SG2042 Technical Reference Manual

Nov 06, 2024

CONTENTS:

CHAPTER

ONE

ABSTRACT

SG2042 is server grade chip with high performance, low power consumption and high data throughput

1.1 Key features

- 64 RISC-V cpu cores which implements IMAFDC
- 4 cores per cluster, 16 clusters on chip
- RISC-V vector 0.7
- 2.0GHz CPU frequency
- 64KiB L1 I-Cache and 64KiB L1 D-Cache per core
- 1MiB unified L2 cache per cluster
- 64MiB system level L3 cache
- 512G ops/s for 8bit integer and 256G ops/s for 16bit floating point
- TDP 120W
- 4 DRAM controller, support DDR4 UDIMM/SODIMM/RDIMM up to 3200MT/s with ECC byte
- Max 256GiB DRAM with single chip, 256GiB with dual chips system
- 2 PCIe controller, support PCIe gen4 up to 16GT/s/lan. 32 lans in total.
- 1 1Gbps ethernet RGMII
- 2 eMMC/SDIO, support eMMC 5.1 or SDIO 3.0. 4bit data width
- 2 SPI flash interface
- 1 LPC
- 4 UART
- 4 I2C, support 100K/400K/1M clock frequency
- 2 general SPI controller
- 4 PWM generator for fan control
- 4 Fan speed counter
- 32 GPIO pins
- FCBGA, ball pitch 1mm, package size 57mm x 57mm

CHAPTER

TWO

SYSTEM

2.1 System architecture

SG2042 is a typical NoC(network-on-chip) architecture. All transactions are routed by the router in network. SoC architecture is shown in figure *Mesh architecture*

As you have seen, four CPUs are partitioned into one cluster, totally 16 clusters are connected into the mesh network. Each SLC(System Level Cache) is 4MiB in size totally 16 SLCs are connected. They are shared by all CPUs. Four DRAM controllers locate on the left and right side respectively. They can be accessed by all masters connected on the network.

SG2042 support 2 socket mode through CCIX ports on mesh. Each CCIX port bind to a PCIe controller. CCIX0 bind to PCIe0, CCIX1 bind to PCIe1. Customers can pick each of them for dual socket connection.

PCI device maps bars of PCI devices into SoC address space. PCI master nodes handle requests from PCI devices, like DMA transactions.

SCP(System CoProcessor) is a "out of mesh" CPU subsystem. it has no cache coherence with other CPUs in mesh network. Its responsibility is initilazing basic platform specific devices. Mesh network, DRAM controller, PCIe controller and so on.

2.2 Memory organization

SG2042 implements RISC-V Sv39 virtual address scheme with 40bits physicall addressing ability.

Program memory, data memory, registers and I/O ports are organized within the same linear 1TiB address space.

The bytes are coded in memory in Little Endian format. The lowest numbered byte in a word is considered the word's least significant byte and the highest numbered byte the most significant.

As SG2042 supports two way CPU technolog. When working at two way mode, the first CPU we naming it as CHIP0, the second CPU we naming it as CHIP1. All resources in CHIP0 are organized within the 512GiB address space(low 39bits). Resources in CHIP1 are organized from 512GiB to 1TiB address space(the most significiant bit of address is 1).

For example, CHIP0 PCIe0 Link0 slave address is located from 0x40_0000_0000 to 0x40_3000_0000. CHIP1 PCIe0 Link0 slave address is located from 0xc0_0000_0000 to 0xc0_3000_0000.

Detailed memory layout is show in table *Memory map*

Table 1: Memory map

Start Address	End Address	Devices	Memory Size
000:0000:0000	00F:FFFF:FFFF	DDR0	64G

continues on next page

Table 1 – continued from previous page

Start Address	End Address	Devices	Memory Size
010:0000:0000	01F:FFFF:FFFF	DDR1	64G
020:0000:0000	02F:FFFF:FFFF	DDR2	64G
030:0000:0000	03F:FFFF:FFFF	DDR3	64G
040:0000:0000	043:FFFF:FFFF	PCIE0_LINK0_SLAVE	16G
044:0000:0000	047:FFFF:FFFF	PCIE0_LINK1_SLAVE	16G
048:0000:0000	04B:FFFF:FFFF	PCIE1_LINK0_SLAVE	16G
04C:0000:0000	04F:FFFF:FFFF	PCIE1_LINK1_SLAVE	16G
070:0014:0000	070:0014:FFFF	SCP_ROM	64K
070:0018:0000	070:0117:FFFF	Serial_Flash0	16M
070:0218:0000	070:0317:FFFF	Serial_Flash1	16M
070:0800:0000	070:0FFF:FFFF	LPC	128M
070:1000:0000	070:100F:FFFF	SRAM0	1M
070:1010:0000	070:101F:FFFF	SRAM1	1M
070:3000:0000	070:3000:0FFF	EFUSE0	4K
070:3000:1000	070:3000:1FFF	EFUSE1	4K
070:3000:2000	070:3000:2FFF	RTC	4K
070:3000:3000	070:3000:3FFF	TIMER	4K
070:3000:4000	070:3000:4FFF	WDT	4K
070:3000:5000	070:3000:5FFF	I2C0	4K
070:3000:6000	070:3000:6FFF	I2C1	4K
070:3000:7000	070:3000:7FFF	I2C2	4K
070:3000:8000	070:3000:8FFF	I2C3	4K
070:3000:9000	070:3000:9FFF	GPIO0	4K
070:3000:A000	070:3000:AFFF	GPIO1	4K
070:3000:B000	070:3000:BFFF	GPIO2	4K
070:3000:C000	070:3000:CFFF	PWM	4K
070:3001:0000	070:3001:0FFF	SYS_CTRL	4K
070:3001:1000	070:3001:1FFF	PINMUX	4K
070:3001:2000	070:3001:2FFF	CLOCK	4K
070:3001:3000	070:3001:3FFF	RESET	4K
070:4000:0000	070:4000:0FFF	UART0	4K
070:4000:1000	070:4000:1FFF	UART1	4K
070:4000:2000	070:4000:2FFF	UART2	4K
070:4000:3000	070:4000:3FFF	UART3	4K
070:4000:4000	070:4000:4FFF	SPI0	4K
070:4000:5000	070:4000:5FFF	SPI1	4K
070:4000:6000	070:4001:5FFF	SYS_DMA	64K
070:4001:6000	070:4002:5FFF	HS_DMA	64K
070:4002:6000	070:4002:9FFF	ETH0	16K
070:4002:A000	070:4002:AFFF	EMMC0	4K
070:4002:B000	070:4002:BFFF	EMMC1	4K
070:400A:0000	070:4029:FFFF	TOP_Monitor	2M
070:402A:0000	070:4049:FFFF	HSPERI_Monitor	2M
070:5000:0000	070:51FF:FFFF	DDR0_CFG	32M
070:5200:0000	070:53FF:FFFF	DDR1_CFG	32M
070:5400:0000	070:55FF:FFFF	DDR2_CFG	32M
070:5600:0000	070:57FF:FFFF	DDR3_CFG	32M
070:6000:0000	070:61FF:FFFF	PCIE0_CFG	32M
070:6200:0000	070:63FF:FFFF	PCIE1_CFG	32M
070:7000:0000	070:7FFF:FFFF	MESH	256M

continues on next page

Table 1 – continued from previous page

Start Address	End Address	Devices	Memory Size
070:9000:0000	070:93FF:FFFF	PLIC	64M
070:9400:0000	070:97FF:FFFF	CLINT_IPI	64M
070:AC00:0000	070:AFFF:FFFF	CLINT_TIMER	64M

2.3 System coprocessor

SG2042 has two CPU subsystem, one is the main 64 cores RISC-V subsystem and the other is system coprocessor(SCP).

After chip power on, system boots from SCP. All RISC-V cores are stay in reset status. SCP will do some platform initialization, then release all 64 RISC-V cores. These platform initializations including:

- Setup PCIe topology. Set PCIe controll to a given mode. Link with PCIe devices.
- Setup DRAM by reading SPD through I2C bus.
- Setup mesh.
- Setup chip to chip CCIX link if dual socket mode is enabled.
- Load RISC-V zero stage bootloader(zsbl.bin)
- Setup RISC-V CPU reset address to where zsbl.bin is loaded.
- Release all RISC-V CPUs, now all CPUs run from zero stage bootloader.

So, RISC-V CPUs do not have a so called bootrom. Zero stage bootloader(zsbl.bin) is the first boot stage of RISC-V CPUs.

2.4 Boot

Boot sequency is controlled by both hardware and software

2.4.1 Power on reset

After power-on reset sequence, SCP reset will be automatically de-asserted by hardware. Customers can select boot devices by pull BOOT_SEL[1] up or down. RISC-V CPUs are asserted remain.

SG2042 provides 8-bit boot strap pins BOOT_SEL[7:0], the usage is shown as table *Boot select*

Table 2: Boot select

Pin	Detect by	Value	Description
BOOT_SEL0	Software	Recommend to 1	0: Disable SD card boot. 1: Try SD card boot first, then try SPI flash boot
BOOT_SEL1	Hard-ware	Recommend to 0	0: Boot from on-chip bootrom. 1: Bootrom from external SPI flash
BOOT_SEL2	Software	Recommend to 0	Enable SCP console
BOOT_SEL3	Software	Must be 0	Enter system level test mode
BOOT_SEL4-7	Reserved	Must be 0	Reserved, but must be pulled down

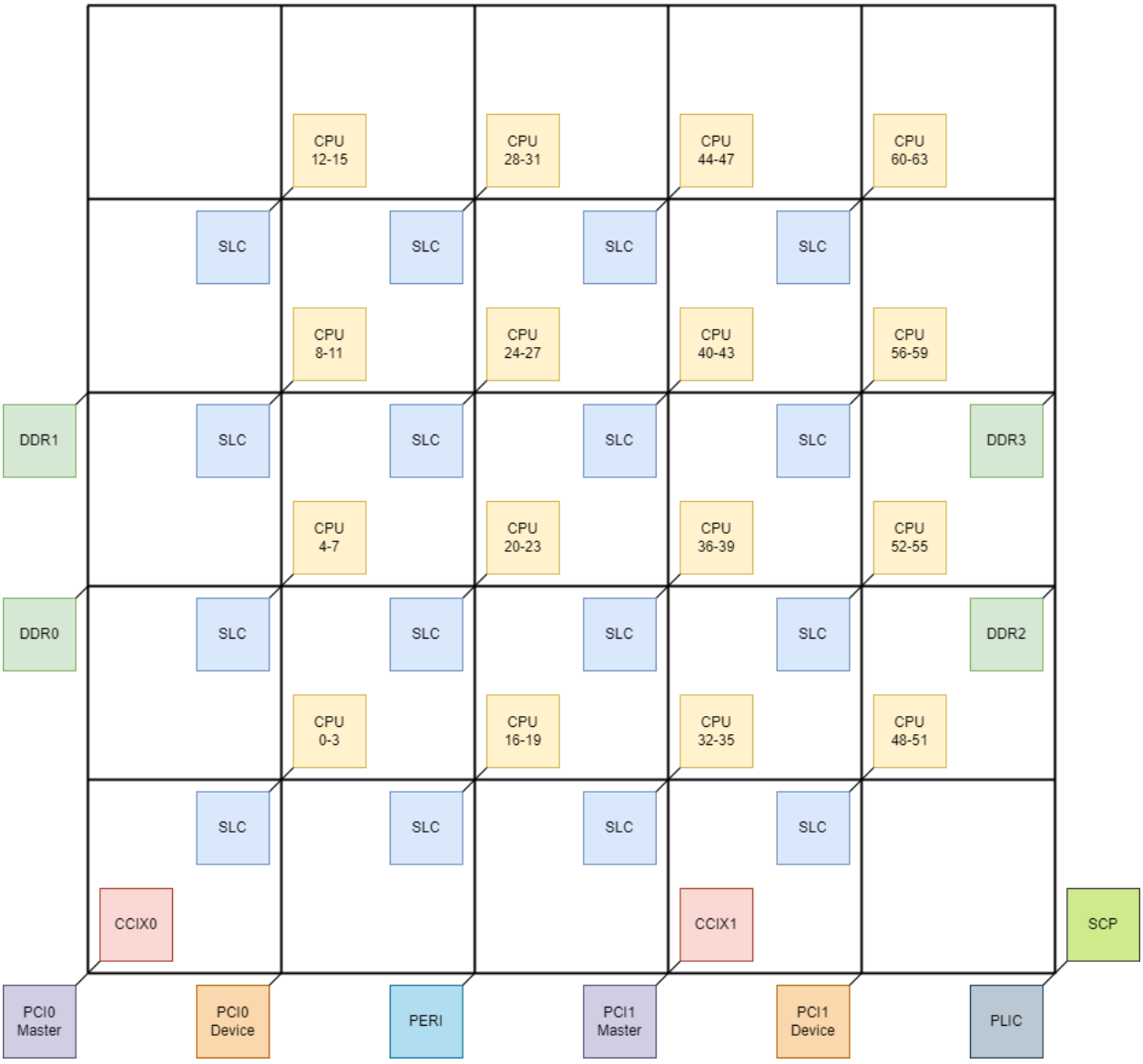

Fig. 1: Mesh architecture

When boot from SPI Flash, IO SPIF*_CLK_SEL1 and SPIF*_CLK_SEL0 are used to determine the clock frequency of SPI interface as shown in table *SPI flash clock selection*

Table 3: SPI flash clock selection

SPIx_CLK_SEL1	SPIx_CLK_SEL0	SPI clock frequency
0	0	2.5MHz
0	1	0.5MHz
1	0	10MHz
1	1	25MHz

For SPIx_CLK_SELy, x stands for SPI0 or SPI1, y stands for SEL0 and SEL1.

2.4.2 Bootrom

Bootrom supports loading SCP firmware from SPI flash or SD card.

When booting from SPI flash, bootrom loads SCP firmware from a given offset in flash. When booting from SD card, some restrictions are listed below:

- SD card MUST contain a partition table MBR format.
- The first partition MUST be formatted with FAT32 file system.
- SCP firmware MUST be named fip.bin.
- fip.bin MUST be put into the first partition.

Suggest partitioning and formatting SD card on linux based PC for compatible considerations. You can do it by following commands (assume your SD card's device file is /dev/sda):

```
$ sudo parted -s /dev/sda -- mklabel msdos mkpart primary fat32 1MiB -1s
$ sudo mkfs.vfat -F32 /dev/sda1
```

2.4.3 SCP firmware

SCP firmware loads RISC-V zero stage bootloader zsbl.bin from SPI or SD card.

SPI flash layout

TODO: add SPI flash layout

SCP firmware load zsbl.bin from the first partition of SD card. zsbl.bin should locate at the root of this partition.

CHAPTER

THREE

PIN MUX

SG2042 pins consist of digital pins, analog pins and power supply pins

3.1 Digital pins

3.1.1 Pin list

Digital pins Digital pins are listed in table *Digital pins*

Table 1: Digital pins

Signal Name	I/O	Voltage	Description	Speed MHz
LPC_LCLK	I	1.8	LPC Host Clock	50
LPC_LFRAME	I	1.8	LPC LFRAME	50
LPC_LAD0	I	1.8	LPC LAD	50
LPC_LAD1	I	1.8	LPC LAD	50
LPC_LAD2	I	1.8	LPC LAD	50
LPC_LAD3	I	1.8	LPC LAD	50
LPC_LDRQ0	I	1.8	LPC Encoded DMA/Bus Master Request	50
LPC_LDRQ1	I	1.8	LPC Encoded DMA/Bus Master Request	50
LPC_SERIRQ	I	1.8	LPC Serialized IRQ	50
LPC_CLKRUN	I	1.8	LPC CLKRUN	50
LPC_LPME	I	1.8	LPC LPME	50
LPC_LPCPD	I	1.8	LPC LPCPD	50
LPC_LSMI	I	1.8	LPC LSMI	50
PCIE0_L0_RESET_X	O	1.8	PCIe0 Link0 Reset	10
PCIE0_L1_RESET_X	O	1.8	PCIE0 Link1 Reset	10
PCIE0_L0_WAKEUP_X	O	1.8	PCIE0 Link0 Wakeup	10
PCIE0_L1_WAKEUP_X	O	1.8	PCIE0 Link1 Wakeup	10
PCIE0_L0_CLKREQ_IN_X	O	1.8	PCIE0 Link0 Clock Req	10
PCIE0_L1_CLKREQ_IN_X	O	1.8	PCIE0 Link1 Clock Req	10
PCIE1_L0_RESET_X	O	1.8	PCIE1 Link0 Reset	10
PCIE1_L1_RESET_X	O	1.8	PCIE1 Link1 Reset	10
PCIE1_L0_WAKEUP_X	O	1.8	PCIE1 Link0 Wakeup	10
PCIE1_L1_WAKEUP_X	O	1.8	PCIE1 Link1 Wakeup	10
PCIE1_L0_CLKREQ_IN_X	O	1.8	PCIE1 Link0 Clock Req	10
PCIE1_L1_CLKREQ_IN_X	O	1.8	PCIE1 Link1 Clock Req	10
SPIF0_CLK_SEL1	I	1.8	SPI0 flash clock select	50
SPIF0_CLK_SEL0	I	1.8	SPI0 flash clock select	50
SPIF0_WP_X	IO	1.8	SPI0 flash Write Protect	50

continues on next page

Table 1 – continued from previous page

Signal Name	I/O	Voltage	Description	Speed MHz
SPIF0_HOLD_X	IO	1.8	SPI0 flash Hold	50
SPIF0_SDI	IO	1.8	SPI0 flash data input	50
SPIF0_CS_X	O	1.8	SPI0 flash chip select	50
SPIF0_SCK	O	1.8	SPI0 flash clock	50
SPIF0_SDO	IO	1.8	SPI0 flash data output	50
SPIF1_CLK_SEL1	I	1.8	SPI1 flash clock select	50
SPIF1_CLK_SEL0	I	1.8	SPI1 flash clock select	50
SPIF1_WP_X	IO	1.8	SPI1 flash Write Protect	50
SPIF1_HOLD_X	IO	1.8	SPI1 flash Hold	50
SPIF1_SDI	IO	1.8	SPI1 flash data input	50
SPIF1_CS_X	O	1.8	SPI1 flash chip select	50
SPIF1_SCK	O	1.8	SPI1 flash clock	50
SPIF1_SDO	IO	1.8	SPI1 flash data output	50
EMMC_WP	IO	1.8	eMMC write protect signal	50
EMMC_CD_X	IO	1.8	eMMC card detect signal, low active	50
EMMC_RST_X	O	1.8	eMMC card reset signal	50
EMMC_PWR_EN	O	1.8	eMMC card power enable signal	50
SDIO_CD_X	I	1.8	SDIO card detect signal, low active	50
SDIO_WP	I	1.8	SDIO write protect signal	50
SDIO_RST_X	O	1.8	SDIO card reset signal	50
SDIO_PWR_EN	O	1.8	SDIO card power enable signal	50
RGMII0_TXD0	O	1.8	RGMII transmit data	250
RGMII0_TXD1	O	1.8	RGMII transmit data	250
RGMII0_TXD2	O	1.8	RGMII transmit data	250
RGMII0_TXD3	O	1.8	RGMII transmit data	250
RGMII0_TXCTRL	O	1.8	RGMII transmit control	250
RGMII0_RXD0	I	1.8	RGMII receive data	250
RGMII0_RXD1	I	1.8	RGMII receive data	250
RGMII0_RXD2	I	1.8	RGMII receive data	250
RGMII0_RXD3	I	1.8	RGMII receive data	250
RGMII0_RXCTRL	I	1.8	RGMII receive control	250
RGMII0_TXC	O	1.8	RGMII transmit clock	250
RGMII0_RXC	I	1.8	RGMII receive clock	250
RGMII0_REFCLKO	O	1.8	Reference clock output	250
RGMII0_IRQ	I	1.8	Interrupt request from PHY	250
RGMII0_MDC	O	1.8	RGMII management clock	250
RGMII0_MDIO	IO	1.8	RGMII management data IO	250
PWM0	O	1.8	Outputs of PWM0	10
PWM1	O	1.8	Outputs of PWM1	10
PWM2	O	1.8	Outputs of PWM2	10
PWM3	O	1.8	Outputs of PWM3	10
FAN0	IO	1.8	Outputs of FAN0	10
FAN1	IO	1.8	Outputs of FAN1	10
FAN2	IO	1.8	Outputs of FAN2	10
FAN3	IO	1.8	Outputs of FAN3	10
IIC0_SDA	IO	1.8	IIC0 SDA	10
IIC0_SCL	IO	1.8	IIC0 SCL	10
IIC1_SDA	IO	1.8	IIC1 SDA	10
IIC1_SCL	IO	1.8	IIC1 SCL	10
IIC2_SDA	IO	1.8	IIC2 SDA	10

continues on next page

Table 1 – continued from previous page

Signal Name	I/O	Voltage	Description	Speed MHz
IIC2_SCL	IO	1.8	IIC2 SCL	10
IIC3_SDA	IO	1.8	IIC3 SDA	10
IIC3_SCL	IO	1.8	IIC3 SCL	10
UART0_TX	O	1.8	UART0 transmit data	10
UART0_RX	I	1.8	UART0 receive data	10
UART0_RTS	O	1.8	UART0 RTS	10
UART0_CTS	I	1.8	UART0 CTS	10
UART1_TX	O	1.8	UART1 transmit data	10
UART1_RX	I	1.8	UART1 receive data	10
UART1_RTS	O	1.8	UART1 RTS	10
UART1_CTS	I	1.8	UART1 CTS	10
UART2_TX	IO	1.8	UART2 transmit data	10
UART2_RX	IO	1.8	UART2 receive data	10
UART2_RTS	O	1.8	UART2 RTS	10
UART2_CTS	I	1.8	UART2 CTS	10
UART3_TX	IO	1.8	UART3 transmit data	10
UART3_RX	IO	1.8	UART3 receive data	10
UART3_RTS	O	1.8	UART3 RTS	10
UART3_CTS	I	1.8	UART3 CTS	10
SPI0_CS0_X	O	1.8	SPI0 CS0	50
SPI0_CS1_X	O	1.8	SPI0 CS1	50
SPI0_SDI	I	1.8	SPI0 SDI	50
SPI0_SDO	IO	1.8	SPI0 SDO	50
SPI0_SCK	O	1.8	SPI0 SCK	50
SPI1_CS0_X	O	1.8	SPI1 CS0	50
SPI1_CS1_X	O	1.8	SPI1 CS1	50
SPI1_SDI	I	1.8	SPI1 SDI	50
SPI1_SDO	IO	1.8	SPI1 SDO	50
SPI1_SCK	O	1.8	SPI1 SCK	50
JTAG0_TDO	IO	1.8	JTAG0 TDO	50
JTAG0_TCK	IO	1.8	JTAG0 TCK	50
JTAG0_TDI	IO	1.8	JTAG0 TDI	50
JTAG0_TMS	IO	1.8	JTAG0 TMS	50
JTAG0_TRST_X	IO	1.8	JTAG0 TRST	50
JTAG0_SRST_X	IO	1.8	JTAG0 SRST	50
JTAG1_TDO	IO	1.8	JTAG1 TDO	50
JTAG1_TCK	IO	1.8	JTAG1 TCK	50
JTAG1_TDI	IO	1.8	JTAG1 TDI	50
JTAG1_TMS	IO	1.8	JTAG1 TMS	50
JTAG1_TRST_X	IO	1.8	JTAG1 TRST	50
JTAG1_SRST_X	IO	1.8	JTAG1 SRST	50
JTAG2_TDO	IO	1.8	JTAG2 TDO (for DFT)	50
JTAG2_TCK	IO	1.8	JTAG2 TCK	50
JTAG2_TDI	IO	1.8	JTAG2 TDI	50
JTAG2_TMS	IO	1.8	JTAG2 TMS	50
JTAG2_TRST_X	IO	1.8	JTAG2 TRST	50
JTAG2_SRST_X	IO	1.8	JTAG2 SRST	50
GPIO0	IO	1.8	GPIO0	10
GPIO1	IO	1.8	GPIO1	10
GPIO2	IO	1.8	GPIO2	10

continues on next page

Table 1 – continued from previous page

Signal Name	I/O	Voltage	Description	Speed MHz
GPIO3	IO	1.8	GPIO3	10
GPIO4	IO	1.8	GPIO4	10
GPIO5	IO	1.8	GPIO5	10
GPIO6	IO	1.8	GPIO6	10
GPIO7	IO	1.8	GPIO7	10
GPIO8	IO	1.8	GPIO8	10
GPIO9	IO	1.8	GPIO9	10
GPIO10	IO	1.8	GPIO10	10
GPIO11	IO	1.8	GPIO11	10
GPIO12	IO	1.8	GPIO12	10
GPIO13	IO	1.8	GPIO13	10
GPIO14	IO	1.8	GPIO14	10
GPIO15	IO	1.8	GPIO15	10
GPIO16	IO	1.8	GPIO16	10
GPIO17	IO	1.8	GPIO17	50
GPIO18	IO	1.8	GPIO18	50
GPIO19	IO	1.8	GPIO19	50
GPIO20	IO	1.8	GPIO20	50
GPIO21	IO	1.8	GPIO21	50
GPIO22	IO	1.8	GPIO22	50
GPIO23	IO	1.8	GPIO23	10
GPIO24	IO	1.8	GPIO24	10
GPIO25	IO	1.8	GPIO25	10
GPIO26	IO	1.8	GPIO26	10
GPIO27	IO	1.8	GPIO27	10
GPIO28	IO	1.8	GPIO28	10
GPIO29	IO	1.8	GPIO29	10
GPIO30	IO	1.8	GPIO30	10
GPIO31	IO	1.8	GPIO31	10
MODE_SEL0	I	1.8	Mode Select0	10
MODE_SEL1	I	1.8	Mode Select1	10
MODE_SEL2	I	1.8	Mode Select2	10
BOOT_SEL0	I	1.8	Boot Select0	10
BOOT_SEL1	I	1.8	Boot Select1	10
BOOT_SEL2	I	1.8	Boot Select2	10
BOOT_SEL3	I	1.8	Boot Select3	10
BOOT_SEL4	I	1.8	Boot Select4	10
BOOT_SEL5	I	1.8	Boot Select5	10
BOOT_SEL6	I	1.8	Boot Select6	10
BOOT_SEL7	I	1.8	Boot Select7	10
MULTI_SCKT	I	1.8	Multi-socket Enable	10
SCKT_ID0	I	1.8	Socket ID0	10
SCKT_ID1	I	1.8	Socket ID1	10
PLL_CLK_IN_MAIN	I	1.8	PLL reference clock input	50
PLL_CLK_IN_DDR_L	I	1.8	DPLL (Left) reference clock input	50
PLL_CLK_IN_DDR_R	I	1.8	DPLL (right) reference clock input	50
XTAL_32K	I	1.8	XTAL 32K clock input	10
SYS_RST_X	I	1.8	System Reset, active-low	10
PWR_BUTTON	I	1.8	System Reset, active-low	10
TEST_EN	I	1.8	TEST Mode Enable	10

continues on next page

Table 1 – continued from previous page

Signal Name	I/O	Voltage	Description	Speed MHz
TEST_MODE_MBIST	I	1.8	TEST MBIST Mode	10
TEST_MODE_SCAN	I	1.8	TEST Scan Mode	10
TEST_MODE_BSD	I	1.8	TEST BSD Mode	10
BISR_BYP	I	1.8	BISR Bypass	10

3.1.2 Pin function

Most digital pins have multiple functions. Each digital pin can have at most four functions. Pins and functions are listed in table *Digital pin functions* The letter in parentheses means, I: input, O: output, IO: input and output.

Table 2: Digital pin functions

Signal Name	Function0	Function1	Function2	Function3
LPC_LCLK	LPC_LCLK(O)	GPIO32(IO)		
LPC_LFRAME	LPC_LFRAME(O)	GPIO33(IO)		
LPC_LAD0	LPC_LAD0(IO)	GPIO34(IO)		
LPC_LAD1	LPC_LAD1(IO)	GPIO35(IO)		
LPC_LAD2	LPC_LAD2(IO)	GPIO36(IO)		
LPC_LAD3	LPC_LAD3(IO)	GPIO37(IO)		
LPC_LDRQ0	LPC_DRQ0(I)	GPIO38(IO)		
LPC_LDRQ1	LPC_DRQ1(I)	GPIO39(IO)		
LPC_SERIRQ	LPC_SERIRQ(IO)	GPIO40(IO)		
LPC_CLKRUN	LPC_CLKRUN(IO)	GPIO41(IO)		
LPC_LPME	LPC_LPME(IO)	GPIO42(IO)		
LPC_LPCPD	LPC_LPCPD(O)	GPIO43(IO)		
LPC_LSMI	LPC_LSMI(I)	GPIO44(IO)		
PCIE0_L0_RESET_X	PCIE0_L0_RESET_X(I)			
PCIE0_L1_RESET_X	PCIE0_L1_RESET_X(I)			
PCIE0_L0_WAKEUP_X	PCIE0_L0_WAKEUP_X(IO)			
PCIE0_L1_WAKEUP_X	PCIE0_L1_WAKEUP_X(IO)			
PCIE0_L0_CLKREQ_IN_X	PCIE0_L0_CLKREQ_IN_X(IO)			
PCIE0_L1_CLKREQ_IN_X	PCIE0_L1_CLKREQ_IN_X(IO)			
PCIE1_L0_RESET_X	PCIE1_L0_RESET_X(I)			
PCIE1_L1_RESET_X	PCIE1_L1_RESET_X(I)			
PCIE1_L0_WAKEUP_X	PCIE1_L0_WAKEUP_X(IO)			
PCIE1_L1_WAKEUP_X	PCIE1_L1_WAKEUP_X(IO)			
PCIE1_L0_CLKREQ_IN_X	PCIE1_L0_CLKREQ_IN_X(IO)			
PCIE1_L1_CLKREQ_IN_X	PCIE1_L1_CLKREQ_IN_X(IO)			
SPIF0_CLK_SEL1	SPIF0_CLK_SEL1(I)			
SPIF0_CLK_SEL0	SPIF0_CLK_SEL0(I)			
SPIF0_WP_X	SPIF0_WP_X(IO)			
SPIF0_HOLD_X	SPIF0_HOLD_X(IO)			
SPIF0_SDI	SPIF0_SDI(IO)			
SPIF0_CS_X	SPIF0_CS_X(O)			
SPIF0_SCK	SPIF0_SCK(O)			
SPIF0_SDO	SPIF0_SDO(IO)			
SPIF1_CLK_SEL1	SPIF1_CLK_SEL1(I)			
SPIF1_CLK_SEL0	SPIF1_CLK_SEL0(I)			
SPIF1_WP_X	SPIF1_WP_X(IO)			
SPIF1_HOLD_X	SPIF1_HOLD_X(IO)			

continues on next page

Table 2 – continued from previous page

Signal Name	Function0	Function1	Function2	Function3
SPIF1_SDI	SPIF1_SDI(IO)			
SPIF1_CS_X	SPIF1_CS_X(O)			
SPIF1_SCK	SPIF1_SCK(O)			
SPIF1_SDO	SPIF1_SDO(IO)			
EMMC_WP	EMMC_WP(I)			
EMMC_CD_X	EMMC_CD_X(I)			
EMMC_RST_X	EMMC_RST_X(O)			
EMMC_PWR_EN	EMMC_PWR_EN(O)			
SDIO_CD_X	SDIO_CD_X(I)			
SDIO_WP	SDIO_WP(I)			
SDIO_RST_X	SDIO_RST_X(O)			
SDIO_PWR_EN	SDIO_PWR_EN(O)			
RGMII0_TXD0	RGMII0_TXD0(O)			
RGMII0_TXD1	RGMII0_TXD1(O)			
RGMII0_TXD2	RGMII0_TXD2(O)			
RGMII0_TXD3	RGMII0_TXD3(O)			
RGMII0_TXCTRL	RGMII0_TXCTRL(O)			
RGMII0_RXD0	RGMII0_RXD0(I)			
RGMII0_RXD1	RGMII0_RXD1(I)			
RGMII0_RXD2	RGMII0_RXD2(I)			
RGMII0_RXD3	RGMII0_RXD3(I)			
RGMII0_RXCTRL	RGMII0_RXCTRL(I)			
RGMII0_TXC	RGMII0_TXC(O)			
RGMII0_RXC	RGMII0_RXC(I)			
RGMII0_REFCLKO	RGMII0_REFCLKO(O)			
RGMII0_IRQ	RGMII0_IRQ(I)			
RGMII0_MDC	RGMII0_MDC(O)			
RGMII0_MDIO	RGMII0_MDIO(IO)			
PWM0	PWM0(O)			
PWM1	PWM1(O)			
PWM2	PWM2(O)	GPIO45(IO)		
PWM3	PWM3(O)	GPIO46(IO)		
FAN0	FAN0(I)			
FAN1	FAN1(I)			
FAN2	FAN2(I)	GPIO47(IO)		
FAN3	FAN3(I)	GPIO48(IO)		
IIC0_SDA	IIC0_SDA(IO)	GPIO49(IO)		
IIC0_SCL	IIC0_SCL(O)	GPIO50(IO)		
IIC1_SDA	IIC1_SDA(IO)	GPIO51(IO)		
IIC1_SCL	IIC1_SCL(O)	GPIO52(IO)		
IIC2_SDA	IIC2_SDA(IO)	GPIO53(IO)		
IIC2_SCL	IIC2_SCL(O)	GPIO54(IO)		
IIC3_SDA	IIC3_SDA(IO)	GPIO55(IO)		
IIC3_SCL	IIC3_SCL(O)	GPIO56(IO)		
UART0_TX	UART0_TX(O)			
UART0_RX	UART0_RX(I)			
UART0_RTS	UART0_RTS(O)	GPIO57(IO)		
UART0_CTS	UART0_CTS(I)	GPIO58(IO)		
UART1_TX	UART1_TX(O)			
UART1_RX	UART1_RX(I)			

continues on next page

Table 2 – continued from previous page

Signal Name	Function0	Function1	Function2	Function3
UART1_RTS	UART1_RTS(O)	GPIO59(IO)		
UART1_CTS	UART1_CTS(I)	GPIO60(IO)		
UART2_TX	UART2_TX(O)			
UART2_RX	UART2_RX(I)			
UART2_RTS	UART2_RTS(O)	GPIO61(IO)		
UART2_CTS	UART2_CTS(I)	GPIO62(IO)		
UART3_TX	UART3_TX(O)			
UART3_RX	UART3_RX(I)			
UART3_RTS	UART3_RTS(O)	GPIO63(IO)		
UART3_CTS	UART3_CTS(I)	GPIO64(IO)		
SPI0_CS0_X	SPI0_CS0_X(O)	GPIO65(IO)		
SPI0_CS1_X	SPI0_CS1_X(O)	GPIO66(IO)		
SPI0_SDI	SPI0_SDI(I)	GPIO67(IO)		
SPI0_SDO	SPI0_SDO(IO)	GPIO68(IO)		
SPI0_SCK	SPI0_SCK(O)	GPIO69(IO)		
SPI1_CS0_X	SPI1_CS0_X(O)	GPIO70(IO)		
SPI1_CS1_X	SPI1_CS1_X(O)	GPIO71(IO)		
SPI1_SDI	SPI1_SDI(I)	GPIO72(IO)		
SPI1_SDO	SPI1_SDO(IO)	GPIO73(IO)		
SPI1_SCK	SPI1_SCK(O)	GPIO74(IO)		
JTAG0_TDO	JTAG0_TDO(IO)	GPIO75(IO)		
JTAG0_TCK	JTAG0_TCK(I)	GPIO76(IO)		
JTAG0_TDI	JTAG0_TDI(I)	GPIO77(IO)		
JTAG0_TMS	JTAG0_TMS(I)	GPIO78(IO)		
JTAG0_TRST_X	JTAG0_TRST_X(I)	GPIO79(IO)		
JTAG0_SRST_X	JTAG0_SRST_X(I)	GPIO80(IO)		
JTAG1_TDO	JTAG1_TDO(O)	GPIO81(IO)		
JTAG1_TCK	JTAG1_TCK(I)	GPIO82(IO)		
JTAG1_TDI	JTAG1_TDI(I)	GPIO83(IO)		
JTAG1_TMS	JTAG1_TMS(I)	GPIO84(IO)		
JTAG1_TRST_X	JTAG1_TRST_X(I)	GPIO85(IO)		
JTAG1_SRST_X	JTAG1_SRST_X(I)	GPIO86(IO)		
JTAG2_TDO	JTAG2_TDO(O)	GPIO87(IO)		
JTAG2_TCK	JTAG2_TCK(I)	GPIO88(IO)		
JTAG2_TDI	JTAG2_TDI(I)	GPIO89(IO)		
JTAG2_TMS	JTAG2_TMS(I)	GPIO90(IO)		
JTAG2_TRST_X	JTAG2_TRST_X(I)	GPIO91(IO)		
JTAG2_SRST_X	JTAG2_SRST_X(I)	GPIO92(IO)		
GPIO0	GPIO0(IO)		DEBUG_0(O)	
GPIO1	GPIO1(IO)		DEBUG_1(O)	
GPIO2	GPIO2(IO)		DEBUG_2(O)	
GPIO3	GPIO3(IO)		DEBUG_3(O)	
GPIO4	PLL_LOCKO(O)	GPIO4(IO)	DEBUG_4(O)	
GPIO5	GPIO5(IO)		DEBUG_5(O)	
GPIO6	GPIO6(IO)		DEBUG_6(O)	
GPIO7	GPIO7(IO)		DEBUG_7(O)	
GPIO8	GPIO8(IO)		DEBUG_8(O)	
GPIO9	GPIO9(IO)		DEBUG_9(O)	
GPIO10	GPIO10(IO)		DEBUG_10(O)	
GPIO11	GPIO11(IO)		DEBUG_11(O)	

continues on next page

Table 2 – continued from previous page

Signal Name	Function0	Function1	Function2	Function3
GPIO12	GPIO12(IO)		DEBUG_12(O)	
GPIO13	GPIO13(IO)		DEBUG_13(O)	
GPIO14	GPIO14(IO)		DEBUG_14(O)	
GPIO15	GPIO15(IO)		DEBUG_15(O)	
GPIO16	GPIO16(IO)		DEBUG_16(O)	
GPIO17	GPIO17(IO)		DEBUG_17(O)	
GPIO18	GPIO18(IO)		DEBUG_18(O)	
GPIO19	GPIO19(IO)		DEBUG_19(O)	
GPIO20	GPIO20(IO)		DEBUG_20(O)	
GPIO21	GPIO21(IO)		DEBUG_21(O)	
GPIO22	GPIO22(IO)		DEBUG_22(O)	
GPIO23	GPIO23(IO)		DEBUG_23(O)	
GPIO24	GPIO24(IO)		DEBUG_24(O)	
GPIO25	GPIO25(IO)		DEBUG_25(O)	
GPIO26	GPIO26(IO)		DEBUG_26(O)	
GPIO27	GPIO27(IO)		DEBUG_27(O)	
GPIO28	GPIO28(IO)		DEBUG_28(O)	
GPIO29	DBG_I2C_SCL(I)	GPIO29(IO)	DEBUG_29(O)	
GPIO30	DBG_I2C_SDA(IO)	GPIO30(IO)	DEBUG_30(O)	
GPIO31	DBG_I2C_SDA_OE(O)	GPIO31(IO)	DEBUG_31(O)	
MODE_SEL0	MODE_SEL0(I)			
MODE_SEL1	MODE_SEL1(I)			
MODE_SEL2	MODE_SEL2(I)			
BOOT_SEL0	BOOT_SEL0(I)			
BOOT_SEL1	BOOT_SEL1(I)			
BOOT_SEL2	BOOT_SEL2(I)			
BOOT_SEL3	BOOT_SEL3(I)			
BOOT_SEL4	BOOT_SEL4(I)			
BOOT_SEL5	BOOT_SEL5(I)			
BOOT_SEL6	BOOT_SEL6(I)			
BOOT_SEL7	BOOT_SEL7(I)			
MULTI_SCKT	MULTI_SCKT(I)			
SCKT_ID0	SCKT_ID0(I)			
SCKT_ID1	SCKT_ID1(I)			
PLL_CLK_IN_MAIN	PLL_CLK_IN_MAIN(I)			
PLL_CLK_IN_DDR_L	PLL_CLK_IN_DDR_L(I)			
PLL_CLK_IN_DDR_R	PLL_CLK_IN_DDR_R(I)			
XTAL_32K	XTAL_32K(I)			
SYS_RST_X	SYS_RST_X(I)			
PWR_BUTTON	PWR_BUTTON(I)			
TEST_EN	TEST_EN(I)			
TEST_MODE_MBIST	TEST_MODE_MBIST(I)			
TEST_MODE_SCAN	TEST_MODE_SCAN(I)			
TEST_MODE_BSD	TEST_MODE_BSD(I)			
BISR_BYP	BISR_BYP(I)			

3.1.3 Registers

Pinmux module controls attributes of pins. These attributes including function, pull up, pull down or no pull, if schmitt trigger is enabled and driving strength.

The base address is listed in table *Memory map*, PINMUX device.

REG00: offset 0x0000

Table 3: LPC_LCLK and LPC_LFRAME

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for LPC_LFRAME. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for LPC_LFRAME. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for LPC_LFRAME
21:20	RW	0x0	Pin Mux Selector for LPC_LFRAME
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for LPC_LFRAME. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for LPC_LFRAME.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for LPC_LCLK. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for LPC_LCLK. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for LPC_LCLK
5:4	RW	0x0	Pin Mux Selector for LPC_LCLK
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for LPC_LCLK. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for LPC_LCLK.

REG01: offset 0x0004

Table 4: LPC_LAD0 and LPC_LAD1

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for LPC_LAD1. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for LPC_LAD1. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for LPC_LAD1
21:20	RW	0x0	Pin Mux Selector for LPC_LAD1
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for LPC_LAD1. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for LPC_LAD1.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for LPC_LAD0. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for LPC_LAD0. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for LPC_LAD0
5:4	RW	0x0	Pin Mux Selector for LPC_LAD0
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for LPC_LAD0. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for LPC_LAD0.

REG02: offset 0x0008

Table 5: LPC_LAD2 and LPC_LAD3

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for LPC_LAD3. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for LPC_LAD3. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for LPC_LAD3
21:20	RW	0x0	Pin Mux Selector for LPC_LAD3
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for LPC_LAD3. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for LPC_LAD3.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for LPC_LAD2. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for LPC_LAD2. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for LPC_LAD2
5:4	RW	0x0	Pin Mux Selector for LPC_LAD2
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for LPC_LAD2. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for LPC_LAD2.

REG03: offset 0x000c

Table 6: LPC_LDRQ0 and LPC_LDRQ1

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for LPC_LDRQ1. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for LPC_LDRQ1. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for LPC_LDRQ1
21:20	RW	0x0	Pin Mux Selector for LPC_LDRQ1
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for LPC_LDRQ1. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for LPC_LDRQ1.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for LPC_LDRQ0. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for LPC_LDRQ0. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for LPC_LDRQ0
5:4	RW	0x0	Pin Mux Selector for LPC_LDRQ0
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for LPC_LDRQ0. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for LPC_LDRQ0.

REG04: offset 0x0010

Table 7: LPC_SERIRQ and LPC_CLKRUN

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for LPC_CLKRUN. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for LPC_CLKRUN. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for LPC_CLKRUN
21:20	RW	0x0	Pin Mux Selector for LPC_CLKRUN
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for LPC_CLKRUN. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for LPC_CLKRUN.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for LPC_SERIRQ. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for LPC_SERIRQ. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for LPC_SERIRQ
5:4	RW	0x0	Pin Mux Selector for LPC_SERIRQ
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for LPC_SERIRQ. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for LPC_SERIRQ.

REG05: offset 0x0014

Table 8: LPC_LPME and LPC_LPCPD

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for LPC_LPCPD. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for LPC_LPCPD. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for LPC_LPCPD
21:20	RW	0x0	Pin Mux Selector for LPC_LPCPD
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for LPC_LPCPD. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for LPC_LPCPD.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for LPC_LPME. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for LPC_LPME. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for LPC_LPME
5:4	RW	0x0	Pin Mux Selector for LPC_LPME
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for LPC_LPME. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for LPC_LPME.

REG06: offset 0x0018

Table 9: LPC_LSMI and PCIE0_L0_RESET_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for PCIE0_L0_RESET_X. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for PCIE0_L0_RESET_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for PCIE0_L0_RESET_X
21:20	RW	0x0	Pin Mux Selector for PCIE0_L0_RESET_X
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for PCIE0_L0_RESET_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for PCIE0_L0_RESET_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for LPC_LSMI. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for LPC_LSMI. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for LPC_LSMI
5:4	RW	0x0	Pin Mux Selector for LPC_LSMI
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for LPC_LSMI. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for LPC_LSMI.

REG07: offset 0x001c

Table 10: PCIE0_L1_RESET_X and PCIE0_L0_WAKEUP_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for PCIE0_L0_WAKEUP_X. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for PCIE0_L0_WAKEUP_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for PCIE0_L0_WAKEUP_X
21:20	RW	0x0	Pin Mux Selector for PCIE0_L0_WAKEUP_X
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for PCIE0_L0_WAKEUP_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for PCIE0_L0_WAKEUP_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for PCIE0_L1_RESET_X. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for PCIE0_L1_RESET_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for PCIE0_L1_RESET_X
5:4	RW	0x0	Pin Mux Selector for PCIE0_L1_RESET_X
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for PCIE0_L1_RESET_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for PCIE0_L1_RESET_X.

REG08: offset 0x0020

Table 11: PCIE0_L1_WAKEUP_X and PCIE0_L0_CLKREQ_IN_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for PCIE0_L0_CLKREQ_IN_X. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for PCIE0_L0_CLKREQ_IN_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for PCIE0_L0_CLKREQ_IN_X
21:20	RW	0x0	Pin Mux Selector for PCIE0_L0_CLKREQ_IN_X
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for PCIE0_L0_CLKREQ_IN_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for PCIE0_L0_CLKREQ_IN_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for PCIE0_L1_WAKEUP_X. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for PCIE0_L1_WAKEUP_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for PCIE0_L1_WAKEUP_X
5:4	RW	0x0	Pin Mux Selector for PCIE0_L1_WAKEUP_X
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for PCIE0_L1_WAKEUP_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for PCIE0_L1_WAKEUP_X.

REG09: offset 0x0024

Table 12: PCIE0_L1_CLKREQ_IN_X and PCIE1_L0_RESET_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for PCIE1_L0_RESET_X. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for PCIE1_L0_RESET_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for PCIE1_L0_RESET_X
21:20	RW	0x0	Pin Mux Selector for PCIE1_L0_RESET_X
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for PCIE1_L0_RESET_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for PCIE1_L0_RESET_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for PCIE0_L1_CLKREQ_IN_X. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for PCIE0_L1_CLKREQ_IN_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for PCIE0_L1_CLKREQ_IN_X
5:4	RW	0x0	Pin Mux Selector for PCIE0_L1_CLKREQ_IN_X
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for PCIE0_L1_CLKREQ_IN_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for PCIE0_L1_CLKREQ_IN_X.

REG0a: offset 0x0028

Table 13: PCIE1_L1_RESET_X and PCIE1_L0_WAKEUP_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for PCIE1_L0_WAKEUP_X. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for PCIE1_L0_WAKEUP_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for PCIE1_L0_WAKEUP_X
21:20	RW	0x0	Pin Mux Selector for PCIE1_L0_WAKEUP_X
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for PCIE1_L0_WAKEUP_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for PCIE1_L0_WAKEUP_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for PCIE1_L1_RESET_X. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for PCIE1_L1_RESET_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for PCIE1_L1_RESET_X
5:4	RW	0x0	Pin Mux Selector for PCIE1_L1_RESET_X
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for PCIE1_L1_RESET_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for PCIE1_L1_RESET_X.

