPUT**	RW	0x0	Indicates that the output buffer pixels should retain their alpha value from the computed alpha for that pixel. If 0, then the ALPHA field from the output buffer parameters register will be used.
21	**IN_PLACE**	RW	0x0	When set, this enables the PXP to perform an alpha blend operation on an existing buffer (output buffer is set to S0 buffer). In this case, the PXP will perform the alpha blending of the overlays into the source buffer. Since only pixels containing an overlay are processed, the PXP does this very efficiently.
20	**DELTA**	RO	0x0	Reserved for future use.
19	**CROP**	RW	0x0	Indicates that the S0 plane should use the cropping register to provide the extents for the output S0 buffer cropping. If not set, the input video cropping extents will be inferred from the S0 WIDTH and HEIGHT fields. When scaling, the CROP bit and controls should be used to specify the scaled image size in the output buffer.
18	**SCALE**	RW	0x0	This bit indicates that the output image should be scaled (only YUV/YCbCr images may be scaled -- RGB scaling is not supported). The XSCALE and YSCALE Register should be programmed accordingly. In addition, the CROP bit and the S0CROP Register should be programmed to ensure that the scaled image is properly cropped in the output buffer. When this bit is zero, the contents of the scaling Register are ignored.
17	**UPSAMPLE**	RO	0x0	Reserved for future use.
16	**SUBSAMPLE**	RO	0x0	Reserved for future use.
15:12	**S0_FORMAT**	RW	0x0	Source 0 buffer format. To select between YUV and YCbCr formats, see bit 31 of the CSCCOEFF0 register. RGB888 = 0x1 32-bit pixels (unpacked 24-bit format) RGB565 = 0x4 16-bit pixels RGB555 = 0x5 16-bit pixels YUV422 = 0x8 16-bit pixels YUV420 = 0x9 16-bit pixels UYVY1P422 = 0xA 16-bit pixels (1-plane U0,Y0,V0,Y1 interleaved bytes) VYUY1P422 = 0xB 16-bit pixels (1-plane V0,Y0,U0,Y1 interleaved bytes) YUV2P422 = 0xC 16-bit pixels (2-plane UV interleaved bytes) YUV2P420 = 0xD 16-bit pixels YVU2P422 = 0xE 16-bit pixels (2-plane VU interleaved bytes) YVU2P420 = 0xF 16-bit pixels

11	**VFLIP**	RW	0x0	Indicates that the output buffer should be flipped vertically (effect applied before rotation).
10	**HFLIP**	RW	0x0	Indicates that the output buffer should be flipped horizontally (effect applied before rotation).
9:8	**ROTATE**	RW	0x0	Indicates the clockwise rotation to be applied at the output buffer. The rotation effect is defined as occurring after the FLIP_X and FLIP_Y permutation. ROT_0 = 0x0 ROT_90 = 0x1 ROT_180 = 0x2 ROT_270 = 0x3
7:4	**OUTBUF_FORMAT**	RW	0x0	Output framebuffer format. The UV byte lanes are synonymous with CbCr byte lanes for YUV output pixel formats. For example, the YUV2P420 format should be selected when the output is YCbCr 2-plane 420 output format. ARGB8888 = 0x0 32-bit pixels RGB888 = 0x1 32-bit pixels (unpacked 24-bit pixel in 32 bit DWORD.) RGB888P = 0x2 24-bit pixels (packed 24-bit format) ARGB1555 = 0x3 16-bit pixels RGB565 = 0x4 16-bit pixels RGB555 = 0x5 16-bit pixels YUV444 = 0x7 32-bit pixels (1-plane XYUV unpacked) UYVY1P422 = 0xA 16-bit pixels (1-plane U0,Y0,V0,Y1 interleaved bytes) VYUY1P422 = 0xB 16-bit pixels (1-plane V0,Y0,U0,Y1 interleaved bytes) YUV2P422 = 0xC 16-bit pixels (2-plane UV interleaved bytes) YUV2P420 = 0xD 16-bit pixels (2-plane UV) YVU2P422 = 0xE 16-bit pixels (2-plane VU interleaved bytes) YVU2P420 = 0xF 16-bit pixels (2-plane VU)
3	**ENABLE_LCD_HANDSHAKE**	RW	0x0	Enable handshake with LCD controller. When this is set, the PXP will not process an entire framebuffer, but will instead process rows of NxN blocks in a double-buffer handshake with the LCDIF. This enables the use of the onboard SRAM for a partial frame buffer.
2	**NEXT_IRQ_ENABLE**	RW	0x0	Next command interrupt enable. When set, the PXP will issue an interrupt when a queued command initiated by a write to the PXP_NEXT register has been loaded into the PXP's Register. This interrupt also indicates that a new command may now be queued.
1	**IRQ_ENABLE**	RW	0x0	Interrupt enable. NOTE: When using the HW_PXP_NEXT functionality to reprogram the PXP, the new value of this bit will be used and may therefore enable or disable an interrupt unintentionally.
0	**ENABLE**	RW	0x0	Enables PXP operation with specified parameters. The ENABLE bit will remain set while the PXP is active and will be cleared once the current operation completes. Software should use the IRQ bit in the HW_PXP_STAT when polling for PXP completion.

PXP Status Register

Offset_Address: 0x0000_0010

Bits	Name	RW	Reset	Description
31:24	**BLOCKX**	RO	0x00	Indicates the X coordinate of the block currently being rendered.
23:1	**BLOCKY**	RO	0x00	Indicates the X coordinate of the block

6				currently being rendered.
15:8	**RSVD2**	RO	0x000000	Reserved, always set to zero.
7:4	**AXI_ERROR_ID**	RO	0x0	Indicates the AXI ID of the failing bus operation.
3	**NEXT_IRQ**	RW	0x0	Indicates that a command issued with the "Next Command"
				functionality has been issued and that a new command may be
				initiated with a write to the PXP_NEXT register.
2	**AXI_READ_ERROR**	RW	0x0	Indicates PXP encountered an AXI read error and processing
				has been terminated.
1	**AXI_WRITE_ERROR**	RW	0x0	Indicates PXP encountered an AXI write error and processing
				has been terminated.
0	**IRQ**	RW	0x0	Indicates current PXP interrupt status. The IRQ is routed
				through the pxp_irq when the IRQ_ENABLE bit in the control
				register is set.

Output Frame Buffer Pointer

Offset_Address: 0x0000_0020

Bits	Name	R W	Reset	Description
31:0	**ADDR**	RW	0x0	Current address pointer for the output frame buffer. The address MUST be word-aligned for proper PXP operation.

Output Frame Buffer Pointer #2

Offset_Address: 0x0000_0030

Bits	Name	R W	Reset	Description
31:0	**ADDR**	RW	0x0	Current address pointer for the output frame buffer. The address MUST be word-aligned for proper PXP operation. Output Framebuffer Pointer #2. This register points to the beginning of the output frame buffer for eitherfield 1 when generating interlaced output or for the UV buffer when in YUV 2-plane output modes. Both interlaced output AND 2-plane output modes are not supported. This register is NOT used as the pointer to the 2nd buffer when in LCDIF_HANDSHAKE mode.

PXP Output Buffer Size

Offset_Address: 0x0000_0040

Bits	Name	R W	Reset	Description

31:24	**ALPHA**	RW	0x00	When generating an output buffer with an alpha component, the value in this field will be used.
23:12	**WIDTH**	RW	0x0	Indicates number of horizontal PIXELS in the image (non-rotated). The image size is not required to be a multiple of 8 pixels. The PXP will handle clipping the pixel output at this boundary.
11:0	**HEIGHT**	RW	0x0	Indicates the number of vertical PIXELS in the image (non-rotated). The image size is not required to be a multiple of 8 pixels. The PXP will handle clipping the pixel output at this

PXP Source 0 (video) Input Buffer Pointer

Offset_Address: 0x0000_0050

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the S0 RGB or Y (luma) input buffer. The address MUST be word-aligned for proper PXP operation.

Source 0 U/Cb or 2 Plane UV Input Buffer Pointer

Offset_Address: 0x0000_0060

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the S0 (video) U/Cb or 2 plane UV Chroma input buffer. The address MUST be word-aligned for proper PXP operation.

Source 0 V/Cr Input Buffer Pointer

Offset_Address: 0x0000_0070

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the S0 (video) U/Cb or 2 plane UV Chroma input buffer. The address MUST be word-aligned for proper PXP operation.

PXP Source 0 (video) Buffer Parameters

Offset_Address: 0x0000_0080

Bits	Name	R W	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the horizontal offset location (in NxN block) of the S0 buffer within the output frame buffer.
23:16	**YBASE**	RW	0x00	This field indicates the vertical offset location (in NxN block) of the S0 buffer within the output frame buffer.
15:8	**WIDTH**	RW	0x0	Indicates number of horizontal NxN blocks in the

				image (non-rotated).
7:0	**HEIGHT**	RW	0x0	Indicates the number of vertical NxN blocks in the image (non-rotated).

Source 0 Background Color

Offset_Address: 0x0000_0090

Bits	Name	R W	Reset	Description
31:0	**COLOR**	RW	0x0	Background color (in 32bpp format) for any pixels not in the S0 buffer range specified in the S0SIZE register.

Source 0 Cropping Register

Offset_Address: 0x0000_00A0

Bits	Name	R W	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the horizontal offset (in terms of N-pixel blocks) into the S0 buffer which is considered the origin of the image. This allows selection of a subset of a source image for processing.
23:16	**YBASE**	RW	0x00	This field indicates the vertical offset (in terms of N-pixel blocks) into the S0 buffer which is considered the origin of the image. This allows selection of a subset of a source image for processing.
15:8	**WIDTH**	RW	0x00	Ouput buffer cropped video width (in terms of N pixel blocks). This field should be programmed to the desired cropped width of the S0 plane in the output buffer. When scaling is not used, this value is effectively the width of the input buffer that should appear in the output buffer. For scaling operations, it's important that this field be programmed to the width of the scaled size of the S0 output image.
7:0	**HEIGHT**	RW	0x00	Input buffer cropped video height (in terms of N pixel blocks). This field should be programmed to the desired cropped height of the S0 plan in the output buffer. When scaling is not used, this value is effectively the height of the input buffer that should appear in the output buffer. For scaling operations, it's important that this field be programmed to the height of the scaled size of the S0 output image.

Source 0 Scale Factor Register

Offset_Address: 0x0000_00B0

Bits	Name	R W	Reset	Description

31	**RSVD2**	RO	0x00	Reserved, always set to zero.
30:16	**YSCALE**	RW	0x1000	This is a three bit integer and 12 bit fractional representation (###.####_####_####) of the Y scaling factor for the S0 source buffer. The maximum value programmed should be 4 since scaling down by a factor greater than 4 is not supported.
15	**RSVD1**	RO	0x00	Reserved, always set to zero.
14:0	**XSCALE**	RW	0x1000	This is a three bit integer and 12 bit fractional representation (###.####_####_####) of the X scaling factor for the S0 source buffer. The maximum value programmed should be 4 since scaling down by a factor greater than 4 is not supported.

Source 0 Scale Offset Register

Offset_Address: 0x0000_00C0

Bits	Name	R W	Reset	Description
31:28	**RSVD2**	RO	0x00	Reserved, always set to zero.
27:16	**YOFFSET**	RW	0x100	This is a 12 bit fractional representation (0.####_####_####) of the Y scaling offset. This represents a fixed block offset which gets added to the scaled block address to determine source data for the scaling engine.
15:12	**RSVD1**	RO	0x00	Reserved, always set to zero.
8:0	**X_OFFSET**	RW	0x000	This is a 12 bit fractional representation (0.####_####_####) of the X scaling offset. This represents a fixed block offset which gets added to the scaled block address to determine source data for the scaling engine.

Color Space Conversion Coefficient Register0

Offset_Address: 0x0000_00D0

Bits	Name	R W	Reset	Description
31	**YCBCR_MODE**	RW	0x0	Set to 1 when performing YCbCr conversion to RGB. Set to 0 when converting YUV to RGB data. This bit changes the behavior of the scaler when performing U/V scaling.
30:29	**RSVD1**	RO	0x00	Reserved, always set to zero.
28:18	**C0**	RW	0x100	Two's compliment Y multiplier coefficient. YUV=0x100 (1.000) YCbCr=0x12A (1.164)

17:9	**UV_OFFSET**	RW	0x000	Two's compliment phase offset implicit for CbCr data. Generally used for YCbCr to RGB conversion. YCbCr=0x180, to 0.5 range)
8:0	**Y_OFFSET**	RW	0x000	Two's compliment amplitude offset implicit in the Y data. For YUV, this is typically 0 and for YCbCr, this is typically -16 (0x1F0)

Color Space Conversion Coefficient Register1

Offset_Address: 0x0000_00E0

Bits	Name	R W	Reset	Description
31:27	**RSVD1**	RO	0x00	Reserved, always set to zero.
26:16	**C1**	RW	0x123	Two's compliment Red V/Cr multiplier coefficient. YUV=0x123 (1.140) YCbCr=0x198 (1.596)
15:11	**RSVD0**	RO	0x00	Reserved, always set to zero.
10:0	**C4**	RW	0x208	Two's compliment Blue U/Cb multiplier coefficient. YUV=0x208 (2.032) YCbCr=0x204 (2.017)

Color Space Conversion Coefficient Register2

Offset_Address: 0x0000_00F0

Bits	Name	R W	Reset	Description
31:27	**RSVD1**	RO	0x00	Reserved, always set to zero.
26:16	**C2**	RW	0x79b	Two's compliment Green V/Cr multiplier coefficient. YUV=0x76B (-0.581) YCbCr=0x730 (-0.813)
15:11	**RSVD0**	RO	0x00	Reserved, always set to zero.
10:0	**C3**	RW	0x76c	Two's compliment Green U/Cb multiplier coefficient. YUV=0x79C (-0.394) YCbCr=0x79C (-0.392)

PXP Next Frame Pointer

Offset_Address: 0x0000_0100

Bits	Name	R W	Reset	Description
31:2	**POINTER**	RW	0x0	A pointer to a data structure containing register values to be used when processing the next frame. The pointer must be 32-bit aligned and should reside in on-chip or off-chip memory.
1	**RSVD**	RO	0x0	Reserved, always set to zero.

0	**ENABLED**	RO	0x0	Indicates that the "next frame" functionality has been enabled. This bit reflects the status of the hardware semaphore indicating that a reload operation is pending at the end of the current frame.

PXP S0 Color Key Low

Offset_Address: 0x0000_0180

Bits	Name	R W	Reset	Description
31:24	**RSVD1**	RO	0x00	Reserved, always set to zero.
23:0	**PIXEL**	RW	0xFFFFFF	Low range of RGB color key applied to S0 buffer. To disable S0 colorkeying, set the low colorkey to 0xFFFFFF and the high colorkey to 0x000000.

PXP S0 Color Key High

Offset_Address: 0x0000_0190

Bits	Name	R W	Reset	Description
31:24	**RSVD1**	RO	0x00	Reserved, always set to zero.
23:0	**PIXEL**	RW	0x0	High range of RGB color key applied to S0 buffer. To disable S0 olorkeying, set the low colorkey to 0xFFFFFF and the high olorkey to 0x000000.

PXP Overlay Color Key Low

Offset_Address: 0x0000_01A0

Bits	Name	R W	Reset	Description
31:24	**RSVD1**	RO	0x00	Reserved, always set to zero.
23:0	**PIXEL**	RW	0xFFFFFF	Low range of RGB color key applied to OL buffer. Each overlay as an independent colorkey enable.

PXP Overlay Color Key High

Offset_Address: 0x0000_01B0

Bits	Name	R W	Reset	Description
31:24	**RSVD1**	RO	0x00	Reserved, always set to zero.
23:0	**PIXEL**	RW	0x0	High range of RGB color key applied to OL buffer. Each overlay as an independent colorkey enable..

PXP Debug Control Register

Offset_Address: 0x0000_01D0

Bits	Name	R W	Reset	Description
31:9	**RSVD**	RO	0x0	Reserved, always set to zero.
8	**Reset_TLB_STATS**	RW	0x0	Fixed read-only value reflecting the MINOR field of the RTL version.
7:0	**SELECT**	RW	0x00	Index into one of the PXP debug Register. The data for the selected register will be returned NONE = 0x0 None CTRL = 0x1 Control Debug S0REGS = 0x2 S0 Debug S0BAX = 0x3 S0 BA X Scale S0BAY = 0x4 S0 BA Y Scale PXBUF = 0x5 PXBUF Debug ROTATION = 0x6 Rotation Debug ROTBUF0 = 0x7 Rotation Buffer 0 ROTBUF1 = 0x8 Rotation Buffer 1 TLBCOUNT = 0xF0 TLB Lookup Count TLBHIT = 0xF1 TLB Hit Count TLBMISS = 0xF2 TLB Miss Count TLBLAT = 0xF3 TLB Latency Count TLBSTATE = 0xF8 TLB State Information

PXP Debug Register

Offset_Address: 0x0000_01E0

Bits	Name	R W	Reset	Description
31:0	**DATA**	RO	0x0	Debug data

ARK PXP VER

Offset_Address: 0x0000_01F0

Bits	Name	R W	Reset	Description
31:0	ARK PXP VER	RO	0x0	ARKMICRO VER Information

PXP Overlay 0 Buffer Pointer

Offset_Address: 0x0000_0200

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the overlay 0 buffer. The address MUST be word-aligned for proper PXP operation.

PXP Overlay 0 Size

Offset_Address: 0x0000_0210

Bits	Name	R W	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the X-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
23:16	**YBASE**	RW	0x00	This field indicates the Y-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
15:8	**WIDTH**	RW	0x0	Indicates number of horizontal NxN blocks in the image (non-rotated).
7:0	**HEIGHT**	RW	0x0	Indicates the number of vertical NxN blocks in the image (non-rotated).

PXP Overlay 0 Parameters

Offset_Address: 0x0000_0220

Bits	Name	R W	Reset	Description
31:20	**RSVD1**	RO	0x0000	Reserved, always set to zero.
19:16	**ROP**	RW	0x0	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CNTL field. MASKOL = 0x0 OL AND S0 MASKNOTOL = 0x1 nOL AND S0 MASKOLNOT = 0x2 OL AND nS0 MERGEOL = 0x3 OL OR S0 MERGENOTOL = 0x4 nOL OR S0 MERGEOLNOT = 0x5 OL OR nS0 NOTCOPYOL = 0x6 nOL NOT = 0x7 nS0 NOTMASKOL = 0x8 OL NAND S0 NOTMERGEOL = 0x9 OL NOR S0 XOROL = 0xA OL XOR S0 NOTXOROL = 0xB OL XNOR S0
15:8	**ALPHA**	RW	0x0	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE bits are set. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when enabled in the ALPHA_CNTL field.
7:4	**FORMAT**	RW	0x0	Indicates the input buffer format for overlay 0. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x1 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x3 16-bit pixels with alpha RGB565 = 0x4 16-bit pixels without alpha RGB555 = 0x5 16-bit pixels without alpha
3	**ENABLE_COLORKEY**	RW	0x0	Indicates that colorkey functionality is enabled for this overlay. Pixels found in the overlay colorkey range will be displayed as transparent (the S0 pixel will be used).
2:1	**ALPHA_CNTL**	RW	0x0	Determines how the alpha value is constructed for this overlay. Indicates that

				the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the OL pixel alpha value will be used to blend the OL with S0. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the overlay and S0 pixels.
0	**ENABLE**	RW	0x0	Indicates that the overlay is active for this operation.

PXP Overlay 0 Parameters2

Offset_Address: 0x0000_0230

Bits	Name	R W	Reset	Description
31:0	**RSVD**	RO	0x0000	Reserved, always set to zero.

PXP Overlay 1 Buffer Pointer

Offset_Address: 0x0000_0240

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the overlay 1 buffer. The address MUST be word-aligned for proper PXP operation.

PXP Overlay 1 Size

Offset_Address: 0x0000_0250

Bits	Name	R W	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the X-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
23:16	**YBASE**	RW	0x00	This field indicates the Y-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
15:8	**WIDTH**	RW	0x0	Indicates number of horizontal NxN blocks in the image (non-rotated).
7:0	**HEIGHT**	RW	0x0	Indicates the number of vertical NxN blocks in the image (non-rotated).

PXP Overlay 1 Parameters

Offset_Address: 0x0000_0260

Bits	Name	R W	Reset	Description
31:20	**RSVD1**	RO	0x0000	Reserved, always set to zero.
19:16	**ROP**	RW	0x0	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CNTL field. MASKOL = 0x0 OL AND S0 MASKNOTOL = 0x1 nOL AND S0 MASKOLNOT = 0x2 OL AND nS0 MERGEOL = 0x3 OL OR S0 MERGENOTOL = 0x4 nOL OR S0 MERGEOLNOT = 0x5 OL OR nS0 NOTCOPYOL = 0x6 nOL NOT = 0x7 nS0 NOTMASKOL = 0x8 OL NAND S0 NOTMERGEOL = 0x9 OL NOR S0 XOROL = 0xA OL XOR S0 NOTXOROL = 0xB OL XNOR S0
15:8	**ALPHA**	RW	0x0	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE bits are set. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when enabled in the ALPHA_CNTL field.
7:4	**FORMAT**	RW	0x0	Indicates the input buffer format for overlay 0. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x1 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x3 16-bit pixels with alpha RGB565 = 0x4 16-bit pixels without alpha RGB555 = 0x5 16-bit pixels without alpha
3	**ENABLE_COLORKEY**	RW	0x0	Indicates that colorkey functionality is enabled for this overlay. Pixels found in the overlay colorkey range will be displayed as transparent (the S0 pixel will be used).
2:1	**ALPHA_CNTL**	RW	0x0	Determines how the alpha value is constructed for this overlay. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the OL pixel alpha value will be used to blend the OL with S0. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the overlay and S0 pixels.
0	**ENABLE**	RW	0x0	Indicates that the overlay is active for this operation.

PXP Overlay 1 Parameters2

Offset_Address: 0x0000_0270

Bits	Name	R W	Reset	Description
31:0	**RSVD**	RO	0x0000	Reserved, always set to zero.

PXP Overlay 2 Buffer Pointer

Offset_Address: 0x0000_0280

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the overlay 2 buffer. The address MUST be word-aligned for proper PXP operation.

PXP Overlay 2 Size

Offset_Address: 0x0000_0290

Bits	Name	R W	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the X-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
23:16	**YBASE**	RW	0x00	This field indicates the Y-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
15:8	**WIDTH**	RW	0x0	Indicates number of horizontal NxN blocks in the image (non-rotated).
7:0	**HEIGHT**	RW	0x0	Indicates the number of vertical NxN blocks in the image (non-rotated).

PXP Overlay 2 Parameters

Offset_Address: 0x0000_02A0

Bits	Name	R W	Reset	Description
31:20	**RSVD1**	RO	0x0000	Reserved, always set to zero.

19:16	**ROP**	RW	0x0	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CNTL field. MASKOL = 0x0 OL AND S0 MASKNOTOL = 0x1 nOL AND S0 MASKOLNOT = 0x2 OL AND nS0 MERGEOL = 0x3 OL OR S0 MERGENOTOL = 0x4 nOL OR S0 MERGEOLNOT = 0x5 OL OR nS0 NOTCOPYOL = 0x6 nOL NOT = 0x7 nS0 NOTMASKOL = 0x8 OL NAND S0 NOTMERGEOL = 0x9 OL NOR S0 XOROL = 0xA OL XOR S0 NOTXOROL = 0xB OL XNOR S0
15:8	**ALPHA**	RW	0x0	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE bits are set. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when enabled in the ALPHA_CNTL field.
7:4	**FORMAT**	RW	0x0	Indicates the input buffer format for overlay 0. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x1 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x3 16-bit pixels with alpha RGB565 = 0x4 16-bit pixels without alpha RGB555 = 0x5 16-bit pixels without alpha
3	**ENABLE_COLORKEY**	RW	0x0	Indicates that colorkey functionality is enabled for this overlay. Pixels found in the overlay colorkey range will be displayed as transparent (the S0 pixel will be used).
2:1	**ALPHA_CNTL**	RW	0x0	Determines how the alpha value is constructed for this overlay. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the OL pixel alpha value will be used to blend the OL with S0. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the overlay and S0 pixels.
0	**ENABLE**	RW	0x0	Indicates that the overlay is active for this operation.

