

ESWIN

EIC7700X SoC Technical Reference Manual

Public Release / Rev 1.1

2024/01/31

Notice

Copyright © 2024 by Beijing ESWIN Computing Technology Co., Ltd., and its ("ESWIN Computing"). All rights reserved.

ESWIN Computing retain all intellectual property and proprietary rights in and to this product. Without as expressly authorized, no part of the software may be released, copied, distributed, reproduced, modified, adapted, translated, or created derivative work of, in whole or in part.

INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND DOES NOT REPRESENT A COMMITMENT ON THE PART OF ESWIN. UNLESS OTHERWISE SPECIFIED IN THE CONTRACT, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT ARE PROVIDED "AS IS" WITHOUT WARRANTIES, GUARANTEES, OR REPRESENTATIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED.

"ESWIN" logo, and other ESWIN icons, are trademarks of Beijing ESWIN Computing Technology Co., Ltd. and its parent company.

All other trademarks and trade names mentioned in this document are the property of their respective holders.

If applicable, Open-Source Software and/or free software licensing notices are available in the product installation.

Change History

Version No	Date	Descriptions
V1.1	2024-1-31	

Content

10 HIGH-SPEED INTERFACE.....	1
10.1 PCI-EXPRESS	1
10.2 GMAC	77
10.3 USB	263
11 SYSTEM SECURITY.....	353
12 PERIPHERAL DEVICES.....	353
12.1 I/O CONTROL (CLMM)	354
12.2 I2C.....	396
12.3 I2S.....	484
12.4 UART	519
12.5 SPI.....	564
12.6 GPIO.....	602
12.7 JTAG	617
12.8 PWM/TIMER	617
12.9 FAN_CTRL.....	629

Content of Figures

Figure 10-1 PCI-Express subsystem block diagram	1
Figure 10-2 One Inbound Region address map	3
Figure 10-3 Two Inbound Regions address map	4
Figure 10-4 Data transfer of DMA write channel	6
Figure 10-5 Data transfer of DMA read channel	6
Figure 10-6 Element structure of DMA Linked list	8
Figure 10-7 A platform of ATS	10
Figure 10-8 Initialization sequence in RC mode	11
Figure 10-9 Initialization sequence in EP mode.....	11
Figure 10-10 GMAC subsystem block diagram	78
Figure 10-11 RGMII TX Timing diagram.....	79
Figure 10-12 RGMII RX Timing diagram	80
Figure 10-13 RMII Timing diagram	80
Figure 10-14 USB Diagram.....	264
Figure 12-1 I2C block diagram	396
Figure 12-2 TX fifo Data format	397
Figure 12-3 Spike Suppression Example.....	397
Figure 12-4 IC_SDA_HOLD Register	399
Figure 12-5 RX Hold time diagram.....	399
Figure 12-6 TX Hold time diagram.....	400
Figure 12-7 I2C MASTER programming example	482
Figure 12-8 I2C slave programming example	483
Figure 12-9 I2C timing sequence	484
Figure 12-10 I2S block diagram	484
Figure 12-11 master transmiting process	517
Figure 12-12 I2S MASTER receiving flow chart	518
Figure 12-13 I2S timing sequence	518
Figure 12-14 uart block diagram.....	519
Figure 12-15 Serial Data Format	520
Figure 12-16 uart Auto Flow Control.....	521
Figure 12-17 Auto CTS timing	521
Figure 12-18 Auto CTS Timing	522

Figure 12-19 Flowchart of Interrupt Generation for Programmable THRE Interrupt Mode.....	522
Figure 12-20 Flowchart for uart Transmit Programming Example.....	563
Figure 12-21 Flowchart for uart Receive Programming Example.....	564
Figure 12-22 SPI block diagram.....	565
Figure 12-23 Motorola Serial Peripheral Interface(SCPH = 0, SSTE = 1).....	566
Figure 12-24 Motorola Serial Peripheral Interface(SCPH = 0, SSTE = 0).....	566
Figure 12-25 SPI Serial Format Continuous Transfer (SCPH = 1)	567
Figure 12-26 Single Master Microwire Serial Transfer (MDD=0).....	567
Figure 12-27 Instruction Transmitted in Standard and Address Transmitted in Enhanced SPI Format	568
Figure 12-28 Typical Read Operation in Enhanced SPI Mode	568
Figure 12-29 Master SPI Transfer Flow	600
Figure 12-30 Master Microwire Transfer Flow	601
Figure 12-31 SPI timing sequence	601
Figure 12-32 GPIO block diagram	602
Figure 12-33 JTAG timing sequence.....	617
Figure 12-34 PWM/TIMER block diagram.....	618

Content of Tables

Table 10-1 PCIe local address mapping	2
Table 10-2 Remote host access address map.....	2
Table 10-3 iATU Inbound register map.....	4
Table 10-4 iATU Outbound register map.....	5
Table 10-5 Third channel of read register map.....	7
Table 10-6 Register configuration of Start LL mode.....	7
Table 10-7 Low power state	8
Table 10-8 RGMII Transmit Timing Requirement	79
Table 10-9 RGMII Receive Timing Requirement	79
Table 10-10 RMII Timing Requirement	80
Table 12-1 Peripheral address space.....	353
Table 12-2 UART Interrupt Control Functions	523

10 High-Speed Interface

10.1 PCI-Express

10.1.1 Overview

PCI-Express is a high performance I/O bus used to interconnect peripheral device such as computing and communication platforms, and has following key features.

- PCIe Root-Complex(RC) and End-Point(EP) dual mode.
- Support PCIe Gen3 x4, compatible with PCIe Specification Revision 3.0
- EP mode supports one physical function.
- EP mode supports 4 BARs, each BAR size is independent configurable.
- EP mode supports INTX, MSI, MSI-X interrupt and PVM function.
- RC mode supports receiving INTX, MSI interrupt.
- Embedded 8 DMA read channels and 8 DMA write channels, and support DMA link-list mode.
- Support iATU(address mapping),4 inbound regions and 32 outbound regions, each outbound size is configurable which up to 8GB.
- Support L0s/L1/L1sub and D1/D2/D3hot low power mode , support PM and ASPM mechanism
- Support PERST#, Hot Reset, FLR reset .
- RC mode supports ATS.
- Max Payload Size is 512Byte, Non-Posted outstanding up to 32.

10.1.2 Block diagram

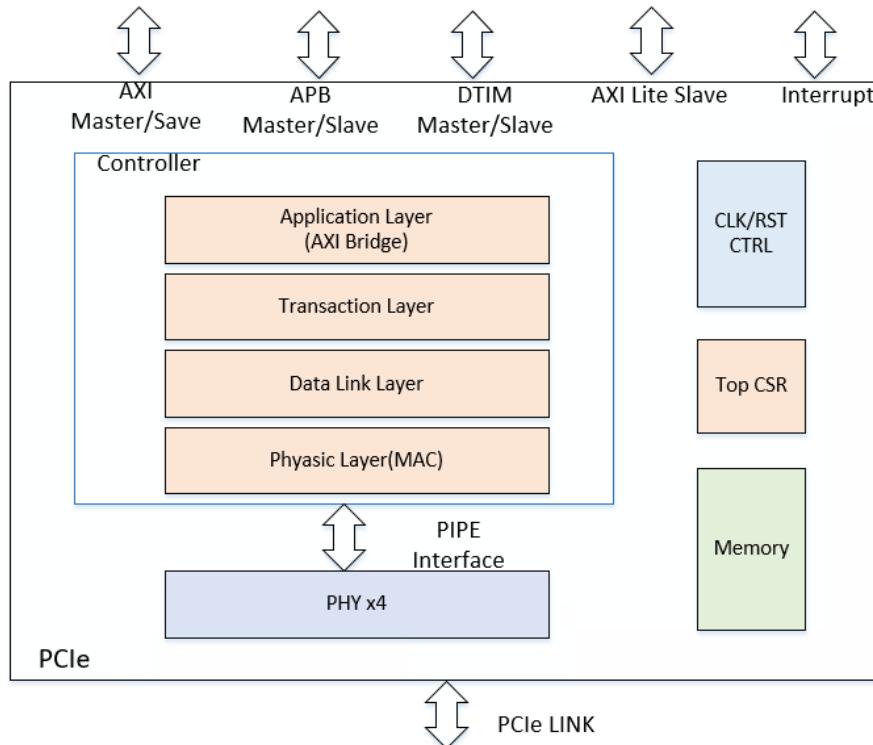


Figure 10-1 PCI-Express subsystem block diagram

10.1.3 Function description

10.1.3.1 Local address map

PCIe local address map as bellow table, there are three regions according to different interface.

Table 10-1 PCIe local address mapping

Interface	Address region	description
APB Slave	0x000_5000_0000 ~ 0x000_5007_ffff	PCIe Top Csr register space
	0x000_5008_0000 ~ 0x000_500f_ffff	PCIe Phy Register Space
AXI-lite Slave	0x000_5400_0000 ~ 0x000_5400_0fff	PCIe 4KB Configure Register Space
	0x000_5410_1000 ~ 0x000_5410_ffff	PCIe Controller Shadow and CS2 Register Space
	0x000_5430_0000 ~ 0x000_5437_ffff	iATU Register Space
	0x000_5438_0000 ~ 0x000_543f_ffff	DMA Register Space
	0x000_54B0_0000 ~ 0x000_54B0_7fff	MSI-X Table Address Space
	0x000_54B0_8000 ~ 0x000_54B0_ffff	MSI-X PBA Address Space
AXI slave	0x000_4000_0000 ~ 0x000_4fff_ffff	Access remote EP Configure and BAR0 Register Space in RC mode
	0x080_0000_0000 ~ 0x09f_ffff_ffff	Outbound Access remote RC Memory in EP mode or Memory Map Access remote EP in RC mode

10.1.3.2 BARs space

As an endpoint, PCIe has five address spaces, such as 4KB configure register space, BAR0, BAR1, BAR2 and BAR4. BAR0 and BAR1 are memory type spaces and non-prefetchable with 32bits address width, BAR2 and BAR4 are also memory type spaces and prefetchable with 64bits address width. BAR0 is responsible for configuring local chip register via remote host, BAR1 corresponds to register address of iATU/DMA/MSI-X table/MSI-X PBA in PCIe subsystem, BAR2 and BAR4 are corresponds to memory map address space for accessing local chip memory as bellow table.

Table 10-2 Remote host access address map

Access Type	Address Region	Access Address
Cfg Request	Configure Space	BDF + Register offset
MRd/MWr	BAR0	BAR0 + Register offset
MRd/MWr	BAR1	BAR1 + DMA Register offset (register of DMA)
		BAR1 + 0x2000 + iATU Register offset (register of iATU)
		BAR1 + 0x8000 + MSI-X Table Register offset(MSI-X Table)
		BAR1 + 0x8400 + MSI-X PBA Register offset(MSI-X PBA)
MRd/MWr	BAR2	BAR2 + Memory Address offset
MRd/MWr	BAR4	BAR4 + Memory Address offset

10.1.3.3 Interrupt

- MSI

As an endpoint, MSI supports 32 vectors, vector[0] is used by PCIe dma interrupt and vector[31:1] is for local mcpu, the 511 interrupts of mcpu are combined into 31 groups, each group corresponds to a vector, vector[1] for mcpu_interrupt[15:0], vector[2] for mcpu_interrupt[31:16]..., and the last vector[31] for mcpu_interrupt[511:480].

As an RC, when the address of received MWr matches MSI_CTRL_ADDR_OFF and MSI_CTRL_UPPER_ADDR_OFF register, a MSI interrupt of USP is detected, RC assert a local interrupt signal (msi_ctrl_int) to mcpu, then the mcpu read MSI_CTRL_INT_0_STATUS_OFF register or MSI_CTRL_INT_0_EN_OFF register (if MSI is masked) and identify the MSI vector.

- MSI-X

RC mode doesn't support MSI-X, when in EP mode, PCIe supports 512 interrupt sources. there are two methods to trigger MSI-X request, one is doorbell through configure the VECTOR and TC fields of MSIX_DOORBELL_OFF register; second is that issues a MWr at AXI slave interface when an address equal to MSIX_ADDRESS_MATCH_LOW_OFF and MSIX_ADDRESS_MATCH_HIGH_OFF, the PCIe controller extracts the vector and TC information from the MWr payload and create the MSI-X request.

- Legacy interrupt

Software configures the SYS_INT field of PCIE_TOP_CSR register to asserts the level-sensitive sys_int and notify the controller that it should send an interrupt message in EP mode, the controller generates two message TLPs, Assert_INTA and Deassert_INTA, in response to the assertion and the de-assertion of this signal. The controller decodes received Assert_INTx/Deassert_INTx messages and pulses the radm_intx_asserted and radm_intx_deasserted output to local mcpu in RC mode, where x=A,B,C,D.

Notes : recommend using MSIX interrupt in EP mode.

10.1.3.4 iATU

The PCIe uses the iATU(internal address translation unit) to implement a local address translation scheme that replaces the TLP address and TLP header fields in the current TLP request header as bellow diagram.

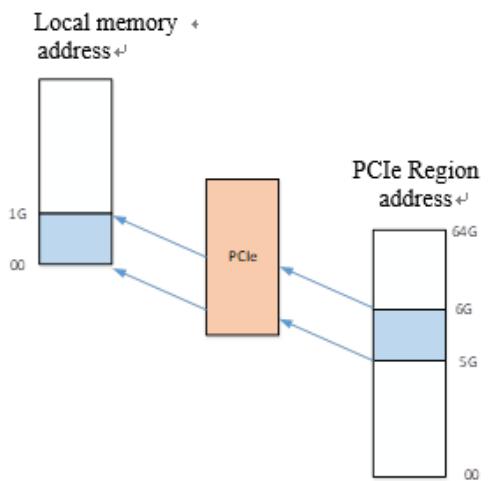


Figure 10-2 One Inbound Region address map

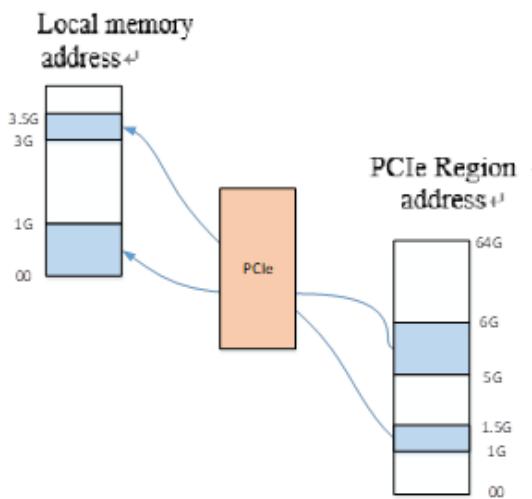


Figure 10-3 Two Inbound Regions address map

- Inbound

There are two inbound regions and iATU processes inbound requests as two translation methods which are address match mode and BAR match mode. TLPs destined for BAR0 or BAR1 in an upstream port are not translated, TLPs that are not error-free(ECRC,malformed,and so on) are not translated

When inbound is configured as address match mode, the field of each request TLP is checked to see if falls into any of the enabled address regions. When an address match is found then the TLP address field is modified as follows:

$$\text{Address} = \text{Address} - \text{Base_Address} + \text{Target_Address}$$

When the TLP address field matches more than one of inbound address regions, then the first enabled region to be matched is used.

When inbound is configured as BAR match mode, BAR matching mechanism check if the address field of any MEM and I/O request TLP falls into any address region defined by the enabled BAR addresses and masks, when a matched BAR is found, the iATU compares the BAR ID to the BAR number field in the IATU_REGION_CTRL_2_OFF_INBOUND_i register for all enabled regions

Inbound programming as following table and provides details of programming examples, you can access the iATU registers through the local CPU or through BAR matched MEM/IO requests.

Table 10-3 iATU Inbound register map

Offset address	Label(i=Region Num)	Description
0x100	IATU_REGION_CTRL_1_OFF_INBOUND_i	Region Control 1
0x104	IATU_REGION_CTRL_2_OFF_INBOUND_i	Region Control 2
0x108	IATU_LWR_BASE_ADDR_OFF_INBOUND_i	Lower Base Address
0x10C	IATU_UPPER_BASE_ADDR_OFF_INBOUND_i	Upper Base Address
0x110	IATU_LIMIT_ADDR_OFF_INBOUND_i	Limit Address
0x114	IATU_LWR_TARGET_ADDR_OFF_INBOUND_i	Lower Target Address
0x118	IATU_UPPER_TARGET_ADDR_OFF_INBOUND_i	Upper Target Address
0x11c	IATU_REGION_CTRL_3_OFF_INBOUND_i	Region Control 3
0x120	IATU_UPPER_LIMIT_ADDR_OFF_INBOUND_i	Upper Limit Address

- Outbound

There are 32 outbound regions, each region size can be configured to 4KB~8GB. Address translation is used for mapping different address ranges to different memory spaces, a typical example maps local memory space to PCI memory space, the iATU also supports type translation.

The iATU supports TYPE translation of MEM and I/O TLPs to Msg/MsgD TLPs. When there is a successful address match on an outbound MEM TLP, and the translated TLP type field is MSG, then the message code field of the TLP is set to the value in the Message Code field of the IATU_REGION_CTRL_2_OFF_OUTBOUND_i register. A MWr with an effective length of 0 is converted to Msg and all other MWr TLPs are converted to MsgD.

The iATU supports translation of I/O and MEM TLPs to CFG TLPs. The 16-bit BDF is located at bit[31:16] of the translated address where:

$$\text{Translated_Address} = \text{Original_Address} - \text{Base_Address} + \text{Target_Address}$$

As an example

$$\{13'h0, \text{function_num}[2:0]\}=\text{Original_Address}[31:16]$$

$$\text{Base_Address}[31:16]=16'h0$$

$$\text{Target_Address}[31:16]=\{\text{bus_num}[7:0],\text{device_num}[4:0],3h0\}$$

Then

$$\text{Translated_Address}[31:16]=\text{BDF}=\{\text{bus_num}[7:0],\text{device_num}[4:0],\text{bus_num}[7:0],\text{device_num}[4:0]\}$$

).

For CFG transactions created directly by software, you must ensure that the BDF field does not match any programmed iATU address region or else unintentional type translation could occur.

Outbound programming as below table which provides details of programming examples, you can access the iATU register through local CPU or through BAR matched MEM/IO requests.

Table 10-4 iATU Outbound register map

Offset address	Label(i=Region Num)	Description
0x000	IATU_REGION_CTRL_1_OFF_OUTBOUND_i	Region Control 1
0x004	IATU_REGION_CTRL_2_OFF_OUTBOUND_i	Region Control 2
0x008	IATU_LWR_BASE_ADDR_OFF_OUTBOUND_i	Lower Base Address
0x00C	IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_i	Upper Base Address
0x010	IATU_LIMIT_ADDR_OFF_OUTBOUND_i	Limit Address
0x014	IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i	Lower Target Address
0x018	IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_i	Upper Target Address
0x01c	IATU_REGION_CTRL_3_OFF_OUTBOUND_i	Region Control 3
0x020	IATU_UPPER_LIMIT_ADDR_OFF_OUTBOUND_i	Upper Limit Address

10.1.3.5 DMA

You can configure the DMA(HDMA) to have one to read channels and one to write channels, it can simultaneously perform the following types of memory translations. DMA write, transfer of a block of data from local memory to remote memory; DMA read, transfer of a block of data from remote memory to local memory. Therefore, the DMA supports full duplex operation, processing read and write transfers at the same time, and in parallel with normal(non-DMA) traffic. Upon completion of a DMA transfer or

an error, the DMA optionally interrupts the local CPU or sends an interrupt IMWr to the remote CPU. The DMA is highly configurable, and you can program it using the local CPU or over the PCIe wire through BAR1 address space.

The DMA can operate in non-linked list mode or linked list mode. In linked list mode, the DMA fetches the transfer control information for each transfer from a list of DMA elements that you have constructed in local memory.

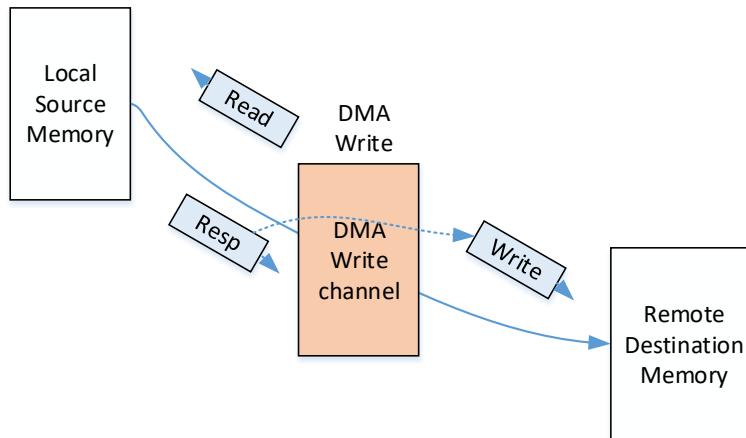


Figure 10-4 Data transfer of DMA write channel

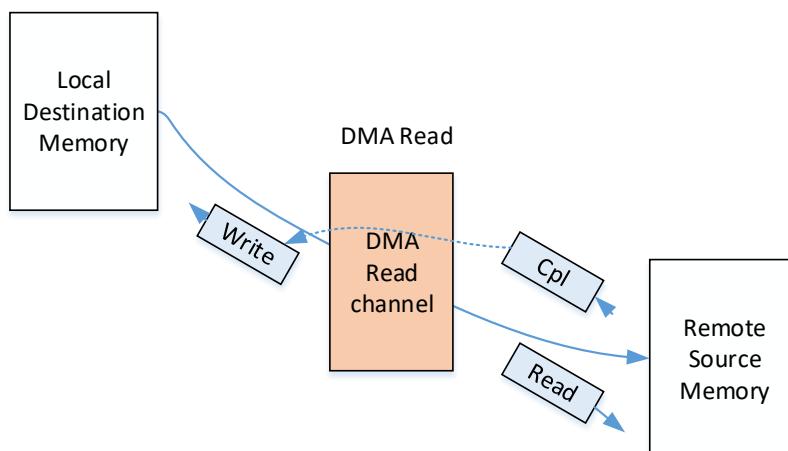


Figure 10-5 Data transfer of DMA read channel

The DMA generates the following interrupts per channel:

- Done: the DMA successfully completes the transfer
- Watermark: Interrupt generated at the end of each watermark event(end of linked list element). Generated in linked list mode only.
- Abort: The DMA fails to complete the transfer, or an error occurs during the transfer.

The interrupts are signaled to the software on local CPU, using one or both of the following mechanisms:

- Locally through the edma_int bus
- Remotely using a posted memory write(IMWr), which can be interpreted as an MSI or MSI-X when directed toward the remote RC.
- There are three programmable IMWr address per channel, one each for the done, watermark, and abort interrupts. For MSI, you must program all IMWr address registers with the same MSI address, as PCIe only supports a single MSI address per function.

- A single IMWr data register is used for all the interrupts, so you must read HDMA_INT_STATUS_OFF_[WR|RD]_i to identify the interrupt type.

Following is the programming example for enable third channel of read ,and enable remote interrupt.

Table 10-5 Third channel of read register map

Local access address	Remote access address	description	Value
0x54380000+0x2C	BAR1+0x2C	DMA read engine enable	0x1
0x54380000+0xCC/D0	BAR1+0xCC/D0	DMA read done IMWR addr	
0x54380000+0xD4/D8	BAR1+0xD4/D8	DMA read abort IMWR addr	
0x54380000+0xE4	BAR1+0xE4	DMA read channel 3 IMWr Data	
0x54380000+0x800	BAR1+0x800	DMA channel control1 register Local interrupt Enable (LIE)=0 Remote interrupt Enable (RIE)=1	0x4000008
0x54380000+0x808	BAR1+0x808	DMA transfer size	
0x54380000+0x80C	BAR1+0x80C	DMA SAR low	
0x54380000+0x810	BAR1+0x810	DMA SAR high	
0x54380000+0x814	BAR1+0x814	DMA DAR low	
0x54380000+0x818	BAR1+0x818	DMA DAR high	
0x54380000+0x010	BAR1+0x010	DMA write doorbell	0x3

The DMA provides a linked list (LL) mode to efficiently move data from source to destination with minimal intervention from the local CPU. This mode provides an alternative to programming the DMA multiple times to transfer multiple blocks of data. The programming information (address, size, and so on) for each block of memory is pre-programmed by your software into a LL element (also known as a descriptor) in local memory. Each element (called a data element) in the LL structure (called a transfer list) can transfer up to 4 GB of data.

You enable LL operation for a channel, by setting the LLEN field of the HDMA_CONTROL1_OFF_[WR|RD]CH_i register to 1. You can enable LL mode independently for each channel. When you enable LL for more than one channel, then you must have a separate LL structure in local memory for each channel. Your must produce the LL element structure in local memory as shown in Figure xx. Normally, all of the elements are contiguous (one after the other) in memory, and each element has six DWORDs containing the information about the block of data to be transferred, RIE is remote interrupt enable,LIE is local interrupt enable,LLP is next linked list pointer,TCB indicate whether repeat to operate this linked list and CB indicate this element is the last one. You program the channel context registers (HDMA_LL_P_LOW_OFF_[WR|RD]CH_i and HDMA_LL_P_HIGH_OFF_[WR|RD]CH_i) with the location of where you have placed the LL element structure in local memory. following table is the example programming for LL mode.

Table 10-6 Register configuration of Start LL mode

Local access address	Host access address	description	Value
0x54380000+0x30	BAR1+0x30	DMA write enable	0x1

Local access address	Host access address	description	Value
0x54380000+0x200	BAR1+0x200	DMA channel control1 register Linked list enable (LLE) =1	0x4000300
0x54380000+0x21C	BAR1+0x21C	DMA Linked list pointer low	
0x54380000+0x220	BAR1+0x220	DMA Linked list pointer high	
0x54380000+0x10	BAR1+0x10	DMA Write doorbell	0x0

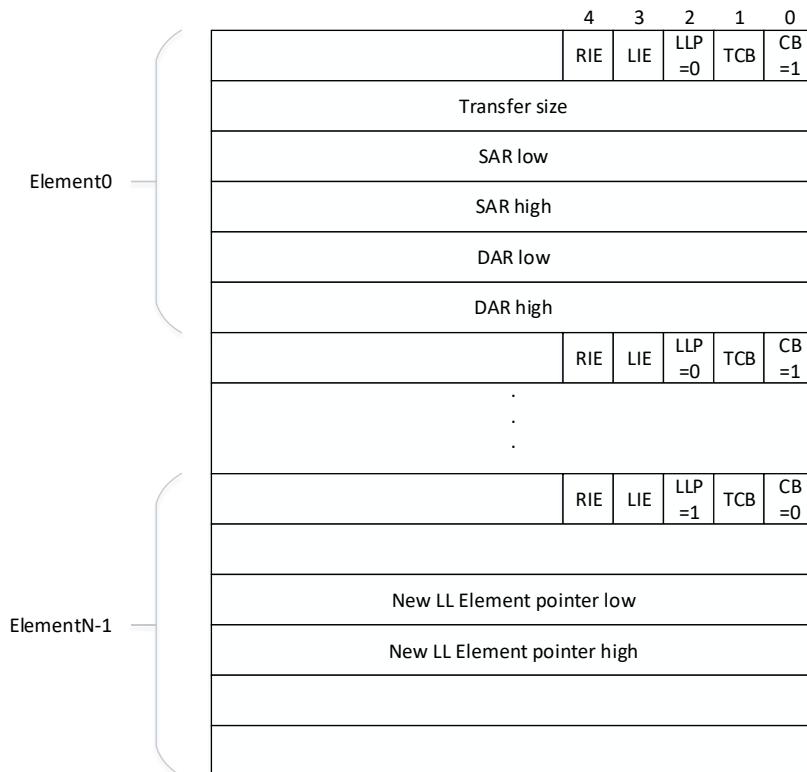


Figure 10-6 Element structure of DMA Linked list

10.1.3.6 Low power

PCIe supports two categories of PM operations to control the device state(D-state) and link state as following table, which are PMSCR and ASPM.

Table 10-7 Low power state

D-State	Link State	State Trigger	PHY State	Active Clock supplies	Pipe_clk Core_clk
D0uninitialized	L0	ASPM	P0, P0s	Refclk/Auxclk	On
D0active	L0	ASPM	P0, P0s	Refclk/Auxclk	On
	L0s			Refclk/Auxclk	On
	L1/L1.sub			Auxclk	Off

D-State	Link State	State Trigger	PHY State	Active Clock supplies	Pipe_clk Core_clk
D1,D2,D3hot	L1	PCI-PM	P1	Refclk/Auxclk	On
D3cold	L2		P2	Auxclk	Off
	L3		P1.2PG	None	Off

- L0s is a low power state enabled by ASPM:
- configure LINK_CONTROL_STATUS_REG register bit[1:0] to 2'b01, enable L0s.
- L0s entry after PCIe being in an idle state for a L0s_ENTRANCE_LATENCY of time if no higher stage of power-down requested.
- L0s exit when any of these condition can be met, such as A DLLP and TLP is pending to be sent or PCIe link partner requests to enter into link recovery.
- L1 is enable by ASPM:
- configure LINK_CONTROL_STATUS_REG register bit[1:0] to 2'b10, enable L1.
- L1s entry after PCIe being in an idle state for a L0s_ENTRANCE_LATENCY of time if no higher stage of power-down requested.
- L0s exit when any of these condition can be met, such as A DLLP and TLP is pending to be sent or PCIe link partner requests to enter into link recovery and neet to negotiate with partner.
- L1 is enable by ASPM when idle timeout in L0:
- configure LINK_CONTROL_STATUS_REG register bit[1:0] to 2'b11, enable L0s and L1.
- L0s exit when any of these condition can be met, such as A DLLP and TLP is pending to be sent or PCIe link partner requests to enter into link recovery and neet to negotiate with partner.
- L1.sub enable and exit by ASPM
- configure LINK_CONTROL_STATUS_REG register to bit[1:0] is 2'b1x, enable L1.
- configure L1SUB_CONTROL_REG register to enable L1.1 or L1.2.
- enter L1.sub after L1 state, PCIe controls CLKREQ# signal to turn off refclk and then L1.sub.
- L1.sub exit conidition is the same as L1, PCIe controls CLKREQ# signal to turn on refclk and negotiate to exit L1 with partner.
- Enter and exit L1 by PCI-PM
- remote host configure PMCSR of USP enable D1、D2 or D3hot.
- USP initiate a link state transition to L1, Both Link partners must negotiate to go this state, the L1-PM negotiation handshake uses PM_ENTER_L1 and PM_Request_Ack DLLPs.
- A DLLP and TLP is pending to be sent or PCIe link partner requests or configure PMCSR to exit L1

10.1.3.7 ATS

PCIe supports ATS function in RC mode(does not support ATS in EP mode), as following diagram, circumscribed SMMU implements address translation, in response to the translation request and ATS invalidation.

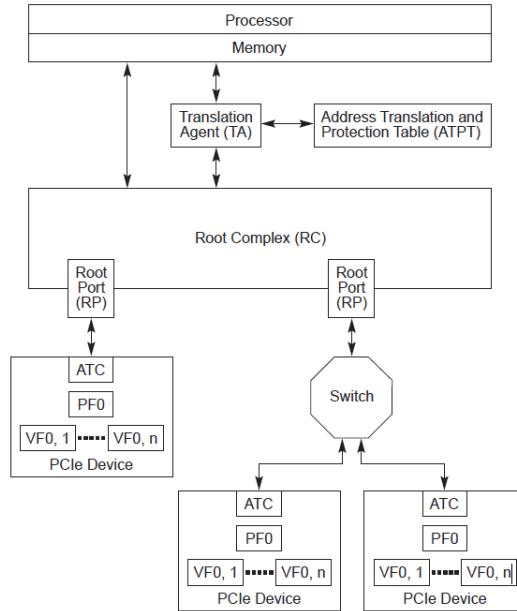


Figure 10-7 A platform of ATS

The AT field of MWr/MRd TLPs from USP illustrates the information of ATS. If AT=2'b10, the address doesn't need to be translated and pass through SMMU; if AT=2'b00, the address should be translated by SMMU; if AT=2'b01, a translation request is detected, the PCIe will translate this TLP to DTIM interface and inform SMMU to respond to this request. If the address page table of system has changed, SMMU will send invalidation request to USP through DTIM interface.

10.1.3.8 Reset

PCIe of EIC7700X supports four reset mechanism, such as Power Up Reset、Perst#、Hot reset and FLR(function reset). These resets are come from POR of EIC7700, PCIe slot, Link reset request of DSP and FLR register configuration. Hot reset can be triggered by Secondary_Bus_Bit register configuration of DSP, then the DSP sends some TS1 TLPs to inform USP to Hot reset, after hot reset the PCIe link will repeat to training. DSP also can configure DEVICE_CONTROL_DEVICE_STATUS register bit[15] to trigger a FLR. If PCIe of EIC7700X complete a hot reset or a FLR, it will generate a interrupt to local cpu.