REG0b: offset 0x002c

Table 14: PCIE1_L1_WAKEUP_X and PCIE1_L0_CLKREQ_IN_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for PCIE1_L0_CLKREQ_IN_X. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for PCIE1_L0_CLKREQ_IN_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for PCIE1_L0_CLKREQ_IN_X
21:20	RW	0x0	Pin Mux Selector for PCIE1_L0_CLKREQ_IN_X
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for PCIE1_L0_CLKREQ_IN_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for PCIE1_L0_CLKREQ_IN_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for PCIE1_L1_WAKEUP_X. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for PCIE1_L1_WAKEUP_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for PCIE1_L1_WAKEUP_X
5:4	RW	0x0	Pin Mux Selector for PCIE1_L1_WAKEUP_X
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for PCIE1_L1_WAKEUP_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for PCIE1_L1_WAKEUP_X.

REG0c: offset 0x0030

Table 15: PCIE1_L1_CLKREQ_IN_X and SPIF0_CLK_SEL1

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPIF0_CLK_SEL1. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for SPIF0_CLK_SEL1. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPIF0_CLK_SEL1
21:20	RW	0x0	Pin Mux Selector for SPIF0_CLK_SEL1
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPIF0_CLK_SEL1. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPIF0_CLK_SEL1.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for PCIE1_L1_CLKREQ_IN_X. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for PCIE1_L1_CLKREQ_IN_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for PCIE1_L1_CLKREQ_IN_X
5:4	RW	0x0	Pin Mux Selector for PCIE1_L1_CLKREQ_IN_X
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for PCIE1_L1_CLKREQ_IN_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for PCIE1_L1_CLKREQ_IN_X.

REG0d: offset 0x0034

Table 16: SPIF0_CLK_SEL0 and SPIF0_WP_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPIF0_WP_X. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for SPIF0_WP_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPIF0_WP_X
21:20	RW	0x0	Pin Mux Selector for SPIF0_WP_X
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPIF0_WP_X. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPIF0_WP_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPIF0_CLK_SEL0. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for SPIF0_CLK_SEL0. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPIF0_CLK_SEL0
5:4	RW	0x0	Pin Mux Selector for SPIF0_CLK_SEL0
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPIF0_CLK_SEL0. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPIF0_CLK_SEL0.

REG0e: offset 0x0038

Table 17: SPIF0_HOLD_X and SPIF0_SDI

Fields	Type	De-fault	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPIF0_SDI. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for SPIF0_SDI. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPIF0_SDI
21:20	RW	0x0	Pin Mux Selector for SPIF0_SDI
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPIF0_SDI. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPIF0_SDI.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPIF0_HOLD_X. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for SPIF0_HOLD_X. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPIF0_HOLD_X
5:4	RW	0x0	Pin Mux Selector for SPIF0_HOLD_X
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPIF0_HOLD_X. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPIF0_HOLD_X.

REG0f: offset 0x003c

Table 18: SPIF0_CS_X and SPIF0_SCK

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPIF0_SCK. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for SPIF0_SCK. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPIF0_SCK
21:20	RW	0x0	Pin Mux Selector for SPIF0_SCK
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPIF0_SCK. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPIF0_SCK.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPIF0_CS_X. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for SPIF0_CS_X. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPIF0_CS_X
5:4	RW	0x0	Pin Mux Selector for SPIF0_CS_X
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPIF0_CS_X. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPIF0_CS_X.

REG10: offset 0x0040

Table 19: SPIF0_SDO and SPIF1_CLK_SEL1

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPIF1_CLK_SEL1. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for SPIF1_CLK_SEL1. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPIF1_CLK_SEL1
21:20	RW	0x0	Pin Mux Selector for SPIF1_CLK_SEL1
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPIF1_CLK_SEL1. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPIF1_CLK_SEL1.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPIF0_SDO. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for SPIF0_SDO. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPIF0_SDO
5:4	RW	0x0	Pin Mux Selector for SPIF0_SDO
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPIF0_SDO. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPIF0_SDO.

REG11: offset 0x0044

Table 20: SPIF1_CLK_SEL0 and SPIF1_WP_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPIF1_WP_X. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for SPIF1_WP_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPIF1_WP_X
21:20	RW	0x0	Pin Mux Selector for SPIF1_WP_X
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPIF1_WP_X. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPIF1_WP_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPIF1_CLK_SEL0. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for SPIF1_CLK_SEL0. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPIF1_CLK_SEL0
5:4	RW	0x0	Pin Mux Selector for SPIF1_CLK_SEL0
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPIF1_CLK_SEL0. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPIF1_CLK_SEL0.

REG12: offset 0x0048

Table 21: SPIF1_HOLD_X and SPIF1_SDI

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPIF1_SDI. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for SPIF1_SDI. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPIF1_SDI
21:20	RW	0x0	Pin Mux Selector for SPIF1_SDI
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPIF1_SDI. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPIF1_SDI.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPIF1_HOLD_X. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for SPIF1_HOLD_X. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPIF1_HOLD_X
5:4	RW	0x0	Pin Mux Selector for SPIF1_HOLD_X
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPIF1_HOLD_X. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPIF1_HOLD_X.

REG13: offset 0x004c

Table 22: SPIF1_CS_X and SPIF1_SCK

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPIF1_SCK. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for SPIF1_SCK. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPIF1_SCK
21:20	RW	0x0	Pin Mux Selector for SPIF1_SCK
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPIF1_SCK. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPIF1_SCK.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPIF1_CS_X. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for SPIF1_CS_X. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPIF1_CS_X
5:4	RW	0x0	Pin Mux Selector for SPIF1_CS_X
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPIF1_CS_X. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPIF1_CS_X.

REG14: offset 0x0050

Table 23: SPIF1_SDO and EMMC_WP

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for EMMC_WP. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for EMMC_WP. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for EMMC_WP
21:20	RW	0x0	Pin Mux Selector for EMMC_WP
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for EMMC_WP. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for EMMC_WP.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPIF1_SDO. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for SPIF1_SDO. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPIF1_SDO
5:4	RW	0x0	Pin Mux Selector for SPIF1_SDO
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPIF1_SDO. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPIF1_SDO.

REG15: offset 0x0054

Table 24: EMMC_CD_X and EMMC_RST_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for EMMC_RST_X. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for EMMC_RST_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for EMMC_RST_X
21:20	RW	0x0	Pin Mux Selector for EMMC_RST_X
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for EMMC_RST_X. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for EMMC_RST_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for EMMC_CD_X. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for EMMC_CD_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for EMMC_CD_X
5:4	RW	0x0	Pin Mux Selector for EMMC_CD_X
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for EMMC_CD_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for EMMC_CD_X.

REG16: offset 0x0058

Table 25: EMMC_PWR_EN and SDIO_CD_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SDIO_CD_X. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for SDIO_CD_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SDIO_CD_X
21:20	RW	0x0	Pin Mux Selector for SDIO_CD_X
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for SDIO_CD_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for SDIO_CD_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for EMMC_PWR_EN. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for EMMC_PWR_EN. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for EMMC_PWR_EN
5:4	RW	0x0	Pin Mux Selector for EMMC_PWR_EN
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for EMMC_PWR_EN. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for EMMC_PWR_EN.

REG17: offset 0x005c

Table 26: SDIO_WP and SDIO_RST_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SDIO_RST_X. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for SDIO_RST_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SDIO_RST_X
21:20	RW	0x0	Pin Mux Selector for SDIO_RST_X
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SDIO_RST_X. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SDIO_RST_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SDIO_WP. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for SDIO_WP. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SDIO_WP
5:4	RW	0x0	Pin Mux Selector for SDIO_WP
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SDIO_WP. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SDIO_WP.

REG18: offset 0x0060

Table 27: SDIO_PWR_EN and RGMII0_TXD0

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for RGMII0_TXD0. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for RGMII0_TXD0. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for RGMII0_TXD0
21:20	RW	0x0	Pin Mux Selector for RGMII0_TXD0
19	RW	0x0	Pull Down Enable for RGMII0_TXD0
18	RW	0x0	Pull Up Enable for RGMII0_TXD0
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SDIO_PWR_EN. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for SDIO_PWR_EN. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SDIO_PWR_EN
5:4	RW	0x0	Pin Mux Selector for SDIO_PWR_EN
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SDIO_PWR_EN. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SDIO_PWR_EN.

REG19: offset 0x0064

Table 28: RGMII0_TXD1 and RGMII0_TXD2

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for RGMII0_TXD2. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for RGMII0_TXD2. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for RGMII0_TXD2
21:20	RW	0x0	Pin Mux Selector for RGMII0_TXD2
19	RW	0x0	Pull Down Enable for RGMII0_TXD2
18	RW	0x0	Pull Up Enable for RGMII0_TXD2
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for RGMII0_TXD1. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for RGMII0_TXD1. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for RGMII0_TXD1
5:4	RW	0x0	Pin Mux Selector for RGMII0_TXD1
3	RW	0x0	Pull Down Enable for RGMII0_TXD1
2	RW	0x0	Pull Up Enable for RGMII0_TXD1
1:0	RW	NA	Reserved

REG1a: offset 0x0068

Table 29: RGMII0_TXD3 and RGMII0_TXCTRL

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for RGMII0_TXCTRL. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for RGMII0_TXCTRL. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for RGMII0_TXCTRL
21:20	RW	0x0	Pin Mux Selector for RGMII0_TXCTRL
19	RW	0x0	Pull Down Enable for RGMII0_TXCTRL
18	RW	0x0	Pull Up Enable for RGMII0_TXCTRL
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for RGMII0_TXD3. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for RGMII0_TXD3. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for RGMII0_TXD3
5:4	RW	0x0	Pin Mux Selector for RGMII0_TXD3
3	RW	0x0	Pull Down Enable for RGMII0_TXD3
2	RW	0x0	Pull Up Enable for RGMII0_TXD3
1:0	RW	NA	Reserved

REG1b: offset 0x006c

Table 30: RGMII0_RXD0 and RGMII0_RXD1

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for RGMII0_RXD1. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for RGMII0_RXD1. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for RGMII0_RXD1
21:20	RW	0x0	Pin Mux Selector for RGMII0_RXD1
19	RW	0x0	Pull Down Enable for RGMII0_RXD1
18	RW	0x0	Pull Up Enable for RGMII0_RXD1
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for RGMII0_RXD0. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for RGMII0_RXD0. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for RGMII0_RXD0
5:4	RW	0x0	Pin Mux Selector for RGMII0_RXD0
3	RW	0x0	Pull Down Enable for RGMII0_RXD0
2	RW	0x0	Pull Up Enable for RGMII0_RXD0
1:0	RW	NA	Reserved

REG1c: offset 0x0070

Table 31: RGMII0_RXD2 and RGMII0_RXD3

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for RGMII0_RXD3. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for RGMII0_RXD3. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for RGMII0_RXD3
21:20	RW	0x0	Pin Mux Selector for RGMII0_RXD3
19	RW	0x0	Pull Down Enable for RGMII0_RXD3
18	RW	0x0	Pull Up Enable for RGMII0_RXD3
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for RGMII0_RXD2. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for RGMII0_RXD2. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for RGMII0_RXD2
5:4	RW	0x0	Pin Mux Selector for RGMII0_RXD2
3	RW	0x0	Pull Down Enable for RGMII0_RXD2
2	RW	0x0	Pull Up Enable for RGMII0_RXD2
1:0	RW	NA	Reserved

REG1d: offset 0x0074

Table 32: RGMII0_RXCTRL and RGMII0_TXC

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for RGMII0_TXC. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for RGMII0_TXC. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for RGMII0_TXC
21:20	RW	0x0	Pin Mux Selector for RGMII0_TXC
19	RW	0x0	Pull Down Enable for RGMII0_TXC
18	RW	0x0	Pull Up Enable for RGMII0_TXC
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for RGMII0_RXCTRL. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for RGMII0_RXCTRL. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for RGMII0_RXCTRL
5:4	RW	0x0	Pin Mux Selector for RGMII0_RXCTRL
3	RW	0x0	Pull Down Enable for RGMII0_RXCTRL
2	RW	0x0	Pull Up Enable for RGMII0_RXCTRL
1:0	RW	NA	Reserved

REG1e: offset 0x0078

Table 33: RGMII0_RXC and RGMII0_REFCLKO

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for RGMII0_REFCLKO. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for RGMII0_REFCLKO. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for RGMII0_REFCLKO
21:20	RW	0x0	Pin Mux Selector for RGMII0_REFCLKO
19	RW	0x0	Pull Down Enable for RGMII0_REFCLKO
18	RW	0x0	Pull Up Enable for RGMII0_REFCLKO
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for RGMII0_RXC. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for RGMII0_RXC. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for RGMII0_RXC
5:4	RW	0x0	Pin Mux Selector for RGMII0_RXC
3	RW	0x0	Pull Down Enable for RGMII0_RXC
2	RW	0x0	Pull Up Enable for RGMII0_RXC
1:0	RW	NA	Reserved

REG1f: offset 0x007c

Table 34: RGMII0_IRQ and RGMII0_MDC

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for RGMII0_MDC. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for RGMII0_MDC. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for RGMII0_MDC
21:20	RW	0x0	Pin Mux Selector for RGMII0_MDC
19	RW	0x0	Pull Down Enable for RGMII0_MDC
18	RW	0x0	Pull Up Enable for RGMII0_MDC
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for RGMII0_IRQ. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for RGMII0_IRQ. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for RGMII0_IRQ
5:4	RW	0x0	Pin Mux Selector for RGMII0_IRQ
3	RW	0x0	Pull Down Enable for RGMII0_IRQ
2	RW	0x0	Pull Up Enable for RGMII0_IRQ
1:0	RW	NA	Reserved

REG20: offset 0x0080

Table 35: RGMII0_MDIO and PWM0

Fields	Type	De-fault	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for PWM0. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for PWM0. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for PWM0
21:20	RW	0x0	Pin Mux Selector for PWM0
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for PWM0. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for PWM0.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for RGMII0_MDIO. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for RGMII0_MDIO. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for RGMII0_MDIO
5:4	RW	0x0	Pin Mux Selector for RGMII0_MDIO
3	RW	0x0	Pull Down Enable for RGMII0_MDIO
2	RW	0x0	Pull Up Enable for RGMII0_MDIO
1:0	RW	NA	Reserved

REG21: offset 0x0084

Table 36: PWM1 and PWM2

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for PWM2. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for PWM2. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for PWM2
21:20	RW	0x0	Pin Mux Selector for PWM2
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for PWM2. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for PWM2.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for PWM1. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for PWM1. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for PWM1
5:4	RW	0x0	Pin Mux Selector for PWM1
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for PWM1. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for PWM1.

REG22: offset 0x0088

Table 37: PWM3 and FAN0

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for FAN0. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for FAN0. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for FAN0
21:20	RW	0x0	Pin Mux Selector for FAN0
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for FAN0. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for FAN0.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for PWM3. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for PWM3. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for PWM3
5:4	RW	0x0	Pin Mux Selector for PWM3
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for PWM3. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for PWM3.

REG23: offset 0x008c

Table 38: FAN1 and FAN2

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for FAN2. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for FAN2. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for FAN2
21:20	RW	0x0	Pin Mux Selector for FAN2
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for FAN2. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for FAN2.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for FAN1. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for FAN1. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for FAN1
5:4	RW	0x0	Pin Mux Selector for FAN1
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for FAN1. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for FAN1.

REG24: offset 0x0090

Table 39: FAN3 and IIC0_SDA

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for IIC0_SDA. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for IIC0_SDA. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for IIC0_SDA
21:20	RW	0x0	Pin Mux Selector for IIC0_SDA
19	RW	0x0	Pull Down Enable for IIC0_SDA
18	RW	0x0	Pull Up Enable for IIC0_SDA
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for FAN3. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for FAN3. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for FAN3
5:4	RW	0x0	Pin Mux Selector for FAN3
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for FAN3. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for FAN3.

REG25: offset 0x0094

Table 40: IIC0_SCL and IIC1_SDA

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for IIC1_SDA. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for IIC1_SDA. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for IIC1_SDA
21:20	RW	0x0	Pin Mux Selector for IIC1_SDA
19	RW	0x0	Pull Down Enable for IIC1_SDA
18	RW	0x0	Pull Up Enable for IIC1_SDA
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for IIC0_SCL. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for IIC0_SCL. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for IIC0_SCL
5:4	RW	0x0	Pin Mux Selector for IIC0_SCL
3	RW	0x0	Pull Down Enable for IIC0_SCL
2	RW	0x0	Pull Up Enable for IIC0_SCL
1:0	RW	NA	Reserved

REG26: offset 0x0098

Table 41: IIC1_SCL and IIC2_SDA

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for IIC2_SDA. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for IIC2_SDA. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for IIC2_SDA
21:20	RW	0x0	Pin Mux Selector for IIC2_SDA
19	RW	0x0	Pull Down Enable for IIC2_SDA
18	RW	0x0	Pull Up Enable for IIC2_SDA
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for IIC1_SCL. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for IIC1_SCL. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for IIC1_SCL
5:4	RW	0x0	Pin Mux Selector for IIC1_SCL
3	RW	0x0	Pull Down Enable for IIC1_SCL
2	RW	0x0	Pull Up Enable for IIC1_SCL
1:0	RW	NA	Reserved

REG27: offset 0x009c

Table 42: IIC2_SCL and IIC3_SDA

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for IIC3_SDA. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for IIC3_SDA. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for IIC3_SDA
21:20	RW	0x0	Pin Mux Selector for IIC3_SDA
19	RW	0x0	Pull Down Enable for IIC3_SDA
18	RW	0x0	Pull Up Enable for IIC3_SDA
17:16	RW	NA	Reserved
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for IIC2_SCL. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for IIC2_SCL. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for IIC2_SCL
5:4	RW	0x0	Pin Mux Selector for IIC2_SCL
3	RW	0x0	Pull Down Enable for IIC2_SCL
2	RW	0x0	Pull Up Enable for IIC2_SCL
1:0	RW	NA	Reserved

REG28: offset 0x00a0

Table 43: IIC3_SCL and UART0_TX

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for UART0_TX. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for UART0_TX. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for UART0_TX
21:20	RW	0x0	Pin Mux Selector for UART0_TX
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for UART0_TX. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for UART0_TX.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for IIC3_SCL. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for IIC3_SCL. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for IIC3_SCL
5:4	RW	0x0	Pin Mux Selector for IIC3_SCL
3	RW	0x0	Pull Down Enable for IIC3_SCL
2	RW	0x0	Pull Up Enable for IIC3_SCL
1:0	RW	NA	Reserved

REG29: offset 0x00a4

Table 44: UART0_RX and UART0_RTS

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for UART0_RTS. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for UART0_RTS. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for UART0_RTS
21:20	RW	0x0	Pin Mux Selector for UART0_RTS
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for UART0_RTS. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for UART0_RTS.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for UART0_RX. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for UART0_RX. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for UART0_RX
5:4	RW	0x0	Pin Mux Selector for UART0_RX
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for UART0_RX. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for UART0_RX.

REG2a: offset 0x00a8

Table 45: UART0_CTS and UART1_TX

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for UART1_TX. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for UART1_TX. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for UART1_TX
21:20	RW	0x0	Pin Mux Selector for UART1_TX
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for UART1_TX. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for UART1_TX.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for UART0_CTS. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for UART0_CTS. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for UART0_CTS
5:4	RW	0x0	Pin Mux Selector for UART0_CTS
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for UART0_CTS. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for UART0_CTS.

REG2b: offset 0x00ac

Table 46: UART1_RX and UART1_RTS

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for UART1_RTS. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for UART1_RTS. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for UART1_RTS
21:20	RW	0x0	Pin Mux Selector for UART1_RTS
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for UART1_RTS. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for UART1_RTS.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for UART1_RX. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for UART1_RX. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for UART1_RX
5:4	RW	0x0	Pin Mux Selector for UART1_RX
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for UART1_RX. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for UART1_RX.

REG2c: offset 0x00b0

Table 47: UART1_CTS and UART2_TX

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for UART2_TX. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for UART2_TX. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for UART2_TX
21:20	RW	0x0	Pin Mux Selector for UART2_TX
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for UART2_TX. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for UART2_TX.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for UART1_CTS. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for UART1_CTS. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for UART1_CTS
5:4	RW	0x0	Pin Mux Selector for UART1_CTS
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for UART1_CTS. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for UART1_CTS.

REG2d: offset 0x00b4

Table 48: UART2_RX and UART2_RTS

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for UART2_RTS. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for UART2_RTS. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for UART2_RTS
21:20	RW	0x0	Pin Mux Selector for UART2_RTS
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for UART2_RTS. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for UART2_RTS.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for UART2_RX. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for UART2_RX. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for UART2_RX
5:4	RW	0x0	Pin Mux Selector for UART2_RX
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for UART2_RX. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for UART2_RX.

REG2e: offset 0x00b8

Table 49: UART2_CTS and UART3_TX

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for UART3_TX. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for UART3_TX. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for UART3_TX
21:20	RW	0x0	Pin Mux Selector for UART3_TX
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for UART3_TX. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for UART3_TX.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for UART2_CTS. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for UART2_CTS. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for UART2_CTS
5:4	RW	0x0	Pin Mux Selector for UART2_CTS
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for UART2_CTS. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for UART2_CTS.

REG2f: offset 0x00bc

Table 50: UART3_RX and UART3_RTS

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for UART3_RTS. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for UART3_RTS. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for UART3_RTS
21:20	RW	0x0	Pin Mux Selector for UART3_RTS
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for UART3_RTS. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for UART3_RTS.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for UART3_RX. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for UART3_RX. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for UART3_RX
5:4	RW	0x0	Pin Mux Selector for UART3_RX
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for UART3_RX. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for UART3_RX.

REG30: offset 0x00c0

Table 51: UART3_CTS and SPI0_CS0_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPI0_CS0_X. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for SPI0_CS0_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPI0_CS0_X
21:20	RW	0x0	Pin Mux Selector for SPI0_CS0_X
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPI0_CS0_X. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPI0_CS0_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for UART3_CTS. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for UART3_CTS. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for UART3_CTS
5:4	RW	0x0	Pin Mux Selector for UART3_CTS
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for UART3_CTS. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for UART3_CTS.

REG31: offset 0x00c4

Table 52: SPI0_CS1_X and SPI0_SDI

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPI0_SDI. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for SPI0_SDI. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPI0_SDI
21:20	RW	0x0	Pin Mux Selector for SPI0_SDI
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPI0_SDI. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPI0_SDI.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPI0_CS1_X. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for SPI0_CS1_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPI0_CS1_X
5:4	RW	0x0	Pin Mux Selector for SPI0_CS1_X
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPI0_CS1_X. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPI0_CS1_X.

REG32: offset 0x00c8

Table 53: SPI0_SDO and SPI0_SCK

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPI0_SCK. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for SPI0_SCK. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPI0_SCK
21:20	RW	0x0	Pin Mux Selector for SPI0_SCK
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPI0_SCK. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPI0_SCK.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPI0_SDO. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for SPI0_SDO. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPI0_SDO
5:4	RW	0x0	Pin Mux Selector for SPI0_SDO
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPI0_SDO. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPI0_SDO.

REG33: offset 0x00cc

Table 54: SPI1_CS0_X and SPI1_CS1_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPI1_CS1_X. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for SPI1_CS1_X. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPI1_CS1_X
21:20	RW	0x0	Pin Mux Selector for SPI1_CS1_X
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPI1_CS1_X. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPI1_CS1_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPI1_CS0_X. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for SPI1_CS0_X. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPI1_CS0_X
5:4	RW	0x0	Pin Mux Selector for SPI1_CS0_X
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPI1_CS0_X. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPI1_CS0_X.

REG34: offset 0x00d0

Table 55: SPI1_SDI and SPI1_SDO

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for SPI1_SDO. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for SPI1_SDO. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for SPI1_SDO
21:20	RW	0x0	Pin Mux Selector for SPI1_SDO
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SPI1_SDO. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for SPI1_SDO.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPI1_SDI. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for SPI1_SDI. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPI1_SDI
5:4	RW	0x0	Pin Mux Selector for SPI1_SDI
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPI1_SDI. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPI1_SDI.

REG35: offset 0x00d4

Table 56: SPI1_SCK and JTAG0_TDO

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for JTAG0_TDO. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for JTAG0_TDO. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for JTAG0_TDO
21:20	RW	0x0	Pin Mux Selector for JTAG0_TDO
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for JTAG0_TDO. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for JTAG0_TDO.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for SPI1_SCK. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for SPI1_SCK. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for SPI1_SCK
5:4	RW	0x0	Pin Mux Selector for SPI1_SCK
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SPI1_SCK. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for SPI1_SCK.

REG36: offset 0x00d8

Table 57: JTAG0_TCK and JTAG0_TDI

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for JTAG0_TDI. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for JTAG0_TDI. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for JTAG0_TDI
21:20	RW	0x0	Pin Mux Selector for JTAG0_TDI
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for JTAG0_TDI. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for JTAG0_TDI.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for JTAG0_TCK. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for JTAG0_TCK. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for JTAG0_TCK
5:4	RW	0x0	Pin Mux Selector for JTAG0_TCK
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for JTAG0_TCK. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for JTAG0_TCK.

REG37: offset 0x00dc

Table 58: JTAG0_TMS and JTAG0_TRST_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for JTAG0_TRST_X. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for JTAG0_TRST_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for JTAG0_TRST_X
21:20	RW	0x0	Pin Mux Selector for JTAG0_TRST_X
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for JTAG0_TRST_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for JTAG0_TRST_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for JTAG0_TMS. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for JTAG0_TMS. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for JTAG0_TMS
5:4	RW	0x0	Pin Mux Selector for JTAG0_TMS
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for JTAG0_TMS. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for JTAG0_TMS.

REG38: offset 0x00e0

Table 59: JTAG0_SRST_X and JTAG1_TDO

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for JTAG1_TDO. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for JTAG1_TDO. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for JTAG1_TDO
21:20	RW	0x0	Pin Mux Selector for JTAG1_TDO
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for JTAG1_TDO. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for JTAG1_TDO.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for JTAG0_SRST_X. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for JTAG0_SRST_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for JTAG0_SRST_X
5:4	RW	0x0	Pin Mux Selector for JTAG0_SRST_X
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for JTAG0_SRST_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for JTAG0_SRST_X.

REG39: offset 0x00e4

Table 60: JTAG1_TCK and JTAG1_TDI

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for JTAG1_TDI. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for JTAG1_TDI. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for JTAG1_TDI
21:20	RW	0x0	Pin Mux Selector for JTAG1_TDI
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for JTAG1_TDI. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for JTAG1_TDI.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for JTAG1_TCK. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for JTAG1_TCK. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for JTAG1_TCK
5:4	RW	0x0	Pin Mux Selector for JTAG1_TCK
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for JTAG1_TCK. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for JTAG1_TCK.

REG3a: offset 0x00e8

Table 61: JTAG1_TMS and JTAG1_TRST_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for JTAG1_TRST_X. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for JTAG1_TRST_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for JTAG1_TRST_X
21:20	RW	0x0	Pin Mux Selector for JTAG1_TRST_X
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for JTAG1_TRST_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for JTAG1_TRST_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for JTAG1_TMS. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for JTAG1_TMS. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for JTAG1_TMS
5:4	RW	0x0	Pin Mux Selector for JTAG1_TMS
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for JTAG1_TMS. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for JTAG1_TMS.

REG3b: offset 0x00ec

Table 62: JTAG1_SRST_X and JTAG2_TDO

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for JTAG2_TDO. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for JTAG2_TDO. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for JTAG2_TDO
21:20	RW	0x0	Pin Mux Selector for JTAG2_TDO
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for JTAG2_TDO. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for JTAG2_TDO.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for JTAG1_SRST_X. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for JTAG1_SRST_X. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for JTAG1_SRST_X
5:4	RW	0x0	Pin Mux Selector for JTAG1_SRST_X
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for JTAG1_SRST_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for JTAG1_SRST_X.

REG3c: offset 0x00f0

Table 63: JTAG2_TCK and JTAG2_TDI

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for JTAG2_TDI. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for JTAG2_TDI. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for JTAG2_TDI
21:20	RW	0x0	Pin Mux Selector for JTAG2_TDI
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for JTAG2_TDI. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for JTAG2_TDI.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for JTAG2_TCK. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for JTAG2_TCK. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for JTAG2_TCK
5:4	RW	0x0	Pin Mux Selector for JTAG2_TCK
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for JTAG2_TCK. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for JTAG2_TCK.

REG3d: offset 0x00f4

Table 64: JTAG2_TMS and JTAG2_TRST_X

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for JTAG2_TRST_X. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for JTAG2_TRST_X. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for JTAG2_TRST_X
21:20	RW	0x0	Pin Mux Selector for JTAG2_TRST_X
19:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for JTAG2_TRST_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for JTAG2_TRST_X.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for JTAG2_TMS. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for JTAG2_TMS. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for JTAG2_TMS
5:4	RW	0x0	Pin Mux Selector for JTAG2_TMS
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for JTAG2_TMS. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for JTAG2_TMS.

REG3e: offset 0x00f8

Table 65: JTAG2_SRST_X and GPIO0

Fields	Type	De-fault	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO0. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for GPIO0. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO0
21:20	RW	0x0	Pin Mux Selector for GPIO0
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO0. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO0.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for JTAG2_SRST_X. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for JTAG2_SRST_X. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for JTAG2_SRST_X
5:4	RW	0x0	Pin Mux Selector for JTAG2_SRST_X
3:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for JTAG2_SRST_X. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for JTAG2_SRST_X.

REG3f: offset 0x00fc

Table 66: GPIO1 and GPIO2

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO2. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for GPIO2. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO2
21:20	RW	0x0	Pin Mux Selector for GPIO2
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO2. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO2.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO1. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for GPIO1. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO1
5:4	RW	0x0	Pin Mux Selector for GPIO1
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO1. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO1.

REG40: offset 0x0100

Table 67: GPIO3 and GPIO4

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO4. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for GPIO4. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO4
21:20	RW	0x0	Pin Mux Selector for GPIO4
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO4. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO4.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO3. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for GPIO3. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO3
5:4	RW	0x0	Pin Mux Selector for GPIO3
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO3. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO3.

REG41: offset 0x0104

Table 68: GPIO5 and GPIO6

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO6. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for GPIO6. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO6
21:20	RW	0x0	Pin Mux Selector for GPIO6
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO6. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO6.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO5. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for GPIO5. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO5
5:4	RW	0x0	Pin Mux Selector for GPIO5
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO5. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO5.

REG42: offset 0x0108

Table 69: GPIO7 and GPIO8

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO8. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for GPIO8. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO8
21:20	RW	0x0	Pin Mux Selector for GPIO8
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO8. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO8.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO7. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for GPIO7. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO7
5:4	RW	0x0	Pin Mux Selector for GPIO7
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO7. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO7.

REG43: offset 0x010c

Table 70: GPIO9 and GPIO10

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO10. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for GPIO10. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO10
21:20	RW	0x0	Pin Mux Selector for GPIO10
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO10. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO10.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO9. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for GPIO9. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO9
5:4	RW	0x0	Pin Mux Selector for GPIO9
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO9. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO9.

REG44: offset 0x0110

Table 71: GPIO11 and GPIO12

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO12. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for GPIO12. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO12
21:20	RW	0x0	Pin Mux Selector for GPIO12
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO12. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO12.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO11. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for GPIO11. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO11
5:4	RW	0x0	Pin Mux Selector for GPIO11
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO11. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO11.

REG45: offset 0x0114

Table 72: GPIO13 and GPIO14

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO14. '1' enables PAD output mode under IP's drive
26	RW	0x1	Schmitt trigger enable for GPIO14. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO14
21:20	RW	0x0	Pin Mux Selector for GPIO14
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO14. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO14.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO13. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for GPIO13. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO13
5:4	RW	0x0	Pin Mux Selector for GPIO13
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO13. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO13.

REG46: offset 0x0118

Table 73: GPIO15 and GPIO16

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO16. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for GPIO16. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO16
21:20	RW	0x0	Pin Mux Selector for GPIO16
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO16. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO16.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO15. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for GPIO15. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO15
5:4	RW	0x0	Pin Mux Selector for GPIO15
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO15. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO15.

REG47: offset 0x011c

Table 74: GPIO17 and GPIO18

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO18. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x0	Schmitt trigger enable for GPIO18. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO18
21:20	RW	0x0	Pin Mux Selector for GPIO18
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO18. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO18.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO17. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x0	Schmitt trigger enable for GPIO17. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO17
5:4	RW	0x0	Pin Mux Selector for GPIO17
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO17. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO17.

REG48: offset 0x0120

Table 75: GPIO19 and GPIO20

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO20. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for GPIO20. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO20
21:20	RW	0x0	Pin Mux Selector for GPIO20
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO20. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO20.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO19. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for GPIO19. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO19
5:4	RW	0x0	Pin Mux Selector for GPIO19
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO19. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO19.

REG49: offset 0x0124

Table 76: GPIO21 and GPIO22

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO22. '1' enables PAD output mode under IP's drive
26	RW	0x0	Schmitt trigger enable for GPIO22. '1' enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO22
21:20	RW	0x0	Pin Mux Selector for GPIO22
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO22. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO22.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO21. '1' enables PAD output mode under IP's drive
10	RW	0x0	Schmitt trigger enable for GPIO21. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO21
5:4	RW	0x0	Pin Mux Selector for GPIO21
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO21. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO21.

REG4a: offset 0x0128

Table 77: GPIO23 and GPIO24

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO24. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for GPIO24. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO24
21:20	RW	0x0	Pin Mux Selector for GPIO24
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO24. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO24.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO23. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for GPIO23. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO23
5:4	RW	0x0	Pin Mux Selector for GPIO23
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO23. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO23.

REG4b: offset 0x012c

Table 78: GPIO25 and GPIO26

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO26. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for GPIO26. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO26
21:20	RW	0x0	Pin Mux Selector for GPIO26
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO26. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO26.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO25. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for GPIO25. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO25
5:4	RW	0x0	Pin Mux Selector for GPIO25
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO25. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO25.

REG4c: offset 0x0130

Table 79: GPIO27 and GPIO28

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO28. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for GPIO28. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO28
21:20	RW	0x0	Pin Mux Selector for GPIO28
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO28. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for GPIO28.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO27. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for GPIO27. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO27
5:4	RW	0x0	Pin Mux Selector for GPIO27
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO27. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for GPIO27.

REG4d: offset 0x0134

Table 80: GPIO29 and GPIO30

Fields	Type	Default	Function
31:28	RO	NA	Reserved
27	RW	0x1	PAD OEX enable for GPIO30. ‘1’ enables PAD output mode under IP’s drive
26	RW	0x1	Schmitt trigger enable for GPIO30. ‘1’ enables Schmitt trigger input function
25:22	RW	0x8	Driving Selector for GPIO30
21:20	RW	0x0	Pin Mux Selector for GPIO30
19:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for GPIO30. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for GPIO30.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO29. ‘1’ enables PAD output mode under IP’s drive
10	RW	0x1	Schmitt trigger enable for GPIO29. ‘1’ enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO29
5:4	RW	0x0	Pin Mux Selector for GPIO29
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO29. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for GPIO29.

REG4e: offset 0x0138

Table 81: GPIO31 and MODE_SEL0

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for MODE_SEL0. '1' enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for MODE_SEL0. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for MODE_SEL0.
15:12	RO	NA	Reserved
11	RW	0x1	PAD OEX enable for GPIO31. '1' enables PAD output mode under IP's drive
10	RW	0x1	Schmitt trigger enable for GPIO31. '1' enables Schmitt trigger input function
9:6	RW	0x8	Driving Selector for GPIO31
5:4	RW	0x0	Pin Mux Selector for GPIO31
3:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for GPIO31. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for GPIO31.

REG4f: offset 0x013c

Table 82: MODE_SEL1 and MODE_SEL2

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for MODE_SEL2. '1' enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for MODE_SEL2. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for MODE_SEL2.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for MODE_SEL1. '1' enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for MODE_SEL1. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for MODE_SEL1.

REG50: offset 0x0140

Table 83: BOOT_SEL0 and BOOT_SEL1

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for BOOT_SEL1. '1' enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for BOOT_SEL1. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for BOOT_SEL1.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for BOOT_SEL0. '1' enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for BOOT_SEL0. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for BOOT_SEL0.

REG51: offset 0x0144

Table 84: BOOT_SEL2 and BOOT_SEL3

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for BOOT_SEL3. '1' enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for BOOT_SEL3. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for BOOT_SEL3.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for BOOT_SEL2. '1' enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for BOOT_SEL2. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for BOOT_SEL2.

REG52: offset 0x0148

Table 85: BOOT_SEL4 and BOOT_SEL5

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for BOOT_SEL5. '1' enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for BOOT_SEL5. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for BOOT_SEL5.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for BOOT_SEL4. '1' enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for BOOT_SEL4. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for BOOT_SEL4.

REG53: offset 0x014c

Table 86: BOOT_SEL6 and BOOT_SEL7

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for BOOT_SEL7. '1' enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for BOOT_SEL7. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for BOOT_SEL7.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for BOOT_SEL6. '1' enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for BOOT_SEL6. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for BOOT_SEL6.

REG54: offset 0x0150

Table 87: MULTI_SCKT and SCKT_ID0

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for SCKT_ID0. ‘1’ enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for SCKT_ID0. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for SCKT_ID0.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for MULTI_SCKT. ‘1’ enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for MULTI_SCKT. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for MULTI_SCKT.

REG55: offset 0x0154

Table 88: SCKT_ID1 and PLL_CLK_IN_MAIN

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x0	Schmitt trigger enable for PLL_CLK_IN_MAIN. ‘1’ enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for PLL_CLK_IN_MAIN. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for PLL_CLK_IN_MAIN.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for SCKT_ID1. ‘1’ enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for SCKT_ID1. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for SCKT_ID1.

REG56: offset 0x0158

Table 89: PLL_CLK_IN_DDR_L and PLL_CLK_IN_DDR_R

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x0	Schmitt trigger enable for PLL_CLK_IN_DDR_R. ‘1’ enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for PLL_CLK_IN_DDR_R. (0:pull down; 1:pull up)
16	RW	0x0	Pull Enable for PLL_CLK_IN_DDR_R.
15:11	RO	NA	Reserved
10	RW	0x0	Schmitt trigger enable for PLL_CLK_IN_DDR_L. ‘1’ enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for PLL_CLK_IN_DDR_L. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for PLL_CLK_IN_DDR_L.

REG57: offset 0x015c

Table 90: XTAL_32K and SYS_RST_X

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for SYS_RST_X. '1' enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x1	Pull Selector for SYS_RST_X. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for SYS_RST_X.
15:11	RO	NA	Reserved
10	RW	0x0	Schmitt trigger enable for XTAL_32K. '1' enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for XTAL_32K. (0:pull down; 1:pull up)
0	RW	0x0	Pull Enable for XTAL_32K.

REG58: offset 0x0160

Table 91: PWR_BUTTON and TEST_EN

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RO	0x1	Schmitt trigger enable for TEST_EN. '1' enables Schmitt trigger input function
25:18	RO	NA	Reserved
17	RO	0x0	Pull Selector for TEST_EN. (0:pull down; 1:pull up)
16	RO	0x1	Pull Enable for TEST_EN.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for PWR_BUTTON. '1' enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x1	Pull Selector for PWR_BUTTON. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for PWR_BUTTON.

REG59: offset 0x0164

Table 92: TEST_MODE_MBIST and TEST_MODE_SCAN

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for TEST_MODE_SCAN. '1' enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for TEST_MODE_SCAN. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for TEST_MODE_SCAN.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for TEST_MODE_MBIST. '1' enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for TEST_MODE_MBIST. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for TEST_MODE_MBIST.

REG5a: offset 0x0168

Table 93: TEST_MODE_BSD and BISR_BYP

Fields	Type	Default	Function
31:27	RO	NA	Reserved
26	RW	0x1	Schmitt trigger enable for BISR_BYP. '1' enables Schmitt trigger input function
25:18	RW	NA	Reserved
17	RW	0x0	Pull Selector for BISR_BYP. (0:pull down; 1:pull up)
16	RW	0x1	Pull Enable for BISR_BYP.
15:11	RO	NA	Reserved
10	RW	0x1	Schmitt trigger enable for TEST_MODE_BSD. '1' enables Schmitt trigger input function
9:2	RW	NA	Reserved
1	RW	0x0	Pull Selector for TEST_MODE_BSD. (0:pull down; 1:pull up)
0	RW	0x1	Pull Enable for TEST_MODE_BSD.

CHAPTER

FOUR

CLOCK

4.1 Clock sources

Three 25MHz clock generation chips are required to provide reference clocks for SG2042.

Clock gen chips are the only option for reference clock input and they should be connected to PLL_CLK_IN_MAIN, PLL_CLK_IN_DPLL_L and PLL_CLK_IN_DPLL_R.

External clock sources are listed in table *External clock sources*

Table 1: External clock sources

Clock Name	Frequency	Description
PLL_CLK_IN_MAIN	25MHz	Main clock
PLL_CLK_IN_DPLL_L	25MHz	DDR0 and DDR1
PLL_CLK_IN_DPLL_R	25MHz	DDR2 and DDR3
PCIE0_REFCLK_M	100MHz	PCIe0 reference clock
PCIE0_REFCLK_P	100MHz	PCIe0 reference clock
PCIE1_REFCLK_M	100MHz	PCIe1 reference clock
PCIE1_REFCLK_P	100MHz	PCIe1 reference clock

4.2 PLL

Different parts inside the chip works on different frequencies. As there is no "one-size-fits-all" PLL, SG2042 instantiates 4 PLLs to satisfy logic's clock requirements.

- MPLL: the name is short for Main PLL. The output clocks of this PLL are mainly used in RP subsystem and AP subsystem.
- FPLL: the name is shoft for Fixed PLL. This PLL generates fixed frequency clock, with output clock at 1.0 GHz. The output clocks of this PLL are mainly used in data and configuraiton bus.
- DPLL0/1: the name is short for DDR PLL. The output clocks of thes PLLs are mainly used in DDR subsystem.

And in order to reconfigure PLL clock frequency on the fly, MPLL and DPLL0/1 use FPLL as a backup.

The micro architecture of a pll cell looks like figure *PLL micro architecture*

The output clock frequency is influenced by:

- FREF: Reference Clock Input (10MHz to 800MHz). SG2042 uses 25MHz reference clock.
- FOUTPOSTDIV: Output Clock (16MHz to 3200MHz)
- REFDIV: Reference divide value (1 to 63)

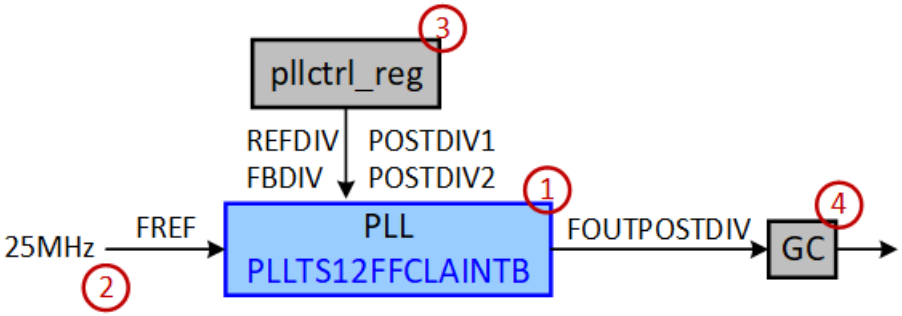

Fig. 1: PLL micro architecture

- FBDIV: Feedback divide value (16 to 320)
- POSTDIV1: Post Divide 1 setting (1 to 7)
- POSTDIV2: Post Divide 2 setting (1 to 7)

The output clock frequency is calculated as:

$$FOUTPOSTDIV = FREF * FBDIV/REFDIV/(POSTDIV1 * POSTDIV2)$$

For reference clock, it is used in reset sequence. Only after certain reset sequence (1.5ms), PLL starts to work.

Software together with a dedicated hardware module are in charge of the PLL control, especially the modification of PLL DIV values (REFDIV, FBDIV, POSTDIV1, POSTDIV2).

After Power-On Reset, embedded hardware is able to select the proper initial REFDIV, FBDIV, POSTDIV1 and POSTDIV2 values so that each PLL will generated clocks with expected frequency based on current chip mode.

During runtime, user can alter PLL's output by progamming DIV values inside PLL Control Registers.

Take DPLL0 configuration as an example:

1. Gate PLL output by clearing PLL Clock Enable Control Reg (0x70_300100C4) bit[4]
2. Modify DPLL0 Control Register (0x70_300100F8)
3. Polling PLL Status Register (0x70_300100C0) until: (1). PLL is locked again (bit[12] == 1) and (2). Updating sequence is finished (bit[4] == 0)
4. Un-gate PLL output clock by Setting PLL Clock Enable Control Reg (0x70_300100C4) bit[4]

When user programs the PLL Control Registers, internal hardware sequence is as figure *PLL hardware sequence*

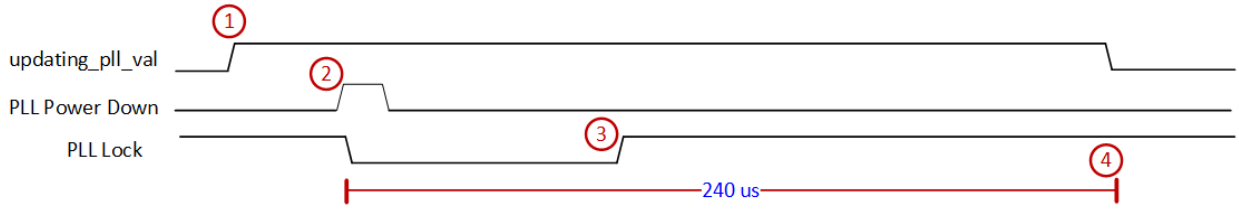

Fig. 2: PLL hardware sequence

1. The updating_pll_val bit is asserted immediately after user writes to PLL Control Registers, and user can check the value of this bit in PLL status register.
2. After hardware logic prepares the new DIV value for PLL, PLL's PD (Global Power Down) signal will be toggled so that PLL will work on the updated value.

3. PLL Lock goes high again when PLL's output is stable on new frequency.
4. Besides LOCK signal from PLL, internal logic will also wait for 240us then determine the modification sequence is finished and de-assert "updating_pll_val" bit.

User should keep polling PLL Status Register so as to ensure "updating_pll_val" bit field is de-asserted and whole sequece is finished. When the sequence is ongoing, internal logic will prevent initiating another modification.

4.3 Clock gate

When users modify the frequency of a PLL, the output frequency may overshoot/exceed the expected frequency before PLL finally gets stable. This will lead to unwanted behavior or errors.

So output clock of PLL should be gated during configuration.

The generation of clock enable signal is shown as figure *Clock gate*

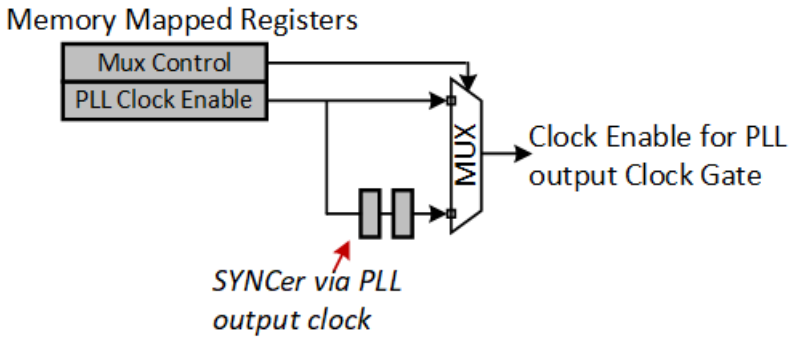

Fig. 3: Clock gate

User is able to control the above logic (Enable/MUX) by programming PLL Clock Enable Control Register (0x70_3001_00C4).

Both original PLL Clock Enable register and its synced version can be selected as Clock Enable PLL. This is because when PLL's frequency overshoots, the synchonizer may fail to work. There has to be a backup path.

Note that the address of register for controlling PLL Gating shall not be the same as those mentioned in previous section. Cos, once you touch the DIV related register, PLL will be powered-down.