PXP Overlay 2 Parameters2

Offset_Address: 0x0000_02B0

Bits	Name	R W	Reset	Description
31:0	**RSVD**	RO	0x0000	Reserved, always set to zero.

PXP Overlay 3 Buffer Pointer

Offset_Address: 0x0000_02C0

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the overlay 3 buffer. The address MUST be word-aligned for proper PXP operation.

PXP Overlay 3 Size

Offset_Address: 0x0000_02D0

Bits	Name	R W	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the X-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
23:16	**YBASE**	RW	0x00	This field indicates the Y-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
15:8	**WIDTH**	RW	0x0	Indicates number of horizontal NxN blocks in the image (non-rotated).
7:0	**HEIGHT**	RW	0x0	Indicates the number of vertical NxN blocks in the image (non-rotated).

PXP Overlay 3 Parameters

Offset_Address: 0x0000_02E0

Bits	Name	R W	Reset	Description
31:20	**RSVD1**	RO	0x0000	Reserved, always set to zero.

19:16	**ROP**	RW	0x0	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CNTL field. MASKOL = 0x0 OL AND S0 MASKNOTOL = 0x1 nOL AND S0 MASKOLNOT = 0x2 OL AND nS0 MERGEOL = 0x3 OL OR S0 MERGENOTOL = 0x4 nOL OR S0 MERGEOLNOT = 0x5 OL OR nS0 NOTCOPYOL = 0x6 nOL NOT = 0x7 nS0 NOTMASKOL = 0x8 OL NAND S0 NOTMERGEOL = 0x9 OL NOR S0 XOROL = 0xA OL XOR S0 NOTXOROL = 0xB OL XNOR S0
15:8	**ALPHA**	RW	0x0	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE bits are set. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when enabled in the ALPHA_CNTL field.
7:4	**FORMAT**	RW	0x0	Indicates the input buffer format for overlay 0. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x1 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x3 16-bit pixels with alpha RGB565 = 0x4 16-bit pixels without alpha RGB555 = 0x5 16-bit pixels without alpha
3	**ENABLE_COLORKEY**	RW	0x0	Indicates that colorkey functionality is enabled for this overlay. Pixels found in the overlay colorkey range will be displayed as transparent (the S0 pixel will be used).
2:1	**ALPHA_CNTL**	RW	0x0	Determines how the alpha value is constructed for this overlay. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the OL pixel alpha value will be used to blend the OL with S0. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the overlay and S0 pixels.
0	**ENABLE**	RW	0x0	Indicates that the overlay is active for this operation.

PXP Overlay 3 Parameters2

Offset_Address: 0x0000_02F0

Bits	Name	R W	Reset	Description
31:0	**RSVD**	RO	0x0000	Reserved, always set to zero.

PXP Overlay 4 Buffer Pointer

Offset_Address: 0x0000_0300

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the overlay 4 buffer. The address MUST be word-aligned for proper PXP operation.

PXP Overlay 4 Size

Offset_Address: 0x0000_0310

Bits	Name	R W	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the X-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
23:16	**YBASE**	RW	0x00	This field indicates the Y-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
15:8	**WIDTH**	RW	0x0	Indicates number of horizontal NxN blocks in the image (non-rotated).
7:0	**HEIGHT**	RW	0x0	Indicates the number of vertical NxN blocks in the image (non-rotated).

PXP Overlay 4 Parameters

Offset_Address: 0x0000_0320

Bits	Name	R W	Reset	Description
31:20	**RSVD1**	RO	0x0000	Reserved, always set to zero.

19:16	**ROP**	RW	0x0	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CNTL field. MASKOL = 0x0 OL AND S0 MASKNOTOL = 0x1 nOL AND S0 MASKOLNOT = 0x2 OL AND nS0 MERGEOL = 0x3 OL OR S0 MERGENOTOL = 0x4 nOL OR S0 MERGEOLNOT = 0x5 OL OR nS0 NOTCOPYOL = 0x6 nOL NOT = 0x7 nS0 NOTMASKOL = 0x8 OL NAND S0 NOTMERGEOL = 0x9 OL NOR S0 XOROL = 0xA OL XOR S0 NOTXOROL = 0xB OL XNOR S0
15:8	**ALPHA**	RW	0x0	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE bits are set. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when enabled in the ALPHA_CNTL field.
7:4	**FORMAT**	RW	0x0	Indicates the input buffer format for overlay 0. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x1 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x3 16-bit pixels with alpha RGB565 = 0x4 16-bit pixels without alpha RGB555 = 0x5 16-bit pixels without alpha
3	**ENABLE_COLORKEY**	RW	0x0	Indicates that colorkey functionality is enabled for this overlay. Pixels found in the overlay colorkey range will be displayed as transparent (the S0 pixel will be used).
2:1	**ALPHA_CNTL**	RW	0x0	Determines how the alpha value is constructed for this overlay. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the OL pixel alpha value will be used to blend the OL with S0. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the overlay and S0 pixels.
0	**ENABLE**	RW	0x0	Indicates that the overlay is active for this operation.

PXP Overlay 4 Parameters2

Offset_Address: 0x0000_0330

Bits	Name	R W	Reset	Description
31:0	**RSVD**	RO	0x0000	Reserved, always set to zero.

PXP Overlay 5 Buffer Pointer

Offset_Address: 0x0000_0340

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the overlay 5 buffer. The address MUST be word-aligned for proper PXP operation.

PXP Overlay 5 Size

Offset_Address: 0x0000_0350

Bits	Name	R W	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the X-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
23:16	**YBASE**	RW	0x00	This field indicates the Y-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
15:8	**WIDTH**	RW	0x0	Indicates number of horizontal NxN blocks in the image (non-rotated).
7:0	**HEIGHT**	RW	0x0	Indicates the number of vertical NxN blocks in the image (non-rotated).

PXP Overlay 5 Parameters

Offset_Address: 0x0000_0360

Bits	Name	R W	Reset	Description
31:20	**RSVD1**	RO	0x0000	Reserved, always set to zero.

19:16	**ROP**	RW	0x0	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CNTL field. MASKOL = 0x0 OL AND S0 MASKNOTOL = 0x1 nOL AND S0 MASKOLNOT = 0x2 OL AND nS0 MERGEOL = 0x3 OL OR S0 MERGENOTOL = 0x4 nOL OR S0 MERGEOLNOT = 0x5 OL OR nS0 NOTCOPYOL = 0x6 nOL NOT = 0x7 nS0 NOTMASKOL = 0x8 OL NAND S0 NOTMERGEOL = 0x9 OL NOR S0 XOROL = 0xA OL XOR S0 NOTXOROL = 0xB OL XNOR S0
15:8	**ALPHA**	RW	0x0	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE bits are set. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when enabled in the ALPHA_CNTL field.
7:4	**FORMAT**	RW	0x0	Indicates the input buffer format for overlay 0. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x1 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x3 16-bit pixels with alpha RGB565 = 0x4 16-bit pixels without alpha RGB555 = 0x5 16-bit pixels without alpha
3	**ENABLE_COLORKEY**	RW	0x0	Indicates that colorkey functionality is enabled for this overlay. Pixels found in the overlay colorkey range will be displayed as transparent (the S0 pixel will be used).
2:1	**ALPHA_CNTL**	RW	0x0	Determines how the alpha value is constructed for this overlay. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the OL pixel alpha value will be used to blend the OL with S0. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the overlay and S0 pixels.
0	**ENABLE**	RW	0x0	Indicates that the overlay is active for this operation.

PXP Overlay 5 Parameters2

Offset_Address: 0x0000_0370

Bits	Name	RW	Reset	Description
31:0	**RSVD**	RO	0x0000	Reserved, always set to zero.

PXP Overlay 6 Buffer Pointer

Offset_Address: 0x0000_0380

Bits	Name	RW	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the overlay 6 buffer. The address MUST be word-aligned for proper PXP operation.

PXP Overlay 6 Size

Offset_Address: 0x0000_0390

Bits	Name	RW	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the X-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
23:16	**YBASE**	RW	0x00	This field indicates the Y-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
15:8	**WIDTH**	RW	0x0	Indicates number of horizontal NxN blocks in the image (non-rotated).
7:0	**HEIGHT**	RW	0x0	Indicates the number of vertical NxN blocks in the image (non-rotated).

PXP Overlay 6 Parameters

Offset_Address: 0x0000_03A0

Bits	Name	RW	Reset	Description
31:20	**RSVD1**	RO	0x0000	Reserved, always set to zero.

19:16	**ROP**	RW	0x0	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CNTL field. MASKOL = 0x0 OL AND S0 MASKNOTOL = 0x1 nOL AND S0 MASKOLNOT = 0x2 OL AND nS0 MERGEOL = 0x3 OL OR S0 MERGENOTOL = 0x4 nOL OR S0 MERGEOLNOT = 0x5 OL OR nS0 NOTCOPYOL = 0x6 nOL NOT = 0x7 nS0 NOTMASKOL = 0x8 OL NAND S0 NOTMERGEOL = 0x9 OL NOR S0 XOROL = 0xA OL XOR S0 NOTXOROL = 0xB OL XNOR S0
15:8	**ALPHA**	RW	0x0	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE bits are set. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when enabled in the ALPHA_CNTL field.
7:4	**FORMAT**	RW	0x0	Indicates the input buffer format for overlay 0. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x1 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x3 16-bit pixels with alpha RGB565 = 0x4 16-bit pixels without alpha RGB555 = 0x5 16-bit pixels without alpha
3	**ENABLE_COLORKEY**	RW	0x0	Indicates that colorkey functionality is enabled for this overlay. Pixels found in the overlay colorkey range will be displayed as transparent (the S0 pixel will be used).
2:1	**ALPHA_CNTL**	RW	0x0	Determines how the alpha value is constructed for this overlay. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the OL pixel alpha value will be used to blend the OL with S0. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the overlay and S0 pixels.
0	**ENABLE**	RW	0x0	Indicates that the overlay is active for this operation.

PXP Overlay 6 Parameters2

Offset_Address: 0x0000_03B0

Bits	Name	R W	Reset	Description
31:0	**RSVD**	RO	0x0000	Reserved, always set to zero.

PXP Overlay 7 Buffer Pointer

Offset_Address: 0x0000_03C0

Bits	Name	R W	Reset	Description
31:0	ADDR	RW	0x0	Address pointer for the overlay 7 buffer. The address MUST be word-aligned for proper PXP operation.

PXP Overlay 7 Size

Offset_Address: 0x0000_03D0

Bits	Name	R W	Reset	Description
31:24	**XBASE**	RW	0x00	This field indicates the X-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
23:16	**YBASE**	RW	0x00	This field indicates the Y-coordinate (in blocks) of the top-left NxN block in the overlay within the output frame buffer.
15:8	**WIDTH**	RW	0x0	Indicates number of horizontal NxN blocks in the image (non-rotated).
7:0	**HEIGHT**	RW	0x0	Indicates the number of vertical NxN blocks in the image (non-rotated).

PXP Overlay 7 Parameters

Offset_Address: 0x0000_03E0

Bits	Name	RW	Reset	Description
31:20	**RSVD1**	RO	0x0000	Reserved, always set to zero.

19:16	**ROP**	RW	0x0	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CNTL field. MASKOL = 0x0 OL AND S0 MASKNOTOL = 0x1 nOL AND S0 MASKOLNOT = 0x2 OL AND nS0 MERGEOL = 0x3 OL OR S0 MERGENOTOL = 0x4 nOL OR S0 MERGEOLNOT = 0x5 OL OR nS0 NOTCOPYOL = 0x6 nOL NOT = 0x7 nS0 NOTMASKOL = 0x8 OL NAND S0 NOTMERGEOL = 0x9 OL NOR S0 XOROL = 0xA OL XOR S0 NOTXOROL = 0xB OL XNOR S0
15:8	**ALPHA**	RW	0x0	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE bits are set. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when enabled in the ALPHA_CNTL field.
7:4	**FORMAT**	RW	0x0	Indicates the input buffer format for overlay 0. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x1 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x3 16-bit pixels with alpha RGB565 = 0x4 16-bit pixels without alpha RGB555 = 0x5 16-bit pixels without alpha
3	**ENABLE_COLORKEY**	RW	0x0	Indicates that colorkey functionality is enabled for this overlay. Pixels found in the overlay colorkey range will be displayed as transparent (the S0 pixel will be used).
2:1	**ALPHA_CNTL**	RW	0x0	Determines how the alpha value is constructed for this overlay. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the OL pixel alpha value will be used to blend the OL with S0. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the overlay and S0 pixels.
0	**ENABLE**	RW	0x0	Indicates that the overlay is active for this operation.

PXP Overlay 7 Parameters2

Offset_Address: 0x0000_03F0

Bits	Name	R W	Reset	Description
31:0	**RSVD**	RO	0x0000	Reserved, always set to zero.

33. eCSPI Controller

33.1 eCSPI Controller features

- Full-duplex synchronous serial interface
- Transfer continuation function allows unlimited length data transfers
- 32-bit wide by 64-entry FIFO for both transmit and receive data
- 32-bit wide by 16-entry FIFO for HT message data
- Polarity and phase of the Chip Select(SS) and SPI Clock (SCLK) are configurable
- DMA support

33.2 eCSPI Register Descriptions

SSI_BASE: 0x5020_0000

33.2.1CONTROLEG: Control Register

Address: SSI_BASE + 0x08

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

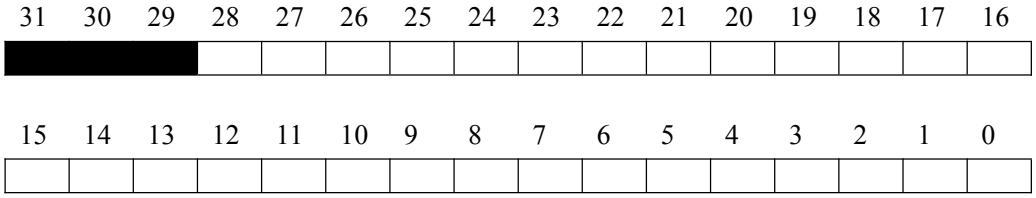
Bit	Description
31-20 BURST LENGTH	This field defines number of bits to be transferred in one SPI burst. SS will remain asserted until all bits in a SPI burst are shifted out .A maximum of 2^12 bits Canberra transferred in a single SPI burst . In master mode, it control the number of bits per SPI burst .since the shift register always loads 32-bit data from TXFIFO ,only the n least-significant (n = BURST LENGTH[4:0] + 1) bits in the first word will be shifted out. The remaining bits in first word will be ignored. All bits in other words will be shifted out. In slave mode, only when SSCTL is cleared, this field will take effect in SPI transfer. Number of valid bits in a SPI burst. 0x000 A SPI burst contains least 1 bit in a TXFIFO word. 0x001 A SPI burst contains least 2 bit in a TXFIFO word. 0x002 A SPI burst contains least 3 bit in a TXFIFO word. ..... 0x01F A SPI burst contains all 32 bits in a TXFIFO word. 0x020 A SPI burst contains least 1 bit in first TXFIFO word and all 32 bits

	in second TXFIFO word. 0x021 A SPI burst contains least 2 bits in first TXFIFO word and all 32 bits in second TXFIFO word. ..... 0xFFE A SPI burst contains least 31 bits in first TXFIFO word and 2^7 -1 TXFIFO words. 0xFFF A SPI burst contains 2^7 TXFIFO words
19-18 CHANNEL SELECT	SPI CHANNEL SELECT bits. It selects which SPI channel is selected to be active. 00 SPI Channel 0 is selected. 01 SPI Channel 1 is selected. 10 SPI Channel 2 is selected. 11 SPI Channel 3 is selected.
17-16 DRCTL	SPI DATA Ready Control. This 2-bit field selects the utilization of the SPI_RDY check this fields before it start a SPI burst. 00 Don't care SPI_RDY 01 Burst will be triggered by falling edge of SPI_RDY. 10 Burst will be triggered by low level of SPI_RDY. 11 RSV.
15–12 PRE DIVIDER	SPI Pre Divider bits. eCSPI uses a two stages divider structure to generate the SPI clock. This four-bit field defines the ipg_clk 4-bit pre-divider. 0000Divide by 1. 0001Divide by 2. 0010Divide by 3. ..... 1101Divide by 14. 1110Divide by 15. 1111Divide by 16.
11 –8 POST DIVIDER	SPI Post Divider bits. eCSPI uses a two stages divider structure to generate the SPI clock. This four-bit field defines the 4-bit post-divider. 0000Divide by 1. 0001Divide by 2. 0010Divide by 4. ..... 1110Divide by 2^14. 1111Divide by 2^15.
Bit	**Description**
7–4 CHANNEL MODE	SPI CHANNEL MODE defines mode of each SPI channel. CHANNEL MODE [0] defines the mode of SPI channel 0. CHANNEL MODE [1] defines the mode of SPI channel 1. CHANNEL MODE [2] defines the mode of SPI channel 2. CHANNEL MODE [3] defines the mode of SPI channel 3. 0 Slave Mode. 1 Master Mode.
3 SMC	Start Mode Control. This bit is used in master mode only and it controls how eCSPI start a SPI burst. 0 Normal mode, the XCH bit controls when a SPI burst can start. Write a 1 to XCH bit will start a SPI burst or multiple bursts. (controlled by SSCTL) 1 Automatic Mode, start a SPI burst when a data is written in TXFIFO immediately.
2	SPI Exchange Bit. If the SMC bit is cleared, writing a 1 to this bit starts

XCH	one SPI bursts/multiple SPI bursts according to SSCTL bit. This bit remains set while either the exchange is in progress, or the eCSPI is waiting for active input if SPIRDY is enabled through DRCTL. This bit is cleared automatically when one SPI burst has been done (SSCTL is clear) or TXFIFO is empty (SSCTL is set). In Slave mode, this bit is ignored. 0 Idle. 1 Initiates exchange (write) or busy (read).
1 HW	HW Trigger Enable. This bit is used in master mode only and it enables hardware trigger mode. 0 HT Mode is disabled. 1 HT Mode is enabled.
0 EN	SPI Module Enable Control. This bit enables the eCSPI. This bit must be asserted before writing to other Register or initiating an exchange. Writing zero to this bit disables the module and resets the internal logic with the exception of the CONTROLREG. The module's internal clocks are gated off whenever the module is disabled. 1 eCSPI is enabled. 0 eCSPI is disabled.

33.2.2CONFIGREG: Config Register

Address: SSI_BASE + 0x0C



Bit	Description
31–29	Reserved, all bits can be read/write, but have no meaning.
28–24 HTLENGTH	The HT LENGTH defines the length of message content in HT Mode. The number of bits of one message content is HT LENGTH + 1;
23–20 SCLK CTL	The SCLK CTL control inactive state of SCLK line for each SPI channel. Bit 27 controls the SCLK inactive state of SPI channel 3. Bit 26 controls the SCLK inactive state of SPI channel 2. Bit 25 controls the SCLK inactive state of SPI channel 1. Bit 24 controls the SCLK inactive state of SPI channel 0. 0 stay low. 1 stay high.
19–16 DATA CTL	The Data CTL control inactive state of Data line for each SPI channel. Bit 23 controls the data polarity of SPI channel 3. Bit 22 controls the data polarity of SPI channel 2. Bit 21 controls the data polarity of SPI channel 1. Bit 20 controls the data polarity of SPI channel 0. 0 stay high. 1 stay low.
15–12 SSB POL	The SSB POL control polarity of SSB for each SPI channel. Bit 15 controls the SSB Polarity of SPI channel 3. Bit 14 controls the SSB Polarity of SPI channel 2.

	Bit 13 controls the SSB Polarity of SPI channel 1. Bit 12 controls the SSB Polarity of SPI channel 0. 0 Active Low. 1 Active High.
11–8 SSB CTRL	The SSB CTL control behavior of SSB for each SPI channel. Bit 11 controls the SSB behavior of SPI channel 3. Bit 10 controls the SSB behavior of SPI channel 2. Bit 9 controls the SSB behavior of SPI channel 1. Bit 8 controls the SSB behavior of SPI channel 0. In Master mode, this bit defines if it is in multiple burst mode when SMC bit is cleared. When SMC bit is set, this bit will be ignored. 0 Single burst Mode. 1 Multiple bursts Mode. In Slave mode, this bit controls when a SPI burst is completed. 0 SPI burst completed when (BURST_LENGTH + 1) bits are received. 1 SPI burst completed when SSB input negated.
7–4 SCLK POL	The SCLK POL control polarity of SCLK for each SPI channel. Bit 7 controls the SCLK Polarity of SPI channel 3. Bit 6 controls the SCLK Polarity of SPI channel 2. Bit 5 controls the SCLK Polarity of SPI channel 1. Bit 4 controls the SCLK Polarity of SPI channel 0. 0 Active high polarity (0 = Idle). 1 Active low polarity (1 = Idle).
3–0 SCLK PHA	The SCLK PHA control data phase of SCLK for each SPI channel. Bit 3 controls the SSB Polarity of SPI channel 3. Bit 2 controls the SSB Polarity of SPI channel 2. Bit 1 controls the SSB Polarity of SPI channel 1. Bit 0 controls the SSB Polarity of SPI channel 0. 0 Phase 0 operation. 1 Phase 1 operation.

33.2.3INTREG: Interrupt Control Register

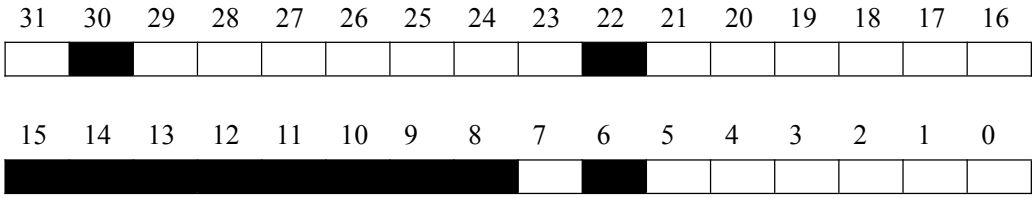
Address: SSI_BASE + 0x10

Bit	Description
31-8	Reserved, all bits should read zero.
7 TCEN	Transfer Completed Interrupt enable. This bit enables the Transfer Completed Interrupt. 0 Disable 1 Enable
6 ROEN	RXFIFO Overflow Interrupt enable. The bit enables the RXFIFO Overflow Interrupt. 0 Disable 1 Enable
5 RFEN	RXFIFO Full Interrupt enable. The bit enables the RXFIFO Full Interrupt. 0 Disable 1 Enable

4 RDREN	RXFIFO Data Request Interrupt enable. The bit enables the RXFIFO Data Request Interrupt when number of data in RXFIFO is great than RX WATER MARK when RXTDEN is clear. 0 Disable 1 Enable
3 RREN	RXFIFO Ready Interrupt enable. The bit enables the RXFIFO Ready Interrupt. 0 Disable 1 Enable
2 TFEN	TXFIFO Full Interrupt enable. The bit enables the TXFIFO Full Interrupt. 0 Disable 1 Enable
1 TDREN	TXFIFO Data Request Interrupt enable. The bit enables the TXFIFO Data Request Interrupt when number of data in TXFIFO is no more than TX WATER MARK. 0 Disable 1 Enable
0 TEEN	TXFIFO Empty Interrupt enable. The bit enables the TXFIFO Empty Interrupt. 0 Disable 1 Enable

33.2.4DMAREG: DMA Control Register

Address: SSI_BASE + 0x14

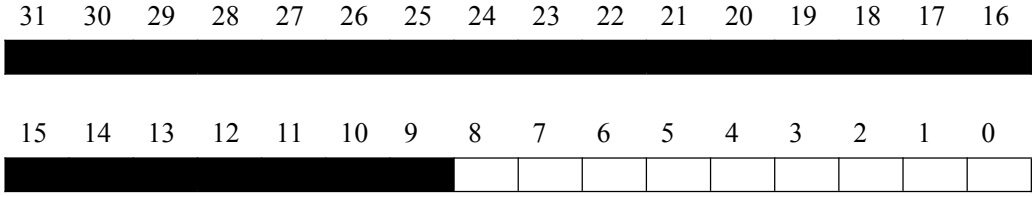


Bit	Description
31 RXTDEN	RXFIFO TAIL DMA Request Enable. When this bit is set, an internal counter is enabled and is increased at each read of RXFIFO. The counter will be cleared automatically when it reach RX DMA LENGTH. If number of words remained in RXFIFO is great or equal to RX DMA LENGTH - content of the counter, a DMA request will be generated even if it is less than RX WATER MARK. 0 Disable 1 Enable
30	Reserved, all bits should read zero.
29–24 RX DMA LENGTH	RX DMA LENGTH. These bits define how many words will be retrieved in one DMA access cycle. (Only used when RXTDEN is set.)
23 RXDEN	RXFIFO DMA Request Enable. This bit enables/disables the RXFIFO DMA Request. 0 Disable 1 Enable
22	Reserved. Should be cleared.
21-16	RX WATER MARK. These bits control when eCSPI issue a RX DMA

RXWATER MARK	request. A RX DMA/Interrupt request will be issued when number of data in RXFIFO is great than RX WATER MARK.
15-8	Reserved, should be cleared.
7 TXDEN	TXFIFO DMA Request Enable. This bit enables/disables the TXFIFO DMA Request. 0 Disable 1 Enable
6	Reserved, should be cleared.
5–0 TXWATER MARK	TX WATER MARK. These bits control when eCSPI issue a TX DMA request. A DMA/Interrupt request will be issued when number of data in TXFIFO is no more than TX WATER MARK.