10.1.4 Initialization sequence

- The initialization sequence in RC mode as following diagram

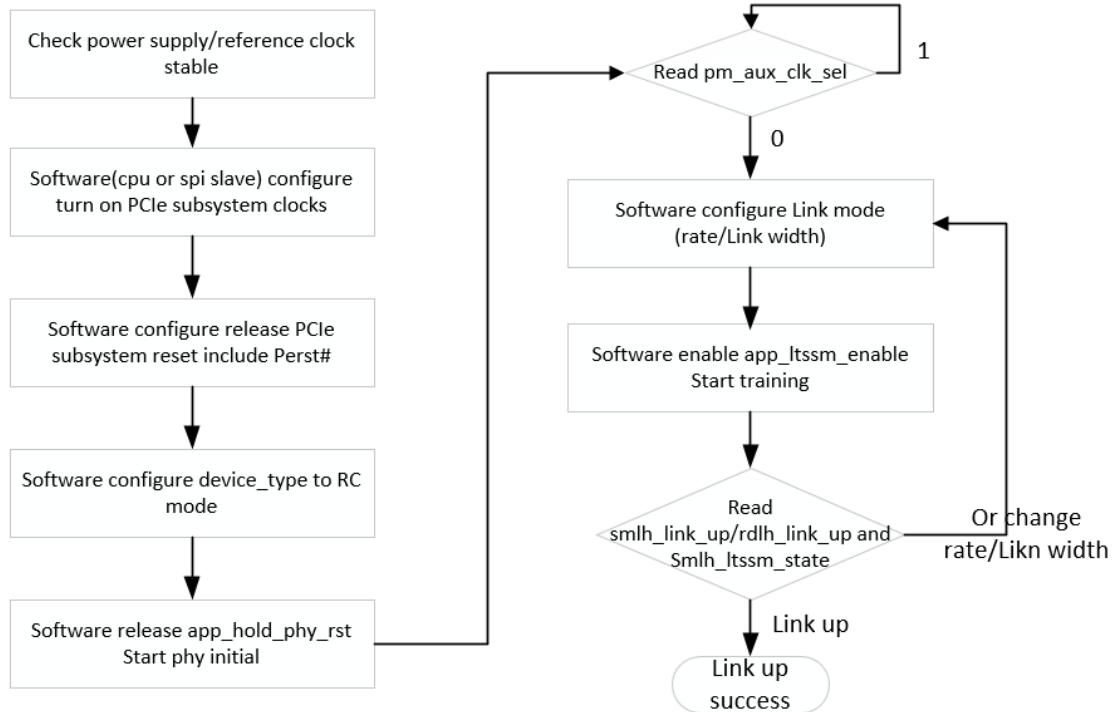


Figure 10-8 Initialization sequence in RC mode

- Above initialization sequence in RC mode
- the initialization sequence in EP mode as following diagram

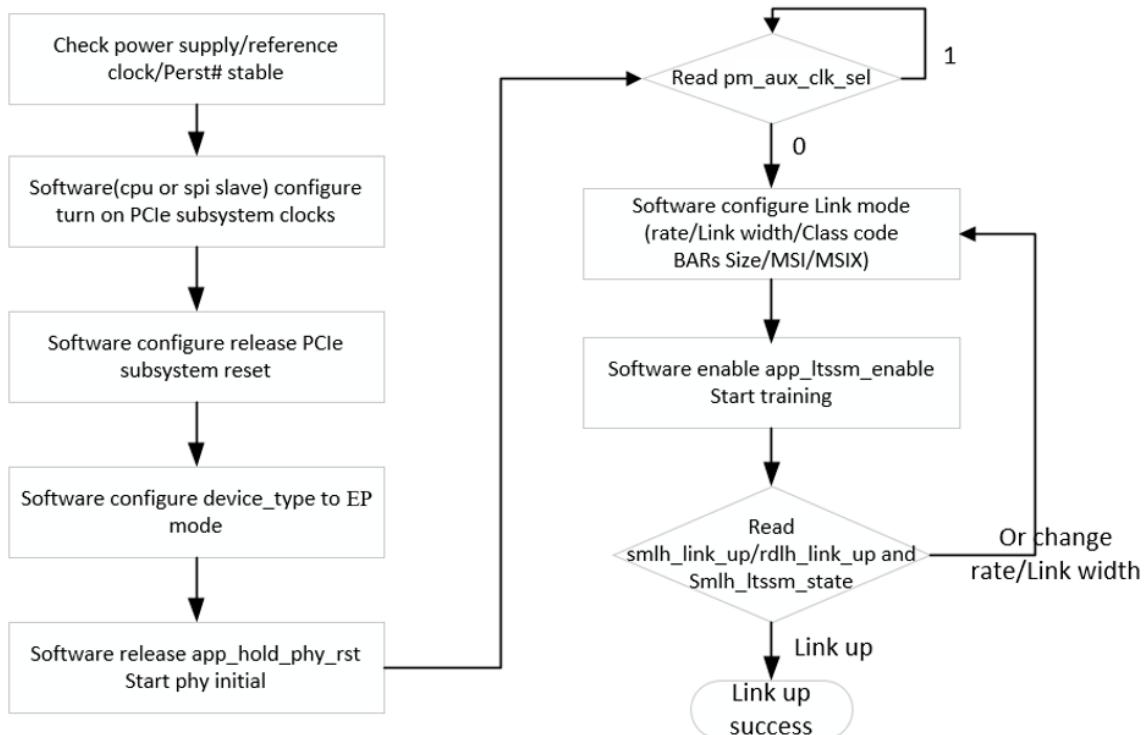


Figure 10-9 Initialization sequence in EP mode

10.1.5 Register description

10.1.5.1 Register list

PCIE Top Csr register list

Block Name	Address Offset	Attributes	Description
PCIE_TOP_CSR/PCIE_CTRL_CFG0	0x0	Address Range: 0x4 Width: 32 Volatile: 0 Endian: little	PCIe subsystem top csr configuration register 0
PCIE_TOP_CSR/PCIE_CTRL_CFG1	0x04	Address Range: 0x4 Width: 32 Volatile: 0 Endian: little	PCIe subsystem top csr configuration register 1
PCIE_TOP_CSR/PCIE_CTRL_CFG11	0x02c	Address Range: 0x4 Width: 32 Volatile: 0 Endian: little	PCIe subsystem top csr configuration register 11
PCIE_TOP_CSR/PCIE_CTRL_CFG12	0x030	Address Range: 0x4 Width: 32 Volatile: 0 Endian: little	PCIe subsystem top csr configuration register 12
PCIE_TOP_CSR/PCIE_CTRL_CFG14	0x034	Address Range: 0x4 Width: 32 Volatile: 0 Endian: little	PCIe subsystem top csr configuration register 14
PCIE_TOP_CSR/PCIE_CTRL_CFG13	0x1000	Address Range: 0x4 Width: 32 Volatile: 0 Endian: little	PCIe subsystem top csr configuration register 13
PCIE_TOP_CSR/PCIE_STATUS0	0x100	Address Range: 0x4 Width: 32 Volatile: 0 Endian: little	PCIe subsystem top csr status register 0

Controller Address Block List

Block Name	Address Offset	Attributes	Description
PCIE_USP/PF0_TYPE0_HDR PCIE_DSP/PF0_TYPE1_HDR	0x0	Address Range: 0x40 Width: 32 Volatile: 0	PF PCI-Compatible Configuration Space Header Type0/Type1

Block Name	Address Offset	Attributes	Description
		Endian: little	
PCIE_USP/PF0_PM_CAP PCIE_DSP/PF0_PM_CAP	0x40	Address Range: 0x8 Width: 32 Volatile: 0 Endian: little	PF PCI Power Management Capability Structure
PCIE_USP/PF0_MSI_CAP PCIE_DSP/PF0_MSI_CAP	0x50	Address Range: 0x18 Width: 32 Volatile: 0 Endian: little	PF MSI Capability Structure
PCIE_USP/PF0_PCIE_CAP PCIE_DSP/PF0_PCIE_CAP	0x70	Address Range: 0x3c Width: 32 Volatile: 0 Endian: little	PF PCI Express Capability Structure
PCIE_USP/PF0_MSIX_CAP PCIE_DSP/PF0_MSIX_CAP	0xb0	Address Range: 0xd Width: 32 Volatile: 0 Endian: little	PF MSI-X Capability Structure
PCIE_USP/PF0_AER_CAP PCIE_DSP/PF0_AER_CAP	0x100	Address Range: 0x48 Width: 32 Volatile: 0 Endian: little	PF Advanced Error Reporting Capability Structure
PCIE_USP/PF0_SPCIE_CAP PCIE_DSP/PF0_SPCIE_CAP	0x148	Address Range: 0x20 Width: 32 Volatile: 0 Endian: little	Secondary PCI Express Capability Structure
PCIE_USP/PF0_ATS_CAP PCIE_DSP/PF0_ATS_CAP	0x168	Address Range: 0x8 Width: 32 Volatile: 0 Endian: little	PF Address Translation Services Capability Structure
PCIE_USP/PF0_PORT_LOGIC PCIE_DSP/PF0_PORT_LOGIC	0x700	Address Range: 0x600 Width: 32 Volatile: 0 Endian: little	Port Logic
PCIE_USP/PF0_TYPE0_HDR_DBI2	Local: 0x100000 Remote: NA	Address Range: 0x40 Width: 32 Volatile: 0 Endian: little	DBI2 Shadow Block: PF PCI-Compatible Configuration Space Header Type0
PCIE_USP/PF0_ATU_CAP PCIE_DSP/PF0_ATU_CAP	Local: 0x300000 Remote:	Address Range: 0x1ff24 Width: 32 Volatile: 0	ATU Unroll Structure

Block Name	Address Offset	Attributes	Description
	BAR1	Endian: little	
PCIE_USP/PF0_HDMA_CAP	Loacal: 0x380000	Address Range: 0x10000 Width: 32	
PCIE_DSP/PF0_HDMA_CAP	Remote: BAR1	Volatile: 0 Endian: little	HDMA Unroll Structure

Registers for Address Block: PF0_TYPE0_HDR_DBI2

Register Name	Address Offset	Memory Access	Attributes	Description
BAR0_MASK_REG	0x10	write-only	DisplayName: BAR\$ Mask Register. Register Size: 32 Value After Reset: 0xffffffff	This register is the mask for BAR\$_REG. If implemented, it exists as a shadow register at the BAR\$_REG address. Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching. The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options. Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.
BAR1_MASK_REG	0x14	write-only	DisplayName: BAR1 Mask	This register is the mask for BAR\$_REG. If implemented, it

Register Name	Address Offset	Memory Access	Attributes	Description
			Register. Register Size: 32 Value After Reset: 0x1fffff	exists as a shadow register at the BAR\$_REG address. Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching. The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options. Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.
BAR2_MASK_REG	0x18	write-only	DisplayName: BAR2 Mask Register. Register Size: 32 Value After Reset: 0x1fffffff	This register is the mask for BAR2_REG. If implemented, it exists as a shadow register at the BAR\$_REG address. Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching.

Register Name	Address Offset	Memory Access	Attributes	Description
				<p>The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options.</p> <p>Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.</p>
BAR3_MASK_REG	0x1c	write-only	<p>DisplayName: BAR3 Mask Register. Register Size: 32 Value After Reset: 0x1ffe</p>	<p>This register is the mask for BAR3_REG. If implemented, it exists as a shadow register at the BAR\$_REG address.</p> <p>Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software. The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching.</p> <p>The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options.</p> <p>Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is</p>

Register Name	Address Offset	Memory Access	Attributes	Description
				accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.
BAR4_MASK_REG	0x20	write-only	<p>DisplayName: BAR4 Mask Register.</p> <p>Register Size: 32</p> <p>Value After Reset: 0x1fffffff</p>	<p>This register is the mask for BAR4_REG. If implemented, it exists as a shadow register at the BAR\$_REG address.</p> <p>Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software.</p> <p>The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching.</p> <p>The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options.</p> <p>Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.</p>
BAR5_MASK_REG	0x24	write-only	<p>DisplayName: BAR5 Mask Register.</p> <p>Register Size: 32</p> <p>Value After Reset: 0x1ffe</p>	<p>This register is the mask for BAR5_REG. If implemented, it exists as a shadow register at the BAR\$_REG address.</p> <p>Normally, the BAR masks are used for indicating the amount of address space that each BAR requests from host software.</p>

Register Name	Address Offset	Memory Access	Attributes	Description
				<p>The BAR masks determine which bits in each BAR are non-writable by host software, which determines the size of the address space claimed by each BAR. The BAR mask values indicate the range of low-order bits, in each implemented BAR, not to use for address matching. The BAR mask value also indicates the range of low-order bits in the BAR that cannot be written from the host. The application can write to all BAR bits to allow setting of memory, I/O, and other standard BAR options.</p> <p>Your local CPU can change the mask at runtime using the DBI. The mask register is invisible to the PCIe wire but visible to your local CPU through the DBI. You cannot read the mask register but you can write to it. It is accessed by asserting dbi_cs2 and dbi_cs. If you only assert dbi_cs then you will access the BAR which is the primary register at that location. Use CS2 instead of dbi_cs2 when you are using the AHB/AXI bridge.</p>
EXP_ROM_BAR_MASK_REG	0x30	write-only	<p>DisplayName: Expansion ROM BAR Mask Register.</p> <p>Register Size: 32</p> <p>Value After Reset: 0x1ffe</p>	<p>This register is the mask for EXP_ROM_BASE_ADDR_REG register. If implemented, it exists as a shadow register at EXP_ROM_BAR_MASK_REG address. The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to this register.</p>

Registers for Address Block: PFO_PORT_LOGIC

Register Name	Address Offset	Memory Access	Attributes	Description
DSP/MSI_CTRL_A DDR_OFF	0x120	read-write	<p>DisplayName: Integrated MSI Reception Module (iMRM) Address Register.</p> <p>Register Size: 32</p> <p>Value After Reset: 0x0</p>	<p>This register holds the integrated MSI reception module address.</p>
DSP/MSI_CTRL_U	0x124	read-	DisplayName: Integrated MSI	This register holds the

Register Name	Address Offset	Memory Access	Attributes	Description
PPER_ADDR_OFF		write	Reception Module Upper Address Register. Register Size: 32 Value After Reset: 0x0	integrated MSI reception module upper address.
DSP/MSI_CTRL_INTERRUPT_0_EN_OFF	0x128	read-write	DisplayName: Integrated MSI Reception Module Interrupt0 Enable Register. Register Size: 32 Value After Reset: 0x0	This register enables integrated MSI reception module interrupt0.
DSP/MSI_CTRL_INTERRUPT_0_MASK_OFF	0x12c	read-write	DisplayName: Integrated MSI Reception Module Interrupt0 Mask Register. Register Size: 32 Value After Reset: 0x0	This register provides information regarding the MSI Interrupt0 mask.
DSP/MSI_CTRL_INTERRUPT_0_STATUS_OF_F	0x130	read-write	DisplayName: Integrated MSI Reception Module Interrupt0 Status Register. Register Size: 32 Value After Reset: 0x0	This register provides the MSI Interrupt0 status.
DSP/MSI_CTRL_INTERRUPT_1_EN_OFF	0x134	read-write	DisplayName: Integrated MSI Reception Module Interrupt1 Enable Register. Register Size: 32 Value After Reset: 0x0	This register enables integrated MSI reception module interrupt1.
DSP/MSI_CTRL_INTERRUPT_1_MASK_OFF	0x138	read-write	DisplayName: Integrated MSI Reception Module Interrupt1 Mask Register. Register Size: 32 Value After Reset: 0x0	This register provides information regarding the MSI Interrupt1 mask.
DSP/MSI_CTRL_INTERRUPT_1_STATUS_OF_F	0x13c	read-write	DisplayName: Integrated MSI Reception Module Interrupt1 Status Register. Register Size: 32 Value After Reset: 0x0	This register provides the MSI Interrupt1 status.
DSP/MSI_CTRL_INTERRUPT_2_EN_OFF	0x140	read-write	DisplayName: Integrated MSI Reception Module Interrupt2 Enable Register. Register Size: 32 Value After Reset: 0x0	This register enables integrated MSI reception module interrupt2.
DSP/MSI_CTRL_INTERRUPT_2_MASK_OFF	0x144	read-write	DisplayName: Integrated MSI Reception Module Interrupt2 Mask Register. Register Size: 32 Value After Reset: 0x0	This register provides information regarding the MSI Interrupt2 mask.
DSP/MSI_CTRL_INTERRUPT_2_STATUS_OF_F	0x148	read-write	DisplayName: Integrated MSI Reception Module Interrupt2 Status Register. Register Size: 32 Value After Reset: 0x0	This register provides the MSI Interrupt2 status.

Register Name	Address Offset	Memory Access	Attributes	Description
DSP/MSI_CTRL_INTERRUPT_3_EN_OFF	0x14c	read-write	DisplayName: Integrated MSI Reception Module Interrupt3 Enable Register. Register Size: 32 Value After Reset: 0x0	This register enables integrated MSI reception module interrupt3.
DSP/MSI_CTRL_INTERRUPT_3_MASK_OFF	0x150	read-write	DisplayName: Integrated MSI Reception Module Interrupt3 Mask Register. Register Size: 32 Value After Reset: 0x0	This register provides information regarding the MSI Interrupt3 mask.
DSP/MSI_CTRL_INTERRUPT_3_STATUS_OF_F	0x154	read-write	DisplayName: Integrated MSI Reception Module Interrupt3 Status Register. Register Size: 32 Value After Reset: 0x0	This register provides the MSI Interrupt3 status.
DSP/MSI_CTRL_INTERRUPT_4_EN_OFF	0x158	read-write	DisplayName: Integrated MSI Reception Module Interrupt4 Enable Register. Register Size: 32 Value After Reset: 0x0	This register enables integrated MSI reception module interrupt4.
DSP/MSI_CTRL_INTERRUPT_4_MASK_OFF	0x15c	read-write	DisplayName: Integrated MSI Reception Module Interrupt4 Mask Register. Register Size: 32 Value After Reset: 0x0	This register provides information regarding the MSI Interrupt4 mask.
DSP/MSI_CTRL_INTERRUPT_4_STATUS_OF_F	0x160	read-write	DisplayName: Integrated MSI Reception Module Interrupt4 Status Register. Register Size: 32 Value After Reset: 0x0	This register provides the MSI Interrupt4 status.
DSP/MSI_CTRL_INTERRUPT_5_EN_OFF	0x164	read-write	DisplayName: Integrated MSI Reception Module Interrupt5 Enable Register. Register Size: 32 Value After Reset: 0x0	This register enables integrated MSI reception module interrupt5.
DSP/MSI_CTRL_INTERRUPT_5_MASK_OFF	0x168	read-write	DisplayName: Integrated MSI Reception Module Interrupt5 Mask Register. Register Size: 32 Value After Reset: 0x0	This register provides information regarding the MSI Interrupt5 mask.
DSP/MSI_CTRL_INTERRUPT_5_STATUS_OF_F	0x16c	read-write	DisplayName: Integrated MSI Reception Module Interrupt5 Status Register. Register Size: 32 Value After Reset: 0x0	This register provides the MSI Interrupt5 status.
DSP/MSI_CTRL_INTERRUPT_6_EN_OFF	0x170	read-write	DisplayName: Integrated MSI Reception Module Interrupt6 Enable Register. Register Size: 32 Value After Reset: 0x0	This register enables integrated MSI reception module interrupt6.

Register Name	Address Offset	Memory Access	Attributes	Description
DSP/MSI_CTRL_INTERRUPT_6_MASK_OFF	0x174	read-write	DisplayName: Integrated MSI Reception Module Interrupt6 Mask Register. Register Size: 32 Value After Reset: 0x0	This register provides information regarding the MSI Interrupt6 mask.
DSP/MSI_CTRL_INTERRUPT_6_STATUS_OF_F	0x178	read-write	DisplayName: Integrated MSI Reception Module Interrupt6 Status Register. Register Size: 32 Value After Reset: 0x0	This register provides the MSI Interrupt6 status.
DSP/MSI_CTRL_INTERRUPT_7_EN_OFF	0x17c	read-write	DisplayName: Integrated MSI Reception Module Interrupt7 Enable Register. Register Size: 32 Value After Reset: 0x0	This register enables integrated MSI reception module interrupt7.
DSP/MSI_CTRL_INTERRUPT_7_MASK_OFF	0x180	read-write	DisplayName: Integrated MSI Reception Module Interrupt7 Mask Register. Register Size: 32 Value After Reset: 0x0	This register provides information regarding the MSI Interrupt7 mask.
DSP/MSI_CTRL_INTERRUPT_7_STATUS_OF_F	0x184	read-write	DisplayName: Integrated MSI Reception Module Interrupt7 Status Register. Register Size: 32 Value After Reset: 0x0	This register provides the MSI Interrupt7 status.
DSP/MSI_GPIO_IO_OFF	0x188	read-write	DisplayName: Integrated MSI Reception Module General Purpose IO Register. Register Size: 32 Value After Reset: 0x0	The contents of this register drive the top-level GPIO msi_ctrl_io[31:0].
MSIX_ADDRESS_MATCH_LOW_OFF	0x240	read-write	DisplayName: MSI-X Address Match Low Register. Register Size: 32 Value After Reset: 0x0	When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more information, see the Interrupts section in the "Controller Operations" chapter of the Databook. This register is only used in AXI configurations. When your local AXI application writes (MWr) to the address defined in this register (and

Register Name	Address Offset	Memory Access	Attributes	Description
				MSIX_ADDRESS_MATCH _HIGH_OFFSET), the controller will load the MSIX_DOORBELL_OFFSET register with the contents of the MWr and subsequently create and send MSI-X TLPs
MSIX_ADDRESS_MATCH_HIGH_OF F	0x244	read-write	DisplayName: MSIX Address Match High Register. Register Size: 32 Value After Reset: 0x0	MSI-X Address Match High Register. When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more information, see the Interrupts section in the "Controller Operations" chapter of the Databook. This register is only used in AXI configurations. When your local AXI application writes (MWr) to the address defined in this register (and MSIX_ADDRESS_MATCH _LOW_OFFSET), the controller will load the MSIX_DOORBELL_OFFSET register with the contents of the MWr and subsequently create and send MSI-X TLPs
MSIX_DOORBELL_OFFSET	0x248	write-only	DisplayName: MSI-X Doorbell Register. Register Size: 32 Value After Reset: 0x0	When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more information, see the Interrupts section in the "Controller Operations" chapter of the Databook. - For AXI configurations: when your local application writes (MWr) to the address defined in

Register Name	Address Offset	Memory Access	Attributes	Description
				MSIX_ADDRESS_MATCH _LOW_OFF, the controller will load this register with the contents of the MWr and subsequently create and send MSI-X TLPs. - For non-AMBA configurations: when your local application writes to this register, the controller will create and send MSI-X TLPs.
MSIX_RAM_CTRL_OFF	0x24c	read-write	DisplayName: MSI-X RAM power mode and debug control register. Register Size: 32 Value After Reset: 0x0	When you enable the MSI-X Table RAM feature (MSIX_TABLE_EN=1), the controller implements the logic and RAM required to generate MSI-X requests. For more information, see the Interrupts section in the "Controller Operations" chapter of the Databook.

Registers for Address Block: PFO_ATU_CAP

Register Name	Address Offset	Memory Access	Attributes	Description
IATU_REGION_CTRL_1_OFFSET_OUTBOUND_i	0x200*i+0x0	read-write	DisplayName: iATU Region Control 1 Register. Register Size: 32	This register controls the iATU outbound region access based on the optional iATU outbound features enabled using iATU Region Control 2 Register.
IATU_REGION_CTRL_2_OFFSET_OUTBOUND_i	0x200*i+0x4	read-write	DisplayName: iATU Region Control 2 Register. Register Size: 32	Using this register you can enable/disable the outbound iATU optional features.
IATU_LWR_BASE_ADDRESS_OFFSET_OUTBOUND_i	0x200*i+0x8	read-write	DisplayName: iATU Lower Base Address Register. Register Size: 32 Value After Reset: 0x0	The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB

Register Name	Address Offset	Memory Access	Attributes	Description
				boundaries. More precisely, the lower log2(CX_ATU_MIN_REGION_SIZE) bits are zero.
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_i	0x200*i+0xc	read-write	DisplayName: iATU Upper Base Address Register. Register Size: 32 Value After Reset: 0x0	This register holds the upper 32-bits of the start (and end) address of the address region to be translated.
IATU_LIMIT_ADDR_OFF_OUTBOUND_i	0x200*i+0x10	read-write	DisplayName: iATU Limit Address Register. Register Size: 32 Value After Reset: 0xffff	This register holds the end address of the address region to be translated.
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i	0x200*i+0x14	read-write	DisplayName: iATU Lower Target Address Register. Register Size: 32 Value After Reset: 0x0	This register holds the Lower Target part of the new address of the translated region, or the outbound TLP header information, depending on the setting of HEADER_SUBSTITUTE_EN field of IATU_REGION_CTRL_2_VIEWPORT_OFFSET_OUTBOUND_i register.
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_i	0x200*i+0x18	read-write	DisplayName: iATU Upper Target Address Register. Register Size: 32 Value After Reset: 0x0	This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.
IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_i	0x200*i+0x20	read-write	DisplayName: iATU Upper Limit Address Register. Register Size: 32 Value After Reset: 0x0	The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0->32, 8=1TB) specifies the maximum region size of an address translation region. This register is only used when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFFSET_OUTBOUND_i is '1'.
IATU_REGION_CTRL_1_OFFSET_INBOUND_i	0x200*i+0x100	read-write	DisplayName: iATU Region Control 1 Register. Register Size: 32	This register controls the iATU inbound region access based on the optional iATU inbound features enabled using iATU Region Control 2 Register.
IATU_REGION_CTRL_2_OFFSET_INBOUND_i	0x200*i+0x104	read-write	DisplayName: iATU Region	Using this register you can enable/disable the inbound iATU

Register Name	Address Offset	Memory Access	Attributes	Description
			Control 2 Register. Register Size: 32	optional features.
IATU_LWR_BASE_A DDR_OFF_INBOUND_i	0x200*i+ 0x108	read-write	DisplayName: iATU Lower Base Address Register. Register Size: 32 Value After Reset: 0x0	The CX_ATU_MIN_REGION_SIZE configuration parameter (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower $\log_2(\text{CX_ATU_MIN_REGION_SIZE})$ bits are zero.
IATU_UPPER_BASE_ ADDR_OFF_INBOUN D_i	0x200*i+ 0x10c	read-write	DisplayName: iATU Upper Base Address Register. Register Size: 32 Value After Reset: 0x0	This register holds the upper 32 bits of the start (and end) address of the address region to be translated.
IATU_LIMIT_ADDR_ OFF_INBOUND_i	0x200*i+ 0x110	read-write	DisplayName: iATU Limit Address Register. Register Size: 32 Value After Reset: 0xff0	This register holds the end address of the address region to be translated.
IATU_LWR_TARGET _ADDR_OFF_INBOU ND_i	0x200*i+ 0x114	read-write	DisplayName: iATU Lower Target Address Register. Register Size: 32 Value After Reset: 0x0	This register holds the Lower Target part of the new address of the translated region.
IATU_UPPER_TARG ET_ADDR_OFF_INB OUND_i	0x200*i+ 0x118	read-write	DisplayName: iATU Upper Target Address Register. Register Size: 32 Value After Reset: 0x0	This register holds the upper 32 bits of the start address (Upper Target part) of the new address of the translated region.
IATU_UPPR_LIMIT_ ADDR_OFF_INBOUN D_i	0x200*i+ 0x120	read-write	DisplayName: iATU Upper Limit Address Register. Register Size: 32 Value After Reset: 0x0	The CX_MAX_ATU_REGION_SIZE configuration parameter (Value Range: 0(4GB) to 32(16 EB)) specifies the maximum region size of an address translation region. This register is only used

Register Name	Address Offset	Memory Access	Attributes	Description
				when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_INBOUND_i is '1'.

Registers for Address Block: PF0_HDMA_CAP

Register Name	Address Offset	Memory Access	Attributes	Description
HDMA_EN_OFF_WRCH_i	0x200*i +0x0	read-write	DisplayName: HDMA Write Channel Enable Register. Register Size: 32	This register enables an HDMA write channel.
HDMA_DOORBELL_OF_F_WRCH_i	0x200*i +0x4	write-only	DisplayName: HDMA Write Channel Doorbell Register. Register Size: 32	This register controls the Doorbell state of an HDMA write channel.
HDMA_ELEM_PF_OFF_WRCH_i	0x200*i +0x8	read-write	DisplayName: HDMA Write Channel Prefetch Register. Register Size: 32	This register holds information regarding descriptor prefetch of an HDMA write channel.
HDMA_LL_P_LOW_OFF_WRCH_i	0x200*i +0x10	read-write	DisplayName: HDMA Write Channel Linked List Pointer Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the address of a write channel transfer list. It is used in linked list mode only.
HDMA_LL_P_HIGH_OFF_WRCH_i	0x200*i +0x14	read-write	DisplayName: HDMA Write Channel Linked List Pointer High Register. Register Size: 32 Value After Reset: 0x0	This register holds the higher 32 bits of the address of a write channel transfer list. It is used in linked list mode only.
HDMA_CYCLE_OFF_W_RCH_i	0x200*i +0x18	read-write	DisplayName: HDMA Write Channel Producer-	This register is used to synchronize the producer (software) and the consumer (HDMA) (used in linked list mode)

Register Name	Address Offset	Memory Access	Attributes	Description
			Consumer Cycle Synchronization Register. Register Size: 32	only). For more information, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization" in the <i>Databook</i> . <i>Note:</i> The HDMA updates this register while operating on a linked list.
HDMA_XFERSIZE_OFF _WRCH_i	0x200*i +0x1c	read-write	DisplayName: HDMA Write Channel Transfer Size Register. Register Size: 32 Value After Reset: 0x0	This register holds the transfer size of an HDMA channel.
HDMA_SAR_LOW_OFF _WRCH_i	0x200*i +0x20	read-write	DisplayName: HDMA Write Channel SAR Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Source Address Register (SAR) of an HDMA write channel.
HDMA_SAR_HIGH_OFF _WRCH_i	0x200*i +0x24	read-write	DisplayName: HDMA Write Channel SAR High Register. Register Size: 32 Value After Reset: 0x0	This register holds the higher 32 bits of the Source Address Register (SAR) of an HDMA write channel.
HDMA_DAR_LOW_OFF _WRCH_i	0x200*i +0x28	read-write	DisplayName: HDMA Write Channel DAR Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Destination Address Register (DAR) of an HDMA write channel.
HDMA_DAR_HIGH_OFF _WRCH_i	0x200*i +02c	read-write	DisplayName: HDMA Write Channel DAR High Register. Register Size: 32 Value After Reset: 0x0	This register holds the higher 32 bits of the Destination Address Register (DAR) of an HDMA write channel.
HDMA_WATERMARK_E N_OFF_WRCH_i	0x200*i +0x30	read-write	DisplayName: HDMA Write Channel Linked-list Watermark Enable Register.	This register controls the watermark interrupts generated after processing a watermarked LL element. HDMA updates this field as LL elements are

Register Name	Address Offset	Memory Access	Attributes	Description
			Register Size: 32	processed.
HDMA_CONTROL1_OF_F_WRCH_i	0x200*i +0x34	read-write	DisplayName: HDMA Write Channel Control Settings 1 Register. Register Size: 32	This register controls the operation of the HDMA write channel.
HDMA_FUNC_NUM_OF_F_WRCH_i	0x200*i +0x38	read-write	DisplayName: HDMA Write Channel Function Number Register. Register Size: 32	This register controls HDMA write channel to Physical/Virtual function mapping.
HDMA_QOS_OFF_WRC_H_i	0x200*i +0x3c	read-write	DisplayName: HDMA Write Channel QoS Settings Register. Register Size: 32	This register controls the HDMA write channel QoS features.
HDMA_STATUS_OFF_WRCH_i	0x200*i +0x80	read-only	DisplayName: HDMA Write Channel Status Register. Register Size: 32	This register specifies the current operational state of a channel.
HDMA_INT_STATUS_OFF_WRCH_i	0x200*i +0x84	read-only	DisplayName: HDMA Write Channel Interrupt Status Register. Register Size: 32	This register provides information regarding the HDMA write channel interrupt status.
HDMA_INT_SETUP_OF_F_WRCH_i	0x200*i +0x88	read-write	DisplayName: HDMA Write Channel Interrupt Setup Register. Register Size: 32	This register is used to configure interrupts for a write channel.
HDMA_INT_CLEAR_OF_F_WRCH_i	0x200*i +0x8c	write-only	DisplayName: HDMA Write Channel Interrupt Clear Register. Register Size: 32	This register indicates interrupt clear status of a write channel. It is a self-clearing register. Reads to this register returns '0'.

Register Name	Address Offset	Memory Access	Attributes	Description
HDMA_MSI_STOP_LOW_OFFSET_WRCH_i	0x200*i +0x90	read-write	DisplayName: HDMA Read Stop Remote Interrupt Address Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Stop Interrupt MWr TLP address.
HDMA_MSI_STOP_HIGH_OFFSET_WRCH_i	0x200*i +0x94	read-write	DisplayName: HDMA Read Stop Remote Interrupt Address High Register. Register Size: 32 Value After Reset: 0x0	This register holds the higher 32 bits of the Stop Interrupt MWr TLP address.
HDMA_MSI_WATERMARK_LOW_OFFSET_WRCH_i	0x200*i +0x98	read-write	DisplayName: HDMA Read Watermark Remote Interrupt Address Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Watermark Interrupt MWr TLP address.
HDMA_MSI_WATERMARK_HIGH_OFFSET_WRCH_i	0x200*i +0x9c	read-write	DisplayName: HDMA Read Watermark Remote Interrupt Address High Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Watermark Interrupt MWr TLP address.
HDMA_MSI_ABORT_LOW_OFFSET_WRCH_i	0x200*i +0xa0	read-write	DisplayName: HDMA Read Abort Remote Interrupt Address Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Abort Interrupt MWr TLP address.
HDMA_MSI_ABORT_HIGH_OFFSET_WRCH_i	0x200*i +0xa4	read-write	DisplayName: HDMA Read	This register holds the lower 32 bits of the Abort Interrupt MWr

Register Name	Address Offset	Memory Access	Attributes	Description
			Abort Remote Interrupt Address High Register. Register Size: 32 Value After Reset: 0x0	TLP address.
HDMA_MSI_MSGD_OF_F_WRCH_i	0x200*i +0xa8	read-write	DisplayName: HDMA Write Channel Remote Interrupt Data Register. Register Size: 32	This register holds the IMWr TLP Data of an HDMA write channel.
HDMA_EN_OFF_RDCH_i	0x200*i +0x100	read-write	DisplayName: HDMA Read Channel Enable. Register Size: 32	This register enables an HDMA read channel.
HDMA_DOORBELL_OF_F_RDCH_i	0x200*i +0x104	write-only	DisplayName: HDMA Read Channel Doorbell Register. Register Size: 32	This register controls the Doorbell state of an HDMA read channel.
HDMA_ELEM_PF_OFF_RDCH_i	0x200*i +0x108	read-write	DisplayName: HDMA Read Channel Prefetch Register. Register Size: 32	This register holds information regarding descriptor prefetch of an HDMA read channel.
HDMA_LL_P_LOW_OFF_RDCH_i	0x200*i +0x110	read-write	DisplayName: HDMA Read Channel Linked List Pointer Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the address of a read channel transfer list. It is used in linked list mode only.
HDMA_LL_P_HIGH_OFF_RDCH_i	0x200*i +0x114	read-write	DisplayName: HDMA Read Channel Linked List Pointer High Register. Register Size: 32 Value After Reset: 0x0	This register holds the higher 32 bits of the address of a read channel transfer list. It is used in linked list mode only.
HDMA_CYCLE_OFF_RD	0x200*i	read-	DisplayName:	This register is used to

Register Name	Address Offset	Memory Access	Attributes	Description
CH_i	+0x118	write	HDMA Read Channel Producer-Consumer Cycle Synchronization Register. Register Size: 32	synchronize the producer (software) and the consumer (HDMA) (used in linked list mode only). For more information, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization" in the <i>Databook</i> . <i>Note:</i> The HDMA updates this register while operating on a linked list.
HDMA_XFERSIZE_OFF _RDCH_i	0x200*i +0x11c	read-write	DisplayName: HDMA Read Channel Transfer Size Register. Register Size: 32 Value After Reset: 0x0	This register holds the transfer size of an HDMA channel.
HDMA_SAR_LOW_OFF _RDCH_i	0x200*i +0x120	read-write	DisplayName: HDMA Read Channel SAR Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Source Address Register (SAR) of an HDMA read channel.
HDMA_SAR_HIGH_OFF _RDCH_i	0x200*i +0x124	read-write	DisplayName: HDMA Read Channel SAR High Register. Register Size: 32 Value After Reset: 0x0	This register holds the higher 32 bits of the Source Address Register (SAR) of an HDMA read channel.
HDMA_DAR_LOW_OFF _RDCH_i	0x200*i +0x128	read-write	DisplayName: HDMA Read Channel DAR Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Destination Address Register (DAR) of an HDMA read channel.
HDMA_DAR_HIGH_OFF _RDCH_i	0x200*i +0x12c	read-write	DisplayName: HDMA Read Channel DAR High Register. Register Size: 32 Value After Reset: 0x0	This register holds the higher 32 bits of the Destination Address Register (DAR) of an HDMA read channel.
HDMA_WATERMARK_E	0x200*i	read-	DisplayName: HDMA Read	This register controls the watermark interrupts generated

Register Name	Address Offset	Memory Access	Attributes	Description
N_OFF_RDCH_i	+0x130	write	Channel Linked-list Watermark Enable Register. Register Size: 32	after processing a watermarked LL element. HDMA updates this field as LL elements are processed.
HDMA_CONTROL1_OF_F_RDCH_i	0x200*i +0x134	read-write	DisplayName: HDMA Read Channel Control Settings 1 Register. Register Size: 32	This register controls the operation of the HDMA read channel.
HDMA_FUNC_NUM_OF_F_RDCH_i	0x200*i +0x138	read-write	DisplayName: HDMA Read Channel Function Number Register. Register Size: 32	This register controls HDMA read channel to Physical/Virtual function mapping.
HDMA_QOS_OFF_RDC_H_i	0x200*i +0x13c	read-write	DisplayName: HDMA Read Channel QoS Settings Register. Register Size: 32	This register controls the HDMA read channel QoS features.
HDMA_STATUS_OFF_RDCH_i	0x200*i +0x180	read-only	DisplayName: HDMA Read Channel Status Register. Register Size: 32	This register specifies the current operational state of a channel.
HDMA_INT_STATUS_OFFSET_RDCH_i	0x200*i +0x184	read-only	DisplayName: HDMA Read Channel Interrupt Status Register. Register Size: 32	This register provides information regarding the HDMA read channel interrupt status.
HDMA_INT_SETUP_OF_F_RDCH_i	0x200*i +0x188	read-write	DisplayName: HDMA Read Channel Interrupt Setup Register. Register Size: 32	This register is used to configure interrupts for a read channel.
HDMA_INT_CLEAR_OF_F_RDCH_i	0x200*i +0x18c	write-only	DisplayName: HDMA Read Channel Interrupt Clear	This register indicates interrupt clear status of a read channel. It is a self-clearing register. Reads to this register returns '0'.

Register Name	Address Offset	Memory Access	Attributes	Description
			Register. Register Size: 32	
HDMA_MSI_STOP_LO_W_OFF_RDCH_i	0x200*i +0x190	read-write	DisplayName: HDMA Read Stop Remote Interrupt Address Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Stop Interrupt MWr TLP address.
HDMA_MSI_STOP_HIGH_OFF_RDCH_i	0x200*i +0x194	read-write	DisplayName: HDMA Read Stop Remote Interrupt Address High Register. Register Size: 32 Value After Reset: 0x0	This register holds the higher 32 bits of the Stop Interrupt MWr TLP address.
HDMA_MSI_WATERMARK_LOW_OFF_RDCH_i	0x200*i +0x198	read-write	DisplayName: HDMA Read Watermark Remote Interrupt Address Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Watermark Interrupt MWr TLP address.
HDMA_MSI_WATERMARK_HIGH_OFF_RDCH_i	0x200*i +0x19c	read-write	DisplayName: HDMA Read Watermark Remote Interrupt Address High Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Watermark Interrupt MWr TLP address.
HDMA_MSI_ABORT_LO_W_OFF_RDCH_i	0x200*i +0x1a0	read-write	DisplayName: HDMA Read Abort Remote Interrupt Address Low Register. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Abort Interrupt MWr TLP address.

Register Name	Address Offset	Memory Access	Attributes	Description
HDMA_MSI_ABORT_HI_GH_OFF_RDCH_i	0x200*i +0x1a4	read-write	DisplayName: HDMA Read Abort Remote Interrupt Address High. Register Size: 32 Value After Reset: 0x0	This register holds the lower 32 bits of the Abort Interrupt mwr TLP address.
HDMA_MSI_MSGD_OF_F_RDCH_i	0x200*i +0x1a8	read-write	DisplayName: HDMA Read Channel Remote Interrupt Data Register. Register Size: 32	This register holds the IMWr TLP Data of an HDMA read channel.