4.4 Clock tree

SG2042 TOP level clock structure is shown as figure *Clock tree*

4.5 Default clock frequency

There three clock modes controlled by clock mode pins. They are safe, normal and fast modes. The tree mode select pins are listed in table *Digital pin functions*

The status of mode select pins is show in table *Mode select*

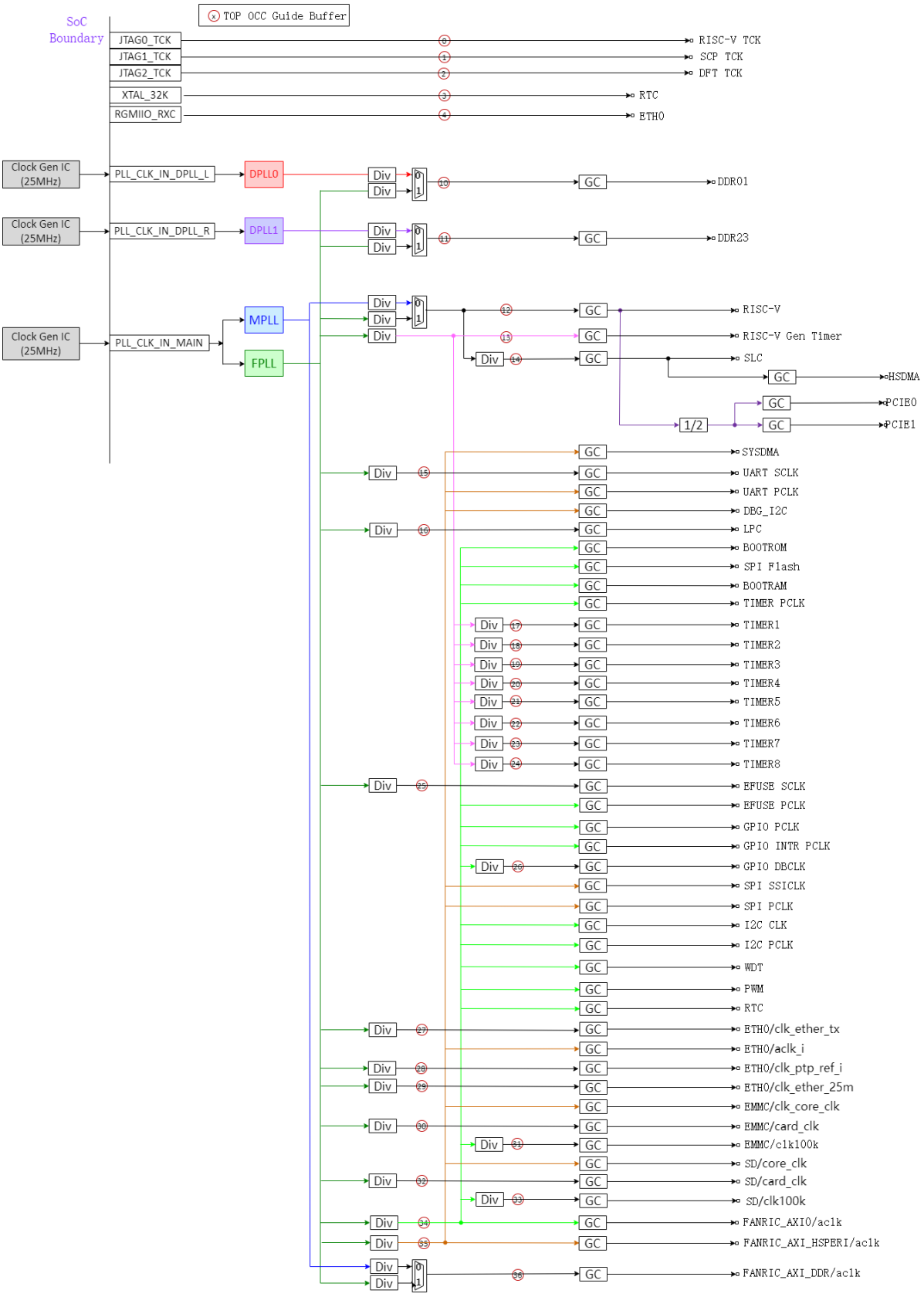

Fig. 4: Clock tree

Table 2: Mode select

MODE_SEL2	MODE_SEL1	MODE_SEL0	MODE
x	0	0	Normal
x	0	1	Fast
x	1	0	Safe
x	1	1	Bypass

MODE_SEL2 pin is not used.

The default clock frequency is show as table *Default clock frequency*

Table 3: Default clock frequency

Clock	NORMAL(MHz)	FAST(MHz)	SAFE(MHz)	BYPASS(MHz)
MPLL	1600	2000	1000	25
FPLL	1000	1000	1000	25
DPLL0	1200	1600	800	25
DPLL1	1200	1600	800	25
RISC-V CPU	1600	2000	1000	25
RISC-V Timer	50	50	50	25
SLC	800	1000	500	25
SYSDMA	250	250	250	25
UART SCLK	500	500	250	25
UART PCLK	250	250	250	25
DBG_I2C	250	250	250	25
LPC	200	200	200	25
BOOTROM	100	100	100	25
SPI Flash	100	100	100	25
BOOTRAM	100	100	100	25
TIMER PCLK	100	100	100	25
TIMER1	50	50	50	25
TIMER2	50	50	50	25
TIMER3	50	50	50	25
TIMER4	50	50	50	25
TIMER5	50	50	50	25
TIMER6	50	50	50	25
TIMER7	50	50	50	25
TIMER8	50	50	50	25
EFUSE CLK	25	25	25	25
EFUSE PCLK	100	100	100	25
GPIO PCLK	100	100	100	25
GPIO INTR CLK	100	100	100	25
GPIO DBCLK	0.1	0.1	0.1	0.1
SPI SSI	250	250	250	25
SPI PCLK	250	250	250	25
IIC ICCLK	100	100	100	25
IIC PCLK	100	100	100	25
WDT	100	100	100	25
PWM	100	100	100	25
RTC	100	100	100	25
PCIE0/1/2/3	800	1000	500	25

continues on next page

Table 3 – continued from previous page

Clock	NORMAL(MHz)	FAST(MHz)	SAFE(MHz)	BYPASS(MHz)
HSDMA	800	1000	500	25
DDR0/1/2/3	1200	1600	800	25

4.6 Registers

There are a set of control registers to control SoC clocks. Clock gate, divider, mux registers are described in this chapter. PLL controll registers are described in chapter *System control*.

The base address is listed in table *Memory map*, CLOCK device.

4.6.1 CLKENREG0: offset 0x0000

Table 4: Clock enable register 0

Bits	Attribute	Default	Description
31	RW	0x1	Clock Enable for clk_axi_eth0 (1: Enable; 0: Gate)
30	RW	0x1	Clock Enable for clk_tx_eth0 (1: Enable; 0: Gate)
29	RW	0x1	Clock Enable for clk_apb_rtc (1: Enable; 0: Gate)
28	RW	0x1	Clock Enable for clk_apb_pwm (1: Enable; 0: Gate)
27	RW	0x1	Clock Enable for clk_apb_wdt (1: Enable; 0: Gate)
26	RW	0x1	Clock Enable for clk_apb_i2c (1: Enable; 0: Gate)
25	RW	0x1	Clock Enable for clk_apb_spi (1: Enable; 0: Gate)
24	RW	0x1	Clock Enable for clk_gpio_db (1: Enable; 0: Gate)
23	RW	0x1	Clock Enable for clk_apb_gpio_intr (1: Enable; 0: Gate)
22	RW	0x1	Clock Enable for clk_apb_gpio (1: Enable; 0: Gate)
21	RW	0x1	Clock Enable for clk_apb_efuse (1: Enable; 0: Gate)
20	RW	0x1	Clock Enable for clk_efuse (1: Enable; 0: Gate)
19	RW	0x1	Clock Enable for clk_timer_8 (1: Enable; 0: Gate)
18	RW	0x1	Clock Enable for clk_timer_7 (1: Enable; 0: Gate)
17	RW	0x1	Clock Enable for clk_timer_6 (1: Enable; 0: Gate)
16	RW	0x1	Clock Enable for clk_timer_5 (1: Enable; 0: Gate)
15	RW	0x1	Clock Enable for clk_timer_4 (1: Enable; 0: Gate)
14	RW	0x1	Clock Enable for clk_timer_3 (1: Enable; 0: Gate)
13	RW	0x1	Clock Enable for clk_timer_2 (1: Enable; 0: Gate)
12	RW	0x1	Clock Enable for clk_timer_1 (1: Enable; 0: Gate)
11	RW	0x1	Clock Enable for clk_apb_timer (1: Enable; 0: Gate)
10	RW	0x1	Clock Enable for clk_axi_sram (1: Enable; 0: Gate)
9	RW	0x1	Clock Enable for clk_ahb_sf (1: Enable; 0: Gate)
8	RW	0x1	Clock Enable for clk_ahb_rom (1: Enable; 0: Gate)
7	RW	0x1	Clock Enable for clk_ahb_lpc (1: Enable; 0: Gate)
6	RW	0x1	Clock Enable for clk_axi_dbg_i2c (1: Enable; 0: Gate)
5	RW	0x1	Clock Enable for clk_apb_uart (1: Enable; 0: Gate)
4	RW	0x1	Clock Enable for clk_uart_500m (1: Enable; 0: Gate)
3	RW	0x1	Clock Enable for clk_sysdma_axi (1: Enable; 0: Gate)
2	RW	0x1	Clock Enable for clk_slc (1: Enable; 0: Gate)
1	RW	0x1	Clock Enable for clk_scp_timer (1: Enable; 0: Gate)
0	RW	0x1	Clock Enable for clk_rp_cpu_normal (1: Enable; 0: Gate)

4.6.2 CLKENREG1: offset 0x0004

Table 5: Clock enable register 1

Bits	Attribute	Default	Description
31:16	RW	NA	Reserved
15	RW	0x1	Clock Enable for clk_ddr23 (1: Enable; 0: Gate)
14	RW	0x1	Clock Enable for clk_ddr01 (1: Enable; 0: Gate)
13	RW	0x1	Clock Enable for clk_axi_ddr (1: Enable; 0: Gate)
12	RW	0x1	Clock Enable for clk_top_axi_hsperi (1: Enable; 0: Gate)
11	RW	0x1	Clock Enable for clk_top_axi0 (1: Enable; 0: Gate)
10	RW	0x1	Clock Enable for clk_hsdma (1: Enable; 0: Gate)
9	RW	0x1	Clock Enable for clk_axi_pcie1 (1: Enable; 0: Gate)
8	RW	0x1	Clock Enable for clk_axi_pcie0 (1: Enable; 0: Gate)
7	RW	0x1	Clock Enable for clk_100k_sd (1: Enable; 0: Gate)
6	RW	0x1	Clock Enable for clk_sd (1: Enable; 0: Gate)
5	RW	0x1	Clock Enable for clk_axi_sd (1: Enable; 0: Gate)
4	RW	0x1	Clock Enable for clk_100k_emmc (1: Enable; 0: Gate)
3	RW	0x1	Clock Enable for clk_emmc (1: Enable; 0: Gate)
2	RW	0x1	Clock Enable for clk_axi_emmc (1: Enable; 0: Gate)
1	RW	0x1	Clock Enable for clk_ref_eth0 (1: Enable; 0: Gate)
0	RW	0x1	Clock Enable for clk_ptp_ref_i_eth0 (1: Enable; 0: Gate)

4.6.3 CLKSELREG0: offset 0x0020

Table 6: Clock select register 0

Bits	Attribute	Default	Description
31:4	RW	NA	Reserved
3	RW	0x1	Clock Select for DDR23's clock core_ddrc_core_clk (aka clk_ddr23) 1: Select in_dpll1_clk as clock source 0: Select in_fpll_clk as clock source
2	RW	0x1	Clock Select for DDR01's clock core_ddrc_core_clk (aka clk_ddr01) 1: Select in_dpll0_clk as clock source 0: Select in_fpll_clk as clock source
1	RW	0x1	Clock Select for FABRIC_AXI_DDR's clock aclk (aka clk_axi_ddr) 1: Select in_mpll_clk as clock source 0: Select in_fpll_clk as clock source
0	RW	0x1	Clock Select for RP's clock top_rp_cpu_clk_normal (aka clk_rp_cpu_normal) 1: Select in_mpll_clk as clock source 0: Select in_fpll_clk as clock source

4.6.4 CLKDIVREG0: offset 0x0040

Table 7: Clock divider 0 control of RISC-V core

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:5	RW	NA	Reserved
4	RW	0x0	Clock Enable for this Branch Divider 0: Gate this Branch Divider 1: Enable this Branch Divider
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.5 CLKDIVREG1: offset 0x0044

Table 8: Clock divider 1 control of RISC-V core

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:5	RW	NA	Reserved
4	RW	0x0	Clock Enable for this Branch Divider 0: Gate this Branch Divider 1: Enable this Branch Divider
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.6 CLKDIVREG2: offset 0x0048

Table 9: Clock divider control of SCP timer

Bits	Attribute	Default	Description
31:24	RW	NA	Reserved
23:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.7 CLKDIVREG3: offset 0x004c

Table 10: Clock divider control of SLC

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.8 CLKDIVREG4: offset 0x0050

Table 11: Clock divider control of UART

Bits	Attribute	Default	Description
31:23	RW	NA	Reserved
22:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.9 CLKDIVREG5: offset 0x0054

Table 12: Clock divider control of LPC

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.10 CLKDIVREG6: offset 0x0058

Table 13: Clock divider control of TIMER1

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.11 CLKDIVREG7: offset 0x005c

Table 14: Clock divider control of TIMER2

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.12 CLKDIVREG8: offset 0x0060

Table 15: Clock divider control of TIMER3

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.13 CLKDIVREG9: offset 0x0064

Table 16: Clock divider control of TIMER4

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.14 CLKDIVREG10: offset 0x0068

Table 17: Clock divider control of TIMER5

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.15 CLKDIVREG11: offset 0x006c

Table 18: Clock divider control of TIMER6

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.16 CLKDIVREG12: offset 0x0070

Table 19: Clock divider control of TIMER7

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.17 CLKDIVREG13: offset 0x0074

Table 20: Clock divider control of TIMER8

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.18 CLKDIVREG14: offset 0x0078

Table 21: Clock divider control of eFuse

Bits	Attribute	Default	Description
31:23	RW	NA	Reserved
22:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.19 CLKDIVREG15: offset 0x007c

Table 22: Clock divider control of GPIO DB

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.20 CLKDIVREG16: offset 0x0080

Table 23: Clock divider control of ETH TX

Bits	Attribute	Default	Description
31:27	RW	NA	Reserved
26:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.21 CLKDIVREG17: offset 0x0084

Table 24: Clock divider control of ETH PTP

Bits	Attribute	Default	Description
31:24	RW	NA	Reserved
23:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.22 CLKDIVREG18: offset 0x0088

Table 25: Clock divider control of ETH

Bits	Attribute	Default	Description
31:24	RW	NA	Reserved
23:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.23 CLKDIVREG19: offset 0x008c

Table 26: Clock divider control of eMMC

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.24 CLKDIVREG20: offset 0x0090

Table 27: Clock divider control of eMMC 100k

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.25 CLKDIVREG21: offset 0x0094

Table 28: Clock divider control of SDIO

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.26 CLKDIVREG22: offset 0x0098

Table 29: Clock divider control of SDIO 100k

Bits	Attribute	Default	Description
31:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.27 CLKDIVREG23: offset 0x009c

Table 30: Clock divider control of TOP AXI0

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RO	0x0	Select Divide Factor from Register This bit is reserved for this divider.
2	RO	0x0	Select High Wide Control from Register This bit is reserved for this divider.
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RO	0x1	Divider Reset Control This bit is reserved for this divider.

4.6.28 CLKDIVREG24: offset 0x00a0

Table 31: Clock divider control of TOP AXI0

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:4	RW	NA	Reserved
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.29 CLKDIVREG25: offset 0x00a4

Table 32: Clock divider 0 control of AXI DDR

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:5	RW	NA	Reserved
4	RW	0x0	Clock Enable for this Branch Divider 0: Gate this Branch Divider 1: Enable this Branch Divider
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.30 CLKDIVREG26: offset 0x00a8

Table 33: Clock divider 1 control of AXI DDR

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:5	RW	NA	Reserved
4	RW	0x0	Clock Enable for this Branch Divider 0: Gate this Branch Divider 1: Enable this Branch Divider
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.31 CLKDIVREG27: offset 0x00ac

Table 34: Clock divider 0 control of DDR01

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:5	RW	NA	Reserved
4	RW	0x0	Clock Enable for this Branch Divider 0: Gate this Branch Divider 1: Enable this Branch Divider
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.32 CLKDIVREG28: offset 0x00b0

Table 35: Clock divider 1 control of DDR01

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:5	RW	NA	Reserved
4	RW	0x0	Clock Enable for this Branch Divider 0: Gate this Branch Divider 1: Enable this Branch Divider
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.33 CLKDIVREG29: offset 0x00b4

Table 36: Clock divider 0 control of DDR23

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:5	RW	NA	Reserved
4	RW	0x0	Clock Enable for this Branch Divider 0: Gate this Branch Divider 1: Enable this Branch Divider
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

4.6.34 CLKDIVREG30: offset 0x00b8

Table 37: Clock divider 1 control of DDR23

Bits	Attribute	Default	Description
31:21	RW	NA	Reserved
20:16	RW	0x0	Clock Divider Factor
15:5	RW	NA	Reserved
4	RW	0x0	Clock Enable for this Branch Divider 0: Gate this Branch Divider 1: Enable this Branch Divider
3	RW	0x0	Select Divide Factor from Register 0: Select initial value 1: Select Divide Factor from this register
2	RW	0x0	Select High Wide Control from Register 0: Select initial value 1: Select High Wide from this register
1	RW	0x0	High Wide Control (when Divider Factor is odd) 0: Low level of the clock is wider 1: High level of the clock is wider
0	RW	0x1	Divider Reset Control 0: Assert Reset 1: De-assert Reset

CHAPTER

FIVE

RESET

5.1 SG2042 Reset Overview and Sequence

SG2042 system reset is controlled by two chip IO: SYS_RST_X and PWR_BUTTON (reserved in SG2042, tie 1). These IOs all have schmitt trigger to improve the signal quality.

When SYS_RST_X is asserted (pull low), SG2042 will start the reset sequence as shown below.

External system reset is first debounced by slow clock for about 30ms to further avoid false reset caused by glitch on the signal input.

This 30ms also ensure the reference clock input of PLL goes stable after chip power on, which needs at least 1ms.

When the root reset release is released, the PLL will get out of power down mode and start to work.

Note that as we have BISR (built-in-self-repair) in SG2042, only after BISR operation finishes, will the chip continue following sequence.

Reset for clock divider, reset for the system and reset for the AP will be released one by one after all PLLs are locked .

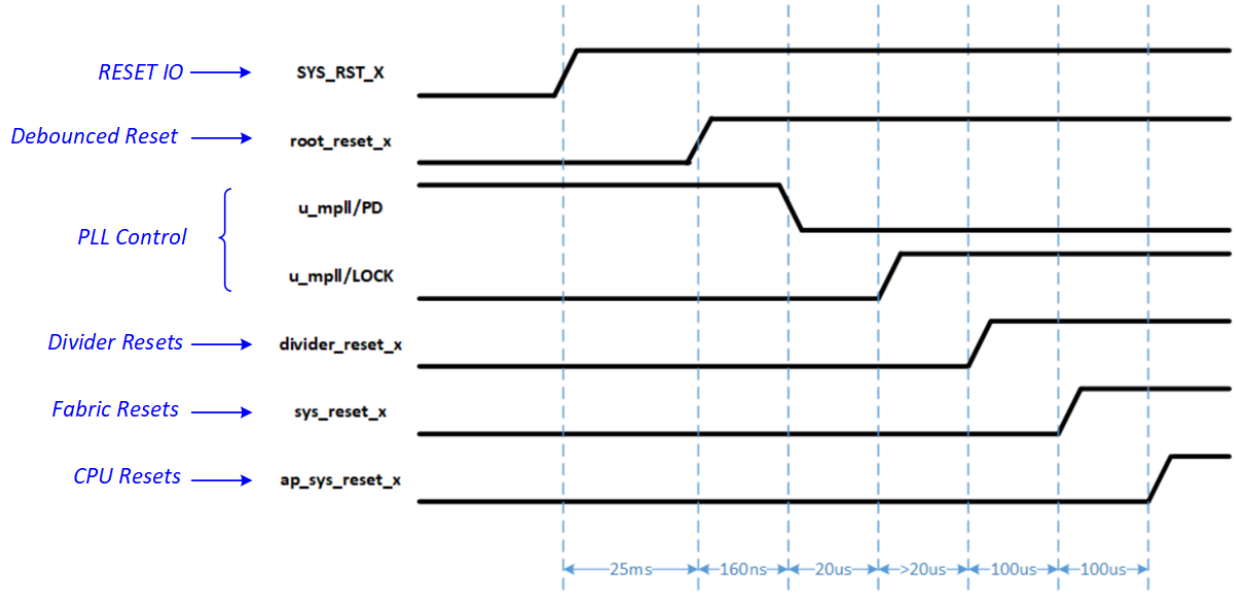

Fig. 1: image2021

During chip runtime, multiple sources are able to trigger the global reset: watch dog reset and ARM warm reset.

These resets may assert PLL power down, reset for clock divider, reset for the system and reset for the AP/RP active. The whole chip will be reset under this condition, except tiny logic related to REFCLK, IO reset.

These software resets must be active for at least 1us before take effect to ensure PLL is reset correctly.

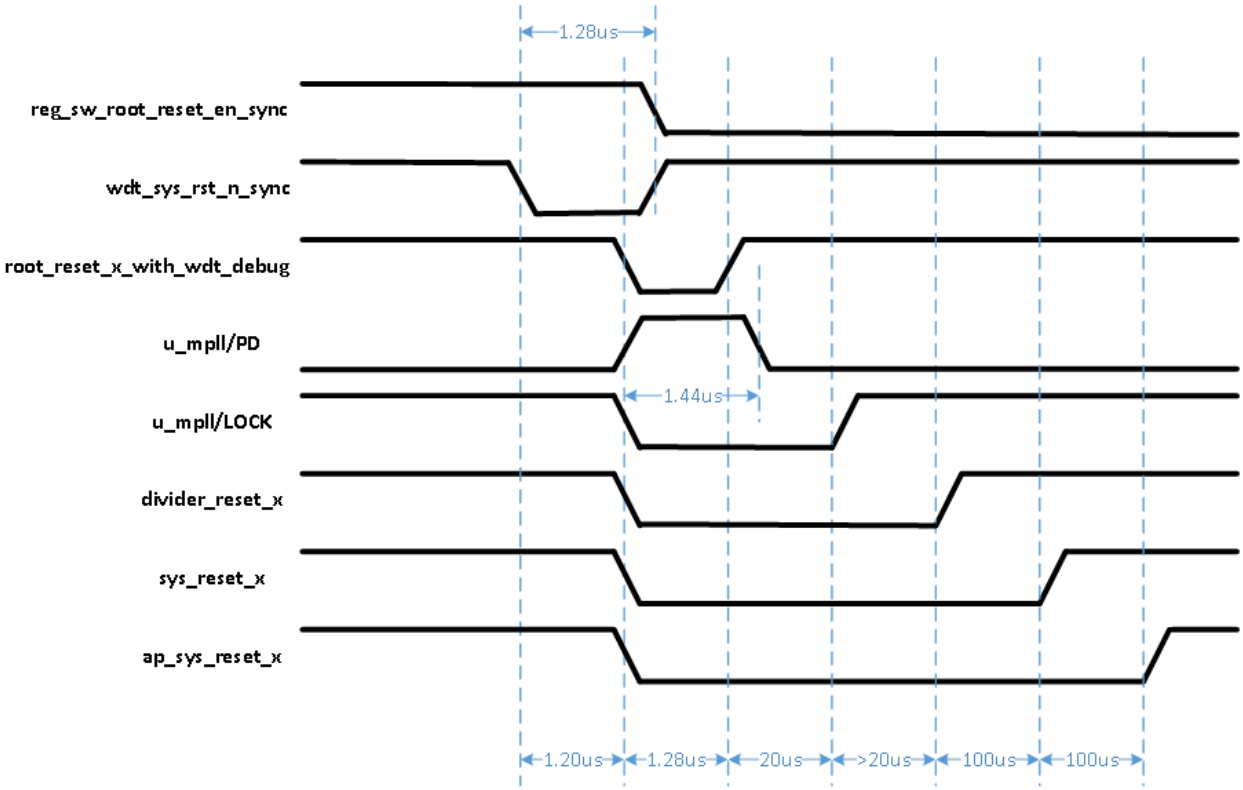

Fig. 2: wdt reset

5.2 Soft Reset

Every module in SG2042 owns a dedicated soft reset control bit for software usage.The relevant register information is as follows.

The base address is listed in table *Memory map*, RESET device.

Table 1: Software Reset Register 0 (0x00000000)

Bits	Attribute	Default	Description
31	RW	0x1	Software Reset for uart2 (Active Low)
30	RW	0x1	Software Reset for uart1 (Active Low)
29	RW	0x1	Software Reset for uart0 (Active Low)
28	RW	0x1	Software Reset for sd (Active Low)
27	RW	0x1	Software Reset for emmc (Active Low)
26	RW	0x1	Software Reset for eth0 (Active Low)
25	RW	0x1	Software Reset for lpc (Active Low)
24	RW	0x1	Software Reset for sf1 (Active Low)
23	RW	0x1	Software Reset for sf0 (Active Low)
22	RW	0x1	Software Reset for axi_sram1 (Active Low)

continues on next page

Table 1 – continued from previous page

Bits	Attribute	Default	Description
21	RW	0x1	Software Reset for axi_sram0 (Active Low)
20	RW	0x1	Software Reset for pwm (Active Low)
19	RW	0x1	Software Reset for gpio2 (Active Low)
18	RW	0x1	Software Reset for gpio1 (Active Low)
17	RW	0x1	Software Reset for gpio0 (Active Low)
16	RW	0x1	Software Reset for i2c3 (Active Low)
15	RW	0x1	Software Reset for i2c2 (Active Low)
14	RW	0x1	Software Reset for i2c1 (Active Low)
13	RW	0x1	Software Reset for i2c0 (Active Low)
12	RW	0x1	Software Reset for ahb_rom1 (Active Low)
11	RW	0x1	Software Reset for ahb_rom0 (Active Low)
10	RW	0x1	Software Reset for wdt (Active Low)
9	RW	0x1	Software Reset for timer (Active Low)
8	RW	0x1	Software Reset for rtc (Active Low)
7	RW	0x1	Software Reset for efuse1 (Active Low)
6	RW	0x1	Software Reset for efuse0 (Active Low)
5	RW	0x1	Software Reset for sysdma (Active Low)
4	RW	0x1	Software Reset for hsdma (Active Low)
3	RW	0x1	Software Reset for rp_sys cmn (Active Low)
2	RW	0x1	Software Reset for rp_sys low speed logic (Active Low)
1	RW	0x0	Software Reset for rp_sys cpu (Active Low)
0	RW	0x1	Software Reset for ap_sys (Active Low)

Table 2: Software Reset Register 1 (0x00000004)

Bits	Attribute	Default	Description
31	RW	0x1	Software Reset for RXU (Active Low)
30	RW	0x1	Software Reset for RXU (Active Low)
29	RW	0x1	Software Reset for RXU (Active Low)
28	RW	0x1	Software Reset for RXU (Active Low)
27	RW	0x1	Software Reset for RXU (Active Low)
26	RW	0x1	Software Reset for RXU (Active Low)
25	RW	0x1	Software Reset for RXU (Active Low)
24	RW	0x1	Software Reset for RXU (Active Low)
23	RW	0x1	Software Reset for RXU (Active Low)
22	RW	0x1	Software Reset for RXU (Active Low)
21	RW	0x1	Software Reset for RXU (Active Low)
20	RW	0x1	Software Reset for RXU (Active Low)
19	RW	0x1	Software Reset for RXU (Active Low)
18	RW	0x1	Software Reset for RXU (Active Low)
17	RW	0x1	Software Reset for RXU (Active Low)
16	RW	0x1	Software Reset for RXU (Active Low)
15	RW	0x1	Software Reset for RXU (Active Low)
14	RW	0x1	Software Reset for RXU (Active Low)
13	RW	0x1	Software Reset for RXU (Active Low)
12	RW	0x1	Software Reset for fau2 (Active Low)
11	RW	0x1	Software Reset for fau1 (Active Low)
10	RW	0x1	Software Reset for fau0 (Active Low)
9	RW	0x1	Software Reset for ddr3 (Active Low)

continues on next page

Table 2 – continued from previous page

Bits	Attribute	Default	Description
8	RW	0x1	Software Reset for ddr2 (Active Low)
7	RW	0x1	Software Reset for ddr1 (Active Low)
6	RW	0x1	Software Reset for ddr0 (Active Low)
5	RW	0x1	Software Reset for pcie1 (Active Low)
4	RW	0x1	Software Reset for pcie0 (Active Low)
3	RW	0x1	Software Reset for dbg_i2c (Active Low)
2	RW	0x1	Software Reset for spi1 (Active Low)
1	RW	0x1	Software Reset for spi0 (Active Low)
0	RW	0x1	Software Reset for uart3 (Active Low)

Table 3: Software Reset Register 2 (0x00000008)

Bits	Attribute	Default	Description
31:13	RW	NA	Reserved
12	RW	0x1	Software Reset for RXU (Active Low)
11	RW	0x1	Software Reset for RXU (Active Low)
10	RW	0x1	Software Reset for RXU (Active Low)
9	RW	0x1	Software Reset for RXU (Active Low)
8	RW	0x1	Software Reset for RXU (Active Low)
7	RW	0x1	Software Reset for RXU (Active Low)
6	RW	0x1	Software Reset for RXU (Active Low)
5	RW	0x1	Software Reset for RXU (Active Low)
4	RW	0x1	Software Reset for RXU (Active Low)
3	RW	0x1	Software Reset for RXU (Active Low)
2	RW	0x1	Software Reset for RXU (Active Low)
1	RW	0x1	Software Reset for RXU (Active Low)
0	RW	0x1	Software Reset for RXU (Active Low)

CHAPTER

SIX

POWER DOMAIN AND POWER SEQUENCE

SG2042 implements single-voltage design to minimize the design effort. Different blocks will work at identical voltage (core: 0.8v) to alleviate further implementation.

So there will be no content on power domain partition, power good indication, isolation and level shifter insertion, etc. The following sections focus on power-up sequence.

Table 1: power bump

	Power Bump	Voltage	Variation	Comment
Top	VDDC	0.8	10%	Core Power
PLL	VDDIO_FPLL	0.8	10%	PLL Power
	VDDIO_MPLL			
	VDDIO_DPLL0			
	VDDIO_DPLL1			
eFUSE	VDD_EFUSE	0.8	10%	
PCIe	cmn_avdd_clk	0.8	0.76v~0.88v	pma power
	rx_avdd_clk_ln_0~15			
	tx_avdd_ln_0~15	0.8	0.76v~0.88v	pma power
	rx_avdd_ln_0~15			
	cmn_avdd			
DDR	VDD	0.8	0.72v~0.88v	core supply
	VDDPLL	0.8	0.72v~0.88v	PLL control supply
	VDDQ	1.2	1.14v~1.26v	I/O drive supply
	VDDQCK	1.2	1.14v~1.26v	I/O drive supply
DDR4	VDDQ	1.2	1.14v~1.26v	DQ Power supply
	VDD	1.2	1.14v~1.26v	Power supply
	cmn_avdd_h	1.5/1.8	1.425v~1.98v	pma power
	xcvr_avdd_h_ln_0~15			
	VQPS	1.8	5%	efuse 1.8v for programming
IO	VDDIO_EMMC_18	1.8	10%	PHY power for EMMC and SD
	VDDIO_SENSOR	1.8	10%	
	VPP	2.5	2.375v~2.75v	DRAM Activating Power supply
	VDDIO_EMMC_33	3.3	10%	PHY power for EMMC and SD
	VDDIO_RGM_33	1.8	10%	1*RGMII 1682 Test data 11mA
	VDDIO_RGM_18	1.8	10%	1*RGMII 1682 Test data 11mA

6.1 Power Up Sequence

6.1.1 DDR Power-Up Sequence

The DDR spec requires the core supply should be sequenced on before, or concurrently with the IO supply.

The recommended power up sequence is:

1. Turn on core supply (VDD) and PLL supply (VDDPLL)
2. Assert rst_n and dll_rst_n ports on the PHY
3. Turn on IO supply (VDDQ, VDDQCK, VDDQX)

The Power Up requirement of particles

1. The power voltage ramp time between 300mV to VDD min must be no greater than 200ms
2. Power on sequence is VPP $\rightarrow$ VDD $\rightarrow$ VDDQ

6.1.2 PCIe Power-Up Sequence

PMA power supply has no sequence requirements.(see integration guide 4.3.4.3)

Based on Vendor's reply: There are no power supply sequence requirements for PHY/Controller, power supplies can be enabled in any order.

6.1.3 eFUSE Power-Up Sequence

The recommended power up sequence of eFUSE is

1. Turn on 0.8v VDD core
2. Turn on 1.8v VQPS

6.1.4 IO Power-Up Sequence

1.8V tphn12ffcllgv18e

1. Turn on 1.8v VDDPST I/O power
2. Turn on 0.8v vdd core power

   (Based on app note: power up I/O power and core power simultaneously is also acceptable)

3.3/1.8 tphn12ffcll_18od33rgmii

- In 3.3V mode, First power up VDDPST18 through PVDD18RGM; after at least 20us, power up VDDPST33 through PVDD3CDGRGM
- In 1.8V mode, VDDPST33 is 1.8V
- In our chip, only 1.8V mode is used.
- based on zhuoming's feedback: VDDPST18 and VDDPST33 can be power on simultaneous or can connect together in PCB if only used at 1.8V mode.
- MS1/MS2 should come form always-on core main.But as there are POC control in IO,when VDD not power on, MS1/MS2 will be set to a 1.8V mode. So these is no problem at the case.

3.3/1.8V tphn12ffcll_18od33sdio

- 0.8V VDD core -> 3.3/1.8V power up.

- based on zhuoming's feedback: MS signal should come form a always-on core domain. If no such always-on core domain. You can refer to option 3. (which is power up VDD core first)

6.1.5 Whole Chip Power-Up Sequence

0ms VDDIO (1.8v VDDPST) VDDIO_RGM_33, VDDIO_RGM_18 ->

1ms VDDC (0.8V) →

2ms VDDIO_EMMC_33 (3.3V) ->

3ms VDD_PCIE (0.8V PCIe PHY +PCIe Controller), VDD_PMA (1.5v/1.8v); DDR VDD (0.8v), DDR PLL (0.8v VDDPLL) ->

4ms DDR IO supply (1.2V, VDDQ,VDDQCK, VDDQX) ->

5ms VQPS (1.8V)->

6ms Release IO reset signals: SYS_RST_X and PWR_BUTTON

CHAPTER

SEVEN

LOW POWER

SG2042 implements dynamic and static clock gating for power saving.

This page is mainly for dynamic clock gating.

7.1 Fabric Auto Clock Gating

SG2042 uses the Low Power Controller (LPC) to dynamically clock gate Fabric Components:

- The Clock originally assigned to the Fabric and its downstream Register Slice passes through the clock-gate logic in the LPC first. Registers can be configured to bypass these clocks without performing any operation.
- LPC can send low-power requests to the Fabric through LPI (Low Power Interface) after N cycles are not transmitted on the Fabric bus (N is register configurable). If the Fabric responds to the request, The LPC will gate the clock.
- Once there is an upstream transmission (LPC finds awvalid, arvalid pulls up), LPC will open the clock.

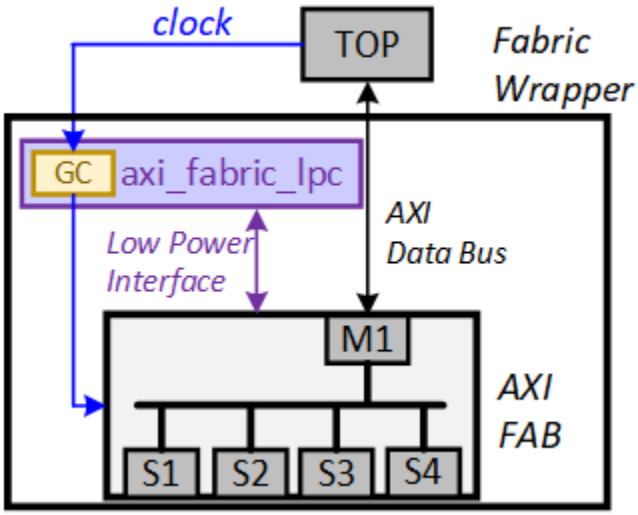

Fig. 1: Fabric Auto Clock Gating

7.2 Low Power Interface Signals

Table 1: low power inferface signals

Signal	Source	Description
CSYSREQ	Clock controller	Low power requirements of the system. This signal is a request from the system clock controller to enter a low power state.
CSYSACK	Peripheral device	Confirm the low power requirements. The signal is a low power request from the peripheral confirming the system.
CACTIVE	Peripheral device	The clock works. This signal indicates that the peripheral needs its clock signal:1=Need a peripheral clock 0=No peripheral clock required

7.3 Address of LPC Registers

All registers that control clock gate are in sys ctrl:

Among them:

All LPCS are divided into two different domains, and all interfaces in the domain are controlled by a corresponding register.For example, a register in the top clk domain controls all the LPC interfaces in the top clk domain.

Table 2: related lpc

Clock Domain	Related LPC
top clk domain	Fab_lpc7 Fab_lpc8 Fab_lpc9 Fab_lpc10 Fab_lpc11 Fab_lpc12 Fab_lpc13 Fab_lpc14 Fab_lpc15
hsperi clk domain	Fab_lpc1 Fab_lpc2 Fab_lpc3 Fab_lpc4 Fab_lpc5 Fab_lpc6 Fab_lpc16 Fab_lpc17

Table 3: address of lpc registers

Register	Address
top_fab_gate_enable	address=0x7030010000+0x20:[bit 0]
hsperi_fab_gate_enable	address=0x7030010000+0x20:[bit 1]
top_fab_gate_cnt_cycle	address=0x7030010000+0x24:[bit:(07:00)]
hsperi_fab_gate_cnt_cycle	address=0x7030010000+0x24:[bit:(15:08)]
Note: LPC control of AP is not supported	

7.4 Program Guide

- set the value in top/hsperi/ap_fab_gate_cnt_cycle register
- set the auto enable of the reg_auto_gate_ena register
- if the bus have no data to transfer,and after the cnt cycle , the fabric will be gate. if the bus is transferring data , the LPC wait the idle of the bus and gate

CHAPTER

EIGHT

PWM AND FAN

8.1 Overview

The pwmx4 contains two functions:

- Generate PWM waveforms.
- Detect the number of pulses over a period of time.

Figure 1 provides the overview of pwmx4. All control of the pwmx4 is performed via the APB interface.

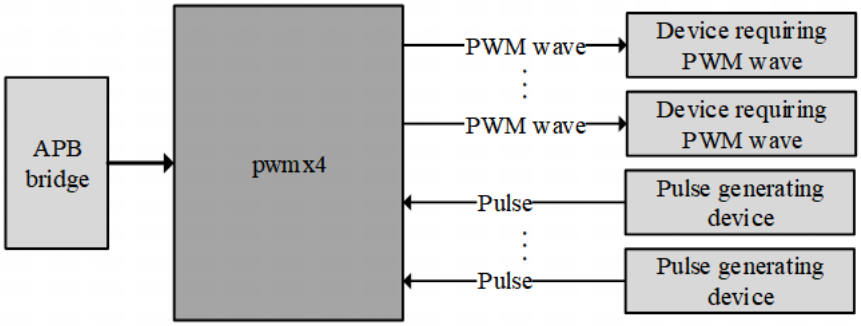

Fig. 1: pwmx4 Overview

8.1.1 Features and Parameters

The pwmx4 provides the following features:

- 4 PWM wave output channel.
- 4 Pulse detect channel.
- Ability to filter glitch that may be included in the pulse.
- All configurations and results are read and written via the APB interface.

8.1.2 Terms and Abbreviations

PWM Pulse-width modulation

8.2 Top Interface

8.2.1 Interface Signal Description

Figure 2 illustrates the top connection of the module.

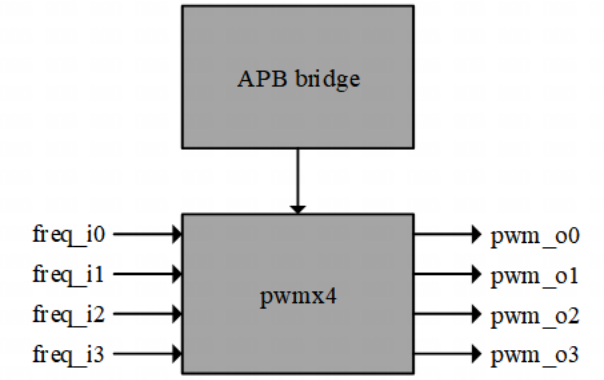

Fig. 2: Connection Overview

The detailed signals and their description are shown in Table 1.

Table 1: Top Interface

Name	I/O	Connection	Description
APB Signal			
p_clk	I	APB bridge	Clock from APB bridge. The only clock source of this module.
p_resetn	I	APB bridge	Async reset (active low) from APB bridge. The only reset source of this module
p_enable	I	APB bridge	APB enable signal.
p_write	I	APB bridge	APB write signal.
p_sel	I	APB bridge	APB sel signal.
p_addr[5:0]	I	APB bridge	APB address signal. Connected to the lower 6 bits of the APB address bus. The lowest 2 bits are not used.
p_wdata[31:0]	I	APB bridge	APB write data signal.
p_rdata[31:0]	O	APB bridge	APB read data signal.
Output Signal			
pwm_o0	O	PWM receiver	PWM channel 0 output.
pwm_o1	O	PWM receiver	PWM channel 1 output.
pwm_o2	O	PWM receiver	PWM channel 2 output.
pwm_o3	O	PWM receiver	PWM channel 3 output.
Input Signal			
freq_i0	I	Pulse source	Pulse detect channel 0 input.
freq_i1	I	Pulse source	Pulse detect channel 1 input.
freq_i2	I	Pulse source	Pulse detect channel 2 input.
freq_i3	I	Pulse source	Pulse detect channel 3 input.

8.3 Integration Requirement

8.3.1 Synchronization of clock and reset

p_clk is the only clock signal for this module.

p_resetn is the only reset signal for this module. There is no synchronization of reset_n inside the module. Therefore, p_resetn needs to be synchronized outside the module with p_clk.

8.4 Function Description

The pwmx4 contains two functions:

- Generate PWM waveforms.
- Detect the number of pulses over a period of time.

8.4.1 PWM generation

The generation of PWM waves is controlled by two registers, PERIODx and HLPERIODx. Where x can be 0, 1, 2, 3, corresponding to 4 PWM channels. Figure 3 shows how registers PERIODx and HLPERIODx affect the PWM waveform.

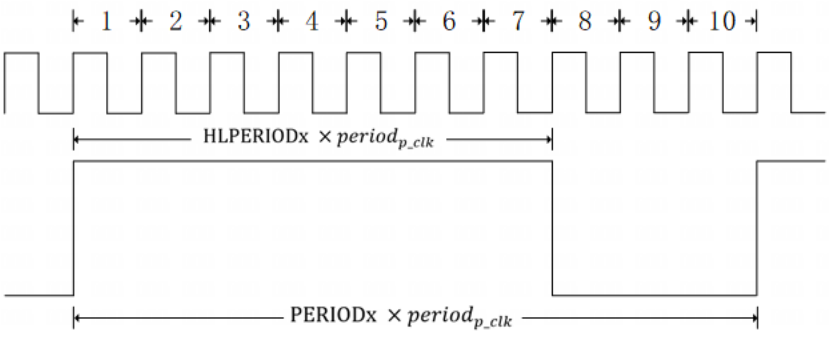

Fig. 3: The effect of registers on PWM waves

Period of PWM

The period of the output PWM wave is

$$period_{PWM} = PERIODx \times period_{p\_clk}$$

The range of PERIODx is 3 to 4294967295.

Attention: The register PERIODx can be written to all values in the range of 0 to 4294967295, but only the above range is legal. Exceeding this range may result in no output waveform or output waveform frequency error.

Duty Cycle of PWM

The duty cycle of the PWM wave is

$$DutyCycle = \frac{HLPERIODx}{PERIODx}$$

The range of HLPERIODx is 2 to 4294967295.

Attention: The register HLPERIODx can be written to all values in the range of 0 to 4294967295 and must be smaller than the corresponding PERIODx, but only the above range is legal. Exceeding this range may result in no output waveform or output waveform frequency error.

8.5 Pulse detection

The time of each pulse detection is controlled by the register FRExNUM, and the result is written to the FRExDATA register after each detection count is completed. The pulse detection circuit will always work cyclically.Figure 4 shows how FRExNUM affects the pulse detection time and how FRExDATA records the detection count results. Where x can be 0, 1, 2, 3, corresponding to 4 pulse detection channels.

The signal is passed through Glitch Filter to eliminate glitch before it is detected and counted.

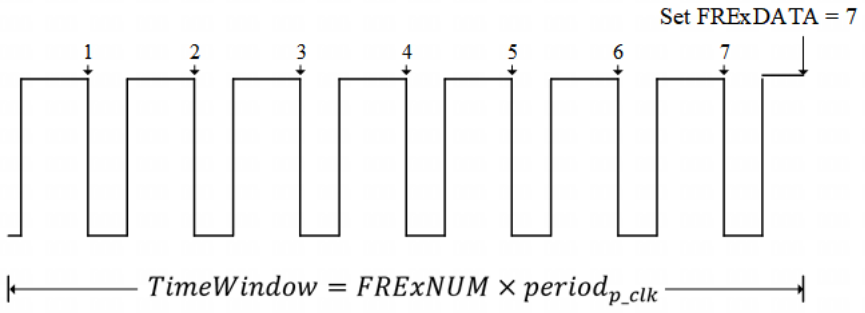

Fig. 4: The effect of registers on Pulse Detection

8.5.1 Time Window

The time window width of one pulse detection is

$$TimeWindow = FRExNUM \times period_{p\_clk}$$

The range of FRExNUM is 1 to 4294967295.

Attention: The register FRExNUM cannot be set to 0, otherwise the FRExDATA value will always be 0. But this does not mean that the circuit stops working, the circuit will continue to work and the power consumption will not decrease.

8.5.2 Glitch Filter

Pulse Detection Filters the input signal using the Glitch filter before counting the pulses. Figure 5 shows the role of Glitch Filter.

Any pulse narrower than GlitchWidth is defined as a glitch.Where

$$GlitchWidth = 16 \times period_{p\_clk}$$

The Glitch filter considers all pulses with a width greater than or equal to Glitchwidth to be valid, and the rest are glitch. And the glitch in the input signal will be removed and output.

Figure 6 shows how the glitch filter works according to GlitchWidth.

Attention 1: The width of the pulse must be greater than GlitchWidth to pass the glitch filter and be detected and counted.

Attention 2: A glitch with a width greater than GlitchWidth cannot be removed.