33.2.5STATREG: Status Register

Address: SSI_BASE + 0x18

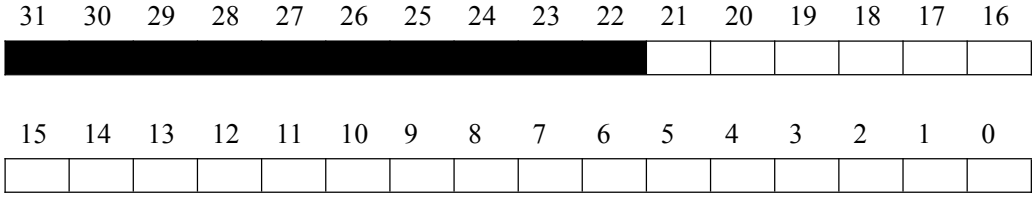


Bit	Description
31–8	Reserved, should be cleared.
8	Rxfifo enable
7 TC	Transfer Completed. When set, this bit indicates current transmission is completed. Writing 1 to this bit clears it. 0 Busy. 1 Transfer Completed.
6 RO	RXFIFO Overflow. When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it. 0 RXFIFO is available. 1 RXFIFO has overflowed.
5 RF	RXFIFO Full. This bit is set when the RXFIFO is full (64 words). 0 Not Full. 1 RXFIFO is Full. (64 words)
4 RDR	RXFIFO Data Request. When RXTDE is set, 0 Number of data in RXFIFO is less than RX DMA WATER MARK. 1 Number of data in RXFIFO is great than RX DMA WATER MARK or a DMA TAIL DMA condition is match. When RXTDE is clear, 0 Number of data in RXFIFO is less than RX DMA WATER MARK. 1 Number of data in RXFIFO is great than RX DMA WATER MARK.
3 RR	RXFIFO Ready. This bit is set any time there is one or more words stored in RXFIFO (≥ 1 words). 0 No valid data in RXFIFO. 1 At least 1 word in RXFIFO.
2 TF	TXFIFO Full. This bit is set when if the TXFIFO is full (64 words). 0 TXFIFO is not Full. 1 TXFIFO is Full.
1	TXFIFO Data Request.

TDR	0 Number of empty slots in TXFIFO is great than TX DMA WATER MARK. 1 Number of empty slots in TXFIFO is no more than TX DMA WATER MARK.
0 TE	TXFIFO Empty. This bit is set if the TXFIFO is empty. 0 TXFIFO contains one or more words. 1 TXFIFO is empty.

33.2.6PERIODREG: Sample Period Control Register

Address: SSI_BASE + 0x1C



Bit	Description
31-22	Reserved, all bits should read zero.
21-16 CSDCTRL	Chip Select Delay Control bits. These bits define how many SPI clocks will be inserted between chip select active edge and first SPI clock edge.
15 CSRC	Clock Source Control. This bit selects the clock source for the sample period counter. 0 SPI Clock (SCLK) 1 CKIL (32.768 KHz)
14–0 SAMPLE PERIOD	Sample Period Control. These bits define the number of wait states to be inserted between data transfers.

33.2.7TESTREG: Test Control Register

Address: SSI_BASE + 0x20

Bit	Description
31 LBC	Loop Back Control. This bit is used in Master mode only. When this bit is set, the eCSPI module connects the transmitter and receiver sections internally, and the data shifted out from the most-significant bit of the Shift register is looped back into the least-significant bit of the Shift register. In this way, a self-test of the complete transmit/receive path can be made. The output pins are not affected, and the input pins are ignored. 0 Not connected. 1 Internally connected.
30	Reserved, all bits should read zero.
29–28 CL	Catch Latency bits. These bits define the catch latency mode which will give more time on path delay between chip and external Slave devices. 00 Normal mode. 01 The data latch event will be delayed for half SPI clock cycle.

	10 The data latch event will be delayed for one SPI clock cycle. 11 The data latch event will be delayed for one and half SPI clock cycle.
27–15	Reserved, all bits should read zero.
14–8 RXCNT	RXFIFO Counter. These bits indicate the number of words in RXFIFO.
7	Reserved, all bits should read zero.
6–0 TXCNT	TXFIFO Counter. These bits indicate the number of words in TXFIFO.

33.2.8MSGDATA: Message Data Register

Address: SSI_BASE + 0x24

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

Bit	Description
31–0 MSGDATA	MSGDATA holds the top word of MSG Data FIFO. The MSG Data FIFO is advanced for each write of this register. The data read is zero. The data write to this register will be stored into MSG Data FIFO.

33.2.9RXDATA: RXFIFO DATA Register

Address: SSI_BASE + 0x50

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

Bit	Description
31-0	RXDATA holds the top word of RXFIFO .The RXFIFO is advanced for each read of this register. The data read is zero when the Receive Data Ready (RR) bit in the Interrupt Control/Status register is cleared. The data write to this register will be ignored.

33.2.10TXDATA : TXFIFO DATA Register

Address: SSI_BASE + 0x460

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

Bit	Description

31-0	TXDATA holds the top word of TXFIFO. The TXFIFO is advanced for each write of this register. The data read is zero. The data write to this register will be stored into TXFIFO.

34. RTC

34.1 Features

Seconds counter.
Alarm interrupts in the normal mode.
Periodic interrupts.
Wakeup function.

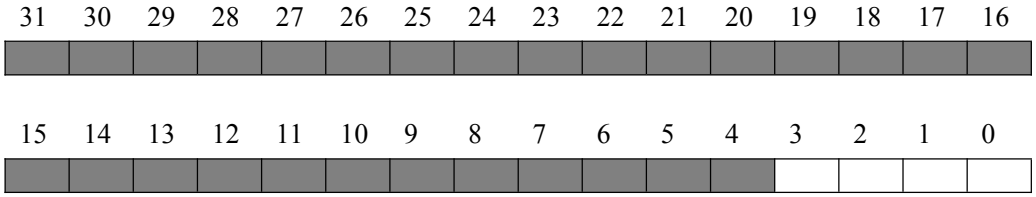
34.2 RTC Register Descriptions

RTC_BASE: 0x51000000

34.2.1 RTCCTL

The RTCCTL register can reset the RTC.

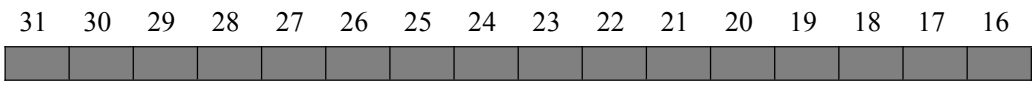
Offset_Address: 0x0000_0000

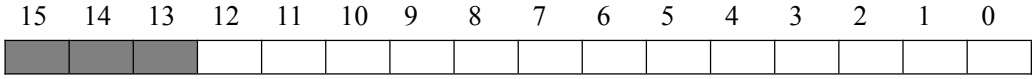


Bit	Type	Reset	Description
31-4	R	0	Reserved
3	R/W	0	almwen almdat register write enable
2	R/W	0	pinten period interrupt enable.
1	R/W	0	ainten alarm interrupt enable.
0	R/W	0	rtcrst

34.2.2 ANAWEN

Offset_Address: 0x0000_0004

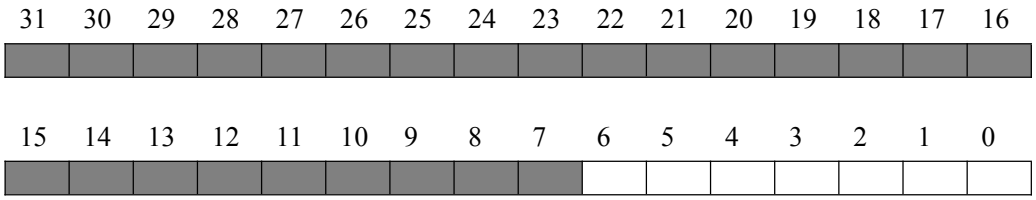




Bit	Type	Reset	Description
31-0	R	0	Reserved
12	R/W	0	rtccnt_wen RTC counter register write enable
11	R/W	0	para_wen ram[127:96](software) write enable
10	R/W	0	ramh_wen ram[95:64](delay_on_timer[31:0]) write enable
9	R/W	0	gpio_wen ram[63:54](RTC gpio) write enable
8	R/W	0	raml_wen ram[53:32](PMC_H) write enable
7	R/W	0	pmc_wen ram[31:0] (PMC_L) write enable
6	R/W	0	isof_wen isof_s write enable
5	R/W	0	slp_wen sleep_s write enable
4	R/W	0	iso_wen iso write enable
3	R/W	0	wdg_wen watchdog_en write enable
2	R/W	0	cksel_wen clk32k_sel write enable
1	R/W	0	donen_wen delay_on_en write enable
0	R/W	0	foff_wen force_off write enable

34.2.3 ANACTL

Offset_Address: 0x0000_0008



Bit	Type	Reset	Description
31-7	R	0	Reserved
6	R/W	0	isof_s ANACTL write operation will set isof_s in analog block when isof_wen=35.
5	R/W	0	sleep_s ANACTL write operation will set sleep_s in analog block when slp_wen=35.
4	R/W	0	iso ANACTL write operation will set iso in analog block when iso_wen=35.
3	R/W	0	wdgen ANACTL write operation will set watchdog_en in analog block when wdg_wen=35.

2	R/W	0	cksel ANACTL write operation will set clk32k_sel in analog block when cksel_wen=35.
1	R/W	0	donen ANACTL write operation will set delay_on_enable in analog block when donen_wen=35.
0	R/W	0	foffen ANACTL write operation will set force_off_enable in analog block when wdg_wen=35.

34.2.4 RTCIM

Offset_Address:0x0000_000C

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

Bit	Type	Reset	Description
30-7	R	0	Reserved.
6-2	R/W	0	period interrupt control 5'b00000: 1/1024s 5'b00001: 1/512s 5'b00010: 1/256s 5'b00011: 1/128s 5'b00100: 1/64s 5'b00101: 1/32s 5'b00110: 1/16s 5'b00111: 1/8s 5'b01000: 1/4s 5'b01001: 1/2s 5'b01010: 1s 5'b01011: 2s 5'b01100: 4s 5'b01101: 8s 5'b01110: 16s 5'b01111: 32s 5'b10000: 64s 5'b10001: 128s 5'b10010: 256s 5'b10011: 512s 5'b10100: 1024s
1	R/W	0	prdim period interrupt mask
0	R/W	0	almim alarm interrupt mask.

34.2.5 RTCSTA

Offset_Address: 0x0000_0010

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

Bit	Type	Reset	Description
31-20	R	0	RTC_GPIO signals from analog block [31] rtc_gpio1_E2 [30] rtc_gpio1_E1 [29] rtc_gpio1_ie [28] rtc_gpio1_oe [27] rtc_gpio1_out [26] rtc_gpio0_E2 [25] rtc_gpio0_E1 [24] rtc_gpio0_ie [23] rtc_gpio0_oe [22] rtc_gpio0_out [21] rtc_gpio1_in [20] rtc_gpio0_in
19-3	R	0	Signals from analog block [19] ACC_IN_V09 [18] CAN1_IN_V09 [17] CAN0_IN_V09 [16] GPIO1_IN_V09 [15] GPIO0_IN_V09 [14] RTC_ACT_V09 [13] ACC_ACT_V09 [12] CAN1_ACT_V09 [11] CAN0_ACT_V09 [10] GPIO1_ACT_V09 [9] GPIO0_ACT_V09 [8] OSC24M_EN_V0P9 [7] OSC32K_SEL_V0P9 [6] ISO_FORCE_V09 [5] ISO_CTL_V09 [4] ISO_ENB_V09 [3] SLEEP_MODE_V09
2	R	0	Busy Busy =1 when writing analog block.
1	R/W	0	pint period interrupt
0	R/W	0	aint alarm interrupt

34.2.6 ALMDAT

Offset_Address: 0x0000_0014

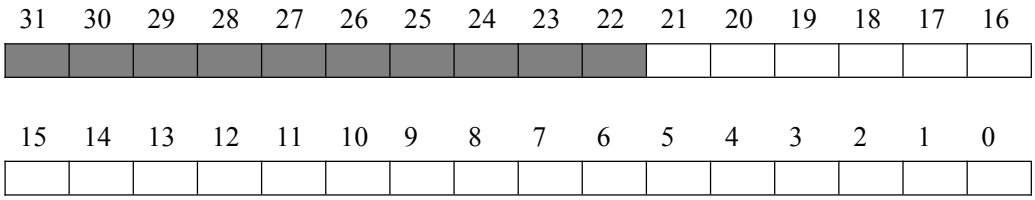
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bit	Type	Reset	Description
31-0	R/W	55AA 55AA	ALMDAT alarm data register can be set when almwen=35.

34.2.7 RAML

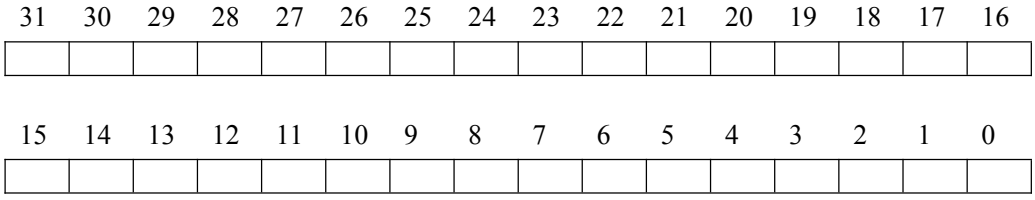
Offset_Address: 0x0000_0018



Bit	Type	Reset	Description
30-22	R	0	Reserved.
21-0	R/W	0	ram[53:32] (PMC_H) write operation will set ram[53:32] in analog block when raml_wen=35.

34.2.8 RAMH

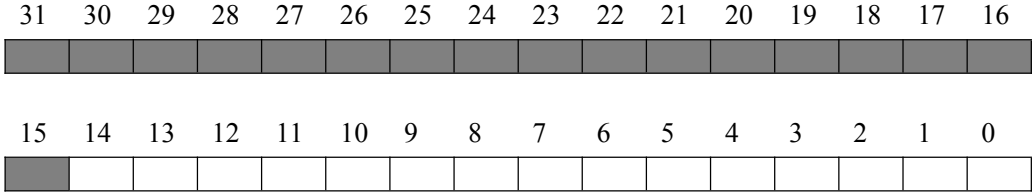
Offset_Address:0x0000_001C



Bit	Type	Reset	Description
31-0	R/W	0	ram[95:64] write operation will set ram[95:64] in analog block when ramh_wen=35.

34.2.9 RTC_VAL_L

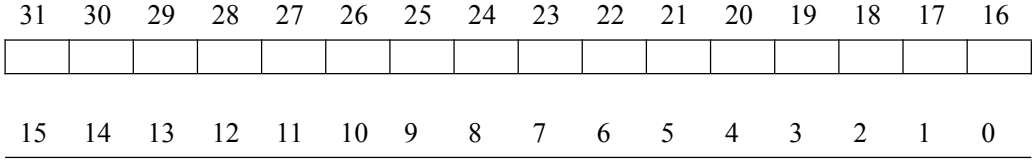
Offset_Address: 0x0000_0020



Bit	Type	Reset	Description
31-15	R	0	Reserved.
14-0	R	0	rtc_value[14:0] from analog block.

34.2.10 RTC_VAL_H

Offset_Address: 0x0000_0024



Bit	Type	Reset	Description
31-0	R/W	0	rtc_value[46:15] from analog block. write operation will set rtc_value[46:15] in analog block when rtccnt_wen=35.

34.2.11 RTC_GPIO

Offset_Address: 0x0000_0028

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

Bit	Type	Reset	Description
31-22	R/W	0	RTC_GPIO config write operation will set rtc_gpio in analog block when gpio_wen=35. [31] rtc_gpio1_E2, [30] rtc_gpio1_E1, [29] rtc_gpio1_ie, [28] rtc_gpio1_oe, [27] rtc_gpio1_out, [26] rtc_gpio0_E2, [25] rtc_gpio0_E1, [24] rtc_gpio0_ie, [23] rtc_gpio0_oe, [22] rtc_gpio0_out
21-0	R	0	reserved

34.2.12 PMCL

Offset_Address: 0x0000_002C

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

Bit	Type	Reset	Description
31-0	R/W	0	PMC_L write operation will set PMC_L(ram[31:0]) in analog block when pmc_wen=35.

34.2.13 RAM_PARA

Offset_Address: 0x0000_0030

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

Bit	Type	Reset	Description
31-0	R/W	0	RAM_PARA write operation will set PMC_PARA(ram[127:96]) in analog block when para_wen=35.

35. ADC

35.1 Features

3x SAR ADC, 12bit resolution,1M sample rate.
8-CH multiplexed with per ADC.
Built-in comparer to generate interrupts.

35.2 ADC Register Descriptions

ADC_BASE: 0x51100000

35.2.1 ADC_CTRL

Offset_Address: 0x0000_0000

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

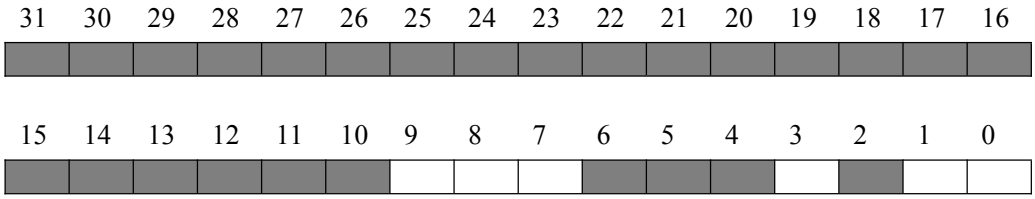
Bit	Type	Reset	Description
31	R/W	0	**ADC_AUX7_CMD** 0: aux7 force transform disable. 1: aux7 force transform enable. We can use force transform enable when some part of the module is damaged.
30	R/W	0	**ADC_AUX6_CMD** 0: aux6 force transform disable. 1: aux6 force transform enable.
29	R/W	0	**ADC_AUX5_CMD** 0: aux5 force transform disable. 1: aux5 force transform enable.
28	R/W	0	**ADC_AUX4_CMD** 0: aux4 force transform disable. 1: aux4 force transform enable.
27	R/W	0	**ADC_AUX7_DET_INV** 0: aux7_det low valid. 1: aux7_det high valid.

26	R/W	0	**ADC_AUX6_DET_INV** 0: aux6_det low valid. 1: aux6_det high valid.
25	R/W	0	**ADC_AUX5_DET_INV** 0: aux5_det low valid. 1: aux5_det high valid.
24	R/W	0	**ADC_AUX4_DET_INV** 0: aux4_det low valid. 1: aux4_det high valid.
23	R/W	0	**ADC_AUX7_ENA** 0: aux7 disable. 1: aux7 enable.
22	R/W	0	**ADC_AUX6_ENA** 0: aux6 disable. 1: aux6 enable.
21	R/W	0	**ADC_AUX5_ENA** 0: aux5 disable. 1: aux5 enable.
20	R/W	0	**ADC_AUX4_ENA** 0: aux4 disable. 1: aux4 enable.
19-18	R/W	0	Reserved.
17	R/W	0	**SOF** 0: hardware mode. 1: software mode.
16	R/W	0	**ADC_AUX3_CMD** 0: aux3 force transform disable. 1: aux3 force transform enable.
15	R/W	0	**ADC_AUX2_CMD** 0: aux2 force transform disable. 1: aux2 force transform enable.
14	R/W	0	**ADC_AUX1_CMD** 0: aux1 force transform disable. 1: aux1 force transform enable.
13	R/W	0	**ADC_AUX0_CMD** 0: aux0 force transform disable. 1: aux0 force transform enable.
12	R/W	0	Reserved.
11	R/W	0	**ADC_AUX3_DET_INV** 0: aux3_det low valid. 1: aux3_det high valid.
10	R/W	0	**ADC_AUX2_DET_INV** 0: aux2_det low valid. 1: aux2_det high valid.
9	R/W	0	**ADC_AUX1_DET_INV** 0: aux1_det low valid. 1: aux1_det high valid.
8	R/W	0	**ADC_AUX0_DET_INV** 0: aux0_det low valid. 1: aux0_det high valid.
7	R/W	0	Reserved.
6	R/W	0	**ADC_AUX3_ENA** 0: aux3 disable. 1: aux3 enable.
5	R/W	0	**ADC_AUX2_ENA** 0: aux2 disable. 1: aux2 enable.
4	R/W	0	**ADC_AUX1_ENA** 0: aux1 disable. 1: aux1 enable.
3	R/W	0	**ADC_AUX0_ENA** 0: aux0 disable. 1: aux0 enable.

2-1	R/W	0	Reserved.
0	R/W	0	**ADC_RST** ADC soft rst : 0 : normal. 1 : reset.

35.2.2 ADC_CFG

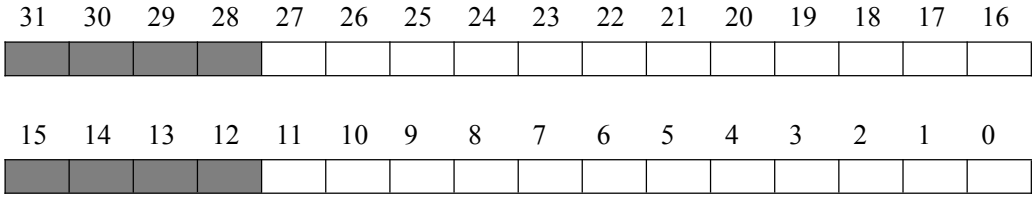
Offset_Address: 0x0000_0004



Bit	Type	Reset	Description
31-10	R	0	Reserved.
9-7	R/W	110	AUX_ADR aux address
6-4	R/W	010	Reserved.
3	R/W	0	ADC_MOD
2	R/W	0	Reserved.
1-0	R/W	10	ADC_PD To analog block.