10.1.5.2 Register detail

PF0_TYPE0_HDR/PF0_TYPE1_HDR/PF0_PM_CAP/PF_MSI_CAP//PF_PCIE_CAP/PF_MSIX_CAP/ PF_AER_CAP/ PF_SPCIE_CAP/ PF_ATS_CAP register detail please reference to PCI-Express Specification.

PCIE_TOP_CSR/PCIE_CTRL_CFG0

Field Name	Bit Offset	Memory Access	Attributes	Description
apps_pm_xmt_turnoff	31	read-write	Register Size: 1 Value After Reset: 0x0	Request from your application to generate a PM_Turn_Off message
app_unlock_msg	30	read-write	Register Size: 1 Value After Reset: 0x0	Request from your application to generate an Unlock message
sys_eml_interlock_engaged	29	read-write	Register Size: 1 Value After Reset: 0x0	System Electromechanical Interlock Engaged
sys_cmd_cpled_int	28	read-write	Register Size: 1 Value After Reset: 0x0	Command completed Interrupt. Indicates that the Hot-Plug controller completed a command
sys_pre_det_chged	27	read-write	Register Size: 1 Value After Reset: 0x0	Presence Detect Changed. Indicates that the state of card present detector has changed
sys_mrl_sensor_chged	26	read-write	Register Size: 1 Value After Reset: 0x0	MRL Sensor Changed. Indicates that the state of MRL sensor has Changed
sys_pwr_fault_det	25	read-write	Register Size: 1 Value After Reset:	Power Fault Detected. Indicates the power controller

Field Name	Bit Offset	Memory Access	Attributes	Description
			0x0	detected a power fault at this slot.
sys_mrl_sensor_state	24	read-write	Register Size: 1 Value After Reset: 0x0	MRL Sensor State. Indicates the state of the manually-operated retention latch (MRL) sensor
sys_pre_det_state	23	read-write	Register Size: 1 Value After Reset: 0x0	Presence Detect State. Indicates whether or not a card is present in the slot:
sys_atten_button_pressed	22	read-write	Register Size: 1 Value After Reset: 0x0	Attention Button Pressed. Indicates that the system attention button has been pressed
apps_pm_xmt_pme	21	read-write	Register Size: 1 Value After Reset: 0x0	Wake Up
tx_lane_flip_en	20	read-write	Register Size: 1 Value After Reset: 0x0	Performs manual lane reversal for transmit lanes
rx_lane_flip_en	19	read-write	Register Size: 1 Value After Reset: 0x0	Performs manual lane reversal for receive lanes
dbg_pba	18	read-write	Register Size: 1 Value After Reset: 0x0	MSIX PBA RAM Debug Mode
dbg_table	17	read-write	Register Size: 1 Value After Reset: 0x0	MSIX Table RAM Debug Mode
app_dbi_ro_wr_disable	16	read-write	Register Size: 1 Value After Reset: 0x0	DBI Read-only Write Disable
app_pf_req_retry_en	15	read-write	Register Size: 1 Value After Reset: 0x0	Provides a per Physical Function (PF) capability to defer incoming configuration requests until initialization is complete
app_req_retry_en	14	read-write	Register Size: 1 Value After Reset: 0x0	Provides a capability to defer incoming configuration requests until initialization is complete
app_xfer_pending	13	read-write	Register Size: 1 Value After Reset: 0x0	Indicates that your application has transfers pending and prevents the controller from entering L1
app_req_exit_l1	12	read-write	Register Size: 1 Value After Reset: 0x0	Application request to Exit L1

Field Name	Bit Offset	Memory Access	Attributes	Description
app_ready_entr_l23	11	read-write	Register Size: 1 Value After Reset: 0x0	Application Ready to Enter L23
app_req_entr_l1	10	read-write	Register Size: 1 Value After Reset: 0x0	Application request to Enter L1 ASPM state
app_init_RST	9	read-write	Register Size: 1 Value After Reset: 0x0	Request from your application to send a hot reset to the upstream port
app_clk_pm_en	8	read-write	Register Size: 1 Value After Reset: 0x0	Clock PM feature enabled by application
app_clk_req_n	7	read-write	Register Size: 1 Value After Reset: 0x1	Indicates that the application logic is ready to have reference clock Removed,active low
app_hold_phy_RST	6	read-write	Register Size: 1 Value After Reset: 0x1	Set this signal to one before the de-assertion of power on reset to hold the PHY in reset. This can be used to configure your PHY
app_ltssm_enable	5	read-write	Register Size: 1 Value After Reset: 0x0	allow the LTSSM to continue link establishment.
sys_aux_pwr_det	4	read-write	Register Size: 1 Value After Reset: 0x0	Auxiliary Power Detected. Used to report to the host software that auxiliary power (Vaux) is present , active high
device_type	0	read-write	Register Size: 4 Value After Reset: 0x0	4'b0000 : ep ; 4'b0001 : legacy ep ; 4'b0100 : RC

PCIE_TOP_CSR/PCIE_CTRL_CFG1

Field Name	Bit Offset	Memory Access	Attributes	Description
reserved	28	Read-only	NA	
phy0_cr_para_sel	27	read-write	Register Size: 1 Value After Reset: 0x0	Phy configure interface enable
pciet_sys_int	26	read-write	Register Size: 1 Value After Reset: 0x0	INTx interrupt trigger
refclk_req_n_en	25	read-	Register Size: 1 Value After Reset:	Reference clock request

Field Name	Bit Offset	Memory Access	Attributes	Description
		write	0x0	enable
app_bus_num	17	read-write	Register Size: 8 Value After Reset: 0x0	Local PCIe bus number input
app_dev_num	12	read-write	Register Size: 5 Value After Reset: 0x0	Local PCIe device number input
app_l1sub_disable	11	read-write	Register Size: 1 Value After Reset: 0x0	L1 sub low power disable
restore_state_ack	10	read-write	Register Size: 1 Value After Reset: 0x0	Acknowledges that saved state has been restored after pwr-up
save_state_ack	9	read-write	Register Size: 1 Value After Reset: 0x0	Acknowledges that state retention task is completed
app_l1_pwr_off_en	8	read-write	Register Size: 1 Value After Reset: 0x0	L1 low power off enable
outband_pwrup_cmd	6	read-write	Register Size: 2 Value After Reset: 0x0	Causes the controller to wake up from L1
app_ras_des_tba_ctrl	4	read-write	Register Size: 2 Value After Reset: 0x0	NA
app_ras_des_sd_hold_ltssm	3	read-write	Register Size: 1 Value After Reset: 0x0 Register Size: 1 Value After Reset: 0x0	Hold ltssm state enable
diag_ctrl_bus	0	read-write	Register Size: 3 Value After Reset: 0x0	Controller internal debug select

PCIE_TOP_CSR/PCIE_CTRL_CFG11

Field Name	Bit Offset	Memory Access	Attributes	Description
Cfg_int_priority	0	read-write	Register Size: 31 Value After Reset: 0x0	Local cpu 31 groups MSI interrupt priority

PCIE_TOP_CSR/PCIE_CTRL_CFG12

Field Name	Bit Offset	Memory Access	Attributes	Description
Cfg_int_mask	0	read-write	Register Size: 32 Value After Reset: 0x0	Local cpu 31 groups MSI interrupt mask

PCIE_TOP_CSR/PCIE_CTRL_CFG14

Field Name	Bit Offset	Memory Access	Attributes	Description
Cfg_bar0_base_addr	0	read-write	Register Size: 31 Value After Reset: 0x5000_0000	Base address of Bar0 pass through APB master configuration

PCIE_TOP_CSR/PCIE_CTRL_CFG13

Field Name	Bit Offset	Memory Access	Attributes	Description
Pcie_qos_sel	8	read-write	Register Size: 1 Value After Reset: 0x0	Pcie axi master qos configuration: 0: axi master qos generated by controller 1: axi master qos generated by register cfg
Pcie_arqos	4	read-write	Register Size: 4 Value After Reset: 0x0	Axi master arqos configuration
Pcie_awqos	0	read-write	Register Size: 4 Value After Reset: 0x0	Axi master awqos configuration

PCIE_TOP_CSR/PCIE_CTRL_CFG13

Field Name	Bit Offset	Memory Access	Attributes	Description
Cfg_bar0_base_addr	0	read-write	Register Size: 31 Value After Reset: 0x0	Base address of Bar0 pass through APB master configuration

PCIE_TOP_CSR/PCIE_STATUS0

Field Name	Bit Offset	Memory Access	Attributes	Description
reserved	17	Read-only	NA	

Field Name	Bit Offset	Memory Access	Attributes	Description
pm_sel_aux_clk	16	Read-only	Register Size: 1 Value After Reset: 0x1	When convert to 0 indicate the phy initial successfully and the aux_clk convert to pipe_clock
phy0_sram_init_done	15	Read-only	Register Size: 1 Value After Reset: 0x0	Phy sram initial done
training_RST_N	14	Read-only	Register Size: 1 Value After Reset: 0x0	Hot reset indication
link_req_RST_NOT	13	Read-only	Register Size: 1 Value After Reset: 0x0	Link reset indication, low pulse when receive a hot reset request
smlh_req_RST_NOT	12	Read-only	Register Size: 1 Value After Reset: 0x0	phy reset indication, low pulse when receive a hot reset request
smlh_ltssm_state_rcvry_eq	11	Read-only	Register Size: 1 Value After Reset: 0x0	Indicate ltssm state is in rcvry_eq state
pm_curnt_state	8	Read-only	Register Size: 3 Value After Reset: 0x0	Current low power state
smlh_ltssm_state	2	Read-only	Register Size: 6 Value After Reset: 0x0	Link ltssm state indication
smlh_link_up	1	Read-only	Register Size: 1 Value After Reset: 0x0	Phy link up indication
rdlh_link_up	0	Read-only	Register Size: 1 Value After Reset: 0x0	Data link layer link up indication

PF0_TYPE0_HDR_DBI2/BAR0_MASK_REG

Field Name	Bit Offset	Memory Access	Attributes	Description
PCI_TYPE0_BAR0_ENABLED	0	write-only	Register Size: 1 Value After Reset: BAR0_ENABLED_n Testable: untestable	- BAR0 Mask Enabled. - The name and the function of this field depends on the value of CX_CXL_ENABLE parameter as follows: 1: RCRB MEMBAR0[0]. - The MEMBAR0 register field is implemented in Type0 and Type1 Config Header capability when target protocol CXL is enabled and when the memorymap region is RCRB. - This is 32bits field, we use this to write MEMBAR0 address into RCRB space. MEMBAR0 registers should be able to access BAR0_REG

Field Name	Bit Offset	Memory Access	Attributes	Description
				offset 10h of PCIE memory-mapped space. 0: BAR0 Mask Enabled. <i>Note:</i> This register field is sticky.
PCI_TYPE0_BAR0_MASK	1	write-only	Register Size: 31 Value After Reset: BAR0_MASK_n Testable: untestable	- BAR0 Mask. - The name and the function of this field depends on the value of CX_CXL_ENABLE parameter as follows: 1: RCRB MEMBAR0[31:1]. - The MEMBAR0 register field is implemented in Type0 and Type1 Config Header capability when target protocol CXL is enabled and when the memorymap region is RCRB. - This is 32bits field, we use this to write MEMBAR0 address into RCRB space. MEMBAR0 registers should be able to access BAR0_REG offset 10h of PCIE memory-mapped space. 0: BAR0 Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) <i>Note:</i> This register field is sticky.

PF0_TYPE0_HDR_DBI2/BAR1_MASK_REG

Field Name	Bit Offset	Memory Access	Attributes	Description
PCI_TYPE0_BAR1_ENABLED	0	write-only	Register Size: 1 Value After Reset: BAR1_ENABLE_D_n Testable: untestable	- BAR1 Mask Enabled. - The name and the function of this field depends on the value of CX_CXL_ENABLE parameter as follows: 1: RCRB MEMBAR1[0]. - The MEMBAR1 register field is implemented in Type0 and Type1 Config Header capability when target protocol CXL is enabled and when the memorymap region is RCRB. - This is 32bits field, we use this to write MEMBAR1 address into RCRB space. MEMBAR1 registers should be able to access BAR1_REG offset 14h of PCIE memory-mapped space. 0: BAR1 Mask Enabled. <i>Note:</i> This register field is sticky.
PCI_TYPE0_BAR1_MASK	1	write-only	Register Size: 31 Value After Reset: BAR1_MASK_n Testable: untestable	- BAR1 Mask. The name and the function of this field depends on the value of CX_CXL_ENABLE parameter as follows: 1: RCRB MEMBAR1[31:0]. - The MEMBAR1 register field is implemented in Type0 and Type1 Config Header capability when target protocol CXL is enabled and when the memorymap region is RCRB. - This is 32bits field, we use this to write MEMBAR1 address into RCRB space. MEMBAR1 registers should be able to access BAR1_REG offset 14h of PCIE memory-

Field Name	Bit Offset	Memory Access	Attributes	Description
				<p>mapped space. 0: BAR1 Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) <i>Note:</i> This register field is sticky.</p>

PF0_TYPE0_HDR_DBI2/BAR2_MASK_REG

Field Name	Bit Offset	Memory Access	Attributes	Description
PCI_TYPE0_BAR2_ENABLED	0	write-only	Register Size: 1 Value After Reset: BAR2_ENABLED_n Testable: untestable	BAR2 Mask Enabled. <i>Note:</i> This register field is sticky.
PCI_TYPE0_BAR2_MASK	1	write-only	Register Size: 31 Value After Reset: BAR2_MASK_n Testable: untestable	BAR2 Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) <i>Note:</i> This register field is sticky.

PF0_TYPE0_HDR_DBI2/BAR3_MASK_REG

Field Name	Bit Offset	Memory Access	Attributes	Description
PCI_TYPE0_BAR3_ENABLED	0	write-only	Register Size: 1 Value After Reset: BAR3_ENABLED_n Testable: untestable	BAR3 Mask Enabled. <i>Note:</i> This register field is sticky.
PCI_TYPE0_BAR3_MASK	1	write-only	Register Size: 31 Value After Reset: BAR3_MASK_n Testable: untestable	BAR3 Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) <i>Note:</i> This register field is sticky.

Register Fields for PF0_TYPE0_HDR_DBI2/BAR4_MASK_REG

Field Name	Bit Offset	Memory Access	Attributes	Description
PCI_TYPE0_BAR4_ENABLED	0	write-	Register Size: 1	BAR4 Mask Enabled.

Field Name	Bit Offset	Memory Access	Attributes	Description
		only	Value After Reset: BAR4_ENABLED_n Testable: untestable	<i>Note:</i> This register field is sticky.
PCI_TYPE0_BAR4_MASK	1	write-only	Register Size: 31 Value After Reset: BAR4_MASK_n Testable: untestable	BAR4 Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) <i>Note:</i> This register field is sticky.

Register Fields for PF0_TYPE0_HDR_DBI2/BAR5_MASK_REG

Field Name	Bit Offset	Memory Access	Attributes	Description
PCI_TYPE0_BAR5_ENABLED	0	write-only	Register Size: 1 Value After Reset: BAR5_ENABLED_n Testable: untestable	BAR5 Mask Enabled. <i>Note:</i> This register field is sticky.
PCI_TYPE0_BAR5_MASK	1	write-only	Register Size: 31 Value After Reset: BAR5_MASK_n Testable: untestable	BAR5 Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: W (sticky) <i>Note:</i> This register field is sticky.

Register Fields for PF0_TYPE0_HDR_DBI2/EXP_ROM_BAR_MASK_REG

Field Name	Bit Offset	Memory Access	Attributes	Description
ROM_BAR_ENABLED	0	write-only	Register Size: 1 Value After Reset: 0x0 Testable: untestable	Expansion ROM Bar Mask Register Enabled. <i>Note:</i> The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if ROM_MASK_WRITABLE then W <i>Note:</i> This register field is sticky.
ROM_MASK	1	write-only	Register Size: 31 Value After Reset: 0xffff Testable: untestable	Expansion ROM Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: No access - Dbi: No access - Dbi2: if ROM_BAR_ENABLED && ROM_MASK_WRITABLE then W <i>Note:</i> This register field is sticky.

Register Fields for PCIE_DSP/PF0_PORT_LOGIC/MSI_CTRL_ADDR_OFF

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_CTRL_ADDR	0	read-write	Register Size: 32 Value After Reset: 0x0	<p>Integrated MSI Reception Module Address. System specified address for MSI memory write transaction termination. Within the AXI Bridge, every received Memory Write request is examined to see if it targets the MSI Address that has been specified in this register; and also to see if it satisfies the definition of an MSI interrupt request. When these conditions are satisfied the Memory Write request is marked as an MSI request.</p> <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PCIE_DSP/PF0_PORT_LOGIC/MSI_CTRL_UPPER_ADDR_OFF

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_CTRL_UPPER_ADDR	0	read-write	Register Size: 32 Value After Reset: 0x0	<p>Integrated MSI Reception Module Upper Address. System specified upper address for MSI memory write transaction termination. Allows functions to support a 64-bit MSI address.</p> <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PCIE_DSP/PF0_PORT_LOGIC/MSI_CTRL_INT_0_EN_OFF

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_CTRL_INT_0_EN	0	read-write	Register Size: 32 Value After Reset: 0x0	<p>MSI Interrupt0 Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector.</p> <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PCIE_DSP/PF0_PORT_LOGIC/MSI_CTRL_INT_0_MASK_OFF

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_CTRL_INT_0_MASK	0	read-write	Register Size: 32 Value After	MSI Interrupt0 Mask. Allows enabled interrupts to be masked. When an MSI is received for a

Field Name	Bit Offset	Memory Access	Attributes	Description
			Reset: 0x0	masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. <i>Note:</i> This register field is sticky.

Register Fields for PCIE_DSP/PF0_PORT_LOGIC/MSI_CTRL_INT_i_STATUS_OFF

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_CTRL_INT_0_STATUS	0	read-write	Register Size: 32 Value After Reset: 0x0 Write Behavior: oneToClear Testable: writeAsRead	MSI Interrupt0 Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

Register Fields for PCIE_DSP/PF0_PORT_LOGIC/MSI_GPIO_IO_OFF

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_GPIO_REG	0	read-write	Register Size: 32 Value After Reset: 0x0	MSI GPIO Register. The contents of this register drive the top-level GPIO msi_ctrl_io[31:0]. <i>Note:</i> This register field is sticky.

Register Fields for PF0_PORT_LOGIC/MSIX_ADDRESS_MATCH_HIGH_OFF

Field Name	Bit Offset	Memory Access	Attributes	Description
MSIX_ADDRESS_MATCH_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0	MSI-X Address Match High Address. <i>Note:</i> This register field is sticky.

Register Fields for PF0_PORT_LOGIC/MSIX_DOORBELL_OFF

Field Name	Bit Offset	Memory Access	Attributes	Description
MSIX_DOORBELL_VECTOR	0	write-only	Register Size: 11	MSI-X Doorbell Vector. This register determines

Field Name	Bit Offset	Memory Access	Attributes	Description
			Value After Reset: 0x0	which vector to generate the MSI-X transaction for.
MSIX_DOORBELL_RESERVED_11	11	write-only	Register Size: 1 Value After Reset: 0x0	Reserved.
MSIX_DOORBELL_TC	12	write-only	Register Size: 3 Value After Reset: 0x0	MSIX Doorbell Traffic Class. This register determines which traffic class to generate the MSI-X transaction with.
MSIX_DOORBELL_VF_ACTIVE	15	write-only	Register Size: 1 Value After Reset: 0x0	MSIX Doorbell Virtual Function Active. This register determines whether a Virtual Function is used to generate the MSI-X transaction.
MSIX_DOORBELL_VF	16	write-only	Register Size: 8 Value After Reset: 0x0	MSIX Doorbell Virtual Function. This register determines the Virtual Function for the MSI-X transaction.
MSIX_DOORBELL_PF	24	write-only	Register Size: 5 Value After Reset: 0x0	MSIX Doorbell Physical Function. This register determines the Physical Function for the MSI-X transaction.
MSIX_DOORBELL_RESERVED_29_31	29	write-only	Register Size: 3 Value After Reset: 0x0	Reserved.

Register Fields for PF0_PORT_LOGIC/MSIX_RAM_CTRL_OFF

Field Name	Bit Offset	Memory Access	Attributes	Description
MSIX_RAM_CTRL_TABLE_DS	0	read-write	Register Size: 1 Value After Reset: 0x0	MSIX Table RAM Deep Sleep. <i>Note:</i> This register field is sticky.
MSIX_RAM_CTRL_TABLE_SD	1	read-write	Register Size: 1 Value After Reset: 0x0	MSIX Table RAM Shut Down. <i>Note:</i> This register field is sticky.

Field Name	Bit Offset	Memory Access	Attributes	Description
MSIX_RAM_CTRL_RESERVED_2_7	2	read-only	Register Size: 6 Value After Reset: 0x0	Reserved. <i>Note:</i> This register field is sticky.
MSIX_RAM_CTRL_PBA_DS	8	read-write	Register Size: 1 Value After Reset: 0x0	MSIX PBA RAM Deep Sleep. <i>Note:</i> This register field is sticky.
MSIX_RAM_CTRL_PBA_SD	9	read-write	Register Size: 1 Value After Reset: 0x0	MSIX PBA RAM Shut Down. <i>Note:</i> This register field is sticky.
MSIX_RAM_CTRL_RESERVED_10_15	10	read-only	Register Size: 6 Value After Reset: 0x0	Reserved. <i>Note:</i> This register field is sticky.
MSIX_RAM_CTRL_BYPASS	16	read-write	Register Size: 1 Value After Reset: 0x0	MSIX RAM Control Bypass. It is up to the application to ensure the RAMs are in the proper power state before trying to access them. Moreover, the application needs to observe all timing requirements of the RAM low power signals before trying to use the MSIX functionality. <i>Note:</i> This register field is sticky.
MSIX_RAM_CTRL_RESERVED_17_23	17	read-only	Register Size: 7 Value After Reset: 0x0	Reserved. <i>Note:</i> This register field is sticky.
MSIX_RAM_CTRL_DBG_TABLE	24	read-write	Register Size: 1 Value After Reset: 0x0	MSIX Table RAM Debug Mode. You can also use the dbg_table input to activate debug mode. Debug mode turns off the PF/VF/Offset-based addressing into the RAM and maps the entire table linearly from the base address of the BAR

Field Name	Bit Offset	Memory Access	Attributes	Description
				(indicated by the BIR) in function 0. <i>Note:</i> This register field is sticky.
MSIX_RAM_CTRL_DBG_PBA	25	read-write	Register Size: 1 Value After Reset: 0x0	MSIX PBA RAM Debug Mode. You can also use the dbg_pba input to activate debug mode. Debug mode turns off the PF/VF/Offset-based addressing into the RAM and maps the entire table linearly from the base address of the BAR (indicated by the BIR) in function 0. <i>Note:</i> This register field is sticky.
MSIX_RAM_CTRL_RESERVED_26_31	26	read-only	Register Size: 6 Value After Reset: 0x0	Reserved. <i>Note:</i> This register field is sticky.

Register Fields for PF0_ATU_CAP/IATU_REGION_CTRL_1_OFF_OUTBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
TYPE	0	read-write	Register Size: 5 Value After Reset: 0x0	When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. <i>Note:</i> This register field is sticky.
TC	5	read-write	Register Size: 3 Value After Reset: 0x0	When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. <i>Note:</i> This register field is sticky.
TD	8	read-write	Register Size: 1 Value After Reset: 0x0	This is a reserved field. Do not use. <i>Note:</i> This register field is sticky.
ATTR	9	read-write	Register Size: 2 Value After Reset: 0x0	When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. <i>Note:</i> This register field is sticky.
INCREASE_REGION_SIZE	13	read-write	Register Size: 1	Increase the maximum ATU Region size. When set, the maximum ATU

Field Name	Bit Offset	Memory Access	Attributes	Description
			Value After Reset: 0x0	Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). <i>Note:</i> This register field is sticky.
AT	16	read-write	Register Size: 2 Value After Reset: 0x0	AT. When the address of an outbound TLP is matched to this region, then the AT field of the TLP is changed to the value in this register. Only valid when the ATS_ENABLE configuration parameter is '1'. <i>Note:</i> This register field is sticky.
CTRL_1_FUNC_NUM	20	read-write	Register Size: 3 Value After Reset: 0x0	Function Number. When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Reserved. You must set this bit to '0' Function Number. When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Region Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. The value in this register must be 0x0 unless multifunction operation in the controller is enabled (CX_NFUNC > 1). <i>Note:</i> This register field is sticky.

Register Fields for PF0_ATU_CAP/IATU_REGION_CTRL_2_OFF_OUTBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSG_CODE	0	read-write	Register Size: 8 Value After	MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or

Field Name	Bit Offset	Memory Access	Attributes	Description
			Reset: 0x0	<p>MsgD; then the message field of the TLP is changed to the value in this register.</p> <p>Memory TLPs: (ST: Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Only Valid when the CX_TPH_ENABLE configuration parameter is 1.</p> <p><i>Note:</i> This register field is sticky.</p>
TAG	8	read-write	Register Size: 8 Value After Reset: 0x0	<p>TAG.</p> <p>The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set.</p> <p><i>Note:</i> This register field is sticky.</p>
TAG_SUBSTITUTE_EN	16	read-write	Register Size: 1 Value After Reset: 0x0	<p>TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/Msgd.</p> <p><i>Note (CX_TPH_ENABLE=1):</i> TAG substitution for MWr will not occur because this field (byte 6) in the TLP header is the ST field. ST substitution can still take place using the MSG_CODE field in IATU_REGION_CTRL_2_OFF_OUTBOUND_i.</p> <p><i>Note (CX_10BIT_TAG = 1):</i> For 10-bit tags, TAG substitution only applies to the 8 least significant bits of the TAG field (bits 7:0 of Byte 4 of the TLP Header). T9 and T8 of the Header are not substituted.</p> <p><i>Note:</i> This register field is sticky.</p>
FUNC_BYPASS	19	read-write	Register Size: 1 Value After Reset: 0x0	<p>Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Region Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register."</p> <p><i>Note:</i> This register field is sticky.</p>
SNP	20	read-write	Register Size: 1 Value After Reset: 0x0	<p>Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding.</p> <p><i>Note:</i> This register field is sticky.</p>
TLP_HEADER_FIELDS_BYPASS	21	read-write	Register Size: 1 Value After Reset: 0x0	<p>TLP Header Fields Translation Bypass. In this mode header fields of the translated TLP is taken from your application transmit interface or, if AMBA is configured, from the AMBA sideband bus (slv_awmisc_info) and not from the</p>

Field Name	Bit Offset	Memory Access	Attributes	Description
				<p>corresponding fields of the IATU_REGION_CTRL_1_OFF_OUTBOUND_i or IATU_REGION_CTRL_2_OFF_OUTBOUND_i registers. The header fields are - TC - PH - TH - ST - AT - Attr (IDO, RO and NS). <i>Note:</i> This register field is sticky.</p>
INHIBIT_PAYLOAD	22	read-write	Register Size: 1 Value After Reset: 0x0	<p>Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. Encoding are define as above. <i>Note:</i> This register field is sticky.</p>
HEADER_SUBSTITUTE_EN	23	read-write	Register Size: 1 Value After Reset: 0x0	<p>Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. Encodings are as above. <i>Note:</i> This register field is sticky.</p>
DMA_BYPASS	27	read-write	Register Size: 1 Value After Reset: 0x0	<p>DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. <i>Note:</i> This field is reserved for the SW product. You must set it to '0'. <i>Note:</i> This register field is sticky.</p>
CFG_SHIFT_MODE	28	read-write	Register Size: 1 Value After Reset: 0x0	<p>CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. <i>Note:</i> This register field is sticky.</p>
INVERT_MODE	29	read-write	Register Size: 1 Value After Reset: 0x0	<p>Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). <i>Note:</i> This register field is sticky.</p>
REGION_EN	31	read-write	Register Size: 1 Value After	Region Enable. This bit must be set to '1' for address translation to take place.

Field Name	Bit Offset	Memory Access	Attributes	Description
			Reset: 0x0	<i>Note:</i> This register field is sticky.

Register Fields for PF0_ATU_CAP/IATU_LWR_BASE_ADDR_OFF_OUTBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
LWR_BASE_HW	0	read-only	Register Size: 12 Value After Reset: 0x0	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is $\log_2(\text{CX_ATU_MIN_REGION_SIZE})$
LWR_BASE_RW	12	read-write	Register Size: 20 Value After Reset: 0x0	Forms bits [31:n] of the start address of the address region to be translated. n is $\log_2(\text{CX_ATU_MIN_REGION_SIZE})$ <i>Note:</i> This register field is sticky.

Register Fields for PF0_ATU_CAP/IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
UPPER_BASE_RW	0	read-write	Register Size: 32 Value After Reset: 0x0	Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. <i>Note:</i> This register field is sticky.

Register Fields for PF0_ATU_CAP/IATU_LIMIT_ADDR_OFF_OUTBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
CBUF_INCR	0	read-only	Register Size: 4 Value After Reset: 0xf	Circular Buffer. <i>Note:</i> This register field is sticky.
LIMIT_ADDR_HW	4	read-only	Register Size: 8 Value After Reset: 0xff	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. A write to this location is ignored by the PCIe controller. <i>Note:</i> This register field is sticky.
LIMIT_ADDR_RW	12	read-write	Register Size: 20 Value After Reset: 0x0	Forms upper bits of the end address of the address region to be translated. <i>Note:</i> This register field is sticky.

Register Fields for PF0_ATU_CAP/IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
LWR_TARGET_RW_OUTBOUND	0	read-write	Register Size: 32 Value After Reset: 0x0	<p>When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(CX_ATU_MIN_REGION_SIZE). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header.</p> <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PF0_ATU_CAP/IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
UPPER_TARGET_RW	0	read-write	Register Size: 32 Value After Reset: 0x0	<p>Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region.</p> <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PF0_ATU_CAP/IATU_UPPR_LIMIT_ADDR_OFF_OUTBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
UPPR_LIMIT_ADDR_RW	0	read-write	Register Size: 1 Value After Reset: 0x0	<p>Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'</p> <p><i>Note:</i> This register field is sticky.</p>
UPPR_LIMIT_ADDR_HW	1	read-only	Register Size: 31 Value After Reset: 0x0	<p>Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_OUTBOUND_i is '1'. These bits are always '0'.</p>

Register Fields for PF0_ATU_CAP/iATU_REGION_CTRL_1_OFF_INBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
TYPE	0	read-write	Register Size: 5 Value After Reset: 0x0	When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). <i>Note:</i> This register field is sticky.
TC	5	read-write	Register Size: 3 Value After Reset: 0x0	When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Region Control 2 Register" is set. <i>Note:</i> This register field is sticky.
TD	8	read-write	Register Size: 1 Value After Reset: 0x0	When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Region Control 2 Register" is set. <i>Note:</i> This register field is sticky.
ATTR	9	read-write	Register Size: 2 Value After Reset: 0x0	When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Region Control 2 Register" is set. <i>Note:</i> This register field is sticky.
INCREASE_REGION_SIZE	13	read-write	Register Size: 1 Value After Reset: 0x0	Increase the maximum ATU Region size. When set, the maximum ATU Region size is determined by CX_ATU_MAX_REGION_SIZE. When clear, the maximum ATU Region size is 4 GB (default). <i>Note:</i> This register field is sticky.
AT	16	read-write	Register Size: 2 Value After Reset: 0x0	When the AT field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "AT Match Enable" bit of the "iATU Region Control 2 Register" is set. Only valid when the ATS_ENABLE configuration parameter is '1'. <i>Note:</i> This register field is sticky.
CTRL_1_FUNC_NUM	20	read-write	Register Size: 3 Value After Reset: 0x0	Reserved. You must set this bit to '0' Function Number. MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Region Control 2 Register" is set. CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match

Field Name	Bit Offset	Memory Access	Attributes	Description
				Enable" bit of the "iATU Region Control 2 Register" is set. <i>Note:</i> This register field is sticky.

Register Fields for PF0_ATU_CAP/IATU_REGION_CTRL_2_OFF_INBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSG_CODE	0	read-write	Register Size: 8 Value After Reset: 0x0	MSG TLPs: (Message Code). When the MSG_CODE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Region Control 2 Register" is set and the TYPE field of the inbound TLP is Msg/MsgD. Memory TLPs: (ST: Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Region Control 2 Register" is set. The setting is independent of the setting of the TH field. Only Valid when the CX_TPH_ENABLE configuration parameter is '1'. <i>Note:</i> This register field is sticky.
BAR_NUM	8	read-write	Register Size: 3 Value After Reset: 0x0	BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Region Control 2 Register" is set. IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR.Reserved. You must set this bit to '0' <i>Note:</i> This register field is sticky.
MSG_TYPE_MATCH_MODE	13	read-write	Register Size: 1 Value After Reset: 0x0	Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the IATU_REGION_CTRL_1_VIEWPORT_OFF_INBOUND_i register (TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. <i>Note:</i> This register field is sticky.
TC_MATCH_EN	14	read-write	Register Size: 1 Value After Reset: 0x0	TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Region Control 1 Register") occurs for address translation to proceed. <i>Note:</i> This register field is sticky.

Field Name	Bit Offset	Memory Access	Attributes	Description
TD_MATCH_EN	15	read-write	Register Size: 1 Value After Reset: 0x0	TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Region Control 1 Register") occurs for address translation to proceed. <i>Note:</i> This register field is sticky.
ATTR_MATCH_EN	16	read-write	Register Size: 1 Value After Reset: 0x0	ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Region Control 1 Register") occurs for address translation to proceed. <i>Note:</i> This register field is sticky.
AT_MATCH_EN	18	read-write	Register Size: 1 Value After Reset: 0x0	AT Match Enable. Ensures that a successful AT TLP field comparison match (see AT field of the "iATU Region Control 1 Register") occurs for address translation to proceed. Only valid when the ATS_ENABLE configuration parameter is '1'. <i>Note:</i> This register field is sticky.
FUNC_NUM_MATCH_EN	19	read-write	Register Size: 1 Value After Reset: 0x0	Reserved. You must set this bit to '0'. Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Region Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. <i>Note:</i> This register field is sticky.
MSG_CODE_MATCH_EN	21	read-write	Register Size: 1 Value After Reset: 0x0	Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Region Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPS). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Region Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Only Valid when the CX_TPH_ENABLE configuration parameter is '1'. <i>Note:</i> This register field is sticky.
SINGLE_ADDR_LOC_TRANS_EN	23	read-write	Register Size: 1 Value After Reset: 0x0	Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. <i>Note:</i> This register field is sticky.
RESPONSE_CODE	24	read-write	Register Size: 2 Value After Reset: 0x0	Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. <i>Note:</i> This register field is sticky.
FUZZY_TYPE_	27	read-write	Register Size: 1	Fuzzy Type Match Enable. When enabled, the iATU relaxes the matching of the TLP TYPE field against

Field Name	Bit Offset	Memory Access	Attributes	Description
MATCH_CODE			Value After Reset: 0x0	<p>the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd, and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap, and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Region Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0, or CfgWr1 TLP.</p> <p><i>Note:</i> This register field is sticky.</p>
CFG_SHIFT_MODE	28	read-write	Register Size: 1 Value After Reset: 0x0	<p>CFG Shift Enable. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address.</p> <p><i>Note:</i> This register field is sticky.</p>
INVERT_MODE	29	read-write	Register Size: 1 Value After Reset: 0x0	<p>Invert Mode Enable. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). When set all regions of that type must use address match mode.</p> <p><i>Note:</i> This register field is sticky.</p>
MATCH_MODE	30	read-write	Register Size: 1 Value After Reset: 0x0	<p>Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows:</p> <p>For MEM I/O TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. - 1: BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: - 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. - 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. - 1: Vendor ID Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and</p>

Field Name	Bit Offset	Memory Access	Attributes	Description
				<p>Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE =1, then Match Mode is ignored.</p> <p><i>Note:</i> This register field is sticky.</p>
REGION_EN	31	read-write	Register Size: 1 Value After Reset: 0x0	<p>Region Enable. This bit must be set to '1' for address translation to take place.</p> <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PF0_ATU_CAP/IATU_LWR_BASE_ADDR_OFF_INBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
LWR_BASE_HW	0	read-only	Register Size: 12 Value After Reset: 0x0	<p>Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.</p> <p>n is log2(CX_ATU_MIN_REGION_SIZE)</p>
LWR_BASE_RW	12	read-write	Register Size: 20 Value After Reset: 0x0	<p>Forms bits [31:n] of the start address of the address region to be translated.</p> <p>n is log2(CX_ATU_MIN_REGION_SIZE)</p> <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PF0_ATU_CAP/IATU_UPPER_BASE_ADDR_OFF_INBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
UPPER_BASE_RW	0	read-write	Register Size: 32 Value After Reset: 0x0	<p>Forms bits [63:32] of the start (and end) address of the address region to be translated.</p> <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PF0_ATU_CAP/IATU_LIMIT_ADDR_OFF_INBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
CBUF_INCR	0	read-write	Register Size: 4 Value After Reset: 0x0	<p>Circular Buffer Increment. When CX_ATU_SLOC_CBUF = 0, then this field is Read-only and forms the lowest bits of the end address of the address region to be translated. When CX_ATU_SLOC_CBUF = 1, then this field is R/W and forms the upper bits of the Circular Buffer Increment size (CBUF_INCR) field for Single</p>

Field Name	Bit Offset	Memory Access	Attributes	Description
				<p>Location Address translation. The increment value (in bytes) is decoded as follows: Note: A write to any bit in the CBUF_INCR field resets the circular buffer pointer - that is, the next matched received Message will be translated to the start address of the Circular Buffer. This field must be written to AFTER the target and limit registers have been updated.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R (sticky) - Dbi: R (sticky) <i>Note:</i> This register field is sticky.</p>
LIMIT_ADDR_HW	4	read-only	Register Size: 8 Value After Reset: 0xff	<p>Forms lower bits of the end address of the address region to be translated. The end address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary, so these bits are always all ones. When SINGLE_ADDR_LOC_TRANS_EN = 1, MSG_TYPE_MATCH_MODE = 1, and CBUF_INCR > 0 then this field forms lower bits of the limit address for the circular buffer. A write to this location is ignored by the PCIe controller.</p> <p><i>Note:</i> This register field is sticky.</p>
LIMIT_ADDR_RW	12	read-write	Register Size: 20 Value After Reset: 0x0	<p>Forms upper bits of the end address of the address region to be translated. When SINGLE_ADDR_LOC_TRANS_EN = 1, MSG_TYPE_MATCH_MODE = 1, and CBUF_INCR > 0 then this field forms the upper bits of the limit address for the circular buffer.</p> <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PF0_ATU_CAP/IATU_LWR_TARGET_ADDR_OFF_INBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
LWR_TARGET_HW	0	read-only	Register Size: 12 Value After Reset: 0x0	<p>Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to a CX_ATU_MIN_REGION_SIZE kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size.</p> <p>A write to this location is ignored by the PCIe controller.</p> <ul style="list-style-type: none"> - Field size depends on log2(CX_ATU_MIN_REGION_SIZE) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode.
LWR_TARGET_RW	12	read-write	Register Size: 20 Value After Reset: 0x0	<p>Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(CX_ATU_MIN_REGION_SIZE) in address match mode.</p> <ul style="list-style-type: none"> - Field size depends on log2(BAR_MASK+1) in BAR match mode. <p><i>Note:</i> This register field is sticky.</p>

Register Fields for PF0_ATU_CAP/IATU_UPPER_TARGET_ADDR_OFF_INBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
UPPER_TARGET_RW	0	read-write	Register Size: 32 Value After Reset: 0x0	Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. <i>Note:</i> This register field is sticky.