8.6 Internal Blocks

8.6.1 Partition Overview

The pwmx4 contains three modules internally, of which PWM Generator and Pulse Detector implement PWM waveform generation and pulse detection respectively. APB slave is used to read and write the internal registers of the other

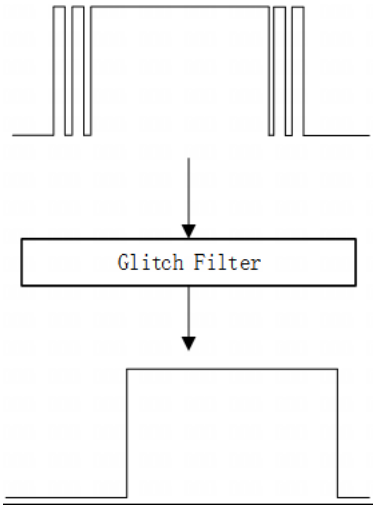

Fig. 5: The role of Glitch Filter

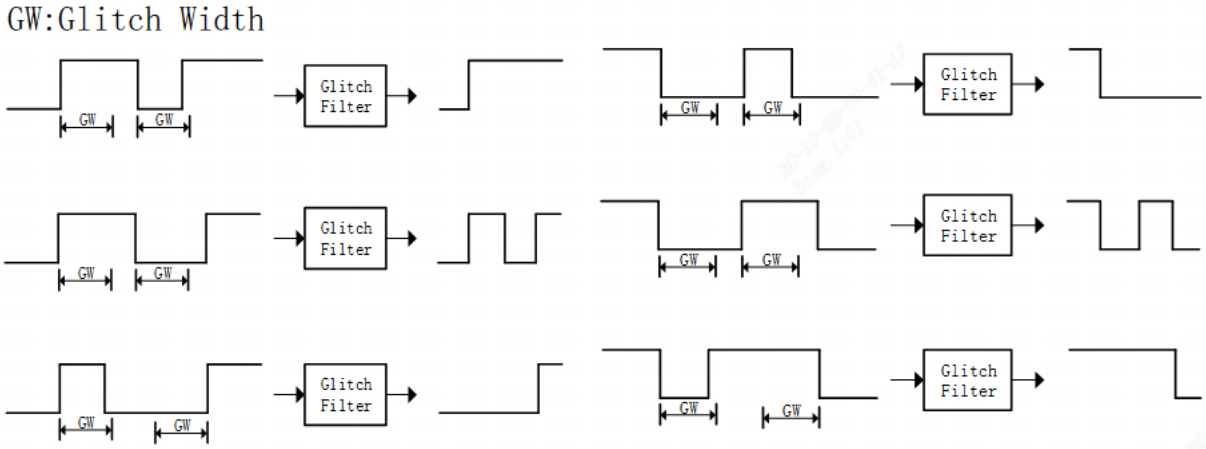

Fig. 6: The operation of the glitch filter on the rising and falling edge of the pulse

two modules. The implementation of the APB slave will be omitted below.

Figure 7 shows the internal parathion of pwmx4:

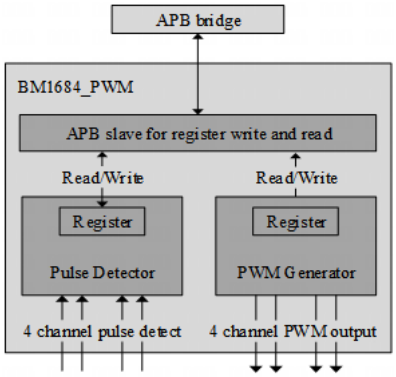

Fig. 7: The microarchitecture of pwmx4

8.6.2 PWM Generator Module

The PWM Generator provides 4 channels of PWM waveform output with frequency and duty cycle controlled by registers PERIODx and HLPERIODx. Where x can be 0, 1, 2, 3, corresponding to 4 channels. Figure 8 shows the microarchitecture of the PWM generator.

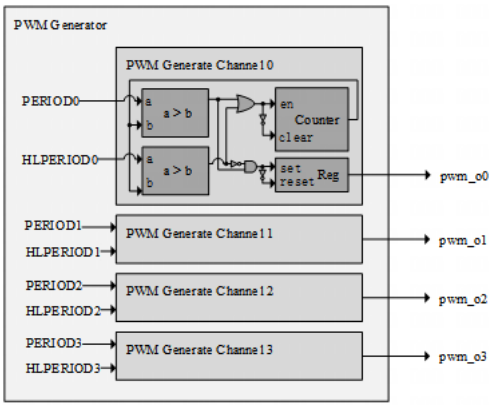

Fig. 8: The microarchitecture of the PWM generator

8.6.3 Pulse Detector Module

The Pulse Detector provides 4 channels of pulse detection input. The time window length of one test is configured by FREQxNUM, and the count result is automatically loaded into FREQxDATA. Where x can be 0, 1, 2, 3, corresponding to 4 channels.

The signal of each input pulse detection channel will enter the glitch filter first. The behavior of the glitch filter is detailed in the function description. This filter is based on a 16-bit shift register.

The signal processed by the glitch filter is converted to a single p_clk clock cycle pulse using a single-cycle pulse generator. This single-cycle pulse signal is used to enable the counter to complete the counting of the pulses.

When a frequently enabled counter is used for pulse detection, the detection result will be written to FREQxDATA when it reaches the time window length.

Figure 9 shows the microarchitecture of the pulse detector.

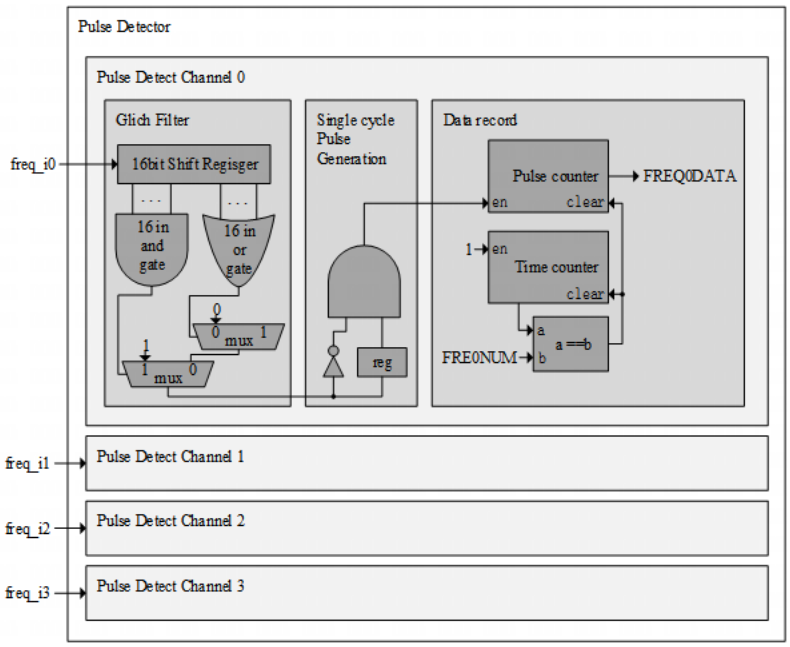

Fig. 9: The microarchitecture of the pulse detector

8.7 Register Definition

8.7.1 Memory Mapped Registers Summary

Table 2: Memory Mapped Registers Summary

Offset	Register Name	Default	Attribute
0x0000	HLPERIOD0	32'h00000000	RW
0x0004	PERIOD0	32'h00000000	RW
0x0008	HLPERIOD1	32'h00000000	RW
0x000C	PERIOD1	32'h00000000	RW
0x0010	HLPERIOD2	32'h00000000	RW
0x0014	PERIOD2	32'h00000000	RW
0x0018	HLPERIOD3	32'h00000000	RW
0x001C	PERIOD3	32'h00000000	RW
0x0020	FREQ0NUM	32'h00000000	RW
0x0024	FERQ0DATA	32'h00000000	RO
0x0028	FREQ1NUM	32'h00000000	RW
0x002C	FERQ1DATA	32'h00000000	RO
0x0030	FREQ2NUM	32'h00000000	RW
0x0034	FERQ2DATA	32'h00000000	RO
0x0038	FREQ3NUM	32'h00000000	RW
0x003C	FERQ3DATA	32'h00000000	RO

8.7.2 Register Description

HLPERIOD0(0x0000)

Table 3: HLPERIOD0(0x0000)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The time that the PWM wave inchannel 0 remains high for one cycle. The actual duration is p_clk clock period multiplied by this register value.

PERIOD0(0x0004)

Table 4: PERIOD0(0x0004)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The PWM wave period of channel 0 based on p_clk. The actual period is the p_clk period multiplied by this register value.

HLPERIOD1(0x0008)

Table 5: HLPERIOD1(0x0008)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The time that the PWM wave in channel 1 remains high for one cycle.The actual duration is p_clk clock period multiplied by this register value

PERIOD1(0x000C)

Table 6: PERIOD1(0x000C)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The PWM wave period of channel 1 based on p_clk. The actual period is the p_clk period multiplied by this register value.

HLPERIOD2(0x0010)

Table 7: HLPERIOD2(0x0010)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The time that the PWM wave in channel 2 remains high for one cycle.The actual duration is p_clk clock period multiplied by this register value

PERIOD2(0x0014)

Table 8: PERIOD2(0x0014)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The PWM wave period of channel 2 based on p_clk. The actual period is the p_clk period multiplied by this register value.

HLPERIOD3(0x0018)

Table 9: HLPERIOD3(0x0018)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The PWM wave period of channel 2 based on p_clk. The actual period is the p_clk period multiplied by this register value.

PERIOD3(0x001C)

Table 10: PERIOD3(0x001C)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The PWM wave period of channel 3 based on p_clk. The actual period is the p_clk period multiplied by this register value.

FREQ0NUM(0x0020)

Table 11: FREQ0NUM(0x0020)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The length of time used for the pulse detection channel 0. The actual length of time is the value of this register multiplied by the p_clk period

FEQ0DATA(0x0024)

Table 12: FEQ0DATA(0x0024)

Bit	Attribute	Default	Description
31:0	RO	32'h00000000	The number of pulses detected by pulse detection channel 0 within the length of time defined by register FRE0NUM.

FREQ1NUM(0x0028)

Table 13: FREQ1NUM(0x0028)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The length of time used for the pulse detection channel 1. The actual length of time is the value of this register multiplied by the p_clk period

FEQ1DATA(0x002C)

Table 14: FEQ1DATA(0x002C)

Bit	Attribute	Default	Description
31:0	RO	32'h00000000	The number of pulses detected by pulse detection channel 1 within the length of time defined by register FRE1NUM.

FREQ2NUM(0x0030)

Table 15: FREQ2NUM(0x0030)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The length of time used for the pulse detection channel 2. The actual length of time is the value of this register multiplied by the p_clk period

FEQ2DATA(0x0034)

Table 16: FEQ2DATA(0x0034)

Bit	Attribute	Default	Description
31:0	RO	32'h00000000	The number of pulses detected by pulse detection channel 1 within the length of time defined by register FRE1NUM.

FREQ3NUM(0x0038)

Table 17: FREQ3NUM(0x0038)

Bit	Attribute	Default	Description
31:0	RW	32'h00000000	The length of time used for the pulse detection channel 3. The actual length of time is the value of this register multiplied by the p_clk period

FEQ3DATA(0x003C)

Table 18: FEQ3DATA(0x003C)

Bit	Attribute	Default	Description
31:0	RO	32'h00000000	The number of pulses detected by pulse detection channel 3 within the length of time defined by register FRE3NUM.

8.8 Software Program Guide

The PWM generation and pulse detection functions are independent of each other. The following two sections will explain how to use these two functions.

8.8.1 PWM generation

Reset State

After reset, the output of the PWM channel is always high.

And the value of PERIODx is 0x00000000.

The value of HLPERIODx is 0x00000000.

Start PWM

The steps to start the PWM waveform output on one channel are as follows:

1.Configure PERIODx. The calculation formula for this value is given in Function Description.

2.Configure HLPERIODx. The calculation formula for this value is given in Function Description. The PWM wave output does not stop until if is no other operation.

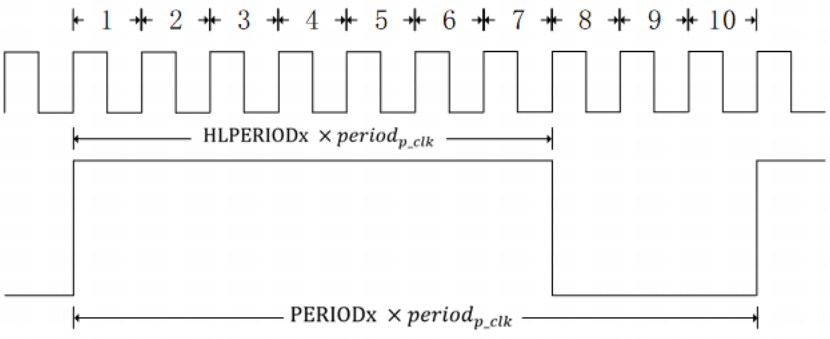

Fig. 10: The effect of registers on PWM waves

Pull High

The steps to stop the PWM wave output and pull the output high are as follows:

1.Configure PERIODx to 0x00000000.

2.Configure HLPERIODx to 0x00000000.

Pull Low

The steps to stop the PWM wave output and pull the output low are as follows:

3.Configure HLPERIODx to 0x00000001.

4.Configure PERIODx to 0x00000000.

Example of starting PWM

Assume that the current p_clk frequency is 100M, that is, its period is 10 ns.

The PWM wave frequency that needs to be output is 10M, that is, its period is 100ns.

The duty cycle required for the PWM wave is 70 percent.

The channel that needs to output the PWM wave is channel 0.

According to the formula given by the function description, you can get a PERIODx value of 10, and a HLPERIODx value of 7.

The steps to start the PWM wave on channel 0 are as follows:

1.Configure PERIODx to 0x0000000a.

2.Configure HLPERIODx to 0x00000007.

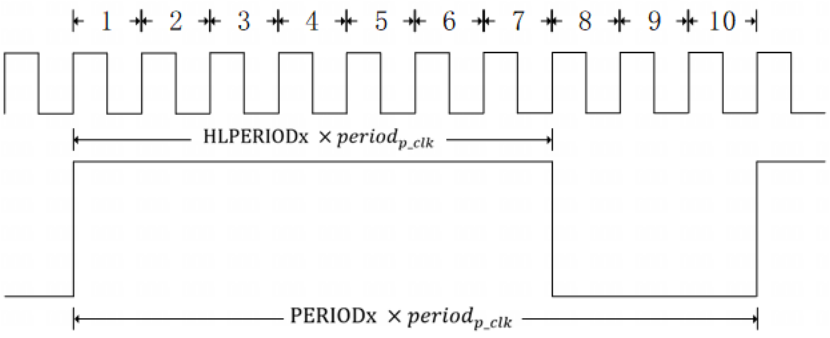

Fig. 11: An example of generating a PWM wave

8.8.2 Pulse detection

Reset State

After reset, the value of FERQxNUM is 0x00000000.

The value of FREQxDATA is 0x00000000

Start pulse detection

The steps to start a pulse detection are as follows:

1.Configure FREQxNUM. The calculation formula for this value is given in Function Description.

2.Wait for a while until the test is completed at least once. This time must be long enough to ensure that at least one time window is run after the registers in the module are configured.

3.Reads the value in FREQxDATA, which is the number of pulses detected in a time window. Pulse detection cannot be turned off. If you need to clear the value in FREQxDATA, you can set FREQxNUM to 0.

An example of starting pulse detection

Assume that the current p_clk frequency is 100M, that is, its period is 10 ns.

The time window width required for pulse detection is 1 ms.

Use channel 0 for pulse detection.

According to the formula given by the function description, the value of FREQ0NUM is 100000.

The steps to start a pulse detection are as follows:

1.Configure FREQ0NUM to 0x000186A0.

2.Wait for a while to ensure that the module runs for more than 1 ms after the register is written. For example, you can wait for 1.5ms under normal conditions.

3.Reads the value in FREQ0DATA.

Figure 12 shows the process of this example.

8.9 Known Issues and Future Work

TBD

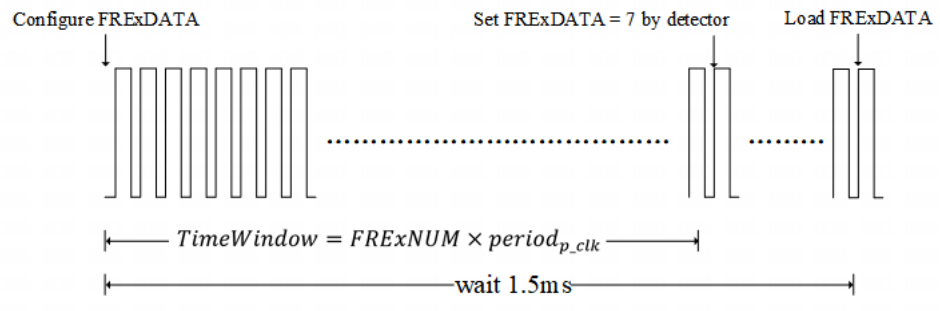

Fig. 12: An example of pulse detection

CHAPTER

NINE

INTERRUPT

1. AP (8-core A53) Interrupt processing: GIC (0x3100_0000)
2. RISC-V interrupt processing: PLIC (0x9000_0000)

Table 1: interrupt information

IP	Comments	Chip Interrupt Signal	RV Interrupt ID
SYS_CTRL	0	reg_gp_intr0[0]	64
SYS_CTRL	0	reg_gp_intr0[1]	65
SYS_CTRL	0	reg_gp_intr0[2]	66
SYS_CTRL	0	reg_gp_intr0[3]	67
SYS_CTRL	0	reg_gp_intr0[4]	68
SYS_CTRL	0	reg_gp_intr0[5]	69
SYS_CTRL	0	reg_gp_intr0[6]	70
SYS_CTRL	0	reg_gp_intr0[7]	71
SYS_CTRL	0	reg_gp_intr0[8]	72
SYS_CTRL	0	reg_gp_intr0[9]	73
SYS_CTRL	0	reg_gp_intr0[10]	74
SYS_CTRL	0	reg_gp_intr0[11]	75
SYS_CTRL	0	reg_gp_intr0[12]	76
SYS_CTRL	0	reg_gp_intr0[13]	77
SYS_CTRL	0	reg_gp_intr0[14]	78
SYS_CTRL	0	reg_gp_intr0[15]	79
SYS_CTRL	0	reg_gp_intr0[16]	80
SYS_CTRL	0	reg_gp_intr0[17]	81
SYS_CTRL	0	reg_gp_intr0[18]	82
SYS_CTRL	0	reg_gp_intr0[19]	83
SYS_CTRL	0	reg_gp_intr0[20]	84
SYS_CTRL	0	reg_gp_intr0[21]	85
SYS_CTRL	0	reg_gp_intr0[22]	86
SYS_CTRL	0	reg_gp_intr0[23]	87
SYS_CTRL	0	reg_gp_intr0[24]	88
SYS_CTRL	0	reg_gp_intr0[25]	89
SYS_CTRL	0	reg_gp_intr0[26]	90
SYS_CTRL	0	reg_gp_intr0[27]	91
SYS_CTRL	0	reg_gp_intr0[28]	92
SYS_CTRL	0	reg_gp_intr0[29]	93
SYS_CTRL	0	reg_gp_intr0[30]	94
SYS_CTRL	0	reg_gp_intr0[31]	95
GPIO	0	gpio0_intr	96

continues on next page

Table 1 – continued from previous page

IP	Comments	Chip Interrupt Signal	RV Interrupt ID
GPIO	0	gpio1_intr	97
GPIO	0	gpio2_intr	98
WDT	0	wdt_intr	99
TIMER	0	timer_intr	100
I2C	0	i2c0_intr	101
I2C	0	i2c1_intr	102
I2C	0	i2c2_intr	103
I2C	0	i2c3_intr	104
RTC	0	rtc_intr	105
SYSDMA	0	sysdma_intr	106
LPC	0	lpc_intr	107
SF	0	spif0_intr	108
SF	0	spif1_intr	109
SPI	0	spi0_intr	110
SPI	0	spi1_intr	111
UART	0	uart0_intr	112
UART	0	uart1_intr	113
UART	0	uart2_intr	114
UART	0	uart3_intr	115
GPM_HSPERI	0	hsperi_sys_gpm_intr	116
GPM_TOP	0	top_gpm_intr	117
	0	ddr0_controller_int	118
	0	ddr1_controller_int	119
	0	ddr2_controller_int	120
	0	ddr3_controller_int	121
	0	pcie0_subsys_int_o	122
	0	pcie1_subsys_int_o	123
	0	ether0_sbd_intr	132
	0	emmc_wakeup_intr	133
	0	emmc_intr	134
	0	sd_wakeup_intr	135
	0	sd_intr	136

CHAPTER

TEN

SYSTEM CONTROL

These registers control system behaviours or hold informations of this chip. This block is more like a gather of misc functions of a SoC.

The base address is listed in table *Memory map*, SYS_CTRL device.

10.1 Registers

10.1.1 CHIP_VERSION: offset 0x000

Chip Version

Table 1: CHIP_VERSION(0x000)

Field Name	MSB	LSB	Type	Reset value	Field Description
CHIP_ID	31	16	RO	16'h2042	Chip ID
CHIP_VERSION	15	0	RO	0	Chip version

10.1.2 CONF_INFORMATION: offset 0x004

Configuration Information

Table 2: CONF_INFORMATION(0x004)

Field Name	MSB	LSB	Type	Reset value	Field Description
REG_DEBUG_SEL	31	24	RW	0	reg_debug_sel register to select which group of debug signal to output 0: debug_dout = 32'h2042 1: debug_dout = pcie0_debug_out 2: debug_dout = pcie1_debug_out
reserved	23	18	RO	0	Reserved
SOCKET_ID	17	16	RO	1	socket_id Socket ID of current chip bit[1] is reserved in SG2042.
MULTI_SOCKET_ENABLE	15	15	RO	0	multi_socket_enable 0: single socket mode 1: multi-socket mode"
REG_DBG_SEL_DIN_AND	14	14	RW	1	reg_dbg_sel_din_and register to mask dbg_sel_din, this register will do logic and with dbg_sel_din before it drives any logic.(Reserved in SG2042)
DBG_SEL_DIN	13	13	RO	0	dbg_sel_din read only register for dbg_sel_din (Reserved in SG2042)
reserved	12	11	RO	0	Reserved
MODE_SEL	10	8	RO	4	mode_sel read only register for mode_sel IO
BOOT_SEL	7	0	RO	0	boot_sel read only register for boot_sel IO

10.1.3 TOP_MISC_CONTROL_REGISTER: offset 0x008

Top Misc Control Registers

Table 3: TOP_MISC_CONTROL_REGISTER(0x008)

Field Name	MSB	LSB	Type	Reset value	Field Description
REG_HAD _ENTRY _CTRL	31	16	RW	0	reg_had_entry_ctrl 0: CPU clusters can be accessed by debug server 1: CPU clusters can NOT be accessed by debug server
reserved	15	7	RO	0	Reserved
REG_HSDMA _WR_ID _RE- ORDER_EN	6	6	RW	0	reg_hsdma_wr_id_reorder_en 1: Reorder HSDMA write transactions' ID to improve HS-DMA DDR access performance
REG_HSDMA _RD_ID _RE- ORDER_EN	5	5	RW	0	reg_hsdma_rd_id_reorder_en 1: Reorder HSDMA read transactions' ID to improve HS-DMA DDR access performance
REG_A53_CMN _TRANS_CTRL	4	4	RW	0	reg_a53_cmn_trans_ctrl 1: modify a53 transactions on CMN to non-cacheable
REG_FAU_INTF _SEL	3	3	RW	0	reg_fau_intf_sel 0: AXI; 1: GIF
REG_SW_ROOT _RESET_EN	2	2	RW	0	reg_sw_root_reset_en register to enable software reset whole chip by watchdog, ap debug reset.
AH- BROM1_BOOT _FINISH	1	1	RW	0	ahbrom1_boot_finish after rom1 boot finish set this bit to 1 to put boot ROM in sleep mode.
AH- BROM0_BOOT _FINISH	0	0	RW	0	ahbrom0_boot_finish after rom0 boot finish set this bit to 1 to put boot ROM in sleep mode.

10.1.4 VMON_OR_TMON_MUX_SELECT: offset 0x00C

Temperture And Voltage Mux Select

Table 4: VMON_OR_TMON_MUX_SELECT(0x00C)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	13	RO	0	Reserved
REG_VOLTAGE_MUX_SEL	12	8	RW	0	Voltage Monitor Mux Select
reserved	7	5	RO	0	Reserved
REG_TEMPERATURE_MUX_SEL	4	0	RW	0	Temperature Monitor Mux Select

10.1.5 PROCESS_MONITOR_CONTROL_REGISTER: offset 0x010

Process Monitor Control Register

Table 5: PROCESS_MONITOR_CONTROL_REGISTER(0x010)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	4	RO	0	Reserved
REG_PM_EN	3	3	RW	0	reg_pm_en Enable signal for process monitor clock Step1: Set reg_pm_en Step2: Configure reg_pm_select Step3: Set reg_pm_start Step4: Read toreg_pm_count
REG_PM_SELECT	2	1	RW	0	reg_pm_select Selection of process monitor 2'b00: ulvt16 2'b01: ulvt20 2'b10: lvt16 2'b11: lvt20
REG_PM_START	0	0	RW	0	reg_pm_start Start trigger of process monitor

10.1.6 WATCHDOG_RESET_STAT: offset 0x01C

WATCHDOG RESET Happened

Table 6: WATCHDOG_RESET_STAT(0x01C)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
WDT_RST_HAPPENED	0	0	W1C	0	Watch-Dog Reset Happened 1: Watch-Dog Reset happened This register is used to indicate whether Watch-Dog Reset is happened. SW writes 1 to clear this bit.

10.1.7 CLOCK_GATING_ENABLE_REGISTER_0: offset 0x020

Auto Clock Gating Enable Control

Table 7: CLOCK_GATING_ENABLE_REGISTER_0(0x020)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	2	RW	0	Reserved
HSPERI_CLK_GATING_EN	1	1	RW	0	[HSPERI] High-Speed Peripheral Subsystem Auto Clock Gating Enable
TOP_CLK_GATING_EN	0	0	RW	0	[TOP] Top Fabric Auto Clock Gating Enable

10.1.8 CLOCK_GATING_ENABLE_REGISTER_1: offset 0x024

Auto Clock Gating Enable Control

Table 8: CLOCK_GATING_ENABLE_REGISTER_1(0x024)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	16	RO	0	Reserved
HSPERI_CLK_GATING_IDLE_THRESHOLD	15	8	RW	8'h20	[HSPERI] Fabric Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[1] of Auto Clock Gating Enable Control Register 0(0x20) is cleared. The function is only valid when bit[1] of Auto Clock Gating Enable Control Register 0(0x20) is set.
TOP_CLK_GATING_IDLE_THRESHOLD	7	0	RW	8'h10	[Top] Fabric Auto Clock Gating Idle Threshold. After N cycles (N is defined by this register) of Fabric Idle, Fabric Low Power Controller will start Auto Clock Gating. This field can only be modified when bit[0] of Auto Clock Gating Enable Control Register 0(0x20) is cleared. The function is only valid when bit[0] of Auto Clock Gating Enable Control Register 0(0x20) is set.

10.1.9 DEBUG_I2C_ID: offset 0x040

Debug I2C ID

Table 9: DEBUG_I2C_ID(0x040)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	8	RO	0	Reserved
DBG_I2C_ID	7	0	RW	8'hc0	System Debug I2C ID Note the real Debug I2C Slave Address = {DBG_I2C_ID[7:2], Chip_socket_id[1:0]}

10.1.10 DEBUG_I2C_QOS_CONTROL: offset 0x044

DEBUG_I2C_QOS_CONTROL

Table 10: DEBUG_I2C_QOS_CONTROL(0x044)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	8	RO	0	Reserved
REG_QOS_DBG_I2C_ARQOS	7	4	RW	0	DBG_I2C_ARQOS
REG_QOS_DBG_I2C_AWQOS	3	0	RW	0	DBG_I2C_AWQOS

10.1.11 ETH0_QOS_CONTROL: offset 0x048

ETH0_QOS_CONTROL

Table 11: ETH0_QOS_CONTROL(0x048)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	8	RO	0	Reserved
REG_QOS_ETH0_ARQOS	7	4	RW	0	ETH0_ARQOS
REG_QOS_ETH0_AWQOS	3	0	RW	0	ETH1_AWQOS

10.1.12 HSPERI_MEM_REMAP_MODE: offset 0x04C

HSPERI_MEM_REMAP_MODE

Table 12: HSPERI_MEM_REMAP_MODE(0x04C)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
REG_HSPERI_MEM_REMAP_MODE	0	0	RW	0	1'b0:auto mode(address is extended with Chip socket id) 1'b1:fixed mode(address is extended with hsperi_mem_remap_reg) new_addr[43:0] = hsperi_mem_remap_mode ? { 4'h0,hsperi_mem_remap_reg[0], ori_addr[38:0] }:{ 4'h0, socket_id[0], ori_addr[38:0] }

10.1.13 HSPERI_MEM_REMAP_REG: offset 0x050

HSPERI_MEM_REMAP_REG

Table 13: HSPERI_MEM_REMAP_REG(0x050)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	10	RO	0	Reserved
REG_HSPERI_MEM_ARADDR_REMAP	9	8	RW	0	REG_HSPERI_MEM_ARADDR_REMAP bit[9] is reserved in SG2042
reserved	7	2	RO	0	Reserved
REG_HSPERI_MEM_AWADDR_REMAP	1	0	RW	0	REG_HSPERI_MEM_AWADDR_REMAP bit[1] is reserved in SG2042

10.1.14 DDR_SIZE_REG: offset 0x054

Table 14: DDR_SIZE_REG(0x054)

Field Name	MSB	LSB	Type	Reset value	Field Description
DDR3_SIZE_RE(	31	24	RW	8'h4	DDR3 Size
DDR2_SIZE_RE(	23	16	RW	8'h4	DDR2 Size
DDR1_SIZE_RE(	15	8	RW	8'h4	DDR1 Size
DDR0_SIZE_REG	7	0	RW	8'h4	DDR0 Size: 8'h0: ddr size is 1TB, bypass 40bit address 8'h1: ddr size is 512GB, tie 1-bit MSB of CMN-> DDR address to 0 8'h2: ddr size is 256GB, tie 2-bit MSB of CMN-> DDR address to 0 8'h3: ddr size is 128GB, tie 3-bit MSB of CMN-> DDR address to 0 8'h4: ddr size is 64GB, tie 4-bit MSB of CMN-> DDR address to 0 8'h5: ddr size is 32GB, tie 5-bit MSB of CMN-> DDR address to 0 8'h6: ddr size is 16GB, tie 6-bit MSB of CMN-> DDR address to 0 8'h7: ddr size is 8GB, tie 7-bit MSB of CMN-> DDR address to 0 8'h8: ddr size is 4GB, tie 8-bit MSB of CMN-> DDR address to 0 8'h9: ddr size is 2GB, tie 9-bit MSB of CMN-> DDR address to 0 8'hA: ddr size is 1GB, tie 10-bit MSB of CMN-> DDR address to 0 other: NA

10.1.15 DDR_CTRL_REG: offset 0x058

Table 15: DDR_CTRL_REG(0x058)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	28	RO	0	Reserved
DDR_CORE _RST_CNT	27	24	RW	4'h6	DDR Core Reset Counter Threshold
DDR_MEM _RST_CNT	23	20	RW	4'h9	DDR MEM Reset Counter Threshold
DDR_REG _RST_CNT	19	16	RW	4'h8	DDR REG Reset Counter Threshold
reserved	15	1	RO	0	Reserved
DDR_AW_W _ALIGN _ENABLE	0	0	RW	1'h0	DDR AW W ALIGN Enable 0: Disable DDR AW W Alignment 1: Enable DDR AW W Alignment (The write request will be sent to DDR only when the write data is also shown on DDR port.)

10.1.16 AP_WIFI_STAT: offset 0x080

AP WFI Status Register

Table 16: AP_WIFI_STAT(0x080)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	18	RO	0	Reserved
AP_CL1_ACINACTM	17	17	RW	0	AP system Cluster 1 ACINACTM: 0: cluster 1 may be snooped by external system 1: cluster 1 will not be snooped by external system
AP_CL0_ACINACTM	16	16	RW	0	AP system Cluster 0 ACINACTM: 0: cluster 0 may be snooped by external system 1: cluster 0 will not be snooped by external system
reserved	15	10	RO	0	Reserved
CL1_STANDBYWFIL2	9	9	RO	0	AP system Cluster 1 WFI State
CL1_STANDBYWFI	8	5	RO	0	AP system Core4-7 WFI State
CL0_STANDBYWFIL2	4	4	RO	0	AP system Cluster 0 WFI State
CL0_STANDBYWFI	3	0	RO	0	AP system Core0-3 WFI State

10.1.17 AP_WARM_RESET: offset 0x084

AP Warm Reset Control and Status

Table 17: AP_WARM_RESET(0x084)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	3	RO	0	Reserved
JTAG _WARM_RST _DISABLE	2	2	RW	0	JTAG Warm Reset Disable bit[9] is reserved in SG2042
AP_SYS _WARM_RST _ACT	1	1	RO	0	AP System Warm Reset Active signal. This bit reflect the current status of AP System Warm Reset Active signal (ap_sys_warm_rst_act).
CLR_AP_SYS _WARM_RST _ACT	0	0	RW	0	Clear AP System Warm Reset Active signal. Writing 1 into this bit will clear the AP System Warm Reset Active signal (ap_sys_warm_rst_act)

10.1.18 ARM_BOOT_ADDR_L: offset 0x088

ARM boot start address

Table 18: ARM_BOOT_ADDR_L(0x088)

Field Name	MSB	LSB	Type	Reset value	Field Description
AP_RVBARADDR_L	31	0	RW	32'h0218_0000	ap_rvbaraddr_full[31:0] ARM boot start address. ap_rvbaraddr_full: default value is decided by boot_sel[1]: 1'b0: 40'h00_0014_0000 (ROM1) 1'b1: 40'h00_0218_0000 (Serial Flash1) ap_rvbaraddr = ap_rvbaraddr_full[39:2]

10.1.19 ARM_BOOT_ADDR_H: offset 0x08C

ARM boot start address

Table 19: ARM_BOOT_ADDR_H(0x08C)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	8	RO	0	Reserved
AP_RVBARADDR_H	7	0	RW	0	ap_rvbaraddr_full[39:32] ARM boot start address

10.1.20 AP_QOS_CONTROL: offset 0x094

Table 20: AP_QOS_CONTROL(0x094)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	16	RO	0	Reserved
REG_QOS_AP_MEM0_ARQOS	15	12	RW	0	AP_MEM0_ARQOS
REG_QOS_AP_MEM0_AWQOS	11	8	RW	0	AP_MEM0_AWQOS
REG_QOS_AP_REG_ARQOS	7	4	RW	0	AP_REG_ARQOS
REG_QOS_AP_REG_AWQOS	3	0	RO	0	AP_REG_AWQOS

10.1.21 AP_MEM_ADDRESS_REMAP_REGISTER: offset 0x098

Table 21: AP_MEM_ADDRESS_REMAP_REGISTER(0x098)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	13	RO	0	Reserved
REG_AP_MEM _ARADDR_REMAP	12	8	R0	0	REG_AP_MEM_ARADDR_REMAP This register is reserved in SG2042.
reserved	7	5	RO	0	Reserved
REG_AP_MEM _AWADDR_REMAP	4	0	R0	0	REG_AP_MEM_AWADDR_REMAP This register is reserved in SG2042.

10.1.22 PLL_STAT: offset 0x0C0

Pll Status

Table 22: PLL_STAT(0x0C0)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	14	RO	0	Reserved
DPLL1_LOCK	13	13	RO	0	DPLL1 LOCK
DPLL0_LOCK	12	12	RO	0	DPLL0_LOCK
FPLL_LOCK	11	11	RO	0	FPLL_LOCK
reserved	10	10	RO	0	Reserved
reserved	9	9	RO	0	Reserved
MPLL_LOCK	8	8	RO	0	MPLL LOCK
reserved	7	6	RO	0	Reserved
UPDATING_DPLL1_VAL	5	5	RO	0	updating_dpll1_val
UPDATING_DPLL0_VAL	4	4	RO	0	updating_dpll0_val
UPDATING_FPLL_VAL	3	3	RO	0	updating_fpll_val
reserved	2	2	RO	0	Reserved
reserved	1	1	RO	0	Reserved
MPLL_LOCK	0	0	RO	0	updating_mpll_val

10.1.23 PLL_CLKEN_CONTROL: offset 0x0C4

PLL Clock Enable Control

Table 23: PLL_CLKEN_CONTROL(0x0C4)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	14	RO	0	Reserved
DPLL1_CLKEN_MUX_SEL	13	13	RW	0	DPLL1 Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
DPLL0_CLKEN_MUX_SEL	12	12	RW	0	DPLL0 Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
FPLL_CLKEN_MUX_SEL	11	11	RW	0	FPLL Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
reserved	10	10	RO	0	Reserved
reserved	9	9	RO	0	Reserved
MPLL_CLKEN_MUX_SEL	8	8	RW	0	MPLL Clock Enable Mux Control 0: Select Unsynced PLL Clock Enable 1: Select Synced version of PLL Clock Enable
reserved	7	6	RO	0	Reserved
DPLL1_CLK_EN	5	5	RW	1	DPLL1 Clock Enable
DPLL0_CLK_EN	4	4	RW	1	DPLL0 Clock Enable
FPLL_CLK_EN	3	3	RW	1	FPLL Clock Enable
reserved	2	2	RO	1	Reserved
reserved	1	1	RO	1	Reserved
MPLL_CLK_EN	0	0	RW	1	MPLL Clock Enable

10.1.24 MPLL_CONTROL: offset 0x0E8

Main PLL Control

Table 24: MPLL_CONTROL(0x0E8)

Field Name	MSB	LSB	Type	Reset value	Field Description
MPLL_FAST_CONFIG_EN	31	31	RW	0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.
reserved	30	28	RO	0	Reserved
MPLL_FBDIV	27	16	RW	12'h40	FBDIV Normal Mode: 'h40 Fast Mode: 'h50 Safe Mode: 'h28
reserved	15	15	RO	0	Reserved
MPLL_POST-DIV2	14	12	RW	1	POSTDIV2
reserved	11	11	RO	0	Reserved
MPLL_POST-DIV1	10	8	RW	1	POSTDIV1
reserved	7	6	RO	0	Reserved
MPLL_REF-DIV	5	0	RW	1	REFDIV

10.1.25 FPLL_CONTROL: offset 0x0F4

Fixed PLL Control

Table 25: FPLL_CONTROL(0x0F4)

Field Name	MSB	LSB	Type	Reset value	Field Description
FPLL_FAST_CONFIG_EN	31	31	WO	0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.
reserved	30	28	RO	0	Reserved
FPLL_FBDIV	27	16	RW	12'h40	FBDIV Normal Mode: 'h28 Fast Mode: 'h28 Safe Mode: 'h28
reserved	15	15	RO	0	Reserved
FPLL_POSTDIV2	14	12	RW	1	POSTDIV2
reserved	11	11	RO	0	Reserved
FPLL_POSTDIV1	10	8	RW	1	POSTDIV1
reserved	7	6	RO	0	Reserved
FPLL_REFDIV	5	0	RW	1	REFDIV

10.1.26 DPLL0_CONTROL: offset 0x0F8

DDR PLL 0 Control

Table 26: DPLL0_CONTROL(0x0F8)

Field Name	MSB	LSB	Type	Reset value	Field Description
DPLL0_FAST_CONFIG_EN	31	31	WO	0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.
reserved	30	28	RO	0	Reserved
DPLL0_FBDIV	27	16	RW	12'h30	FBDIV Normal Mode: 'h35 Fast Mode: 'h40 Safe Mode: 'h20
reserved	15	15	RO	0	Reserved
DPLL0_POSTDIV2	14	12	RW	1	POSTDIV2
reserved	11	11	RO	0	Reserved
DPLL0_POSTDIV1	10	8	RW	1	POSTDIV1
reserved	7	6	RO	0	Reserved
DPLL0_REFDIV	5	0	RW	1	REFDIV

10.1.27 DPLL1_CONTROL: offset 0x0FC

DDR PLL 1 Control

Table 27: DPLL1_CONTROL(0x0FC)

Field Name	MSB	LSB	Type	Reset value	Field Description
DPLL1_FAST_CONFIG_EN	31	31	WO	0	Fast Config Mode Enable 1: Enable Fast Config Mode. In this mode, only FBDIV can be modified, and there will be no PLL Power-Down sequence in PLL frequency update. 0: Disable Fast Config Mode.
reserved	30	28	RO	0	Reserved
DPLL1_FBDIV	27	16	RW	12'h30	FBDIV Normal Mode: 'h35 Fast Mode: 'h40 Safe Mode: 'h20
reserved	15	15	RO	0	Reserved
DPLL1_POSTDIV2	14	12	RW	1	POSTDIV2
reserved	11	11	RO	0	Reserved
DPLL1_POSTDIV1	10	8	RW	1	POSTDIV1
reserved	7	6	RO	0	Reserved
DPLL1_REFDIV	5	0	RW	1	REFDIV

10.1.28 DEVICE_LOCK_REGISTER

Device Lock

The read operation will return the value then assert this bit.

The write operation will de-assert the bit.

Table 28: DEVICE_LOCK_REGISTER0: offset 0x140

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG0	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 29: DEVICE_LOCK_REGISTER1: offset 0x144

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG1	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 30: DEVICE_LOCK_REGISTER2: offset 0x148

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG2	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 31: DEVICE_LOCK_REGISTER3: offset 0x14C

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG3	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 32: DEVICE_LOCK_REGISTER4: offset 0x150

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG4	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 33: DEVICE_LOCK_REGISTER5: offset 0x154

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG5	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 34: DEVICE_LOCK_REGISTER6: offset 0x158

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG6	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 35: DEVICE_LOCK_REGISTER7: offset 0x15C

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG7	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 36: DEVICE_LOCK_REGISTER8: offset 0x160

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG8	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 37: DEVICE_LOCK_REGISTER9: offset 0x164

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG9	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 38: DEVICE_LOCK_REGISTER10: offset 0x168

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG10	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 39: DEVICE_LOCK_REGISTER11: offset 0x16C

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG11	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 40: DEVICE_LOCK_REGISTER12: offset 0x170

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG12	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 41: DEVICE_LOCK_REGISTER13: offset 0x174

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG13	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 42: DEVICE_LOCK_REGISTER14: offset 0x178

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG14	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

Table 43: DEVICE_LOCK_REGISTER11: offset 0x17C

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
DEV_LOCK_REG15	0	0	RW	0	Lock Control and Status (1).The read operation will return the value then assert this bit. (2). Write operation will de-assert the bit.

10.1.29 GENERAL_PURPOSE_REGISTER

General purpose register for sw usage

The Field Description for all of the registers in table 44:General purpose register

Table 44: GENERAL_PURPOSE_REGISTER

Offset	Reg Name	Field Name	MSB	LSB	Type	Reset value
0x1C0	GENERAL_PURPOSE_REGISTER0	GP_REG0	31	0	RW	0
0x1C4	GENERAL_PURPOSE_REGISTER1	GP_REG1	31	0	RW	0
0x1C8	GENERAL_PURPOSE_REGISTER2	GP_REG2	31	0	RW	0
0x1CC	GENERAL_PURPOSE_REGISTER3	GP_REG3	31	0	RW	0
0x1D0	GENERAL_PURPOSE_REGISTER4	GP_REG4	31	0	RW	0
0x1D4	GENERAL_PURPOSE_REGISTER5	GP_REG5	31	0	RW	0
0x1D8	GENERAL_PURPOSE_REGISTER6	GP_REG6	31	0	RW	0
0x1DC	GENERAL_PURPOSE_REGISTER7	GP_REG7	31	0	RW	0
0x1E0	GENERAL_PURPOSE_REGISTER8	GP_REG8	31	0	RW	0
0x1E4	GENERAL_PURPOSE_REGISTER9	GP_REG9	31	0	RW	0
0x1E8	GENERAL_PURPOSE_REGISTER10	GP_REG10	31	0	RW	0
0x1EC	GENERAL_PURPOSE_REGISTER11	GP_REG11	31	0	RW	0
0x1F0	GENERAL_PURPOSE_REGISTER12	GP_REG12	31	0	RW	0
0x1F4	GENERAL_PURPOSE_REGISTER13	GP_REG13	31	0	RW	0
0x1F8	GENERAL_PURPOSE_REGISTER14	GP_REG14	31	0	RW	0
0x1FC	GENERAL_PURPOSE_REGISTER15	GP_REG15	31	0	RW	0
0x200	GENERAL_PURPOSE_REGISTER16	GP_REG16	31	0	RW	0
0x204	GENERAL_PURPOSE_REGISTER17	GP_REG17	31	0	RW	0
0x208	GENERAL_PURPOSE_REGISTER18	GP_REG18	31	0	RW	0
0x20C	GENERAL_PURPOSE_REGISTER19	GP_REG19	31	0	RW	0
0x210	GENERAL_PURPOSE_REGISTER20	GP_REG20	31	0	RW	0
0x214	GENERAL_PURPOSE_REGISTER21	GP_REG21	31	0	RW	0
0x218	GENERAL_PURPOSE_REGISTER22	GP_REG22	31	0	RW	0
0x21C	GENERAL_PURPOSE_REGISTER23	GP_REG23	31	0	RW	0
0x220	GENERAL_PURPOSE_REGISTER24	GP_REG24	31	0	RW	0
0x224	GENERAL_PURPOSE_REGISTER25	GP_REG25	31	0	RW	0
0x228	GENERAL_PURPOSE_REGISTER26	GP_REG26	31	0	RW	0
0x22C	GENERAL_PURPOSE_REGISTER27	GP_REG27	31	0	RW	0
0x230	GENERAL_PURPOSE_REGISTER28	GP_REG28	31	0	RW	0
0x234	GENERAL_PURPOSE_REGISTER29	GP_REG29	31	0	RW	0
0x238	GENERAL_PURPOSE_REGISTER30	GP_REG30	31	0	RW	0
0x23C	GENERAL_PURPOSE_REGISTER31	GP_REG31	31	0	RW	0

10.1.30 PM_COUNT_REGISTER

Process Monitor Counter Register

Table 45: PM_COUNT_REGISTER

Offset	Reg Name	Field Name	MSB	LSB	Type	Reset value	Field Description
0x2A0	PM_COUNT_REGISTER0	reserved	31	16	RO	0	Reserved
		PM_COUNT_0	15	0	RO	0	toreg_pm_count0 count value of process monitor
0x2A4	PM_COUNT_REGISTER1	reserved	31	16	RO	0	Reserved
		PM_COUNT_1	15	0	RO	0	toreg_pm_count1 count value of process monitor
0x2A8	PM_COUNT_REGISTER2	reserved	31	16	RO	0	Reserved
		PM_COUNT_2	15	0	RO	0	toreg_pm_count2 count value of process monitor

10.1.31 GP_INTR_REGISTER

General Purpose Interrupt Register

Table 46: GP_INTR_REGISTER

Offset	Reg Name	Field Name	MSB	LSB	Type	Reset value	Field Description
0x2E0	GP_INTR_REGISTER_0	REG_GP_INTR0	31	0	RO	0	General Purpose Interrupt Register (reg_gp_intr)
0x2E4	GP_INTR_REGISTER_1	REG_GP_INTR1	31	0	RO	0	General Purpose Interrupt Register (reg_gp_intr)

10.1.32 GP_INTR0_SET: offset 0x300

REG_GP_INTR0 Set Register

Table 47: GP_INTR0_SET(0x300)

Field Name	MSB	LSB	Type	Reset value	Field Description
REG_GP_INTR0_SET	31	0	WO	0	Write 1 into this register will also set the corresponding bit in General Purpose Interrupt Register 0 (REG_GP_INTR0). When SW writes value into this register, the behavior is shown as: REG_GP_INTR0 <= REG_GP_INTR0 | this_reg

10.1.33 GP_INTR0_CLR: offset 0x304

REG_GP_INTR0 CLR Register

Table 48: GP_INTR0_CLR(0x304)

Field Name	MSB	LSB	Type	Reset value	Field Description
REG_GP_INTR0_CLR	31	0	WO	0	Write 1 into this register will also clear the corresponding bit in General Purpose Interrupt Register 0 (REG_GP_INTR0). When SW writes value into this register, the behavior is shown as: REG_GP_INTR0 <= REG_GP_INTR0 & ~this_reg

10.1.34 GP_INTR1_SET: offset 0x308

REG_GP_INTR1 Set Register

Table 49: GP_INTR1_SET(0x308)

Field Name	MSB	LSB	Type	Reset value	Field Description
REG_GP_INTR1_SET	31	0	WO	0	Write 1 into this register will also set the corresponding bit in General Purpose Interrupt Register 1 (REG_GP_INTR1). When SW writes value into this register, the behavior is shown as: REG_GP_INTR1 <= REG_GP_INTR1 | this_reg

10.1.35 GP_INTR1_CLR: offset 0x30C

REG_GP_INTR1 CLR Register

Table 50: GP_INTR1_CLR(0x30C)

Field Name	MSB	LSB	Type	Reset value	Field Description
REG_GP_INTR1_CLR	31	0	WO	0	Write 1 into this register will also clear the corresponding bit in General Purpose Interrupt Register 1 (REG_GP_INTR1). When SW writes value into this register, the behavior is shown as: REG_GP_INTR1 <= REG_GP_INTR1 & ~this_reg

10.1.36 RP_CPU_VENDOR_ID_L: offset 0x340

Table 51: RP_CPU_VENDOR_ID_L(0x340)

Field Name	MSB	LSB	Type	Reset value	Field Description
TOP_RP_CPU_VENDORID	31	0	RW	0	RP_CPU_VENDOR_ID_L SW program the correct value after boot.