35.2.3 ADC_IMR

Offset_Address: 0x0000_008



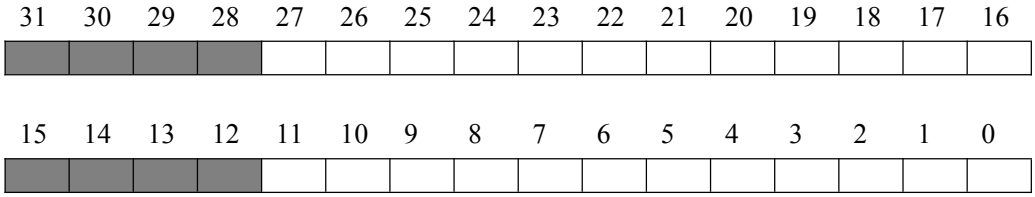
Bit	Type	Reset	Description
31-28	R	0	Reserved.
27	R/W	1	ADC_AUX7_VALUE_IM ADC aux7 value interrupt mask: 0: mask disabled. 1: mask enabled.
26	R/W	1	ADC_AUX7_STOP_IM ADC aux7 stop interrupt mask: 0: mask disabled. 1: mask enabled.
25	R/W	1	ADC_AUX7_START_IM ADC aux7 start interrupt mask: 0: mask disabled. 1: mask enabled.
24	R/W	1	ADC_AUX6_VALUE_IM

			ADC aux6 value interrupt mask: 0: mask disabled. 1: mask enabled.
23	R/W	1	ADC_AUX6_STOP_IM ADC aux6 stop interrupt mask: 0: mask disabled. 1: mask enabled.
22	R/W	1	ADC_AUX6_START_IM ADC aux6 start interrupt mask: 0: mask disabled. 1: mask enabled.
21	R/W	1	ADC_AUX5_VALUE_IM ADC aux5 value interrupt mask: 0: mask disabled. 1: mask enabled.
20	R/W	1	ADC_AUX5_STOP_IM ADC aux5 stop interrupt mask: 0: mask disabled. 1: mask enabled.
19	R/W	1	ADC_AUX5_START_IM ADC aux5 start interrupt mask: 0: mask disabled. 1: mask enabled.
18	R/W	1	ADC_AUX4_VALUE_IM ADC aux4 value interrupt mask: 0: mask disabled. 1: mask enabled.
17	R/W	1	ADC_AUX4_STOP_IM ADC aux4 stop interrupt mask: 0: mask disabled. 1: mask enabled.
16	R/W	1	ADC_AUX4_START_IM ADC aux4 start interrupt mask: 0: mask disabled. 1: mask enabled.
15-12	R/W	1	Reserved.
11	R/W	1	ADC_AUX3_VALUE_IM ADC aux3 value interrupt mask: 0: mask disabled. 1: mask enabled.
10	R/W	1	ADC_AUX3_STOP_IM ADC aux3 stop interrupt mask: 0: mask disabled. 1: mask enabled.
9	R/W	1	ADC_AUX3_START_IM ADC aux3 start interrupt mask: 0: mask disabled. 1: mask enabled.
8	R/W	1	ADC_AUX2_VALUE_IM ADC aux2 value interrupt mask: 0: mask disabled. 1: mask enabled.
7	R/W	1	ADC_AUX2_STOP_IM ADC aux2 stop interrupt mask: 0: mask disabled. 1: mask enabled.
6	R/W	1	ADC_AUX2_START_IM ADC aux2 start interrupt mask: 0: mask disabled. 1: mask enabled.
5	R/W	1	ADC_AUX1_VALUE_IM ADC aux1 value interrupt mask: 0: mask disabled. 1: mask enabled.

4	R/W	1	ADC_AUX1_STOP_IM ADC aux1 stop interrupt mask: 0: mask disabled. 1: mask enabled.
3	R/W	1	ADC_AUX1_START_IM ADC aux1 start interrupt mask: 0: mask disabled. 1: mask enabled.
2	R/W	1	ADC_AUX0_VALUE_IM ADC aux0 value interrupt mask: 0: mask disabled. 1: mask enabled.
1	R/W	1	ADC_AUX0_STOP_IM ADC aux0 stop interrupt mask: 0: mask disabled. 1: mask enabled.
0	R/W	1	ADC_AUX0_START_IM ADC aux0 start interrupt mask: 0: mask disabled. 1: mask enabled.

35.2.4 ADC_STA

Offset_Address:0x0000_000C

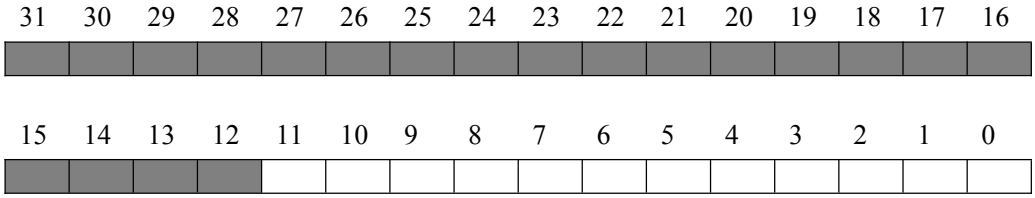


Bit	Type	Reset	Description
30-28	R	0	Reserved.
27	R/W	0	ADC_AUX7_VALUE_INT Aux7 value interrupt status.
26	R/W	0	ADC_AUX7_STOP_INT Aux7 stop interrupt status.
25	R/W	0	ADC_AUX7_START_INT Aux7 start interrupt status.
24	R/W	0	ADC_AUX6_VALUE_INT Aux6 value interrupt status.
23	R/W	0	ADC_AUX6_STOP_INT Aux6 stop interrupt status.
22	R/W	0	ADC_AUX6_START_INT Aux6 start interrupt status.
21	R/W	0	ADC_AUX5_VALUE_INT Aux5 value interrupt status.
20	R/W	0	ADC_AUX5_STOP_INT Aux5 stop interrupt status.
19	R/W	0	ADC_AUX5_START_INT Aux5 start interrupt status.
18	R/W	0	ADC_AUX4_VALUE_INT Aux4 value interrupt status.
17	R/W	0	ADC_AUX4_STOP_INT Aux4 stop interrupt status.
16	R/W	0	ADC_AUX4_START_INT Aux4 start interrupt status.
15-12	R	0	Reserved.
11	R/W	0	ADC_AUX3_VALUE_INT Aux3 value interrupt status.

10	R/W	0	ADC_AUX3_STOP_INT Aux3 stop interrupt status.
9	R/W	0	ADC_AUX3_START_INT Aux3 start interrupt status.
8	R/W	0	ADC_AUX2_VALUE_INT Aux2 value interrupt status.
7	R/W	0	ADC_AUX2_STOP_INT Aux2 stop interrupt status.
6	R/W	0	ADC_AUX2_START_INT Aux2 start interrupt status.
5	R/W	0	ADC_AUX1_VALUE_INT Aux1 value interrupt status.
4	R/W	0	ADC_AUX1_STOP_INT Aux1 stop interrupt status.
3	R/W	0	ADC_AUX1_START_INT Aux1 start interrupt status.
2	R/W	0	ADC_AUX0_VALUE_INT Aux0 value interrupt status.
1	R/W	0	ADC_AUX0_STOP_INT Aux0 stop interrupt status.
0	R/W	0	ADC_AUX0_START_INT Aux0 start interrupt status.

35.2.5 ADC_AUX0

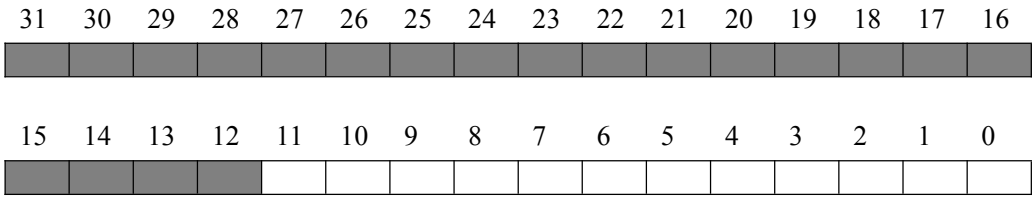
Offset_Address: 0x0000_0014



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC_AUX0** ADC AUX0 value.

35.2.6 ADC_AUX1

Offset_Address: 0x0000_0018

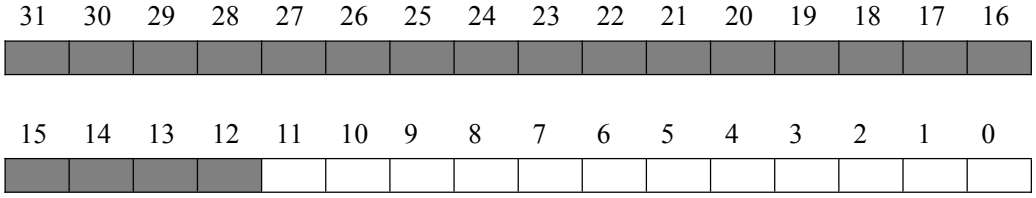


Bit	Type	Reset	Description

31-12	R	0	Reserved..
11-0	R	0	**ADC_AUX1** ADC AUX1 value.

35.2.7 ADC_AUX2

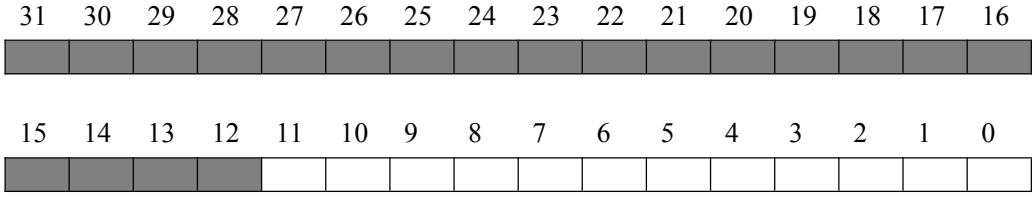
Offset_Address: 0x0000_001C



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC_AUX2** ADC AUX1 value.

35.2.8 ADC_AUX3

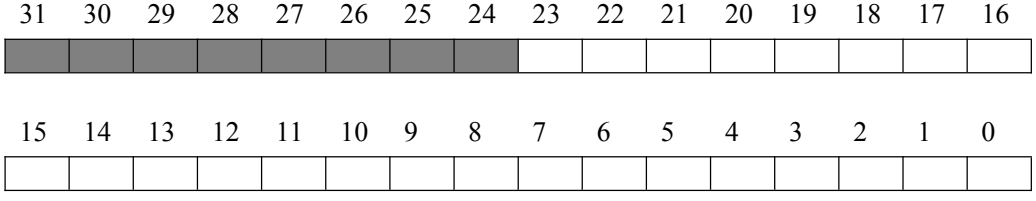
Offset_Address: 0x0000_0020



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC_AUX3** ADC AUX1 value.

35.2.9 ADC_DBNCNT

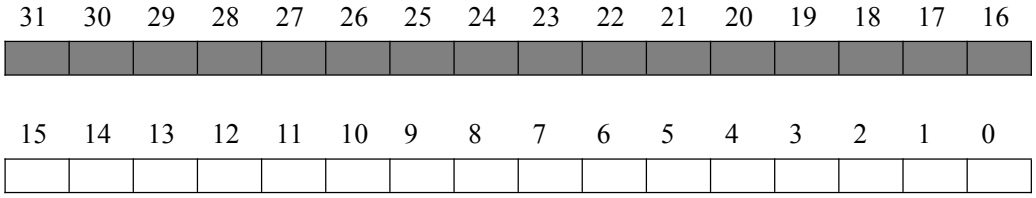
Offset_Address: 0x0000_002C



Bit	Type	Reset	Description
31-24	R	0	Reserved..
23-0	R/W	24'hffffff	**ADC_DBNCNT** Debounce counter of APB Clock for eliminating bounce . After the time of **DBNCNT,** we'll think the value of the signal is stable and valid. Note: we must set the value of **DBNCNT** before enable a channel.

35.2.10 ADC_INTER

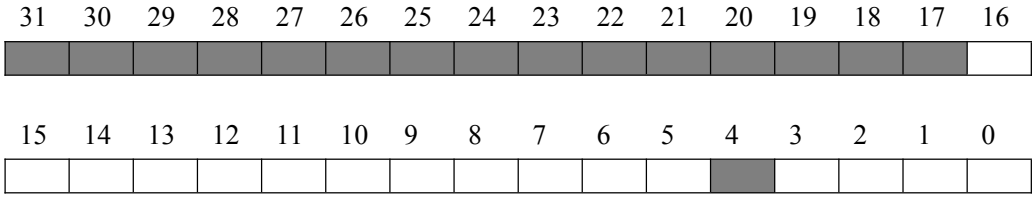
Offset_Address: 0x0000_0030



Bit	Type	Reset	Description
31-16	R	0	Reserved..
15-0	R/W	16'hffff	**ADC_INTER** ADC transform interval, ie. The counter of ADC Sample clock..The time between two transforms

35.2.11 ADC_SCTL

Offset_Address: 0x0000_0034

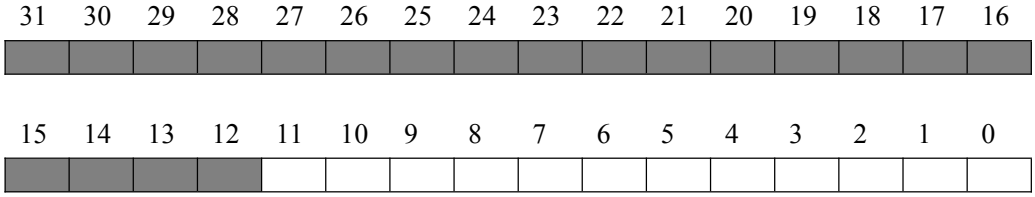


Bit	Type	Reset	Description
31-17	R	0	reserved.
16	R	0	ADC aux7_req_s
15	R	0	ADC aux6_req_s
14	R	0	ADC aux5_req_s
13	R	0	ADC aux4_req_s
12-10	W/R	0	aux_sel_s[2:0]
9	W/R	0	adc_csn_s
8	W/R	0	adc_si_s
7	W/R	0	adc_clk_s

6	R	0	adc_so_s
5	R	0	adc_busy_s
4	R	0	reserved.
3	R	0	ADC aux3_req_s
2	R	0	ADC aux2_req_s
1	R	0	ADC aux1_req_s
0	R	0	ADC aux0_req_s

35.2.12 ADC_AUX4

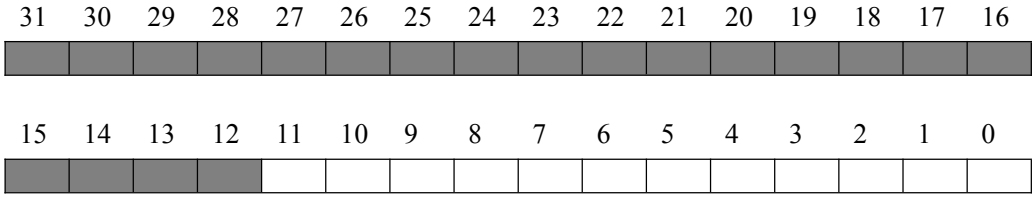
Offset_Address: 0x0000_0038



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC_AUX4** ADC AUX4 value.

35.2.13 ADC_AUX5

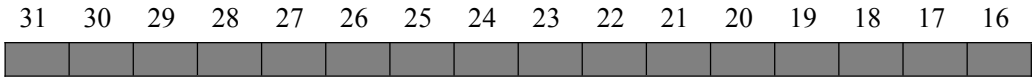
Offset_Address: 0x0000_003C

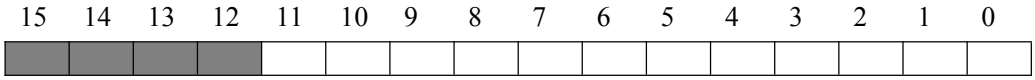


Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC_AUX5** ADC AUX5 value.

35.2.14 ADC_AUX6

Offset_Address: 0x0000_0040

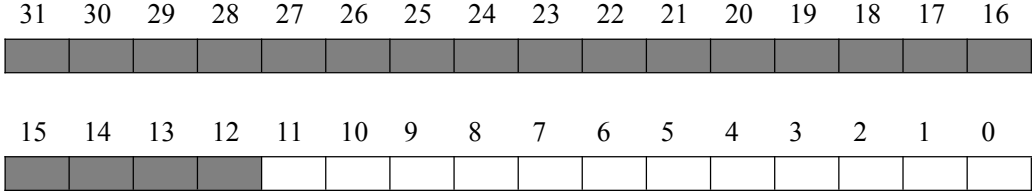




Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC_AUX6** ADC AUX6 value.

35.2.15 ADC_AUX7

Offset_Address: 0x0000_0044



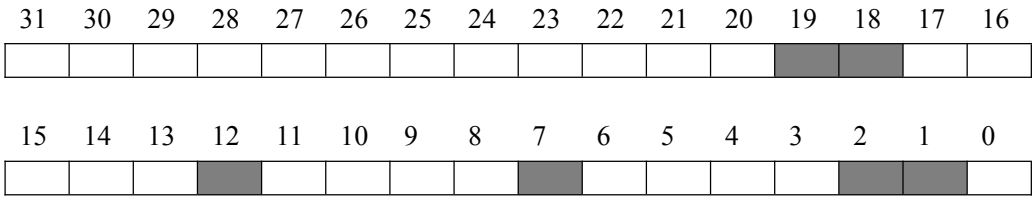
Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC_AUX7** ADC AUX7 value.

35.3 ADC1 Register Descriptions

ADC1_BASE: 0x51200000

35.3.1 ADC1_CTRL

Offset_Address: 0x0000_0000



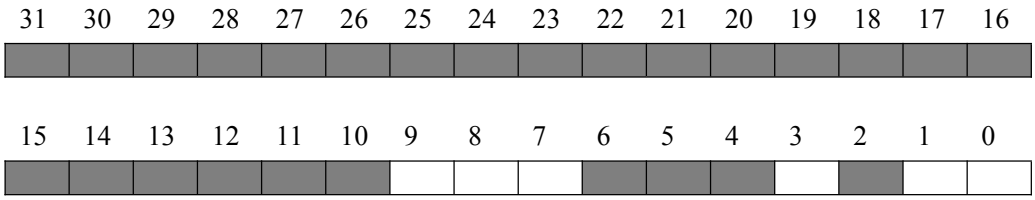
Bit	Type	Reset	Description
31	R/W	0	**ADC1_AUX7_CMD** 0: aux7 force transform disable. 1: aux7 force transform enable. We can use force transform enable when some part of the module is damaged.
30	R/W	0	**ADC1_AUX6_CMD** 0: aux6 force transform disable. 1: aux6 force transform enable.
29	R/W	0	**ADC1_AUX5_CMD** 0: aux5 force transform disable. 1: aux5 force transform enable.

28	R/W	0	**ADC1_AUX4_CMD** 0: aux4 force transform disable. 1: aux4 force transform enable.
27	R/W	0	**ADC1_AUX7_DET_INV** 0: aux7_det low valid. 1: aux7_det high valid.
26	R/W	0	**ADC1_AUX6_DET_INV** 0: aux6_det low valid. 1: aux6_det high valid.
25	R/W	0	**ADC1_AUX5_DET_INV** 0: aux5_det low valid. 1: aux5_det high valid.
24	R/W	0	**ADC1_AUX4_DET_INV** 0: aux4_det low valid. 1: aux4_det high valid.
23	R/W	0	**ADC1_AUX7_ENA** 0: aux7 disable. 1: aux7 enable.
22	R/W	0	**ADC1_AUX6_ENA** 0: aux6 disable. 1: aux6 enable.
21	R/W	0	**ADC1_AUX5_ENA** 0: aux5 disable. 1: aux5 enable.
20	R/W	0	**ADC1_AUX4_ENA** 0: aux4 disable. 1: aux4 enable.
19-18	R/W	0	Reserved.
17	R/W	0	**SOF** 0: hardware mode. 1: software mode.
16	R/W	0	**ADC1_AUX3_CMD** 0: aux3 force transform disable. 1: aux3 force transform enable.
15	R/W	0	**ADC1_AUX2_CMD** 0: aux2 force transform disable. 1: aux2 force transform enable.
14	R/W	0	**ADC1_AUX1_CMD** 0: aux1 force transform disable. 1: aux1 force transform enable.
13	R/W	0	**ADC1_AUX0_CMD** 0: aux0 force transform disable. 1: aux0 force transform enable.
12	R/W	0	Reserved.
11	R/W	0	**ADC1_AUX3_DET_INV** 0: aux3_det low valid. 1: aux3_det high valid.
10	R/W	0	**ADC1_AUX2_DET_INV** 0: aux2_det low valid. 1: aux2_det high valid.
9	R/W	0	**ADC1_AUX1_DET_INV** 0: aux1_det low valid. 1: aux1_det high valid.
8	R/W	0	**ADC1_AUX0_DET_INV** 0: aux0_det low valid. 1: aux0_det high valid.
7	R/W	0	Reserved.
6	R/W	0	**ADC1_AUX3_ENA** 0: aux3 disable. 1: aux3 enable.
5	R/W	0	**ADC1_AUX2_ENA** 0: aux2 disable. 1: aux2 enable.

4	R/W	0	**ADC1_AUX1_ENA** 0: aux1 disable. 1: aux1 enable.
3	R/W	0	**ADC1_AUX0_ENA** 0: aux0 disable. 1: aux0 enable.
2-1	R/W	0	Reserved.
0	R/W	0	**ADC1_RST** ADC1 soft rst : 0 : normal. 1 : reset.

35.3.2 ADC1_CFG

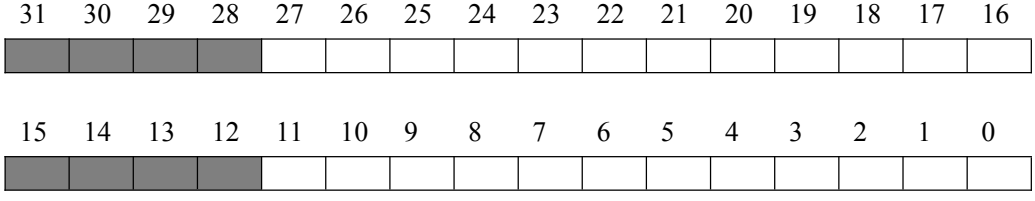
Offset_Address: 0x0000_0004



Bit	Type	Reset	Description
31-10	R	0	Reserved.
9-7	R/W	110	AUX_ADR aux address
6-4	R/W	010	Reserved.
3	R/W	0	ADC1_MOD
2	R/W	0	Reserved.
1-0	R/W	10	ADC1_PD To analog block.

35.3.3 ADC1_IMR

Offset_Address: 0x0000_008



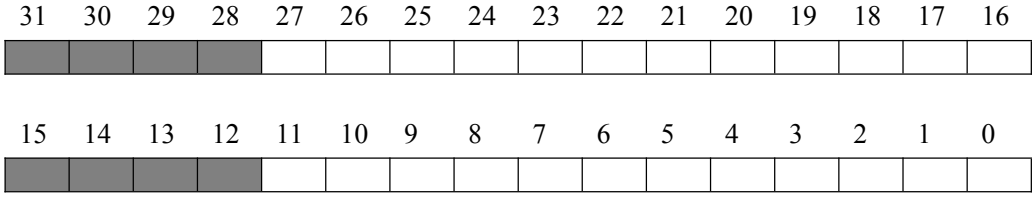
Bit	Type	Reset	Description
31-28	R	0	Reserved.
27	R/W	1	ADC1_AUX7_VALUE_IM ADC1 aux7 value interrupt mask: 0: mask disabled. 1: mask enabled.
26	R/W	1	ADC1_AUX7_STOP_IM ADC1 aux7 stop interrupt mask: 0: mask disabled.