Register Fields for PF0_ATU_CAP/IATU_UPPR_LIMIT_ADDR_OFF_INBOUND_i

Field Name	Bit Offset	Memory Access	Attributes	Description
UPPR_LIMIT_ADDR_RW	0	read-write	Register Size: 1 Value After Reset: 0x0	Forms the LSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_INBOUND_i is '1' <i>Note:</i> This register field is sticky.
UPPR_LIMIT_ADDR_HW	1	read-only	Register Size: 31 Value After Reset: 0x0	Forms MSB's of the Upper Limit part of the region "end address" to be translated. Only applies to 64-bit systems and when the INCREASE_REGION_SIZE field in IATU_REGION_CTRL_1_OFF_INBOUND_i is '1'. These bits are always '0'.

Register Fields for PF0_HDMA_CAP/HDMA_EN_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
ENABLE	0	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Write Channel Enable. The controller checks this field for power management purposes. If this field is enabled for any one of the read, or write channel, the controller exits low power state. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_DOORBELL_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
DB_START	0	write-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Write Channel Doorbell Start. You must set this field to start the write transfer for this channel.
DB_STOP	1	write-only	Register Size: 1 Value After Reset: 0x0 Testable:	HDMA Write Channel Doorbell Stop. You must set this field to stop the write transfer for this channel. HDMA stops the write transfer for this channel at the earliest when this field is set.

			writeAsRead	<i>Note:</i> Only after the HDMA_STATUS_OFF_WRCH_i.STATUS =0x03, you can consider this channel to be in stop state.
--	--	--	-------------	---

Register Fields for PF0_HDMA_CAP/HDMA_ELEM_PF_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
ELEMENT_PREFETCH	0	read-write	Register Size: 7 Value After Reset: 0x0 Testable: writeAsRead	<p>This field controls the number of linked list elements (descriptors) this HDMA write channel prefetches. The linked list element pointer loaded by HDMA updates this field.</p> <p><i>Note:</i> The actual prefetch value is ELEMENT_PREFETCH plus '1'. For example, if ELEMENT_PREFETCH =0, one descriptor is prefetched by this channel.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_LL_P_LOW_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
ELEMENT_LIST_PTR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	<p>Lower 32 bits of the address of the transfer list in the local memory. Used in linked list mode only. HDMA fetches descriptors from local memory respecting the following patterns:</p> <ul style="list-style-type: none"> - When the current element is a data element; HDMA increments this field by 6 DWORDs times the prefetch depth. For more information, see registers HDMA_ELEM_PF_OFF_WRCH* and HDMA_QOS_OFF_WRCH* - When the current element is a link element; HDMA overwrites this field with the LL Element Pointer of the next LL element structure. The current fetched memory address pointer is not directly visible in this register, instead HDMA updates this register on the following occurrences: <ul style="list-style-type: none"> - Watermark interrupt event - Channel Status ABORT event - Channel Status STOP event <p>On a watermark interrupt event this register points to the descriptor's address that triggered it. A channel status STOP or ABORT events are end of transfer events that makes this register point to the next data to transfer after the event.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_LL_P_HIGH_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description	
ELEMENT_LIST_PTR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable:	Higher 32 bits of the address of the transfer list in the local memory. HDMA fetches descriptors from local memory respecting the following patterns:	<ul style="list-style-type: none"> - When the current element is a data element; HDMA increments this field by 6

Field Name	Bit Offset	Memory Access	Attributes	Description
			writeAsRead	<p>DWORDs times the prefetch depth. For more information, see registers HDMA_ELEM_PF_OFF_WRCH* and HDMA_QOS_OFF_WRCH* - When the current element is a link element; HDMA overwrites this field with the LL Element Pointer of the next LL element structure. The current fetched memory address pointer is not directly visible in this register, instead HDMA updates this register on the following occurrences:</p> <ul style="list-style-type: none"> - Watermark interrupt event - Channel Status ABORT event - Channel Status STOP event <p>On a watermark interrupt event this register points to the descriptor's address that triggered it. A channel status STOP or ABORT events are end of transfer events that makes this register point to the next data to transfer after the event.</p> <p><i>Note:</i> The access attributes of this field are as follows:</p> <ul style="list-style-type: none"> - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_CYCLE_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
CYCLE_BIT	0	read-write	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	<p>Consumer Cycle Bit.</p> <p><i>Note:</i> The access attributes of this field are as follows:</p> <ul style="list-style-type: none"> - Wire: R/W - Dbi: R/W
CYCLE_STATE	1	read-write	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	<p>Consumer Cycle State. You must initialize this field.</p> <p><i>Note:</i> The access attributes of this field are as follows:</p> <ul style="list-style-type: none"> - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_XFERSIZE_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
XFERSIZE	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	<p>HDMA Write Channel Transfer Size. You program this register with the size of the HDMA transfer. The maximum HDMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). HDMA automatically decrements the value of this field as the write channel transfer progresses. This field indicates the number of bytes remaining to be requested. When all bytes are requested the current transfer size is zero. In LL mode, the HDMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, but it is not reliable for that specific usage, as it is updated after read requests and not after write requests.</p> <p><i>Note:</i> The access attributes of this field are as follows:</p> <ul style="list-style-type: none"> - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_SAR_LOW_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
SAR_PTR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	<p>Source Address Register (lower 32 bits). Indicates the address of the local memory from which HDMA reads. The HDMA increments the SAR as the HDMA transfer progresses. In LL mode, the HDMA overwrites this field with the corresponding dword of the LL element. The SAR is the address of the local memory.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_SAR_HIGH_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
SAR_PTR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	<p>Source Address Register (higher 32 bits).</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_DAR_LOW_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
DAR_PTR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	<p>Destination Address Register (lower 32 bits). Indicates the address to which HDMA writes. The HDMA increments the DAR as the HDMA transfer progresses. In LL mode, the HDMA overwrites this field with the corresponding dword of the LL element. The DAR is the address of the remote memory.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_DAR_HIGH_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
DAR_PTR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	<p>Destination Address Register (higher 32 bits).</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_WATERMARK_EN_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
RWIE	0	read-write	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	Enables Remote Interrupts at watermark events (end of linked list element) in an HDMA write channel. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
LWIE	1	read-write	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	Enables Local Interrupts at watermark events (end of linked list element) in an HDMA write channel. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_CONTROL1_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
LLEN	0	read-write	Register Size: 1 Value After Reset: 0x0	This field enables the HDMA write channel Linked List mode. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
MEM_TYPE	1	read-write	Register Size: 1 Value After Reset: 0x0	Master AXI ACE-Lite Cache Coherency Control. This field sets the HDMA write channel memory type of the address space of the data transfer as follows: For more information, see "ACE-Lite Features and Limitations" section of the <i>Databook</i> . Irrespective of the value of this field, Linked list descriptor requests have this attribute set to 0x1. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
SRC_SNOOP	2	read-write	Register Size: 1 Value After Reset: 0x0	Source No Snoop TLP Header Bit. The HDMA write channel uses this TLP header field when generating MRd (SAR addressing space) TLPs. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
DST_SNOOP	3	read-write	Register Size: 1 Value After Reset: 0x0	Destination No Snoop TLP Header Bit. The HDMA write channel uses this TLP header field when generating MWr/IMWr (DAR addressing space) TLPs. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
RO	4	read-write	Register Size: 1 Value After Reset: 0x0	Relaxed Ordering TLP Header Bit. HDMA write channel uses this TLP header field when generating MRd/MWr/IMWr TLPs. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
AT	5	read-write	Register Size: 2 Value After Reset: 0x0	Address Translation Services TLP Header Bit (AT). The HDMA write channel uses this TLP header field when generating MRd/MWr/IMWr TLPs. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_FUNC_NUM_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
PF	0	read-write	Register Size: 5 Value After Reset: 0x0	<p>DMA Channel Physical Function Number.</p> <p>DMA channel wide physical function of the generated TLP. The controller uses this field to form the requester ID for requests. Function numbering starts at '0'. When you have enabled SR-IOV, refer to VF and VF_EN fields of this register as well. The behavior is undefined if any value outside of the valid range is set to this field.</p> <p>At the signal interfaces, DMA drives the XADM / RADM signals d_client[0 1].tlp_func_num / d_radm_trgt1_func_num.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_QOS_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
TC	0	read-write	Register Size: 3 Value After Reset: 0x0	<p>Traffic Class TLP Header Field (TC). The HDMA write channel uses this TLP header field when generating MRd/MWr/IMWWr TLPS.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>
WEIGHT	3	read-write	Register Size: 5 Value After Reset: 0x1	<p>HDMA Write Channel Weight. Specifies the maximum number of TLP MRd requests that the HDMA can issue for a channel, before it returns to the arbitration routine. When a channel reaches its weight count or HDMA channel request transfer size reaches zero, the weighted round robin (WRR) arbiter selects the next channel to be processed. Your software must initialize this field before ringing the doorbell. For more information, see "Multichannel Arbitration" in the <i>Databook</i>.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>
PF_DEPTH	16	read-write	Register Size: 10 Value After Reset: 0x0	<p>This field controls the linked list prefetch allocation space for an HDMA write channel. Each write channel stores PF_DEPTH plus '1' linked list elements. For example, if PF_DEPTH =0, the write channel stores one linked list element. This field must be within the range: [0 : floor(CX_DMA_WR_LLQ * CC_NUM_DMA_WR_CHAN / number_of_active_channels) - 1].</p> <p><i>Note:</i></p> <p>If this field is set to a value outside the specified range, data corruption or channel crosstalk may happen.</p> <p>If PF_DEPTH is changed at run time, only channels in the range: [0 : number_of_active_channels - 1] can be doorbelled.</p> <p>This field cannot be changed while any write engine channel is in doorbell state. Doing so might yield undefined results.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_STATUS_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
STATUS	0	read-only	Register Size: 3 Value After Reset: 0x3 Testable: writeAsRead	HDMA Write Channel Status.

Register Fields for PF0_HDMA_CAP/HDMA_INT_STATUS_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
STOP	0	read-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Write Channel Stop Interrupt Status. STOP and ABORT register fields are mutually exclusive.
WATERMARK	1	read-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Write Channel Watermark Interrupt Status.
ABORT	2	read-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Write Channel Abort Interrupt Status. ABORT and STOP register fields are mutually exclusive.
ERROR	3	read-only	Register Size: 4 Value After Reset: 0x0 Testable: writeAsRead	HDMA Write Channel Interrupt Error Status. Specifies the error that caused the HDMA to change HDMA_STATUS_OFF_WRCH_i.STATUS to ABORTED state.

Register Fields for PF0_HDMA_CAP/HDMA_INT_SETUP_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
STOP_MASK	0	read-write	Register Size: 1 Value After Reset: 0x1	HDMA Write Channel Stop Interrupt Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
WATERMARK_MASK	1	read-write	Register Size: 1 Value After Reset: 0x1	HDMA Write Channel Watermark Interrupt Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Field Name	Bit Offset	Memory Access	Attributes	Description
ABORT_MASK	2	read-write	Register Size: 1 Value After Reset: 0x1	HDMA Write Channel Abort Interrupt Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
RSIE	3	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Write Channel Remote Stop Interrupt Enable. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
LSIE	4	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Write Channel Local Stop Interrupt Enable. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
RAIE	5	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Write Channel Remote Abort Interrupt Enable. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
LAIE	6	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Write Channel Local Abort Interrupt Enable. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_INT_CLEAR_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
STOP_CLEAR	0	write-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Write Channel Stop Interrupt Clear.
WATERMARK_CLEAR	1	write-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Write Channel Watermark Interrupt Clear.
ABORT_CLEAR	2	write-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Abort Channel Abort Interrupt Clear.

Register Fields for PF0_HDMA_CAP/HDMA_MSI_STOP_LOW_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_STOP_ADDR_LOW	0	read-write	Register Size: 32	The HDMA uses this field to generate bits [31:0] of the address field for the Stop

Field Name	Bit Offset	Memory Access	Attributes	Description
			Value After Reset: 0x0	Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_STOP_HIGH_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_STOP_ADDR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [63:32] of the address field for the Stop Interrupt MWr TLP. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_WATERMARK_LOW_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_WATERMARK_ADDR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [31:0] of the address field for the Watermark Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_WATERMARK_HIGH_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_WATERMARK_ADDR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [31:0] of the address field for the Watermark Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_ABORT_LOW_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_ABORT_ADDR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [31:0] of the address field for the Abort Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_ABORT_HIGH_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_ABORT_ADDR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [31:0] of the address field for the Abort Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_MSGD_OFF_WRCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_MESSAGE	0	read-write	Register Size: 16 Value After Reset: 0x0	The HDMA write channel uses this field to generate the data field for every IMWr TLPs it generates. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_EN_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
ENABLE	0	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Read Channel Enable. The controller checks this field for power management purposes. If this field is enabled for any one of the read, or write channel, the controller exits low power state. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_DOORBELL_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
DB_START	0	write-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Read Channel Doorbell Start. You must set this field to start the read transfer for this channel.
DB_STOP	1	write-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Read Channel Doorbell Stop. You must set this field to stop the read transfer for this channel. HDMA stops the read transfer for this channel at the earliest when this field is set. <i>Note:</i> Only after the HDMA_STATUS_OFF_RDCH_i.STATUS =0x03, you can consider this channel to be in stop state.

Register Fields for PF0_HDMA_CAP/HDMA_ELEM_PF_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
ELEMENT_PREFETCH	0	read-write	Register Size: 7 Value After Reset: 0x0 Testable: writeAsRead	<p>This field controls the number of linked list elements (descriptors) this HDMA read channel prefetches. The linked list element pointer loaded by HDMA updates this field.</p> <p><i>Note:</i> The actual prefetch value is ELEMENT_PREFETCH plus '1'. For example, if ELEMENT_PREFETCH =0, one descriptor is prefetched by this channel.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for DWC_PCIE_USP/PF0_HDMA_CAP/HDMA_LL_P_LOW_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
ELEMENT_LIST_PTR_LO_W	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	<p>Lower 32 bits of the address of the transfer list in the local memory. Used in linked list mode only. HDMA fetches descriptors from local memory respecting the following patterns: - When the current element is a data element; HDMA increments this field by 6 DWORDs times the prefetch depth. For more information, see registers HDMA_ELEM_PF_OFF_RDCH* and HDMA_QOS_OFF_RDCH* - When the current element is a link element; HDMA overwrites this field with the LL Element Pointer of the next LL element structure. The current fetched memory address pointer is not directly visible in this register, instead HDMA updates this register on the following occurrences: - Watermark interrupt event - Channel Status ABORT event - Channel Status STOP event On a watermark interrupt event this register points to the descriptor's address that triggered it. A channel status STOP or ABORT events are end of transfer events that makes this register point to the next data to transfer after the event</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_LL_P_HIGH_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
ELEMENT_LIST_PTR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable:	<p>Higher 32 bits of the address of the transfer list in the local memory. HDMA fetches descriptors from local memory respecting the following patterns: - When the current element is a data element; HDMA increments this field by 6 DWORDs times the prefetch depth. For more</p>

Field Name	Bit Offset	Memory Access	Attributes	Description
			writeAsRead	<p>information, see registers HDMA_ELEM_PF_OFF_RDCH* and HDMA_QOS_OFF_RDCH* - When the current element is a link element; HDMA overwrites this field with the LL Element Pointer of the next LL element structure. The current fetched memory address pointer is not directly visible in this register, instead HDMA updates this register on the following occurrences:</p> <ul style="list-style-type: none"> - Watermark interrupt event - Channel Status ABORT event - Channel Status STOP event <p>On a watermark interrupt event this register points to the descriptor's address that triggered it. A channel status STOP or ABORT events are end of transfer events that makes this register point to the next data to transfer after the event.</p> <p><i>Note:</i> The access attributes of this field are as follows:</p> <ul style="list-style-type: none"> - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_CYCLE_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
CYCLE_BIT	0	read-write	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	<p>Consumer Cycle Bit.</p> <p><i>Note:</i> The access attributes of this field are as follows:</p> <ul style="list-style-type: none"> - Wire: R/W - Dbi: R/W
CYCLE_STATE	1	read-write	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	<p>Consumer Cycle State. You must initialize this field.</p> <p><i>Note:</i> The access attributes of this field are as follows:</p> <ul style="list-style-type: none"> - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_XFERSIZE_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
XFERSIZE	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	<p>HDMA Read Channel Transfer Size. You program this register with the size of the HDMA transfer. The maximum HDMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). HDMA automatically decrements the value of this field as the read channel transfer progresses. This field indicates the number of bytes remaining to be requested. When all bytes are requested the current transfer size is zero. In LL mode, the HDMA overwrites this register with the corresponding dword of the LL element.</p> <p>You can read this register to monitor the transfer progress, but it is not reliable for that specific usage, as it is updated after read requests and not after write requests.</p>

Field Name	Bit Offset	Memory Access	Attributes	Description
				<i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_SAR_LOW_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
SAR_PTR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	Source Address Register (lower 32 bits). Indicates the address of the local memory from which HDMA reads. The HDMA increments the SAR as the HDMA transfer progresses. In LL mode, the HDMA overwrites this field with the corresponding dword of the LL element. The SAR is the address of the remote memory. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_SAR_HIGH_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
SAR_PTR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	Source Address Register (higher 32 bits). <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_DAR_LOW_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
DAR_PTR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0 Testable: writeAsRead	Destination Address Register (lower 32 bits). Indicates the address to which HDMA writes. The HDMA increments the DAR as the HDMA transfer progresses. In LL mode, the HDMA overwrites this field with the corresponding dword of the LL element. The DAR is the address of the local memory. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_DAR_HIGH_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
DAR_PTR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0	Destination Address Register (higher 32 bits). <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Field Name	Bit Offset	Memory Access	Attributes	Description
			Testable: writeAsRead	

Register Fields for PF0_HDMA_CAP/HDMA_WATERMARK_EN_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
RWIE	0	read-write	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	Enables Remote Interrupts at watermark events (end of linked list element) in an HDMA read channel. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
LWIE	1	read-write	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	Enables Local Interrupts at watermark events (end of linked list element) in an HDMA read channel. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_CONTROL1_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
LLEN	0	read-write	Register Size: 1 Value After Reset: 0x0	This field enables the HDMA read channel Linked List mode. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
MEM_TYPE	1	read-write	Register Size: 1 Value After Reset: 0x0	Master AXI ACE-Lite Cache Coherency Control. This field sets the HDMA read channel memory type of the address space of the data transfer as follows: For more information, see "ACE-Lite Features and Limitations" section of the <i>Databook</i> . Irrespective of the value of this field, Linked list descriptor requests have this attribute set to 0x1. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
SRC_SNOOP	2	read-write	Register Size: 1 Value After Reset: 0x0	Source No Snoop TLP Header Bit. The HDMA read channel uses this TLP header field when generating MRd (SAR addressing space) TLPs. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
DST_SNOOP	3	read-write	Register Size: 1 Value After Reset: 0x0	Destination No Snoop TLP Header Bit. The HDMA read channel uses this TLP header field when generating MWr/IMWr (DAR addressing space) TLPs. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Field Name	Bit Offset	Memory Access	Attributes	Description
RO	4	read-write	Register Size: 1 Value After Reset: 0x0	Relaxed Ordering TLP Header Bit. HDMA read channel uses this TLP header field when generating MRd/MWr/IMWr TLPS. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
AT	5	read-write	Register Size: 2 Value After Reset: 0x0	Address Translation Services TLP Header Bit (AT). The HDMA read channel uses this TLP header field when generating MRd/MWr/IMWr TLPS. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_FUNC_NUM_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
PF	0	read-write	Register Size: 5 Value After Reset: 0x0	DMA Channel Physical Function Number. DMA channel wide physical function of the generated TLP. The controller uses this field to form the requester ID for requests. Function numbering starts at '0'. When you have enabled SR-IOV, refer to VF and VF_EN fields of this register as well. The behavior is undefined if any value outside of the valid range is set to this field. At the signal interfaces, DMA drives the XADM / RADM signals d_client[0 1]_tlp_func_num / d_radm_trgt1_func_num. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_QOS_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
TC	0	read-write	Register Size: 3 Value After Reset: 0x0	Traffic Class TLP Header Field (TC). The HDMA read channel uses this TLP header field when generating MRd/MWr/IMWr TLPS. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
WEIGHT	3	read-write	Register Size: 5 Value After Reset: 0x1	HDMA Read Channel Weight. Specifies the maximum number of TLP MRd requests that the HDMA can issue for a channel, before it returns to the arbitration routine. When a channel reaches its weight count or HDMA channel request transfer size reaches zero, the weighted round robin (WRR) arbiter selects the next channel to be processed. Your software must initialize this field before ringing the doorbell. For more information, see "Multichannel Arbitration" in the <i>Databook</i> . <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
PF_DEPTH	16	read-write	Register Size: 10 Value After Reset: 0x0	This field controls the linked list prefetch allocation space for an HDMA read channel. Each read channel stores PF_DEPTH plus '1' linked list elements. For example, if PF_DEPTH = 0, the read channel stores one linked list element. This field must be within the range: [0 : floor(CX_DMA_RD_LLQ * CC_NUM_DMA_RD_CHAN / number_of_active_channels) - 1].

Field Name	Bit Offset	Memory Access	Attributes	Description
				<p><i>Note:</i> - If this field is set to a value outside the specified range, data corruption or channel crosstalk may happen. - If PF_DEPTH is changed at run time, only channels in the range: [0 : number_of_active_channels - 1] can be doorbelled. - This field cannot be changed while any read engine channel is in doorbell state. Doing so might yield undefined results.</p> <p><i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W</p>

Register Fields for PF0_HDMA_CAP/HDMA_STATUS_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
STATUS	0	read-only	Register Size: 3 Value After Reset: 0x3 Testable: writeAsRead	HDMA Read Channel Status.

Register Fields for PF0_HDMA_CAP/HDMA_INT_STATUS_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
STOP	0	read-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Read Channel Stop Interrupt Status. STOP and ABORT register fields are mutually exclusive.
WATERMARK	1	read-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Read Channel Watermark Interrupt Status.
ABORT	2	read-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Read Channel Abort Interrupt Status. ABORT and STOP register fields are mutually exclusive.
ERROR	3	read-only	Register Size: 4 Value After Reset: 0x0 Testable: writeAsRead	HDMA Read Channel Error Interrupt Status. Specifies the error that caused the HDMA to change HDMA_STATUS_OFF_RDCH_i.STATUS to ABORTED state.

Register Fields for PF0_HDMA_CAP/HDMA_INT_SETUP_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
STOP_MASK	0	read-write	Register Size: 1 Value After Reset: 0x1	HDMA Read Channel Stop Interrupt Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
WATERMARK_MASK	1	read-write	Register Size: 1 Value After Reset: 0x1	HDMA Read Channel Watermark Interrupt Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
ABORT_MASK	2	read-write	Register Size: 1 Value After Reset: 0x1	HDMA Read Channel Abort Interrupt Mask. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
RSIE	3	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Read Channel Remote Stop Interrupt Enable. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
LSIE	4	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Read Channel Local Stop Interrupt Enable. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
RAIE	5	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Read Channel Remote Abort Interrupt Enable. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W
LAIE	6	read-write	Register Size: 1 Value After Reset: 0x0	HDMA Read Channel Local Abort Interrupt Enable. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_INT_CLEAR_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
STOP_CLEAR	0	write-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Read Channel Stop Interrupt Clear.
WATERMARK_CLEAR	1	write-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Read Channel Watermark Interrupt Clear.
ABORT_CLEAR	2	write-only	Register Size: 1 Value After Reset: 0x0 Testable: writeAsRead	HDMA Abort Channel Abort Interrupt Clear. Setting this field clears the ABORT and ERROR fields.

Register Fields for PF0_HDMA_CAP/HDMA_MSI_STOP_LOW_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_STOP_ADDR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [31:0] of the address field for the Stop Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_STOP_HIGH_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_STOP_ADDR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [63:32] of the address field for the Stop Interrupt MWr TLP. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_WATERMARK_LOW_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_WATERMARK_ADDR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [31:0] of the address field for the Watermark Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_WATERMARK_HIGH_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_WATERMARK_ADDR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [31:0] of the address field for the Watermark Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_ABORT_LOW_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_ABORT_ADDR_LOW	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [31:0] of the address field for the Abort Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_ABORT_HIGH_OFF_RDCH_i

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_ABORT_ADDR_HIGH	0	read-write	Register Size: 32 Value After Reset: 0x0	The HDMA uses this field to generate bits [31:0] of the address field for the Abort Interrupt MWr TLP. Bits [1:0] must be '00' as this address must be dword aligned. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

Register Fields for PF0_HDMA_CAP/HDMA_MSI_MSGD_OFF_RDCH_j

Field Name	Bit Offset	Memory Access	Attributes	Description
MSI_MESSAGE	0	read-write	Register Size: 16 Value After Reset: 0x0	The HDMA read channel uses this field to generate the data field for every IMWr TLPs it generates. <i>Note:</i> The access attributes of this field are as follows: - Wire: R/W - Dbi: R/W

10.2 GMAC

10.2.1 Overview

EIC7700X has two lanes of Ethernet module, which interact with the internet through high-speed IO and can directly connect to a switch or router. and has following key features:

- The two Ethernet modules can share the task of data transmission and implement Ethernet bonding in the protocol stack to achieve load balancing while jointly bearing the data transmission load.
- Implementing Gigabit Ethernet and downward compatibility with 100M and 10M Ethernet.
- Supports full- and half-duplex modes, and flow control under corresponding conditions.
- Support RGMII, RMII interface
- Support IP v4 and IP v6
- Support SA insertion and replacement, and VLAN insertion, replacement and delete
- Supports controlling and managing external PHY chips through MDIO
- Supports up to 8 MAC filtering addresses, supports DA hash filtering, supports VLAN hash filtering, and supports layer 3/4 packet filtering.
- Supports TX/RX queues with a maximum length of 4.
- Supports TCP checksum calculation and insertion.

10.2.2 Block Diagram

The block diagram of a single Ethernet module group is shown above, with each group connected to external PHY interfaces. Inside the module, there are TX and RX sides. The MAC handles Ethernet protocol-related tasks, while the MTL handles data buffering and distribution queue work at the transaction layer. The DMA is responsible for data exchange with the system bus, and CSRs at different levels are responsible for configuring registers in the corresponding modules.

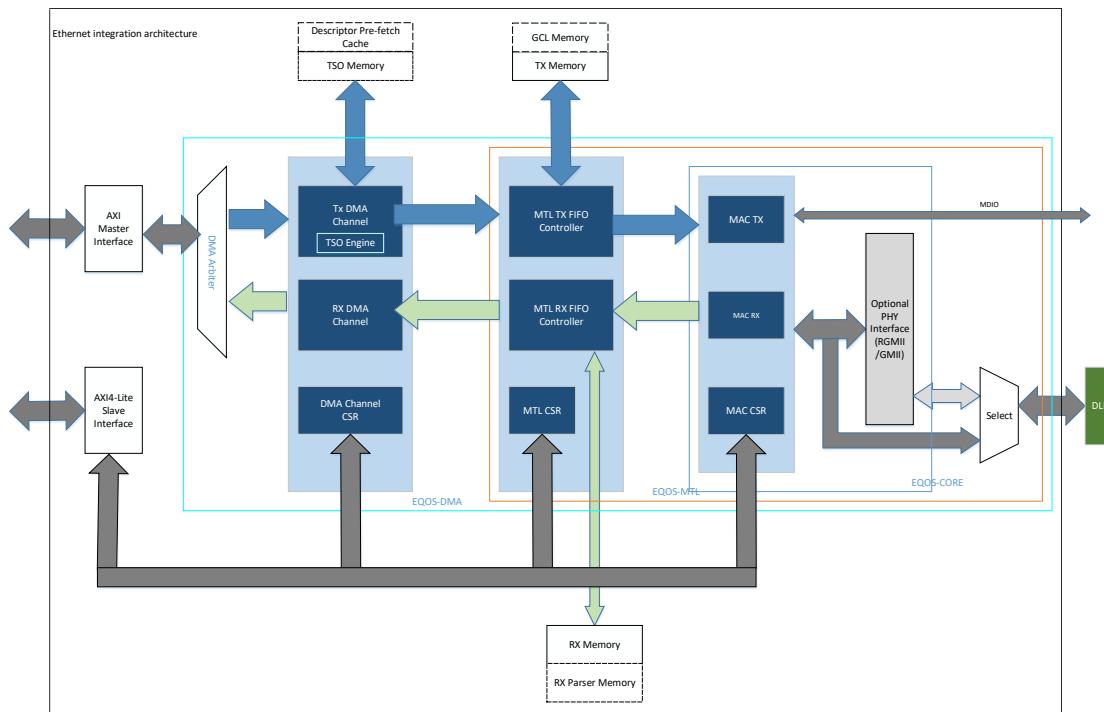


Figure 10-10 GMAC subsystem block diagram

10.2.3 Initialization Sequence

10.2.3.1 DMA Initialization

1. Write register DMA_Mode SWR, reset DMA
2. Poll DMA_Mode SWR=0, reset done
3. Write register DMA_SysBus_Mode
4. Create TX or RX descriptor
5. Configure Tx or Rx RING descriptor length, write register DMA_CH(#i)_TxDesc_Ring_Length
6. Initialize Tx or Rx descriptor address, write register DMA_CH(#i)_TxDesc_List_Address
7. Configure packet length, write register DMA_CH(#i)_Control
8. Enable interrupt, write register DMA_CH(#i)_Interrupt_Enable
9. Start Tx or Rx process, write register DMA_CH(#i)_TX_Control, DMA_CH(#i)_RX_Control

10.2.3.2 MTL Initialization

1. Set Tx or Rx arbitration strategy, write MTL_Operation_Mode
2. Configure mapping from queue to channel, write register MTL_RxQ_DMA_Map0 and MTL_RxQ_DMA_Map1

3. Initialize Tx buffering setting, write register MTL_TxQ0_Operation_Mode
4. Initialize Rx buffering setting, write register MTL_RxQ0_Operation_Mode

10.2.3.3 MAC Initialization

1. Configure mac address, write register MAC_Address0_Low and MAC_Address0_High
2. Configure MAC filtering, write register MAC_Packet_Filter
3. Configure MAC flow control, write register MAC_Q(#i)_Tx_Flow_Ctrl
4. Enable interrupt, write register MAC_Interrupt_Enable
5. Write register MAC_Configuration.TE, MAC_Configuration.RE, start MAC transmit or receive process

10.2.4 Timing Diagrams

RGMII Interface Timing

- TskewT represents the timing offset between the clock edge and data edge on the transmitting side, with an effective range of 0-0.5ns.
- TskewR represents the timing offset between the clock edge and data edge on the receiving side, with an effective range of 1-2.6ns

Table 10-8 RGMII Transmit Timing Requirement

Parameter	Description	MIN	TYP	MAX
T1	Transmit Clock Period		8ns	
T2	Transmit Data-to-Clock Output Skew	-600ps		600ps
T3	Transmit Data-to-Clock Output Setup	1.2ns	2ns	
T4	Transmit Data-to-Clock Output Hold	1.2ns	2ns	

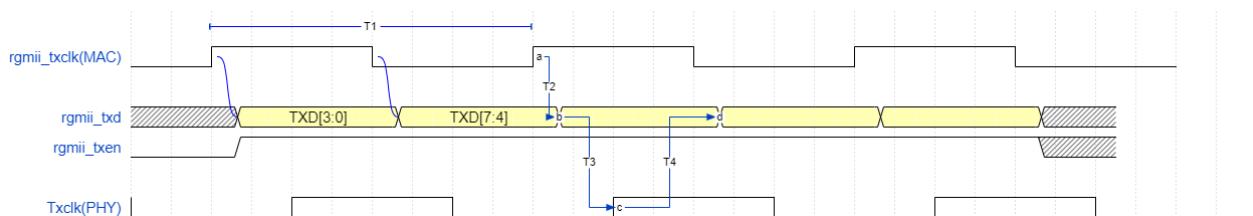


Figure 10-11 RGMII TX Timing diagram

Table 10-9 RGMII Receive Timing Requirement

Parameter	Description	MIN	TYP	MAX
T1	Receive Clock Period	-	8ns	-
T2	Receive Data-to-Clock Output Skew	-500ps	-	500ps
T3	Receive Data-to-Clock Output Setup	1ns	2ns	-

T4	Receive Data-to-Clock Output Hold	1ns	2ns	-
rgmii_rxclk(MAC)	T1	rgmii_rxv	rgmii_rx	RxD[3:0] RxD[7:4]

The diagram illustrates the RGMII RX timing requirements. It shows the reference clock rgmii_rxclk (MAC), receive valid rgmii_rxv, receive data rgmii_rx, and receive clock Rxclk(PHY). The timing intervals are labeled: T1 (reference clock period), T2 (delay from ref_clk high to txd[1:0] valid), T3 (setup time before ref_clk), and T4 (hold time after ref_clk).

Figure 10-12 RGMII RX Timing diagram

RMII Interface Timing

In the EIC7700, the RMII timing setup/hold requirements are shown as below:

Table 10-10 RMII Timing Requirement

Parameter	Description	MIN	TYP	MAX
T1	Reference Clock Period	-	20ns	-
T2	Delay Time, ref_clk high to txd[1:0] valid	1.2ns	-	12ns
T3	Setup Time, rxd[1:0] valid before ref_clk	4ns	-	-
T4	Hold Time, rxd[1:0] valid after ref_clk	2ns	-	-

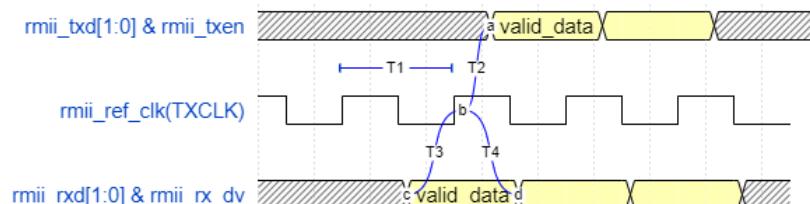


Figure 10-13 RMII Timing diagram

10.2.5 Register Description

10.2.5.1 Register Overview

GMAC 0 Base_addr:5040_0000

GMAC 1 Base_addr:5041_0000

Registers for Address Block: DWC_eqos_top_map/EQOS_MAC

Register	Offset	Description
MAC_Configuration	0x0	The MAC Configuration Register establishes the operating mode of the MAC.
MAC_Ext_Configuration	0x4	The MAC Extended Configuration Register establishes the operating mode of the MAC.