10.1.37 RP_CPU_VENDOR_ID_H: offset 0x344

Table 52: RP_CPU_VENDOR_ID_H(0x344)

Field Name	MSB	LSB	Type	Reset value	Field Description
TOP_RP_CPU_VENDORID	31	0	RW	0	RP_CPU_VENDOR_ID_H SW program the correct value after boot.

10.1.38 RP_CPU_APB_BASE_L: offset 0x348

Table 53: RP_CPU_APB_BASE_L(0x348)

Field Name	MSB	LSB	Type	Reset value	Field Description
TOP_RP_CPU_APB_BASE	31	0	RW	32'hA8000000	RP_CPU_APB_BASE_L Access towards APB Base will not be routed to CMN, this address space is mainly for CLINT Timer and RPU register configuration.

10.1.39 RP_CPU_APB_BASE_H: offset 0x34C

Table 54: RP_CPU_APB_BASE_H(0x34C)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	16	RO	0	Reserved
TOP_RP_CPU_APB_BASE	15	0	RW	0	RP_CPU_APB_BASE_H

10.1.40 RP_CPU_RVBA_L: offset 0x350

Table 55: RP_CPU_RVBA_L(0x350)

Field Name	MSB	LSB	Type	Reset value	Field Description
TOP_RP_CPU_RVBA	31	0	RW	32'h0018_0000	top_rp_cpu_rvba[31:0] RISC-V boot start address. top_rp_cpu_rvba: default value is decided by boot_sel[1]: 1'b0: 48'h0000_0010_0000 (ROM0) 1'b1: 48'h0000_0018_0000 (Serial Flash0)

10.1.41 RP_CPU_RVBA_H: offset 0x354

Table 56: RP_CPU_RVBA_H(0x354)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	16	RO	0	Reserved
TOP_RP_CPU_RVBA	15	0	RW	0	top_rp_cpu_rvba[47:32] RISC-V boot start address.

10.1.42 RP_CFGM_PERIPHBASE_L: offset 0x358

Table 57: RP_CFGM_PERIPHBASE_L(0x358)

Field Name	MSB	LSB	Type	Reset value	Field Description
TOP_RP_CFGM_PERIPHBASE	31	0	RW	32'h70000000	RP_CFGM_PERIPHBASE_L Start address for CMN register configuration.

10.1.43 RP_CFGM_PERIPHBASE_H: offset 0x35C

Table 58: RP_CFGM_PERIPHBASE_H(0x35C)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	12	RO	0	Reserved
TOP_RP_CFGM_PERIPHBASE	11	0	RW	0	RP_CFGM_PERIPHBASE_H

10.1.44 RP_CPU_SEC_ACC: offset 0x360

Table 59: RP_CPU_SEC_ACC(0x360)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	1	RO	0	Reserved
TOP_RP_CPU_SEC_ACC	0	0	RW	0	TOP_RP_CPU_SEC_ACC Control the Security Bit of prot signal.

10.1.45 RP_CPU_MEMMAP_EXPA: offset 0x364

Table 60: RP_CPU_MEMMAP_EXPA(0x364)

Field Name	MSB	LSB	Type	Reset value	Field Description
reserved	31	9	RO	0	Reserved
TOP_RP_CPU_MEMMAP_EXPA	8	0	RW	0	TOP_RP_CPU_MEMMAP_EXPA

10.1.46 RP_RXU_CLK_ENABLE: offset 0x368

Table 61: RP_RXU_CLK_ENABLE(0x368)

Field Name	MSB	LSB	Type	Reset value	Field Description
TOP_RP_RXU_CLK_ENABLE	31	0	RW	32'hFFFF_FFFF	Clock Enable for RXU bit[31]: clock enable for rxu31 ... bit[0]: clock enable for rxu0

10.1.47 MP_REG

Table 62: MP_REG

Offset	Reg Name	Field Name	MSB	LSB	Type	Reset value	Field Description
0x380	MP0_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP0	9	0	RW	0	MP0_CLUSTER_ID
0x384	MP0_CONTROI_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP0	0	0	RW	1	Clock Enable for MP0 C910
0x388	MP1_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP1	9	0	RW	1	MP1_CLUSTER_ID
0x38C	MP1_CONTROI_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP1	0	0	RW	1	Clock Enable for MP1 C910
0x390	MP2_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP2	9	0	RW	2	MP2_CLUSTER_ID
0x394	MP2_CONTROI_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP2	0	0	RW	1	Clock Enable for MP2 C910
0x398	MP3_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP3	9	0	RW	3	MP3_CLUSTER_ID
0x39C	MP3_CONTROI_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP3	0	0	RW	1	Clock Enable for MP3 C910
0x3A0	MP4_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP4	9	0	RW	4	MP4_CLUSTER_ID
0x3A4	MP4_CONTROI_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP4	0	0	RW	1	Clock Enable for MP4 C910
0x3A8	MP5_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP5	9	0	RW	5	MP5_CLUSTER_ID
0x3AC	MP5_CONTROI_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP5	0	0	RW	1	Clock Enable for MP5 C910
0x3B0	MP6_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP6	9	0	RW	6	MP6_CLUSTER_ID
0x3B4	MP6_CONTROI_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP6	0	0	RW	1	Clock Enable for MP6 C910

Off-set	Reg Name	Field Name	MSB	LSB	Type	Re-set value	Field Description
0x3B8	MP7_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP7	9	0	RW	7	MP7_CLUSTER_ID
0x3BC	MP7_CONTROL_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP7	0	0	RW	1	Clock Enable for MP7 C910
0x3C0	MP8_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP8	9	0	RW	8	MP8_CLUSTER_ID
0x3C4	MP8_CONTROL_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP8	0	0	RW	1	Clock Enable for MP8 C910
0x3C8	MP9_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP9	9	0	RW	9	MP9_CLUSTER_ID
0x3CC	MP9_CONTROL_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP9	0	0	RW	1	Clock Enable for MP9 C910
0x3D0	MP10_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP10	9	0	RW	10'ha	MP10_CLUSTER_ID
0x3D4	MP10_CONTRO_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP10	0	0	RW	1	Clock Enable for MP10 C910
0x3D8	MP11_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP11	9	0	RW	10'hb	MP11_CLUSTER_ID
0x3DC	MP11_CONTRO_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP11	0	0	RW	1	Clock Enable for MP11 C910
0x3E0	MP12_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP12	9	0	RW	10'hc	MP12_CLUSTER_ID
0x3E4	MP12_CONTRO_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP12	0	0	RW	1	Clock Enable for MP12 C910
0x3E8	MP13_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP13	9	0	RW	10'hd	MP13_CLUSTER_ID
0x3EC	MP13_CONTRO_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP13	0	0	RW	1	Clock Enable for MP13 C910

Offset	Reg Name	Field Name	MSB	LSB	Type	Reset value	Field Description
0x3F0	MP14_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP14	9	0	RW	10'he	MP14_CLUSTER_ID
0x3F4	MP14_CONTRO_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP14	0	0	RW	1	Clock Enable for MP14 C910
0x3F8	MP15_STATUS_REG	reserved	31	10	RO	0	Reserved
		TOP_RP_CLUSTER_ID_MP15	9	0	RW	10'hf	MP15_CLUSTER_ID
0x3FC	MP15_CONTRO_REG	reserved	31	1	RO	0	Reserved
		TOP_RP_CPU_CLK_EN_MP15	0	0	RW	1	Clock Enable for MP15 C910

CHAPTER

ELEVEN

I2C

11.1 Registers

This section describes the programmable registers of the DW_apb_i2c.

Note:There are references to both hardware parameters and software registers throughout this chapter. Parameters and many of the register bits are prefixed with an IC_*. However, the software register bits are distinguished in this chapter by italics. For instance, IC_MAX_SPEED_MODE is a hardware parameter and configured once using Synopsys coreConsultant, whereas the IC_SLAVE_DISABLE bit in the IC_CON register controls whether I2C has its slave disabled.

11.1.1 Register Memory Map

Table 1 provides the details of the DW_apb_i2c memory map.

Table 1: Memory Map of DW_apb_i2c

Name	Address Offset	width	R/W	Description
IC_CON	0x00	20 bits	R/W or R-Only on bit 4 and bit 9 to 19.	I2C control R/W: I2C_DYNAMIC_TAR_UPDATE=1, bit 4 is read only. IC_RX_FULL_HLD_BUS_EN =0, bit 9 is read only. IC_STOP_DET_IF_MASTER_ACTIVE =0, bit 10 is read only. IC_BUS_CLEAR_FEATURE=0, bit 11 is read only IC_OPTIONAL_SAR=0, bit 16 is read only IC_SMBUS=0, bit 17 is read only IC_SMBUS_ARP=0, bits 18 and 19 are read only. Reset Value: 19: IC_PERSISTANT_SLV_ADDR_DEFAULT 17 to 18 : 0 16: IC_OPTIONAL_SAR_DEFAULT 15 to 7: 0 6: IC_SLAVE_DISABLE 5: IC_RESTART_EN 4: IC_10BITADDR_MASTER 3: IC_10BITADDR_SLAVE 2:1:IC_MAX_SPEED_MODE 0: IC_MASTER_MODE
IC_TAR	0x04	12,13, 14 or 16 bits	R/W	I2C Target Address Width: If I2C_DYNAMIC_TAR_UPDATE=1, 13 bits If IC_DEVICE_ID=1, 14 bits If IC_SMBUS=1, 17 bits otherwise 12 bits Reset Value: Reset values for the four bit fields correspond to the following 13: 0x0 12: IC_10BITADDR_MASTER configuration parameter 11: 0x0 10: 0x0 9:0: IC_DEFAULT_TAR_SLAVE_ADDR
IC_SAR	0x08	10 bits	R/W	Slave Target Address Reset Value: IC_DEFAULT_SLAVE_ADDR
IC_HS_MADDR	0x0C	3 bits	R/W	I2C HS Master Mode Code Address Reset Value: IC_HS_MASTER_CODE

Name	Address Offset	width	R/W	Description
IC_DATA_CMD	0x10	Refer to Description	R/W	I2C Rx/Tx Data Buffer and Command Reset Value: 0x0 Width: Write: 11 bits when IC_EMPTYFIFO_HOLD_MASTER_EN=1 9 bits when IC_EMPTYFIFO_HOLD_MASTER_EN=0 Read: 12 bits when IC_FIRST_DATA_BYTE_STATUS =1 8 bits when IC_FIRST_DATA_BYTE_STATUS = 0 Notes: With nine or eleven bits required for writes the DW_apb_i2c requires 16-bit data on the APB bus transfers when writing into the transmit FIFO. Eight-bit transfers remain for reads from the receive FIFO. In order for the DW_apb_i2c to continue acknowledging reads,a read command should be written for every byte that is to be received;otherwise the DW_apb_i2c will stop acknowledging
IC_SS_SCL_HCNT	0x14	16bits	R/W	Standard speed I2C Clock SCL High Count Reset Value: IC_SS_SCL_HIGH_COUNT
IC_SS_SCL_LCNT	0x18	16bits	R/W	Standard speed I2C Clock SCL Low Count Reset Value: IC_SS_SCL_LOW_COUNT
IC_FS_SCL_HCNT	0x1C	16bits	R/W	Fast Mode and Fast Mode Plus I2C Clock SCL High Count Reset Value: IC_FS_SCL_HIGH_COUNT
IC_FS_SCL_LCNT	0x20	16bits	R/W	Fast Mode and Fast Mode Plus I2C Clock SCL Low Count Reset Value: IC_FS_SCL_LOW_COUNT
IC_HS_SCL_HCNT	0x24	16bits	R/W	High speed I2C Clock SCL High Count Reset Value: IC_HS_SCL_HIGH_COUNT
IC_HS_SCL_LCNT	0x28	16bits	R/W	High speed I2C Clock SCL Low Count Reset Value: IC_HS_SCL_LOW_COUNT
IC_INTR_STAT	0x2c	15bits	R	I2C Interrupt Status Reset Value: 0x0

Name	Address Offset	width	R/W	Description
IC_INTR_MASK	0x30	15 bits	R/W or Read-only on bits 12 to 14	I2C Interrupt Mask Reset Value: If IC_BUS_CLEAR_FEATURE=0, 14'h8ff If IC_BUS_CLEAR_FEATURE=1, 15'h48ff
IC_RAW_INTR_STAT	0x34	15 bits	R	I2C Raw Interrupt Status Reset Value: 0x0
IC_RX_TL	0x38	8 bits	R/W	I2C Receive FIFO Threshold Reset Value: IC_RX_TL configuration parameter
IC_TX_TL	0x3C	8 bits	R/W	I2C Transmit FIFO Threshold Reset Value: IC_TX_TL configuration parameter
IC_CLR_INTR	0x40	1 bit	R	Clear Combined and Individual Interrupts Reset Value: 0x0
IC_CLR_RX_UNDER	0x44	1 bit	R	Clear RX_UNDER Interrup Reset Value: 0x0
IC_CLR_RX_OVER	0x48	1 bit	R	Clear RX_OVER Interrup Reset Value: 0x0
IC_CLR_TX_OVER	0x4C	1 bit	R	Clear TX_OVER Interrupt Reset Value: 0x0
IC_CLR_RD_REQ	0x50	1 bit	R	Clear RD_REQ Interrupt Reset Value: 0x0
IC_CLR_TX_ABRT	0x54	1 bit	R	Clear TX_ABRT Interrupt Reset Value: 0x0
IC_CLR_RX_DONE	0x58	1 bit	R	Clear RX_DONE Interrupt Reset Value: 0x0
IC_CLR_ACTIVITY	0x5c	1 bit	R	Clear ACTIVITY Interrup Reset Value: 0x0
IC_CLR_STOP_DET	0x60	1 bit	R	Clear STOP_DET Interrupt Reset Value: 0x0
IC_CLR_START_DET	0x64	1 bit	R	Clear START_DET Interrup Reset Value: 0x0
IC_CLR_GEN_CALL	0x68	1 bit	R	Clear GEN_CALL Interrupt Reset Value: 0x0

Name	Address Offset	width	R/W	Description
IC_ENABLE	0x6c	Refer to Description	R/W	I2C Enable Width: 2 bits if IC_TX_CMD_BLOCK = 0 3 bits if IC_TX_CMD_BLOCK = 1 4 bits if IC_BUS_CLEAR_FEATURE = 1 17 bits if IC_SMBUS=1 19 bits if IC_SMBUS_SUSPEND_ALERT=1 Reset Value: 0x0
IC_STATUS	0x70	Refer to Description	R	I2C Status register Width: 7 bits if IC_STAT_FOR_CLK_STRETCH = 0 11 bits if IC_STAT_FOR_CLK_STRETCH = 1 12 bits if IC_BUS_CLEAR_FEATURE=1 17 bits if IC_SMBUS=1 19 bits if IC_SMBUS_ARP=1 21 bits if IC_SMBUS_SUSPEND_ALERT=1 Reset Value: 0x6
IC_TXFLR	0x74	TX_ABW+1	R	Transmit FIFO Level Register Reset Value: 0x0
IC_RXFLR	0x78	RX_ABW+1	R	Receive FIFO Level Register Reset Value: 0x0
IC_SDA_HOLD	0x7C	24 bits	R/W	SDA hold time length register Reset Value: IC_DEFAULT_SDA_HOLD
IC_TX_ABRT_SOURCE	0x80	32 bits	R	I2C Transmit Abort Status Register Reset Value: 0x0
IC_SLV_DATA_NACK_ONLY	0x84	1 bit	R/W	Generate SLV_DATA_NACK Register Reset Value: 0x0
IC_DMA_CR	0x88	2 bits	R/W	DMA Control Register for transmit and receive handshaking interface Reset Value: 0x0
IC_DMA_TDLR	0x8c	TX_ABW	R/W	DMA Transmit Data Level Reset Value: 0x0
IC_DMA_RDLR	0x90	RX_ABW	R/W	DMA Receive Data Level Reset Value: 0x0
IC_SDA_SETUP	0x94	8 bits	R/W	I2C SDA Setup Register Reset Value: IC_DEFAULT_SDA_SETUP configuration parameter
IC_ACK_GENERAL_CALL	0x98	1 bit	R/W	I2C ACK General Call Registe Reset Value: IC_DEFAULT_ACK_GENERAL_CALL configuration paramete
IC_ENABLE_STATUS	0x9C	3 bits	R	I2C Enable Status Register Reset Value: 0x0
IC_FS_SPKLEN	0xA0	8 bits	R/W	ISS and FS spike suppression limit Reset Value: IC_DEFAULT_FS_SPKLEN configuration parameter
IC_HS_SPKLEN	0xA4	8 bits	R/W	HS spike suppression limit Reset Value: IC_DEFAULT_HS_SPKLEN configuration parameter
IC_CLR_RESTART_DET	0xA8	1 bit	R	Clear RESTART_DET Interrupt Reset Value: 0x0

Name	Address Offset	width	R/W	Description
IC_COMP _PARAM_1	0xf4	32 bits	R	Component Parameter Register Reset Value: Reset value depends on configuration parameters.
IC_COMP _VERSION	0xf8	32 bits	R	Component Version ID Reset Value: See the releases table in the AMBA 2 release notes
IC_COMP _TYPE	0xfc	32 bits	R	DesignWare Component Type Register Reset Value: 0x44570140
IC_SCL_STU CK_AT_LOW _TIMEOUT	0xAC	32 bits	R/W	I2C SCL stuck at low timeout register Reset Value: IC_SCL_STUCK_TIMEOUT_DEFAULT
IC_SDA_STU CK_AT_LOW _TIMEOUT	0xB0	32 bits	R/W	I2C SDA Stuck at Low Timeout Reset Value: IC_SDA_STUCK_TIMEOUT_DEFAULT
IC_CLR_SCL _STUCK _DET	0xB4	1 bit	R	Clear SCL Stuck at Low Detect Interrupt Register Reset Value: 0x0
IC_DEVICE _ID	0xB8	24 bits	R	I2C Device ID Reset Value: IC_DEVICE_ID_VALUE
IC_UFM_SCL _HCNT	0x14	16 bits	R/W	Ultra-Fast mode I2C Clock High Count Register Reset Value: IC_UFM_SCL_HIGH_COUNT
IC_UFM_SCL _LCNT	0x18	16 bits	R/W	Ultra-Fast mode I2C Clock Low Count Register Reset Value: IC_UFM_SCL_LOW_COUNT
IC_UFM_TBUF _CNT	0x1c	16 bits	R/W	Ultra-Fast mode TBuf Idle Count Register Reset Value: IC_UFM_TBUF_CNT_DEFAULT
IC_UFM _SPKLEN	0xA0	8 bits	R/W	I2C Ultra-Fast mode Spike suppression Register Reset Value: IC_DEFAULT_UFM_SPKLEN
IC_SMBUS _CLOCK_LOW _SEXT	0xBC	32 bits	R/W	SMBUS Slave Clock Extend Timeout Register
IC_SMBUS _CLOCK_LOW _MEXT	0xC0	32 bits	R/W	SMBUS Master extend clock Timeout Register
IC_SMBUS _THIGH_MAX _IDLE_COUNT	0xC4	16 bits	R/W	SMBus Thigh MAX Bus-Idle count Register
IC_SMBUS _INTR_STAT	0xC8	32 bits	R	I2C SMBUS Interrupt Status Register
IC_SMBUS _INTR_MASK	0xCC	32 bits	R/W	I2C Interrupt Mask Register Register

IC_SMBUS _INTR_RAW _STATUS	0xD0	32 bits	R	I2C SMBUS Raw Interrupt Status Register
IC_CLR _SMBUS_INTR	0xD4	32 bits	W	Clear SMBUS Interrupt Register
IC_OPTIONAL _SAR	0xD8	7 bits	R/W	I2C Optional Slave Address Register
IC_SMBUS _UDID_LSB	0xDC	32 bits	R/W	SMBUS ARP UDID LSB Register

11.1.2 Operation of Interrupt Registers

Table 2 lists the operation of the DW_apb_i2c interrupt registers and how they are set and cleared. Some bits are set by hardware and cleared by software, whereas other bits are set and cleared by hardware.

Table 2: Clearing and Setting of Interrupt Registers

Interrupt Bit Fields	Set by Hardware/Cleared by Software	Set and Cleared by Hardware
MST_ON_HOLD	No	Yes
RESTART_DET	Yes	No
GEN_CALL	Yes	No
START_DET	Yes	No
STOP_DET	Yes	No
ACTIVITY	Yes	No
RX_DONE	Yes	No
TX_ABRT	Yes	No
RD_REQ	Yes	No
TX_EMPTY	No	Yes
TX_OVER	Yes	No
RX_FULL	No	Yes
RX_OVER	Yes	No
RX_UNDER	Yes	No

Figure 1 shows the operation of the interrupt registers where the bits are set by hardware and cleared by software.

11.1.3 Registers and Field Descriptions

This section describes the registers listed in Table 1. Registers are on the pclk domain, but status bits reflect actions that occur in the ic_clk domain. Therefore, there is delay when the pclk register reflects the activity that occurred on the ic_clk side.

Some registers may be written only when the DW_apb_i2c is disabled, programmed by the IC_ENABLE register. Software should not disable the DW_apb_i2c while it is active. If the DW_apb_i2c is in the process of transmitting when it is disabled, it stops as well as deletes the contents of the transmit buffer after the current transfer is complete. The slave continues receiving until the remote master aborts the transfer, in which case the DW_apb_i2c could be disabled. Registers that cannot be written to when the DW_apb_i2c is enabled are indicated in their descriptions.

Unless the clocks pclk and ic_clk are identical (IC_CLK_TYPE = 0), there is a two-register delay for synchronous and asynchronous modes.

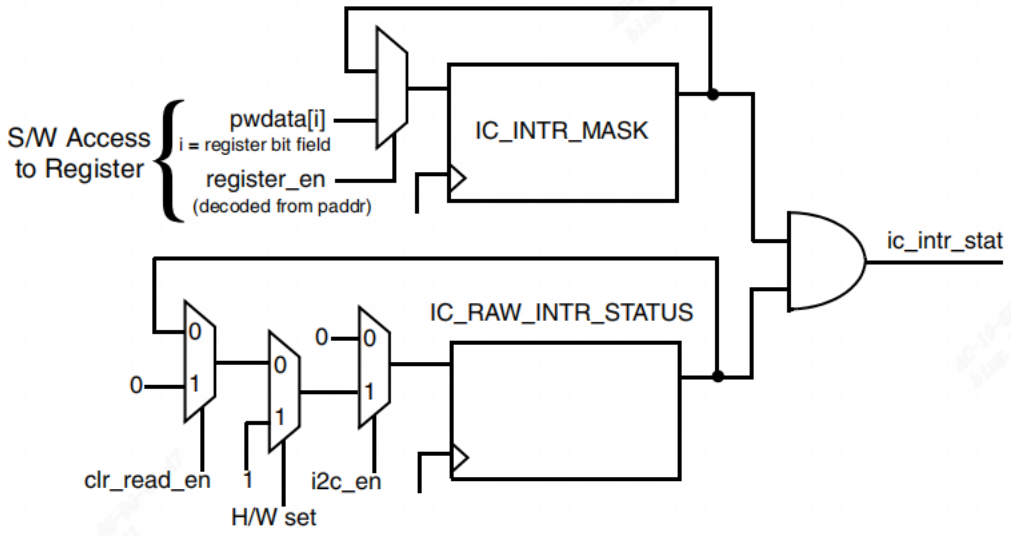

Fig. 1: Interrupt Scheme

IC_CON

- Name: I2C Control Register
- Size: 20 bits
- Address Offset: 0x00
- 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.

This register can be written only when the DW_apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. Writes at other times have no effect.

Table 3: IC_CON Register Fields

Bits	Name	R/W	Description
31:20	Reserved	N/A	Reserved
19	SMBUS_PERSISTANT _SLV_ADDR_EN		This bit controls to enable DW_apb_i2c slave as persistent or non-persistent slave. If the slave is non-PSA then DW_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. Dependencies: This register bit is applicable only when the IC_SMBUS_ARP configuration parameter is set to 1. This bit is applicable only in Slave mode. Reset Value: IC_PERSISTANT_SLV_ADDR_DEFAULT
18	SMBUS_ARP_EN	R/W	This bit controls whether DW_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 DW_apb_i2c slave also includes the generation/validity of PEC byte for Address Resolution Protocol commands. This bit is applicable only in Slave mode. Dependencies: This register bit is applicable only when the IC_SMBUS_ARP configuration parameter is set to 1. Reset Value: 0x0
17	SMBUS_SLAVE_QUICK_CMD_EN	R/W	If this bit is set to 1, DW_apb_i2c slave only receives Quick commands in SMBus Mode. If this bit is set to 0, DW_apb_i2c slave receives all bus protocols but not Quick commands. This bit is applicable only in slave mode. Dependencies: This register bit is applicable only when the IC_SMBUS configuration parameter is set to 1. Reset Value: 0x0

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. Dependencies: This register bit is valid only if configuratio -n parameter IC_OPTIONAL_SAR is set to 1 Reset Value: IC_OPTIONAL_SAR_DEFAULT
15:12	Reserved	R.W	Reserved
11	BUS_CLEAR_FEATURE_CTRL	R/W	In Master Mode: 1'b1: Bus Clear Feature is enabled 1'b0: Bus Clear Feature is disabled In Slave Mode, this register bit is not applicable. Reset Value: 1'b0 Dependencies: This register bit value is applicable only when IC_BUS_CLEAR_FEATURE=1. This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1)
10	STOP_DET_IF_MASTER _ACTIVE	R/W	In Master Mode: 1'b1: Issues the STOP_DET interrupt only when the master is active 1'b0: Issues the STOP_DET irrespective of whether the master is active Reset value: 1'b0 Dependencies: This Register bit value is applicable only when IC_STOP_DET_IF_MASTER_ACTIVE=1. This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1)

9	RX_FIFO_FULL_HLD_CTRL	R/W or R	This bit controls whether DW_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. Dependencies: This register bit value is applicable only when the IC_RX_FULL_HLD_BUS_EN configuration parameter is set to 1. If IC_RX_FULL_HLD_BUS_EN = 0, then this bit is read-only. If IC_RX_FULL_HLD_BUS_EN = 1, then this bit can be read or write. This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0
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. Reset value: 0x0
7	STOP_DET_IFADDRESSED	R/W	In slave mode: 1'b1 – issues the STOP_DET interrupt only when it is addressed. 1'b0 – issues the STOP_DET irrespective of whether it's addressed or not. Dependencies: This register bit value is applicable in the slave mode only (MASTER_MODE = 1'b0) Reset value: 1'b0 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).
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), DW_apb_i2c functions only as a master and does not perform any action that requires a slave. 0: slave is enabled 1: slave is disabled Reset value: IC_SLAVE_DISABLE configuration parameter NOTE: Software should ensure that if this bit is written with '0,' then bit 0 should also be written with a '0'.

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 DW_apb_i2c operations. 0: disable 1: enable When the RESTART is disabled, the DW_apb_i2c master is incapable of performing the following functions: Sending a START BYTE Performing any 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 DW_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. Reset value: IC_RESTART_EN configuration parameter

4	IC_10BITADDR_MASTER or IC_10BITADDR_MASTER _rd_only	R/W or R	If the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to “No” (0), this bit is named IC_10BITADDR_MASTER and controls whether the DW_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 0: 7-bit addressing 1: 10-bit addressing Dependencies: If I2C_DYNAMIC_TAR_UPDATE = 1, then this bit is read-only. If I2C_DYNAMIC_TAR_UPDATE = 0, then this bit can be read or write. Reset value: IC_10BITADDR_MASTER configuration parameter
3	IC_10BITADDR_SLAVE	R/W	When acting as a slave, this bit controls whether the DW_apb_i2c responds to 7- or 10-bit addresses. 0: 7-bit addressing. The DW_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 DW_apb_i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register. Reset value: IC_10BITADDR_SLAVE configuration parameter

Note:Bits 3 and 4 of this register can be programmed differently and in any combination depending on which format is required for the transfers. For example, master mode can be configured with 10-bit addressing and slave mode can be configured with 7-bit addressing.

2:1	SPEED	R/W	These bits control at which speed the DW_apb_i2c operates. Hardware protects against illegal values being programmed by software. register 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 (0 to 100 Kb/s) 2: fast mode (<= 400 Kb/s) or fast mode plus (<= 1000 Kb/s) 3: high speed mode (<= 3.4 Mb/s) NOTE: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: IC_MAX_SPEED_MODE configuration
0	MASTER_MODE	R/W	This bit controls whether the DW_apb_i2c master is enabled. 0: master disabled 1: master enabled Reset value: IC_MASTER_MODE configuration parameter NOTE: Software should ensure that if this bit is written with '1,' then bit 6 should also be written with a '1'.

Certain combinations of the IC_SLAVE_DISABLE (bit 6) and MASTER_MODE (bit 0) result in a configuration error. Table 4 lists the states that result from the combinations of these two bits.

Table 4: States for IC_SLAVE_DISABLE (bit 6) and MASTER_MODE (bit 0)

IC_SLAVE_DISABLE(IC_CON[6])	MASTER_MODE(IC_CON[0])	State
0	0	Slave Device
0	1	Config Error
1	0	Config Error
1	1	Master Device

Note:Because the DW_apb_i2c should only be used either as an I2C master or I2C slave (but not both) at any one time, care should be taken in software that certain combinations of the two bits IC_SLAVE_DISABLE and IC_MASTER_MODE are not programmed into the "IC_CON". In particular, IC_SLAVE_DISABLE and IC_MASTER_MODE must not be set to '0' and '1,' respectively at any given time.

IC_TAR

- Name: I2C Target Address Register
- Size: 12 bits; when I2C_DYNAMIC_TAR_UPDATE = 0 and IC_DEVICE_ID = 0

  13 bits; when I2C_DYNAMIC_TAR_UPDATE = 1 and IC_DEVICE_ID = 0

  14 bits; when IC_DEVICE_ID = 1 irrespective of I2C_DYNAMIC_TAR_UPDATE is set

  17 bits; when IC_SMBUS=1
- Address Offset: 0x04
- Read/Write Access: Read/Write

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. Writes to this register succeed 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:

- DW_apb_i2c is NOT enabled (IC_ENABLE[0] is set to 0); or
- DW_apb_i2c is enabled (IC_ENABLE[0]=1); AND

  DW_apb_i2c is NOT engaged in any Master (tx, rx) operation (IC_STATUS[5]=0); AND

  DW_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)^1

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

- DW_apb_i2c is enabled (IC_ENABLE[0]=1); AND IC_EMPTYFIFO_HOLD_MASTER_EN configuration parameter is set to 1; AND DW_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);1

If the software or application is aware the the DW_apb_i2c is not using the TAR address for the pending commands inthe Tx FIFO, then it is possible to update the TAR address even while the Tx FIFO has entries (IC_STATUS[2]= 0).

Table 5: IC_TAR Register Fields

Bits	Name	R/W	Description
31:17	Reserved	N/A	Reserved
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 DW_apb_i2c. Dependencies: This register bit is applicable only when the IC_SMBUS configuration parameter is set to 1. Reset Value: 0x0
15:14	Reserved	N/A	Reserved
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[6:0] is to be performed by the DW_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. Dependencies: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset Value: 0x0
12	IC_10BITADDR_MASTER	R/W	This bit controls whether the DW_apb_i2c starts its transfers in 7-or10-bit addressing mode when acting as a master. 0: 7-bit addressing 1: 10-bit addressing Dependencies: This bit exists in this register only if the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to Yes (1) Reset value: IC_10BITADDR_MASTER configuration parameter

11	SPECIAL	R/W	This bit indicates whether software performs a Device-ID, 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 Reset value: 0x0
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 DW_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 DW_apb_i2c remains in General Call mode until the SPECIAL bit value (bit 11) is cleared. 1: START BYTE Reset value: 0x0
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. Reset value: IC_DEFAULT_TAR_SLAVE_ADDR configuration parameter 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. Only one direction loopback mode is supported (simplex), not duplex. A master cannot transmit to itself; it can transmit to only a slave.

Note:It is not necessary to perform any write to this register if DW_apb_i2c is enabled as an I2C slave only

IC_SAR

- Name: I2C Slave Address Register
- Size: 10 bits
- Address Offset: 0x08
- Read/Write Access: Read/Write

Table 6: IC_SAR Register Fields

Bits	Name	R/W	Description
31:10	Reserved	N/A	Reserved
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 IC_ENABLE[0] 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. Reset value: IC_DEFAULT_SLAVE_ADDR configuration parameter

Note:It is not necessary to perform any write to this register if DW_apb_i2c is enabled as an I2C master only

IC_HS_MADDR

- Name: I2C High Speed Master Mode Code Address Register
- Size: 3 bits
- Address Offset: 0x0c
- Read/Write Access: Read/Write

This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE = 1).

Table 7: IC_HS_MADDR Register Fields

Bits	Name	R/W	Description
31:3	Reserved	N/A	Reserved
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 IC_ENABLE[0] being set to 0. Writes at other times have no effect. Reset value: IC_HS_MASTER_CODE configuration parameter

Note:It is not necessary to perform any write to this register if DW_apb_i2c is enabled as an I2C slave only.

IC_DATA_CMD

- Name: 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
- Size:
  - Write
    - 11 bits when IC_EMPTYFIFO_HOLD_MASTER_EN=1
    - 9 bits when IC_EMPTYFIFO_HOLD_MASTER_EN=0
  - Read
    - 12 bits when IC_FIRST_DATA_BYTE_STATUS = 1
    - 8 bits when IC_FIRST_DATA_BYTE_STATUS = 0
- Address Offset: 0x10
- Read/Write Access: Read/Write

Note:In order for the DW_apb_i2c to continue acknowledging reads, a read command should be written for every byte that is to be received; otherwise the DW_apb_i2c will stop acknowledging

Table 8: IC_DATA_CMD Register Fields

Bits	Name	R/W	Description
31:12	Reserved	N/A	Reserved
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. Reset value: 0x0 Dependencies: This Register bit value is applicable only when FIRST_DATA_BYTE_STATUS=1. Note: In case of APB_DATA_WIDTH=8: 1.You must perform two APB Reads to IC_DATA_CMD to get status on 11 bit. 2.To read the 11 bit, you must perform the first data byte read [7:0] (offset 0x10) and then perform the second read[15:8](offset 0x11) to know the status of 11 bit (whether the data received in previous read is a first data byte) 3.The 11th bit is an optional read field. You can ignore 2nd byte read [15:8] (offset 0x11) if not interested in the FIRST_DATA_BYTE status.
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.

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.
8	CMD	W	This bit controls whether a read or a write is performed. This bit does not control the direction when the DW_apb_i2c acts as a slave. It controls only the direction when it acts as a master. 1 = Read 0 = Write 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. Dependencies: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0

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 DW_apb_i2c. However, when you read this register, these bits return the value of data received on the DW_apb_i2c interface. Reset value: 0x0

IC_SS_SCL_HCNT

- Name: Standard Speed I2C Clock SCL High Count Register
- Size: 16 bits
- Address Offset: 0x14
- Read/Write Access: Read/Write

This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE = 1).

Table 9: IC_SS_SCL_HCNT Register Fields

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	IC_SS _SCL _HCNT	R/W^1	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 IC_ENABLE[0] being set to 0. Writes at other times have no effect The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the DW_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 DW_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. Reset value: IC_SS_SCL_HIGH_COUNT configuration parameter
Read-only if IC_HC_COUNT_VALUES = 1.			

IC_SS_SCL_LCNT

- Name: Standard Speed I2C Clock SCL Low Count Register
- Size: 16 bits
- Address Offset: 0x18
- Read/Write Access: Read/Write

This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE = 1).

Table 10: IC_SS_SCL_LCNT Register Fields

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	IC_SS _SCL _LCNT	R/W^1	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 IC_ENABLE[0] being set to 0. Writes at other times have no effect The minimum valid value is 8; hardware prevents values less than this being written, and if attempted, results in 8 being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the DW_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. Reset value: IC_SS_SCL_LOW_COUNT configuration parameter
Read-only if IC_HC_COUNT_VALUES = 1.			

IC_FS_SCL_HCNT

- Name: Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
- Size: 16 bits
- Address Offset: 0x1c
- Read/Write Access: Read/Write

This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE = 1).

Table 11: IC_FS_SCL_HCNT

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	IC_FS _SCL _HCNT	R/W^1	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 IC_ENABLE[0] being set to 0. Writes at other times have no effect. The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the DW_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. Reset value: IC_FS_SCL_HIGH_COUNT configuration parameter
Read-only if IC_HC_COUNT_VALUES = 1.			

IC_FS_SCL_LCNT

- Name: Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
- Size: 16 bits
- Address Offset: 0x20
- Read/Write Access: Read/Write

This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE = 1).

Table 12: IC_FS_SCL_LCNT Register Fields

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	IC_FS _SCL _LCNT	R/W^1	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 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 IC_ENABLE[0] being set to 0. Writes at other times have no effect The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. Reset value: IC_FS_SCL_LOW_COUNT configuration parameter
Read-only if IC_HC_COUNT_VALUES = 1.			

IC_HS_SCL_HCNT

- Name: High Speed I2C Clock SCL High Count Register
- Size: 16 bits
- Address Offset: 0x24
- Read/Write Access: Read/Write

This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE = 1).

Table 13: IC_HS_SCL_HCNT Register Fields

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	IC_HS _SCL _HCNT	R/W^1	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. 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 IC_ENABLE[0] being set to 0. Writes at other times have no effect The minimum valid value is 6; hardware prevents values less than this being written, and if attempted results in 6 being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the DW_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. Reset value: IC_HS_SCL_HIGH_COUNT configuration parameter
Read-only if IC_HC_COUNT_VALUES = 1.			

IC_HS_SCL_LCNT

- Name: High Speed I2C Clock SCL Low Count Register
- Size: 16 bits
- Address Offset: 0x28
- Read/Write Access: Read/Write

This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE = 1).

Table 14: IC_HS_SCL_LCNT Register Fields

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	IC_HS _SCL _LCNT	R/W^1	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 IC_ENABLE[0] being set to 0. Writes at other times have no effect The minimum valid value is 8; hardware prevents values less than this being written, and if attempted results in 8 being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the DW_apb_i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than 8 then the count value gets changed to 8. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. Reset value: IC_HS_SCL_LOW_COUNT configuration parameter
Read-only if IC_HC_COUNT_VALUES = 1.			

IC_INTR_STAT

- Name: I2C Interrupt Status Register
- Size: 15 bits
- Address Offset: 0x2C
- Read/Write Access: Read

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.

Table 15: IC_INTR_STAT Register Fields

Bits	Name	R/W	Description
31:15	Reserved	N/A	Reserved
14	R_SCL _STUCK _AT _LOW	R	See IC_RAW_INTR_STAT for a detailed description of this bit. Dependencies: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA _FAST_MODE=1). Reset value: 0x0
13	R_MST_ON_HOLD	R	See “IC_RAW_INTR_STAT” for a detailed description of this bit. Reset value: 0x0
12	R_RESTART_DET	R	See IC_RAW_INTR_STAT for a detailed description of these bits.
11	R_GEN_CALL		Dependencies: R_RX_DONE and R_RD_REQ are not applicable in
10	R_START_DET		Ultra Fast speed mode (IC_ULTRA_FAST_MODE = 1).
9	R_STOP_DET		Reset value: 0x0
8	R_ACTIVITY		
7	R_RX_DONE		
6	R_TX_ABRT		
5	R_RD_REQ		
4	R_TX_EMPTY		
3	R_TX_OVER		
2	R_RX_FULL		
1	R_RX_OVER		
0	R_RX_UNDER		

IC_INTR_MASK

- Name: I2C Interrupt Mask Register
- Size: 15 bits
- Address Offset: 0x30
- Read/Write Access: Read/Write
  - If configuration parameter IC_SLV_RESTART_DET = 0, bit 13 is read only.
  - If configuration parameter I2C_DYNAMIC_TAR_UPDATE = 0

    or IC_EMPTYFIFO_HOLD_MASTER_EN = 0, bit 14 is read only.
  - If configuration parameter IC_BUS_CLEAR_FEATURE = 0, bit 15 is read only.

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.

Table 16: IC_INTR_MASK Register Fields

Bits	Name	R/W	Description
31:15	Reserved	N/A	Reserved
14	M_SCL_STUCK _AT_LOW	R or R/W	This bit masks the R_SCL_STUCK_AT_LOW interrupt bit in the IC_INTR_STAT register This bit is enabled only when IC_BUS_CLEAR_FEATURE = 1. Dependencies: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset Value: 0x1
13	M_MST_ON_HOLD	R or R/W	This bit masks the R_MST_ON_HOLD interrupt bit in the IC_INTR_STAT register Dependencies: If I2C_DYNAMIC_TAR_UPDATE = 1 and IC_EMPTYFIFO_HOLD_MASTER_EN = 1, then M_MST_ON_HOLD is read/write. Otherwise M_MST_ON_HOLD is read-only. Reset value: 14'h8ff
12	M_RESTART_DET	R or R/W	This bit masks the R_RESTART_DET interrupt status bit in the IC_INTR_STAT register. Dependencies: If IC_SLV_RESTART_DET_EN = 1, then M_RESTART_DET is read/write. Otherwise M_RESTART_DET is read-only. Reset value: 14'h8ff
11	M_GEN_CALL	R/W	These bits mask their corresponding interrupt status bits in the IC_INTR_STAT register. Dependencies: M_RX_DONE and M_RD_REQ are not applicable in Ultra Fast speed mode (IC_ULTRA_FAST_MODE = 1). Reset value: 14'h8ff
10	M_START_DET		
9	M_STOP_DET		
8	M_ACTIVITY		
7	M_RX_DONE		
6	M_TX_ABRT		
5	M_RD_REQ		
4	M_TX_EMPTY		
3	M_TX_OVER		
2	M_RX_FULL		
1	M_RX_OVER		
0	M_RX_UNDER		

IC_RAW_INTR_STAT

- Name: I2C Raw Interrupt Status Register
- Size: 15 bits
- Address Offset: 0x34
- Read/Write Access: Read Unlike the IC_INTR_STAT register, these bits are not masked so they always show the true status of the DW_apb_i2c.

Table 17: IC_RAW_INTR_STAT Register Fields

Bits	Name	R/W	Description
31:15	Reserved	N/A	Reserved
14	SCL_STUCK _AT_LOW	R	Indicates whether the SCL Line is stuck at low for the IC_SCL_STUCK_LOW_TIMOUT number of ic_clk periods. Enabled only when IC_BUS_CLEAR_FEATURE = 1 Dependencies: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA _FAST_MODE=1) Reset Value: 0x0
13	MST_ON_HOLD	R	Indicates whether a master is holding the bus and the Tx FIFO is empty. Enabled only when I2C_DYNAMIC_TAR_UPDATE = 1 and IC_EMPTYFIFO_HOLD_MASTER_EN = 1 Reset value: 0X0
12	RESTART_DET	R	Indicates whether a RESTART condition has occurred on the I2C interface when DW_apb_i2c is operating in slave mode and the slave is the addressed slave 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 DW_apb_i2c does not generate the RESTART_DET interrupt. Reset value: 0x0
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 DW_apb_i2c or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. DW_apb_i2c stores the received data in the Rx buffer. Reset value: 0x0
10	START_DET	R	Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether DW_apb_i2c is operating in slave or master mode. Reset value: 0x0

9	STOP_DET	R	Indicates whether a STOP condition has occurred on the I2C interface regardless of whether DW_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 is generated only if the 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 is issued only if the master is active. If IC_CON[10]=1'b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is issued irrespective of whether the master is active. Reset value: 0x0
8	ACTIVITY	R	This bit captures DW_apb_i2c activity and stays set until it is cleared. There are four ways to clear it: Disabling the DW_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 DW_apb_i2c module is idle, this bit remains set until cleared,indicating that there was activity on the bus. Reset value: 0x0

7	RX_DONE	R	When the DW_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. Dependencies: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0
6	TX_ABRT	R	This bit indicates if DW_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 DW_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. RX FIFO is flushed because of TX_ABRT is controlled by the coreConsultant parameter IC_AVOID_RX_FIFO_FLUSH_ON_TX_ABRT. Reset value: 0x0

5	RD_REQ	R	This bit is set to 1 when DW_apb_i2c is acting as a slave and another I2C master is attempting to read data from DW_apb_i2c. The DW_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. Dependencies: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0
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. Reset value: 0x0

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. Reset value: 0x0
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 IC_ENABLE[0] is set to 0, regardless of the activity that continues. Reset value: 0x0
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 DW_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. Reset value: 0x0
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. Reset value: 0x0

IC_RX_TL

- Name: I2C Receive FIFO Threshold Register
- Size: 8bits
- Address Offset: 0x38
- Read/Write Access: Read/Write

Table 18: IC_RX_TL Register Fields

Bits	Name	R/W	Description
31:8	Reserved	N/A	Reserved
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. Reset value: IC_RX_TL configuration parameter

IC_TX_TL

- Name: I2C Transmit FIFO Threshold Register
- Size: 8 bits
- Address Offset: 0x3c
- Read/Write Access: Read/Write

Table 19: IC_TX_TL Register Fields

Bits	Name	R/W	Description
31:8	Reserved	N/A	Reserved
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 256 entries. Reset value: IC_TX_TL configuration parameter

IC_CLR_INTR

- Name: Clear Combined and Individual Interrupt Register
- Size: 1 bit
- Address Offset: 0x40
- Read/Write Access: Read

Table 20: IC_CLR_INTR Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
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. Reset value: 0x0

IC_CLR_RX_UNDER

- Name: Clear RX_UNDER Interrupt Register
- Size: 1 bit
- Address Offset: 0x44
- Read/Write Access: Read

Table 21: IC_CLR_RX_UNDER Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_RX_UNDER	R	Read this register to clear the RX_UNDER interrupt (bit 0) of the IC_RAW_INTR_STAT register. Reset value: 0x0

IC_CLR_RX_OVER

- Name: Clear RX_OVER Interrupt Register
- Size: 1 bit
- Address Offset: 0x48
- Read/Write Access: Read

Table 22: IC_CLR_RX_OVER Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_RX_OVER	R	Read this register to clear the RX_OVER interrupt (bit 1) of the IC_RAW_INTR_STAT register. Reset value: 0x0

IC_CLR_TX_OVER

- Name: Clear TX_OVER Interrupt Register
- Size: 1 bit
- Address Offset: 0x4c
- Read/Write Access: Read

Table 23: IC_CLR_TX_OVER Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_TX_OVER	R	Read this register to clear the TX_OVER interrupt (bit 3) of the IC_RAW_INTR_STAT register. Reset value: 0x0

IC_CLR_RD_REQ

- Name: Clear RD_REQ Interrupt Register
- Size: 1 bit
- Address Offset: 0x50
- Read/Write Access: Read
- Dependencies: This Register is not applicable in Ultra-Fast speed mode(IC_ULTRA_FAST_MODE=1)

Table 24: IC_CLR_RD_REQ Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_RD_REQ	R	Read this register to clear the RD_REQ interrupt (bit 5) of the IC_RAW_INTR_STAT register. Reset value: 0x0

IC_CLR_TX_ABRT

- Name: Clear TX_ABRT Interrupt Register
- Size: 1 bit
- Address Offset: 0x54
- Read/Write Access: Read

Table 25: IC_CLR_TX_ABRT Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
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. Reset value: 0x0

IC_CLR_RX_DONE

- Name: Clear RX_DONE Interrupt Register
- Size: 1 bit
- Address Offset: 0x58
- Read/Write Access: Read
- Dependencies: This Register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1)

Table 26: IC_CLR_RX_DONE Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_RX_DONE	R	Read this register to clear the RX_DONE interrupt (bit 7) of the IC_RAW_INTR_STAT register. Reset value: 0x0

IC_CLR_ACTIVITY

- Name: Clear ACTIVITY Interrupt Register
- Size: 1 bit
- Address Offset: 0x5c
- Read/Write Access: Read

Table 27: IC_CLR_ACTIVITY Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
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. Reset value: 0x0

IC_CLR_STOP_DET

- Name: Clear STOP_DET Interrupt Register
- Size: 1 bit
- Address Offset: 0x60
- Read/Write Access: Read

Table 28: IC_CLR_STOP_DET Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_STOP _DET	R	Read this register to clear the STOP_DET interrupt (bit 9) of the IC_RAW_INTR_STAT register. Reset value: 0x0

IC_CLR_START_DET

- Name: Clear START_DET Interrupt Register
- Size: 1 bit
- Address Offset: 0x64
- Read/Write Access: Read

Table 29: IC_CLR_START_DET Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_START _DET	R	Read this register to clear the START_DET interrupt (bit 10) of the IC_RAW_INTR_STAT register. Reset value: 0x0

IC_CLR_GEN_CALL

- Name: Clear GEN_CALL Interrupt Register
- Size: 1 bit
- Address Offset: 0x68
- Read/Write Access: Read

Table 30: IC_CLR_GEN_CALL Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_GEN _CALL	R	Read this register to clear the GEN_CALL interrupt (bit 11) of IC_RAW_INTR_STAT register. Reset value: 0x0

IC_ENABLE

- Name: I2C Enable Register
- Size: 19 bits
- Address Offset: 0x6c
- Read/Write Access: Read/Write
  - Bit 2 is read only when IC_TX_CMD_BLOCK_DEFAULT=0
  - Bit 3 is read only when IC_BUS_CLEAR_FEATURE = 0
  - Bit 16 is read only when IC_SMBUS=0.
  - Bit 17 and 18 are read only when IC_SMBUS_SUSPEND_ALERT=0.