			1: mask enabled.
25	R/W	1	ADC1_AUX7_START_IM ADC1 aux7 start interrupt mask: 0: mask disabled. 1: mask enabled.
24	R/W	1	ADC1_AUX6_VALUE_IM ADC1 aux6 value interrupt mask: 0: mask disabled. 1: mask enabled.
23	R/W	1	ADC1_AUX6_STOP_IM ADC1 aux6 stop interrupt mask: 0: mask disabled. 1: mask enabled.
22	R/W	1	ADC1_AUX6_START_IM ADC1 aux6 start interrupt mask: 0: mask disabled. 1: mask enabled.
21	R/W	1	ADC1_AUX5_VALUE_IM ADC1 aux5 value interrupt mask: 0: mask disabled. 1: mask enabled.
20	R/W	1	ADC1_AUX5_STOP_IM ADC1 aux5 stop interrupt mask: 0: mask disabled. 1: mask enabled.
19	R/W	1	ADC1_AUX5_START_IM ADC1 aux5 start interrupt mask: 0: mask disabled. 1: mask enabled.
18	R/W	1	ADC1_AUX4_VALUE_IM ADC1 aux4 value interrupt mask: 0: mask disabled. 1: mask enabled.
17	R/W	1	ADC1_AUX4_STOP_IM ADC1 aux4 stop interrupt mask: 0: mask disabled. 1: mask enabled.
16	R/W	1	ADC1_AUX4_START_IM ADC1 aux4 start interrupt mask: 0: mask disabled. 1: mask enabled.
15-12	R/W	1	Reserved.
11	R/W	1	ADC1_AUX3_VALUE_IM ADC1 aux3 value interrupt mask: 0: mask disabled. 1: mask enabled.
10	R/W	1	ADC1_AUX3_STOP_IM ADC1 aux3 stop interrupt mask: 0: mask disabled. 1: mask enabled.
9	R/W	1	ADC1_AUX3_START_IM ADC1 aux3 start interrupt mask: 0: mask disabled. 1: mask enabled.
8	R/W	1	ADC1_AUX2_VALUE_IM ADC1 aux2 value interrupt mask: 0: mask disabled. 1: mask enabled.
7	R/W	1	ADC1_AUX2_STOP_IM ADC1 aux2 stop interrupt mask: 0: mask disabled. 1: mask enabled.
6	R/W	1	ADC1_AUX2_START_IM ADC1 aux2 start interrupt mask:

			0: mask disabled. 1: mask enabled.
5	R/W	1	ADC1_AUX1_VALUE_IM ADC1 aux1 value interrupt mask: 0: mask disabled. 1: mask enabled.
4	R/W	1	ADC1_AUX1_STOP_IM ADC1 aux1 stop interrupt mask: 0: mask disabled. 1: mask enabled.
3	R/W	1	ADC1_AUX1_START_IM ADC1 aux1 start interrupt mask: 0: mask disabled. 1: mask enabled.
2	R/W	1	ADC1_AUX0_VALUE_IM ADC1 aux0 value interrupt mask: 0: mask disabled. 1: mask enabled.
1	R/W	1	ADC1_AUX0_STOP_IM ADC1 aux0 stop interrupt mask: 0: mask disabled. 1: mask enabled.
0	R/W	1	ADC1_AUX0_START_IM ADC1 aux0 start interrupt mask: 0: mask disabled. 1: mask enabled.

35.3.4 ADC1_STA

Offset_Address:0x0000_000C

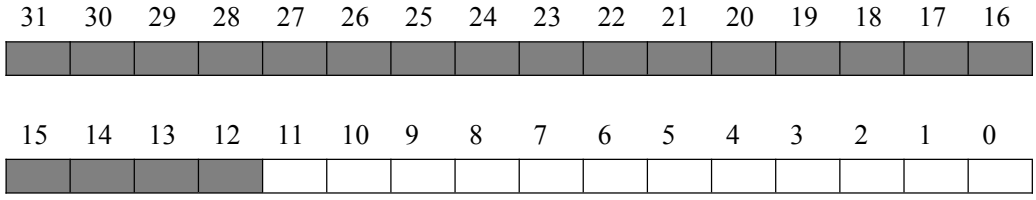


Bit	Type	Reset	Description
30-28	R	0	Reserved.
27	R/W	0	ADC1_AUX7_VALUE_INT Aux7 value interrupt status.
26	R/W	0	ADC1_AUX7_STOP_INT Aux7 stop interrupt status.
25	R/W	0	ADC1_AUX7_START_INT Aux7 start interrupt status.
24	R/W	0	ADC1_AUX6_VALUE_INT Aux6 value interrupt status.
23	R/W	0	ADC1_AUX6_STOP_INT Aux6 stop interrupt status.
22	R/W	0	ADC1_AUX6_START_INT Aux6 start interrupt status.
21	R/W	0	ADC1_AUX5_VALUE_INT Aux5 value interrupt status.
20	R/W	0	ADC1_AUX5_STOP_INT Aux5 stop interrupt status.
19	R/W	0	ADC1_AUX5_START_INT Aux5 start interrupt status.
18	R/W	0	ADC1_AUX4_VALUE_INT Aux4 value interrupt status.
17	R/W	0	ADC1_AUX4_STOP_INT

			Aux4 stop interrupt status.
16	R/W	0	ADC1_AUX4_START_INT Aux4 start interrupt status.
15-12	R	0	Reserved.
11	R/W	0	ADC1_AUX3_VALUE_INT Aux3 value interrupt status.
10	R/W	0	ADC1_AUX3_STOP_INT Aux3 stop interrupt status.
9	R/W	0	ADC1_AUX3_START_INT Aux3 start interrupt status.
8	R/W	0	ADC1_AUX2_VALUE_INT Aux2 value interrupt status.
7	R/W	0	ADC1_AUX2_STOP_INT Aux2 stop interrupt status.
6	R/W	0	ADC1_AUX2_START_INT Aux2 start interrupt status.
5	R/W	0	ADC1_AUX1_VALUE_INT Aux1 value interrupt status.
4	R/W	0	ADC1_AUX1_STOP_INT Aux1 stop interrupt status.
3	R/W	0	ADC1_AUX1_START_INT Aux1 start interrupt status.
2	R/W	0	ADC1_AUX0_VALUE_INT Aux0 value interrupt status.
1	R/W	0	ADC1_AUX0_STOP_INT Aux0 stop interrupt status.
0	R/W	0	ADC1_AUX0_START_INT Aux0 start interrupt status.

35.3.5 ADC1_AUX0

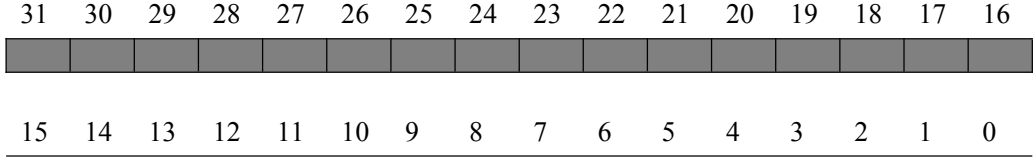
Offset_Address: 0x0000_0014

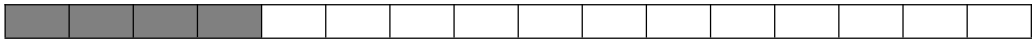


Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC1_AUX0** ADC1 AUX0 value.

35.3.6 ADC1_AUX1

Offset_Address: 0x0000_0018

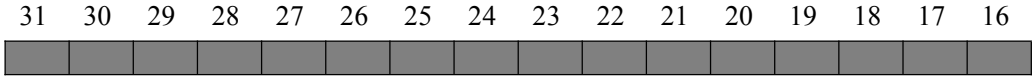


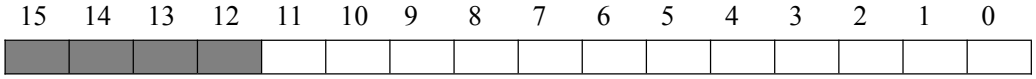


Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC1_AUX1** ADC1 AUX1 value.

35.3.7 ADC1_AUX2

Offset_Address: 0x0000_001C

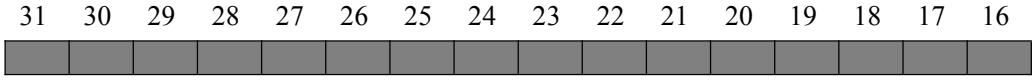


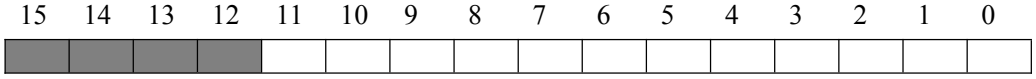


Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC1_AUX2** ADC1 AUX1 value.

35.3.8 ADC1_AUX3

Offset_Address: 0x0000_0020

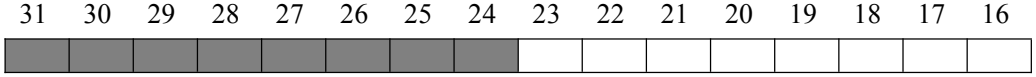


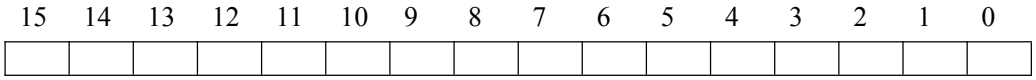


Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC1_AUX3** ADC1 AUX1 value.

35.3.9 ADC1_DBNCNT

Offset_Address: 0x0000_002C

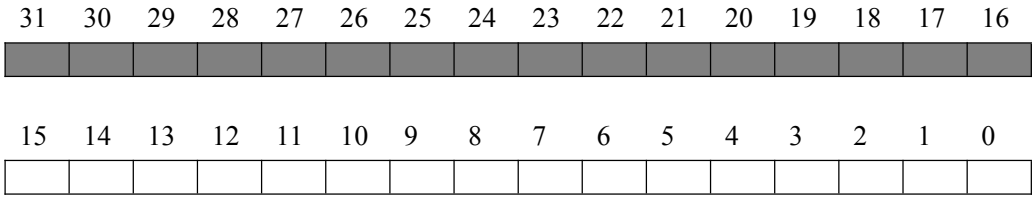




Bit	Type	Reset	Description
31-24	R	0	Reserved..
23-0	R/W	24'hfffff	**ADC1_DBNCNT** Debounce counter of APB Clock for eliminating bounce . After the time of **DBNCNT,** we'll think the value of the signal is stable and valid. Note: we must set the value of **DBNCNT** before enable a channel.

35.3.10 ADC1_INTER

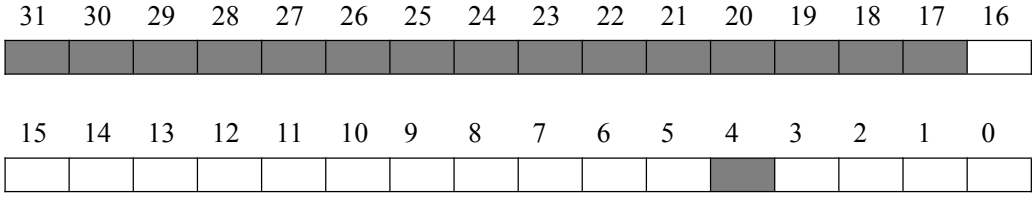
Offset_Address: 0x0000_0030



Bit	Type	Reset	Description
31-16	R	0	Reserved..
15-0	R/W	16'hfff	**ADC1_INTER** ADC1 transform interval, ie. The counter of ADC Sample clock..The time between two transforms

35.3.11 ADC1_SCTL

Offset_Address: 0x0000_0034

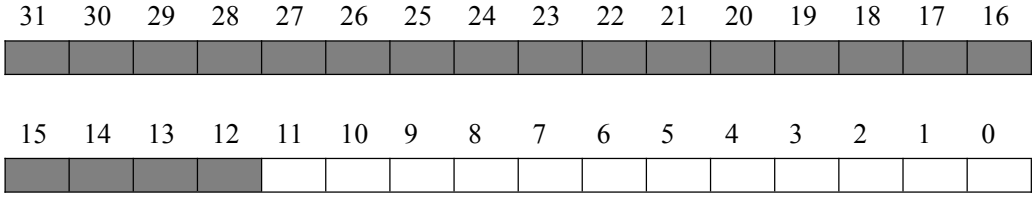


Bit	Type	Reset	Description
31-17	R	0	reserved.
16	R	0	ADC1 aux7_req_s
15	R	0	ADC1 aux6_req_s
14	R	0	ADC1 aux5_req_s
13	R	0	ADC1 aux4_req_s
12-10	W/R	0	aux_sel_s[2:0]
9	W/R	0	adc_csn_s

8	W/R	0	adc_si_s
7	W/R	0	adc_clk_s
6	R	0	adc_so_s
5	R	0	adc_busy_s
4	R	0	reserved.
3	R	0	ADC1 aux3_req_s
2	R	0	ADC1 aux2_req_s
1	R	0	ADC1 aux1_req_s
0	R	0	ADC1 aux0_req_s

35.3.12 ADC1_AUX4

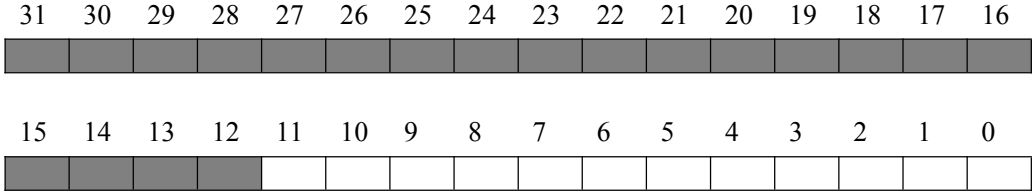
Offset_Address: 0x0000_0038



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC1_AUX4** ADC1 AUX4 value.

35.3.13 ADC1_AUX5

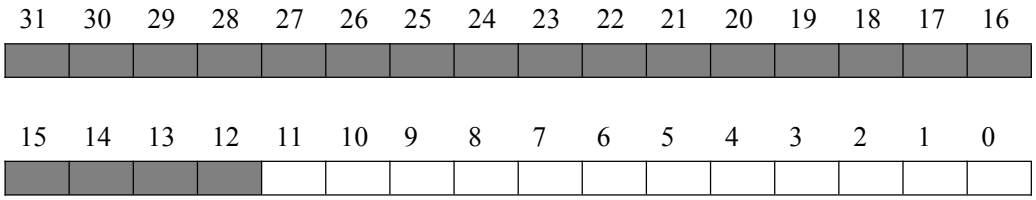
Offset_Address: 0x0000_003C



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC1_AUX5** ADC AUX5 value.

35.3.14 ADC1_AUX6

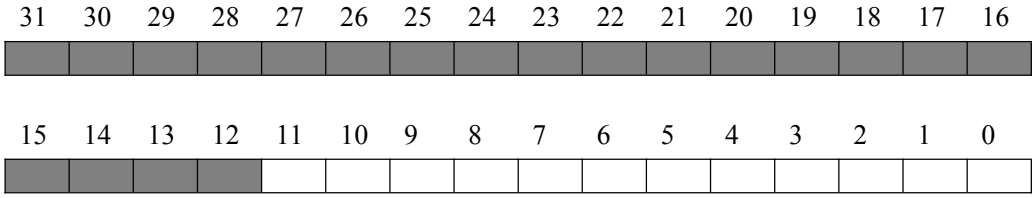
Offset_Address: 0x0000_0040



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC1_AUX6** ADC AUX6 value.

35.3.15 ADC1_AUX7

Offset_Address: 0x0000_0044



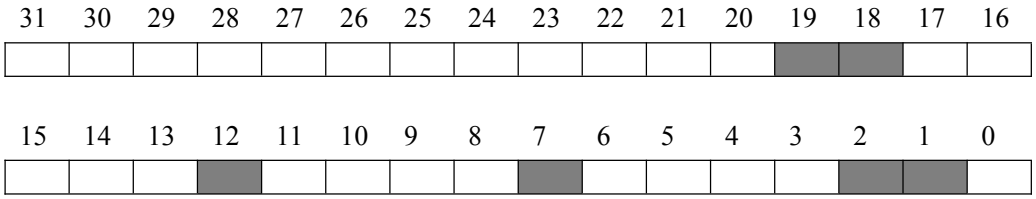
Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC1_AUX7** ADC AUX7 value.

35.4 ADC2 Register Descriptions

ADC2_BASE: 0x51300000

35.4.1 ADC2_CTRL

Offset_Address: 0x0000_0000



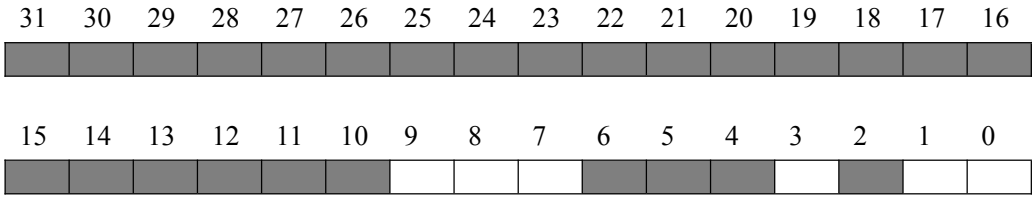
Bit	Type	Reset	Description
31	R/W	0	**ADC2_AUX7_CMD** 0: aux7 force transform disable. 1: aux7 force transform enable. We can use force transform enable when some part of the module is damaged.
30	R/W	0	**ADC2_AUX6_CMD** 0: aux6 force transform disable. 1: aux6 force transform enable.

29	R/W	0	**ADC2_AUX5_CMD** 0: aux5 force transform disable. 1: aux5 force transform enable.
28	R/W	0	**ADC2_AUX4_CMD** 0: aux4 force transform disable. 1: aux4 force transform enable.
27	R/W	0	**ADC2_AUX7_DET_INV** 0: aux7_det low valid. 1: aux7_det high valid.
26	R/W	0	**ADC2_AUX6_DET_INV** 0: aux6_det low valid. 1: aux6_det high valid.
25	R/W	0	**ADC2_AUX5_DET_INV** 0: aux5_det low valid. 1: aux5_det high valid.
24	R/W	0	**ADC2_AUX4_DET_INV** 0: aux4_det low valid. 1: aux4_det high valid.
23	R/W	0	**ADC2_AUX7_ENA** 0: aux7 disable. 1: aux7 enable.
22	R/W	0	**ADC2_AUX6_ENA** 0: aux6 disable. 1: aux6 enable.
21	R/W	0	**ADC2_AUX5_ENA** 0: aux5 disable. 1: aux5 enable.
20	R/W	0	**ADC2_AUX4_ENA** 0: aux4 disable. 1: aux4 enable.
19-18	R/W	0	Reserved.
17	R/W	0	**SOF** 0: hardware mode. 1: software mode.
16	R/W	0	**ADC2_AUX3_CMD** 0: aux3 force transform disable. 1: aux3 force transform enable.
15	R/W	0	**ADC2_AUX2_CMD** 0: aux2 force transform disable. 1: aux2 force transform enable.
14	R/W	0	**ADC2_AUX1_CMD** 0: aux1 force transform disable. 1: aux1 force transform enable.
13	R/W	0	**ADC2_AUX0_CMD** 0: aux0 force transform disable. 1: aux0 force transform enable.
12	R/W	0	Reserved.
11	R/W	0	**ADC2_AUX3_DET_INV** 0: aux3_det low valid. 1: aux3_det high valid.
10	R/W	0	**ADC2_AUX2_DET_INV** 0: aux2_det low valid. 1: aux2_det high valid.
9	R/W	0	**ADC2_AUX1_DET_INV** 0: aux1_det low valid. 1: aux1_det high valid.
8	R/W	0	**ADC2_AUX0_DET_INV** 0: aux0_det low valid. 1: aux0_det high valid.
7	R/W	0	Reserved.
6	R/W	0	**ADC2_AUX3_ENA** 0: aux3 disable. 1: aux3 enable.

5	R/W	0	**ADC2_AUX2_ENA** 0: aux2 disable. 1: aux2 enable.
4	R/W	0	**ADC2_AUX1_ENA** 0: aux1 disable. 1: aux1 enable.
3	R/W	0	**ADC2_AUX0_ENA** 0: aux0 disable. 1: aux0 enable.
2-1	R/W	0	Reserved.
0	R/W	0	**ADC2_RST** ADC soft rst : 0 : normal. 1 : reset.

35.4.2 ADC2_CFG

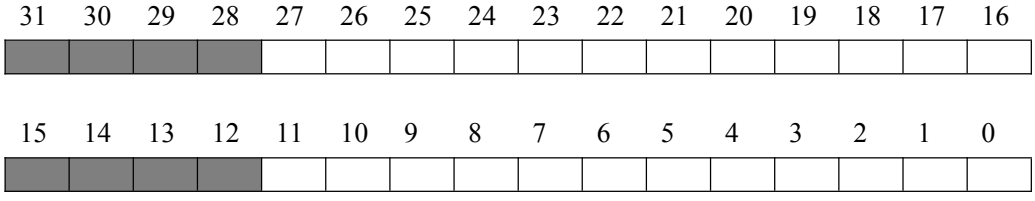
Offset_Address: 0x0000_0004



Bit	Type	Reset	Description
31-10	R	0	Reserved.
9-7	R/W	110	AUX_ADR aux address
6-4	R/W	010	Reserved.
3	R/W	0	ADC2_MOD
2	R/W	0	Reserved.
1-0	R/W	10	ADC2_PD To analog block.

35.4.3 ADC2_IMR

Offset_Address: 0x0000_008



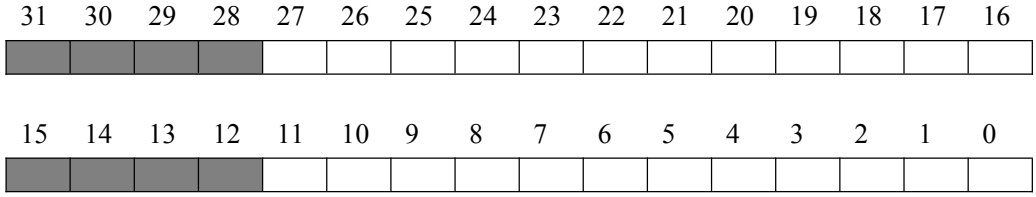
Bit	Type	Reset	Description
31-28	R	0	Reserved.
27	R/W	1	ADC2_AUX7_VALUE_IM ADC2 aux7 value interrupt mask: 0: mask disabled. 1: mask enabled.

26	R/W	1	ADC2_AUX7_STOP_IM ADC2 aux7 stop interrupt mask: 0: mask disabled. 1: mask enabled.
25	R/W	1	ADC2_AUX7_START_IM ADC2 aux7 start interrupt mask: 0: mask disabled. 1: mask enabled.
24	R/W	1	ADC2_AUX6_VALUE_IM ADC2 aux6 value interrupt mask: 0: mask disabled. 1: mask enabled.
23	R/W	1	ADC2_AUX6_STOP_IM ADC2 aux6 stop interrupt mask: 0: mask disabled. 1: mask enabled.
22	R/W	1	ADC2_AUX6_START_IM ADC2 aux6 start interrupt mask: 0: mask disabled. 1: mask enabled.
21	R/W	1	ADC2_AUX5_VALUE_IM ADC2 aux5 value interrupt mask: 0: mask disabled. 1: mask enabled.
20	R/W	1	ADC2_AUX5_STOP_IM ADC2 aux5 stop interrupt mask: 0: mask disabled. 1: mask enabled.
19	R/W	1	ADC2_AUX5_START_IM ADC2 aux5 start interrupt mask: 0: mask disabled. 1: mask enabled.
18	R/W	1	ADC2_AUX4_VALUE_IM ADC2 aux4 value interrupt mask: 0: mask disabled. 1: mask enabled.
17	R/W	1	ADC2_AUX4_STOP_IM ADC2 aux4 stop interrupt mask: 0: mask disabled. 1: mask enabled.
16	R/W	1	ADC2_AUX4_START_IM ADC2 aux4 start interrupt mask: 0: mask disabled. 1: mask enabled.
15-12	R/W	1	Reserved.
11	R/W	1	ADC2_AUX3_VALUE_IM ADC2 aux3 value interrupt mask: 0: mask disabled. 1: mask enabled.
10	R/W	1	ADC2_AUX3_STOP_IM ADC2 aux3 stop interrupt mask: 0: mask disabled. 1: mask enabled.
9	R/W	1	ADC2_AUX3_START_IM ADC2 aux3 start interrupt mask: 0: mask disabled. 1: mask enabled.
8	R/W	1	ADC2_AUX2_VALUE_IM ADC2 aux2 value interrupt mask: 0: mask disabled. 1: mask enabled.
7	R/W	1	ADC2_AUX2_STOP_IM ADC2 aux2 stop interrupt mask: 0: mask disabled.