Register	Offset	Description
MAC_Packet_Filter	0x8	The MAC Packet Filter register contains the filter controls for receiving packets. Some of the controls...
MAC_Watchdog_Timeout	0xc	The Watchdog Timeout register controls the watchdog timeout for received packets.
MAC_Hash_Table_Reg0	0x10	The Hash Table Register 0 contains the first 32 bits of the hash table, when the width of the hash...
MAC_Hash_Table_Reg1	0x14	The Hash Table Register 1 contains the second 32 bits of the hash table. You can specify the width...
MAC_VLAN_Tag_Ctrl	0x50	This register is the redefined format of the MAC VLAN Tag Register. It is used for indirect addressing....
MAC_VLAN_Tag_Data	0x54	This register holds the read/write data for Indirect Access of the Per VLAN Tag registers. During...
MAC_VLAN_Tag_Filter(#i) (for i = 0; i <= 3)	0x54	This register contains VLAN Tag filter \${i} control information.
MAC_VLAN_Hash_Table	0x58	When VTHM bit of the MAC_VLAN_Tag register is set, the 16-bit VLAN Hash Table register is used for...
MAC_VLAN_Incl	0x60	The VLAN Tag Inclusion or Replacement register contains the VLAN tag for insertion or replacement...
MAC_Q0_Tx_Flow_Ctrl	0x70	The Flow Control register controls the generation and reception of the Control (Pause Command) packets...
MAC_Rx_Flow_Ctrl	0x90	The Receive Flow Control register controls the pausing of MAC Transmit based on the received Pause...
MAC_RxQ_Ctrl4	0x94	The Receive Queue Control 4 register controls the routing of unicast and multicast packets that...
MAC_RxQ_Ctrl0	0xa0	The Receive Queue Control 0 register controls the queue management in the MAC Receiver. Note: In...
MAC_RxQ_Ctrl1	0xa4	The Receive Queue Control 1 register controls the routing of multicast, broadcast, AV, DCB, and...
MAC_RxQ_Ctrl2	0xa8	This register controls the routing of tagged packets based on the USP (user Priority) field of the...
MAC_Interrupt_Status	0xb0	The Interrupt Status register contains the status of interrupts.
MAC_Interrupt_Enable	0xb4	The Interrupt Enable register contains the masks for generating the interrupts.
MAC_Rx_Tx_Status	0xb8	The Receive Transmit Status register contains the

Register	Offset	Description
		Receive and Transmit Error status.
MAC_PHYIF_Control_Status	0xf8	The PHY Interface Control and Status register indicates the status signals received by the SGMII,...
MAC_Version	0x110	The version register identifies the version of the DWC_ether_qos. This register contains two bytes:...
MAC_Debug	0x114	The Debug register provides the debug status of various MAC blocks.
MAC_HW_Feature0	0x11c	This register indicates the presence of first set of the optional features or functions of the DWC_ether_qos....
MAC_HW_Feature1	0x120	This register indicates the presence of second set of the optional features or functions of the...
MAC_HW_Feature2	0x124	This register indicates the presence of third set of the optional features or functions of the DWC_ether_qos....
MAC_HW_Feature3	0x128	This register indicates the presence of fourth set the optional features or functions of the DWC_ether_qos....
MAC_MDIO_Address	0x200	The MDIO Address register controls the management cycles to external PHY through a management...
MAC_MDIO_Data	0x204	The MDIO Data register stores the Write data to be written to the PHY register located at the address...
MAC_CSR_SW_Ctrl	0x230	This register contains software programmable controls for changing the CSR access response and status...
MAC_Address0_High	0x300	The MAC Address0 High register holds the upper 16 bits of the first 6-byte MAC address of the station....
MAC_Address0_Low	0x304	The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address of the...
MAC_Address(#i)_High (for i = 1; i <= 8)	(0x0008*i)+0x0300	The MAC Address1 High register holds the upper 16 bits of the second 6-byte MAC address of the station. If...
MAC_Address(#i)_Low (for i = 1; i <= 8)	(0x0008*i)+0x0304	The MAC Address1 Low register holds the lower 32 bits of the second 6-byte MAC address of the...
MAC_L3_L4_Control(#i) (for i = 0; i <= 3)	(0x0030*i)+0x0900	The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer...
MAC_Layer4_Address(#i) (for i = 0; i <= 3)	(0x0030*i)+0x0904	The MAC_Layer4_Address(#i), MAC_L3_L4_Control(#i), MAC_Layer3_Addr0_Reg(#i), MAC_Layer3_Addr1_Reg(#i),...

Register	Offset	Description
MAC_Layer3_Addr0_Reg(#i) (for i = 0; i <= 3)	(0x0030*i)+0x0910	For IPv4 packets, the Layer 3 Address 0 Register 0 register contains the 32-bit IP Source Address...
MAC_Layer3_Addr1_Reg(#i) (for i = 0; i <= 3)	(0x0030*i)+0x0914	For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address...
MAC_Layer3_Addr2_Reg(#i) (for i = 0; i <= 3)	(0x0030*i)+0x0918	The Layer 3 Address 2 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains...
MAC_Layer3_Addr3_Reg(#i) (for i = 0; i <= 3)	(0x0030*i)+0x091C	The Layer 3 Address 3 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains...
MAC_Indir_Access_Ctrl	0xa70	This register provides the Indirect Access control and status for MAC_<MSEL>_<AOFF> registers.
MAC_Indir_Access_Data	0xa74	This register holds the read/write data for Indirect Access of MAC_<MSEL>_<AOFF> registers.
MAC_TMRQ_Regs(#i) (for i = 0; i <= 7)	0xa74	This register contains the type, associated queue number and packet type (Preemption/Express) related...

Registers for Address Block: DWC_eqos_top_map/EQOS_MTL

Register	Offset	Description
MTL_Operation_Mode	0xc00	The Operation Mode register establishes the Transmit and Receive operating modes and commands.
MTL_Interrupt_Status	0xc20	The software driver (application) reads this register during interrupt service routine or polling...
MTL_RxQ_DMA_Map0	0xc30	The Receive Queue and DMA Channel Mapping 0 register is reserved in EQOS-CORE and EQOS-MTL...

Registers for Address Block: DWC_eqos_top_map/EQOS_MTL_Q0

Register	Offset	Description
MTL_TxQ0_Operation_Mode	0xd00	The Queue 0 Transmit Operation Mode register establishes the Transmit queue operating modes and...
MTL_TxQ0_Underflow	0xd04	The Queue 0 Underflow Counter register contains the counter for packets aborted because of Transmit...
MTL_TxQ0_Debug	0xd08	The Queue 0 Transmit Debug register gives the debug status of various blocks related to the Transmit...
MTL_TxQ0_ETS_Status	0xd14	The Queue 0 ETS Status register provides the average traffic transmitted in Queue 0.
MTL_TxQ0_Quantum_Weight	0xd18	The Queue 0 Quantum or Weights register contains the quantum value for Deficit Weighted Round Robin...

Register	Offset	Description
MTL_Q0_Interrupt_Control_Status	0xd2c	This register contains the interrupt enable and status bits for the queue 0 interrupts.
MTL_RxQ0_Operation_Mode	0xd30	The Queue 0 Receive Operation Mode register establishes the Receive queue operating modes and command. The...
MTL_RxQ0_Missed_Packet_Overflow_Cnt	0xd34	The Queue 0 Missed Packet and Overflow Counter register contains the counter for packets missed...
MTL_RxQ0_Debug	0xd38	The Queue 0 Receive Debug register gives the debug status of various blocks related to the Receive...
MTL_RxQ0_Control	0xd3c	The Queue Receive Control register controls the receive arbitration and passing of received packets...

Registers for Address Block: DWC_eqos_top_map/EQOS_MTL_Q1

Register	Offset	Description
MTL_TxQ(#i)_Operation_Mode (for i = 1; i <= 3)	(0x0040*i)+0x0D00	The Queue 1 Transmit Operation Mode register establishes the Transmit queue operating modes and...
MTL_TxQ(#i)_Underflow (for i = 1; i <= 3)	(0x0040*i)+0x0D04	The Queue 1 Underflow Counter register contains the counter for packets aborted because of Transmit...
MTL_TxQ(#i)_Debug (for i = 1; i <= 3)	(0x0040*i)+0x0D08	The Queue 1 Transmit Debug register gives the debug status of various blocks related to the Transmit...
MTL_TxQ(#i)_ETS_Status (for i = 1; i <= 3)	(0x0040*i)+0x0D14	The Queue 1 ETS Status register provides the average traffic transmitted in Queue 1.
MTL_TxQ(#i)_Quantum_Weight (for i = 1; i <= 3)	(0x0040*i)+0x0D18	The Queue 1 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted...
MTL_Q(#i)_Interrupt_Control_Status (for i = 1; i <= 3)	(0x0040*i)+0x0D2C	This register contains the interrupt enable and status bits for the queue 1 interrupts.
MTL_RxQ(#i)_Operation_Mode (for i = 1; i <= 3)	(0x0040*i)+0x0D30	The Queue 1 Receive Operation Mode register establishes the Receive queue operating modes and command. The...
MTL_RxQ(#i)_Missed_Packet_Overflow_Cnt (for i = 1; i <= 3)	(0x0040*i)+0x0D34	The Queue 1 Missed Packet and Overflow Counter register contains

Register	Offset	Description
		the counter for packets missed...
MTL_RxQ(#i)_Debug (for i = 1; i <= 3)	(0x0040*i)+0x0D38	The Queue 1 Receive Debug register gives the debug status of various blocks related to the Receive...
MTL_RxQ(#i)_Control (for i = 1; i <= 3)	(0x0040*i)+0x0D3C	The Queue Receive Control register controls the receive arbitration and passing of received packets...

Registers for Address Block: DWC_eqos_top_map/EQOS_DMA

Register	Offset	Description
DMA_Mode	0x1000	The Bus Mode register establishes the bus operating modes for the DMA.
DMA_SysBus_Mode	0x1004	The System Bus mode register controls the behavior of the AHB or AXI master. It mainly controls...
DMA_Interrupt_Status	0x1008	The application reads this Interrupt Status register during interrupt service routine or polling...
DMA_Debug_Status0	0x100c	The Debug Status 0 register gives the Receive and Transmit process status for DMA Channel 0-Channel...
DMA_Debug_Status1	0x1010	The Debug Status1 register gives the Receive and Transmit process status for DMA Channel 3-Channel...
AXI_LPI_Entry_Interval	0x1040	This register is used to control the AXI LPI entry interval.

Registers for Address Block: DWC_eqos_top_map/EQOS_DMA_CH0

Register	Offset	Description
DMA_CH(#i)_Control (for i = 0; i <= 3)	(0x0080*i)+0x1100	The DMA Channeli Control register specifies the MSS value for segmentation, length to skip between...
DMA_CH(#i)_Tx_Control (for i = 0; i <= 3)	(0x0080*i)+0x1104	The DMA Channeli Transmit Control register controls the Tx features such as PBL, TCP segmentation,...
DMA_CH(#i)_Rx_Control (for i = 0; i <= 3)	(0x0080*i)+0x1108	The DMA Channeli Receive Control register controls the Rx features such as PBL, buffer size, and...
DMA_CH(#i)_TxDesc_List_HAddress (for i = 0; i <= 3)	(0x0080*i)+0x1110	The Channeli Tx Descriptor List HAddress register has the higher 8 or 16 bits of the start address...
DMA_CH(#i)_TxDesc_List_Address (for i = 0; i <= 3)	(0x0080*i)+0x1114	The Channeli Tx Descriptor List Address register points the DMA to the start of Transmit descriptor...
DMA_CH(#i)_RxDesc_List_HAddress	(0x0080*i)+0x1118	The Channeli Rx Descriptor List HAddress register has the higher 8

Register	Offset	Description
(for i = 0; i <= 3)		or 16 bits of the start address...
DMA_CH(#i)_RxDesc_List_Address (for i = 0; i <= 3)	(0x0080*i)+0x111C	The Channeli Rx Descriptor List Address register points the DMA to the start of Receive descriptor...
DMA_CH(#i)_TxDesc_Tail_Pointer (for i = 0; i <= 3)	(0x0080*i)+0x1120	The Channeli Tx Descriptor Tail Pointer register points to an offset from the base and indicates...
DMA_CH(#i)_RxDesc_Tail_Pointer (for i = 0; i <= 3)	(0x0080*i)+0x1128	The Channeli Rx Descriptor Tail Pointer Points to an offset from the base and indicates the location...
DMA_CH(#i)_TxDesc_Ring_Length (for i = 0; i <= 3)	(0x0080*i)+0x112C	The Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.
DMA_CH(#i)_Rx_Control2 (for i = 0; i <= 3)	(0x0080*i)+0x1130	The Channeli Receive Control register controls the Rx features such as Rx Descriptor Ring Length...
DMA_CH(#i)_Interrupt_Enable (for i = 0; i <= 3)	(0x0080*i)+0x1134	The Channeli Interrupt Enable register enables the interrupts reported by the Status register.
DMA_CH(#i)_Rx_Interrupt_Watchdog_Timer (for i = 0; i <= 3)	(0x0080*i)+0x1138	The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt...
DMA_CH(#i)_Current_App_TxDesc (for i = 0; i <= 3)	(0x0080*i)+0x1144	The Channeli Current Application Transmit Descriptor register points to the current Transmit descriptor...
DMA_CH(#i)_Current_App_RxDesc (for i = 0; i <= 3)	(0x0080*i)+0x114C	The Channeli Current Application Receive Descriptor register points to the current Receive descriptor...
DMA_CH(#i)_Current_App_TxBuffer_H (for i = 0; i <= 3)	(0x0080*i)+0x1150	The Channeli Current Application Transmit Buffer Address High register has the higher 8 or 16 bits...
DMA_CH(#i)_Current_App_TxBuffer (for i = 0; i <= 3)	(0x0080*i)+0x1154	The Channeli Current Application Transmit Buffer Address register points to the current Tx buffer...
DMA_CH(#i)_Current_App_RxBuffer_H (for i = 0; i <= 3)	(0x0080*i)+0x1158	The Channeli Current Application Receive Buffer Address High register has the higher 8 or 16 bits...
DMA_CH(#i)_Current_App_RxBuffer (for i = 0; i <= 3)	(0x0080*i)+0x115C	The Channeli Current Application Receive Buffer Address register points to the current Rx buffer...
DMA_CH(#i)_Status (for i = 0; i <= 3)	(0x0080*i)+0x1160	The software driver (application) reads the Status register during

Register	Offset	Description
		interrupt service routine or...
DMA_CH(#i)_Miss_Frame_Cnt (for i = 0; i <= 3)	(0x0080*i)+0x1164	This register has the number of packet counter that got dropped by the DMA either due to Bus Error...

10.2.5.2 Register Detail

MAC_Configuration

Description: The MAC Configuration Register establishes the operating mode of the MAC.

Size: 32 bits

Offset: 0x0

31	30: 28	2	26: 24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6: 5	4	3:2	1	0
Rs vd	SA RC	IP C	IPG CE	GPSL KP	S2 T	CS S	AC D	W E	B D	J E	J S	P M	FE M	D M	L M	ECRS FD	D O	DC RS	D R	Reserv ed_7	B L	D C	PREL EN	T E	R E

Fields for Register: MAC_Configuration

Bits	Name	Memory Access	Description
31			Reserved Field: Yes
30:28	SARC	R/W	<p>Source Address Insertion or Replacement Control</p> <p>This field controls the source address insertion or replacement for all transmitted packets. Bit 30 specifies which MAC Address register (0 or 1) is used for source address insertion or replacement based on the values of Bits[29:28]:</p> <p>2'b0x:</p> <p>The mti_sa_ctrl_i and ati_sa_ctrl_i input signals control the SA field generation.</p> <p>2'b10:</p> <p>If Bit 30 is set to 0, the MAC inserts the content of the MAC Address 0 registers in the SA field of all transmitted packets.</p> <p>If Bit 30 is set to 1 and the Enable MAC Address Register 1 option is selected while configuring the IP, the MAC inserts the content of the MAC Address 1 registers in the SA field of all transmitted packets.</p> <p>2'b11:</p> <p>If Bit 30 is set to 0, the MAC replaces the content of the MAC Address 0 registers in the SA field of all transmitted packets.</p> <p>If Bit 30 is set to 1 and the MAC Address Register 1 is enabled, the MAC replaces the content of the MAC Address 1 registers in the</p>

Bits	Name	Memory Access	Description
			<p>SA field of all transmitted packets.</p> <p>Note: Changes to this field take effect only on the start of a packet. If you write to this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (SA_CTRL_IN): mti_sa_ctrl_i and ati_sa_ctrl_i input signals control the SA field generation 0x2 (MAC0_INS_SA): Contents of MAC Addr-0 inserted in SA field 0x3 (MAC0 REP_SA): Contents of MAC Addr-0 replaces SA field 0x6 (MAC1_INS_SA): Contents of MAC Addr-1 inserted in SA field 0x7 (MAC1 REP_SA): Contents of MAC Addr-1 replaces SA field <p>Value After Reset:0x0</p>
27	IPC	R/W	<p>Checksum Offload</p> <p>When set, this bit enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled.</p> <p>The Layer 3 and Layer 4 Packet Filter and Enable Split Header features automatically selects the IPC Full Checksum Offload Engine on the Receive side. When any of these features are enabled, you must set the IPC bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): IP header/payload checksum checking is disabled 0x1 (ENABLE): IP header/payload checksum checking is enabled <p>Value After Reset:0x0</p>
26:24	IPG	R/W	<p>Inter-Packet Gap</p> <p>These bits control the minimum IPG between packets during transmission. This range of minimum IPG is valid in full-duplex mode.</p> <p>In the half-duplex mode, the minimum IPG can be configured only for 64-bit times (IPG = 100). Lower values are not considered.</p> <p>When a JAM pattern is being transmitted because of backpressure activation, the MAC does not consider the minimum IPG.</p> <p>This function (IPG less than 96 bit times) is valid only when EIPGEN bit in MAC_Ext_Configuration register is reset. When EIPGEN is set, then the minimum IPG (greater than 96 bit times) is controlled as per the description given in EIPG field in MAC_Ext_Configuration register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (IPG96): 96 bit times IPG

Bits	Name	Memory Access	Description
			<p>0x1 (IPG88): 88 bit times IPG 0x2 (IPG80): 80 bit times IPG 0x3 (IPG72): 72 bit times IPG 0x4 (IPG64): 64 bit times IPG 0x5 (IPG56): 56 bit times IPG 0x6 (IPG48): 48 bit times IPG 0x7 (IPG40): 40 bit times IPG</p> <p>Value After Reset:0x0</p>
23	GPSLCE	R/W	<p>Giant Packet Size Limit Control Enable</p> <p>When this bit is set, the MAC considers the value in GPSL field in MAC_Ext_Configuration register to declare a received packet as Giant packet. This field must be programmed to more than 1,518 bytes. Otherwise, the MAC considers 1,518 bytes as giant packet limit.</p> <p>When this bit is reset, the MAC considers a received packet as Giant packet when its size is greater than 1,518 bytes (1522 bytes for tagged packet).</p> <p>The watchdog timeout limit, Jumbo Packet Enable and 2K Packet Enable have higher precedence over this bit, that is the MAC considers a received packet as Giant packet when its size is greater than 9,018 bytes (9,022 bytes for tagged packet) with Jumbo Packet Enabled and greater than 2,000 bytes with 2K Packet Enabled. The watchdog timeout, if enabled, terminates the received packet when watchdog limit is reached. Therefore, the programmed giant packet limit should be less than the watchdog limit to get the giant packet status.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Giant Packet Size Limit Control is disabled 0x1 (ENABLE): Giant Packet Size Limit Control is enabled <p>Value After Reset:0x0</p>
22	S2KP	R/W	<p>IEEE 802.3as Support for 2K Packets</p> <p>When this bit is set, the MAC considers all packets with up to 2,000 bytes length as normal packets. When the JE bit is not set, the MAC considers all received packets of size more than 2K bytes as Giant packets.</p> <p>When this bit is reset and the JE bit is not set, the MAC considers all received packets of size more than 1,518 bytes (1,522 bytes for tagged) as giant packets. For more information about how the setting of this bit and the JE bit impact the Giant packet status, see the Table, Giant Packet Status based on S2KP and JE Bits.</p> <p>Note: When the JE bit is set, setting this bit has no effect on the giant packet status.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (DISABLE): Support upto 2K packet is disabled 0x1 (ENABLE): Support upto 2K packet is Enabled Value After Reset:0x0</p>
21	CST	R/W	<p>CRC stripping for Type packets When this bit is set, the last four bytes (FCS) of all packets of Ether type (type field greater than 1,536) are stripped and dropped before forwarding the packet to the application. Note: For information about how the settings of the ACS bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bits.</p> <p>Values:</p> <p>0x0 (DISABLE): CRC stripping for Type packets is disabled 0x1 (ENABLE): CRC stripping for Type packets is enabled Value After Reset:0x0</p>
20	ACS	R/W	<p>Automatic Pad or CRC Stripping When this bit is set, the MAC strips the Pad or FCS field on the incoming packets only if the value of the length field is less than 1,536 bytes. All received packets with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field. When this bit is reset, the MAC passes all incoming packets to the application, without any modification. Note: For information about how the settings of CST bit and this bit impact the packet length, see the Table, Packet Length based on the CST and ACS Bit .</p> <p>Values:</p> <p>0x0 (DISABLE): Automatic Pad or CRC Stripping is disabled 0x1 (ENABLE): Automatic Pad or CRC Stripping is enabled Value After Reset:0x0</p>
19	WD	R/W	<p>Watchdog Disable When this bit is set, the MAC disables the watchdog timer on the receiver. The MAC can receive packets of up to 16,383 bytes. When this bit is reset, the MAC does not allow more than 2,048 bytes (10,240 if JE is set high) of the packet being received. The MAC cuts off any bytes received after 2,048 bytes.</p> <p>Values:</p> <p>0x0 (ENABLE): Watchdog is enabled 0x1 (DISABLE): Watchdog is disabled</p>

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
18	BE	R/W	<p>Packet Burst Enable</p> <p>When this bit is set, the MAC allows packet bursting during transmission in the GMII half-duplex mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Packet Burst is disabled 0x1 (ENABLE): Packet Burst is enabled <p>Value After Reset:0x0</p>
17	JD	R/W	<p>Jabber Disable</p> <p>When this bit is set, the MAC disables the jabber timer on the transmitter. The MAC can transfer packets of up to 16,383 bytes.</p> <p>When this bit is reset, if the application sends more than 2,048 bytes of data (10,240 if JE is set high) during transmission, the MAC does not send rest of the bytes in that packet.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): Jabber is enabled 0x1 (DISABLE): Jabber is disabled <p>Value After Reset:0x0</p>
16	JE	R/W	<p>Jumbo Packet Enable</p> <p>When this bit is set, the MAC allows jumbo packets of 9,018 bytes (9,022 bytes for VLAN tagged packets) without reporting a giant packet error in the Rx packet status.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Jumbo packet is disabled 0x1 (ENABLE): Jumbo packet is enabled <p>Value After Reset:0x0</p>
15	PS	R/W	<p>Port Select</p> <p>This bit selects the Ethernet line speed. This bit, along with Bit 14, selects the exact line speed. In the 10/100 Mbps-only (always 1) or 1000 Mbps-only (always 0) configurations, this bit is read-only (RO) with appropriate value. In default 10/100/1000 Mbps configurations, this bit is read-write (R/W). The mac_speed_o[1] signal reflects the value of this bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_1000_2500M): For 1000 or 2500 Mbps operations 0x1 (M_10_100M): For 10 or 100 Mbps operations <p>Value After Reset:0x0</p>

Bits	Name	Memory Access	Description
14	FES	R/W	<p>Speed</p> <p>This bit selects the speed mode. The mac_speed_o[0] signal reflects the value of this bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_10_1000M): 10 Mbps when PS bit is 1 and 1 Gbps when PS bit is 0 0x1 (M_100_2500M): 100 Mbps when PS bit is 1 and 2.5 Gbps when PS bit is 0 <p>Value After Reset:0x0</p>
13	DM	R/W	<p>Duplex Mode</p> <p>When this bit is set, the MAC operates in the full-duplex mode in which it can transmit and receive simultaneously. This bit is RO with default value of 1'b1 in the full-duplex-only configurations.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (HDUPLX): Half-duplex mode 0x1 (FDUPLX): Full-duplex mode <p>Value After Reset:0x0</p>
12	LM	R/W	<p>Loopback Mode</p> <p>When this bit is set, the MAC operates in the loopback mode at GMII or MII. The (G)MII Rx clock input (clk_rx_i) is required for the loopback to work properly. This is because the Tx clock is not internally looped back.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Loopback is disabled 0x1 (ENABLE): Loopback is enabled <p>Value After Reset:0x0</p>
11	ECRSFD	R/W	<p>Enable Carrier Sense Before Transmission in Full-Duplex Mode</p> <p>When this bit is set, the MAC transmitter checks the CRS signal before packet transmission in the full-duplex mode. The MAC starts the transmission only when the CRS signal is low.</p> <p>When this bit is reset, the MAC transmitter ignores the status of the CRS signal.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): ECRSFD is disabled 0x1 (ENABLE): ECRSFD is enabled <p>Value After Reset:0x0</p>
10	DO	R/W	Disable Receive Own

Bits	Name	Memory Access	Description
			<p>When this bit is set, the MAC disables the reception of packets when the gmii_txen_o is asserted in the half-duplex mode. When this bit is reset, the MAC receives all packets given by the PHY.</p> <p>This bit is not applicable in the full-duplex mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): Enable Receive Own 0x1 (DISABLE): Disable Receive Own <p>Value After Reset:0x0</p>
9	DCRS	R/W	<p>Disable Carrier Sense During Transmission</p> <p>When this bit is set, the MAC transmitter ignores the (G)MII CRS signal during packet transmission in the half-duplex mode. As a result, no errors are generated because of Loss of Carrier or No Carrier during transmission.</p> <p>When this bit is reset, the MAC transmitter generates errors because of Carrier Sense. The MAC can even abort the transmission.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): Enable Carrier Sense During Transmission 0x1 (DISABLE): Disable Carrier Sense During Transmission <p>Value After Reset:0x0</p>
8	DR	R/W	<p>Disable Retry</p> <p>When this bit is set, the MAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the MAC ignores the current packet transmission and reports a Packet Abort with excessive collision error in the Tx packet status.</p> <p>When this bit is reset, the MAC retries based on the settings of the BL field. This bit is applicable only in the half-duplex mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): Enable Retry 0x1 (DISABLE): Disable Retry <p>Value After Reset:0x0</p>
7	Reserved_7	R	<p>Reserved.</p> <p>Value After Reset:0x0</p>
6:5	BL	R/W	<p>Back-Off Limit</p> <p>The back-off limit determines the random integer number (r) of slot time delays (4,096 bit times for 1000/2500 Mbps; 512 bit times for 10/100 Mbps) for which the MAC waits before rescheduling a transmission attempt during retries after a collision. n = retransmission attempt.</p>

Bits	Name	Memory Access	Description
			<p>The random integer r takes the value in the range $0 \leq r < 2^k$</p> <p>This bit is applicable only in the half-duplex mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (MIN_N_10): $k = \min(n, 10)$ 0x1 (MIN_N_8): $k = \min(n, 8)$ 0x2 (MIN_N_4): $k = \min(n, 4)$ 0x3 (MIN_N_1): $k = \min(n, 1)$ <p>Value After Reset:0x0</p>
4	DC	R/W	<p>Deferral Check</p> <p>When this bit is set, the deferral check function is enabled in the MAC. The MAC issues a Packet Abort status, along with the excessive deferral error bit set in the Tx packet status, when the Tx state machine is deferred for more than 24,288 bit times in 10 or 100 Mbps mode.</p> <p>If the MAC is configured for 1000/2500 Mbps operation, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit, but it is prevented because of an active carrier sense signal (CRS) on GMII or MII.</p> <p>The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and the CRS signal becomes inactive, the transmitter transmits and collision happens. Because of collision, the transmitter needs to back off and then defer again after back off completion. In such a scenario, the deferral timer is reset to 0, and it is restarted.</p> <p>When this bit is reset, the deferral check function is disabled and the MAC defers until the CRS signal goes inactive.</p> <p>This bit is applicable only in the half-duplex mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Deferral check function is disabled 0x1 (ENABLE): Deferral check function is enabled <p>Value After Reset:0x0</p>
3:2	PRELEN	R/W	<p>Preamble Length for Transmit packets</p> <p>These bits control the number of preamble bytes that are added to the beginning of every Tx packet. The preamble reduction occurs only when the MAC is operating in the full-duplex mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_7BYTES): 7 bytes of preamble 0x1 (M_5BYTES): 5 bytes of preamble 0x2 (M_3BYTES): 3 bytes of preamble

Bits	Name	Memory Access	Description
			<p>0x3 (RESERVED): Reserved Value After Reset:0x0</p>
1	TE	R/W	<p>Transmitter Enable When this bit is set, the Tx state machine of the MAC is enabled for transmission on the GMII or MII interface. When this bit is reset, the MAC Tx state machine is disabled after it completes the transmission of the current packet. The Tx state machine does not transmit any more packets. Values: 0x0 (DISABLE): Transmitter is disabled 0x1 (ENABLE): Transmitter is enabled Value After Reset:0x0</p>
0	RE	R/W	<p>Receiver Enable When this bit is set, the Rx state machine of the MAC is enabled for receiving packets from the GMII or MII interface. When this bit is reset, the MAC Rx state machine is disabled after it completes the reception of the current packet. The Rx state machine does not receive any more packets from the GMII or MII interface. Values: 0x0 (DISABLE): Receiver is disabled 0x1 (ENABLE): Receiver is enabled Value After Reset:0x0</p>

MAC_Ext_Configuration

Description: The MAC Extended Configuration Register establishes the operating mode of the MAC.

Size:32 bits

Offset:0x4

31:30	29:25	24	23	22:19	18	17	16	15:14	13:0
Rsvd	EIPG	EIPGEN	Reserved_23	Rsvd	USP	SPEN	DCRCC	Reserved_15_14	GPSL

Fields for Register: MAC_Ext_Configuration

Bits	Name	Memory Access	Description
31:30			Reserved Field: Yes
29:25	EIPG	R/W	Extended Inter-Packet Gap

Bits	Name	Memory Access	Description
			<p>The value in this field is applicable when the EIPGEN bit is set. This field (as Most Significant bits), along with IPG field in MAC_Configuration register, gives the minimum IPG greater than 96 bit times in steps of 8 bit times: {EIPG, IPG}</p> <p>8'h00 - 104 bit times 8'h01 - 112 bit times 8'h02 - 120 bit times ----- 8'hFF - 2144 bit times</p> <p>Value After Reset:0x0</p>
24	EIPGEN	R/W	<p>Extended Inter-Packet Gap Enable</p> <p>When this bit is set, the MAC interprets EIPG field and IPG field in MAC_Configuration register together as minimum IPG greater than 96 bit times in steps of 8 bit times.</p> <p>When this bit is reset, the MAC ignores EIPG field and interprets IPG field in MAC_Configuration register as minimum IPG less than or equal to 96 bit times in steps of 8 bit times.</p> <p>Note: Enable the extended Inter-Packet Gap feature only when operating in Full-Duplex mode. There might be undesirable effects on back-pressure function and frame transmission if it is enabled in Half-Duplex mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Extended Inter-Packet Gap is disabled 0x1 (ENABLE): Extended Inter-Packet Gap is enabled <p>Value After Reset:0x0</p>
23	Reserved_23	R	<p>Reserved.</p> <p>Value After Reset:0x0</p>
22:19			Reserved Field: Yes
18	USP	R/W	<p>Unicast Slow Protocol Packet Detect</p> <p>When this bit is set, the MAC detects the Slow Protocol packets with unicast address of the station specified in the MAC_Address0_High and MAC_Address0_Low registers. The MAC also detects the Slow Protocol packets with the Slow Protocols multicast address (01-80-C2-00-00-02).</p> <p>When this bit is reset, the MAC detects only Slow Protocol packets with the Slow Protocol multicast address specified in the IEEE 802.3-2015, Section 5.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (DISABLE): Unicast Slow Protocol Packet Detection is disabled</p> <p>0x1 (ENABLE): Unicast Slow Protocol Packet Detection is enabled</p> <p>Value After Reset:0x0</p>
17	SPEN	R/W	<p>Slow Protocol Detection Enable</p> <p>When this bit is set, MAC processes the Slow Protocol packets (Ether Type 0x8809) and provides the Slow Protocol Sub-Type and Code fields in Rx status.</p> <p>When this bit is reset, the MAC forwards all error-free Slow Protocol packets to the application. The MAC considers such packets as normal Type packets.</p> <p>Values:</p> <p>0x0 (DISABLE): Slow Protocol Detection is disabled</p> <p>0x1 (ENABLE): Slow Protocol Detection is enabled</p> <p>Value After Reset:0x0</p>
16	DCRCC	R/W	<p>Disable CRC Checking for Received Packets</p> <p>When this bit is set, the MAC receiver does not check the CRC field in the received packets. When this bit is reset, the MAC receiver always checks the CRC field in the received packets.</p> <p>Values:</p> <p>0x0 (ENABLE): CRC Checking is enabled</p> <p>0x1 (DISABLE): CRC Checking is disabled</p> <p>Value After Reset:0x0</p>
15:14	Reserved_15_14	R	<p>Reserved.</p> <p>Value After Reset:0x0</p>
13:0	GPSL	R/W	<p>Giant Packet Size Limit</p> <p>If the received packet size is greater than the value programmed in this field in units of bytes, the MAC declares the received packet as Giant packet. The value programmed in this field must be greater than or equal to 1,518 bytes. Any other programmed value is considered as 1,518 bytes.</p> <p>For VLAN tagged packets, the MAC adds 4 bytes to the programmed value. When the Enable Double VLAN Processing option is selected, the MAC adds 8 bytes to the programmed value for double VLAN tagged packets. The value in this field is applicable when the GPSLCE bit is set in MAC_Configuration register.</p> <p>Value After Reset:0x0</p>

MAC_Packet_Filter

Description: The MAC Packet Filter register contains the filter controls for receiving packets. Some of the controls from this register go to the address check block of the MAC which performs the first level of address filtering. The second level of filtering is performed on the incoming packet based on other controls such as Pass Bad Packets and Pass Control Packets.