Table 31: IC_ENABLE Register Fields

Bits	Name	R/W	Description
31:19	Reserved	N/A	Reserved
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. Dependencies: This Register bit value is applicable only when IC_SMBUS_SUSPEND_ALERT=1 Reset value: 0x0
17	SMBUS _SUS- PEND _EN	R/W	The SMBUS_SUSPEND_EN register bit is used to control assertion and deassertion of SMBSUS signal. 0: De-assert SMBSUS signal 1: Assert SMBSUS signal Dependencies: This Register bit value is applicable only when IC_SMBUS_SUSPEND_ALERT=1 Reset value: 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. Dependencies: This Register bit value is applicable only when IC_SMBUS=1 Reset value: 0x0
15:4	Reserved	N/A	Reserved

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. This bit is enabled only when IC_BUS_CLEAR_FEATURE = 1. Dependencies: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset Value: 0x0
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. Reset value: IC_TX_CMD_BLOCK_DEFAULT Dependencies: This Register bit value is applicable only when IC_TX_CMD_BLOCK =1. 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 the master is in the Idle state (IC_STATUS[5] == 0). Any further commands put in the Tx FIFO are not executed until TX_CMD_BLOCK bit is unset.

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. Reset value: 0x0
0	ENABLE	R/W	Controls whether the DW_apb_i2c is enabled. 0: Disables DW_apb_i2c (TX and RX FIFOs are held in an erased state) 1: Enables DW_apb_i2c Software can disable DW_apb_i2c while it is active. However, it is important that care be taken to ensure that DW_apb_i2c is disabled properly. When DW_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 DW_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 DW_apb_i2c stops the current transfer at the end of the current byte 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 DW_apb_i2c. Reset value: 0x0

IC_STATUS

- Name: I2C Status Register
- Size: 32 bits
- Address Offset: 0x70
- Read/Write Access: Read

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 to 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

Table 32: IC_STATUS Register Fields

Bits	Name	R/W	Description
31:19	Reserved	N/A	Reserved
20	SMBUS _ALERT _STATUS	R	This bit indicates whether the status of the input signal is ic_smbus_alert_in_n. This signal is asserted when the SMBus Alert signal is asserted by the SMBus Device. Dependencies: Enabled only when IC_SMBUS_SUSPEND_ALERT=1 is set to 1. Reset Value: 0x0
19	SMBUS _SUSPEND _STATUS	R	This bit indicates whether the status of the input signal is ic_smbus_sus_in_n. This signal is asserted when the SMBus Suspend signal is asserted by the SMBus Host. Dependencies: Enabled only when IC_SMBUS_SUSPEND_ALERT=1 is set to 1. Reset Value: 0x0
18	SMBUS _SLAVE _ADDR _RESOLVED	R	This bit indicates whether the SMBus Slave address (ic_sar[6:0]) is Resolved by ARP Master. Dependencies: Enabled only when IC_SMBUS_ARP=1 is set to 1. Reset Value: 0x0
17	SMBUS _SLAVE _ADDR _VALID	R	This bit indicates whether the SMBus Slave address (ic_sar[6:0]) is valid or not. Dependencies: Enabled only when IC_SMBUS_ARP=1 is set to 1. Reset Value: 0x0
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. Dependencies: Enabled only when IC_SMBUS=1 is set to 1. Reset Value: 0x0

15:12	Reserved	N/A	Reserved
11	SDA_STUCK_NOT_RECOVERED	R	This bit indicates that an SDA stuck at low is not recovered after the recovery mechanism. This bit is enabled only when IC_BUS_CLEAR_FEATURE = 1. Reset Value: 0x0
10	SLV_HOLD_RX_FIFO_FULL	R	This bit indicates the BUS Hold in Slave mode due to the Rx FIFO being Full and an additional byte being received (this kind of Bus hold is applicable if IC_RX_FULL_HLD_BUS_EN is set to 1). Reset value: 0x0 Dependencies: This Register bit value is applicable only when IC_STAT_FOR_CLK_STRETCH=1.
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. Reset Value: 0x0 Dependencies: This Register bit value is applicable only when IC_STAT_FOR_CLK_STRETCH=1.
8	MST_HOLD_RX_FIFO_FULL_VALID	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). Reset Value: 0x0 Dependencies: This Register bit value is applicable only when IC_STAT_FOR_CLK_STRETCH=1
7	MST_HOLD_TX_FIFO_FULL_VALID		If the IC_EMPTYFIFO_HOLD_MASTER_EN parameter is set to 1, the DW_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). Reset value: 0x0 Dependencies: This Register bit value is applicable only when IC_STAT_FOR_CLK_STRETCH=1
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. 0: Slave FSM is in IDLE state so the Slave part of DW_apb_i2c is not Active 1: Slave FSM is not in IDLE state so the Slave part of DW_apb_i2c is Active Reset value: 0x0
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. 0: Master FSM is in IDLE state so the Master part of DW_apb_i2c is not Active 1: Master FSM is not in IDLE state so the Master part of DW_apb_i2c is Active NOTE: IC_STATUS[0]—that is, ACTIVITY bit—is the OR of SLV_ACTIVITY and MST_ACTIVITY bits. Reset value: 0x0

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. 0: Receive FIFO is not full 1: Receive FIFO is full Reset value: 0x0
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. 0: Receive FIFO is empty 1: Receive FIFO is not empty Reset value: 0x0
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. 0: Transmit FIFO is not empty 1: Transmit FIFO is empty Reset value: 0x1
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. 0: Transmit FIFO is full 1: Transmit FIFO is not full Reset value: 0x1
0	ACTIVITY	R	I2C Activity Status. Reset value: 0x0

IC_TXFLR

- Name: I2C Transmit FIFO Level Register
- Size: TX_ABW + 1
- Address Offset: 0x74
- Read/Write Access: Read

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.

Table 33: IC_TXFLR Register Fields

Bits	Name	R/W	Description
31:TX_ABW+1	Reserved	N/A	Reserved
TX_ABW:0	TXFLR	R	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. Reset value: 0x0

IC_RXFLR

- Name: I2C Receive FIFO Level Register
- Size: RX_ABW + 1
- Address Offset: 0x78
- Read/Write Access: Read

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.

Table 34: IC_RXFLR Register Fields

Bits	Name	R/W	Description
31:RX_ABW+1	Reserved	N/A	Reserved
RX_ABW:0	RXFLR	R	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO. Reset value: 0x0

IC_SDA_HOLD

- Name: I2C SDA Hold Time Length Register
- Size: 24 bits
- Address Offset: 0x7C
- Read/Write Access: Read/Write

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 rare 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.

Table 35: IC_SDA_HOLD Register Fields

Bits	Name	R/W	Description
31:24	Reserved	N/A	Reserved
23:16	IC_SDA _RX_HOLD	R/W	Sets the required SDA hold time in units of ic_clk period, when DW_apb_i2c acts as a reciever. Reset value: IC_DEFAULT_SDA_HOLD
15:0	IC_SDA _TX_HOLD	R/W	Sets the required SDA hold time in units of ic_clk period, when DW_apb_i2c acts as a transmitter. Reset value: IC_DEFAULT_SDA_HOLD

IC_TX_ABRT_SOURCE

- Name: I2C Transmit Abort Source Register
- Size: 32 bits
- Address Offset: 0x80
- Read/Write Access: Read

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.

Table 36: IC_TX_ABRT_SOURCE Register Fields

Bits	Name	R/W	Description	Role of DW_apb_i2c
31:23	TX_FLUSH _CNT	R	This field indicates the number of Tx FIFO data commands that are flushed due to TX_ABRT interrupt. It is cleared whenever I2C is disabled. Reset value: 0x0	Master-Transmitter or Slave-Transmitter
22:21	Reserved	R	These bits are reserved.	
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. Reset Value: 0x0	Master
19	ABRT _DEVICE _SLVADDR _NOACK	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 Reset Value: 0x0	Master
18	ABRT _DEVICE _NOACK	R	This is a master-mode-only bit. Master initiates the DEVICE_ID transfer and the device ID sent is not acknowledged by any slave. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master
17	ABRT_SDA _STUCK _AT_LOW	R	This is a master-mode-only bit. Master detects the SDA is Stuck at low for the IC_SDA_STUCK_AT_LOW_TIMEOUT value of ic_clks. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master
16	ABRT_USER _ABRT	R	This is a master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1]). Reset value: 0x0	Master-Transmitter

15	ABRT _SLVRD _INTX	R	1: 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. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Slave-Transmitter
14	ABRT_SLV _ARBLOST	R	1: Slave lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) 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 DW_apb_i2c no longer own the bus. Reset value: 0x0	Slave-Transmitter
13	ABRT _SLVFLUSH _TXFIFO	R	1: 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. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Slave-Transmitter

12	ARB_LOST	R	1:Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master-Transmitter or Slave-Transmitter
11	ABRT _MASTER _DIS	R	1:User tries to initiate a Master operation with the Master mode disabled. Reset value: 0x0	Master-Transmitter or Master-Receiver
10	ABRT_10B _RD _NORSTRT	R	1:The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the master sends a read command in 10-bit addressing mode. Dependencies: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1). Reset value: 0x0	Master-Receiver

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 re-asserted. 1:The restart is disabled (IC_RESTART_EN bit (IC_CON[5]) = 0) and the user is trying to send a START Byte. Reset value: 0x0	Master
8	ABRT_HS _NORSTRT	R	1: 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. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master-Transmitter or Master-Receiver
7	ABRT _SBYTE _ACKDET	R	1: Master has sent a START Byte and the START Byte was acknowledged (wrong behavior). Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master
6	ABRT_HS _ACKDET	R	1: Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior). Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master
5	ABRT _GCALL _READ	R	1: DW_apb_i2c in master mode sent a General Call but the user programmed the byte following the General Call to be a read rom the bus (IC_DATA _CMD[9] is set to 1). Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master-Transmitter
4	ABRT _GCALL _NOACK	R	1: DW_apb_i2c in master mode sent a General Call and no slave on the bus acknowledged the General Call. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master-Transmitter

3	ABRT _TXDATA _NOACK	R	1: This is a master-mode only bit. Master has received an acknowledgement for the address, but when it sent data byte(s) following the address, it did not receive an acknowledge from the remote slave(s). Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master-Transmitter
2	ABRT _10ADDR2 _NOACK	R	1: Master is in 10-bit address mode and the second address byte of the 10-bit address was not acknowledged by any slave. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master-Transmitter or Master-Receiver
1	ABRT _10ADDR1 _NOACK	R	1: Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master-Transmitter or Master-Receiver
0	ABRT_7B _ADDR _NOACK	R	1: Master is in 10-bit address mode and the the address sent was not acknowledged by any slave. Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1) Reset value: 0x0	Master-Transmitter or Master-Receiver

IC_SLV_DATA_NACK_ONLY

- Name: Generate Slave Data NACK Register
- Size: 1 bit
- Address Offset: 0x84
- Read/Write Access: Read/Write
- Dependency: This Register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1).

The register is used to generate a NACK for the data part of a transfer when DW_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:

- DW_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.

Table 37: IC_SLV_DATA_NACK_ONLY Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	NACK _RX_HOLD	R/W	Generate NACK. This NACK generation only occurs when DW_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 Reset value: 0x0

IC_DMA_CR

- Name: DMA Control Register
- Size: 2 bits
- Address Offset: 0x88
- Read/Write Access: Read/Write

  This register is only valid when DW_apb_i2c is configured with a set of DMA Controller interface signals(IC_HAS_DMA = 1). When DW_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.

Table 38: IC_DMA_CR Register Fields

Bits	Name	R/W	Description
31:2	Reserved	N/A	Reserved
1	TDMAE	R/W	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. 0 = Transmit DMA disabled 1 = Transmit DMA enabled Reset value: 0x0
0	RDMAE	R/W	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. 0 = Receive DMA disabled 1 = Receive DMA enabled Reset value: 0x0

IC_DMA_TDLR

- Name: DMA Transmit Data Level Register
- Size: TX_ABW–1:0
- Address Offset: 0x8c
- Read/Write Access: Read/Write

This register is only valid when the DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_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.

Table 39: IC_DMA_TDLR Register Fields

Bits	Name	R/W	Description
31:TX_ABW	Reserved	N/A	Reserved
TX_ABW–1:0	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. Reset value: 0x0

IC_DMA_RDLR

- Name: I2C Receive Data Level Register
- Size: RX_ABW–1:0
- Address Offset: 0x90
- Read/Write Access: Read/Write

This register is only valid when DW_apb_i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When DW_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.

Table 40: IC_DMA_RDLR Register Fields

Bits	Name	R/W	Description
31:RX_ABW	Reserved	N/A	Reserved
RX_ABW–1:0	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. Reset value: 0x0

IC_SDA_SETUP

- Name: I2C SDA Setup Register
- Size: 8 bits
- Address Offset: 0x94
- Read/Write Access: Read/Write
- Dependency: This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1).

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—by holding SCL low when DW_apb_i2c services a read request while operating as a slave-transmitter. The relevant I2C requirement is tSU:DAT (note 4) as detailed in the I2C Bus Specification. This register must be programmed with a value equal to or greater than 2.

Writes to this register succeed only when IC_ENABLE[0] = 0.

Note:The length of setup time is calculated using [(IC_SDA_SETUP - 1) * (ic_clk_period)], so if the user requires 10 ic_clk periods of setup time, they should program a value of 11.The IC_SDA_SETUP register is only used by the DW_apb_i2c when operating as a slave transmitter.

Table 41: IC_SDA_SETUP Register Fields

Bits	Name	R/W	Description
31:8	Reserved	N/A	Reserved
7:0	SDA_SETUP	R/W	SDA Setup. It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz, IC_SDA_SETUP should be programmed to a value of 11. IC_SDA_SETUP must be programmed with a minimum value of 2. Default Reset value: 0x64, but can be hardcoded by setting the IC_DEFAULT_SDA_SETUP configuration parameter.

IC_ACK_GENERAL_CALL

- Name: I2C ACK General Call Register
- Size: 1 bit
- Address Offset: 0x98
- Read/Write Access: Read/Write
- Dependency: This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1).

The register controls whether DW_apb_i2c responds with an ACK or NACK when it receives an I2C General Call address. This register is applicable only when the DW_apb_i2c is in the slave mode.

Table 42: IC_ACK_GENERAL_CALL Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	ACK_GEN _CALL	R/W	ACK General Call. When set to 1, DW_apb_i2c responds with a ACK (by asserting ic_data_oe) when it receives a General Call. When set to 0, the DW_apb_i2c does not generate General Call interrupts. Default Reset value: 0x1, but can be hardcoded by setting the IC_DEFAULT_ACK_GENERAL_CALL configuration parameter.

IC_ENABLE_STATUS

- Name: I2C Enable Status Register
- Size: 3 bits
- Address Offset: 0x9C
- Read/Write Access: Read

The register is used to report the DW_apb_i2c hardware status when IC_ENABLE[0] is set from 1 to 0; that is, when DW_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 disabling the DW_apb_i2c depends on I2C bus activities.

Table 43: IC_ENABLE_STATUS Register Fields

Bits	Name	R/W	Description
31:3	Reserved	N/A	Reserved
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 setting IC_ENABLE[0] from 1 to 0. When read as 1, DW_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 DW_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, DW_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. Reset value: 0x0
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 setting bit 0 of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to bit 0 of IC_ENABLE while: (a) DW_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, DW_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 DW_apb_i2c (IC_SAR register) OR if the transfer is completed before bit 0 of 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 DW_apb_i2c has a chance to NACK a transfer, and bit 0 of IC_ENABLE has been set to 0, then this bit will also be set to 1. When read as 0, DW_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. Reset value: 0x0
0	IC_EN	R	ic_en Status. This bit always reflects the value driven on the output port ic_en. When read as 1, DW_apb_i2c is deemed to be in an enabled state. When read as 0, DW_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). Reset value: 0x0

IC_FS_SPKLEN

- Name: I2C SS and FS Spike Suppression Limit Register
- Size: 8 bits
- Address Offset: 0xA0
- Read/Write Access: Read/Write
- Dependency: This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1).

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 standard mode, fast mode, or fast mode plus. The relevant I2 C requirement is tSP (Table 4) as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1.

Table 44: IC_FS_SPKLEN Register Fields

Bits	Name	R/W	Description
31:8	Reserved		
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 are filtered out by the spike suppression logic This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] 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. Reset value: IC_DEFAULT_FS_SPKLEN configuration parameter

IC_HS_SPKLEN

- Name: I2C HS Spike Suppression Limit Register
- Size: 8 bits
- Address Offset: 0xA4
- Read/Write Access: Read/Write
- Dependency: This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1).

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 mode. The relevant I2C requirement is tSP (Table 6) 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.

Table 45: IC_HS_SPKLEN Register Fields

Bits	Name	R/W	Description
31:8	Reserved		
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 are filtered out by the spike suppression logic This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] 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. This register 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. Reset value: IC_DEFAULT_HS_SPKLEN configuration parameter

IC_CLR_RESTART_DET

- Name: Clear RESTART_DET Interrupt Register
- Size: 1 bit
- Address Offset: 0xA8
- Read/Write Access: Read

Table 46: IC_CLR_RESTART_DET Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_RESTART_DET	R	Read this register to clear the RESTART_DET interrupt (bit 12) of the IC_RAW_INTR_STAT register. Dependencies: This register is present only when IC_SLV_RESTART_DET_EN = 1. Reset value: 0x0

IC_COMP_PARAM_1

- Name: Component Parameter Register 1
- Size: 32 bits
- Address Offset: 0xf4
- Read/Write Access: Read

Note:This is a constant read-only register that contains encoded information about the component's parameter settings. The reset value depends on coreConsultant parameter(s).

Table 47: IC_COMP_PARAM_1 Register Fields

Bits	Name	R/W	Description
31:24	Reserved	N/A	Reserved
23:16	TX_BUFFER _DEPTH	R	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter. 0x00 = Reserved 0x01 = 2 0x02 = 3 ... 0xFF = 256
15:8	RX_BUFFER _DEPTH	R	The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter. 0x00 = Reserved 0x01 = 2 0x02 = 3 ... 0xFF = 256
7	ADD_ENCODED _PARAMS	R	The value of this register is derived from the IC_ADD_ENCODED_PARAMS coreConsultant parameter. 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 0: False 1: True
6	HAS_DMA	R	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter. 0: False 1: True

6	HAS_DMA	R	The value of this register is derived from the IC_HAS_DMA coreConsultant parameter. 0: False 1: True
5	INTR_IO	R	The value of this register is derived from the IC_INTR_IO coreConsultant parameter. 0: Individual 1: Combined
4	HC_COUNT _VALUES	R	The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter. 0: False 1: True
3:2	MAX_SPEED_MODE	R	The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter. 0x0 = Reserved 0x1 = Standard 0x2 = Fast 0x3 = High Dependency: This field is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1)
1:0	APB_DATA_WIDTH	R	The value of this register is derived from the APB_DATA_WIDTH coreConsultant parameter. 0x0 = 8 bits 0x1 = 16 bits 0x2 = 32 bits 0x3 = Reserved

IC_COMP_VERSION

- Name: I2C Component Version Register
- Size: 32 bits
- Address Offset: 0xf8
- Read/Write Access: Read

Table 48: IC_COMP_VERSION Register Fields

Bits	Name	R/W	Description
31:0	IC_COMP_VERSION	R	Specific values for this register are described in the Releases Table in the AMBA 2 release notes

IC_COMP_TYPE

- Name: I2C Component Type Register
- Size: 32 bits
- Address Offset: 0xfc
- Read/Write Access: Read

Table 49: IC_COMP_TYPE Register Fields

Bits	Name	R/W	Description
31:0	IC_COMP_TYPE	R	Designware Component Type number = 0x44_57_01_40. This assigned unique hex value is constant and is derived from the two ASCII letters “DW” followed by a 16-bit unsigned number.

IC_SCL_STUCK_AT_LOW_TIMEOUT

- Name: I2C SCL Stuck at Low Timeout
- Size: 32 bits
- Address Offset: 0xAC
- Read/Write Access: Read/Write
- Dependencies: This register is not applicable in Ultra-Fast speed mode(IC_ULTRA_FAST_MODE = 1).

This register is used to store the duration, measured in ic_clk cycles, used to generate an Interrupt (SCL_STUCK_AT_LOW) if SCL is held low for the IC_SCL_STUCK_LOW_TIMEOUT duration.

Table 50: IC_SCL_STUCK_AT_LOW_TIMEOUT Register Field

Bits	Name	R/W	Description
31:0	IC_SCL_STUCK LOW_TIMEOUT	R/W	DW_apb_i2c generates the interrupt to indicate SCL stuck at low if it detects the SCL stuck at low for the IC_SCL_STUCK_LOW_TIMEOUT in units of ic_clk period. Reset Value: IC_SCL_STUCK_TIMEOUT_DEFAULT

IC_SDA_STUCK_AT_LOW_TIMEOUT

- Name: I2C SDA Stuck at Low Timeout
- Size: 32 bits
- Address Offset: 0xB0
- Read/Write Access: Read/Write
- Dependencies: This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1).

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.

Table 51: IC_SDA_STUCK_AT_LOW_TIMEOUT Register Field

Bits	Name	R/W	Description
31:0	IC_SDA_STUCK _LOW_TIMEOUT	R/W	DW_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. Reset Value: IC_SDA_STUCK_TIMEOUT_DEFAULT

IC_CLR_SCL_STUCK_DET

- Name: Clear SCL Stuck at Low Detect Interrupt Register
- Size: 1 bit
- Address Offset: 0xB4
- Read/Write Access: Read
- Dependencies: This register is not applicable in Ultra-Fast speed mode (IC_ULTRA_FAST_MODE=1)

Table 52: IC_CLR_SCL_STUCK_DET Register Fields

Bits	Name	R/W	Description
31:1	Reserved	N/A	Reserved
0	CLR_SCL_STUCK	R	Read this register to clear the SCL_STUCK_DET interrupt (bit 14) of the IC_RAW_INTR_STAT register. Reset value: 0x0

IC_DEVICE_ID

- Name: I2C Device ID
- Size: 24 bits
- Address Offset: 0xb8
- Read/Write Access: Read
- Dependencies: This register is not applicable in Ultra-Fast speed mode(IC_ULTRA_FAST_MODE=1).

This register contains the Device-ID of the component, which includes 12 bits of manufacturer name, 9 bits of part identification and 3 bits of die-version.

Table 53: IC_DEVICE_ID Register Fields

Bits	Name	R/W	Description
31:24	Reserved	N/A	Reserved
23:0	DEVICE-ID	R	Contains the Device-ID of the component assigned through the configuration parameter IC_DEVICE_ID_VALUE Reset Value: IC_DEVICE_ID_VALUE

IC_UFM_SCL_HCNT

- Name: Ultra-Fast mode I2C Clock High Count Register
- Size: 16 bits
- Address Offset: 0x14
- Read/Write Access: Read/Write
- Dependencies: This is register is present only if parameter IC_ULTRA_FAST_MODE is set to 1.

Table 54: Ultra-Fast Mode SCL High Counter Register Field Description

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	IC_UFM_SCL_HCNT	R/W^1	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 IC_ENABLE[0] 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 DW_apb_i2c. The lower byte must be programmed first and then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. Reset value: IC_UFM_SCL_HIGH_COUNT configuration parameter

Read-only if IC_HC_COUNT_VALUES = 1.

IC_UFM_SCL_LCNT

- Name: Ultra-Fast mode I2C Clock Low Count Register
- Size: 16 bits
- Address Offset: 0x18
- Read/Write Access: Read
- Dependencies: This is register is present only if parameter IC_ULTRA_FAST_MODE is set to 1.

Table 55: Ultra-Fast Mode SCL Low Counter Register Field Description

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	IC_UFM_SCL_LCNT	R/W^1	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 IC_ENABLE[0] 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 DW_apb_i2c. The lower byte must be programmed and then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. Reset value: IC_UFM_SCL_LOW_COUNT configuration parameter

Read-only if IC_HC_COUNT_VALUES = 1.

IC_UFM_SPKLEN

- Name: I2C Ultra-Fast mode Spike suppression Register
- Size: 8 bits
- Address Offset: 0xA0

- Read/Write Access: Read/Write
- Dependencies: This is register is present only if parameter IC_ULTRA_FAST_MODE is set to 1.

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.

Table 56: UFM Spike Suppression Register

Bits	Name	R/W	Description
31:8	Reserved	N/A	Reserved
7:0	IC_UFM_SPKLEN	R/W	This register must be set before any I2C bus transaction can occur 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 are filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled, which corresponds to IC_ENABLE[0] 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. Reset value: IC_DEFAULT_UFM_SPKLEN configuration parameter.

IC_UFM_TBUF_CNT

- Name: Ultra-Fast mode TBuf Idle Count Register
- Size: 16 bits
- Address Offset: 0x1c
- Read/Write Access: Read/Write
- Dependencies: This is register is present only if parameter IC_ULTRA_FAST_MODE is set to 1.

Table 57: Ultra-Fast Mode Tbuf Counter Register Field Description

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	IC_UFM_TBUF_CNT	R/W^1	This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the tBuf Idle time count for Ultra-Fast speed. This register can be written only when the I2C interface is disabled which corresponds to IC_ENABLE[0] 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 DW_apb_i2c. The lower byte must be programmed first and then the upper byte is programmed. When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only. NOTE:The DW_apb_i2c will add 9 ic_clks after tBuf time is expired to generate START on the Bus. Reset value: IC_UFM_TBUF_CNT_DEFAULT configuration parameter

Read-only if IC_HC_COUNT_VALUES = 1.

IC_SMBUS_CLOCK_LOW_SEXT

- Name: SMBUS Slave Clock Extend Timeout Register
- Size: 32 bits
- Address Offset: 0xBC
- Read/Write Access: Read/Write

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 DW_apb_i2c is disabled, whichcorresponds 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.

Table 58: IC_SMBUS_CLOCK_LOW_SEXT Register Field Description

Bits	Name	R/W	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. Reset Value: IC_SMBUS_CLOCK_LOW_SEXT_DEFAULT

IC_SMBUS_CLOCK_LOW_MEXT

- Name: SMBUS Master extend clock Timeout Register
- Size: 32 bits
- Address Offset: 0xC0
- Read/Write Access: Read/Write

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 DW_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.

Table 59: SMBUS Master extend clock Timeout Register Field Description

Bits	Name	R/W	Description
31:0	SMBUS_CLK_LOW_MEXT_TIMEOUT	R/W	This field is used to detect the Master extend SMBus clock (SCL) 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. Reset Value: IC_SMBUS_CLOCK_LOW_SEXT_DEFAULT

IC_SMBUS_THIGH_MAX_IDLE_COUNT

- Name: SMBus Thigh MAX Bus-Idle count Register
- Size: 16 bits
- Address Offset: 0xC4

- Read/Write Access: Read/Write

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 DW_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.

Table 60: SMBus Thigh MAX Bus-Idle count Register Field Descriptions

Bits	Name	R/W	Description
31:16	Reserved	N/A	Reserved
15:0	SMBUS_THIGH_MAX _BUS_IDLE_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 to ensure that a transfer is not currently in progress. The values in this register are in units of ic_clk period. Reset value: IC_SMBUS_RST_IDLE_CNT_DEFAULT

IC_SMBUS_INTR_STAT

- Name: I2C SMBUS Interrupt Status Register
- Size: 32 bits
- Address Offset: 0xC8
- Read/Write Access: Read

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.

Table 61: I2C SMBUS Interrupt Status Register Field Descriptions

Bits	Name	R/W	Description
31:11	Reserved	N/A	Reserved
10	R_SMBUS_ALERT _DET	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
9	R_SMBUS_SUSPEND _DET	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
8	R_SLV_RX_PEC _NACK	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
7	R_ARP_ASSGN ADDR_CMD_DET	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
6	R_ARP_GET_UDID _CMD_DET	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
5	R_ARP_RST_CMD _DET	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
4	R_ARP_PREPARE CMD_DET	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
3	R_HOST_NOTIFY _MST_DET	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
2	R_QUICK_CMD_DET	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
1	R_MST_CLOCK _TIMEOUT	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0
0	R_SLV_CLOCK _EXTND_TIMEOUT	R	See IC_SMBUS_INTR_RAW_STATUS for a detailed description of this bit. Reset value: 0x0

IC_SMBUS_INTR_MASK

- Name: I2C Interrupt Mask Register
- Size: 32 bits
- Address Offset: 0xcc
- Read/Write Access: Read/Write

Table 62: I2C Interrupt Mask Register Field Descriptions

Bits	Name	R/W	Description
31:11	Reserved	N/A	Reserved
10	M_SMBUS_ALERT	R/W	This bit masks the R_SMBUS_ALERT_DET interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS_SUSPEND_ALERT=1. Reset Value: 0x1
9	M_SMBUS_SUSPEND _DET	R/W	This bit masks the R_SMBUS_SUSPEND_DET interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS_SUSPEND_ALERT=1. Reset Value: 0x1
8	M_SLV_RX_PEC _NACK	R/W	This bit masks the R_SLV_RX_PEC_NACK interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS_ARP=1. Reset Value: 0x1
7	M_ARP_ASSGN _ADDR_CMD_DET	R/W	This bit masks the R_ARP_ASSGN_ADDR_CMD_DET interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS_ARP=1. Reset Value: 0x1
6	M_ARP_GET_UDID _CMD_DET	R/W	This bit masks the R_ARP_GET_UDID_CMD_DET interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS_ARP=1. Reset Value: 0x1
5	M_ARP_RST_CMD _DET	R/W	This bit masks the R_ARP_RST_CMD_DET interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS_ARP=1. Reset Value: 0x1
4	M_ARP_PREPARE _CMD_DET	R/W	This bit masks the R_ARP_PREPARE_CMD_DET interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS_ARP=1. Reset Value: 0x1
3	M_HOST_NOTIFY _MST_DET	R/W	This bit masks the R_HOST_NOTIFY_DET interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS=1. Reset Value: 0x1
2	M_QUICK_CMD_DET	R/W	This bit masks the R_QUICK_CMD_DET interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS=1. Reset Value: 0x1
1	M_MST_CLOCK _EXTND_TIMEOUT	R/W	This bit masks the R_MST_CLOCK_EXTND_TIMEOUT interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS=1. Reset Value: 0x1
0	M_SLV_CLOCK _EXTND_TIMEOUT	R/W	This bit masks the R_SLV_CLOCK_EXTND_TIMEOUT interrupt bit in the IC_SMBUS_INTR_STAT register. This bit is enabled only when IC_SMBUS=1. Reset Value: 0x1

IC_SMBUS_INTR_RAW_STATUS

- Name: I2C SMBUS Raw Interrupt Status Register
- Size: 32 bits
- Address Offset: 0xd0
- Read/Write Access: Read only

Table 63: I2C SMBUS Raw Interrupt Status Register Field Descriptions

Bits	Name	R/W	Description
31:11	Reserved	N/A	Reserved
10	SMBUS_ALERT_DET	R	Indicates whether a SMBALERT (ic_smbalert_in_n) signal is driven low by the slave. Dependencies: This register bit is valid only if configuration parameter IC_SMBUS_SUSPEND_ALERT is set to 1. Reset Value: 0x0
9	SMBUS_SUSPEND _DET	R	Indicates whether a SMBSUS (ic_smbsus_in_n) signal is driven low by the Host. Dependencies: This register bit is valid only if configuration parameter IC_SMBUS_SUSPEND_ALERT is set to 1. Reset Value: 0x0
8	SLV_RX_PEC_NACK	R	Indicates whether a Slave generates a NACK for the PEC Byte of the ARP command from the slave. Dependencies: This register bit is valid only if configuration parameter IC_SMBUS_ARP is set to 1. Reset Value: 0x0
7	ARP_ASSGN_ADDR _CMD_DET	R	Indicates whether an Assign Address ARP command has been received. Dependencies: This register bit is valid only if configuration parameter IC_SMBUS_ARP is set to 1. Reset Value: 0x0
6	ARP_GET_UDID _CMD_DET	R	Indicates whether a General or directed Get UDID ARP command has been received. Dependencies: This register bit is valid only if configuration parameter IC_SMBUS_ARP is set to 1. Reset Value: 0x0

5	ARP_RST_CMD_DET	R	Indicates whether a General or Directed Reset ARP command has been received. Dependencies: This register bit is valid only if configuration parameter IC_SMBUS_ARP is set to 1. Reset Value: 0x0
4	ARP_PREPARE_CMD_DET	R	Indicates whether a Prepare to ARP command has been received. Dependencies: This register bit is valid only if configuration parameter IC_SMBUS_ARP is set to 1. Reset Value: 0x0
3	HOST_NTFY_MST_DET	R	Indicates whether a Host Notify command has been received. Dependencies: This register bit is valid only if configuration parameter IC_SMBUS is set to 1. Reset Value: 0x0
2	QUICK_CMD_DET	R	Indicates whether a Quick command has been received on the SMBus interface regardless of whether DW_apb_i2c is operating in slave or master mode. This bit is enabled only when IC_SMBUS=1 is set to 1. Reset Value: 0x0
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 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
0	SLV_CLOCK_EXTND_TIMEOUT	R	Indicates whether the transaction from Slave (that is, from START to STOP) exceeds IC_SMBUS_CLOCK_LOW_SEXT time. This bit is enabled only when IC_SMBUS=1 IC_CON[0]=1

IC_CLR_SMBUS_INTR

- Name: Clear SMBUS Interrupt Register
- Size: 32 bits
- Address Offset: 0xD4
- Read/Write Access: Write only

Table 64: Clear SMBUS Interrupt Register Field Descriptions

Bits	Name	R/W	Description
31:11	Reserved	N/A	Reserved
10	CLR_SMBUS_ALERT _DET	W	Write this register to clear the SMBUS_ALERT_DET interrupt (bit 10) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
9	CLR_SMBUS _SUSPEND_DET	W	Write this register to clear the R_SMBUS_SUSPEND_DET interrupt (bit 9) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
8	CLR_SLV_RX _PEC_NACK	W	Write this register to clear the SLV_RX_PEC_NACK interrupt (bit 8) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
7	CLR_ARP_ASSGN _ADDR_CMD_DET	W	Write this register to clear the ARP_ASSGN_ADDR_CMD_DET interrupt (bit 7) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
6	CLR_ARP_GET _UDID_CMD_DET	W	Write this register to clear the ARP_GET_UDID_CMD_DET interrupt (bit 6) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
5	CLR_ARP_RST _CMD_DET	W	Write this register to clear the ARP_RST_CMD_DET interrupt (bit 5) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
4	CLR_ARP_PREPARE _CMD_DET	W	Write this register to clear the ARP_PREPARE_CMD_DET interrupt (bit 4) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
3	CLR_HOST_NOTIFY _MST_DET	W	Write this register to clear the HOST_NOTIFY_MST_DET interrupt (bit 3) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
2	CLR_QUICK_CMD _DET	W	Write this register to clear the QUICK_CMD_DET interrupt (bit 2)the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
1	CLR_MST_CLOCK _EXTND_TIMEOUT	W	Write this register to clear the MST_CLOCK_EXTND_TIMEOUT interrupt (bit 1) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0
0	CLR_SLV_CLOCK _EXTND_TIMEOUT	W	Write this register to clear the SLV_CLOCK_EXTND_TIMEOUT interrupt (bit 0) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0

IC_OPTIONAL_SAR

- Name: I2C Optional Slave Address Register
- Size: 7 bits
- Address Offset: 0xD8
- Read/Write Access: Read/Write

Table 65: I2C Optional Slave Address Register Field Descriptions

Bits	Name	R/W	Description
15:11	Reserved	N/A	Reserved
6:1	IC_OPTIONAL_SAR	R/W	Optional Slave address for DW_apb_i2c when operating as a slave in SMBus Mode. Dependencies: This register bit is valid only if configuration parameter IC_OPTIONAL_SAR is set to 1. Reset Value: IC_OPTIONAL_SAR_DEFAULT
0	CLR_SLV_CLOCK _EXTND_TIMEOUT	W	Write this register to clear the SLV_CLOCK_EXTND_TIMEOUT interrupt (bit 0) of the IC_SMBUS_RAW_INTR_STAT register. Reset value: 0x0

IC_SMBUS_UDID_LSB

- Name: SMBUS ARP UDID LSB Register
- Size: 32 bits
- Address Offset: 0xDC
- Read/Write Access: Read/Write
- Dependencies: This register is present only if IC_SMBUS_ARP =1.

This register can be written only when the DW_apb_i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS_ARP 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.

Table 66: SMBUS ARP UDID LSB Register Field Description

Bits	Name	R/W	Description
31:0	IC_SMBUS_ARP _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. Reset Value: IC_SMBUS_UDID_LSB_DEFAULT

CHAPTER

TWELVE

SPI FLASH

12.1 Register Definition

Table 1: SPIFMC Register Summary

Offset	Name	Description	Default	Width
0x00~0x03	SPI_CTRL	SPI Control Register	0x0008_C013	32bit
0x04	CE_CTRL	CE Control Register	0x00	8bit
0x08~0x09	DLY_CTRL	Delay Control Register	0x0300	16bit
0x0c	DMMR	DMMR Control Register	0x01	8bit
0x10~0x11	TRAN_CSR	Transfer Control and Status Register	0x3B00	16bit
0x14~0x15	TRAN_NUM	Transfer Number Register	0x0000	16bit
0x18~0x1b	FF_PORT	FIFO Port Register	NA	32bit
0x20	FF_PT	FIFO Pointer Register	0x00	8bit
0x28	INT_STS	Interrupt Status Register	0x00	8bit
0x2c	INT_EN	Interrupt Enable Register	0x00	8bit

12.1.1 SPI_CTRL,SPI Control(0x00~0x03)

default value:0x0008_C013

Table 2: SPI_CTRL

Bit	Attribute	Default	Description
31:22	R(0)	10'b0	Reserved
21	R(0)/W(1)	0	SRst: Soft Reset
20	RW	0	LSBF: Least Significant Bit First
19:16	RW	4'b1000	FrameLen
15	RW	1	WpOL
14	RW	1	HoldOL
13	RW	0	CPOL
12	RW	0	CPHA
11	R(0)	0	Reserved
10:0	RW	10'h13	SckDiv

SRst: Soft Reset

Write 1 Resets each state machine and interrupt flag bit. If the system switches from SPI Flash boot, the controller switches from boot mode to common mode A soft reset should also be performed beforethe mode.

LSBF: Least Significant Bit First

0: Frame MSB first

1: Frame LSB first

FrameLen: Frame Length

FrameLen is the length of the sent and received frames (in bits). If FrameLen is 0, the frame length is 16 bits. Don't Supports frame length 1.

WpOL: WP Pin Output Level

Output level value of WP pin.

HoldOL: HOLD Pin Output Level

The output level of the HOLD pin.

CPOL: Clock Polarity

0: The SCK is low when it is idle

1: The SCK is high when it is idle

CPHA: Clock Phase

0: SCK starts sampling data on the first clock edge after slice selection is valid

1: After slice selection is valid, SCK starts sampling data along the second clock edge

{CPOL, CPHA} consists of four operating modes of SPI, and its sequence diagram is shown in 5 SPI Mode

SckDiv:SPI Clock Divider

SCK frequency = HCLK frequency / (2(SckDiv+ 1))

12.1.2 CE_CTRL,CE Control(0x04)

default value: 0x00

Table 3: CE_CTRL

Bit	Attribute	Default	Description
7:2	R(0)	0	Reserved
1	RW	0	CEManualEn
0	RW	0	CEManual

CEManualEn: CE Manual Enable

0: The level of the CE pin is controlled by the hardware state machine

1: The level of the CE pin is controlled by the CEManual register

CEManual:

CEManual controls the level value of the CE pin. CEManual is only effective when CEManualEn is 1.

12.1.3 DLY_CTRL, Delay Control (0x08~0x09)

default value: 0x0300

Table 4: DLY_CTRL

Bit	Attribute	Default	Description
15:12	R(0)	4'b0	Reserved
11:8	RW	4'b0011	CET*
7:4	R(0)	4'b0	Reserved
3:0	RW	4'b0	FmIntvl

CET: CE Pre and Post Time

CET controls how long the CE remains in force with respect to the first clock edge of SCK before atransmission begins and how long it remains in force with respect to the last clock edge of SCK afterthe transmission ends. This time is calculated as T = TSCK * (CET+1)

FmIntvl: Frame Interval

FmIntvl controls the frame spacing of two adjacent frames of data: T = TSCK * FmIntvl (no SCK pulsewithin the frame spacing). When FmIntvl is 0, there is no frame spacing.

12.1.4 DMMR, Direct Memory Mapping Read (0x0C)

default value: 0x01

Table 5: DMMR

Bit	Attribute	Default	Description
7:1	R(0)	7'b0	Reserved
0	RW	1	DMMR

When the DMMR bit is 1, the read address on the AHB is mapped directly to the SPI Flash, and the controller automatically reads data from the SPI Flash address without software setting related commands and addresses. In this case, the SPI Flash can be used as ROM.

attention:

1. When DMMR is 1, registers in IP can be written but not read;
2. Before entering the DMMR mode, the software must be correctly configured with BusWidth, FastMode, CntnsRead and AddrBN registers.

12.1.5 TRAN_CSR,Transfer Control and Status Register (0x10~0x11)

default value: 0x3B00

Table 6: TRAN_CSR

Bit	Attribute	Default	Description
15	R/W(1)	0	GoBusy
14	R(0)	0	Reserved
13:12	RW	11	FFTrgLvl
11	RW	1	WithCmd
10:8	RW	011	AddrBN
7	RW	0	MISOLevel
6	RW	0	DmaEn
5:4	RW	0	BusWidth
3	R(0)	0	FastMode
2	RW	0	CntnsRead
1:0	RW	0	TranMode

MISOLevel: MISO Pin Level

MISOLevel is the level value of the miso i pin

GoBusy:

Writing 0 to this bit does not work. Writing 1 to this position 1 starts a transmission. After the transmission ends, this bit is automatically cleared to zero. Before initiating a new transfer, the software should query the register, and a new transfer can be initiated only when the register is 0.

DmaEn: Transmit DMA Enable

0: DMA Disable

1: DMA Enable

When TranMode is set to 11, the sending and receiving process is carried out at the same time, and DMA transmission is not supported. Therefore, DmaEn must be set to 0.

FFTrgLvl: FIFO Trigger Level

FFTrgLvl controls the conditions under which the FIFO generates interrupts and DMA requests.

00: 1 Byte

01: 2 Bytes

10: 4 Bytes

11: 8 Bytes

For Transmit, when the number of free Byte in FIFO is greater than or equal to the number of Byte defined by FFTrgLvl, interrupt and DMA request are generated;

For Receive, an interrupt and DMA request are generated when the number of valid bytes in the FIFO is greater than or equal to the number of bytes defined by FFTrgLvl.

WithCmd: With Command

0: The current transmission does not carry commands

1: indicates the current transport tape command

AddrBN: Address Byte Number

Indicates the number of bytes (including dummy byte and mode byte) of the current Flash transport address field. 0 indicates that there is no address field.

BusWidth: Bus Width

00: 1 bit bus

01: 2 bit bus

10: 4 bit bus

11: Reserved

FastMode:

0: Normal Mode

1: Fast Mode

CntnsRead: Continuous Read

If this bit is 1, the hardware will send 0xa0 as the mode bit after the address to achieve Continuous Read, or 00 as the mode bit (see 8.1BBh and EBh command timing). Note: This bit is only valid if the DMMR is 1.

TranMode: Transfer Mode

00: No Tx, No Rx

01: Rx only

10: Tx only 11: Tx and Rx

TranMode Indicates the sending and receiving mode for transmitting data except commands and addresses.

Since the same FIFO is used for sending and receiving, the amount of data sent when TranMode is 11 cannot exceed the FIFO capacity (8 bytes). The software should first write all the data to be sent into the FIFO and configure TRAN_NUM, and then start the transmission. Software can no longer access FIFO until TranDoneInt is generated; After the TranDoneInt is generated, the FIFO will store TRAN_NUM received data.

12.1.6 TRAN_NUM, Transfer Number (0x14~0x15)

default value: 0x0000

Table 7: TRAN_NUM

Bit	Attribute	Default	Description
15:0	R/W	0	TRAN_NUM

TRAN_NUM Indicates the number of frames (excluding commands and addresses) sent and received in a transmission. Frames are set in SPI_CTRL.

If TRAN_NUM is 0, it indicates 65536 frames. This register is invalid when TranMode is 00.

12.1.7 FF_PORT, FIFO Port (0x18~0x1b)

default value: NA

Table 8: FF_PORT

Bit	Attribute	Default	Description
31:0	RW	XXXX	FF_PORT

Writing the register base address can write data to the FIFO, reading the register base address can read data from the FIFO. Note: Only the base address of the register can be read or written.

For sending frames:

	32-bit write	16-bit write	8-bit write
Frame length less than or equalto 8 (occupies one byte of FIFO space)	Write 4 frames at a time (lower frames are sent first)	Write 2 frames at a time (lower frames are sent first)	Write 1 frame at a time
Frame size of 8 (takes up two bytes of FIFO space)	Write 2 frames at a time (lower frames are sent first)	Write 1 frame at a time	Write a frame twice, first write the lower 8 bits of the frame,then write the higher frame

For receiving frames:

	32-bit read	16-bit read	8-bit read
Frame length less than or equalto 8 (occupies one byte of FIFO space)	Read 4 frames at a time (the lower frame is the first frame received)	Read 2 frames at a time (the lower frame is the first frame received)	Read 1 frame at a time
Frame size of 8 (takes up two bytes of FIFO space)	Read 2 frames at a time (the lower frame is the first frame received)	Read 1 frame at a time	Read a frame twice. first read the lower 8 bits of the frame,then write the higher frame

The data storage format is shown in *FmLen = 6, 1 byte per frame* and *FmLen = 10, 2 bytes per frame*, taking the frame length of 6 and 10 as examples respectively.