			1: mask enabled.
6	R/W	1	ADC2_AUX2_START_IM ADC2 aux2 start interrupt mask: 0: mask disabled. 1: mask enabled.
5	R/W	1	ADC2_AUX1_VALUE_IM ADC2 aux1 value interrupt mask: 0: mask disabled. 1: mask enabled.
4	R/W	1	ADC2_AUX1_STOP_IM ADC2 aux1 stop interrupt mask: 0: mask disabled. 1: mask enabled.
3	R/W	1	ADC2_AUX1_START_IM ADC2 aux1 start interrupt mask: 0: mask disabled. 1: mask enabled.
2	R/W	1	ADC2_AUX0_VALUE_IM ADC2 aux0 value interrupt mask: 0: mask disabled. 1: mask enabled.
1	R/W	1	ADC2_AUX0_STOP_IM ADC2 aux0 stop interrupt mask: 0: mask disabled. 1: mask enabled.
0	R/W	1	ADC2_AUX0_START_IM ADC2 aux0 start interrupt mask: 0: mask disabled. 1: mask enabled.

35.4.4 ADC2_STA

Offset_Address:0x0000_000C

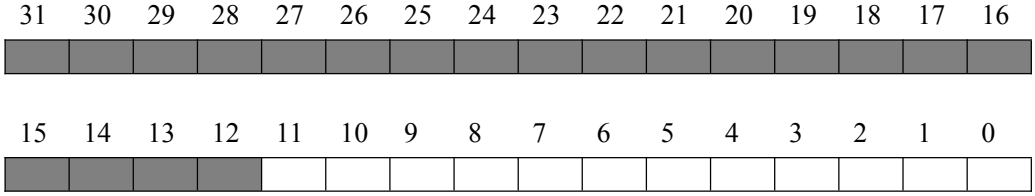


Bit	Type	Reset	Description
30-28	R	0	Reserved.
27	R/W	0	ADC2_AUX7_VALUE_INT Aux7 value interrupt status.
26	R/W	0	ADC2_AUX7_STOP_INT Aux7 stop interrupt status.
25	R/W	0	ADC2_AUX7_START_INT Aux7 start interrupt status.
24	R/W	0	ADC2_AUX6_VALUE_INT Aux6 value interrupt status.
23	R/W	0	ADC2_AUX6_STOP_INT Aux6 stop interrupt status.
22	R/W	0	ADC2_AUX6_START_INT Aux6 start interrupt status.
21	R/W	0	ADC2_AUX5_VALUE_INT Aux5 value interrupt status.
20	R/W	0	ADC2_AUX5_STOP_INT Aux5 stop interrupt status.
19	R/W	0	ADC2_AUX5_START_INT Aux5 start interrupt status.

18	R/W	0	ADC2_AUX4_VALUE_INT Aux4 value interrupt status.
17	R/W	0	ADC2_AUX4_STOP_INT Aux4 stop interrupt status.
16	R/W	0	ADC2_AUX4_START_INT Aux4 start interrupt status.
15-12	R	0	Reserved.
11	R/W	0	ADC2_AUX3_VALUE_INT Aux3 value interrupt status.
10	R/W	0	ADC2_AUX3_STOP_INT Aux3 stop interrupt status.
9	R/W	0	ADC2_AUX3_START_INT Aux3 start interrupt status.
8	R/W	0	ADC2_AUX2_VALUE_INT Aux2 value interrupt status.
7	R/W	0	ADC2_AUX2_STOP_INT Aux2 stop interrupt status.
6	R/W	0	ADC2_AUX2_START_INT Aux2 start interrupt status.
5	R/W	0	ADC2_AUX1_VALUE_INT Aux1 value interrupt status.
4	R/W	0	ADC2_AUX1_STOP_INT Aux1 stop interrupt status.
3	R/W	0	ADC2_AUX1_START_INT Aux1 start interrupt status.
2	R/W	0	ADC2_AUX0_VALUE_INT Aux0 value interrupt status.
1	R/W	0	ADC2_AUX0_STOP_INT Aux0 stop interrupt status.
0	R/W	0	ADC2_AUX0_START_INT Aux0 start interrupt status.

35.4.5 ADC2_AUX0

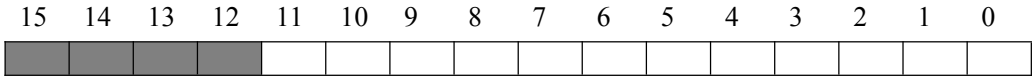
Offset_Address: 0x0000_0014



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC2_AUX0** ADC2 AUX0 value.

35.4.6 ADC2_AUX1

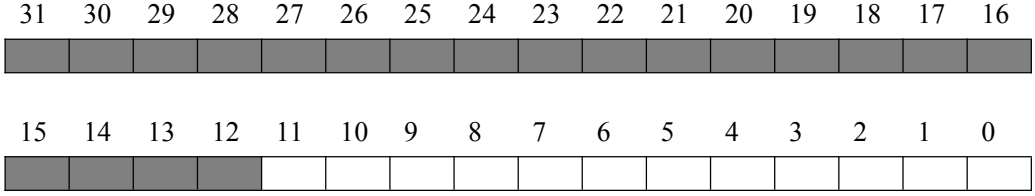
Offset_Address: 0x0000_0018



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC2_AUX1** ADC AUX1 value.

35.4.7 ADC2_AUX2

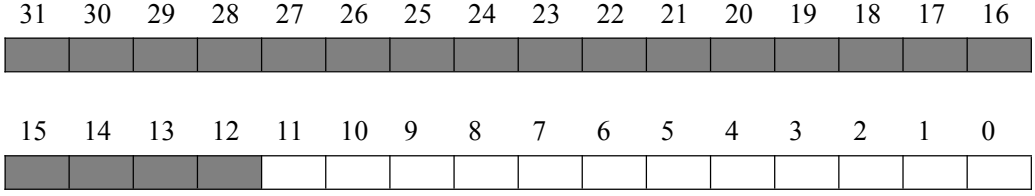
Offset_Address: 0x0000_001C



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC2_AUX2** ADC AUX1 value.

35.4.8 ADC2_AUX3

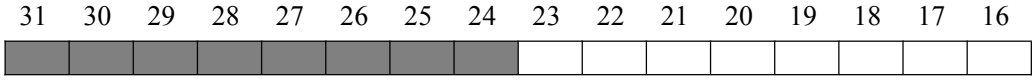
Offset_Address: 0x0000_0020

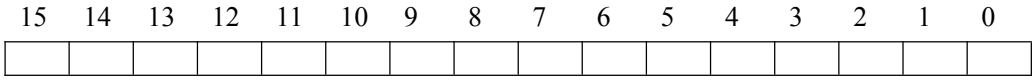


Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC2_AUX3** ADC AUX1 value.

35.4.9 ADC2_DBNCNT

Offset_Address: 0x0000_002C

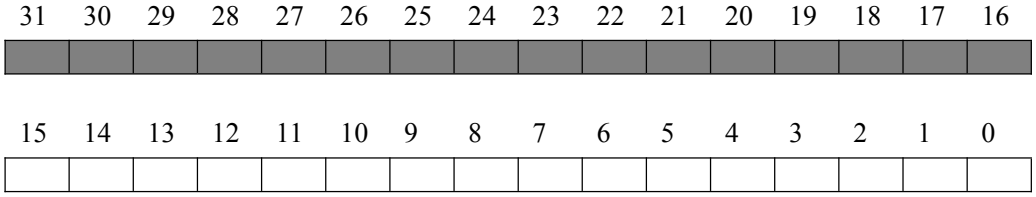




Bit	Type	Reset	Description
31-24	R	0	Reserved..
23-0	R/W	24'hfffff	**ADC2_DBNCNT** Debounce counter of APB Clock for eliminating bounce . After the time of **DBNCNT,** we'll think the value of the signal is stable and valid. Note: we must set the value of **DBNCNT** before enable a channel.

35.4.10 ADC2_INTER

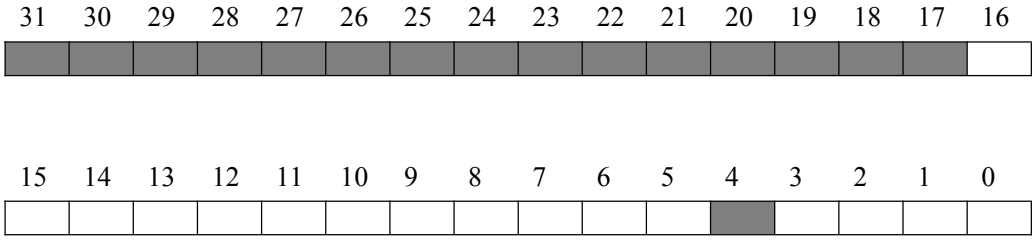
Offset_Address: 0x0000_0030



Bit	Type	Reset	Description
31-16	R	0	Reserved..
15-0	R/W	16'hfff	**ADC2_INTER** ADC transform interval, ie. The counter of ADC Sample clock..The time between two transforms

35.4.11 ADC2_SCTL

Offset_Address: 0x0000_0034

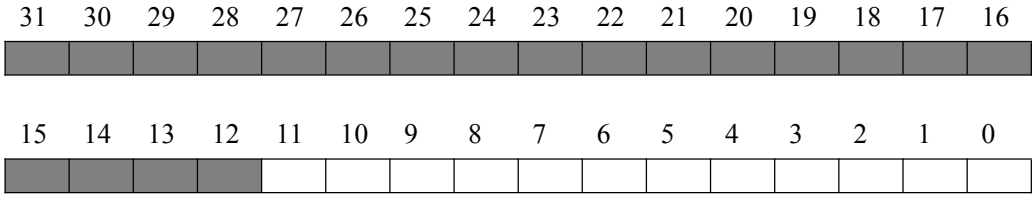


Bit	Type	Reset	Description
31-17	R	0	reserved.
16	R	0	ADC2 aux7_req_s
15	R	0	ADC2 aux6_req_s
14	R	0	ADC2 aux5_req_s
13	R	0	ADC2 aux4_req_s
12-10	W/R	0	aux_sel_s[2:0]
9	W/R	0	adc_csn_s
8	W/R	0	adc_si_s

7	W/R	0	adc_clk_s
6	R	0	adc_so_s
5	R	0	adc_busy_s
4	R	0	reserved.
3	R	0	ADC2 aux3_req_s
2	R	0	ADC2 aux2_req_s
1	R	0	ADC2 aux1_req_s
0	R	0	ADC2 aux0_req_s

35.4.12 ADC2_AUX4

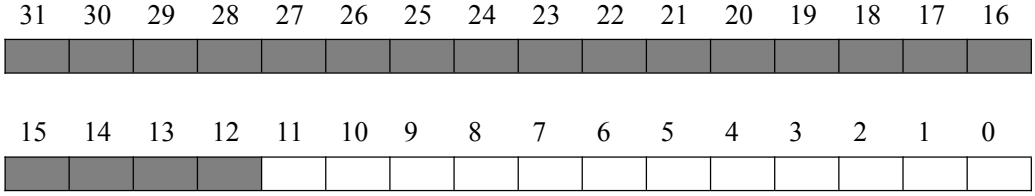
Offset_Address: 0x0000_0038



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC2_AUX4** ADC2 AUX4 value.

35.4.13 ADC2_AUX5

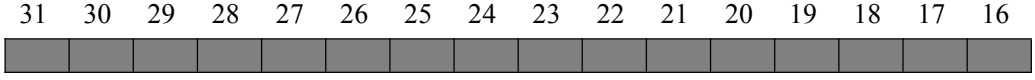
Offset_Address: 0x0000_003C

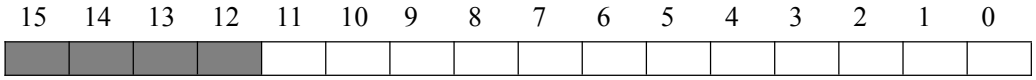


Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC2_AUX5** ADC AUX5 value.

35.4.14 ADC2_AUX6

Offset_Address: 0x0000_0040

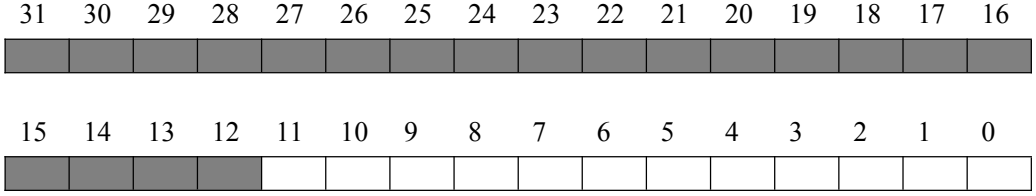




Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC2_AUX6** ADC AUX6 value.

35.4.15 ADC2_AUX7

Offset_Address: 0x0000_0044



Bit	Type	Reset	Description
31-12	R	0	Reserved..
11-0	R	0	**ADC2_AUX7** ADC AUX7 value.

36. RDC

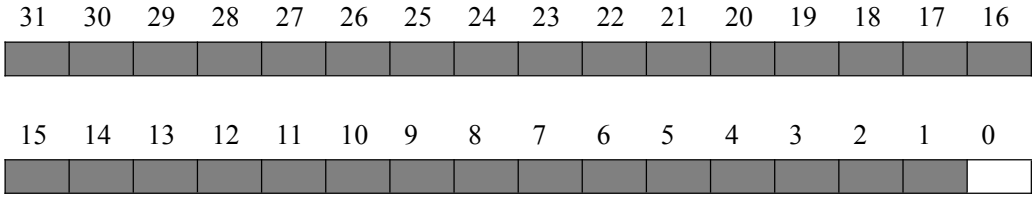
36.1 DDR RDC Register Description

DDR_RDC_BASE: 0x51D00000

36.1.1 CTRL

The CTRL register can lock down the programming configuration of the RDC. It also enables the BLK_IDX autoincrement.

Offset_Address: 0x0000_0000



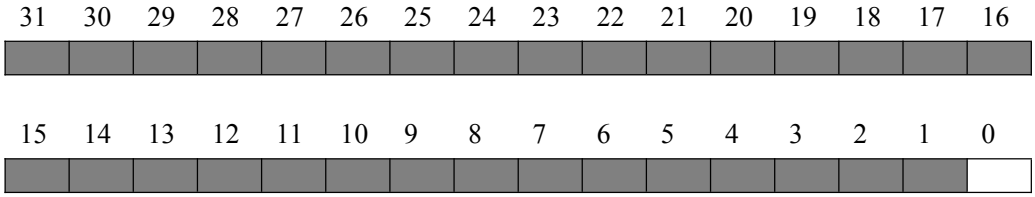
Bit	Type	Reset	Description
31	R/W	0	SEC_CFG_LOCK Security configuration lockdown. When set to 1, CTRL, BLK_LUT, IRQ_EN and IRQ_SET become read-only registers. To set this bit to 0, the RDC must be reset.

30-9	R	0	Reserved.
8	R/W	1	INC_BLK_IDX BLK_IDX autoincrement enable.
7-0	R	0	Reserved.

36.1.2 BLK_MAX

The BLK_MAX register controls the maximum value that can be set in the BLK_IDX register.

Offset_Address: 0x0000_0010

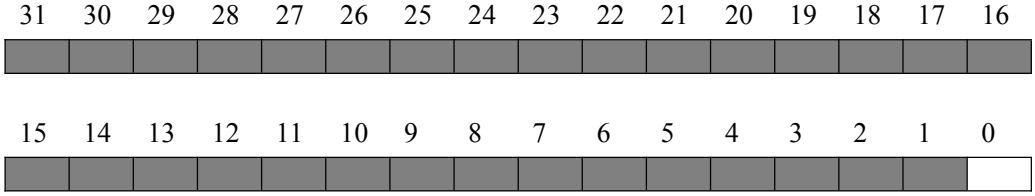


Bit	Type	Reset	Description
31-0	R	0	BLK_MAX The maximum value that the BLK_IDX register can be set to.

36.1.3 BLK_CFG

The BLK_CFG register returns the block size and whether the block initialization is in progress.

Offset_Address: 0x0000_0014

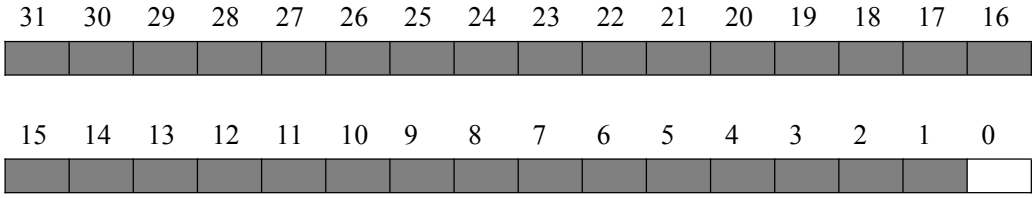


Bit	Type	Reset	Description
31	R	0	INIT_ONGOING When set to 1, it indicates that the block initialization is in progress.
30-4	R	0	Reserved.
3-0	R	0	BLK_SIZE 15 = 1MB.

36.1.4 BLK_IDX_ADDR

The BLK_IDX register controls the index pointer that selects each group of 32 blocks.

Offset_Address: 0x0000_0018

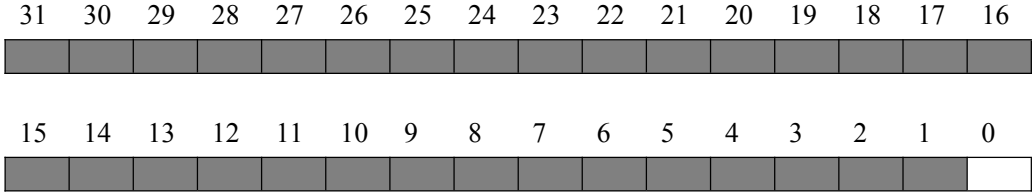


Bit	Type	Reset	Description
30-4	R	0	Reserved.
3-0	R/W	0	BLK_IDX The index value that the RDC uses for accesses to the block-based LUT, using the BLK_LUT register.

36.1.5 BLK_LUT

The BLK_LUT register controls the state of a block. The BLK_LUT register controls up to 32 blocks and the BLK_IDX register sets which 32-block region in the *Look Up Table* (LUT) is accessed.

Offset_Address: 0x0000_001C

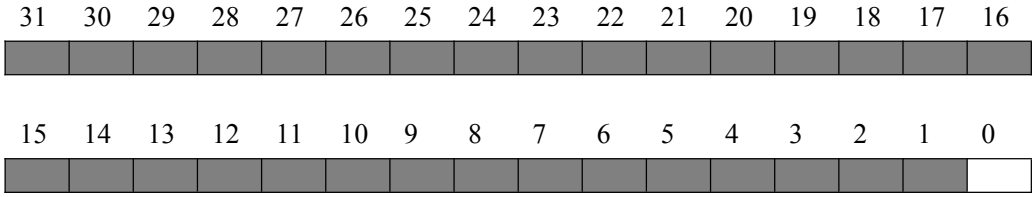


Bit	Type	Reset	Description
31-0	R/W	0	BLK_LUT Block-based gating Look Up Table (LUT). Each bit controls the state of a single block, but the block number depends on the value of the index pointer, BLK_IDX. For example: • If BLK_IDX is 0, then bit[0] contains the state of block #0 and bit[31] contains the state of block #336.1. • If BLK_IDX is 1, then bit[0] contains the state of block #32, bit[31] contains the state of block #63. When: • Bit[n] = 0, the block can be access. • Bit[n] = 1, the block can not be access. If auto-increment is enabled, that is CTRL.INC_BLK_IDX == 1, then the RDC increments the BLK_IDX by one, for each word read or word write of the BLK_LUT register. 512MB = 16 x 32 blocks

36.1.6 IRQ_STAT

The IRQ_STAT register returns the interrupt status.

Offset_Address: 0x0000_0020



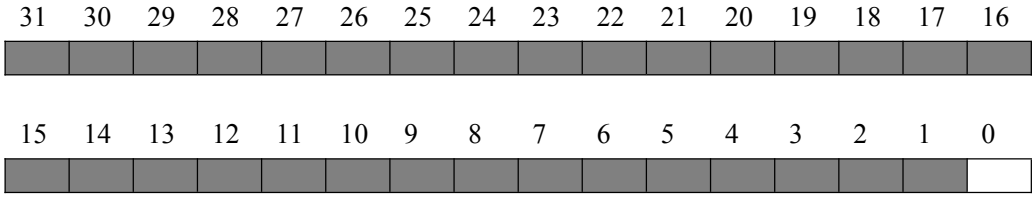
Bit	Type	Reset	Description
30-5	R	0	Reserved.
4-0	R	0	IRQ_STAT [4]: Error int from CPU; [3]: Error int from M4;

			[2]: Error int from CHBUS(DMA/USB/SDC/ETH); [1]: Error int from GBUS(GPU/MFC/QOI/BLEND/PXP/WRAP); [0]: Error int from VBUS(LCD/ITU);

36.1.7 IRQ_CLEAR

The IRQ_CLEAR register clears the interrupt.

Offset_Address: 0x0000_0024

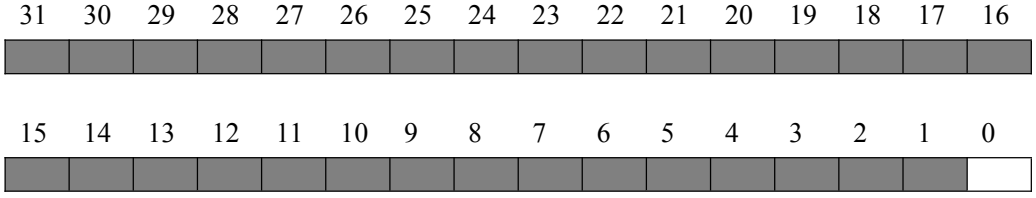


Bit	Type	Reset	Description
30-5	R	0	Reserved.
4-0	R/W	0	IRQ_CLEAR [4]: Clear interrupt from CPU; [3]: Clear interrupt from M4; [2]: Clear interrupt from CHBUS(DMA/USB/SDC/ETH); [1]: Clear interrupt from GBUS(GPU/MFC/QOI/BLEND/PXP/WRAP); [0]: Clear interrupt from VBUS(LCD/ITU);

36.1.8 IRQ_EN

The IRQ_EN register enables the interrupt.

Offset_Address: 0x0000_0028

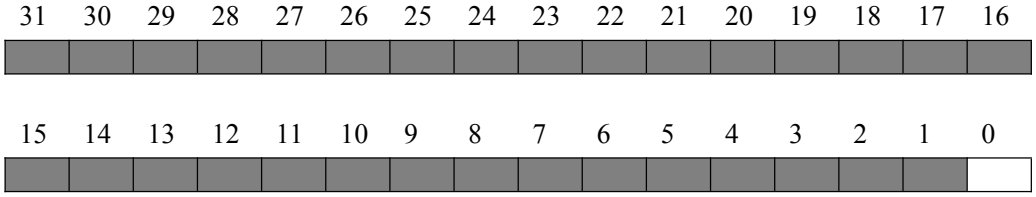


Bit	Type	Reset	Description
30-5	R	0	Reserved.
4-0	R/W	0	IRQ_EN [4]: Enable interrupt from CPU; [3]: Enable interrupt from M4; [2]: Enable interrupt from CHBUS(DMA/USB/SDC/ETH); [1]: Enable interrupt from GBUS(GPU/MFC/QOI/BLEND/PXP/WRAP); [0]: Enable interrupt from VBUS(LCD/ITU);

36.1.9 IRQ_INFO1_VBUS

The IRQ_INFO1 register returns the address of the AXI transaction that triggered the violation Interrupt.