Size: 32 bits

Offset: 0x8

31	30:22	21	20	19:17	16	15:11	10	9	8	7:6	5	4	3	2	1	0
R A	Reserved_30_22	DNT U	IPF E	Reserved_19_17	VTF E	Reserved_15_11	HP F	SA F	S A I F	P C F	D B F	P M	D A I F	H M C	H U C	P R

Fields for Register: MAC_Packet_Filter

Bits	Name	Memory Access	Description
31	RA	R/W	<p>Receive All</p> <p>When this bit is set, the MAC Receiver module passes all received packets to the application, irrespective of whether they pass the address filter or not. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bit in the Rx Status Word.</p> <p>When this bit is reset, the Receiver module passes only those packets to the application that pass the SA or DA address filter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive All is disabled 0x1 (ENABLE): Receive All is enabled <p>Value After Reset: 0x0</p>
30:22	Reserved_30_22	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
21	DNTU	R/W	<p>Drop Non-TCP/UDP over IP Packets</p> <p>When this bit is set, the MAC drops the non-TCP or UDP over IP packets. The MAC forward only those packets that are processed by the Layer 4 filter. When this bit is reset, the MAC forwards all non-TCP or UDP over IP packets.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (FWD): Forward Non-TCP/UDP over IP Packets 0x1 (DROP): Drop Non-TCP/UDP over IP Packets <p>Value After Reset: 0x0</p>
20	IPFE	R/W	Layer 3 and Layer 4 Filter Enable

Bits	Name	Memory Access	Description
			<p>When this bit is set, the MAC drops packets that do not match the enabled Layer 3 and Layer 4 filters. If Layer 3 or Layer 4 filters are not enabled for matching, this bit does not have any effect.</p> <p>When this bit is reset, the MAC forwards all packets irrespective of the match status of the Layer 3 and Layer 4 fields.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Layer 3 and Layer 4 Filters are disabled 0x1 (ENABLE): Layer 3 and Layer 4 Filters are enabled <p>Value After Reset: 0x0</p>
19:17	Reserved_19_17	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
16	VTFE	R/W	<p>VLAN Tag Filter Enable</p> <p>When this bit is set, the MAC drops the VLAN tagged packets that do not match the VLAN Tag. When this bit is reset, the MAC forwards all packets irrespective of the match status of the VLAN Tag.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): VLAN Tag Filter is disabled 0x1 (ENABLE): VLAN Tag Filter is enabled <p>Value After Reset: 0x0</p>
15:11	Reserved_15_11	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
10	HPF	R/W	<p>Hash or Perfect Filter</p> <p>When this bit is set, the address filter passes a packet if it matches either the perfect filtering or hash filtering as set by the HMC or HUC bit.</p> <p>When this bit is reset and the HUC or HMC bit is set, the packet is passed only if it matches the Hash filter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Hash or Perfect Filter is disabled 0x1 (ENABLE): Hash or Perfect Filter is enabled <p>Value After Reset: 0x0</p>
9	SAF	R/W	<p>Source Address Filter Enable</p> <p>When this bit is set, the MAC compares the SA field of the received packets with the values programmed in the enabled SA registers. If the comparison fails, the MAC drops the packet.</p>

Bits	Name	Memory Access	Description
			<p>When this bit is reset, the MAC forwards the received packet to the application with updated SAF bit of the Rx Status depending on the SA address comparison.</p> <p>Note: According to the IEEE specification, Bit 47 of the SA is reserved. However, in DWC_ether_qos, the MAC compares all 48 bits. The software driver should take this into consideration while programming the MAC address registers for SA.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): SA Filtering is disabled 0x1 (ENABLE): SA Filtering is enabled <p>Value After Reset: 0x0</p>
8	SAIF	R/W	<p>SA Inverse Filtering</p> <p>When this bit is set, the Address Check block operates in the inverse filtering mode for SA address comparison. If the SA of a packet matches the values programmed in the SA registers, it is marked as failing the SA Address filter.</p> <p>When this bit is reset, if the SA of a packet does not match the values programmed in the SA registers, it is marked as failing the SA Address filter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): SA Inverse Filtering is disabled 0x1 (ENABLE): SA Inverse Filtering is enabled <p>Value After Reset: 0x0</p>
7:6	PCF	R/W	<p>Pass Control Packets</p> <p>These bits control the forwarding of all control packets (including unicast and multicast Pause packets).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (FLTR_ALL): MAC filters all control packets from reaching the application 0x1 (FW_XCPT_PAU): MAC forwards all control packets except Pause packets to the application even if they fail the Address filter 0x2 (FW_ALL): MAC forwards all control packets to the application even if they fail the Address filter 0x3 (FW_PASS): MAC forwards the control packets that pass the Address filter <p>Value After Reset: 0x0</p>
5	DBF	R/W	Disable Broadcast Packets

Bits	Name	Memory Access	Description
			<p>When this bit is set, the AFM module blocks all the incoming broadcast packets. In addition, it overrides all other filter settings.</p> <p>When this bit is reset, the AFM module passes all received broadcast packets.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): Enable Broadcast Packets 0x1 (DISABLE): Disable Broadcast Packets <p>Value After Reset: 0x0</p>
4	PM	R/W	<p>Pass All Multicast</p> <p>When this bit is set, it indicates that all the received packets with a multicast destination address (first bit in the destination address field is '1') are passed. When this bit is reset, filtering of multicast packet depends on HMC bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Pass All Multicast is disabled 0x1 (ENABLE): Pass All Multicast is enabled <p>Value After Reset: 0x0</p>
3	DAIF	R/W	<p>DA Inverse Filtering</p> <p>When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast packets. When this bit is reset, normal filtering of packets is performed.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): DA Inverse Filtering is disabled 0x1 (ENABLE): DA Inverse Filtering is enabled <p>Value After Reset: 0x0</p>
2	HMC	R/W	<p>Hash Multicast</p> <p>When this bit is set, the MAC performs the destination address filtering of received multicast packets according to the hash table.</p> <p>When this bit is reset, the MAC performs the perfect destination address filtering for multicast packets, that is, it compares the DA field with the values programmed in DA registers.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Hash Multicast is disabled 0x1 (ENABLE): Hash Multicast is enabled <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
1	HUC	R/W	<p>Hash Unicast</p> <p>When this bit is set, the MAC performs the destination address filtering of unicast packets according to the hash table.</p> <p>When this bit is reset, the MAC performs a perfect destination address filtering for unicast packets, that is, it compares the DA field with the values programmed in DA registers.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Hash Unicast is disabled 0x1 (ENABLE): Hash Unicast is enabled <p>Value After Reset: 0x0</p>
0	PR	R/W	<p>Promiscuous Mode</p> <p>When this bit is set, the Address Filtering module passes all incoming packets irrespective of the destination or source address. The SA or DA Filter Fails status bits of the Rx Status Word are always cleared when PR is set.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Promiscuous Mode is disabled 0x1 (ENABLE): Promiscuous Mode is enabled <p>Value After Reset: 0x0</p>

MAC_Watchdog_Timeout

Description: The Watchdog Timeout register controls the watchdog timeout for received packets.

Size: 32 bits

Offset: 0xc

31:9	8	7:4	3:0
Reserved_31_9	PWE	Reserved_7_4	WTO

Fields for Register: MAC_Watchdog_Timeout

Bits	Name	Memory Access	Description
31:9	Reserved_31_9	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
8	PWE	R/W	<p>Programmable Watchdog Enable</p> <p>When this bit is set and the WD bit of the MAC_Configuration register is reset, the WTO field is used as watchdog timeout for a received</p>

Bits	Name	Memory Access	Description
			<p>packet. When this bit is cleared, the watchdog timeout for a received packet is controlled by setting of WD and JE bits in MAC_Configuration register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Programmable Watchdog is disabled 0x1 (ENABLE): Programmable Watchdog is enabled <p>Value After Reset: 0x0</p>
7:4	Reserved_7_4	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
3:0	WTO	R/W	<p>Watchdog Timeout</p> <p>When the PWE bit is set and the WD bit of the MAC_Configuration register is reset, this field is used as watchdog timeout for a received packet. If the length of a received packet exceeds the value of this field, such packet is terminated and declared as an error packet.</p> <p>Note: When the PWE bit is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE 802.3-specified valid tagged packets are declared as error packets and then dropped.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_2KBYTES): 2 KB 0x1 (M_3KBYTES): 3 KB 0x2 (M_4KBYTES): 4 KB 0x3 (M_5KBYTES): 5 KB 0x4 (M_6KBYTES): 6 KB 0x5 (M_7KBYTES): 7 KB 0x6 (M_8KBYTES): 8 KB 0x7 (M_9KBYTES): 9 KB 0x8 (M_10KBYTES): 10 KB 0x9 (M_11KBYTES): 11 KB 0xa (M_12KBYTES): 12 KB 0xb (M_13KBYTES): 13 KB 0xc (M_14KBYTES): 14 KB 0xd (M_15KBYTES): 15 KB 0xe (M_16383BYTES): 16383 Bytes 0xf (RESERVED): Reserved <p>Value After Reset: 0x0</p>

MAC_Hash_Table_Reg0

Description: The Hash Table Register 0 contains the first 32 bits of the hash table, when the width of the hash table is 128 or 256 bits. The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5. The hash value of the destination address is calculated in the following way:

Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

Perform bitwise reversal for the value obtained in Step 1.

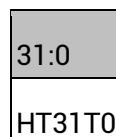
Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values. If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

Size: 32 bits

Offset: 0x10



Fields for Register: MAC_Hash_Table_Reg0

Bits	Name	Memory Access	Description
31:0	HT31T0	R/W	<p>MAC Hash Table First 32 Bits</p> <p>This field contains the first 32 Bits [31:0] of the Hash table.</p> <p>Value After Reset: 0x0</p>

MAC_Hash_Table_Reg1

Description: The Hash Table Register 1 contains the second 32 bits of the hash table. The Hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming packet is passed through the CRC logic and the upper six (seven in 128-bit Hash or eight in 256-bit Hash) bits of the CRC register are used to index the content of the Hash table. The most significant bits determines the register to be used (Hash Table Register X), and the least significant five bits determine the bit within the register. For example, a hash value of 6'b100000 (in 64-bit Hash) selects Bit 0 of the Hash Table Register 1, a value of 7b'1110000 (in 128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5. The hash value of the destination address is calculated in the following way:

128-bit Hash) selects Bit 16 of the Hash Table Register 3 and a value of 8b'10111111 (in 256-bit Hash) selects Bit 31 of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

Calculate the 32-bit CRC for the DA (See IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).

Perform bitwise reversal for the value obtained in Step 1.

Take the upper 6 (or 7 or 8) bits from the value obtained in Step 2.

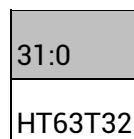
If the corresponding bit value of the register is 1'b1, the packet is accepted. Otherwise, it is rejected. If the PM bit is set in MAC_Packet_Filter, all multicast packets are accepted regardless of the multicast hash values.

If the Hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Hash Table Register X registers are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

Size: 32 bits

Offset: 0x14



Fields for Register: MAC_Hash_Table_Reg1

Bits	Name	Memory Access	Description
31:0	HT63T32	R/W	<p>MAC Hash Table Second 32 Bits</p> <p>This field contains the second 32 Bits [63:32] of the Hash table.</p> <p>Value After Reset: 0x0</p>

MAC_VLAN_Tag_Ctrl

Description: This register is the redefined format of the MAC VLAN Tag Register. It is used for indirect addressing. It contains the address offset, command type and Busy Bit for CSR access of the Per VLAN Tag registers.

Size: 32 bits

Offset: 0x50

31	30	29:2 6	25	24	23	22:2 1	20	19	18	17	16	15:y	3:2	1	0
Rsv d	Reserved _30	Rsv d	VTH M	EVLR XS	Reserved _23	EVL S	DOVL TC	ERSVL M	ESV L	VTI M	ET V	Reserved_1 5_y	OF S	C	O B

Fields for Register: MAC_VLAN_Tag_Ctrl

Bits	Name	Memory Access	Description
31			Reserved Field: Yes
30	Reserved_30	R	Reserved. Value After Reset: 0x0
29:26			Reserved Field: Yes
25	VTHM	R/W	<p>VLAN Tag Hash Table Match Enable</p> <p>When this bit is set, the most significant four bits of CRC of VLAN Tag are used to index the content of the MAC_VLAN_Hash_Table register. A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the packet matched the VLAN hash table.</p> <p>When the ETV bit is set, the CRC of the 12-bit VLAN Identifier (VID) is used for comparison. When the ETV bit is reset, the CRC of the 16-bit VLAN tag is used for comparison.</p> <p>When this bit is reset, the VLAN Hash Match operation is not performed.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): VLAN Tag Hash Table Match is disabled 0x1 (ENABLE): VLAN Tag Hash Table Match is enabled <p>Value After Reset: 0x0</p>
24	EVLRXS	R/W	<p>Enable VLAN Tag in Rx status</p> <p>When this bit is set, MAC provides the outer VLAN Tag in the Rx status. When this bit is reset, the MAC does not provide the outer VLAN Tag in Rx status.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): VLAN Tag in Rx status is disabled 0x1 (ENABLE): VLAN Tag in Rx status is enabled <p>Value After Reset: 0x0</p>
23	Reserved_23	R	Reserved. Value After Reset: 0x0
22:21	EVLS	R/W	<p>Enable VLAN Tag Stripping on Receive</p> <p>This field indicates the stripping operation on the outer VLAN Tag in received packet.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DONOT): Do not strip 0x1 (IFPASS): Strip if VLAN filter passes 0x2 (IFFAIL): Strip if VLAN filter fails

Bits	Name	Memory Access	Description
			<p>0x3 (ALWAYS): Always strip</p> <p>Value After Reset: 0x0</p>
20	DOVLTC	R/W	<p>Disable VLAN Type Check for VLAN Hash Filtering</p> <p>When this bit is set, the MAC VLAN Hash Filter does not check whether the VLAN Tag specified by the ERIVLT bit is of type S-VLAN or C-VLAN.</p> <p>When this bit is reset, the MAC VLAN Hash Filter filters or matches the VLAN Tag specified by the ERIVLT bit only when VLAN Tag type is similar to the one specified by the ERSVLM bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): VLAN Type Check is enabled 0x1 (DISABLE): VLAN Type Check is disabled <p>Value After Reset: 0x0</p>
19	ERSVLM	R/W	<p>Enable Receive S-VLAN Match for VLAN Hash Filtering</p> <p>When this bit is set, the MAC receiver enables VLAN Hash filtering or matching for S-VLAN (Type = 0x88A8) packets. When this bit is reset, the MAC receiver enables VLAN Hash filtering or matching for C-VLAN (Type = 0x8100) packets.</p> <p>The ERIVLT bit determines the VLAN tag position considered for VLAN Hash filtering or matching.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive S-VLAN Match is disabled 0x1 (ENABLE): Receive S-VLAN Match is enabled <p>Value After Reset: 0x0</p>
18	ESVL	R/W	<p>Enable S-VLAN</p> <p>When this bit is set, the MAC transmitter and receiver consider the S-VLAN packets (Type = 0x88A8) as valid VLAN tagged packets.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): S-VLAN is disabled 0x1 (ENABLE): S-VLAN is enabled <p>Value After Reset: 0x0</p>
17	VTIM	R/W	<p>VLAN Tag Inverse Match Enable</p> <p>When this bit is set, this bit enables the VLAN Tag inverse matching. The packets without matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The packets with matched VLAN Tag are marked as matched.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x0 (DISABLE): VLAN Tag Inverse Match is disabled 0x1 (ENABLE): VLAN Tag Inverse Match is enabled</p> <p>Value After Reset: 0x0</p>
16	ETV	R/W	<p>Enable 12-Bit VLAN Tag Comparison for VLAN Hash Filtering</p> <p>When this bit is set, a 12-bit VLAN identifier is used for VLAN Hash filtering instead of the complete 16-bit VLAN tag. Bits[11:0] of VLAN tag in the received VLAN-tagged packet are used for hash-based VLAN filtering.</p> <p>When this bit is reset, all 16 bits of the 15th and 16th bytes of the received VLAN packet are used for VLAN hash filtering.</p> <p>Values:</p> <p>0x0 (DISABLE): 12-Bit VLAN Tag Comparison is disabled 0x1 (ENABLE): 12-Bit VLAN Tag Comparison is enabled</p> <p>Value After Reset: 0x0</p>
15:4	Reserved_15_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
3:2	OFS	R/W	<p>Offset</p> <p>This field holds the address offset of the MAC VLAN Tag Filter Register which the application is trying to access.</p> <p>The width of the field depends on the number of MAC VLAN Tag Registers enabled.</p> <p>Value After Reset: 0x0</p>
1	CT	R/W	<p>Command Type</p> <p>This bit indicates if the current register access is a read or a write.</p> <p>When set, it indicate a read operation. When reset, it indicates a write operation.</p> <p>Values:</p> <p>0x0 (WRITE): Write operation 0x1 (READ): Read operation</p> <p>Value After Reset: 0x0</p>
0	OB	R/W	<p>Operation Busy</p> <p>This bit is set along with a read or write command for initiating the indirect access to per VLAN Tag Filter register. This bit is reset when the read or write command to per VLAN Tag Filter indirect access</p>

Bits	Name	Memory Access	Description
			<p>register is complete. The next indirect register access can be initiated only after this bit is reset.</p> <p>During a write operation, the bit is reset only after the data has been written into the Per VLAN Tag register.</p> <p>During a read operation, the data should be read from the MAC_VLAN_Tag_Data register only after this bit is reset.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Operation Busy is disabled 0x1 (ENABLE): Operation Busy is enabled <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MAC_VLAN_Tag_Data

Description: This register holds the read/write data for Indirect Access of the Per VLAN Tag registers. During the read access, this field contains valid read data only after the OB bit is reset. During the write access, this field should be valid prior to setting the OB bit in the MAC_VLAN_Tag_Ctrl Register.

Size: 32 bits

Offset: 0x54

31:y	26:25	24	23:21	20	19	18	17	16	15:0
Reserved_31_y	DMACHN	DMACHEN	Reserved_23_21	Rsvd	ERSVLM	DOVLTC	ETV	VEN	VID

Fields for Register: MAC_VLAN_Tag_Data

Bits	Name	Memory Access	Description
31:27	Reserved_31_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
26:25	DMACHN	R/W	<p>DMA Channel Number</p> <p>The DMA Channel number to which the VLAN Tagged Frame is to be routed if it passes this VLAN Tag Filter is programmed in this field.</p> <p>If the Routing based on VLAN Tag Filter is not necessary, this field need not be programmed.</p> <p>Value After Reset: 0x0</p>
24	DMACHEN	R/W	<p>DMA Channel Number Enable</p> <p>This bit is the Enable for the DMA Channel Number value programmed in the field DMACH.</p>

Bits	Name	Memory Access	Description
			<p>When this bit is reset, the Routing does not occur based on VLAN Filter result. The frame is routed based on DA Based DMA Channel Routing.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): DMA Channel Number is disabled 0x1 (ENABLE): DMA Channel Number is enabled <p>Value After Reset: 0x0</p>
23:21	Reserved_23_21	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
20			<p>Reserved Field: Yes</p>
19	ERSVLM	R/W	<p>Enable S-VLAN Match for received Frames</p> <p>This bit is valid only when VLAN Tag Enable of the Filter is set.</p> <p>When this bit is set, the MAC receiver enables filtering or matching for S-VLAN (Type = 0x88A8) packets.</p> <p>When this bit is reset, the MAC receiver enables filtering or matching for C-VLAN (Type = 0x8100) packets.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive S-VLAN Match is disabled 0x1 (ENABLE): Receive S-VLAN Match is enabled <p>Value After Reset: 0x0</p>
18	DOVLTC	R/W	<p>Disable VLAN Type Comparison</p> <p>This bit is valid only when VLAN Tag Enable of the Filter is set.</p> <p>When this bit is set, the MAC does not check whether the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit is of type S-VLAN or C-VLAN.</p> <p>When this bit is reset, the MAC filters or matches the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit only when VLAN Tag type is similar to the one specified by the Enable S-VLAN Match for received Frames bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): VLAN type comparison is enabled 0x1 (DISABLE): VLAN type comparison is disabled <p>Value After Reset: 0x0</p>
17	ETV	R/W	<p>12bits or 16bits VLAN comparison</p> <p>This bit is valid only when VEN of the Filter is set.</p>

Bits	Name	Memory Access	Description
			<p>When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged packet.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_16BIT): 16 bit VLAN comparison 0x1 (M_12BIT): 12 bit VLAN comparison <p>Value After Reset: 0x0</p>
16	VEN	R/W	<p>VLAN Tag Enable</p> <p>This bit is used to enable or disable the VLAN Tag.</p> <p>When this bit is set, the MAC compares the VLAN Tag of received packet with the VLAN Tag ID.</p> <p>When this bit is reset, no comparison is performed irrespective of the programming of the other fields.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): VLAN Tag is disabled 0x1 (ENABLE): VLAN Tag is enabled <p>Value After Reset: 0x0</p>
15:0	VID	R/W	<p>VLAN Tag ID</p> <p>This field holds the VLAN Tag value which is used by the MAC for perfect comparison. It is valid when VLAN Tag Enable is set.</p> <p>Value After Reset: 0x0</p>

MAC_VLAN_Tag_Filter(#i) (for i = 0; i <= 3)

Description: This register contains VLAN Tag filter \${i} control information.

Size: 32 bits

Offset: 0x54

31:y	26:25	24	23:21	20	19	18	17	16	15:0
Reserved_31_y	DMACHN	DMACHEN	Reserved_23_21	Rsvd	ERSVLM	DOVLTC	ETV	VEN	VID

Fields for Register: MAC_VLAN_Tag_Filter(#i) (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:27	Reserved_31_y	R	Reserved.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
26:25	DMACHN	R/W	<p>DMA Channel Number</p> <p>The DMA Channel number to which the VLAN Tagged Frame is to be routed if it passes this VLAN Tag Filter is programmed in this field.</p> <p>If the Routing based on VLAN Tag Filter is not necessary, this field need not be programmed.</p> <p>Value After Reset: 0x0</p>
24	DMACHEN	R/W	<p>DMA Channel Number Enable</p> <p>This bit is the Enable for the DMA Channel Number value programmed in the field DMACH.</p> <p>When this bit is reset, the Routing does not occur based on VLAN Filter result. The frame is routed based on DA Based DMA Channel Routing.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): DMA Channel Number is disabled 0x1 (ENABLE): DMA Channel Number is enabled <p>Value After Reset: 0x0</p>
23:21	Reserved_23_21	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
20			Reserved Field: Yes
19	ERSVLM	R/W	<p>Enable S-VLAN Match for received Frames</p> <p>This bit is valid only when VLAN Tag Enable of the Filter is set.</p> <p>When this bit is set, the MAC receiver enables filtering or matching for S-VLAN (Type = 0x88A8) packets.</p> <p>When this bit is reset, the MAC receiver enables filtering or matching for C-VLAN (Type = 0x8100) packets.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive S-VLAN Match is disabled 0x1 (ENABLE): Receive S-VLAN Match is enabled <p>Value After Reset: 0x0</p>
18	DOVLTC	R/W	<p>Disable VLAN Type Comparison</p> <p>This bit is valid only when VLAN Tag Enable of the Filter is set.</p> <p>When this bit is set, the MAC does not check whether the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit is of type S-VLAN or C-VLAN.</p>

Bits	Name	Memory Access	Description
			<p>When this bit is reset, the MAC filters or matches the VLAN Tag specified by the Enable Inner VLAN Tag Comparison bit only when VLAN Tag type is similar to the one specified by the Enable S-VLAN Match for received Frames bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): VLAN type comparison is enabled 0x1 (DISABLE): VLAN type comparison is disabled <p>Value After Reset: 0x0</p>
17	ETV	R/W	<p>12bits or 16bits VLAN comparison</p> <p>This bit is valid only when VEN of the Filter is set.</p> <p>When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged packet.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_16BIT): 16 bit VLAN comparison 0x1 (M_12BIT): 12 bit VLAN comparison <p>Value After Reset: 0x0</p>
16	VEN	R/W	<p>VLAN Tag Enable</p> <p>This bit is used to enable or disable the VLAN Tag.</p> <p>When this bit is set, the MAC compares the VLAN Tag of received packet with the VLAN Tag ID.</p> <p>When this bit is reset, no comparison is performed irrespective of the programming of the other fields.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): VLAN Tag is disabled 0x1 (ENABLE): VLAN Tag is enabled <p>Value After Reset: 0x0</p>
15:0	VID	R/W	<p>VLAN Tag ID</p> <p>This field holds the VLAN Tag value which is used by the MAC for perfect comparison. It is valid when VLAN Tag Enable is set.</p> <p>Value After Reset: 0x0</p>

MAC_VLAN_Hash_Table

Description: When VTHM bit of the MAC_VLAN_Tag register is set, the 16-bit VLAN Hash Table

register is used for group address filtering based on the VLAN tag. For hash filtering, the content of the 16-bit VLAN tag or 12-bit VLAN ID (based on the ETV bit of MAC_VLAN_Tag Register) in the incoming packet is passed through the CRC logic. The upper four bits of the calculated hash value are used to index the contents of the VLAN Hash table. For example, hash value of 4b'1000 selects Bit 8 of the VLAN Hash table.

The hash value of the destination address is calculated in the following way:

Calculate the 32-bit CRC for the VLAN tag or ID (For steps to calculate CRC32, see Section 3.2.8 of IEEE 802.3).

Perform bitwise reversal for the value obtained in step 1.

Take the upper four bits from the value obtained in step 2.

If the VLAN hash Table register is configured to be double-synchronized to the (G)MII clock domain, the synchronization is triggered only when Bits[15:8] (in little-endian mode) or Bits[7:0] (in big-endian mode) of this register are written.

If double-synchronization is enabled, consecutive writes to this register should be performed after at least four clock cycles in the destination clock domain.

Size: 32 bits

Offset: 0x58

31:16	15:0
Reserved_31_16	VLHT

Fields for Register: MAC_VLAN_Hash_Table

Bits	Name	Memory Access	Description
31:16	Reserved_31_16	R	Reserved. Value After Reset: 0x0
15:0	VLHT	R/W	VLAN Hash Table This field contains the 16-bit VLAN Hash Table. Value After Reset: 0x0

MAC_VLAN_Incl

Description: The VLAN Tag Inclusion or Replacement register contains the VLAN tag for insertion or replacement in the Transmit packets. It also contains the VLAN tag insertion controls.

Size: 32 bits

Offset: 0x60

31:30	29:26	25:24	23:22	21	20	19	18	17:16	15:0
Rsvd	Reserved_29_y	Rsvd	Reserved_23_22	Rsvd	VLTI	CSVL	VLP	VLC	VLT

Fields for Register: MAC_VLAN_Incl

Bits	Name	Memory Access	Description
31:30			Reserved Field: Yes
29:26	Reserved_29_y	R	Reserved. Value After Reset: 0x0
25:24			Reserved Field: Yes
23:22	Reserved_23_22	R	Reserved. Value After Reset: 0x0
21			Reserved Field: Yes
20	VLTI	R/W	VLAN Tag Input When this bit is set, it indicates that the VLAN tag to be inserted or replaced in Tx packet should be taken from: The Tx descriptor Values: 0x0 (DISABLE): VLAN Tag Input is disabled 0x1 (ENABLE): VLAN Tag Input is enabled Value After Reset: 0x0
19	CSVL	R/W	C-VLAN or S-VLAN When this bit is set, S-VLAN type (0x88A8) is inserted or replaced in the 13th and 14th bytes of transmitted packets. When this bit is reset, C-VLAN type (0x8100) is inserted or replaced in the 13th and 14th bytes of transmitted packets. Values: 0x0 (C-VLAN): C-VLAN type (0x8100) is inserted or replaced 0x1 (S-VLAN): S-VLAN type (0x88A8) is inserted or replaced Value After Reset: 0x0
18	VLP	R/W	VLAN Priority Control When this bit is set, the control bits[17:16] are used for VLAN deletion, insertion, or replacement. When this bit is reset, the mti_vlan_ctrl_i control input is used and bits[17:16] are ignored. Values: 0x0 (DISABLE): VLAN Priority Control is disabled 0x1 (ENABLE): VLAN Priority Control is enabled Value After Reset: 0x0
17:16	VLC	R/W	VLAN Tag Control in Transmit Packets

Bits	Name	Memory Access	Description
			<p>2'b00: No VLAN tag deletion, insertion, or replacement 2'b01: VLAN tag deletion The MAC removes the VLAN type (bytes 13 and 14) and VLAN tag (bytes 15 and 16) of all transmitted packets with VLAN tags. 2'b10: VLAN tag insertion The MAC inserts VLT in bytes 15 and 16 of the packet after inserting the Type value (0x8100 or 0x88a8) in bytes 13 and 14. This operation is performed on all transmitted packets, irrespective of whether they already have a VLAN tag. 2'b11: VLAN tag replacement The MAC replaces VLT in bytes 15 and 16 of all VLAN-type transmitted packets (Bytes 13 and 14 are 0x8100 or 0x88a8).</p> <p>Note: Changes to this field take effect only on the start of a packet. If you write this register field when a packet is being transmitted, only the subsequent packet can use the updated value, that is, the current packet does not use the updated value.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NONE): No VLAN tag deletion, insertion, or replacement 0x1 (DELETE): VLAN tag deletion 0x2 (INSERT): VLAN tag insertion 0x3 (REPLACE): VLAN tag replacement <p>Value After Reset: 0x0</p>
15:0	VLT	R/W	<p>VLAN Tag for Transmit Packets</p> <p>This field contains the value of the VLAN tag to be inserted or replaced. The value must only be changed when the transmit lines are inactive or during the initialization phase.</p> <p>Bits[15:13] are the User Priority field, Bit 12 is the CFI/DEI field, and Bits[11:0] are the VID field in the VLAN tag.</p> <p>The following list describes the bits of this field:</p> <ul style="list-style-type: none"> Bits[15:13]: User Priority Bit 12: Canonical Format Indicator (CFI) or Drop Eligible Indicator (DEI) Bits[11:0]: VLAN Identifier (VID) field of VLAN tag <p>Value After Reset: 0x0</p>

MAC_Q0_Tx_Flow_Ctrl

Description: The Flow Control register controls the generation and reception of the Control (Pause Command) packets by the Flow control module of the MAC. A Write to a register with the Busy bit set to 1 triggers the Flow Control block to generate a Pause packet. The fields of the control packet are selected as specified in the 802.3x specification, and the Pause Time value from this register is used in the Pause Time field of the control packet. The Busy bit remains set until the control packet is transferred onto the cable. The application must make sure that the Busy bit is cleared before writing to the register.

When the PFCE bit in the MAC_Rx_Flow_Ctrl register is enabled, this register controls the generation of Priority Flow Control (PFC) frames with priorities mapped according to PSRQ0 in the MAC_RxQ_Ctrl2 register.

Size: 32 bits

Offset: 0x70

31:16	15:8	7	6:4	3:2	1	0
PT	Reserved_15_8	DZPQ	PLT	Reserved_3_2	TFE	FCB_BPA

Fields for Register: MAC_Q0_Tx_Flow_Ctrl

Bits	Name	Memory Access	Description
31:16	PT	R/W	<p>Pause Time</p> <p>This field holds the value to be used in the Pause Time field in the Tx control packet. If the Pause Time bits are configured to be double-synchronized to the (G)MII clock domain, consecutive writes to this register should be performed only after at least four clock cycles in the destination clock domain.</p> <p>Value After Reset: 0x0</p>
15:8	Reserved_15_8	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
7	DZPQ	R/W	<p>Disable Zero-Quanta Pause</p> <p>When this bit is set, it disables the automatic generation of the zero-quanta Pause packets on de-assertion of the flow-control signal from the FIFO layer (MTL or external sideband flow control signal sbd_flowctrl_i or mti_flowctrl_i).</p> <p>When this bit is reset, normal operation with automatic zero-quanta Pause packet generation is enabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): Zero-Quanta Pause packet generation is enabled 0x1 (DISABLE): Zero-Quanta Pause packet generation is disabled <p>Value After Reset: 0x0</p>
6:4	PLT	R/W	<p>Pause Low Threshold</p> <p>This field configures the threshold of the Pause timer at which the</p>

Bits	Name	Memory Access	Description
			<p>input flow control signal mti_flowctrl_i (or sbd_flowctrl_i) is checked for automatic retransmission of the Pause packet.</p> <p>The threshold values should be always less than the Pause Time configured in Bits[31:16]. For example, if PT = 100H (256 slot times), and PLT = 001, a second Pause packet is automatically transmitted if the mti_flowctrl_i signal is asserted at 228 (256-28) slot times after the first Pause packet is transmitted.</p> <p>The following list provides the threshold values for different values. The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface.</p> <p>This (approximate) computation is based on the packet size (64, 1518, 2000, 9018, 16384, or 32768) + 2 Pause Packet Size + IPG in Slot Times.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (PT4): Pause Time minus 4 Slot Times (PT -4 slot times) 0x1 (PT28): Pause Time minus 28 Slot Times (PT -28 slot times) 0x2 (PT36): Pause Time minus 36 Slot Times (PT -36 slot times) 0x3 (PT144): Pause Time minus 144 Slot Times (PT -144 slot times) 0x4 (PT256): Pause Time minus 256 Slot Times (PT -256 slot times) 0x5 (PT512): Pause Time minus 512 Slot Times (PT -512 slot times) 0x6 (RSVD): Reserved <p>Value After Reset: 0x0</p>
3:2	Reserved_3_2	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
1	TFE	R/W	<p>Transmit Flow Control Enable</p> <p>Full-Duplex Mode:</p> <p>In the full-duplex mode, when this bit is set, the MAC enables the flow control operation to Tx Pause packets. When this bit is reset, the flow control operation in the MAC is disabled, and the MAC does not transmit any Pause packets.</p> <p>Half-Duplex Mode:</p> <p>In the half-duplex mode, when this bit is set, the MAC enables the backpressure operation. When this bit is reset, the backpressure feature is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Transmit Flow Control is disabled

Bits	Name	Memory Access	Description
			<p>0x1 (ENABLE): Transmit Flow Control is enabled</p> <p>Value After Reset: 0x0</p>
0	FCB_BPA	R/W	<p>Flow Control Busy or Backpressure Activate</p> <p>This bit initiates a Pause packet in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set.</p> <p>Full-Duplex Mode:</p> <p>In the full-duplex mode, this bit should be read as 1'b0 before writing to this register. To initiate a Pause packet, the application must set this bit to 1'b1. During Control packet transfer, this bit continues to be set to indicate that a packet transmission is in progress. When Pause packet transmission is complete, the MAC resets this bit to 1'b0. You should not write to this register until this bit is cleared.</p> <p>Half-Duplex Mode:</p> <p>When this bit is set (and TFE bit is set) in the half-duplex mode, the MAC asserts the backpressure. During backpressure, when the MAC receives a new packet, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the mti_flowctrl_i input signal for the backpressure function. When the MAC is configured for the full-duplex mode, the BPA is automatically disabled.</p> <p>Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Flow Control Busy or Backpressure Activate is disabled 0x1 (ENABLE): Flow Control Busy or Backpressure Activate is enabled <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MAC_Rx_Flow_Ctrl

Description: The Receive Flow Control register controls the pausing of MAC Transmit based on the received Pause packet.

Size: 32 bits

Offset: 0x90

31:9	8	7:2	1	0
Reserved_31_9	Rsvd	Reserved_7_2	UP	RFE

Fields for Register: MAC_Rx_Flow_Ctrl

Bits	Name	Memory Access	Description
31:9	Reserved_31_9	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
8			Reserved Field: Yes
7:2	Reserved_7_2	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
1	UP	R/W	<p>Unicast Pause Packet Detect</p> <p>A pause packet is processed when it has the unique multicast address specified in the IEEE 802.3. When this bit is set, the MAC can also detect Pause packets with unicast address of the station. This unicast address should be as specified in MAC_Address0_High and MAC_Address0_Low.</p> <p>When this bit is reset, the MAC only detects Pause packets with unique multicast address.</p> <p>Note: The MAC does not process a Pause packet if the multicast address is different from the unique multicast address. This is also applicable to the received PFC packet when the Priority Flow Control (PFC) is enabled. The unique multicast address (0x01_80_C2_00_00_01) is as specified in IEEE 802.1 Qbb-2011.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Unicast Pause Packet Detect disabled 0x1 (ENABLE): Unicast Pause Packet Detect enabled <p>Value After Reset: 0x0</p>
0	RFE	R/W	<p>Receive Flow Control Enable</p> <p>When this bit is set and the MAC is operating in full-duplex mode, the MAC decodes the received Pause packet and disables its transmitter for a specified (Pause) time. When this bit is reset or the MAC is operating in half-duplex mode, the decode function of the Pause packet is disabled.</p> <p>When PFC is enabled, flow control is enabled for PFC packets. The MAC decodes the received PFC packet and disables the Transmit queue, with matching priorities, for a duration of received Pause time.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Flow Control is disabled 0x1 (ENABLE): Receive Flow Control is enabled <p>Value After Reset: 0x0</p>

MAC_RxQ_Ctrl4

Description: The Receive Queue Control 4 register controls the routing of unicast and multicast packets that fail the Destination or Source address filter to the Rx queues.

Size: 32 bits

Offset: 0x94

31:y	18:17	16	15:11	10:9	8	7:3	2:1	0
Reserved_31_y	VFFQ	VFFQE	Reserved_15_y	MFFQ	MFFQE	Reserved_7_y	UFFQ	UFFQE

Fields for Register: MAC_RxQ_Ctrl4

Bits	Name	Memory Access	Description
31:19	Reserved_31_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
18:17	VFFQ	R/W	<p>VLAN Tag Filter Fail Packets Queue</p> <p>This field holds the Rx queue number to which the tagged packets failing the Destination or Source Address filter (and UFFQE/MFFQE not enabled) or failing the VLAN tag filter must be routed to. This field is valid only when the VFFQE bit is set.</p> <p>Value After Reset: 0x0</p>
16	VFFQE	R/W	<p>VLAN Tag Filter Fail Packets Queuing Enable</p> <p>When this bit is set, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter, are routed to the Rx Queue Number programmed in the VFFQ. When this bit is reset, the tagged packets which fail the Destination or Source address filter or fail the VLAN tag filter are routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): VLAN tag Filter Fail Packets Queuing is disabled 0x1 (ENABLE): VLAN tag Filter Fail Packets Queuing is enabled <p>Value After Reset: 0x0</p>
15:11	Reserved_15_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
10:9	MFFQ	R/W	<p>Multicast Address Filter Fail Packets Queue.</p> <p>This field holds the Rx queue number to which the Multicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the MFFQE bit is set.</p> <p>Value After Reset: 0x0</p>
8	MFFQE	R/W	Multicast Address Filter Fail Packets Queuing Enable.