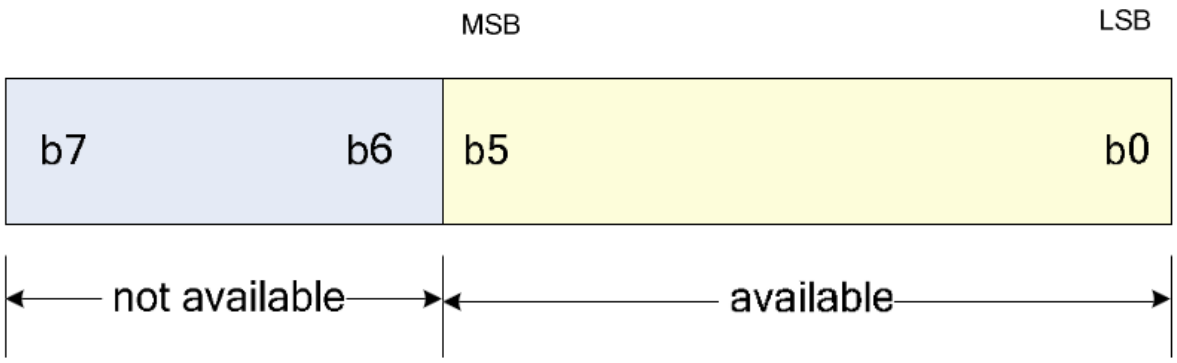

Fig. 1: FmLen = 6, 1 byte per frame

Issues with FF_PORT's data receiving and sending of low-order frames first:

When writing data to the FF_PORT register, the lower frame is the first frame received. For example, when writing 8-bit cmd and 24-bit address addr[23:0] to FF_PORT, The correct format for writing to the FF_PORT register is {addr[7:0],addr[15:8], addr[23:16], cmd[7:0]}, so that data is read and written to addr[23:0] of the SPI Flash.

If SPIFMC is used to read and write the SPI Flash, the written 32bit data[31:0] can be directly written into the FF_PORT. At this time, the data written into the SPI Flash is received first because the lower FIFO frame. The actual data

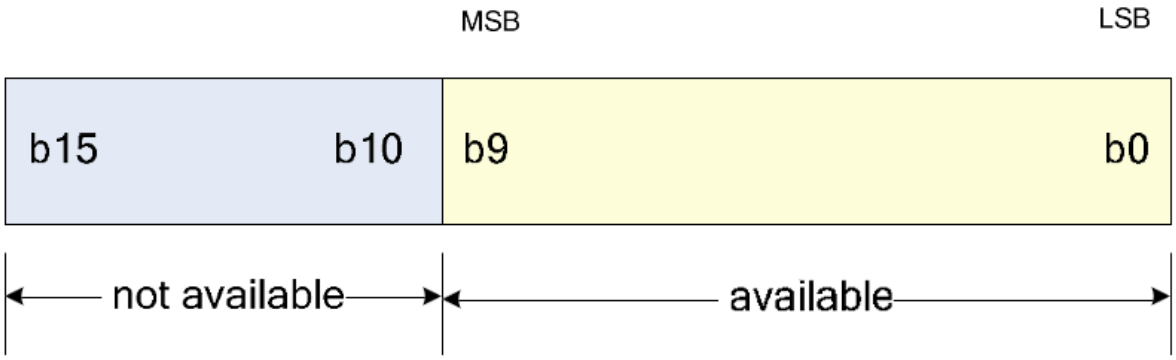

Fig. 2: FmLen = 10, 2 bytes per frame

is {data[7:0], data[15:8], data[23:16],data[31:24]}, but when the data is read by SPIFMC, the lower frame of the FF_PORT data is sent first, and the read data is the correct data[31:0].

If the data written to the SPI Flash needs to be read using a method other than SPIFMC, it should be written to the FF_PORT register {data[7:0], data[15:8], data[23:16], data[31:24]}, The actual data stored in the SPI Flash is data[31:0].

12.1.8 FF_PT, FIFO Pointer (0x20)

default value: 0x00

Table 9: FF_PT

Bit	Attribute	Default	Description
7:4	R(0)	0	Reserved
3:0	RW	0	FF_PT

Read the register to get the number of valid data bytes in the FIFO, write the register to flush the FIFO.

12.1.9 INT_STS, Interrupt Status (0x28)

default value: 0x00

Table 10: INT_STS

Bit	Attribute	Default	Description
7:6	R(0)	00	Reserved
5	R/W(0)	0	TxFrameInt
4	R/W(0)	0	RxFrameInt
3	R/W(0)	0	WrFFInt
2	R/W(0)	0	RdFFInt
1	R(0)	0	Reserved
0	R/W(0)	0	TranDoneInt*

If the CPU writes 0 bits to the Reserved bit, the corresponding bit is cleared and 1 bits are ignored.

TxFrameInt:

This interrupt is generated once for each successful frame of data sent.

RxFrameInt:

This interrupt is generated once for each successfully received frame of data.

WrFFInt: Write FIFO Interrupt

The CPU writes frame data to the FIFO after receiving this interrupt.

RdFFInt: Read FIFO Interrupt

The CPU reads frame data from the FIFO after receiving this interrupt.

TranDoneInt:Transfer Done Interrupt

The interrupt marks the completion of a transfer.

12.1.10 INT_EN, Interrupt Enable (0x2c)

default value: 0x00

Table 11: INT_EN

Bit	Attribute	Default	Description
7:6	R(0)	00	Reserved
5	R/W	0	TxFrameIntEn
4	R/W	0	RxFrameIntEn
3	R/W	0	WrFFIntEn
2	R/W	0	RdFFIntEn
1	R(0)	0	Reserved
0	R/W	0	TranDoneIntEn*

1: indicates that the corresponding interrupt is enabled

0: disables the corresponding Interrupt. For details, see INT_STS, Interrupt Status (0x28).

CHAPTER

THIRTEEN

GPIO

13.1 Registers

This section describes the programmable registers of the DW_apb_gpio.

13.1.1 Bus Interface

The DW_apb_gpio peripheral has a standard AMBA 2.0 APB interface for reading and writing the interna registers. This peripheral supports APB data bus widths of 8, 16, or 32 bits, which is set with the APB_DATA_WIDTH parameter.

Figure 1 shows the read/write busses between the DW_apb and the APB slave.

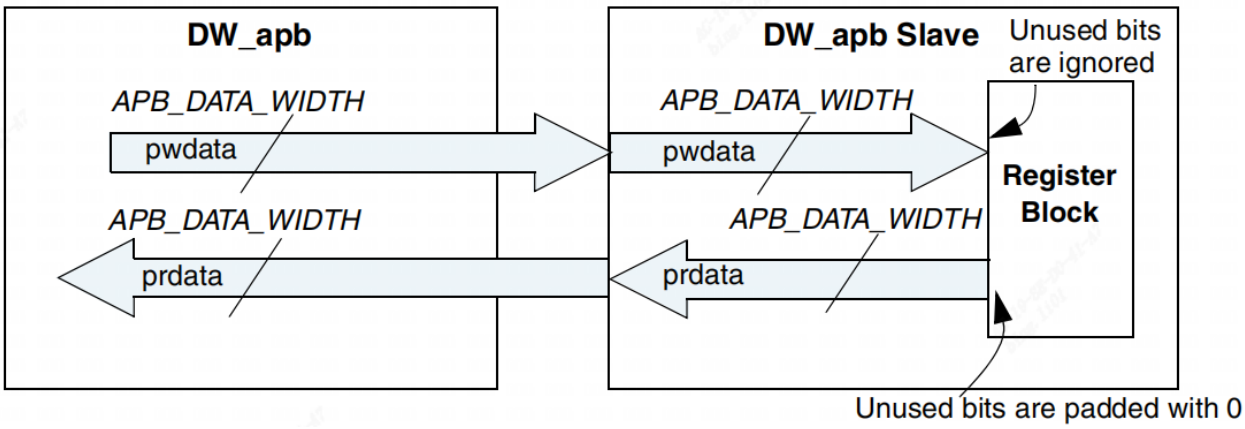

Fig. 1: Relationship Between DW_apb and Slave Data Widths

13.1.2 Register Memory Map

Table 1 shows the memory map for the DW_apb_gpio peripheral.

Table 1: Memory Map of DW_apb_gpio

Name	Address Offset	width	R/W	Description
gpio_swporta_dr	0x00	See Description	R/W	Port A data register Width: GPIO_PWIDTH_A Reset Value: GPIO_SWPORTA_RESET
gpio_swporta_ddr	0x04	See Description	R/W	Port A data direction register Width: GPIO_PWIDTH_A Reset Value: GPIO_DFLT_DIR_A (for all bits)
gpio_swporta_ctl	0x08	See Description	R/W	Port A data source register Width: 1 bit if GPIO_PORTA_SINGLE _CTL = 1, or GPIO_PWIDTH_A otherwise Reset Value: GPIO_DFLT_SRC_A Bit is repeated GPIO_PWIDTH_A times if GPIO_PORTA_SINGLE_CTL = 0
gpio_swporta_dr	0x0c	See Description	R/W	Port B data register Width: GPIO_PWIDTH_B Reset Value: GPIO_SWPORTB_RESET
gpio_swporta_ddr	0x10	See Description	R/W	Port B data direction register Width: GPIO_PWIDTH_B Reset Value: GPIO_DFLT_DIR_B (for all bits)
gpio_swportb_ctl	0x14	See Description	R/W	Port B data source register Width: 1 bit if GPIO_PORTB_SINGLE _CTL = 1, or GPIO_PWIDTH_B otherwise Reset Value: GPIO_DFLT_SRC_B Bit is repeated GPIO_PWIDTH_B times if GPIO_PORTB_SINGLE_CTL = 0
gpio_swportc_dr	0x18	See Description	R/W	Port C data register Width: GPIO_PWIDTH_C Reset Value: GPIO_SWPORTC_RESET

gpio_swportc_ddr	0x1c	See Description	R/W	Port C data direction register Width: GPIO_PWIDTH_C Reset Value: GPIO_DFLT_DIR_C (for all bits)
gpio_swportc_ctl	0x20	See Description	R/W	Port C data source register Width: 1 bit if GPIO_PORTC_SINGLE _CTL = 1, or GPIO_PWIDTH_C otherwise Reset Value: GPIO_DFLT_SRC_C Bit is repeated GPIO_PWIDTH_C times if GPIO_PORTC_SINGLE_CTL = 0
gpio_swportd_dr	0x24	See Description	R/W	Port D data register Width: GPIO_PWIDTH_D Reset Value: GPIO_SWPORTD_RESET
gpio_swportd_ddr	0x28	See Description	R/W	Port D data direction register Width: GPIO_PWIDTH_D Reset Value: GPIO_DFLT_DIR_D (for all bits)
gpio_swportd_ctl	0x2c	See Description	R/W	Port D data source register Width: 1 bit if GPIO_PORTD_SINGLE _CTL = 1, or GPIO_PWIDTH_D otherwise Reset Value: GPIO_DFLT_SRC_D Bit is repeated GPIO_PWIDTH_D times if GPIO_PORTD_SINGLE_CTL = 0
gpio_inten	0x30	See Description	R/W	Interrupt enable register Width: GPIO_PWIDTH_A Reset Value: 0x0
gpio_intmask	0x34	See Description	R/W	Interrupt mask register Width: GPIO_PWIDTH_A Reset Value: 0x0
gpio_inttype _level	0x38	See Description	R/W	Interrupt level register Width: GPIO_PWIDTH_A Reset Value: 0x0

gpio_int _polarity	0x3c	See Description	R/W	Interrupt polarity register Width: GPIO_PWIDTH_A Reset Value: 0x0
gpio_intstatus	0x40	See Description	R	Interrupt status of Port A Width: GPIO_PWIDTH_A Reset Value: 0x0
gpio_raw _intstatus	0x44	See Description	R	Raw interrupt status of Port A (premasking) Width: GPIO_PWIDTH_A Reset Value: 0x0
gpio_debounce	0x48	See Description	R/W	Debounce enable register Width: GPIO_PWIDTH_A Reset Value: 0x0
gpio_porta_eoi	0x4c	See Description	W	Port A clear interrupt register Width: GPIO_PWIDTH_A Reset Value: 0x0
gpio_ext_porta	0x50	See Description	R	Port A clear interrupt register Width: GPIO_PWIDTH_A Reset Value: 0x0
gpio_ext_portb	0x54	See Description	R	Port B external port register Width: GPIO_PWIDTH_B Reset Value: 0x0
gpio_ext_portc	0x58	See Description	R	Port C external port register Width: GPIO_PWIDTH_C Reset Value: 0x0
gpio_ext_portd	0x5c	See Description	R	Port D external port register Width: GPIO_PWIDTH_D Reset Value: 0x0
gpio_ls_sync	0x60	1 bit	R/W	Level-sensitive synchronization enable Register Reset Value: 0x0

gpio_id_code	0x64	See Description	R	ID code register Width: GPIO_ID_WIDTH Reset Value: GPIO_ID_NUM
gpio_int _bothedge	0x68	See Description	R/W	Interrupt both edge type Width: GPIO_PWIDTH_A Reset Value: 0x0
gpio_ver_id_code	0x6c	32 bits	R	Component Version register Reset Value: See the Releases table in the Release Notes
gpio_config_reg1	0x74	32 bits	R	Configuration Register 1 Reset Value: Reset value depends on configuration parameters.
gpio_config_reg2	0x70	32 bits	R	Configuration Register 2 Reset Value: Reset value depends on configuration parameters.

13.1.3 Register and Field Descriptions

The following sections contain the memory diagrams and field descriptions for the individual registers.

gpio_swporta_dr

- Name:Port A Data Register
- Size:GPIO_PWIDTH_A
- Address Offset:0x00
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Port A Data Register	R/W	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. Reset Value: GPIO_SWPORTA_RESET

gpio_swporta_ddr

- Name:Port A Data Direction Register
- Size:GPIO_PWIDTH_A
- Address Offset:0x04
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Port A Data Direction Register	R/W	Values written to this register independently control the direction of the corresponding data bit in Port A. The default direction can be configured as input or output after system reset through the GPIO_DFLT_DIR_A parameter. 0 – Input (default) 1 – Output Reset Value: GPIO_DFLT_DIR_A

gpio_swporta_ctl

- Name:Port A Data Source
- Size:1 bit wide if GPIO_PORTA_SINGLE_CTL = 1

  GPIO_PWIDTH_A bits wide if GPIO_PORTA_SINGLE_CTL = 0
- Address Offset:0x08
- Read/write access:read/write

Bits	Name	R/W	Description
0 -or- 0:GPIO_PWIDTH_A-1	Port A Data Source	R/W	The data and control source for a signal can come from either software or hardware; this bit selects between them.The default source is configurable through the GPIO_DFLT_SRC_A configuration parameter. 0 – Software mode (default) 1 – Hardware mode If GPIO_PORTA_SINGLE_CTL = 0, the register will contain one bit for each bit of the signal. Upon reset in this case,the value of GPIO_DFLT_SRC_A is replicated across all bits of the signal so that all bits power up with the same operating mode. Furthermore, the default source of each bit of the signal can subsequently be changed by writing to the corresponding bit of this register. This register is not available unless GPIO_HW_PORTA = 1. Reset Value: If GPIO_PORTA_SINGLE_CTL = 1, then the reset value is GPIO_DFLT_SRC_A. If GPIO_PORTA_SINGLE_CTL = 0, then the reset value is {GPIO_PWIDTH_A{GPIO_DFLT_SRC_A in each bit}}.

gpio_swportb_dr

- Name:Port B Data Register
- Size:GPIO_PWIDTH_B
- Address Offset:0x0c
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_B	Reserved, read as zero		
GPIO_PWIDTH_B–1:0	Port B Data Register	R/W	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. Reset Value: GPIO_SWPORTB_RESET

gpio_swportb_ddr

- Name:Port B Data Register
- Size:GPIO_PWIDTH_B
- Address Offset:0x10
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_B	Reserved, read as zero		
GPIO_PWIDTH_B–1:0	Port B Data Direction	R/W	Values written to this register independently control the direction of the corresponding data bit in Port B. The default direction can be configured as input or output after system reset through the GPIO_DFLT_DIR_B parameter. 0 – Input (default) 1 – Output Reset Value: GPIO_DFLT_DIR_B

gpio_swportb_ctl

- Name:Port B Data Register
- Size:1 bit wide if GPIO_PORTB_SINGLE_CTL = 1

  GPIO_PWIDTH_B bits wide if GPIO_PORTB_SINGLE_CTL = 0
- Address Offset:0x14
- Read/write access:read/write

Bits	Name	R/W	Description
0 -or- 0:GPIO_PWIDTH_B-1	Port B Data Source	R/W	The data and control source for a signal can come from either software or hardware; this bit selects between them.The default source is configurable through the GPIO_DFLT_SRC_B configuration parameter. 0 – Software mode (default) 1 – Hardware mode If GPIO_PORTA_SINGLE_CTL = 0, the register will contain one bit for each bit of the signal. Upon reset in this case,the value of GPIO_DFLT_SRC_B is replicated across all bits of the signal so that all bits power up with the same operating mode. Furthermore, the default source of each bit of the signal can subsequently be changed by writing to the corresponding bit of this register. This register is not available unless GPIO_HW_PORTB = 1. Reset Value: If GPIO_PORTB_SINGLE_CTL = 1, then the reset value is GPIO_DFLT_SRC_B. If GPIO_PORTB_SINGLE_CTL = 0, then the reset value is {GPIO_PWIDTH_B{GPIO_DFLT_SRC_B in each bit}}.

gpio_swportc_dr

- Name:Port C Data Register
- Size:GPIO_PWIDTH_C
- Address Offset:0x18
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_C	Reserved, read as zero		
GPIO_PWIDTH_C–1:0	Port C Data Register	R/W	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. Reset Value: GPIO_SWPORTC_RESET

gpio_swportc_ddr

- Name:Port C Data Direction
- Size:GPIO_PWIDTH_C
- Address Offset:0x1c
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_C	Reserved, read as zero		
GPIO_PWIDTH_C–1:0	Port C Data Direction	R/W	Values written to this register independently control the direction of the corresponding data bit in Port C. The default direction can be configured as input or output after system reset through the GPIO_DFLT_DIR_C parameter. 0 – Input (default) 1 – Output Reset Value: GPIO_DFLT_DIR_C

gpio_swportc_ctl

- Name:Port C Data Register
- Size:1 bit wide if GPIO_PORTC_SINGLE_CTL = 1

  GPIO_PWIDTH_C bits wide if GPIO_PORTC_SINGLE_CTL = 0
- Address Offset:0x20
- Read/write access:read/write

Bits	Name	R/W	Description
0 -or- 0:GPIO_PWIDTH_C-1	Port C Data Source	R/W	The data and control source for a signal can come from either software or hardware; this bit selects between them.The default source is configurable through the GPIO_DFLT_SRC_C configuration parameter. 0 – Software mode (default) 1 – Hardware mode If GPIO_PORTC_SINGLE_CTL = 0, the register will contain one bit for each bit of the signal. Upon reset in this case,the value of GPIO_DFLT_SRC_C is replicated across all bits of the signal so that all bits power up with the same operating mode. Furthermore, the default source of each bit of the signal can subsequently be changed by writing to the corresponding bit of this register. This register is not available unless GPIO_HW_PORTC = 1. Reset Value: If GPIO_PORTC_SINGLE_CTL = 1, then the reset value is GPIO_DFLT_SRC_C. If GPIO_PORTC_SINGLE_CTL = 0, then the reset value is {GPIO_PWIDTH_C{GPIO_DFLT_SRC_C in each bit}}.

gpio_swportd_dr

- Name:Port D Data Register
- Size:GPIO_PWIDTH_D
- Address Offset:0x24
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_D	Reserved, read as zero		
GPIO_PWIDTH_D–1:0	Port D Data Register	R/W	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. 0 – Input (default) 1 – Output Reset Value: GPIO_SWPORTD_RESET

gpio_swportd_ddr

- Name:Port D Data Direction
- Size:GPIO_PWIDTH_D
- Address Offset:0x24
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_D	Reserved, read as zero		
GPIO_PWIDTH_D–1:0	Port D Data Direction	R/W	Values written to this register independently control the direction of the corresponding data bit in Port D. The default direction can be configured as input or output after system reset through the GPIO_DFLT_DIR_D parameter. 0 – Input (default) 1 – Output Reset Value: GPIO_DFLT_DIR_D

gpio_swportd_ctl

- Name:Port D Data Source
- Size:1 bit wide if GPIO_PORTD_SINGLE_CTL = 1

  GPIO_PWIDTH_D bits wide if GPIO_PORTD_SINGLE_CTL = 0
- Address Offset:0x2c
- Read/write access:read/write

Bits	Name	R/W	Description
0 -or- 0:GPIO_PWIDTH_D-1	Port D Data Source	R/W	The data and control source for a signal can come from either software or hardware; this bit selects between them.The default source is configurable through the GPIO_DFLT_SRC_D configuration parameter. 0 – Software mode (default) 1 – Hardware mode If GPIO_PORTD_SINGLE_CTL = 0, the register will contain one bit for each bit of the signal. Upon reset in this case,the value of GPIO_DFLT_SRC_D is replicated across all bits of the signal so that all bits power up with the same operating mode. Furthermore, the default source of each bit of the signal can subsequently be changed by writing to the corresponding bit of this register. This register is not available unless GPIO_HW_PORTD = 1. Reset Value: If GPIO_PORTD_SINGLE_CTL = 1, then the reset value is GPIO_DFLT_SRC_D. If GPIO_PORTD_SINGLE_CTL = 0, then the reset value is {GPIO_PWIDTH_D{GPIO_DFLT_SRC_D in each bit}}.

gpio_inten

- Name:Interrupt enable
- Size:GPIO_PWIDTH_A

  This register is available only if Port A is configured to generate interrupts

  (GPIO_PORTA_INTR = Include (1))
- Address Offset:0x30

- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Interrupt enable	R/W	Allows each bit of Port A to be configured for interrupts. By default the generation of interrupts is disabled. Whenever a 1 is written to a bit of this register, it configures the corresponding bit on Port A to become an interrupt; otherwise, Port A operates as a normal GPIO signal. Interrupts are disabled on the corresponding bits of Port A if the corresponding data direction register is set to Output or if Port A mode is set to Hardware. 0 – Configure Port A bit as normal GPIO signal (default) 1 – Configure Port A bit as interrupt Reset Value: 0x0

gpio_intmask

- Name:Interrupt mask
- Size:GPIO_PWIDTH_A

  This register is available only if Port A is configured to generate interrupts

  (GPIO_PORTA_INTR = Include (1))
- Address Offset:0x34
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Interrupt mask	R/W	Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. By default, all interrupts bits are unmasked. Whenever a 1 is written to a bit in this register, it masks the interrupt generation capability for this signal; otherwise interrupts are allowed through. The unmasked status can be read as well as the resultant status after masking. 0 – Interrupt bits are unmasked (default) 1 – Mask interrupt Reset Value: 0x0

gpio_inttype_level

- Name:Interrupt level
- Size:GPIO_PWIDTH_A

  This register is available only if Port A is configured to generate interrupts

  (GPIO_PORTA_INTR = Include (1))
- Address Offset:0x38
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Interrupt level	R/W	Controls the type of interrupt that can occur on Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to be level-sensitive; otherwise, it is edge-sensitive. 0 – Level-sensitive (default) 1 – Edge-sensitive Reset Value: 0x0

gpio_int_polariy

- Name:Interrupt polarity
- Size:GPIO_PWIDTH_A

  This register is available only if Port A is configured to generate interrupts

  (GPIO_PORTA_INTR = Include (1))
- Address Offset:0x3c
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Interrupt polarity	R/W	Controls the polarity of edge or level sensitivity that can occur on input of Port A. Whenever a 0 is written to a bit of this register, it configures the interrupt type to falling-edge or active-low sensitive; otherwise, it is rising-edge or active-high sensitive. 0 – Active-low (default) 1 – Active-high Reset Value: 0x0

gpio_intstatus

- Name:Interrupt status
- Size:GPIO_PWIDTH_A

  This register is available only if Port A is configured to generate interrupts

  (GPIO_PORTA_INTR = Include (1))
- Address Offset:0x40
- Read/write access:read

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Interrupt status	R	Interrupt status of Port A Reset Value: 0x0

gpio_raw_intstatus

- Name:Raw interrupt status
- Size:GPIO_PWIDTH_A

  This register is available only if Port A is configured to generate interrupts

  (GPIO_PORTA_INTR = Include (1))
- Address Offset:0x44
- Read/write access:read

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Raw Interru -pt status	R	Raw interrupt of status of Port A (premasking bits) Reset Value: 0x0

gpio_debounce

- Name:Debounce enable
- Size:GPIO_PWIDTH_A

  This register is available only if Port A is configured to generate interrupts

  (GPIO_PORTA_INTR = Include (1)) and when the debounce logic is included

  (GPIO_DEBOUNCE = Include (1)).
- Address Offset:0x48
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Debounce enable	R/W	Controls whether an external signal that is the source of an interrupt needs to be debounced to remove any spurious glitches. Writing a 1 to a bit in this register enables the debouncing circuitry. A signal must be valid for two periods of an external clock before it is internally processed. 0 – No debounce (default) 1 – Enable debounce Reset Value: 0x0

gpio_porta_eoi

- Name:Clear interrupt
- Size:GPIO_PWIDTH_A

  This register is available only if Port A is configured to generate interrupts

  (GPIO_PORTA_INTR = Include (1)).
- Address Offset:0x4c
- Read/write access:write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Clear interrupt	W	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. 0 – No interrupt clear (default) 1 – Clear interrupt Reset Value: 0x0

gpio_ext_porta

- Name:External Port A
- Size:GPIO_PWIDTH_A
- Address Offset:0x50
- Read/write access:read

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	External Port A	R	This register always reflects the signals value on the External Port A Reset Value: 0x0

gpio_ext_portb

- Name:External Port B
- Size:GPIO_PWIDTH_B
- Address Offset:0x54
- Read/write access:read

Bits	Name	R/W	Description
31:GPIO_PWIDTH_B	Reserved, read as zero		
GPIO_PWIDTH_B–1:0	External Port B	R	This register always reflects the signals value on the External Port B Reset Value: 0x0

gpio_ext_portc

- Name:External Port C
- Size:GPIO_PWIDTH_C
- Address Offset:0x58
- Read/write access:read

Bits	Name	R/W	Description
31:GPIO_PWIDTH_C	Reserved, read as zero		
GPIO_PWIDTH_C–1:0	External Port C	R	This register always reflects the signals value on the External Port C Reset Value: 0x0

gpio_ext_portd

- Name:External Port D
- Size:GPIO_PWIDTH_D
- Address Offset:0x5c
- Read/write access:read

Bits	Name	R/W	Description
31:GPIO_PWIDTH_D	Reserved, read as zero		
GPIO_PWIDTH_D–1:0	External Port D	R	This register always reflects the signals value on the External Port D Reset Value: 0x0

gpio_ls_sync

- Name:Synchronization level
- Size:1 bit
- Address Offset:0x60
- Read/write access:

  read/write when Port A is configured to generate interrupts (GPIO_PORTA_INTR = 1)

  read-only when GPIO_PORTA_INTR = 0

Bits	Name	R/W	Description
0	Synchroniza -tion level	R/W	Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr. 0 – No synchronization to pclk_intr (default) 1 – Synchronize to pclk_intr Reset Value: 0x0

gpio_id_code

- Name:GPIO ID code
- Size:GPIO_ID_WIDTH
- Address Offset:0x64
- Read/write access:read

Bits	Name	R/W	Description
31:GPIO_ID_WIDTH	Reserved, read as zero		
GPIO_ID_WIDTH–1:0	GPIO ID code	R	This is a user-specified code that a system can read. It can be used for chip identification, and so on. ** Reset Value: ** GPIO_ID_NUM

gpio_int_bothedge

- Name:Interrupt both edge type
- Size:GPIO_PWIDTH_A

  This register is available only if PORT A is configured to generate interrupts (GPIO_PORTA_INTR = Include(1)) and interrupt detection is configured to generate on both rising and falling edges of external input signal (GPIO_INT_BOTH_EDGE = Include(1)).
- Address Offset:0x68
- Read/write access:read/write

Bits	Name	R/W	Description
31:GPIO_PWIDTH_A	Reserved, read as zero		
GPIO_PWIDTH_A–1:0	Interrupt both edge type	R/W	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 registers 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 registers. 0 – Active-low (default) 1 – Active-high Reset Value: 0x0

gpio_ver_id_code

- Name:GPIO Component Version
- Size:32 bits
- Address Offset:0x6c
- Read/write access:read

Bits	Name	R/W	Description
31:0	GPIO Component Version	R	ASCII value for each number in the version, followed by*.For example 32_30_31_2A represents the version 2.01* Reset Value: See the releases table in the Release Notes

gpio_config_reg1

- Name:GPIO Configuration Register 1
- Size:32 bits
- Address Offset:0x74
- Read/write access:read

This register is present when the configuration parameter GPIO_ADD_ENCODED_PARAMS is set to True. If this parameter is set to False, this register reads back zero (0).

Bits	Name	R/W	Description
31:22	Reserved	R	Reserved
21	INTERRUPT_BOTH_EDGE_TYPE	R	The value of this register is derived from the GPIO_INT_BOTH_EDGE configuration parameter. 0 = Exclude 1 = Include
20:16	ENCODED_ID_WIDTH	R	The value of this register is equal to GPIO_ID_WIDTH-1.
15	GPIO_ID	R	The value of this register is derived from the GPIO_ID configuration parameter. 0 = Exclude 1 = Include
14	ADD_ENCODED_PARAMS	R	The value of this register is derived from the GPIO_ADD_ENCODED_PARAMS configuration parameter. 0 = False 1 = True
13	DEBOUNCE	R	The value of this register is derived from the GPIO_DEBOUNCE configuration parameter. 0 = Exclude 1 = Include
12	PORTA_INTR	R	The value of this register is derived from the GPIO_PORTA_INTR configuration parameter. 0 = Exclude 1 = Include
11	HW_PORTD	R	The value of this register is derived from the GPIO_HW_PORTD configuration parameter. 0 = Exclude 1 = Include

10	HW_PORTC	R	The value of this register is derived from the GPIO_HW_PORTC configuration parameter. 0 = Exclude 1 = Include
9	HW_PORTB	R	The value of this register is derived from the GPIO_HW_PORTB configuration parameter. 0 = Exclude 1 = Include
8	HW_PORTA	R	The value of this register is derived from the GPIO_HW_PORTA configuration parameter. 0 = Exclude 1 = Include
7	PORTD_SINGLE_CTL	R	The value of this register is derived from the GPIO_PORTD_SINGLE_CTL configuration parameter. 0 = False 1 = True
6	PORTC_SINGLE_CTL	R	The value of this register is derived from the GPIO_PORTC_SINGLE_CTL configuration parameter. 0 = False 1 = True
5	PORTB_SINGLE_CTL	R	The value of this register is derived from the GPIO_PORTB_SINGLE_CTL configuration parameter. 0 = False 1 = True
4	PORTA_SINGLE_CTL	R	The value of this register is derived from the GPIO_PORTA_SINGLE_CTL configuration parameter. 0 = False 1 = True
3:2	NUM_PORTS	R	The value of this register is derived from the GPIO_NUM_PORT configuration parameter. 0x0 = 1 0x1 = 2 0x2 = 3 0x3 = 4
1:0	APB_DATA_WIDTH	R	The value of this register is derived from the GPIO_APB_DATA_WIDTH configuration parameter. 0x0 = 8 bits 0x1 = 16 bits 0x2 = 32 bits 0x3 = Reserved

gpio_config_reg2

- Name:GPIO Configuration Register 2
- Size:32 bits
- Address Offset:0x70
- Read/write access:read

This register is a read-only register that is present when the configuration parameter

GPIO_ADD_ENCODED_PARAMS is set to True. If this configuration is set to False, then this register reads back 0.

Bits	Name	R/W	Description
31:20	Reserved	R	Reserved
19:15	ENCODED_ID_PWIDTH_D	R	The value of this register is equal to GPIO_PWIDTH_D-1.
14:10	ENCODED_ID_PWIDTH_C	R	The value of this register is equal to GPIO_PWIDTH_C-1.
9:5	ENCODED_ID_PWIDTH_B	R	The value of this register is equal to GPIO_PWIDTH_B-1.
4:0	ENCODED_ID_PWIDTH_A	R	The value of this register is equal to GPIO_PWIDTH_A-1.

CHAPTER

FOURTEEN

UART

14.1 Registers

This section describes the programmable registers of the UART.

Note:Since UART registers are only located 32-bit boundaries, paddr[1:0] may be tied low permanently, if so desired. This would allow backward compatibility with standard 16550 UART programmability.

14.1.1 1 Register Memory Map

Table -1 summarizes the register memory map for the UART:

Table 1: UART Memory Map

Name	Address Offset	Width	R/W	Description
RBR	0x00	32 bits	R	Receive Buffer Register. Reset Value: 0x0 Dependencies: LCR[7] bit = 0
THR		32 bits	W	Transmit Holding Register Reset Value: 0x0 Dependencies: LCR[7] bit = 0
DLL		32 bits	R/W	Divisor Latch (Low) Reset Value: 0x0 Dependencies: LCR[7] bit = 1
DLH	0x04	32 bits	R/W	Divisor Latch (High) Reset Value: 0x0 Dependencies: LCR[7] bit = 1
IER		32 bits	R/W	Interrupt Enable Register Reset Value: 0x0 Dependencies: LCR[7] bit = 0
IIR	0x08	32 bits	R	Interrupt Identification Register Reset Value: 0x01
FCR	0x08	32 bits	W	FIFO Control Register Reset Value: 0x0
LCR	0x0C	32 bits	R/W	Line Control Register Reset Value: 0x0
MCR	0x10	32 bits	R/W	Modem Control Register Reset Value: 0x0
LSR	0x14	32 bits	R	Line Status Register Reset Value: 0x60

SRBR	0x30 - 0x6C	32 bits	R	Shadow Receive Buffer Register Reset Value: 0x0 Dependencies: LCR[7] bit = 0
STHR	0x70	32 bits	W	Shadow Transmit Holding Register Reset Value: 0x0 Dependencies: LCR[7] bit = 0
FAR	0x74	32 bits	R/W	FIFO Access Register Reset Value: 0x0
TFR	0x78	32 bits	R	Transmit FIFO Read Reset Value: 0x0
RFW	0x7C	32 bits	W	Receive FIFO Write Reset Value: 0x0
USR	0x7C	32 bits	R	UART Status Register Reset Value: 0x6

TFL	0x80	see 2.21	R	Transmit FIFO Level Reset Value: 0x0
RFL	0x84	see 2.22	R	Receive FIFO Level Reset Value: 0x0
SRR	0x88	32 bits	W	Software Reset Register Reset Value: 0x0
SRTS	0x8C	32 bits	R/W	Shadow Request to Send Reset Value: 0x0
SBCR	0x90	32 bits	R/W	Shadow Break Control Register Reset Value: 0x0
SDMAM	0x94	32 bits	R/W	Shadow DMA Mode Reset Value: 0x0
SFE	0x98	32 bits	R/W	Shadow FIFO Enable Reset Value: 0x0

SRT	0x9C	32 bits	R/W	Shadow RCVR Trigger Reset Value: 0x0
STET	0xA0	32 bits	R/W	Shadow TX Empty Trigger Reset Value: 0x0
HTX	0xA4	32 bits	R/W	Halt TX Reset Value: 0x0
DMASA	0xA8	1 bit	W	DMA Software Acknowledge Reset Value: 0x0
TCR	0xAC	32 bits	R/W	Transceiver Control Register Reset Value: 0x6 Dependencies: UART_RS485_INTERFACE_EN=1

DE_EN	0xB0	32 bits	R/W	Driver Output Enable Register Reset Value: 0x0 Dependencies: UART_RS485_INTERFACE_EN=1
RE_EN	0xB4	32 bits	R/W	Receiver Output Enable Register Reset Value: 0x0 Dependencies: UART_RS485_INTERFACE_EN=1
DET	0xB8	32 bits	R/W	Driver Output Enable Timing Register Reset Value: 0x0 Dependencies: UART_RS485_INTERFACE_EN=1
TAT	0xBC	32 bits	R/W	TurnAround Timing Register. Reset Value: 0x0 Dependencies: UART_RS485_INTERFACE_EN=1

DLF	0xC0	32 bits	R/W	Divisor Latch Fractional Value. Reset Value: 0x0 Dependencies: FRACTIONAL_BAUD_DIVISOR_EN=1
RAR	0xC4	32 bits	R/W	Receive Address Register Reset Value: 0x0 Dependencies: *UART* 9BIT_DATA_EN=1
TAR	0xC8	32 bits	R/W	Transmit Address Register Reset Value: 0x0 Dependencies: UART_9BIT_DATA_EN=1
LCR_EXT	0xCC	32 bits	R/W	Line Extended Control Register Reset Value: 0x0 Dependencies: UART_9BIT_DATA_EN=1

——	0xD0 - 0xF0	——	——	——
CPR	0xF4	32 bits	R	Component Parameter Register Reset Value: Configuration-dependent
UCV	0xF8	32 bits	R	UART Component Version Reset Value: See the Releases table in the AMBA 2 release notes.
CTR	0xFC	32 bits	R	Component Type Register Reset Value: 0x44570110

14.1.2 2 Register and Field Descriptions

The following subsections describe the data fields of the UART registers.

2.1 RBR

- 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.

Table 2: RBR Register Fields

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

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.

Table 3: THR Register Fields

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

2.3 DLH

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

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.

Table 4: DLH Register Fields

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 UART clock should be allowed to pass before transmitting or receiving data. Reset Value: 0x0

2.4 DLL

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

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.

Table 5: DLL Register Fields

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 UART clock should be allowed to pass before transmitting or receiving data. Reset Value: 0x0

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.

Table 6: IER Register Fields

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 Transmitter 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

2.6 IIR

- Name: Interrupt Identity Register
- Size: 32 bits
- Address Offset: 0x08
- Read/write access: read-only

Table 7: IIR Register Fields

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 The interrupt priorities are split into several levels that are detailed in Table -2 . **Note** : An interrupt of type 0111 (busy detect) is never indicated if UART_16550_COMPATIBLE = YES in coreConsultant. 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

Table -2 summarizes the Interrupt Control Functions:

Table 8: 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
0	0	1	0	Third	Transmit holding register empty	Transmitter holding register empty (Prog. THRE Mode disabled) or XMIT FIFO above 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 auto 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 the Master has tried to write to the Line Control Register while the UART is busy (USR[0] is set to 1).	Reading the UART status register

2.7 FCR

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

This register is valid only when the UART is configured to have FIFOs implemented (FIFO_MODE != NONE). If FIFOs are not implemented, this register does not exist and writing to this register address has no effect.

Table 9: FCR Register Fields

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 thre_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

3	DMA Mode (or DMAM)	W	DMA Mode. This determines the DMA signalling 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 XIFOR)	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 RIFOR)	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

2.8 LCR

- Name: Line Control Register
- Size: 32 bits
- Address Offset: 0x0C
- Read/write access: read/write

Table 10: LCR Register Fields

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 writable 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 writable 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 logic 1s is transmitted or checked. If set to 0, an odd number of logic 1s is transmitted or checked. Reset Value: 0x0
3	PEN	R/W	Parity Enable. If UART_16550_COMPATIBLE = NO, then writable 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 writable 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 bits **NOTE**: The STOP bit duration implemented by 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
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, always readable. When DLS_E is 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

2.9 MCR

- Name: Modem Control Register
- Size: 32 bits
- Address Offset: 0x10
- Read/write access: read/write

Table 11: MCR Register Fields

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 **Note**: To enable SIR mode, write the appropriate value to the MCR register before writing to the LCR register.
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, Auto Flow Control features are enabled. 0 - Auto Flow Control Mode disabled 1 - Auto Flow Control Mode enabled Reset Value: 0x0
4	Loop-Back (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

2.10 LSR

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

Table 12: LSR Register Fields

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
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 FIFOs 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 list. 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 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

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

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.

Table 13: MSR Register Fields

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 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 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
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

2.12 SCR

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

Table 14: SCR Register Fields

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 UART. Reset Value: 0x0

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 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.

Table 15: LPDLL Register Fields

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 UART clock should be allowed to pass before transmitting or receiving data. Reset Value: 0x0

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 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.

Table 16: LPDLH Register Fields

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 UART clock should be allowed to pass before transmitting or receiving data. Reset Value: 0x0

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 UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers 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.

Table 17: SRBR Register Fields

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

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 UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers 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.

Table 18: STHR Register Fields

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

2.17 FAR

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

Table 19: FAR Register Fields

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

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 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.

Table 20: TFR Register Fields

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

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 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.

Table 21: RFW Register Fields

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

2.20 USR

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

Table 22: USR Register Fields

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 is 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
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 UART is idle or inactive. 0 - UART is idle or inactive 1 - 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 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 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

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 UART is configured to have additional FIFO status registers implemented (FIFO_STAT = YES). If status registers are not implemented, this register does not exist and reading from this register address returns 0.

Table 23: TFL Register Fields

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

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 UART is configured to have additional FIFO status registers implemented (FIFO_STAT = YES). If status registers are not implemented, this register does not exist and reading from this register address returns 0.

Table 24: RFL Register Fields

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

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 UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers are not implemented, this register does not exist and reading from this register address returns 0.

Table 25: SRR Register Fields

Bits	Nam	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 UART and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset. Reset Value: 0x0

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 UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers are not implemented, this register does not exist and reading from this register address returns 0.

Table 26: SRTS Register Fields

Bits	Nam	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 UART is ready to exchange data. When Auto RTS Flow Control is off (MCR[5] = 0), set rts_n low by setting MCR[1] (RTS) 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

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 UART is configured to have additional shadow registers implemented (SHADOW = YES). If shadow registers are not implemented, this register does not exist and reading from this register address returns 0.

Table 27: SBCR Register Fields

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.

2.26 SDMAM

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

This register is valid only when the UART is configured to have additional FIFO registers implemented (FIFO_MODE != None) and additional shadow registers implemented (SHADOW = YES). If these registers are not implemented, this register does not exist and reading from this register address returns 0.

Table 28: SDMAM Register Fields

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 signalling mode for 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

2.27 SFE

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

This register is valid only when the UART is configured to have additional FIFO registers implemented (FIFO_MODE != None) and additional shadow registers implemented (SHADOW = YES). If these registers are not implemented, this register does not exist and reading from this register address returns 0.

Table 29: SFE Register Fields

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. 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

2.28 SRT

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

This register is valid only when the UART is configured to have additional FIFO registers implemented (FIFO_MODE != None) and additional shadow registers implemented (SHADOW = YES). If these registers are not implemented, this register does not exist and reading from this register address returns 0.

Table 30: SRT Register Fields

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

2.29 STET

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

This register is valid only when the UART is configured to have FIFOs implemented (FIFO_MODE != NONE) and THRE interrupt support implemented (THRE_MODE_USER = Enabled) and additional shadow registers implemented (SHADOW = YES). If FIFOs are not implemented or THRE interrupt support is not implemented or shadow registers are not implemented, this register does not exist and reading from this register address returns 0.

Table 31: STET Register Fields

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

2.30 HTX

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

Table 32: HTX Register Fields

Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	Halt TX	R/W	This register is used 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 Dependencies: Writes have no effect when FIFO_MODE = None.

2.31 DMASA

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

Table 33: DMASA Register Fields

Bits	Name	R/W	Description
31:1	Reserved and read as 0		
0	DMA Software Acknowledge	W	This register is used 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 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.

2.32 TCR

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

This register is used to enable or disable RS485 mode and also control the polarity values for Driven enable (de) and Receiver Enable (re) signals.

This register is only valid when the UART is configured to have RS485 interface implemented (UART_RS485_INTERFACE_EN = ENABLED). If RS485 interface is not implemented, this register does not exist and reading from this register address returns zero.

Table 34: TCR Register Fields

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 UART controller will be ready to receive. If the user programs the TX FIFO with data, then 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
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

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.

This register is only valid when the UART is configured to have RS485 interface implemented (UART_RS485_INTERFACE_EN = ENABLED). If RS485 interface is not implemented, this register does not exist and reading from this register address will return zero.

Table 35: DE_EN Register Fields

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

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.

This register is only valid when the UART is configured to have RS485 interface implemented (UART_RS485_INTERFACE_EN = ENABLED). If the RS485 interface is not implemented, this register does not exist and reading from this register address will return zero.

Table 36: RE_EN Register Fields

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

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 're' signal.

This register is only valid when the UART is configured to have RS485 interface implemented (UART_RS485_INTERFACE = ENABLED). If RS485 interface is not implemented, this register does not exist and reading from this register address will return zero.

Table 37: DET Register Fields

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

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.

This register is only valid when the UART is configured to have the RS485 interface implemented (UART_RS485_INTERFACE_EN = ENABLED). If RS485 interface is not implemented, this register does not exist and reading from this register address will return zero.

Table 38: TAT Register Fields

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

2.37 DLF

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

This register is only valid when the UART is configured to have Fractional Baud rate Divisor implemented (FRACTIONAL_BAUD_DIVISOR_EN = ENABLED). If Fractional Baud rate divisor is not implemented, this register does not exist and reading from this register address will return zero.

Table 39: DLF Register Fields

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). Table -3 describes the DLF Values to be programmed for DLF_SIZE=4. Reset Value: 0x0

Table -3 summarizes the Divisor Latch Fractional Values:

Table 40: Divisor Latch Fractional Values

DLF Value	Fraction	Fractional Value
0000	0/16	0.0000
0001	1/16	0.0625
0010	2/16	0.1250
0011	3/16	0.1875
0100	4/16	0.2500
0101	5/16	0.3125
0110	6/16	0.3750
0111	7/16	0.4375
1000	8/16	0.5000
1001	9/16	0.5625
1010	10/16	0.6250
1011	11/16	0.6875
1100	12/16	0.7500
1101	13/16	0.8125
1110	14/16	0.8750
1111	15/16	0.9375

2.38 RAR

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

Table 41: RAR Register Fields

Bits	Nam	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

2.39 TAR

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

Table 42: TAR Register Fields

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 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 UART serial lane, then 'SEND_ADDR' bit will be auto-cleared by the hardware. Reset Value: 0x0

2.40 LCR_EXT

- Name: Line Extended Control Register (LCR_EXT)
- Size: 32 bits
- Address Offset: 0xCC
- Read/write access: read/write

Table 43: LCR_EXT Register Fields

Bits	Name	R/W	Description
31:4	Reserved and read as 0		
3	TRANS-MIT_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 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 9th 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 9th 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

1	ADDR_MATCH	R/W	Address Match Mode. This bit is used to enable the address match feature during receive. 1 - Address match mode; 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 UART starts receiving data. 0 - Normal mode; UART will start to receive the data and 9-bit character will be formed and written into the receive Rx FIFO. 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. Reset Value: 0x0
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

2.41 CPR

- Name: Component Parameter Register
- Size: 32 bits
- Address Offset: 0xF4
- Read/write access: read-only

This register is valid only when UART_ADD_ENCODED_PARAMS = 1. If the UART_ADD_ENCODED_PARAMS parameter is not set, this register does not exist and reading from this register address returns 0.

Table 44: CPR Register Fields

Bits	Name	R/W	Description
31:24	Reserved and read as 0		
23:16	FIFO_MODE	R	0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81-0xff = reserved
15:14	Reserved and read as 0		
13	DMA_EXTRA	R	0 - FALSE 1 - TRUE
12	UART_ADD_ENCODED_PARAM	R	0 - FALSE 1 - TRUE
11	SHADOW	R	0 - FALSE 1 - TRUE
10	FIFO_STAT	R	0 - FALSE 1 - TRUE
9	FIFO_ACCESS	R	0 - FALSE 1 - TRUE
8	ADDITIONAL_FEAT	R	0 - FALSE 1 - TRUE
7	SIR_LP_MODE	R	0 - FALSE 1 - TRUE
6	SIR_MODE	R	0 - FALSE 1 - TRUE
5	THRE_MODE	R	0 - FALSE 1 - TRUE
4	AFCE_MODE	R	0 - FALSE 1 - TRUE
3:2	Reserved and read as 0		
1:0	APB_DATA_WIDTH	R	00 - 8 bits 01 - 16 bits 10 - 32 bits 11 - reserved

2.42 UCV

- Name: UART Component Version
- Size: 32 bits
- Address Offset: 0xF8
- Read/write access: read-only

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

Table 45: UCV Register Fields

Bits	Name	R/W	Description
31:0	UART Component Version	R	ASCII value for each number in the version, followed by*. For example 32_30_31_2A represents the version 2.01* Reset Value: See the releases table in the AMBA 2 release notes.