Offset_Address: 0x0000_002C

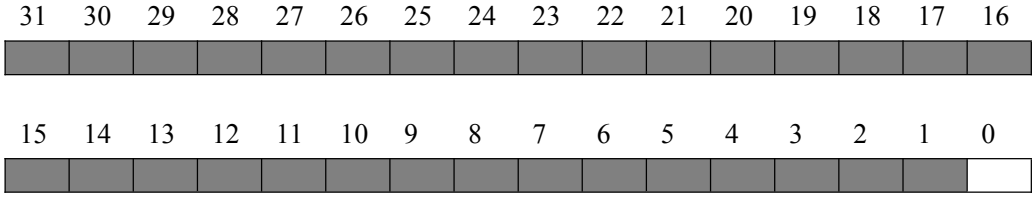


Bit	Type	Reset	Description
31-0	R	0	FIRST_ADDR Returns the address of the violation that triggered the interrupt.

36.1.10 IRQ_INFO2_VBUS

The IRQ_INFO2 register returns extra information about the AXI transaction that triggered the violation interrupt. It also indicates the occurrence of multiple violations and coincident read and write violations.

Offset_Address: 0x0000_0030



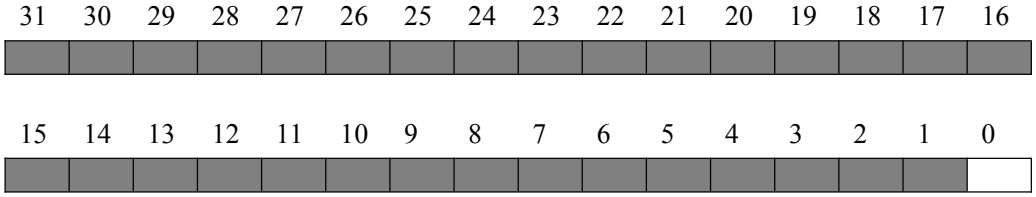
Bit	Type	Reset	Description
31	R	0	AxPROT Returns the **AxPROT[1]** value of the AXI transaction that triggered the violation interrupt: 0 = A Secure AXI transaction triggered the interrupt. 1 = A Non-secure AXI transaction triggered the interrupt.
30-25	R	0	reserved.
24	R	0	ERR_MULTI Indicates whether multiple violations have occurred. 0 = No more violations have occurred since the interrupt was triggered. 1 = More violations have occurred since the interrupt was triggered.
23-21	R	0	reserved.
20	R	0	ERR_BOTH Indicates whether coincident read and write violations triggered the interrupt: 0 = No coincident read and write violations have occurred when the interrupt was triggered. 1 = Coincident read and write violations have occurred when the interrupt was triggered.
19-17	R	0	reserved.
16	R	0	WnR Indicates whether a read or write AXI transaction triggered the violation interrupt: 0 = A read AXI transaction triggered the interrupt.

			1 = A write AXI transaction triggered the interrupt.
15-0	R	0	AxID Returns the ID of the AXI transaction that triggered the violation interrupt [4]: 1'b0: LCD; 1'b1: ITU;

36.1.11 IRQ_INFO1_GBUS

The IRQ_INFO1 register returns the address of the AXI transaction that triggered the violation Interrupt.

Offset_Address: 0x0000_0034

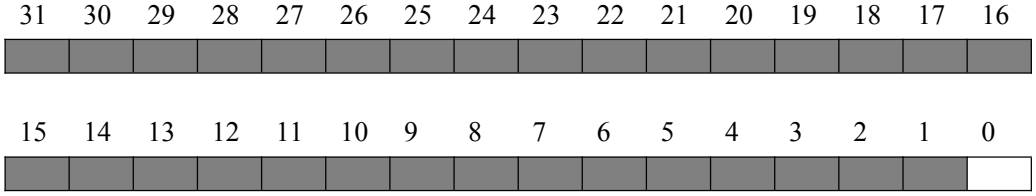


Bit	Type	Reset	Description
31-0	R	0	FIRST_ADDR Returns the address of the violation that triggered the interrupt.

36.1.12 IRQ_INFO2_GBUS

The IRQ_INFO2 register returns extra information about the AXI transaction that triggered the violation interrupt. It also indicates the occurrence of multiple violations and coincident read and write violations.

Offset_Address: 0x0000_0038



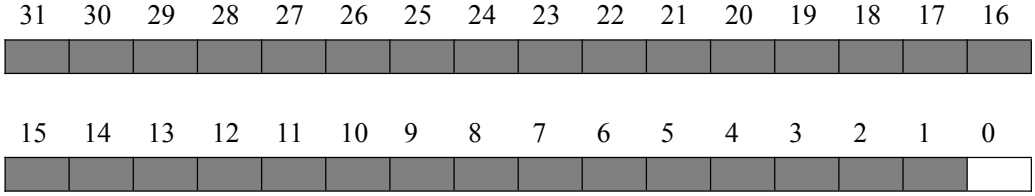
Bit	Type	Reset	Description
31	R	0	AxPROT Returns the **AxPROT[1]** value of the AXI transaction that triggered the violation interrupt: 0 = A Secure AXI transaction triggered the interrupt. 1 = A Non-secure AXI transaction triggered the interrupt.
30-25	R	0	reserved.
24	R	0	ERR_MULTI Indicates whether multiple violations have occurred. 0 = No more violations have occurred since the interrupt was triggered. 1 = More violations have occurred since the interrupt was triggered.

23-21	R	0	reserved.
20	R	0	ERR_BOTH Indicates whether coincident read and write violations triggered the interrupt: 0 = No coincident read and write violations have occurred when the interrupt was triggered. 1 = Coincident read and write violations have occurred when the interrupt was triggered.
19-17	R	0	reserved.
16	R	0	WnR Indicates whether a read or write AXI transaction triggered the violation interrupt: 0 = A read AXI transaction triggered the interrupt. 1 = A write AXI transaction triggered the interrupt.
15-0	R	0	AxID Returns the ID of the AXI transaction that triggered the violation interrupt [6:4]: 0: MFC; 1: GPU; 2: QOI; 3: BLENDING; 4: PXP; 5: WRAP;

36.1.13 IRQ_INFO1_CHBUS

The IRQ_INFO1 register returns the address of the AXI transaction that triggered the violation Interrupt.

Offset_Address: 0x0000_003C

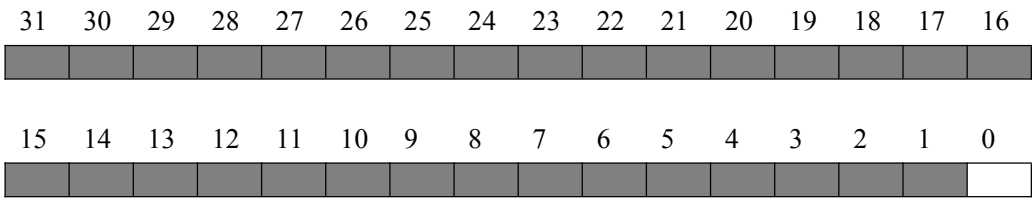


Bit	Type	Reset	Description
31-0	R	0	FIRST_ADDR Returns the address of the violation that triggered the interrupt.

36.1.14 IRQ_INFO2_CHBUS

The IRQ_INFO2 register returns extra information about the AXI transaction that triggered the violation interrupt. It also indicates the occurrence of multiple violations and coincident read and write violations.

Offset_Address: 0x0000_0040

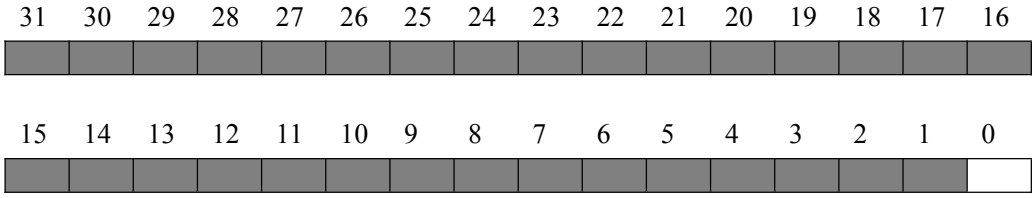


Bit	Type	Reset	Description
31	R	0	AxPROT Returns the **AxPROT[1]** value of the AXI transaction that triggered the violation interrupt: 0 = A Secure AXI transaction triggered the interrupt. 1 = A Non-secure AXI transaction triggered the interrupt.
30-25	R	0	reserved.
24	R	0	ERR_MULTI Indicates whether multiple violations have occurred. 0 = No more violations have occurred since the interrupt was triggered. 1 = More violations have occurred since the interrupt was triggered.
23-21	R	0	reserved.
20	R	0	ERR_BOTH Indicates whether coincident read and write violations triggered the interrupt: 0 = No coincident read and write violations have occurred when the interrupt was triggered. 1 = Coincident read and write violations have occurred when the interrupt was triggered.
19-17	R	0	reserved.
16	R	0	WnR Indicates whether a read or write AXI transaction triggered the violation interrupt: 0 = A read AXI transaction triggered the interrupt. 1 = A write AXI transaction triggered the interrupt.
15-0	R	0	AxID Returns the ID of the AXI transaction that triggered the violation interrupt 10'h80/10'h102: DMA; 10'h101: USB; 10'h83: SD0; 10'h104: SD1; 10'h105: ETH;

36.1.15 IRQ_INFO1_M4

The IRQ_INFO1 register returns the address of the AXI transaction that triggered the violation Interrupt.

Offset_Address: 0x0000_0044



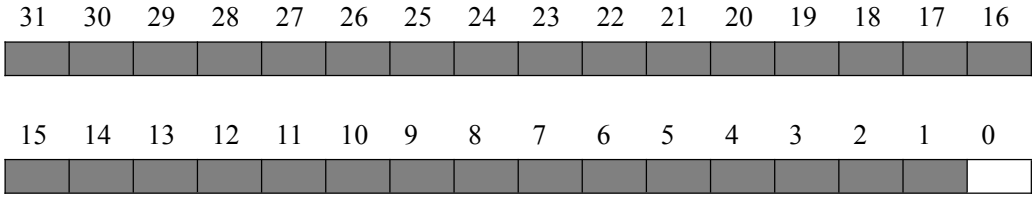
Bit	Type	Reset	Description
31-0	R	0	FIRST_ADDR Returns the address of the violation that triggered the interrupt.

36.1.16 IRQ_INFO2_M4

The IRQ_INFO2 register returns extra information about the AXI transaction that triggered the violation interrupt. It also indicates the occurrence of multiple violations and coincident read and write

violations.

Offset_Address: 0x0000_0048

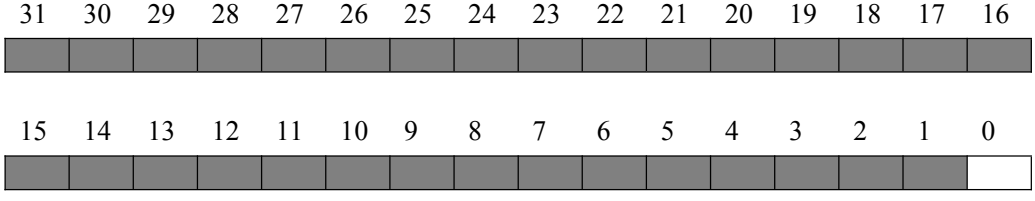


Bit	Type	Reset	Description
31	R	0	AxPROT Returns the **AxPROT[1]** value of the AXI transaction that triggered the violation interrupt: 0 = A Secure AXI transaction triggered the interrupt. 1 = A Non-secure AXI transaction triggered the interrupt.
30-25	R	0	reserved.
24	R	0	ERR_MULTI Indicates whether multiple violations have occurred. 0 = No more violations have occurred since the interrupt was triggered. 1 = More violations have occurred since the interrupt was triggered.
23-21	R	0	reserved.
20	R	0	ERR_BOTH Indicates whether coincident read and write violations triggered the interrupt: 0 = No coincident read and write violations have occurred when the interrupt was triggered. 1 = Coincident read and write violations have occurred when the interrupt was triggered.
19-17	R	0	reserved.
16	R	0	WnR Indicates whether a read or write AXI transaction triggered the violation interrupt: 0 = A read AXI transaction triggered the interrupt. 1 = A write AXI transaction triggered the interrupt.
15-0	R	0	AxID Returns the ID of the AXI transaction that triggered the violation interrupt

36.1.17 IRQ_INFO1_CPU

The IRQ_INFO1 register returns the address of the AXI transaction that triggered the violation Interrupt.

Offset_Address: 0x0000_004C

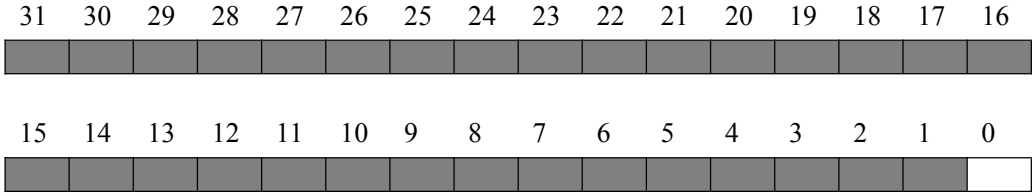


Bit	Type	Reset	Description
31-0	R	0	FIRST_ADDR Returns the address of the violation that triggered the interrupt.

36.1.18 IRQ_INFO2_CPU

The IRQ_INFO2 register returns extra information about the AXI transaction that triggered the violation interrupt. It also indicates the occurrence of multiple violations and coincident read and write violations.

Offset_Address: 0x0000_0050

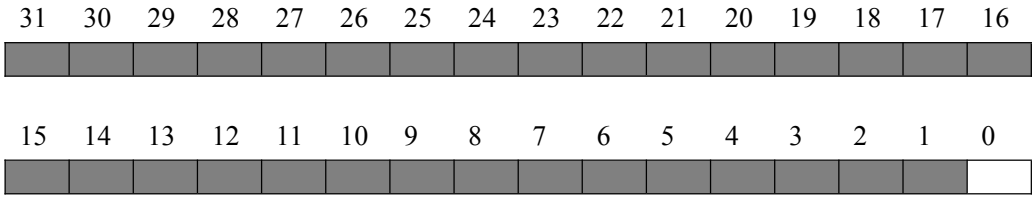


Bit	Type	Reset	Description
31	R	0	AxPROT Returns the **AxPROT[1]** value of the AXI transaction that triggered the violation interrupt: 0 = A Secure AXI transaction triggered the interrupt. 1 = A Non-secure AXI transaction triggered the interrupt.
30-25	R	0	reserved.
24	R	0	ERR_MULTI Indicates whether multiple violations have occurred. 0 = No more violations have occurred since the interrupt was triggered. 1 = More violations have occurred since the interrupt was triggered.
23-21	R	0	reserved.
20	R	0	ERR_BOTH Indicates whether coincident read and write violations triggered the interrupt: 0 = No coincident read and write violations have occurred when the interrupt was triggered. 1 = Coincident read and write violations have occurred when the interrupt was triggered.
19-17	R	0	reserved.
16	R	0	WnR Indicates whether a read or write AXI transaction triggered the violation interrupt: 0 = A read AXI transaction triggered the interrupt. 1 = A write AXI transaction triggered the interrupt.
15-0	R	0	AxID Returns the ID of the AXI transaction that triggered the violation interrupt

36.1.19 IRQ_SET

The IRQ_SET register sets the interrupt.

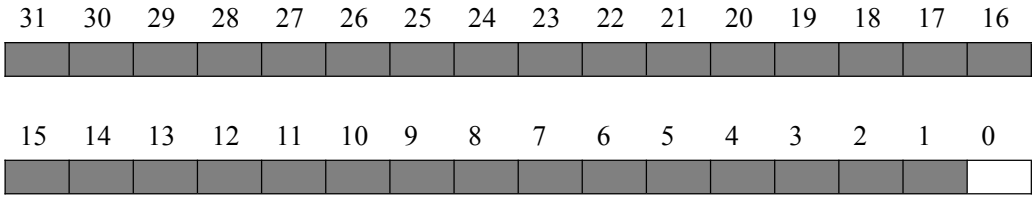
Offset_Address: 0x0000_0054



Bit	Type	Reset	Description
30-5	R	0	Reserved.
4-0	R/W	0	IRQ_SET [4]: Set interrupt from CPU; [3]: Set interrupt from M4; [2]: Set interrupt from CHBUS(DMA/USB/SDC/ETH); [1]: Set interrupt from GBUS(GPU/MFC/QOI/BLEND/PXP/WRAP); [0]: Set interrupt from VBUS(LCD/ITU);

36.1.20 LUT_MASTER_SEL

Offset_Address: 0x0000_0058



Bit	Type	Reset	Description
30-4	R	0	Reserved.
3-0	R/W	0	Lut_master_sel Select which master's lut is going to write or read by software. 0: LCD; 1: ITU; 2: MFC; 3: GPU; 4: QOI; 5: BLEND; 6: PXP; 7: WRAP; 8: DMA; 9: USB; 10: SD0; 11: SD1; 12: ETH; 13: M4; 14: CPU;

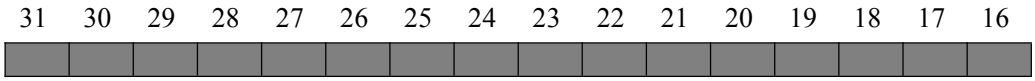
36.2 SRAM RDC Register Description

SRAM_RDC_BASE: 0x51E00000

36.2.1 CTRL

The CTRL register can lock down the programming configuration of the RDC. It also enables the BLK_IDX autoincrement.

Offset_Address: 0x0000_0000

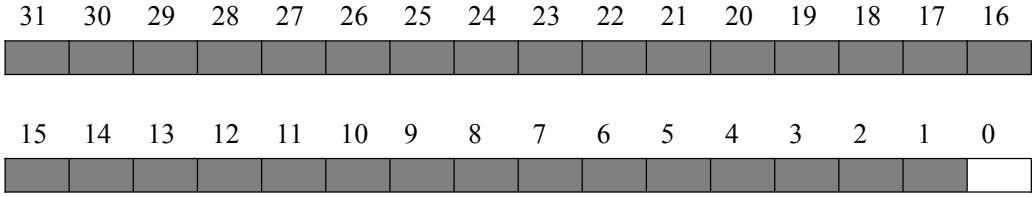


Bit	Type	Reset	Description
31	R/W	0	SEC_CFG_LOCK Security configuration lockdown. When set to 1, CTRL, BLK_LUT, IRQ_EN and IRQ_SET become read-only registers. To set this bit to 0, the RDC must be reset.
30-9	R	0	Reserved.
8	R/W	1	INC_BLK_IDX BLK_IDX autoincrement enable.
7-0	R	0	Reserved.

36.2.2 BLK_MAX

The BLK_MAX register controls the maximum value that can be set in the BLK_IDX register.

Offset_Address: 0x0000_0010

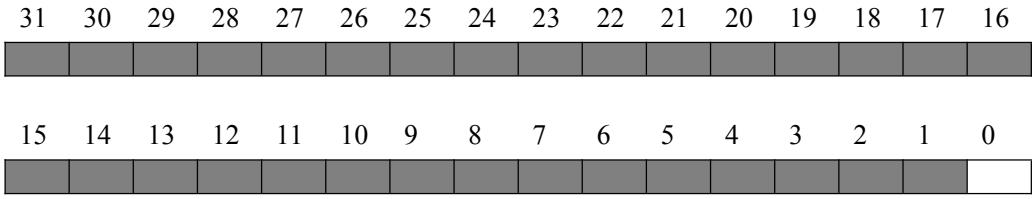


Bit	Type	Reset	Description
31-0	R	0	BLK_MAX The maximum value that the BLK_IDX register can be set to.

36.2.3 BLK_CFG

The BLK_CFG register returns the block size and whether the block initialization is in progress.

Offset_Address: 0x0000_0014



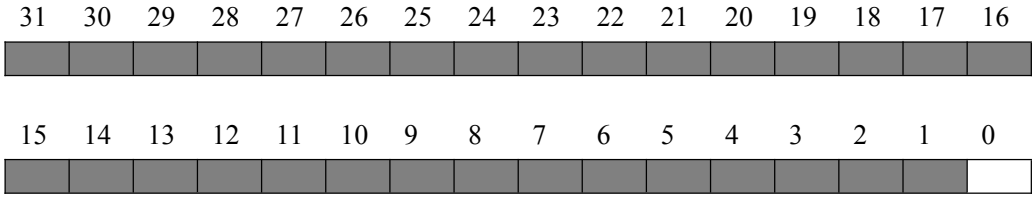
Bit	Type	Reset	Description
31	R/W	0	INIT_ONGOING When set to 1, it indicates that the block initialization is in progress.
30-4	R	0	Reserved.
3-0	R	0	BLK_SIZE This field returns the block size: 12 = 128KB

36.2.4 BLK_LUT

The BLK_LUT register controls the state of a block. The BLK_LUT register controls up to 32 blocks

and the BLK_IDX register sets which 32-block region in the *Look Up Table* (LUT) is accessed.

Offset_Address: 0x0000_001C

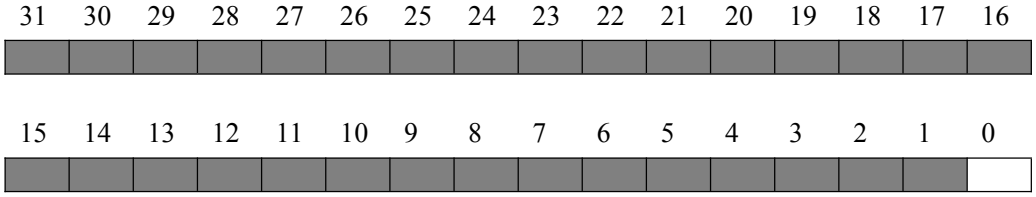


Bit	Type	Reset	Description
31-0	R/W	0	BLK_LUT Block-based gating *Look Up Table* (LUT). Each bit controls the state of a single block, but the block number depends on the value of the index pointer, BLK_IDX. For example: • If BLK_IDX is 0, then bit[0] contains the state of block #0 and bit[31] contains the state of block #336.1. • If BLK_IDX is 1, then bit[0] contains the state of block #32, bit[31] contains the state of block #63. When: • Bit[n] = 0, the block can be access. • Bit[n] = 1, the block can not be access. If auto-increment is enabled, that is CTRL.INC_BLK_IDX == 1, then the RDC increments the BLK_IDX by one, for each word read or word write of the BLK_LUT register. 4MB = 32 blocks

36.2.5 IRQ_STAT

The IRQ_STAT register returns the interrupt status.

Offset_Address: 0x0000_0020

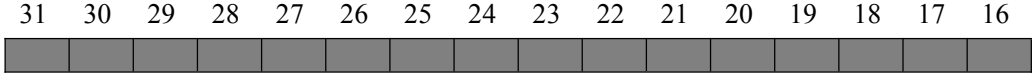


Bit	Type	Reset	Description
30-2	R	0	Reserved.
1-0	R	0	IRQ_STAT [1]: Error int from M4; [0]: Error int from Others(LCD/ITU/GPU/MFC/QOI/BLEND/PXP/WRAP/DMA/USB/SDC/ETH/CPU);

36.2.6 IRQ_CLEAR

The IRQ_CLEAR register clears the interrupt.