Bits	Name	Memory Access	Description
			<p>When this bit is set, the Multicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the MFFQ. When this bit is reset, the Multicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Multicast Address Filter Fail Packets Queuing is disabled 0x1 (ENABLE): Multicast Address Filter Fail Packets Queuing is enabled <p>Value After Reset: 0x0</p>
7:3	Reserved_7_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
2:1	UFFQ	R/W	<p>Unicast Address Filter Fail Packets Queue.</p> <p>This field holds the Rx queue number to which the Unicast packets failing the Destination or Source Address filter are routed to. This field is valid only when the UFFQE bit is set.</p> <p>Value After Reset: 0x0</p>
0	UFFQE	R/W	<p>Unicast Address Filter Fail Packets Queuing Enable.</p> <p>When this bit is set, the Unicast packets which fail the Destination or Source address filter is routed to the Rx Queue Number programmed in the UFFQ. When this bit is reset, the Unicast packets which fail the Destination or Source address filter is routed based on other routing options. This bit is valid only when the RA bit of the MAC_Packet_Filter register is set.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Unicast Address Filter Fail Packets Queuing is disabled 0x1 (ENABLE): Unicast Address Filter Fail Packets Queuing is enabled <p>Value After Reset: 0x0</p>

MAC_RxQ_Ctrl0

Description: The Receive Queue Control 0 register controls the queue management in the MAC Receiver. **Note:** In multiple Rx queues configuration, all the queues are disabled by default. Enable the Rx queue by programming the corresponding field in this register.

Size: 32 bits

Offset: 0xa0

31:16	15:8	7:6	5:4	3:2	1:0
Reserved_31_16	Rsvd	RXQ3EN	RXQ2EN	RXQ1EN	RXQ0EN

Fields for Register: MAC_RxQ_Ctrl0

Bits	Name	Memory Access	Description
31:16	Reserved_31_16	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
15:8			Reserved Field: Yes
7:6	RXQ3EN	R/W	<p>Receive Queue 3 Enable</p> <p>This field is similar to the RXQ0EN field.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Queue not enabled 0x1 (EN_AV): Queue enabled for AV 0x2 (EN_DCB_GEN): Queue enabled for DCB/Generic 0x3 (RSVD): Reserved <p>Value After Reset: 0x0</p>
5:4	RXQ2EN	R/W	<p>Receive Queue 2 Enable</p> <p>This field is similar to the RXQ0EN field.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Queue not enabled 0x1 (EN_AV): Queue enabled for AV 0x2 (EN_DCB_GEN): Queue enabled for DCB/Generic 0x3 (RSVD): Reserved <p>Value After Reset: 0x0</p>
3:2	RXQ1EN	R/W	<p>Receive Queue 1 Enable</p> <p>This field is similar to the RXQ0EN field.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Queue not enabled 0x1 (EN_AV): Queue enabled for AV 0x2 (EN_DCB_GEN): Queue enabled for DCB/Generic 0x3 (RSVD): Reserved <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
1:0	RXQ0EN	R/W	<p>Receive Queue 0 Enable</p> <p>This field indicates whether Rx Queue 0 is enabled for AV or DCB.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Queue not enabled 0x1 (EN_AV): Queue enabled for AV 0x2 (EN_DCB_GEN): Queue enabled for DCB/Generic 0x3 (RSVD): Reserved <p>Value After Reset: 0x0</p>

MAC_RxQ_Ctrl1

Description: The Receive Queue Control 1 register controls the routing of multicast, broadcast, AV, DCB, and untagged packets to the Rx queues.

Size: 32 bits

Offset: 0xa4

31:30	29	28	27	26: 21	20	19	18:1 6	15	14: 12	11	10: 8	7	6: 4	3	2: 0
Reserved_31_30	TB_RQE	OMC_Bcq	Reserved_27	Rs_vd	MCB_CQEN	Reserved_19	MC_Bcq	Reserved_15	UP_Q	Reserved_11	Rs_vd	Reserved_7	Rs_vd	Reserved_3	Rs_vd

Fields for Register: MAC_RxQ_Ctrl1

Bits	Name	Memory Access	Description
31:30	Reserved_31_30	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
29	TBRQE	R/W	<p>Type Field Based Rx Queuing Enable</p> <p>When this bit is set, it enables Type Field based Rx Queuing where the Type field of received packet is compared with programmed TYP field in MAC_TMRQ_Regs(#i) and if a match occurs the packet is routed to the corresponding TMRQ field.</p> <p>Value After Reset: 0x0</p>
28	OMCBCQ	R/W	<p>Over-riding MC-BC queue priority select</p> <p>1: Priority of MCBCQ is reduced and the received packet is first routed to PTPQ, AVCPQ, DCBCPQ depending on packet type.</p>

Bits	Name	Memory Access	Description
			<p>0: Received Multicast/Broadcast packet is routed to MCBCQ.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): overriding MCBCQ priority disabled 0x1 (ENABLE): overriding MCBCQ priority enabled <p>Value After Reset: 0x0</p>
27	Reserved_27	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
26:21			<p>Reserved Field: Yes</p>
20	MCBCQEN	R/W	<p>Multicast and Broadcast Queue Enable</p> <p>This bit specifies that Multicast or Broadcast packets routing to the Rx Queue is enabled and the Multicast or Broadcast packets must be routed to Rx Queue specified in MCBCQ field.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Multicast and Broadcast Queue is disabled 0x1 (ENABLE): Multicast and Broadcast Queue is enabled <p>Value After Reset: 0x0</p>
19	Reserved_19	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
18:16	MCBCQ	R/W	<p>Multicast and Broadcast Queue</p> <p>This field specifies the Rx Queue onto which Multicast or Broadcast Packets are routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Multicast or Broadcast Packets.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (QUEUE0): Receive Queue 0 0x1 (QUEUE1): Receive Queue 1 0x2 (QUEUE2): Receive Queue 2 0x3 (QUEUE3): Receive Queue 3 0x4 (QUEUE4): Receive Queue 4 0x5 (QUEUE5): Receive Queue 5 0x6 (QUEUE6): Receive Queue 6 0x7 (QUEUE7): Receive Queue 7 <p>Value After Reset: 0x0</p>
15	Reserved_15	R	Reserved.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
14:12	UPQ	R/W	<p>Untagged Packet Queue</p> <p>This field indicates the Rx Queue to which Untagged Packets are to be routed. Any Rx Queue enabled for Generic/DCB/AV traffic can be used to route the Untagged Packets.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (QUEUE0): Receive Queue 0 0x1 (QUEUE1): Receive Queue 1 0x2 (QUEUE2): Receive Queue 2 0x3 (QUEUE3): Receive Queue 3 0x4 (QUEUE4): Receive Queue 4 0x5 (QUEUE5): Receive Queue 5 0x6 (QUEUE6): Receive Queue 6 0x7 (QUEUE7): Receive Queue 7 <p>Value After Reset: 0x0</p>
11	Reserved_11	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
10:8			Reserved Field: Yes
7	Reserved_7	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
6:4			Reserved Field: Yes
3	Reserved_3	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
2:0			Reserved Field: Yes

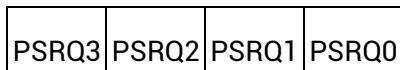
MAC_RxQ_Ctrl2

Description: This register controls the routing of tagged packets based on the USP (user Priority) field of the received packets to the RxQueues 0 to 3.

Size: 32 bits

Offset: 0xa8



**Fields for Register: MAC_RxQ_Ctrl2**

Bits	Name	Memory Access	Description
31:24	PSRQ3	R/W	<p>Priorities Selected in the Receive Queue 3</p> <p>This field decides the priorities assigned to Rx Queue 3. All packets with priorities that match the values set in this field are routed to Rx Queue 3. For example, if PSRQ3[6, 3] are set, packets with USP field equal to 3 or 6 are routed to Rx Queue 3. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.</p> <p>Value After Reset: 0x0</p>
23:16	PSRQ2	R/W	<p>Priorities Selected in the Receive Queue 2</p> <p>This field decides the priorities assigned to Rx Queue 2. All packets with priorities that match the values set in this field are routed to Rx Queue 2. For example, if PSRQ2[1, 0] are set, packets with USP field equal to 1 or 0 are routed to Rx Queue 2. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.</p> <p>Value After Reset: 0x0</p>
15:8	PSRQ1	R/W	<p>Priorities Selected in the Receive Queue 1</p> <p>This field decides the priorities assigned to Rx Queue 1. All packets with priorities that match the values set in this field are routed to Rx Queue 1. For example, if PSRQ1[4] is set, packets with USP field equal to 4 are routed to Rx Queue 1. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.</p> <p>Value After Reset: 0x0</p>
7:0	PSRQ0	R/W	<p>Priorities Selected in the Receive Queue 0</p> <p>This field decides the priorities assigned to Rx Queue 0. All packets with priorities that match the values set in this field are routed to Rx Queue 0. For example, if PSRQ0[5] is set, packets with USP field equal to 5 are routed to Rx Queue 0. The software must ensure that the content of this field is mutually exclusive to the PSRQ fields for other queues, that is, the same priority is not mapped to multiple Rx queues.</p> <p>Value After Reset: 0x0</p>

MAC_Interrupt_Status

Description: The Interrupt Status register contains the status of interrupts.

Size: 32 bits

Offset: 0xb0

31:21	20:1 9	18	17	16	15	14	13	12: 8	7:6	5:4	3	2:1	0
Reserved_31_21	Rsvd	MDIOIS	Rsvd	Reserved_16	Rsvd	RXSTSIS	TXSTSIS	Rsvd	Reserved_7_6	Rsvd	PHYIS	Rsvd	RGSMIIS

Fields for Register: MAC_Interrupt_Status

Bits	Name	Memory Access	Description
31:21	Reserved_31_21	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
20:19			Reserved Field: Yes
18	MDIOIS	R	<p>MDIO Interrupt Status</p> <p>This bit indicates an interrupt event after the completion of MDIO operation. To reset this bit, the application has to read this bit/Write 1 to this bit when RCWE bit in MAC_CSR_SW_Ctrl register is set.</p> <p>Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MDIO Interrupt status not active 0x1 (ACTIVE): MDIO Interrupt status active <p>Value After Reset: 0x0</p>
17			Reserved Field: Yes
16	Reserved_16	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
15			Reserved Field: Yes
14	RXSTSIS	R	<p>Receive Status Interrupt</p> <p>This bit indicates the status of received packets. This bit is set when the RWT bit is set in the MAC_Rx_Tx_Status register. This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set) in the MAC_Rx_Tx_Status register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Receive Interrupt status not active 0x1 (ACTIVE): Receive Interrupt status active <p>Value After Reset: 0x0</p>
13	TXSTSIS	R	Transmit Status Interrupt

Bits	Name	Memory Access	Description
			<p>This bit indicates the status of transmitted packets. This bit is set when any of the following bits is set in the MAC_Rx_Tx_Status register:</p> <ul style="list-style-type: none"> Excessive Collision (EXCOL) Late Collision (LCOL) Excessive Deferral (EXDEF) Loss of Carrier (LCARR) No Carrier (NCARR) Jabber Timeout (TJT) <p>This bit is cleared when the corresponding interrupt source bit is read (or corresponding interrupt source bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set) in the MAC_Rx_Tx_Status register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Transmit Interrupt status not active 0x1 (ACTIVE): Transmit Interrupt status active <p>Value After Reset: 0x0</p>
12:8			Reserved Field: Yes
7:6	Reserved_7_6	R	Reserved. Value After Reset: 0x0
5:4			Reserved Field: Yes
3	PHYIS	R	<p>PHY Interrupt</p> <p>This bit is set when rising edge is detected on the phy_intr_i input. This bit is cleared when this register is read (or this bit is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): PHY Interrupt not detected 0x1 (ACTIVE): PHY Interrupt detected <p>Value After Reset: 0x0</p>
2:1			Reserved Field: Yes
0	RGSMIIIS	R	<p>RGMII or SMII Interrupt Status</p> <p>This bit is set because of any change in value of the Link Status of RGMII or SMII interface (LNKSTS bit in MAC_PHYIF_Control_Status register). This bit is cleared when the MAC_PHYIF_Control_Status register is read (or LNKSTS bit of MAC_PHYIF_Control_Status</p>

Bits	Name	Memory Access	Description
			<p>register is written to 1 when RCWE bit of MAC_CSR_SW_Ctrl register is set).</p> <p>This bit is valid only when you select the optional RGMII or SMII PHY interface.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): RGMII or SMII Interrupt Status is not active 0x1 (ACTIVE): RGMII or SMII Interrupt Status is active <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MAC_Interrupt_Enable

Description: The Interrupt Enable register contains the masks for generating the interrupts.

Size: 32 bits

Offset: 0xb4

31:19	18	17	16	15	14	13	12	11:6	5:4	3	2:1	0
Reserved_31_19	MDIOIE	Rsvd	Reserved_16	Reserved_15	RXSTSIE	TXSTSIE	Rsvd	Reserved_11_6	Rsvd	PHYIE	Rsvd	RGSMIIE

Fields for Register: MAC_Interrupt_Enable

Bits	Name	Memory Access	Description
31:19	Reserved_31_19	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
18	MDIOIE	R/W	<p>MDIO Interrupt Enable</p> <p>When this bit is set, it enables the assertion of the interrupt when MDIOS field is set in the MAC_Interrupt_Status register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): MDIO Interrupt is disabled 0x1 (ENABLE): MDIO Interrupt is enabled <p>Value After Reset: 0x0</p>
17			Reserved Field: Yes
16	Reserved_16	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
15	Reserved_15	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
14	RXSTSIE	R/W	<p>Receive Status Interrupt Enable</p> <p>When this bit is set, it enables the assertion of the interrupt signal because of the setting of RXSTSIS bit in the MAC_Interrupt_Status register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Status Interrupt is disabled 0x1 (ENABLE): Receive Status Interrupt is enabled <p>Value After Reset: 0x0</p>
13	TXSTSIE	R/W	<p>Transmit Status Interrupt Enable</p> <p>When this bit is set, it enables the assertion of the interrupt signal because of the setting of TXSTSIS bit in the MAC_Interrupt_Status register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Timestamp Status Interrupt is disabled 0x1 (ENABLE): Timestamp Status Interrupt is enabled <p>Value After Reset: 0x0</p>
12			Reserved Field: Yes
11:6	Reserved_11_6	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
5:4			Reserved Field: Yes
3	PHYIE	R/W	<p>PHY Interrupt Enable</p> <p>When this bit is set, it enables the assertion of the interrupt signal because of the setting of PHYIS bit in MAC_Interrupt_Status register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): PHY Interrupt is disabled 0x1 (ENABLE): PHY Interrupt is enabled <p>Value After Reset: 0x0</p>
2:1			Reserved Field: Yes
0	RGSMMIIIE	R/W	<p>RGMII or SMII Interrupt Enable</p> <p>When this bit is set, it enables the assertion of the interrupt signal</p>

Bits	Name	Memory Access	Description
			<p>because of the setting of RGSMIIIS bit in MAC_Interrupt_Status register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): RGMII or SMII Interrupt is disabled 0x1 (ENABLE): RGMII or SMII Interrupt is enabled <p>Value After Reset: 0x0</p>

MAC_Rx_Tx_Status

Description: The Receive Transmit Status register contains the Receive and Transmit Error status.

Size: 32 bits

Offset: 0xb8

31:9	8	7:6	5	4	3	2	1	0
Reserved_31_9	RWT	Reserved_7_6	EXCOL	LCOL	EXDEF	LCARR	NCARR	TJT

Fields for Register: MAC_Rx_Tx_Status

Bits	Name	Memory Access	Description
31:9	Reserved_31_9	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
8	RWT	R	<p>Receive Watchdog Timeout</p> <p>This bit is set when a packet with length greater than 2,048 bytes is received (10,240 bytes when Jumbo Packet mode is enabled) and the WD bit is reset in the MAC_Configuration register. This bit is set when a packet with length greater than 16,383 bytes is received and the WD bit is set in the MAC_Configuration register.</p> <p>Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): No receive watchdog timeout 0x1 (ACTIVE): Receive watchdog timed out <p>Value After Reset: 0x0</p>
7:6	Reserved_7_6	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
5	EXCOL	R	Excessive Collisions

Bits	Name	Memory Access	Description
			<p>When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the transmission aborted after 16 successive collisions while attempting to transmit the current packet. If the DR bit is set in the MAC_Configuration register, this bit is set after the first collision and the packet transmission is aborted.</p> <p>Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): No collision 0x1 (ACTIVE): Excessive collision is sensed <p>Value After Reset: 0x0</p>
4	LCOL	R	<p>Late Collision</p> <p>When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the packet transmission aborted because a collision occurred after the collision window (64 bytes including Preamble in MII mode; 512 bytes including Preamble and Carrier Extension in GMII mode).</p> <p>This bit is not valid if the Underflow error occurs.</p> <p>Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): No collision 0x1 (ACTIVE): Late collision is sensed <p>Value After Reset: 0x0</p>
3	EXDEF	R	<p>Excessive Deferral</p> <p>When the DTXSTS bit is set in the MTL_Operation_Mode register and the DC bit is set in the MAC_Configuration register, this bit indicates that the transmission ended because of excessive deferral of over 24,288 bit times (155,680 in 1000/2500 Mbps mode or when Jumbo packet is enabled).</p> <p>Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): No Excessive deferral 0x1 (ACTIVE): Excessive deferral <p>Value After Reset: 0x0</p>
2	LCARR	R	<p>Loss of Carrier</p> <p>When the DTXSTS bit is set in the MTL_Operation_Mode register, this</p>

Bits	Name	Memory Access	Description
			<p>bit indicates that the loss of carrier occurred during packet transmission, that is, the phy_crs_i signal was inactive for one or more transmission clock periods during packet transmission. This bit is valid only for packets transmitted without collision.</p> <p>Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Carrier is present 0x1 (ACTIVE): Loss of carrier <p>Value After Reset: 0x0</p>
1	NCARR	R	<p>No Carrier</p> <p>When the DTXSTS bit is set in the MTL_Operation_Mode register, this bit indicates that the carrier signal from the PHY is not present at the end of preamble transmission.</p> <p>Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Carrier is present 0x1 (ACTIVE): No carrier <p>Value After Reset: 0x0</p>
0	TJT	R	<p>Transmit Jabber Timeout</p> <p>This bit indicates that the Transmit Jabber Timer expired which happens when the packet size exceeds 2,048 bytes (10,240 bytes when the Jumbo packet is enabled) and JD bit is reset in the MAC_Configuration register. This bit is set when the packet size exceeds 16,383 bytes and the JD bit is set in the MAC_Configuration register.</p> <p>Access restriction applies. Clears on read (or write of 1 when RCWE bit in MAC_CSR_SW_Ctrl register is set). Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): No Transmit Jabber Timeout 0x1 (ACTIVE): Transmit Jabber Timeout occurred <p>Value After Reset: 0x0</p>

MAC_PHYIF_Control_Status

Description: The PHY Interface Control and Status register indicates the status signals received by the SGMII, RGMII, or SMII interface (selected at reset) from the PHY. This register is optional.

Size: 32 bits**Offset:** 0xf8

31:22	21:20	19	18:17	16	15:5	4	3	2	1	0
Reserved_31_22	Rsvd	LNKSTS	LNKSPEED	LNKMOD	Reserved_15_5	Rsvd	Reserved_3	Rsvd	LU	T C

Fields for Register: MAC_PHYIF_Control_Status

Bits	Name	Memory Access	Description
31:22	Reserved_31_22	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
21:20			<p>Reserved Field: Yes</p>
19	LNKSTS	R	<p>Link Status</p> <p>This bit indicates whether the link is up (1'b1) or down (1'b0).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Link down 0x1 (ACTIVE): Link up <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
18:17	LNKSPEED	R	<p>Link Speed</p> <p>This bit indicates the current speed of the link.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_2500K): 2.5 MHz 0x1 (M_25M): 25 MHz 0x2 (M_125M): 125 MHz 0x3 (RSVD): Reserved <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
16	LNKMOD	R	<p>Link Mode</p> <p>This bit indicates the current mode of operation of the link.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (HDUPLEX): Half-duplex mode 0x1 (FDUPLEX): Full-duplex mode <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
			Testable: untestable
15:5	Reserved_15_5	R	Reserved. Value After Reset: 0x0
4			Reserved Field: Yes
3	Reserved_3	R	Reserved. Value After Reset: 0x0
2			Reserved Field: Yes
1	LUD	R/W	Link Up or Down This bit indicates whether the link is up or down during transmission of configuration in the RGMII, SGMII, or SMII interface. Values: 0x0 (LINKDOWN): Link down 0x1 (LINKUP): Link up Value After Reset: 0x0
0	TC	R/W	Transmit Configuration in RGMII, SGMII, or SMII When set, this bit enables the transmission of duplex mode, link speed, and link up or down information to the PHY in the RGMII, SMII, or SGMII port. When this bit is reset, no such information is driven to the PHY. The details of this feature are provided in the following sections: "Reduced Gigabit Media Independent Interface" "Serial Media Independent Interface" "Serial Gigabit Media Independent Interface" Values: 0x0 (DISABLE): Disable Transmit Configuration in RGMII, SGMII, or SMII 0x1 (ENABLE): Enable Transmit Configuration in RGMII, SGMII, or SMII Value After Reset: 0x0

MAC_Version

Description: The version register identifies the version of the DWC_ether_qos. This register contains two bytes: one that ESWIN uses to identify the IP release number, and the other that you set while configuring the IP.

Size: 32 bits**Offset:** 0x110

31:16	15:8	7:0
Reserved_31_16	USERVER	SNPSVER

Fields for Register: MAC_Version

Bits	Name	Memory Access	Description
31:16	Reserved_31_16	R	Reserved. Value After Reset: 0x0
15:8	USERVER	R	User-defined Version (configured with coreConsultant) Value After Reset: 0x10
7:0	SNPSVER	R	ESWIN-defined Version Value After Reset: 0x52

MAC_Debug**Description:** The Debug register provides the debug status of various MAC blocks.**Size:** 32 bits**Offset:** 0x114

31:19	18:17	16	15:3	2:1	0
Reserved_31_19	TFCSTS	TPESTS	Reserved_15_3	RFCFCSTS	RPESTS

Fields for Register: MAC_Debug

Bits	Name	Memory Access	Description
31:19	Reserved_31_19	R	Reserved. Value After Reset: 0x0
18:17	TFCSTS	R	MAC Transmit Packet Controller Status This field indicates the state of the MAC Transmit Packet Controller module. Values: 0x0 (IDLE): Idle state 0x1 (WAITING): Waiting for one of the following: Status of the previous packet OR IPG or back off period to be over 0x2 (GEN_TX_PAU): Generating and transmitting a Pause control packet (in full-duplex mode)

Bits	Name	Memory Access	Description
			<p>0x3 (TRNSFR): Transferring input packet for transmission</p> <p>Value After Reset: 0x0</p>
16	TPESTS	R	<p>MAC GMII or MII Transmit Protocol Engine Status</p> <p>When this bit is set, it indicates that the MAC GMII or MII transmit protocol engine is actively transmitting data, and it is not in the Idle state.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MAC GMII or MII Transmit Protocol Engine Status not detected 0x1 (ACTIVE): MAC GMII or MII Transmit Protocol Engine Status detected <p>Value After Reset: 0x0</p>
15:3	Reserved_15_3	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
2:1	RFCFCSTS	R	<p>MAC Receive Packet Controller FIFO Status</p> <p>When this bit is set, this field indicates the active state of the small FIFO Read and Write controllers of the MAC Receive Packet Controller module.</p> <p>Value After Reset: 0x0</p>
0	RPESTS	R	<p>MAC GMII or MII Receive Protocol Engine Status</p> <p>When this bit is set, it indicates that the MAC GMII or MII receive protocol engine is actively receiving data, and it is not in the Idle state.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MAC GMII or MII Receive Protocol Engine Status not detected 0x1 (ACTIVE): MAC GMII or MII Receive Protocol Engine Status detected <p>Value After Reset: 0x0</p>

MAC_HW_Feature0

Description: This register indicates the presence of first set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Note: All bits are set or reset according to the features selected while configuring the IP in coreConsultant.

Size: 32 bits**Offset:** 0x11c

31	30:28	27	26:25	24	23	22:18	17	16	15	14	13	12	11:10	9	8	7	6	5	4	3	2	1	0
Reserved_31	ACTPHYSEL	SAVLANINS	TSSTSSEL	MACADR64SEL	MACADR32SEL	ADDMACADRS	Reserved_17	RXC0ESFI	Reserved_15	TXCOESFI	FEESFI	TSSFI	Reserved_11_1	ARPOFFESFI	MIVCSFI	MGKSF1	RWKSF1	SMAIFI	VI_HASH	PCSSF1	HDSEFI	GMISFI	MISFI

Fields for Register: MAC_HW_Feature0

Bits	Name	Memory Access	Description
31	Reserved_31	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
30:28	ACTPHYSEL	R	<p>Active PHY Selected</p> <p>When you have multiple PHY interfaces in your configuration, this field indicates the sampled value of phy_intf_sel_i during reset de-assertion.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (GMII_MII): GMII or MII 0x1 (RGMII): RGMII 0x2 (SGMII): SGMII 0x3 (TBI): TBI 0x4 (RMII): RMII 0x5 (RTBI): RTBI 0x6 (SMII): SMII 0x7 (REVMIII): RevMII <p>Value After Reset: 0x0</p>
27	SAVLANINS	R	<p>Source Address or VLAN Insertion Enable</p> <p>This bit is set to 1 when the Enable SA and VLAN Insertion on Tx option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Source Address or VLAN Insertion Enable option is not selected 0x1 (ACTIVE): Source Address or VLAN Insertion Enable option is selected <p>Value After Reset: 0x1</p>

Bits	Name	Memory Access	Description
26:25	TSSTSSEL	R	<p>Timestamp System Time Source</p> <p>This bit indicates the source of the Timestamp system time: This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INTRNL): Internal 0x1 (EXTRNL): External 0x2 (BOTH): Both 0x3 (RSVD): Reserved <p>Value After Reset: 0x0</p>
24	MACADDR64SEL	R	<p>MAC Addresses 64-127 Selected</p> <p>This bit is set to 1 when the Enable Additional 64 MAC Address Registers (64-127) option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MAC Addresses 64-127 Select option is not selected 0x1 (ACTIVE): MAC Addresses 64-127 Select option is selected <p>Value After Reset: 0x0</p>
23	MACADDR32SEL	R	<p>MAC Addresses 32-63 Selected</p> <p>This bit is set to 1 when the Enable Additional 32 MAC Address Registers (32-63) option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MAC Addresses 32-63 Select option is not selected 0x1 (ACTIVE): MAC Addresses 32-63 Select option is selected <p>Value After Reset: 0x0</p>
22:18	ADDMACADRSEL	R	<p>MAC Addresses 1-31 Selected</p> <p>This bit is set to 1 when the non-zero value is selected for Enable Additional 1-31 MAC Address Registers option</p> <p>Value After Reset: 0x8</p>
17	Reserved_17	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
16	RXCOESEL	R	<p>Receive Checksum Offload Enabled</p> <p>This bit is set to 1 when the Enable Receive TCP/IP Checksum</p>

Bits	Name	Memory Access	Description
			<p>Check option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Receive Checksum Offload Enable option is not selected 0x1 (ACTIVE): Receive Checksum Offload Enable option is selected <p>Value After Reset: 0x1</p>
15	Reserved_15	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
14	TXCOESEL	R	<p>Transmit Checksum Offload Enabled</p> <p>This bit is set to 1 when the Enable Transmit TCP/IP Checksum Insertion option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Transmit Checksum Offload Enable option is not selected 0x1 (ACTIVE): Transmit Checksum Offload Enable option is selected <p>Value After Reset: 0x0</p>
13	EEESEL	R	<p>Energy Efficient Ethernet Enabled</p> <p>This bit is set to 1 when the Enable Energy Efficient Ethernet (EEE) option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Energy Efficient Ethernet Enable option is not selected 0x1 (ACTIVE): Energy Efficient Ethernet Enable option is selected <p>Value After Reset: 0x0</p>
12	TSSEL	R	<p>IEEE 1588-2008 Timestamp Enabled</p> <p>This bit is set to 1 when the Enable IEEE 1588 Timestamp Support option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): IEEE 1588-2008 Timestamp Enable option is not selected 0x1 (ACTIVE): IEEE 1588-2008 Timestamp Enable option is selected <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
11:10	Reserved_11_10	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
9	ARPOFFSEL	R	<p>ARP Offload Enabled</p> <p>This bit is set to 1 when the Enable IPv4 ARP Offload option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): ARP Offload Enable option is not selected 0x1 (ACTIVE): ARP Offload Enable option is selected <p>Value After Reset: 0x0</p>
8	MMCSEL	R	<p>RMON Module Enable</p> <p>This bit is set to 1 when the Enable MAC Management Counters (MMC) option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): RMON Module Enable option is not selected 0x1 (ACTIVE): RMON Module Enable option is selected <p>Value After Reset: 0x0</p>
7	MGKSEL	R	<p>PMT Magic Packet Enable</p> <p>This bit is set to 1 when the Enable Magic Packet Detection option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): PMT Magic Packet Enable option is not selected 0x1 (ACTIVE): PMT Magic Packet Enable option is selected <p>Value After Reset: 0x0</p>
6	RWKSEL	R	<p>PMT Remote Wake-up Packet Enable</p> <p>This bit is set to 1 when the Enable Remote Wake-Up Packet Detection option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): PMT Remote Wake-up Packet Enable option is not selected 0x1 (ACTIVE): PMT Remote Wake-up Packet Enable option is selected <p>Value After Reset: 0x0</p>
5	SMASEL	R	SMA (MDIO) Interface

Bits	Name	Memory Access	Description
			<p>This bit is set to 1 when the Enable Station Management (MDIO Interface) option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): SMA (MDIO) Interface not selected 0x1 (ACTIVE): SMA (MDIO) Interface selected <p>Value After Reset: 0x1</p>
4	VLHASH	R	<p>VLAN Hash Filter Selected</p> <p>This bit is set to 1 when the Enable VLAN Hash Table Based Filtering option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): VLAN Hash Filter not selected 0x1 (ACTIVE): VLAN Hash Filter selected <p>Value After Reset: 0x1</p>
3	PCSSEL	R	<p>PCS Registers (TBI, SGMII, or RTBI PHY interface)</p> <p>This bit is set to 1 when the TBI, SGMII, or RTBI PHY interface option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): No PCS Registers (TBI, SGMII, or RTBI PHY interface) 0x1 (ACTIVE): PCS Registers (TBI, SGMII, or RTBI PHY interface) <p>Value After Reset: 0x0</p>
2	HDSEL	R	<p>Half-duplex Support</p> <p>This bit is set to 1 when the half-duplex mode is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): No Half-duplex support 0x1 (ACTIVE): Half-duplex support <p>Value After Reset: 0x1</p>
1	GMIISEL	R	<p>1000 Mbps Support</p> <p>This bit is set to 1 when 1000 Mbps is selected as the Mode of Operation</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): No 1000 Mbps support 0x1 (ACTIVE): 1000 Mbps support

Bits	Name	Memory Access	Description
			Value After Reset: 0x1
0	MIISEL	R	<p>10 or 100 Mbps Support This bit is set to 1 when 10/100 Mbps is selected as the Mode of Operation</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): No 10 or 100 Mbps support 0x1 (ACTIVE): 10 or 100 Mbps support <p>Value After Reset: 0x1</p>

MAC_HW_Feature1

Description: This register indicates the presence of second set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Note: All bits are set or reset according to the features selected while configuring the IP in coreConsultant.

Size: 32 bits

Offset: 0x120

31	30:2	26	25:2	23	22	21	20	19	18	17	16	15:1	13	12	11	10:6	5	4:0
Reserved_31	L3L4FNUM	Reserved_26	HASHTBLSZ	POUOST	Reserved_22	RAVSEL	AVSEL	DBGMEMA	TSOEN	SPHEN	DCBEN	ADDR64	ADVTHWORD	PTOPEN	OSTEN	TXFIFOSIZE	SPRAM	RXFIFOSIZE

Fields for Register: MAC_HW_Feature1

Bits	Name	Memory Access	Description
31	Reserved_31	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
30:27	L3L4FNUM	R	<p>Total number of L3 or L4 Filters This field indicates the total number of L3 or L4 filters:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NOFILT): No L3 or L4 Filter

Bits	Name	Memory Access	Description
			<p>0x1 (M_1FILT): 1 L3 or L4 Filter 0x2 (M_2FILT): 2 L3 or L4 Filters 0x3 (M_3FILT): 3 L3 or L4 Filters 0x4 (M_4FILT): 4 L3 or L4 Filters 0x5 (M_5FILT): 5 L3 or L4 Filters 0x6 (M_6FILT): 6 L3 or L4 Filters 0x7 (M_7FILT): 7 L3 or L4 Filters 0x8 (M_8FILT): 8 L3 or L4 Filters</p> <p>Value After Reset: 0x4</p>
26	Reserved_26	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
25:24	HASHTBLSZ	R	<p>Hash Table Size</p> <p>This field indicates the size of the hash table:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_HT): No hash table 0x1 (M_64): 64 0x2 (M_128): 128 0x3 (M_256): 256 <p>Value After Reset: 0x1</p>
23	POUOST	R	<p>One Step for PTP over UDP/IP Feature Enable</p> <p>This bit is set to 1 when the Enable One step timestamp for PTP over UDP/IP feature is selected.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): One Step for PTP over UDP/IP Feature is not selected 0x1 (ACTIVE): One Step for PTP over UDP/IP Feature is selected <p>Value After Reset: 0x0</p>
22	Reserved_22	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
21	RAVSEL	R	<p>Rx Side Only AV Feature Enable</p> <p>This bit is set to 1 when the Enable Audio Video Bridging option on Rx Side Only is selected.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (INACTIVE): Rx Side Only AV Feature is not selected 0x1 (ACTIVE): Rx Side Only AV Feature is selected Value After Reset: 0x0</p>
20	AVSEL	R	<p>AV Feature Enable This bit is set to 1 when the Enable Audio Video Bridging option is selected. Values: 0x0 (INACTIVE): AV Feature is not selected 0x1 (ACTIVE): AV Feature is selected Value After Reset: 0x0</p>
19	DBGMEMA	R	<p>DMA Debug Registers Enable This bit is set to 1 when the Debug Mode Enable option is selected Values: 0x0 (INACTIVE): DMA Debug Registers option is not selected 0x1 (ACTIVE): DMA Debug Registers option is selected Value After Reset: 0x0</p>
18	TSOEN	R	<p>TCP Segmentation Offload Enable This bit is set to 1 when the Enable TCP Segmentation Offloading for TCP/IP Packets option is selected Values: 0x0 (INACTIVE): TCP Segmentation Offload Feature is not selected 0x1 (ACTIVE): TCP Segmentation Offload Feature is selected Value After Reset: 0x0</p>
17	SPHEN	R	<p>Split Header Feature Enable This bit is set to 1 when the Enable Split Header Structure option is selected Values: 0x0 (INACTIVE): Split Header Feature is not selected 0x1 (ACTIVE): Split Header Feature is selected Value After Reset: 0x0</p>
16	DCBEN	R	<p>DCB Feature Enable This bit is set to 1 when the Enable Data Center Bridging option is</p>

Bits	Name	Memory Access	Description
			<p>selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): DCB Feature is not selected 0x1 (ACTIVE): DCB Feature is selected <p>Value After Reset: 0x0</p>
15:14	ADDR64	R	<p>Address Width.</p> <p>This field indicates the configured address width:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_32): 32 0x1 (M_40): 40 0x2 (M_48): 48 0x3 (RSVD): Reserved <p>Value After Reset: 0x2</p>
13	ADVTHWORD	R	<p>IEEE 1588 High Word Register Enable</p> <p>This bit is set to 1 when the Add IEEE 1588 Higher Word Register option is selected</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): IEEE 1588 High Word Register option is not selected 0x1 (ACTIVE): IEEE 1588 High Word Register option is selected <p>Value After Reset: 0x0</p>
12	PTOEN	R	<p>PTP Offload Enable</p> <p>This bit is set to 1 when the Enable PTP Timestamp Offload Feature is selected.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): PTP Offload feature is not selected 0x1 (ACTIVE): PTP Offload feature is selected <p>Value After Reset: 0x0</p>
11	OSTEN	R	<p>One-Step Timestamping Enable</p> <p>This bit is set to 1 when the Enable One-Step Timestamp Feature is selected.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): One-Step Timestamping feature is not selected 0x1 (ACTIVE): One-Step Timestamping feature is selected

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
10:6	TXFIFOSIZE	R	<p>MTL Transmit FIFO Size</p> <p>This field contains the configured value of MTL Tx FIFO in bytes expressed as Log to base 2 minus 7, that is, $\text{Log}_2(\text{TXFIFO_SIZE}) - 7$:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_128B): 128 bytes 0x1 (M_256B): 256 bytes 0x2 (M_512B): 512 bytes 0x3 (M_1024B): 1024 bytes 0x4 (M_2048B): 2048 bytes 0x5 (M_4096B): 4096 bytes 0x6 (M_8192B): 8192 bytes 0x7 (M_16384B): 16384 bytes 0x8 (M_32KB): 32 KB 0x9 (M_64KB): 64 KB 0xa (M_128KB): 128 KB 0xb (RSVD): Reserved
			Value After Reset: 0x5
5	SPRAM	R	<p>Single Port RAM Enable</p> <p>This bit is set to 1 when the Use single port RAM Feature is selected.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Single Port RAM feature is not selected 0x1 (ACTIVE): Single Port RAM feature is selected
4:0	RXFIFOSIZE	R	<p>MTL Receive FIFO Size</p> <p>This field contains the configured value of MTL Rx FIFO in bytes expressed as Log to base 2 minus 7, that is, $\text{Log}_2(\text{RXFIFO_SIZE}) - 7$:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_128B): 128 bytes 0x1 (M_256B): 256 bytes 0x2 (M_512B): 512 bytes 0x3 (M_1024B): 1024 bytes 0x4 (M_2048B): 2048 bytes