2.43 CTR

- Name: Component Type Register
- Size: 32 bits
- Address Offset: 0xFC
- Read/write access: read-only

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

Table 46: CTR Register Fields

Bits	Name	R/W	Description
31:0	Peripheral ID	R	This register contains the peripherals identification code. Reset Value: 0x44570110

CHAPTER

FIFTEEN

SPI

15.1 Registers

This chapter details all possible registers in the controller. They are arranged hierarchically into maps and blocks (banks). For configurable IP titles, your actual configuration might not contain all of these registers.

Attention: For configurable IP titles, do not use this document to determine the exact attributes of your register map. It is for reference purposes only.

15.1.1 1 Memory Access Attributes

The Memory Access attribute is defined as <ReadBehavior>/<WriteBehavior> which are defined in the following table.

Table 1: Table 1-1 : Possible Read and Write Behaviors

Read (or Write) Behavior	Description
RC	A read clears this register field.
RS	A read sets this register field.
RM	A read modifies the contents of this register field.
Wo	You can only write to this register once field.
W1C	A write of 1 clears this register field.
W1S	A write of 1 sets this register field.
W1T	A write of 1 toggles this register field.
W0C	A write of 0 clears this register field.
W0S	A write of 0 sets this register field.
W0T	A write of 0 toggles this register field.
WC	Any write clears this register field.
WS	Any write sets this register field.
WM	Any write toggles this register field.
no Read Behavior attribute	You cannot read this register. It is Write-Only.
no Write Behavior attribute	You cannot write to this register. It is Read-Only.

Table 2: Table 1-2 : Memory Access Examples

Memory Access	Description
R	Read-only register field.
W	Write-only register field.
R/W	Read/write register field.
R/W1C	You can read this register field. Writing 1 clears it.
RC/W1C	Reading this register field clears it. Writing 1 clears it.
R/Wo	You can read this register field. You can only write to it once.

Table 3: Table 1-3 : Optional Attributes

Attribute	Description
Volatile	As defined by the IP-XACT specification. If true, indicates in the case of a write followed by read, or in the case of two consecutive reads, there is no guarantee as to what is returned by the read on the second transaction or that this return value is consistent with the write or read of the first transaction. The element implies there is some additional mechanism by which this field can acquire new values other than by reads/writes/resets and other access methods known to IP-XACT. For example, when the core updates the register field contents.
Testable	As defined by the IP-XACT specification. Possible values are unconstrained, untestable, readOnly, writeAsRead, restore. Untestable means that this field is untestable by a simple automated register test. For example, the read-write access of the register is controlled by a pin or another register. readOnly means that you should not write to this register; only read from it. This might apply for a register that modifies the contents of another register.
Reset Mask	As defined by the IP-XACT specification. Indicates that this register field has an unknown reset value. For example, the reset value is set by another register or an input pin; or the register is implemented using RAM.
*Varies	Indicates that the memory access (or reset) attribute (read, write behavior) is not fixed. For example, the read-write access of the register is controlled by a pin or another register. Or when the access depends on some configuration parameter; in this case the post-configuration report in coreConsultant gives the actual access value.

15.1.2 2 Detailed description of the register

2.1 CTRLR0

- Name: Control Register 0
- Description: This register controls the serial data transfer. It is impossible to write to this register when the SSI is enabled. The SSI is enabled and disabled by writing to the SSIENR register. Reset Value: SSI_CTRLR0_RST
- Size: 32 bits
- Offset: 0x0
- Exists: Always

Table 4: CTRLR0 Register Fields

Bits	Name	Memory Access	Description
31:25	RSVD_CTRLR0	R	SSTE Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
24	SSTE	*Varies	Slave Select Toggle Enable. When 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. If this register field is set to 1 the ss_*_n line will toggle between consecutive data frames, with the serial clock (sclk) being held to its default value while ss_*_n is high; if this register field is set to 0 the ss_*_n willstay low and sclk will run continuously for the duration of thetransfer. Note: This register is only valid when SSI_SCPH0_SSTOGGLE is set to 1. Value After Reset: “(SSI_SCPH0_SSTOGGLE==0) ? \”0\”: \”1\”” Exists: Always Memory Access: “(SSI_SCPH0_SSTOGGLE==0) ? \”read-only\”: \”read-write\””
23	RSVD_CTRLR0_23	R	CTRLR0_23 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
22:21	SPI_FRF	*Varies	SPI Frame Format: Selects data frame format for Transmitting/Receiving the data Bits only valid when SSI_SPI_MODE is either set to “Dual” or “Quad” or “Octal” mode. When SSI_SPI_MODE is configured for “Dual Mode”, 10/11 combination is reserved. When SSI_SPI_MODE is configured for “Quad Mode”, 11 combination is reserved. Values: 0x0 (STD_SPI_FRF): Standard SPI Frame Format 0x1 (DUAL_SPI_FRF): Dual SPI Frame Format 0x2 (QUAD_SPI_FRF): Quad SPI Frame Format 0x3 (OCTAL_SPI_FRF): Octal SPI Frame Format Value After Reset: 0x0 Exists: Always Memory Access: “(SSI_SPI_MODE==0) ? \”read-only\”: \”read-write\””

20:16	DFS_32	*Varies	Data Frame Size in 32-bit transfer size mode. Used to select the data frame size in 32-bit transfer mode. These bits areonly valid when SSI_MAX_XFER_SIZE is configured to 32. When the data frame size is programmed to be less than 32 bits, the receive data are automatically right-justified by the receive logic, with the upper bits of the receive FIFO zero-padded. You are responsible for making sure that transmit data is right-justified before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data. Note: When SSI_SPI_MODE is either set to “Dual” or “Quad” or “Octal” mode and SPI_FRF is not set to 2’b00. DFS value should be multiple of 2 if SPI_FRF = 0x01, DFS value should be multiple of 4 if SPI_FRF = 0x10, DFS value should be multiple of 8 if SPI_FRF = 0x11. Values: 0x3 (FRAME_04BITS): 4-bit serial data transfer 0x4 (FRAME_05BITS): 5-bit serial data transfer 0x5 (FRAME_06BITS): 6-bit serial data transfer 0x6 (FRAME_07BITS): 7-bit serial data transfer 0x7 (FRAME_08BITS): 8-bit serial data transfer 0x8 (FRAME_09BITS): 9-bit serial data transfer 0x9 (FRAME_10BITS): 10-bit serial data transfer 0xa (FRAME_11BITS): 11-bit serial data transfer 0xb (FRAME_12BITS): 12-bit serial data transfer 0xc (FRAME_13BITS): 13-bit serial data transfer 0xd (FRAME_14BITS): 14-bit serial data transfer 0xe (FRAME_15BITS): 15-bit serial data transfer 0xf (FRAME_16BITS): 16-bit serial data transfer 0x10 (FRAME_17BITS): 17-bit serial data transfer 0x11 (FRAME_18BITS): 18-bit serial data transfer 0x12 (FRAME_19BITS): 19-bit serial data transfer 0x13 (FRAME_20BITS): 20-bit serial data transfer 0x14 (FRAME_21BITS): 21-bit serial data transfer 0x15 (FRAME_22BITS): 22-bit serial data transfer 0x16 (FRAME_23BITS): 23-bit serial data transfer 0x17 (FRAME_24BITS): 24-bit serial data transfer 0x18 (FRAME_25BITS): 25-bit serial data transfer 0x19 (FRAME_26BITS): 26-bit serial data transfer 0x1a (FRAME_27BITS): 27-bit serial data transfer 0x1b (FRAME_28BITS): 28-bit serial data transfer 0x1c (FRAME_29BITS): 29-bit serial data transfer 0x1d (FRAME_30BITS): 30-bit serial data transfer 0x1e (FRAME_31BITS): 31-bit serial data transfer

			0x1f (FRAME_32BITS): 32-bit serial data transfer Value After Reset: “(SSI_MAX_XFER_SIZE ==32) ? \”0x7\” : \”0x0\”” Exists: Always Memory Access: “(SSI_MAX_XFER_SIZE ==32) ? \”read-write\” : \”read-only\””
15:12	CFS	R/W	Control Frame Size. Selects the length of the control word for the Microwire frame format. Values: 0x0 (SIZE_01_BIT): 1-bit Control Word 0x1 (SIZE_02_BIT): 2-bit Control Word 0x2 (SIZE_03_BIT): 3-bit Control Word 0x3 (SIZE_04_BIT): 4-bit Control Word 0x4 (SIZE_05_BIT): 5-bit Control Word 0x5 (SIZE_06_BIT): 6-bit Control Word 0x6 (SIZE_07_BIT): 7-bit Control Word 0x7 (SIZE_08_BIT): 8-bit Control Word 0x8 (SIZE_09_BIT): 9-bit Control Word 0x9 (SIZE_10_BIT): 10-bit Control Word 0xa (SIZE_11_BIT): 11-bit Control Word 0xb (SIZE_12_BIT): 12-bit Control Word 0xc (SIZE_13_BIT): 13-bit Control Word 0xd (SIZE_14_BIT): 14-bit Control Word 0xe (SIZE_15_BIT): 15-bit Control Word
			0xf (SIZE_16_BIT): 16-bit Control Word Value After Reset: 0x0 Exists: Always
11	SRL	R/W	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. 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 Values: 0x1 (TESTING_MODE): Test mode: Tx & Rx shift reg connected 0x0 (NORMAL_MODE): Normal mode operation Value After Reset: 0x0 Exists: Always

10	SLV_OE	R/W	Slave 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. Values: 0x1 (DISABLED): Slave Output is disabled 0x0 (ENABLED): Slave Output is enabled Value After Reset: 0x0 Exists: SSI_IS_MASTER == 0

9:8	TMOD	R/W	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 Idevice 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 master device. 00 - Transmit & Receive 01 - Transmit Only

			10 - Receive Only 11 - EEPROM Read When SSI_SPI_MODE is either set to “Dual” or “Quad” or “Octal” mode and SPI_FRF is not set to 2’b00. There are only two valid combinations: 10 - Read 01 - Write Values: 0x0 (TX_AND_RX): Transmit & receive 0x1 (TX_ONLY): Transmit only mode or Write (SPI_FRF != 2’b00) 0x2 (RX_ONLY): Receive only mode or Read (SPI_FRF != 2’b00) 0x3 (EEPROM_READ): EEPROM Read mode Value After Reset: 0x0 Exists: Always

7	SCPOL	*Varies	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. Values: 0x0 (SCLK_LOW): Inactive state of serial clock is low 0x1 (SCLK_HIGH): Inactive state of serial clock is high Value After Reset: SSI_DFLT_SCPOL Exists: Always Memory Access: “(SSI_HC_FRF==0) ? \””Read-write\”” : \’Read-only\””

6	SCPH	*Varies	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. Values: 0x0 (SCPH_MIDDLE): Serial clock toggles in middle of first data bit 0x1 (SCPH_START): Serial clock toggles at start of first data bit Value After Reset: SSI_DFLT_SCPH Exists: Always Memory Access: “(SSI_HC_FRF==0) ? \”Read-write\” : \”Read-only\””

5:4	FRF	*Varies	Frame Format. Selects which serial protocol transfers the data. Values: 0x0 (MOTOROLA_SPI): Motorola SPI Frame Format 0x1 (TEXAS_SSP): Texas Instruments SSP Frame Format 0x2 (NS_MICROWIRE): National Microwire Frame Format 0x3 (RESERVED): Reserved value Value After Reset: SSI_DFLT_FRF Exists: Always Memory Access: “(SSI_HC_FRF==0) ? \”Read-write\” : \”Read-only\””

3:0	DFS	*Varies	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 either set to “Dual” or “Quad” or “Octal” mode and SPI_FRF is not set to 2’b00. DFS value should be multiple of 2 if SPI_FRF = 01, DFS value should be multiple of 4 if SPI_FRF = 10, DFS value should be multiple of 8 if SPI_FRF = 11. Values: 0x3 (FRAME_04BITS): 4-bit serial data transfer 0x4 (FRAME_05BITS): 5-bit serial data transfer
			0x5 (FRAME_06BITS): 6-bit serial data transfer 0x6 (FRAME_07BITS): 7-bit serial data transfer 0x7 (FRAME_08BITS): 8-bit serial data transfer 0x8 (FRAME_09BITS): 9-bit serial data transfer 0x9 (FRAME_10BITS): 10-bit serial data transfer 0xa (FRAME_11BITS): 11-bit serial data transfer 0xb (FRAME_12BITS): 12-bit serial data transfer 0xc (FRAME_13BITS): 13-bit serial data transfer 0xd (FRAME_14BITS): 14-bit serial data transfer 0xe (FRAME_15BITS): 15-bit serial data transfer 0xf (FRAME_16BITS): 16-bit serial data transfer Value After Reset: “(SSI_MAX_XFER_SIZE ==16) ? \”0x7\” : \”0x0\”” Exists: Always Memory Access: “(SSI_MAX_XFER_SIZE ==16) ? \”read-write\” : \”read-only\””

2.2 CTRLR1

- Name: Control Register 1
- Description: This register exists only when the SSI is configured as a master device. When the SSI is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. Control register 1 controls the end of serial transfers when in receive-only mode. It is impossible to write to this register when the SSI is enabled. The SSI is enabled and disabled by writing to the SSIENR register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x4
- Exists: SSI_IS_MASTER == 1

Table 5: CTRLR1 Register Fields

Bits	Name	Memory Access	Description
31:16	RSVD_CTRLR1	R	CTRLR1 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
15:0	NDF	R/W	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 DW_apb_ssi is configured as a a serial slave. Value After Reset: 0x0 Exists: Always

2.3 SSIENR

- Name: SSI Enable Register
- Description: This register enables and disables the SSI.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x8
- Exists: Always

Table 6: SSIENR Register Fields

Bits	Name	Memory Access	Description
31:1	RSVD_SSIENR	R	SSIENR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
0	SSI_EN	R/W	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 registers 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. Values: 0x0 (DISABLE): Disables Serial Transfer 0x1 (ENABLED): Enables Serial Transfer Value After Reset: 0x0 Exists: Always

2.4 MWCR

- Name: Microwire Control Register
- Description: This register controls the direction of the data word for the half-duplex Microwire serial protocol. It is impossible to write to this register when the SSI is enabled. The SSI is enabled and disabled by writing to the SSIENR register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0xc
- Exists: Always

Table 7: MWCR Register Fields

Bits	Name	Memory Access	Description
31:3	RSVD_MWCR	R	MWCR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always

2	MHS	R/W	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 DW_apb_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. Values: 0x0 (DISABLE): Handshaking interface is disabled 0x1 (ENABLED): Handshaking interface is enabled Value After Reset: 0x0 Exists: SSI_IS_MASTER == 1

1	MDD	R/W	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. Values: 0x0 (RECEIVE): SSI receives data 0x1 (TRANSMIT): SSI transmits data Value After Reset: 0x0 Exists: Always
0	MWMOD	R/W	Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or nonsequential. 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. Values: 0x0 (NON_SEQUENTIAL): Non-Sequential Microwire Transfer 0x1 (SEQUENTIAL): Sequential Microwire Transfer Value After Reset: 0x0 Exists: Always

2.5 SER

- Name: Slave Enable Register
- Description: This register is valid only when the SSI is configured as a master device. When the SSI is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. The register enables the individual slave select output lines from the SSI master. Up to 16 slave-select output pins are available on the SSI master. Register bits can be set or cleared when SSI_EN=0. If SSI_EN=1, then register bits can be set (to delay the slave select assertion while TX FIFO is getting filled) but cannot be cleared.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x10
- Exists: SSI_IS_MASTER == 1

Table 8: SER Register Fields

Bits	Name	Memory Access	Description
31:y	RSVD_SER	R	SER Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Range Variable[y]: SSI_NUM_SLAVES
x:0	SER	*Varies	Slave Select Enable Flag. Each bit in this register corresponds to a slave select line (ss_x_n) from the DW_apb_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. Values: 0x0 (NOT_SELECTED): No slave selected 0x1 (SELECTED): Slave is selected Value After Reset: 0x0 Exists: Always Range Variable[x]: SSI_NUM_SLAVES - 1 Memory Access: “(SSI_IS_MASTER==1) ? \”read-write\” : \”read-only\”””

2.6 BAUDR

- Name: Baud Rate Select
- Description: This register is valid only when the SSI is configured as a master device. When the SSI is configured as a serial slave, writing to this location has no effect; reading from this location returns 0. The register derives the frequency of the serial clock that regulates the data transfer. The 16-bit field in this register defines the ssi_clk divider value. It is impossible to write to this register when the SSI is enabled. The SSI is enabled and disabled by writing to the SSIENR register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x14
- Exists: SSI_IS_MASTER == 1

Table 9: BAUDR Register Fields

Bits	Name	Memory Access	Description
31:16	RSVD_BAUDR	R	BAUDR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
15:0	SCKDV	R/W	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. For example: for Fssi_clk = 3.6864MHz and SCKDV =2 Fsclk_out = 3.6864/2 = 1.8432MHz The LSB for this field is always set to 0 and is unaffected by a write Value After Reset: 0x0 Exists: Always

2.7 TXFTLR

- Name: Transmit FIFO Threshold Level
- Description: This register controls the threshold value for the transmit FIFO memory. The SSI is enabled and disabled by writing to the SSIENR register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x18
- Exists: Always

Table 10: TXFTLR Register Fields

Bits	Name	Memory Access	Description
31:y	RSVD_TXFTLR	R	TXFTLR Reserved bits - Read Only Exists: Always Range Variable[y]: TX_ABW
x:0	TFT	R/W	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 this 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. For information on the Transmit FIFO Threshold values, see the “Master SPI and SSP Serial Transfers” in the SSI Databook. ssi_txe_intr is asserted when TFT or less data entries are present in transmit FIFO Value After Reset: 0x0 Exists: Always Range Variable[x]: TX_ABW - 1

2.8 RXFTLR

- Name: Receive FIFO Threshold Level
- Description: This register controls the threshold value for the receive FIFO memory. The SSI is enabled and disabled by writing to the SSIENR register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x1c
- Exists: Always

Table 11: RXFTLR Register Fields

Bits	Name	Memory Access	Description
31:y	RSVD_RXFTLR	R	RXFTLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Range Variable[y]: RX_ABW
x:0	RFT	R/W	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. For information on the Receive FIFO Threshold values, see the "Master SPI and SSP Serial Transfers" in the SSI Databook. Value After Reset: 0x0 Exists: Always Range Variable[x]: RX_ABW - 1

2.9 TXFLR

- Name: Transmit FIFO Level Register
- Description: This register contains the number of valid data entries in the transmit FIFO memory.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x20
- Exists: Always

Table 12: TXFLR Register Fields

Bits	Name	Memory Access	Description
31:y	RSVD_T	R	TXFLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true Range Variable[y]: TX_ABW + 1
x:0	TXFTL	R	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. Value After Reset: 0x0 Exists: Always Volatile: true Range Variable[x]: TX_ABW

2.10 RXFLR

- Name: Receive FIFO Level Register
- Description: This register contains the number of valid data entries in the receive FIFO memory. This register can be ready at any time.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x24
- Exists: Always

Table 13: RXFLR Register Fields

Bits	Name	Memory Access	Description
31:y	RSVD_R	R	RXFLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true Range Variable[y]: RX_ABW + 1
x:0	RXFTL	R	Receive FIFO Level. Contains the number of valid data entries in the receive FIFO. Value After Reset: 0x0 Exists: Always Volatile: true Range Variable[x]: RX_ABW

2.11 SR

- Name: Status Register
- Description: 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.
- Reset Value: 0x6
- Size: 32 bits
- Offset: 0x28
- Exists: Always

Table 14: SR Register Fields

Bits	Name	Memory Access	Description
31:7	RSVD_SR	R	SR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true
6	DCOL	R	Data Collision Error. Relevant only when the SSI is configured as a master device. This bit will be set if ss_in_n input is asserted by other master, when 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 Values: 0x0 (NO_ERROR_CONDITION): No Error 0x1 (TX_COLLISION_ERROR): Transmit Data Collision Error Value After Reset: 0x0 Exists: SSI_IS_MASTER == 1 Volatile: true
5	TXE	R	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. Values: 0x0 (NO_ERROR): No Error 0x1 (TX_ERROR): Transmission Error Value After Reset: 0x0 Exists: SSI_IS_MASTER == 0 Volatile: true
4	RFF	R	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. Values: 0x0 (NOT_FULL): Receive FIFO is not full 0x1 (FULL): Receive FIFO is full Value After Reset: 0x0 Exists: Always Volatile: true
3	RFNE	R	Receive FIFO Not Empty. Set when the receive FIFO contains one or more entries and is cleared to completely empty the receive FIFO. Values: 0x0 (EMPTY): Receive FIFO is empty 0x1 (NOT_EMPTY): Receive FIFO is not empty Value After Reset: 0x0 Exists: Always Volatile: true

2	TFE	R	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. Values: 0x0 (NOT_EMPTY): Transmit FIFO is not empty 0x1 (EMPTY): Transmit FIFO is 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: 0x0 (FULL): Transmit FIFO is full 0x1 (NOT_FULL): Transmit FIFO is not Full Value After Reset: 0x1 Exists: Always Volatile: true
0	BUSY	R	SSI Busy Flag. When set, indicates that a serial transfer is in progress; when cleared indicates that the DW_apb_ssi is idle or disabled. Values: 0x0 (INACTIVE): SSI is idle or disabled 0x1 (ACTIVE): SSI is actively transferring data Value After Reset: 0x0 Exists: Always Volatile: true

2.12 IMR

- Name: Interrupt Mask Register
- Description: This read/write reigster masks or enables all interrupts generated by the SSI. When the SSI is configured as a slave device, the MSTIM bit field is not present. This changes the reset value from 0x3F for serial-master configurations to 0x1F for serial-slave configurations.
- Reset Value: (SSI_IS_MASTER == 1) ? 0x3F : 0x1F
- Size: 32 bits
- Offset: 0x2c
- Exists: Always

Table 15: IMR Register Fields

Bits	Name	Memory Access	Description
31:6	RSVD_IMR	R	IMR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
5	MSTIM	*Varies	Multi-Master Contention Interrupt Mask. This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. Values: 0x0 (MASKED): ssi_mst_intr interrupt is masked 0x1 (UNMASKED): ssi_mst_intr interrupt is not masked Value After Reset: “(SSI_IS_MASTER==1) ? \”1\” : \”0\”” Exists: SSI_IS_MASTER == 1 Memory Access: “(SSI_IS_MASTER==1) ? \”read-write\” : "read-only\””
4	RXFIM	R/W	Receive FIFO Full Interrupt Mask. Values: 0x0 (MASKED): ssi_rxf_intr interrupt is masked 0x1 (UNMASKED): ssi_rxf_intr interrupt is not masked Value After Reset: 0x1 Exists: Always
3	RXOIM	R/W	Receive FIFO Overflow Interrupt Mask. Values: 0x0 (MASKED): ssi_rxo_intr interrupt is masked 0x1 (UNMASKED): ssi_rxo_intr interrupt is not masked Value After Reset: 0x1 Exists: Always
2	RXUIM	R/W	Receive FIFO Underflow Interrupt Mask. Values: 0x0 (MASKED): ssi_rxu_intr interrupt is masked 0x1 (UNMASKED): ssi_rxu_intr interrupt is not masked Value After Reset: 0x1 Exists: Always
1	TXOIM	R/W	Transmit FIFO Overflow Interrupt Mask. Values: 0x0 (MASKED): ssi_txo_intr interrupt is masked 0x1 (UNMASKED): ssi_txo_intr interrupt is not masked Value After Reset: 0x1 Exists: Always
0	TXEIM	R/W	Transmit FIFO Empty Interrupt Mask. Values: 0x0 (MASKED): ssi_txe_intr interrupt is masked 0x1 (UNMASKED): ssi_txe_intr interrupt is not masked Value After Reset: 0x1 Exists: Always

2.13 ISR

- Name: Interrupt Status Register
- Description: This register reports the status of the SSI interrupts after they have been masked.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x30
- Exists: Always

Table 16: ISR Register Fields

Bits | Name		Memory Access	Description
31:6	RSVD_ISR		ISR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true
5	MSTIS	R	Multi-Master Contention Interrupt Status. This bit field is not present if the SSI is configured as a serial-slave device. Values: 0x0 (INACTIVE): ssi_mst_intr interrupt not active after masking 0x1 (ACTIVE): ssi_mst_intr interrupt is active after masking Value After Reset: 0x0 Exists: SSI_IS_MASTER == 1 Volatile: true
4	RXFIS	R	Receive FIFO Full Interrupt Status Values: 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_rxf_intr interrupt is full after masking Exists: Always Volatile: true
3	RXOIS	R	Receive FIFO Overflow Interrupt Status Values: 0x0 (INACTIVE): ssi_rxo_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_rxo_intr interrupt is active after masking Value After Reset: 0x0 Exists: Always Volatile: true

2	RXUIS	R	Receive FIFO Underflow Interrupt Status Values: 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_rxu_intr interrupt is active after masking Value After Reset: 0x0 Exists: Always Volatile: true
1	TXOIS	R	Transmit FIFO Overflow Interrupt Status Values: 0x0 (INACTIVE): ssi_txo_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_txo_intr interrupt is active after masking Value After Reset: 0x0 Exists: Always Volatile: true
0	TXEIS	R	Transmit FIFO Empty Interrupt Status Values: 0x0 (INACTIVE): ssi_txe_intr interrupt is not active after masking 0x1 (ACTIVE): ssi_txe_intr interrupt is active after masking Value After Reset: 0x0 Exists: Always Volatile: true

2.14 RISR

- Name: Raw Interrupt Status Register
- Description: This read-only register reports the status of the SSI interrupts prior to masking.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x34
- Exists: Always

Table 17: RISR Register Fields

Bits	Name	Memory Access	Description
31:6	RSVD_RISR	R	RISR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true
5	MSTIR	R	Multi-Master Contention Raw Interrupt Status. This bit field is not present if the DW_apb_ssi is configured as a serial-slave device. Values: 0x0 (INACTIVE): ssi_mst_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_mst_intr interrupt is active prior masking Value After Reset: 0x0 Exists: SSI_IS_MASTER == 1 Volatile: true

4	RXFIR	R	Receive FIFO Full Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_rxf_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_rxf_intr interrupt is active prior to masking Value After Reset: 0x0 Exists: Always Volatile: true
3	RXOIR	R	Receive FIFO Overflow Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_rxo_intr interrupt is active prior masking 0x1 (ACTIVE): ssi_rxo_intr interrupt is not active prior to masking Value After Reset: 0x0 Exists: Always Volatile: true
2	RXUIR	R	Receive FIFO Underflow Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_rxu_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_rxu_intr interrupt is active prior to masking Value After Reset: 0x0 Exists: Always Volatile: true
1	TXOIR	R	Transmit FIFO Overflow Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_txo_intr interrupt is not active prior masking 0x1 (ACTIVE): ssi_txo_intr interrupt is active prior masking Value After Reset: 0x0 Exists: Always Volatile: true
0	TXEIR	R	Transmit FIFO Empty Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_txe_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_txe_intr interrupt is active prior to masking Value After Reset: 0x0 Exists: Always Volatile: true

2.15 TXOICR

- Name: Transmit FIFO Overflow Interrupt Clear Registers.
- Description: Transmit FIFO Overflow Interrupt Clear Register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x38
- Exists: Always

Table 18: TXOICR Register Fields

Bits	Name	Memory Access	Description
31:1	RSVD_TXOICR	R	TXOICR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true
0	TXOICR	R	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. Value After Reset: 0x0 Exists: Always Volatile: true

2.16 RXOICR

- Name: Receive FIFO Overflow Interrupt Clear Register
- Description: Receive FIFO Overflow Interrupt Clear Register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x3c
- Exists: Always

Table 19: RXOICR Register Fields

Bits	Name	Memory Access	Description
31:1	RSVD_RXOICR	R	RXOICR Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
0	RXOICR	R	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. Value After Reset: 0x0 Exists: Always Volatile: true

2.17 RXUICR

- Name: Receive FIFO Underflow Interrupt Clear Register
- Description: Receive FIFO Underflow Interrupt Clear Register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x40
- Exists: Always

Table 20: RXUICR Register Fields

Bits	Name	Memory Access	Description
31:1	RSVD_RXUICR	R	RXUICR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true
0	RXUICR	R	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. Value After Reset: 0x0 Exists: Always Volatile: true

2.18 MSTICR

- Name: Multi-Master Interrupt Clear Register
- Description: Multi-Master Interrupt Clear Register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x44
- Exists: Always

Table 21: MSTICR Register Fields

Bits	Name	Memory Access	Description
31:1	RSVD_MSTICR	R	MSTICR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true
0	MSTICR	R	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. Value After Reset: 0x0 Exists: Always Volatile: true

2.19 ICR

- Name: Interrupt Clear Register
- Description: Interrupt Clear Register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x48
- Exists: Always

Table 22: ICR Register Fields

Bits	Name	Memory Access	Description
31:1	RSVD_ICR	R	ICR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Volatile: true
0	ICR	R	Clear Interrupt. 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. Value After Reset: 0x0 Exists: Always Volatile: true

2.20 DMACR

- Name: DMA Control Register
- Description: This register is only valid when SSI is configured with a set of DMA Controller interface signals (SSI_HAS_DMA = 1). When SSI is not configured for DMA operation, this register will not exist and writing to the register's address will have no effect; reading from this register address will return zero. The register is used to enable the DMA Controller interface operation.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x4c
- Exists: SSI_HAS_DMA == 1

Table 23: DMACR Register Fields

Bits	Name	Memory Access	Description
31:2	RSVD_DMACR	R	DMACR 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: 0x0 (DISABLE): Transmit DMA disabled 0x1 (ENABLED): Transmit DMA enabled Value After Reset: 0x0 Exists: Always
0	RDMAE	R/W	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel Values: 0x0 (DISABLE): Receive DMA disabled Value After Reset: 0x0 Exists: Always

2.21 DMATDLR

- Name: DMA Transmit Data Level
- Description: This register is only valid when the SSI is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When SSI is not configured for DMA operation, this register will not exist and writing to its address will have no effect; reading from its address will return zero.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x50
- Exists: SSI_HAS_DMA == 1

Table 24: DMATDLR Register Fields

Bits	Name	Memory Access	Description
31:y	RSVD_DMATDLR	R	DMATDLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Range Variable[y]: TX_ABW
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. For information on the DMATDL decode values, see the “Slave SPI and SSP Serial Transfers” section in the DW_apb_ssi Databook. dma_tx_req is asserted when DMATDL or less data entries are present in the transmit FIFO Value After Reset: 0x0 Exists: Always Range Variable[x]: TX_ABW - 1

2.22 DMARDLR

- Name: DMA Receive Data Level
- Description: This register is only valid when SSI is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When SSI is not configured for DMA operation, this register will not exist and writing to its address will have no effect; reading from its address will return zero.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x54
- Exists: SSI_HAS_DMA == 1

Table 25: DMARDLR Register Fields

Bits	Name	Memory Access	Description
31:y	RRSVD_DMARDLRR		DMARDLR Reserved bits - Read Only Value After Reset: 0x0 Exists: Always Range Variable[y]: RX_ABW
x:0	DMATDL	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 above this field value + 1, and RDMAE=1. For information on the DMARDL decode values, see the “Slave SPI and SSP Serial Transfers” section in the DW_apb_ssi Databook. dma_rx_req is asserted when DMARDL or more valid data entries are present in the receive FIFO. Value After Reset: 0x0 Exists: Always Range Variable[x]: RX_ABW - 1

2.23 IDR

- Name: Identification Register
- Description: This register contains the peripherals identification code, which is written into the register at configuration time using coreConsultant.
- Reset Value: SSI_ID
- Size: 32 bits
- Offset: 0x58
- Exists: Always

Table 26: IDR Register Fields

Bits	Name	Memory Access	Description
31:0	IDCODE	R	Identification code. The register contains the peripheral’s identification code, which is written into the register at configuration time using CoreConsultant. Value After Reset: SSI_ID Exists: Always

2.24 SSI_VERSION_ID

- Name: coreKit version ID Register
- Description: This read-only register stores the specific SSI component version.
- Reset Value: SSI_VERSION_ID

- Size: 32 bits
- Offset: 0x5c
- Exists: Always

Table 27: SSI_VERSION_ID Register Fields

Bits	Name	Memory Access	Description
31:0	SSI_COMP_VERSION	R	Contains the hex representation of the Synopsys component version. Consists of ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01*. Value After Reset: SSI_VERSION_ID Exists: Always

2.25 DRx (for x = 0; x <= 35)

- Name: Data Register x
- Description: The SSI data register is a 16/32-bit (depending on SSI_MAX_XFER_SIZE) read/write buffer for the transmit/receive FIFOs. If the configuration parameter SSI_MAX_XFER_SIZE is set to 32, then all 32 bits are valid, otherwise, only 16 bits ([15:0]) of the register are valid. When the register is read, data in the receive FIFO buffer is accessed. When it is written to, data are moved into the transmit FIFO buffer; a write can occur only when SSI_EN = 1. FIFOs are reset when SSI_EN = 0.NOTE: The DR register in the SSI occupies thirty-six 32-bit address locations of the memory map to facilitate AHB burst transfers. Writing to any of these address locations has the same effect as pushing the data from the pwdata bus into the transmit FIFO. Reading from any of these locations has the same effect as popping data from the receive FIFO onto the prdata bus. The FIFO buffers on the SSI are not addressable.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0x60
- Exists: Always

Table 28: DRx Register Fields

Bits	Name	Memory Access	Description
31:16	RSVD_DR	R	DR{i} Reserved bits - Read Only Value After Reset: 0x0 Exists: SSI_MAX_XFER_SIZE == 16 Volatile: true
x:0	DR	R/W	Data Register. When writing to this register, you must rightjustify 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. Value After Reset: 0x0 Exists: Always Volatile: true Range Variable[x]: SSI_MAX_XFER_SIZE - 1

2.26 RX_SAMPLE_DLY

- Name: RX Sample Delay Register
- Description: This register is only valid when the SSI is configured with rxd sample delay logic (SSI_HAS_RX_SAMPLE_DELAY==1). When the SSI is not configured with rxd sample delay logic, this register will not exist and writing to its address location will have no effect; reading from its address will return zero. This register control the number of ssi_clk cycles that are delayed (from the default sample time) before the actual sample of the rxd input occurs. It is impossible to write to this register when the SSI is enabled. The SSI is enabled and disabled by writing to the SSIENR register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0xf0
- Exists: SSI_HAS_RX_SAMPLE_DELAY == 1

Table 29: RX_SAMPLE_DLY Register Fields

Bits	Name	Memory Access	Description
31:8	RSVD_RX_SAMPLE_DLYR		SAMPLE_DLY Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
7:0	RSD	R/W	Rxd Sample Delay. This register is used to delay the sample of the rxd input port. Each value represents a single ssi_clk delay on the sample of rxd. **Note:** If this register is programmed with a value that exceeds the depth of the internal shift registers (SSI_RX_DLY_SR_DEPTH) zero delay will be applied to the rxd sample. Value After Reset: 0x0 Exists: Always

2.27 SPI_CTRLR0

- Name: SPI Control Register
- Description: This register is valid only when SSI_SPI_MODE is either set to “Dual” or “Quad” or “Octal” mode. This register is used to control the serial data transfer in SPI mode of operation. The register is only relevant when SPI_FRF is set to either 01 or 10 or 11. It is not possible to write to this register when the SSI is enabled (SSI_EN=1). The SSI is enabled and disabled by writing to the SSIENR register.
- Reset Value: 0x00000200
- Size: 32 bits
- Offset: 0xf4
- Exists: SSI_SPI_MODE != 0

Table 30: SPI_CTRLR0 Register Fields

Bits	Name	Memory Access	Description
31:19	RSVD_SPI_CTRLR0	R	SPI_CTRLR0 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
18	SPI_RXDS_EN	*Varies	Read data strobe enable bit. Once this bit is set to 1 SSI will use Read data strobe (rxd) to capture read data in DDR mode. Value After Reset: 0x0 Exists: Always Memory Access: “(SSI_HAS_RXDS==0) ? \”Read-only\”:\”read-write\””
17	INST_DDR_EN	*Varies	Instruction DDR Enable bit. This will enable Dual-data rate transfer for Instruction phase. Value After Reset: 0x0 Exists: Always Memory Access: “(SSI_HAS_DDR==0) ? \”Read-only\”:\”read-write\””

16	SPI_DDR_EN	*Varies	SPI DDR Enable bit. This will enable Dual-data rate transfers in Dual/Quad/Octal frame formats of SPI. Value After Reset: 0x0 Exists: Always Memory Access: “(SSI_HAS_DDR==0) ? \”Read-only\”:\”read-write"”
15:11	WAIT_CYCLES	R/W	Wait cycles Number of wait cycles in Dual/Quad/Octal mode between control frames transmit and data reception. This value is specified as number of SPI clock cycles. For information on the WAIT_CYCLES decode value, see “Read Operation in Enhanced SPI Modes” section in the SSI Databook. Value After Reset: 0x0 Exists: Always
10	RSVD_SPI_CTRLR0_10	R	CTRLR0_10 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always

9:8	INST_L	R/W	Instruction Length Dual/Quad/Octal mode instruction length in bits. Values: 0x0 (INST_L_0): 0-bit (No Instruction) 0x1 (INST_L_1): 4-bit Instruction 0x2 (INST_L_2): 8-bit Instruction 0x3 (INST_L_3): 16-bit Instruction Value After Reset: 0x2 Exists: Always
7:6	RSVD_SPI_CTRLR0_6_7	R	CTRLR0_6_7 Reserved bits - Read Only Value After Reset: 0x0 Exists: Always

5:2	ADDR_L	R/W	Address Length. This bit defines Length of Address to be transmitted. Only after this much bits are programmed in to the FIFO the transfer can begin. For information on the ADDR_Ldecode value, see “Read Operation in Enhanced SPI Modes” section in the SSI Databook. Values: 0x0 (ADDR_L_0): 0-bit Address Width 0x1 (ADDR_L_1): 4-bit Address Width 0x2 (ADDR_L_2): 8-bit Address Width 0x3 (ADDR_L_3): 12-bit Address Width 0x4 (ADDR_L_4): 16-bit Address Width 0x5 (ADDR_L_5): 20-bit Address Width 0x6 (ADDR_L_6): 24-bit Address Width 0x7 (ADDR_L_7): 28-bit Address Width 0x8 (ADDR_L_8): 32-bit Address Width 0x9 (ADDR_L_9): 36-bit Address Width 0xa (ADDR_L_10): 40-bit Address Width 0xb (ADDR_L_11): 44-bit Address Width 0xc (ADDR_L_12): 48-bit Address Width 0xd (ADDR_L_13): 52-bit Address Width 0xe (ADDR_L_14): 56-bit Address Width 0xf (ADDR_L_15): 60-bit Address Width Value After Reset: 0x0 Exists: Always

1:0	TRANS_TYPE	R/W	Address and instruction transfer format. Selects whether SSI will transmit instruction/address either in Standard SPI mode or the SPI mode selected in 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 SPI_FRF. 11 - Reserved. Exists: Always

2.28 TXD_DRIVE_EDGE

- Name: Transmit Drive Edge Register
- Description: This Register is valid only when SSI_HAS_DDR is equal to 1. This register is used to control the driving edge of TXD register in DDR mode. It is not possible to write to this register when the SSI is enabled (SSI_EN=1). The SSI is enabled and disabled by writing to the SSIENR register.
- Reset Value: 0x0
- Size: 32 bits
- Offset: 0xf8
- Exists: SSI_HAS_DDR != 0

Table 31: TXD_DRIVE_EDGE Register Fields

Bits	Name	Memory Access	Description
31:8	RRSVD_TXD_DRIVE_EDGE	R	DRIVE_EDGE Reserved bits - Read Only Value After Reset: 0x0 Exists: Always
7:0	TDE	R/W	TXD Drive edge - value of which decides the driving edge of tramit data. The maximum value of this regster is = (BAUDR/2) -1. Value After Reset: 0x0 Exists: Always

2.29 RSVD

- Name: RSVD - Reserved address location
- Description: RSVD - Reserved address location.
- Size: 32 bits
- Offset: 0xfc
- Exists: Always

Table 32: RSVD Register Fields

Bits	Name	Memory Access	Description
31:0	RSVD	R	RSVD 31to0 Reserved address location Value After Reset: 0x0 Exists: Always Volatile: true

CHAPTER

SIXTEEN

LPC

16.1 Software Registers

LPC Host needs dedicated registers to support command types and data passing. Following table shows details of it.

Register	Offset	Reset Value	RW	Description
CONTROL	0x00	0x00	R/W	LPC Host Control Register field, which contains LPC Host information to transfer a valid transaction Bit 0: CTRL_ENABLE Core enable or disable control Bit 1: CTRL_SYNC_TIMEOUT Enable or disable SYNC timeout for long wait sync Bit 2: CTRL_LFRAME_TIMING LFRAME timing mode 0 - TYPICAL[LFRAME timing is typical, one LCLK wide] 1 - EXTENDED[LFRAME timing is extended, two LCLK wide] Bit 18:3: CTRL_PRESCALER This field controls the Prescaler value used to divide the clk to deliver the desired frequency Bit 26:19: CTRL_IFG Inter frame gap Bit 30:27: CTRL_ABORT_WIDTH Control the width of FRAME# while doing abort

COM-MAND	0x04	0x00	R/W	This register implements LPC Host command register Bit 0: CMD_PENDING Host Command frames are pending for processing. Software needs to set this bit to 1, to enable main FSM to send commands on LPC BUS. Hardware will clear it once done with processing of command Bit 1:CMD_ABORT LPC Host command abort in middle, can be set middle of transfer Bit 4:2 CMD_TYPE This field is used to communicate cycle type (memory, I/O, DMA) 3'b000 : I/O 3'b001 : MEMORY 3'b010 : DMA 3'b011 : FIRMWARE 3'b100 : BUS_MASTER 3'b101 : DRIVE_LPCD 3'b110 : DRIVE_SERIRQ 3'b111 : DRIVE_CLKRUN Bit 5: CMD_DIR LPC Host command direction encoding 0 - READ(Field cycle direction read) 1 - WRITE(Field cycle direction write) Bit 7:6 CMD_STATUS LPC Host command status encoding 0 - SYNC_STAT_TIMEOUT(Field indicating sync Timeout) 1 - SYNC_STAT_ERROR(Field indicating sync Error) Bit 10:8 CMD_DMA_CHANNEL

				LPC Host command DMA Channel encoding This field is driven by the host on DMA cycles to indicate to the peripheral which DMA channel has been granted 3'b000 : DMA_CHANNEL_0 3'b001 : DMA_CHANNEL_1 3'b010 : DMA_CHANNEL_2 3'b011 : DMA_CHANNEL_3 3'b100 : DMA_CHANNEL_4 3'b101 : DMA_CHANNEL_5 3'b110 : DMA_CHANNEL_6 3'b111 : DMA_CHANNEL_7 Bit 11: CMD_DMA_SIZE LPC Host command DMA transfer size 0 - DMA_SIZE_8B(DMA transfer size is 8 bits) 1-DMA_SIZE_16B(DMA transfer size is 16 bits) Bit 13: CMD_SIGNAL_VALUE LPC Host command LPCP/SERIRQ/CLKRUN bits values to drive Bit 16:14 : CMD_MSIZE[Firmware Message Size] Bit16[4] Firmware transfer size is 1,2,4,16,128 bytes 3'b000 : MSIZE_1B 3'b001 : MSIZE_2B 3'b010 : MSIZE_4B 3'b011 : MSIZE_16B 3'b100 : MSIZE_128B

IDSEL_MADDR	0x08	0x00	R/W	Register to hold Firmware cycle IDSEL, Maddr fields or Memory/IO address or Peripheral address Bit 27:0 : MADDR Bit 31:28 : IDSEL[Firmware Msize] This field is used to indicates the byte size for FIRMWARE transfer
LPCPD_20MS_TIME	0x0C	0x00	R/W	Bit 19:0: Register to control the 30 microseconds to stop/start LCLK# with respect to LPCPD#
LONG_WAIT_SYNC_TIMEOUT	0x10	0x00	R/W	Bit 31:0: Register to control the timeout of long wait sync
PERIPHERAL_C OMMAND	0x14	0x00	R/W	This register implements LPC Host peripheral command received Bit 1:0 : PCMD_TYPE[LPC Host Peripheral command type encoding] Bit 2 : PCMD_DIR[LPC Host Peripheral command direction encoding] Bit 4:3 : PCMD_SIZE[LPC Host Peripheral command size of transfer]
FIFO_STATUS	0x18	0x00	R	This register is holds FIFO status Bit 0: WR_FIFO_EMPTY This bit would get set when Write FIFO emptyhappens Bit 1 : WR_FIFO_FULL This bit would get set when Write FIFO full happens Bit 2: RD_FIFO_EMPTY This bit would get set when Read FIFO empty happens Bit 3: RD_FIFO_FULL This bit would get set when Read FIFO full happens Bit 4: REQ_FIFO_EMPTY This bit would get set when Request FIFO empty happens Bit 5: REQ_FIFO_FULL This bit would get set when Request FIFO full happens Bit 10 :6 : WR_FIFO_ELEMENTS Write FIFO number of elements Bit 15:11 : RD_FIFO_ELEMENTS Read FIFO number of elements 20:16 : REQ_FIFO_ELEMENTS Request FIFO number of elements

SPI_FIFO_CFG	0x1C	0x00	W	This register controls Write, Read and Receive FIFO flush operation Bit 0: WR_FIFO_FLUSH This field is flush the WR FIFO Bit 1 : RD_FIFO_FLUSH Register to flush the RD FIFO Bit 2 : REQ_FIFO_FLUSH Register to flush the REQ FIFO
IRQ_ENABLE	0x20	0x00	R/W	Host IRQ enable register. This register controls the masking of interrupt. When the particular bit in this register is '0' then the corresponding Interrupt in the IRQ_STATUS register will be masked. Bit 0 : This bit enables IRQ when core has finished sending all commands. Bit 1 : This bit enables IRQ when DMA request from Device/Peripherals Bit 2 : This bit enables IRQ when Bus Master request from Device/Peripherals Bit 3 : This bit enables IRQ when CLKRUN signal is asserted Bit 4 : This bit enables IRQ when LPME signal is asserted Bit 5 :This bit enables IRQ when SERIRQ signal is asserted Bit 6 : This bit enables IRQ when LSMI signal is asserted Bit 7 : This bit enables IRQ when Write FIFO empty Bit 8 : This bit enables IRQ when Write FIFO full Bit 9: This bit enables IRQ when Read FIFO empty Bit 10: This bit enables IRQ when Read FIFO full Bit 11: This bit enables IRQ when Request FIFO empty Bit 12: This bit enables IRQ when Request FIFO full

IRQ_STATUS	0x24	0x00	R/W	Host IRQ Status register. This register controls the masking of interrupt. When the particular bit in this register is '0' then the corresponding Interrupt in the IRQ_STATUS register will be masked. Bit 0 : This bit is set when core has finished sending all commands. Bit 1 : This bit enables IRQ when DMA request from Device/Peripherals Bit 2 : This bit is set when Bus Master request from Device/Peripherals Bit 3 : This bit set when CLKRUN signal is asserted Bit 4 : This bit is set when LPME signal is asserted Bit 5 : This bit is set when SERIRQ signal is asserted Bit 6 : This bit is set when LSMI signal is asserted Bit 7 : This bit is set when Write FIFO empty Bit 8 : This bit is set when Write FIFO full Bit 9: This bit enables IRQ when Read FIFO empty Bit 10: This bit is set when Read FIFO full Bit 11: This bit is set when Request FIFO empty Bit 12: This bit is set when Request FIFO full
SOC_TIMEOUT	0x28	0x00	R/W	This register controls the timeout value for the SOC interface.
WR_FIFO	0x2C	0x00	W	Bit 31:0 : WR_FIFO This field is 2 clocks wide, representing one data byte. The host drives it on target, DMA, I/O, Memory data.
RD_FIFO	0x30	0x00	R	Bit 31:0 : RD_FIFO This field is 2 clocks wide, representing one data byte. The peripheral drives its when data is flowing to the host. Used for Read Transfer
REQ_FIFO	0x34	0x00	R	Bit 4:0 : REQ_FIFO This field is 2 clocks wide, representing one data byte. The peripheral drives it DMA/Bus Master Request.

CHAPTER

SEVENTEEN

INDICES AND TABLES

- genindex
- modindex
- search