Offset_Address: 0x0000_0024

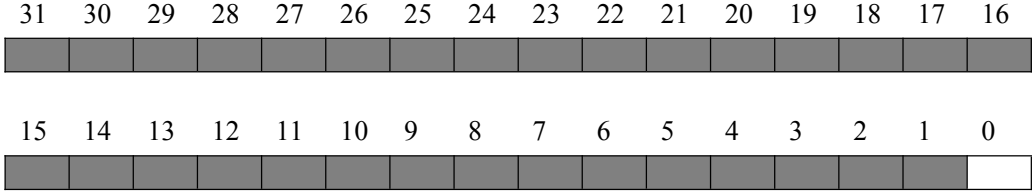


Bit	Type	Reset	Description
30-5	R	0	Reserved.
4-0	R/W	0	IRQ_CLEAR [1]: Clear interrupt from M4; [0]: Clear interrupt from Others(LCD/ITU/GPU/MFC/QOI/BLEND/PXP/WRAP/DMA/USB/SDC/ETH/CPU);

36.2.7 IRQ_EN

The IRQ_EN register enables the interrupt.

Offset_Address: 0x0000_0028

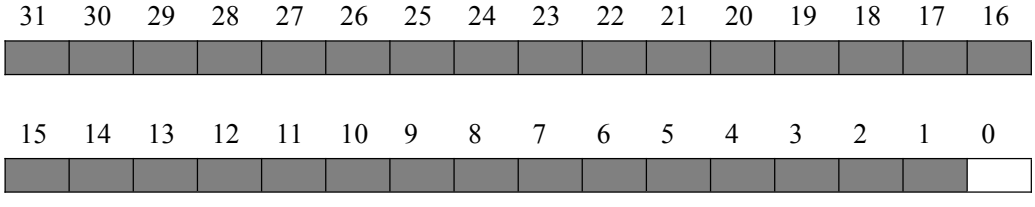


Bit	Type	Reset	Description
30-5	R	0	Reserved.
4-0	R/W	0	IRQ_EN [1]: Enable interrupt from M4; [0]: Enable interrupt from Others(LCD/ITU/GPU/MFC/QOI/BLEND/PXP/WRAP/DMA/USB/SDC/ETH/CPU);

36.2.8 IRQ_INFO1_OTHERS

The IRQ_INFO1 register returns the address of the AXI transaction that triggered the violation Interrupt.

Offset_Address: 0x0000_002C



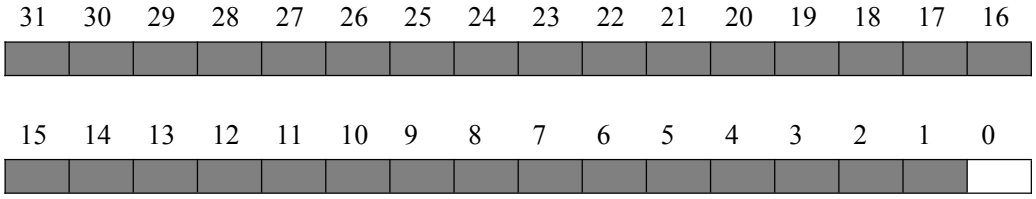
Bit	Type	Reset	Description
31-0	R	0	FIRST_ADDR Returns the address of the violation that triggered the interrupt.

36.2.9 IRQ_INFO2_OTHERS

The IRQ_INFO2 register returns extra information about the AXI transaction that triggered the violation interrupt. It also indicates the occurrence of multiple violations and coincident read and write

violations.

Offset_Address: 0x0000_0030

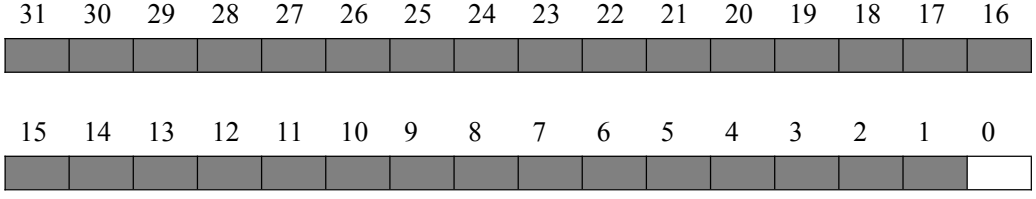


Bit	Type	Reset	Description
31	R	0	AxPROT Returns the **AxPROT[1]** value of the AXI transaction that triggered the violation interrupt: 0 = A Secure AXI transaction triggered the interrupt. 1 = A Non-secure AXI transaction triggered the interrupt.
30-25	R	0	reserved.
24	R	0	ERR_MULTI Indicates whether multiple violations have occurred. 0 = No more violations have occurred since the interrupt was triggered. 1 = More violations have occurred since the interrupt was triggered.
23-21	R	0	reserved.
20	R	0	ERR_BOTH Indicates whether coincident read and write violations triggered the interrupt: 0 = No coincident read and write violations have occurred when the interrupt was triggered. 1 = Coincident read and write violations have occurred when the interrupt was triggered.
19-17	R	0	reserved.
16	R	0	WnR Indicates whether a read or write AXI transaction triggered the violation interrupt: 0 = A read AXI transaction triggered the interrupt. 1 = A write AXI transaction triggered the interrupt.
15-0	R	0	AxID Returns the ID of the AXI transaction that triggered the violation interrupt [9:7]: 0: [6:4]: 0: LCD; 1: ITU;（not use） [9:7]: 1: [6:0]: 0: DMA; 3: SD0; [9:7]: 2: [6:0]: 1: USB; 2: DMA; 4: SD1; 5: ETH; [9:7]: 3: [6:4]: 0: MFC ; 1: GPU; 2:QOI; 3:BLENDING; 4: PXP; 5: WRAP; [9:7]: 4: CPU;

36.2.10 IRQ_INFO1_M4

The IRQ_INFO1 register returns the address of the AXI transaction that triggered the violation Interrupt.

Offset_Address: 0x0000_0044

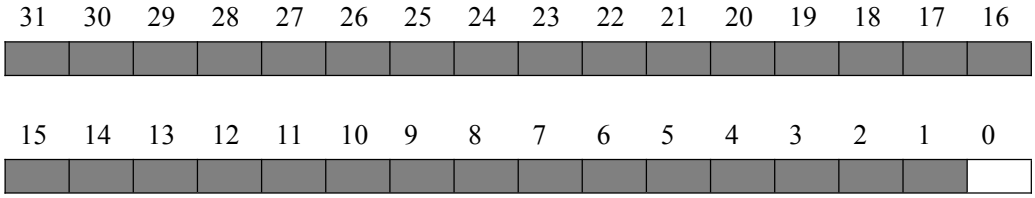


Bit	Type	Reset	Description
31-0	R	0	FIRST_ADDR Returns the address of the violation that triggered the interrupt.

36.2.11 IRQ_INFO2_M4

The IRQ_INFO2 register returns extra information about the AXI transaction that triggered the violation interrupt. It also indicates the occurrence of multiple violations and coincident read and write violations.

Offset_Address: 0x0000_0048

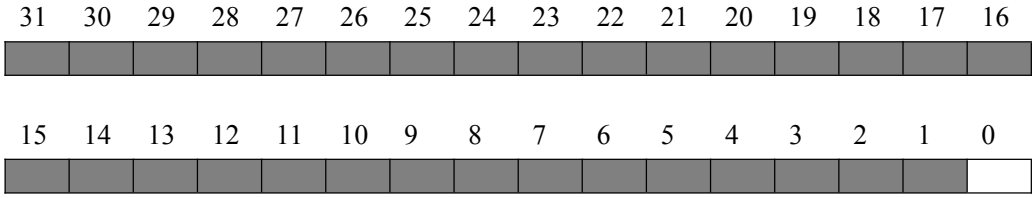


Bit	Type	Reset	Description
31	R	0	AxPROT Returns the **AxPROT[1]** value of the AXI transaction that triggered the violation interrupt: 0 = A Secure AXI transaction triggered the interrupt. 1 = A Non-secure AXI transaction triggered the interrupt.
30-25	R	0	reserved.
24	R	0	ERR_MULTI Indicates whether multiple violations have occurred. 0 = No more violations have occurred since the interrupt was triggered. 1 = More violations have occurred since the interrupt was triggered.
23-21	R	0	reserved.
20	R	0	ERR_BOTH Indicates whether coincident read and write violations triggered the interrupt: 0 = No coincident read and write violations have occurred when the interrupt was triggered. 1 = Coincident read and write violations have occurred when the interrupt was triggered.
19-17	R	0	reserved.
16	R	0	WnR Indicates whether a read or write AXI transaction triggered the violation interrupt: 0 = A read AXI transaction triggered the interrupt. 1 = A write AXI transaction triggered the interrupt.
15-0	R	0	AxID Returns the ID of the AXI transaction that triggered the violation interrupt

36.2.12 IRQ_SET

The IRQ_SET register sets the interrupt.

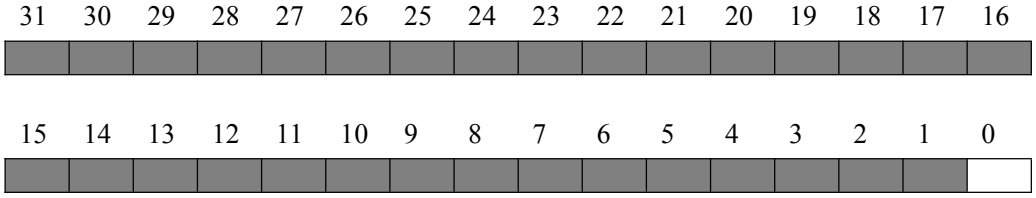
Offset_Address: 0x0000_0054



Bit	Type	Reset	Description
30-5	R	0	Reserved.
4-0	R/W	0	IRQ_SET [1]: Set interrupt from M4; [0]: Set interrupt fromOthers(LCD/ITU（not use）/GPU/MFC/QOI/BLEND/PXP/WRAP/DMA/USB/SDC/ETH/CPU);

36.2.13 LUT_MASTER_SEL

Offset_Address: 0x0000_0058



Bit	Type	Reset	Description
30-4	R	0	Reserved.
3-0	R/W	0	Lut_master_sel Select which master's lut is going to write or read by software. 0: LCD; 1: ITU;（not use） 2: MFC; 3: GPU; 4: QOI; 5: BLEND; 6: PXP; 7: WRAP; 8: DMA; 9: USB; 10: SD0; 11: SD1; 12: ETH; 13: M4; 14: CPU;

37. Mailbox

MAILBOX_BASE: 0x40E8_0000

37.1 Features

16 programmable mailboxes, each one has 16-bit mailboxes source register.

16 interrupt registers.

37.2 Register Descriptions

名		位宽	功能		
Source		[31:0]	Set to define which core is the source and which interrupt line is asserted for the acknowledge interrupt.When the register is programmed, it must be cleared to 0x00000000 before it can be reprogrammed.	0x000(mailbox 0) 0x040(mailbox 1)	32'h40e8_0000
Destination Set		[31:0]	Set Destination	0x004(mailbox 0)	
Destination Clear		[31:0]	Used to clear bits in the Mailbox Destination Register	0x008(mailbox 0)	
Destination Status		[31:0]	Defines which interrupt output to assert for the message.	0x00c(mailbox 0)	
IPCMxMODE	-	[31:2]	Read undefined. Write as zero.	0x10(mailbox 0)	
	Auto Link	[1]	Enable Auto Link.		
	Auto Acknowledge	[0]	Enable Auto Acknowledge.		
Mask Set		[31:0]	Set mask bit	0x014(mailbox 0)	
Mask Clear		[31:0]	Clear mask bit	0x018(mailbox 0)	
Mask Status		[31:0]	For each bit position: 1: enable interrupt 0: disable interrupt	0x01c(mailbox 0)	
IPCMxSEND	-	[31:2]	Read undefined. Write as zero.	0x020(mailbox 0)	
	Send	[1:0]	Send message： 00: idle 01: send message to dest-core 10: send message to source-core 11: reserved		
Data		[31:0]	Message data	0x024(mailbox 0) 0x7E0(mailbox 31)	
MaskIntStat		[31:0]	Mask interrupt	0x800	

RawIntStat	[31:0]	Raw interrupt	0x804	

38. MIPI DSI Host Controller

38.1 Mipi_dsi_host Features

The mipi_dsi_host has the following features:

- Compliant with MIPI Alliance standards
- Supports all commands defined in the MIPI Alliance specification for Display Command Set (DCS)
  - Transmission of all Command mode packets through the APB interface
  - Transmission of commands in low-power and high-speed during Video mode
- Supports 24 RGB up to 1080@60fps resolution
- Supports up to four D-PHY data lanes
- Bidirectional communication and escape mode support through data lane 0
- Supports non-continuous clock in D-PHY clock lane for additional power saving
- Supports Ultra Low-Power mode with PLL disabled
- ECC and Checksum capabilities
- Support for End of Transmission Packet (EoTp)
- Video mode pattern generator:
  - Vertical and horizontal color bar generation
  - BER pattern

39. MIPI CSI Host Controller

39.1 Mipi_csi_host Features

The mipi_csi2_host supports the following features:

- Compliant with MIPI Alliance standards
- Support MIPICSI2/ITU601/656 YUV422/420 input up to 1920x1080
- Support YUV420,YUV422,RGB888 format

- Up to four D-PHY RX data lanes
- Long and Short packet decoding
- Timing accurate signaling of Frame and Line synchronization packets
- Several frame formats
  - General Frame or Digital Interlaced Video with or without accurate sync timing
  - Data type (Packet or Frame level) and virtual channel interleaving
- Error detection and correction
  - PHY level
  - Packet level
  - Line level
  - Frame level

40. PKA

40.1 Features

The Public Key Accelerator(PKA) is designed to offload large arithmetic operations from the host's software. The host loads the parameters in the Public Key Accelerator and specifies the appropriate function to initiate the autonomous program. No further communication between the PKA and the system is required until the Public Key Accelerator finishes its task.

The PKA instance is highly parameterizable allowing it to be tuned to anywhere between the performance or area-centric ends of the application spectrum. The PKA can execute at least the following algorithms and precisions:

- RSA (with or without CRT): 1024, 2048, and 4096-bit.
- ECC(include SM2).

The F/W RAM only needs to be loaded once after reset. Since the PKA is a Harvard architecture, no operations within the PKA itself can modify the contents of the F/W area. Only the host can do this.

41. SPACC

41.1 Features

- Cipher and Hash modes (non-AEAD modes) to perform parallel hashing and encryption or decryption operations on a flexible protocol data unit (PDU) structure, to allow acceleration support of a variety of packet protocols.
  - Cipher algorithms:
    - AES, SM4
  - Cipher modes:
    - ECB, CBC, CTR, OFB, CFB
  - Hash (MAC) algorithms:
    - MD5, SHA-1，SHA-256/512
    - HMAC
    - CRC32
  - Hash modes:
    - Raw, HMAC
- Combined modes (AEAD modes):
  - AES-GCM, AES-CCM
- System bus secure access (Secure Bus) support for systems, that differentiate between secure and normal transaction modes.
- Scatter-Gather DMA through Data Descriptor Table (DDT) packet descriptors.

- The Secure Key Port (SKP) interface supports reading keys from OTP

42. TRNG

42.1 Features

Internal random (re)seed operation

128-bit random number generation

Serial random data stream output for auxiliary uses such as, DPA, TA, IPsec, and so on

Status/control I/O for allowing external monitoring of internal actions and states

43.Electrical Characteristics

43.1 Operation Conditions

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
DVDD_CSI	CSI IO Supply	3.0	3.3	3.6	V
		1.72	1.8	1.98	
DVDD_EMMC	EMMC IO Supply	3.0	3.3	3.6	V
		1.72	1.8	1.98	
DVDD_SD1	SD1 IO Supply	3.0	3.3	3.6	V
		1.72	1.8	1.98	
DVDD_PD1	PD1 IO Supply	3.0	3.3	3.6	V
		1.72	1.8	1.98	
DVDD_PD2	PD2 IO Supply	3.0	3.3	3.6	V
		1.72	1.8	1.98	
DVDD_PD3	PD3 IO Supply	3.0	3.3	3.6	V
		1.72	1.8	1.98	
DVDD_PD4	PD4 IO Supply	3.0	3.3	3.6	V
		1.72	1.8	1.98	
VDD	Core Supply	0.81	0.9	0.99	V
VDDQ15	DDR Supply	1.425	1.5	1.575	V
VDD18_DDR_PLL	DDR PLL Supply	1.72	1.8	1.98	V
AVDD18_CSI	CSI Supply	1.72	1.8	1.98	V
AVDD18_ADC	ADC Supply	1.72	1.8	1.98	V
AVDD09_PLL	PLL Supply	0.81	0.9	0.99	V
AVDD18_LVDS	LVDS Supply	1.72	1.8	1.98	V
AVDD18_DSI	DSI Supply	1.72	1.8	1.98	V
VDD33_USB/ VDD18_USB	USB Supply	3.0	3.3	3.6	V
VQPS	FUSE Supply	3.0	3.3	3.6	V
VDD33_RTC	RTC Supply	3.0	3.3	3.6	V

43.2 Current Consumption

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT
DVDD_CSI DVDD_EMMC DVDD_SD1 DVDD_PD1 DVDD_PD2 DVDD_PD3 DVDD_PD4 VDD33_USB	IO Supply, 3.3V		100		mA
VDD AVDD09_PLL	Core Supply, 0.9V		660		mA

VDDQ15	DDR Supply, 1.5V		160		mA
VDD18_DDR_PLL AVDD18_CSI AVDD18_ADC AVDD18_LVDS AVDD18_DSI	Analog Supply，1.8V		70		mA
VDD33_RTC	RTC Supply, 3.3V		130		uA

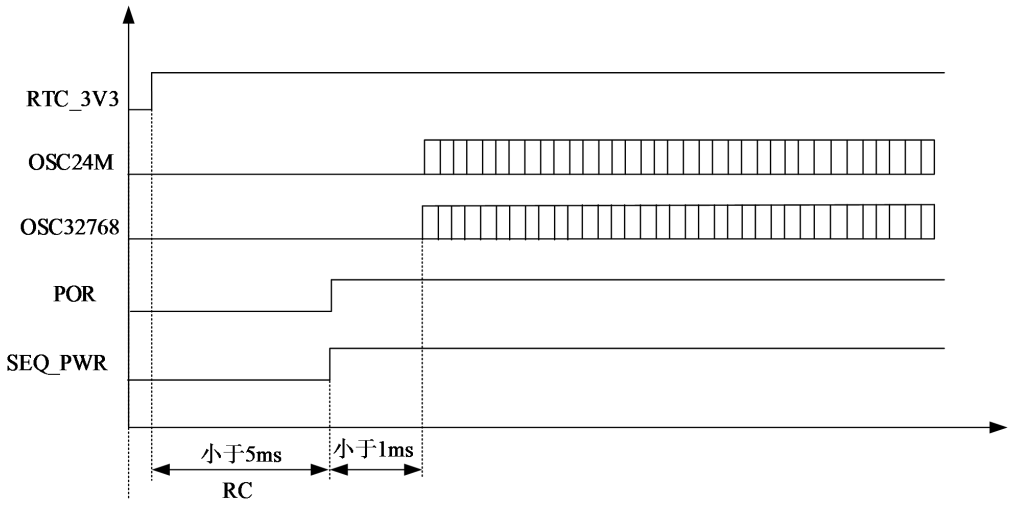
43.3 Thermal Information

Symbol	Parameter	Unit（℃/W）
Theta-Ja	Thermal resistance, junction to ambient(JEDEC PCB 4 Layer； Air flow=0m/s, Ambient 85℃)	31.156
Theta-Jc	Thermal resistance, junction to ambient(JEDEC PCB 4 Layer； Air flow=0m/s, Ambient 25℃)	4.820

***Note:**

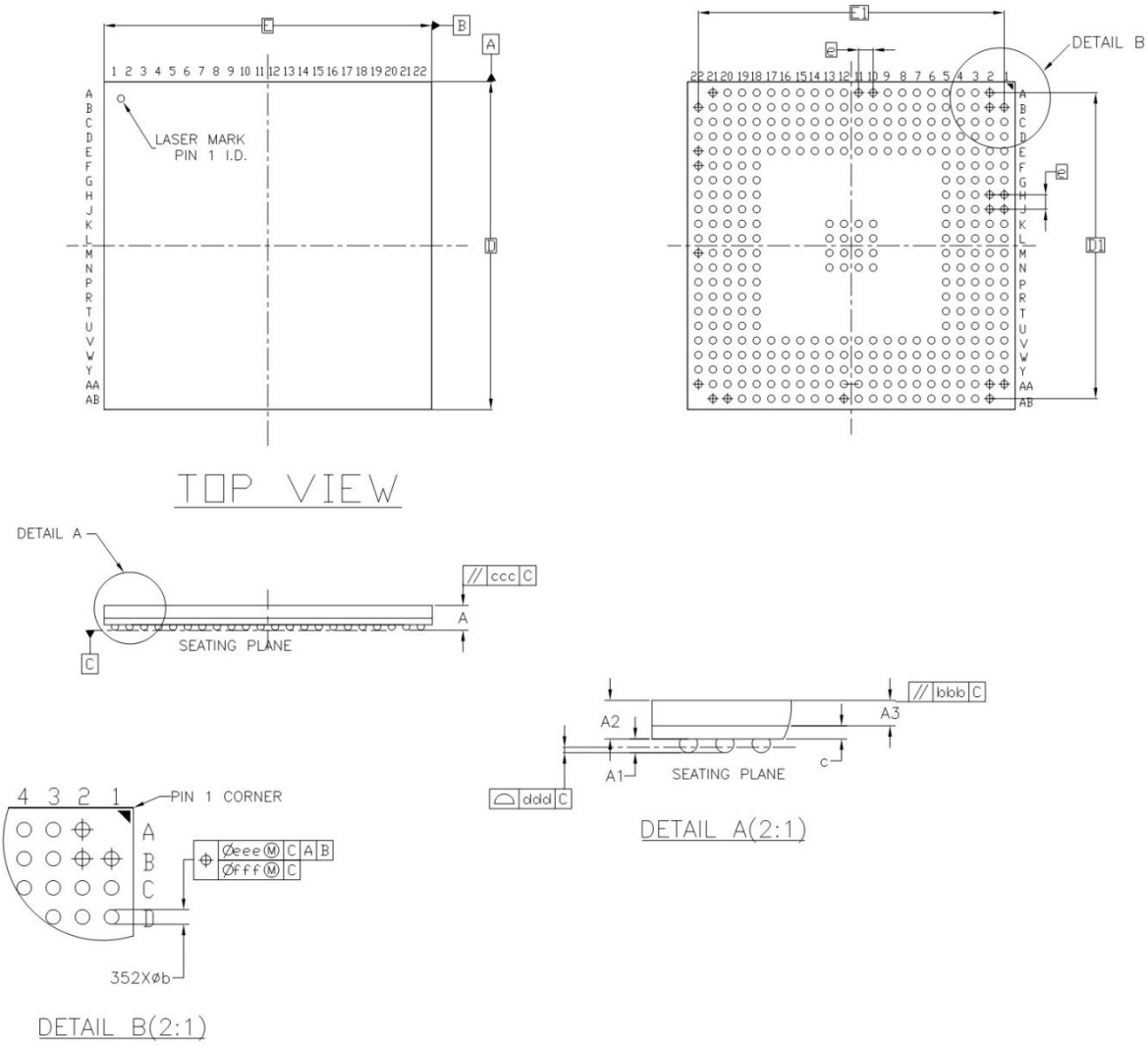
Under typical application condition， the power consumption is 1.28W, and the surface temperature rise of the chip is approximately 40℃. To ensure product reliability, it is recommended to add heat sinks.

43.4 Power Sequence



RTC第一次上电或RTC按键复位自动启动

44. Package Descriptions



SYMBOL	MILLIMETER		
	MIN	NOR	MAX
A	-	-	1.50
A1	0.25	0.30	0.35
A2	1.01	1.06	1.11
A3	0.70 BSC		
c	0.32	0.36	0.40
D	17.90	18.00	18.10
D1	16.80 BSC		
E	17.90	18.00	18.10
E1	16.80 BSC		
e	0.80 BSC		
L	0.35	0.40	0.45
aaa	0.15		
bbb	0.25		
ccc	0.20		
ddd	0.12		
eee	0.20		
fff	0.10		