Bits	Name	Memory Access	Description
			<p>0x5 (M_4096B): 4096 bytes 0x6 (M_8192B): 8192 bytes 0x7 (M_16384B): 16384 bytes 0x8 (M_32KB): 32 KB 0x9 (M_64KB): 64 KB 0xa (M_128KB): 128 KB 0xb (M_256KB): 256 KB 0xc (RSVD): Reserved</p> <p>Value After Reset: 0x5</p>

MAC_HW_Feature2

Description: This register indicates the presence of third set of the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Size: 32 bits

Offset: 0x124

31	30:28	27	26:24	23:22	21:18	17:16	15:12	11:10	9:6	5:4	3:0
Reserved_31	AUXSNAPNUM	Reserved_27	PPSOUTNUM	TDCSZ	TXCHCNT	RDCSZ	RXCHCNT	Reserved_11_10	TXQCNT	Reserved_5_4	RXQCNT

Fields for Register: MAC_HW_Feature2

Bits	Name	Memory Access	Description
31	Reserved_31	R	<p>Reserved. Value After Reset: 0x0</p>
30:28	AUXSNAPNUM	R	<p>Number of Auxiliary Snapshot Inputs This field indicates the number of auxiliary snapshot inputs: Values: 0x0 (NO_AUXI): No auxiliary input 0x1 (M_1_AUXI): 1 auxiliary input</p>

Bits	Name	Memory Access	Description
			<p>0x2 (M_2_AUXI): 2 auxiliary input 0x3 (M_3_AUXI): 3 auxiliary input 0x4 (M_4_AUXI): 4 auxiliary input 0x5 (RSVD): Reserved</p> <p>Value After Reset: 0x0</p>
27	Reserved_27	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
26:24	PPSOUTNUM	R	<p>Number of PPS Outputs</p> <p>This field indicates the number of PPS outputs:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_PPSO): No PPS output 0x1 (M_1_PPSO): 1 PPS output 0x2 (M_2_PPSO): 2 PPS output 0x3 (M_3_PPSO): 3 PPS output 0x4 (M_4_PPSO): 4 PPS output 0x5 (RSVD): Reserved <p>Value After Reset: 0x0</p>
23:22	TDCSZ	R	<p>Tx DMA Descriptor Cache Size in terms of 16 bytes descriptors:</p> <ul style="list-style-type: none"> 00 - Cache Not configured 01 - 4 16 bytes descriptor 10 - 8 16 bytes descriptor 11 - 16 16 bytes descriptor <p>Value After Reset: 0x1</p>
21:18	TXCHCNT	R	<p>Number of DMA Transmit Channels</p> <p>This field indicates the number of DMA Transmit channels:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_1TXCH): 1 MTL Tx Channel 0x1 (M_2TXCH): 2 MTL Tx Channels 0x2 (M_3TXCH): 3 MTL Tx Channels 0x3 (M_4TXCH): 4 MTL Tx Channels 0x4 (M_5TXCH): 5 MTL Tx Channels

Bits	Name	Memory Access	Description
			<p>0x5 (M_6TXCH): 6 MTL Tx Channels 0x6 (M_7TXCH): 7 MTL Tx Channels 0x7 (M_8TXCH): 8 MTL Tx Channels 0x0 (NO_DCACHE): Desc Cache not configured 0x1 (M_1TDCSZ): 4 0x2 (M_2TDCSZ): 8 0x3 (M_3TDCSZ): 16</p> <p>Value After Reset: 0x3</p>
17:16	RDCSZ	R	<p>Rx DMA Descriptor Cache Size in terms of 16 bytes descriptors:</p> <p>00 - Cache Not configured 01 - 4 16 bytes descriptor 10 - 8 16 bytes descriptor 11 - 16 16 bytes descriptor</p> <p>Value After Reset: 0x1</p>
15:12	RXCHCNT	R	<p>Number of DMA Receive Channels</p> <p>This field indicates the number of DMA Receive channels:</p> <p>Values:</p> <p>0x0 (M_1RXCH): 1 MTL Rx Channel 0x1 (M_2RXCH): 2 MTL Rx Channels 0x2 (M_3RXCH): 3 MTL Rx Channels 0x3 (M_4RXCH): 4 MTL Rx Channels 0x4 (M_5RXCH): 5 MTL Rx Channels 0x5 (M_6RXCH): 6 MTL Rx Channels 0x6 (M_7RXCH): 7 MTL Rx Channels 0x7 (M_8RXCH): 8 MTL Rx Channels 0x0 (NO_DCACHE): Desc Cache not configured 0x1 (M_1RDCSZ): 4 0x2 (M_2RDCSZ): 8 0x3 (M_3RDCSZ): 16</p> <p>Value After Reset: 0x3</p>
11:10	Reserved_11_10	R	Reserved.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
9:6	TXQCN	R	<p>Number of MTL Transmit Queues</p> <p>This field indicates the number of MTL Transmit queues:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_1TXQ): 1 MTL Tx Queue 0x1 (M_2TXQ): 2 MTL Tx Queues 0x2 (M_3TXQ): 3 MTL Tx Queues 0x3 (M_4TXQ): 4 MTL Tx Queues 0x4 (M_5TXQ): 5 MTL Tx Queues 0x5 (M_6TXQ): 6 MTL Tx Queues 0x6 (M_7TXQ): 7 MTL Tx Queues 0x7 (M_8TXQ): 8 MTL Tx Queues <p>Value After Reset: 0x3</p>
5:4	Reserved_5_4	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
3:0	RXQCN	R	<p>Number of MTL Receive Queues</p> <p>This field indicates the number of MTL Receive queues:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_1RXQ): 1 MTL Rx Queue 0x1 (M_2RXQ): 2 MTL Rx Queues 0x2 (M_3RXQ): 3 MTL Rx Queues 0x3 (M_4RXQ): 4 MTL Rx Queues 0x4 (M_5RXQ): 5 MTL Rx Queues 0x5 (M_6RXQ): 6 MTL Rx Queues 0x6 (M_7RXQ): 7 MTL Rx Queues 0x7 (M_8RXQ): 8 MTL Rx Queues <p>Value After Reset: 0x3</p>

MAC_HW_Feature3

Description: This register indicates the presence of fourth set the optional features or functions of the DWC_ether_qos. The software driver can use this register to dynamically enable or disable the programs related to the optional blocks.

Size: 32 bits

Offset: 0x128

31:3 0	29:2 8	27	26	25:2 2	21:2 0	19:1 7	16	15	14:1 3	12:1 1	10	9	8:6	5	4	3	2:0
Reserved_31_30	ASP	TBSSEL	FPESEL	Reserved_25_22	ESTWID	ESTDEP	ESTSEL	Reserved_15	FRPES	FRPBS	FRPSEL	PDUPSEL	Reserved_7_6	DVLAN	CBTISEL	Reserved_3	NRVF

Fields for Register: MAC_HW_Feature3

Bits	Name	Memory Access	Description
31:30	Reserved_31_30	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
29:28	ASP	R	<p>Automotive Safety Package</p> <p>Following are the encoding for the different Safety features</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NONE): No Safety features selected 0x1 (ECC_ONLY): Only "ECC protection for external memory" feature is selected 0x2 (AS_NPPE): All the Automotive Safety features are selected without the "Parity Port Enable for external interface" feature 0x3 (AS_PPE): All the Automotive Safety features are selected with the "Parity Port Enable for external interface" feature <p>Value After Reset: 0x0</p>
27	TBSSEL	R	<p>Time Based Scheduling Enable</p> <p>This bit is set to 1 when the Time Based Scheduling feature is selected.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Time Based Scheduling Enable feature is not selected 0x1 (ACTIVE): Time Based Scheduling Enable feature is selected <p>Value After Reset: 0x0</p>
26	FPESEL	R	<p>Frame Preemption Enable</p> <p>This bit is set to 1 when the Enable Frame preemption feature is selected.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (INACTIVE): Frame Preemption Enable feature is not selected</p> <p>0x1 (ACTIVE): Frame Preemption Enable feature is selected</p> <p>Value After Reset: 0x0</p>
25:22	Reserved_25_22	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
21:20	ESTWID	R	<p>Width of the Time Interval field in the Gate Control List</p> <p>This field indicates the width of the Configured Time Interval Field</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NOWIDTH): Width not configured 0x1 (WIDTH16): 16 0x2 (WIDTH20): 20 0x3 (WIDTH24): 24 <p>Value After Reset: 0x0</p>
19:17	ESTDEP	R	<p>Depth of the Gate Control List</p> <p>This field indicates the depth of Gate Control list expressed as Log2(DWC_EQOS_EST_DEP)-5</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NODEPTH): No Depth configured 0x1 (DEPTH64): 64 0x2 (DEPTH128): 128 0x3 (DEPTH256): 256 0x4 (DEPTH512): 512 0x5 (DEPTH1024): 1024 0x6 (RSVD): Reserved <p>Value After Reset: 0x0</p>
16	ESTSEL	R	<p>Enhancements to Scheduled Traffic Enable</p> <p>This bit is set to 1 when the Enable Enhancements to Scheduling Traffic feature is selected.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Enable Enhancements to Scheduling Traffic feature is not selected 0x1 (ACTIVE): Enable Enhancements to Scheduling Traffic feature is selected

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
15	Reserved_15	R	Reserved. Value After Reset: 0x0
14:13	FRPES	R	Flexible Receive Parser Table Entries size This field indicates the Max Number of Parser Entries supported by Flexible Receive Parser. Values: 0x0 (M_64ENTR): 64 Entries 0x1 (M_128ENTR): 128 Entries 0x2 (M_256ENTR): 256 Entries 0x3 (RSVD): Reserved Value After Reset: 0x0
12:11	FRPBS	R	Flexible Receive Parser Buffer size This field indicates the supported Max Number of bytes of the packet data to be Parsed by Flexible Receive Parser. Values: 0x0 (M_64BYTES): 64 Bytes 0x1 (M_128BYTES): 128 Bytes 0x2 (M_256BYTES): 256 Bytes 0x3 (RSVD): Reserved Value After Reset: 0x0
10	FRPSEL	R	Flexible Receive Parser Selected This bit is set to 1 when the Enable Flexible Programmable Receive Parser option is selected. Values: 0x0 (INACTIVE): Flexible Receive Parser feature is not selected 0x1 (ACTIVE): Flexible Receive Parser feature is selected Value After Reset: 0x0
9	PDUPSEL	R	Broadcast/Multicast Packet Duplication This bit is set to 1 when the Broadcast/Multicast Packet Duplication feature is selected. Values: 0x0 (INACTIVE): Broadcast/Multicast Packet Duplication

Bits	Name	Memory Access	Description
			<p>feature is not selected</p> <p>0x1 (ACTIVE): Broadcast/Multicast Packet Duplication feature is selected</p> <p>Value After Reset: 0x0</p>
8:6	Reserved_7_6	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
5	DVLAN	R	<p>Double VLAN Tag Processing Selected</p> <p>This bit is set to 1 when the Enable Double VLAN Processing Feature is selected.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Double VLAN option is not selected 0x1 (ACTIVE): Double VLAN option is selected <p>Value After Reset: 0x0</p>
4	CBTISEL	R	<p>Queue/Channel based VLAN tag insertion on Tx Enable</p> <p>This bit is set to 1 when the Enable Queue/Channel based VLAN tag insertion on Tx Feature is selected.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Enable Queue/Channel based VLAN tag insertion on Tx feature is not selected 0x1 (ACTIVE): Enable Queue/Channel based VLAN tag insertion on Tx feature is selected <p>Value After Reset: 0x0</p>
3	Reserved_3	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
2:0	NRVF	R	<p>Number of Extended VLAN Tag Filters Enabled</p> <p>This field indicates the Number of Extended VLAN Tag Filters selected:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_ERVLAN): No Extended Rx VLAN Filters 0x1 (M_4_ERVLAN): 4 Extended Rx VLAN Filters 0x2 (M_8_ERVLAN): 8 Extended Rx VLAN Filters 0x3 (M_16_ERVLAN): 16 Extended Rx VLAN Filters 0x4 (M_24_ERVLAN): 24 Extended Rx VLAN Filters 0x5 (M_32_ERVLAN): 32 Extended Rx VLAN Filters

Bits	Name	Memory Access	Description
			0x6 (RSVD): Reserved Value After Reset: 0x1

MAC_MDIO_Address

Description: The MDIO Address register controls the management cycles to external PHY through a management interface.

Size: 32 bits

Offset: 0x200

31:28	27	26	25:2	20:1	15	14:1	11:	7:5	4	3	2	1	0
Reserved_31_28	PS_E	BT_B	PA	RDA	Reserved_15	NTC	CR	Reserved_7_5	SKA_P	GOC_1	GOC_0	C45_E	G_B

Fields for Register: MAC_MDIO_Address

Bits	Name	Memory Access	Description
31:28	Reserved_31_28	R	Reserved. Value After Reset: 0x0
27	PSE	R/W	Preamble Suppression Enable When this bit is set, the SMA suppresses the 32-bit preamble and transmits MDIO frames with only 1 preamble bit. When this bit is 0, the MDIO frame always has 32 bits of preamble as defined in the IEEE specifications. Values: 0x0 (DISABLE): Preamble Suppression disabled 0x1 (ENABLE): Preamble Suppression enabled Value After Reset: 0x0
26	BTB	R/W	Back to Back transactions When this bit is set and the NTC has value greater than 0, then the MAC informs the completion of a read or write command at the end of frame transfer (before the trailing clocks are transmitted). The software can thus initiate the next command which is executed immediately irrespective of the number trailing clocks generated for the previous frame. When this bit is reset, then the read/write command completion (GB is cleared) only after the trailing clocks are generated. In this mode, it is ensured that the NTC is always generated after each frame. This bit must not be set when NTC=0.

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x0 (DISABLE): Back to Back transactions disabled 0x1 (ENABLE): Back to Back transactions enabled</p> <p>Value After Reset: 0x0</p>
25:21	PA	R/W	<p>Physical Layer Address</p> <p>This field indicates which Clause 22 PHY devices (out of 32 devices) the MAC is accessing. This field indicates which Clause 45 capable PHYs (out of 32 PHYs) the MAC is accessing.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
20:16	RDA	R/W	<p>Register/Device Address</p> <p>These bits select the PHY register in selected Clause 22 PHY device. These bits select the Device (MMD) in selected Clause 45 capable PHY.</p> <p>Value After Reset: 0x0</p>
15	Reserved_15	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
14:12	NTC	R/W	<p>Number of Trailing Clocks</p> <p>This field controls the number of trailing clock cycles generated on gmii_mdc_o (MDC) after the end of transmission of MDIO frame. The valid values can be from 0 to 7. Programming the value to 3'h3 indicates that there are additional three clock cycles on the MDC line after the end of MDIO frame transfer.</p> <p>Value After Reset: 0x0</p>
11:8	CR	R/W	<p>CSR Clock Range</p> <p>The CSR Clock Range selection determines the frequency of the MDC clock according to the CSR clock frequency used in your design:</p> <ul style="list-style-type: none"> 0000: CSR clock = 60-100 MHz; MDC clock = CSR clock/42 0001: CSR clock = 100-150 MHz; MDC clock = CSR clock/62 0010: CSR clock = 20-35 MHz; MDC clock = CSR clock/16 0011: CSR clock = 35-60 MHz; MDC clock = CSR clock/26 0100: CSR clock = 150-250 MHz; MDC clock = CSR clock/102 0101: CSR clock = 250-300 MHz; MDC clock = CSR clock/124 0110: CSR clock = 300-500 MHz; MDC clock = CSR clock/204

Bits	Name	Memory Access	Description
			<p>0111: CSR clock = 500-800 MHz; MDC clock = CSR clock/324</p> <p>The suggested range of CSR clock frequency applicable for each value (when Bit 11 = 0) ensures that the MDC clock is approximately between 1.0 MHz to 2.5 MHz frequency range.</p> <p>When Bit 11 is set, you can achieve a higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 1010, the resultant MDC clock is of 12.5 MHz which is beyond the range specified in IEEE 802.3. Program the following values only if the interfacing chips support faster MDC clocks:</p> <ul style="list-style-type: none"> 1000: CSR clock/4 1001: CSR clock/6 1010: CSR clock/8 1011: CSR clock/10 1100: CSR clock/12 1101: CSR clock/14 1110: CSR clock/16 1111: CSR clock/18 <p>These bits are not used for accessing RevMII. These bits are read-only if the RevMII interface is selected as single PHY interface.</p> <p>Value After Reset: 0x0</p>
7:5	Reserved_7_5	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
4	SKAP	R/W	<p>Skip Address Packet</p> <p>When this bit is set, the SMA does not send the address packets before read, write, or post-read increment address packets. This bit is valid only when C45E is set.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Skip Address Packet is disabled 0x1 (ENABLE): Skip Address Packet is enabled <p>Value After Reset: 0x0</p>
3	GOC_1	R/W	<p>GMII Operation Command 1</p> <p>This bit is higher bit of the operation command to the PHY or RevMII, GOC_1 and GOC_0 is encoded as follows:</p> <ul style="list-style-type: none"> 00: Reserved 01: Write

Bits	Name	Memory Access	Description
			<p>10: Post Read Increment Address for Clause 45 PHY 11: Read</p> <p>When Clause 22 PHY or RevMII is enabled, only Write and Read commands are valid.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): GMII Operation Command 1 is disabled 0x1 (ENABLE): GMII Operation Command 1 is enabled <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
2	GOC_0	R/W	<p>GMII Operation Command 0</p> <p>This is the lower bit of the operation command to the PHY or RevMII. When in SMA mode (MDIO master) this bit along with GOC_1 determines the operation to be performed to the PHY. When only RevMII is selected in configuration this bit is read-only and tied to 1.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): GMII Operation Command 0 is disabled 0x1 (ENABLE): GMII Operation Command 0 is enabled <p>Value After Reset: 0x0</p>
1	C45E	R/W	<p>Clause 45 PHY Enable</p> <p>When this bit is set, Clause 45 capable PHY is connected to MDIO. When this bit is reset, Clause 22 capable PHY is connected to MDIO.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Clause 45 PHY is disabled 0x1 (ENABLE): Clause 45 PHY is enabled <p>Value After Reset: 0x0</p>
0	GB	R/W	<p>GMII Busy</p> <p>The application sets this bit to instruct the SMA to initiate a Read or Write access to the MDIO slave. The MAC clears this bit after the MDIO frame transfer is completed. Hence the software must not write or change any of the fields in MAC_MDIO_Address and MAC_MDIO_Data registers as long as this bit is set.</p> <p>For write transfers, the application must first write 16-bit data in the GDI field (and also RA field when C45E is set) in MAC_MDIO_Data register before setting this bit. When C45E is set, it should also write into the RA field of MAC_MDIO_Data register before initiating a read transfer. When a read transfer is completed (GB=0), the data read from the PHY register is valid in the GD field of the MAC_MDIO_Data register.</p>

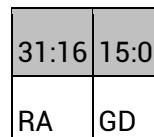
Bits	Name	Memory Access	Description
			<p>Note: Even if the addressed PHY is not present, there is no change in the functionality of this bit.</p> <p>Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): GMII Busy is disabled 0x1 (ENABLE): GMII Busy is enabled <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MAC_MDIO_Data

Description: The MDIO Data register stores the Write data to be written to the PHY register located at the address specified in MAC_MDIO_Address. This register also stores the Read data from the PHY register located at the address specified by MDIO Address register.

Size: 32 bits

Offset: 0x204



Fields for Register: MAC_MDIO_Data

Bits	Name	Memory Access	Description
31:16	RA	R/W	<p>Register Address</p> <p>This field is valid only when C45E is set. It contains the Register Address in the PHY to which the MDIO frame is intended for.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
15:0	GD	R/W	<p>GMII Data</p> <p>This field contains the 16-bit data value read from the PHY or RevMII after a Management Read operation or the 16-bit data value to be written to the PHY or RevMII before a Management Write operation.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MAC_CSR_SW_Ctrl

Description: This register contains software programmable controls for changing the CSR access response and status bits clearing.

Size: 32 bits

Offset: 0x230

31:9	8	7:1	0
Reserved_31_9	SEEN	Reserved_7_1	RCWE

Fields for Register: MAC_CSR_SW_Ctrl

Bits	Name	Memory Access	Description
31:9	Reserved_31_9	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
8	SEEN	R/W	<p>Slave Error Response Enable</p> <p>When this bit is set, the MAC responds with Slave Error for accesses to reserved registers in CSR space.</p> <p>When this bit is reset, the MAC responds with Okay response to any register accessed from CSR space.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Slave Error Response is disabled 0x1 (ENABLE): Slave Error Response is enabled <p>Value After Reset: 0x0</p>
7:1	Reserved_7_1	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
0	RCWE	R/W	<p>Register Clear on Write 1 Enable</p> <p>When this bit is set, the access mode of some register fields changes to Clear on Write 1, the application needs to set that respective bit to 1 to clear it.</p> <p>When this bit is reset, the access mode of these register fields remain as Clear on Read.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Register Clear on Write 1 is disabled 0x1 (ENABLE): Register Clear on Write 1 is enabled <p>Value After Reset: 0x0</p>

MAC_Address0_High

Description: The MAC Address0 High register holds the upper 16 bits of the first 6-byte MAC address of the station. The first DA byte that is received on the (G)MII interface corresponds to the LS byte (Bits [7:0]) of the MAC Address Low register. For example, if 0x112233445566 is received (0x11 in lane 0 of the first column) on the (G)MII as the destination address, then the MacAddress0 Register [47:0] is compared with 0x665544332211.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, then the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address0 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Size: 32 bits

Offset: 0x300

31	30:18	17:16	15:0
AE	Reserved_30_y	DCS	ADDRHI

Fields for Register: MAC_Address0_High

Bits	Name	Memory Access	Description
31	AE	R	<p>Address Enable</p> <p>This bit is always set to 1.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): INVALID : This bit must be always set to 1 0x1 (ENABLE): This bit is always set to 1 <p>Value After Reset: 0x1</p>
30:18	Reserved_30_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
17:16	DCS	R/W	<p>DMA Channel Select</p> <p>If the PDC bit of MAC_Ext_Configuration register is not set:</p> <p>This field contains the binary representation of the DMA Channel number to which an Rx packet whose DA matches the MAC Address0 content is routed.</p> <p>If the PDC bit of MAC_Ext_Configuration register is set:</p> <p>This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address0 content is routed.</p> <p>Value After Reset: 0x0</p>
15:0	ADDRHI	R/W	<p>MAC Address0[47:32]</p> <p>This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. The MAC uses this field for filtering the received packets</p>

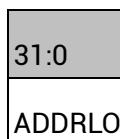
Bits	Name	Memory Access	Description
			<p>and inserting the MAC address in the Transmit Flow Control (Pause) Packets.</p> <p>Value After Reset: 0xffff</p>

MAC_Address0_Low

Description: The MAC Address0 Low register holds the lower 32 bits of the 6-byte first MAC address of the station.

Size: 32 bits

Offset: 0x304



Fields for Register: MAC_Address0_Low

Bits	Name	Memory Access	Description
31:0	ADDRLO	R/W	<p>MAC Address0[31:0]</p> <p>This field contains the lower 32 bits of the first 6-byte MAC address. The MAC uses this field for filtering the received packets and inserting the MAC address in the Transmit Flow Control (Pause) Packets.</p> <p>Value After Reset: 0xffffffff</p>

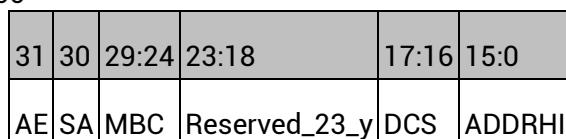
MAC_Address(#i)_High (for i = 1; i <= 8)

Description: The MAC Address1 High register holds the upper 16 bits of the second 6-byte MAC address of the station.

If the MAC address registers are configured to be double-synchronized to the (G)MII clock domains, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the MAC Address1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low Register should be performed after at least four clock cycles in the destination clock domain.

Size: 32 bits

Offset: (0x0008*i)+0x0300



Fields for Register: MAC_Address(#i)_High (for i = 1; i <= 8)

Bits	Name	Memory Access	Description
31	AE	R/W	<p>Address Enable</p> <p>When this bit is set, the address filter module uses the second MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Address is ignored 0x1 (ENABLE): Address is enabled <p>Value After Reset: 0x0</p>
30	SA	R/W	<p>Source Address</p> <p>When this bit is set, the MAC Address1[47:0] is used to compare with the SA fields of the received packet. When this bit is reset, the MAC Address1[47:0] is used to compare with the DA fields of the received packet.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DA): Compare with Destination Address 0x1 (SA): Compare with Source Address <p>Value After Reset: 0x0</p>
29:24	MBC	R/W	<p>Mask Byte Control</p> <p>These bits are mask control bits for comparing each of the MAC Address bytes. When set high, the MAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address1 registers. Each bit controls the masking of the bytes as follows:</p> <ul style="list-style-type: none"> Bit 29: MAC_Address\${i}_High[15:8] Bit 28: MAC_Address\${i}_High[7:0] Bit 27: MAC_Address\${i}_Low[31:24] .. Bit 24: MAC_Address\${i}_Low[7:0] <p>You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.</p> <p>Value After Reset: 0x0</p>
23:18	Reserved_23_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
17:16	DCS	R/W	<p>DMA Channel Select</p> <p>If the PDC bit of MAC_Ext_Configuration register is not set: This field contains the binary representation of the DMA Channel</p>

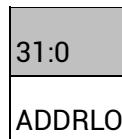
Bits	Name	Memory Access	Description
			<p>number to which an Rx packet whose DA matches the MAC Address(#i) content is routed.</p> <p>If the PDC bit of MAC_Ext_Configuration register is set:</p> <p>This field contains the one-hot representation of one or more DMA Channel numbers to which an Rx packet whose DA matches the MAC Address(#i) content is routed.</p> <p>Value After Reset: 0x0</p>
15:0	ADDRHI	R/W	<p>MAC Address1 [47:32]</p> <p>This field contains the upper 16 bits[47:32] of the second 6-byte MAC address.</p> <p>Value After Reset: 0xffff</p>

MAC_Address(#i)_Low (for i = 1; i <= 8)

Description: The MAC Address1 Low register holds the lower 32 bits of the second 6-byte MAC address of the station.

Size: 32 bits

Offset: (0x0008*i)+0x0304



Fields for Register: MAC_Address(#i)_Low (for i = 1; i <= 8)

Bits	Name	Memory Access	Description
31:0	ADDRLO	R/W	<p>MAC Address1 [31:0]</p> <p>This field contains the lower 32 bits of second 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.</p> <p>Value After Reset: 0xffffffff</p>

MAC_L3_L4_Control(#i) (for i = 0; i <= 3)

Description: The Layer 3 and Layer 4 Control register controls the operations of filter 0 of Layer 3 and Layer 4.

Size: 32 bits

Offset: (0x0030*i)+0x0900

31:2	28	27:2	25:2	23:2	21	20	19	18	17	16	15:1	10:1	5	4	3	2	1	0
------	----	------	------	------	----	----	----	----	----	----	------	------	---	---	---	---	---	---

9		6	4	2						1	6						
Reserved_31_29	DMCHENO	Reserved_27_y	DMCHN0	Reserved_23_22	L4DPIMO	L4DPM0	L4SPIMO	L4SPM0	Reserved_17	L4PENO	L3HDBM0	L3HSBM0	L3DAIM0	L3DAM0	L3SAM0	Reserved_1	L3PENO

Fields for Register: MAC_L3_L4_Control(#i) (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:29	Reserved_31_29	R	Reserved. Value After Reset: 0x0
28	DMCHENO	R/W	DMA Channel Select Enable When set, this bit enables the selection of the DMA channel number for the packet that is passed by this L3_L4 filter. The DMA channel is indicated by the DMCHN bits. When this bit is reset, the DMA channel is not decided by this filter. Values: 0x0 (DISABLE): DMA Channel Select is disabled 0x1 (ENABLE): DMA Channel Select is enabled Value After Reset: 0x0
27:26	Reserved_27_y	R	Reserved. Value After Reset: 0x0
25:24	DMCHN0	R/W	DMA Channel Number When DMCHEN is set high, this field selects the DMA Channel number to which the packet passed by this filter is routed. The width of this field depends on the number of the DMA channels present in your configuration. Value After Reset: 0x0
23:22	Reserved_23_22	R	Reserved. Value After Reset: 0x0
21	L4DPIMO	R/W	Layer 4 Destination Port Inverse Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Destination Port number field is enabled for perfect matching. This bit is valid and applicable only when the L4DPM0 bit is set high. Values: 0x0 (DISABLE): Layer 4 Destination Port Inverse Match is disabled

Bits	Name	Memory Access	Description
			<p>0x1 (ENABLE): Layer 4 Destination Port Inverse Match is enabled Value After Reset: 0x0</p>
20	L4DPM0	R/W	<p>Layer 4 Destination Port Match Enable When this bit is set, the Layer 4 Destination Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Destination Port number field for matching.</p> <p>Values:</p> <p>0x0 (DISABLE): Layer 4 Destination Port Match is disabled 0x1 (ENABLE): Layer 4 Destination Port Match is enabled</p> <p>Value After Reset: 0x0</p>
19	L4SPIMO	R/W	<p>Layer 4 Source Port Inverse Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for inverse matching. When this bit is reset, the Layer 4 Source Port number field is enabled for perfect matching.</p> <p>This bit is valid and applicable only when the L4SPM0 bit is set high.</p> <p>Values:</p> <p>0x0 (DISABLE): Layer 4 Source Port Inverse Match is disabled 0x1 (ENABLE): Layer 4 Source Port Inverse Match is enabled</p> <p>Value After Reset: 0x0</p>
18	L4SPM0	R/W	<p>Layer 4 Source Port Match Enable When this bit is set, the Layer 4 Source Port number field is enabled for matching. When this bit is reset, the MAC ignores the Layer 4 Source Port number field for matching.</p> <p>Values:</p> <p>0x0 (DISABLE): Layer 4 Source Port Match is disabled 0x1 (ENABLE): Layer 4 Source Port Match is enabled</p> <p>Value After Reset: 0x0</p>
17	Reserved_17	R	<p>Reserved. Value After Reset: 0x0</p>
16	L4PEN0	R/W	<p>Layer 4 Protocol Enable When this bit is set, the Source and Destination Port number fields of UDP packets are used for matching. When this bit is reset, the Source and Destination Port number fields of TCP packets are used for matching.</p>

Bits	Name	Memory Access	Description
			<p>The Layer 4 matching is done only when the L4SPM0 or L4DPM0 bit is set.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Layer 4 Protocol is disabled 0x1 (ENABLE): Layer 4 Protocol is enabled <p>Value After Reset: 0x0</p>
15:11	L3HDBM0	R/W	<p>Layer 3 IP DA Higher Bits Match</p> <p>IPv4 Packets:</p> <p>This field contains the number of higher bits of IP Destination Address that are matched in the IPv4 packets. The following list describes the values of this field:</p> <ul style="list-style-type: none"> 0: No bits are masked. 1: LSb[0] is masked 2: Two LSbs [1:0] are masked .. 31: All bits except MSb are masked. <p>IPv6 Packets:</p> <p>Bits[12:11] of this field correspond to Bits[6:5] of L3HSBM0 which indicate the number of lower bits of IP Source or Destination Address that are masked in the IPv6 packets. The following list describes the concatenated values of the L3HDBM0[1:0] and L3HSBM0 bits:</p> <ul style="list-style-type: none"> 0: No bits are masked. 1: LSb[0] is masked. 2: Two LSbs [1:0] are masked .. 127: All bits except MSb are masked. <p>This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set.</p> <p>Value After Reset: 0x0</p>
10:6	L3HSBM0	R/W	<p>Layer 3 IP SA Higher Bits Match</p> <p>IPv4 Packets:</p> <p>This field contains the number of lower bits of IP Source Address that are masked for matching in the IPv4 packets. The following list describes the values of this field:</p> <ul style="list-style-type: none"> 0: No bits are masked.

Bits	Name	Memory Access	Description
			<p>1: LSb[0] is masked 2: Two LSbs [1:0] are masked .. 31: All bits except MSb are masked.</p> <p>IPv6 Packets: This field contains Bits[4:0] of L3HSBMO. These bits indicate the number of higher bits of IP Source or Destination Address matched in the IPv6 packets. This field is valid and applicable only when the L3DAM0 or L3SAM0 bit is set high.</p> <p>Value After Reset: 0x0</p>
5	L3DAIM0	R/W	<p>Layer 3 IP DA Inverse Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for inverse matching. When this bit is reset, the Layer 3 IP Destination Address field is enabled for perfect matching. This bit is valid and applicable only when the L3DAM0 bit is set high.</p> <p>Values: 0x0 (DISABLE): Layer 3 IP DA Inverse Match is disabled 0x1 (ENABLE): Layer 3 IP DA Inverse Match is enabled</p> <p>Value After Reset: 0x0</p>
4	L3DAM0	R/W	<p>Layer 3 IP DA Match Enable When this bit is set, the Layer 3 IP Destination Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Destination Address field for matching.</p> <p>Note: When the L3PENO bit is set, you should set either this bit or the L3SAM0 bit because either IPv6 DA or SA can be checked for filtering.</p> <p>Values: 0x0 (DISABLE): Layer 3 IP DA Match is disabled 0x1 (ENABLE): Layer 3 IP DA Match is enabled</p> <p>Value After Reset: 0x0</p>
3	L3SAIMO	R/W	<p>Layer 3 IP SA Inverse Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for inverse matching. When this bit reset, the Layer 3 IP Source Address field is enabled for perfect matching. This bit is valid and applicable only when the L3SAM0 bit is set.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (DISABLE): Layer 3 IP SA Inverse Match is disabled 0x1 (ENABLE): Layer 3 IP SA Inverse Match is enabled Value After Reset: 0x0</p>
2	L3SAM0	R/W	<p>Layer 3 IP SA Match Enable When this bit is set, the Layer 3 IP Source Address field is enabled for matching. When this bit is reset, the MAC ignores the Layer 3 IP Source Address field for matching. Note: When the L3PEN0 bit is set, you should set either this bit or the L3DAM0 bit because either IPv6 SA or DA can be checked for filtering.</p> <p>Values:</p> <p>0x0 (DISABLE): Layer 3 IP SA Match is disabled 0x1 (ENABLE): Layer 3 IP SA Match is enabled</p> <p>Value After Reset: 0x0</p>
1	Reserved_1	R	<p>Reserved. Value After Reset: 0x0</p>
0	L3PEN0	R/W	<p>Layer 3 Protocol Enable When this bit is set, the Layer 3 IP Source or Destination Address matching is enabled for IPv6 packets. When this bit is reset, the Layer 3 IP Source or Destination Address matching is enabled for IPv4 packets. The Layer 3 matching is done only when the L3SAM0 or L3DAM0 bit is set.</p> <p>Values:</p> <p>0x0 (DISABLE): Layer 3 Protocol is disabled 0x1 (ENABLE): Layer 3 Protocol is enabled</p> <p>Value After Reset: 0x0</p>

MAC_Layer4_Address(#i) (for i = 0; i <= 3)

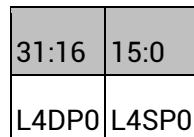
Description: The MAC_Layer4_Address(#i), MAC_L3_L4_Control(#i), MAC_Layer3_Addr0_Reg(#i), MAC_Layer3_Addr1_Reg(#i), MAC_Layer3_Addr2_Reg(#i) and MAC_Layer3_Addr3_Reg(#i) registers are reserved (RO with default value) if Enable Layer 3 and Layer 4 Packet Filter option is not selected while configuring the IP.

You can configure the Layer 3 and Layer 4 Address Registers to be double-synchronized by selecting the Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain option while configuring the IP. When you select this option, the synchronization is triggered only when Bits[31:24] (in little-endian mode) or Bits[7:0] (in big-endian mode) of the Layer 3 and Layer 4 Address Registers are written. For

proper synchronization updates, you should perform consecutive writes to same Layer 3 and Layer 4 Address Registers after at least four clock cycles delay of the destination clock.

Size: 32 bits

Offset: (0x0030*i)+0x0904



Fields for Register: MAC_Layer4_Address(#i) (for i = 0; i <= 3)

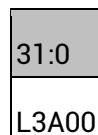
Bits	Name	Memory Access	Description
31:16	L4DP0	R/W	<p>Layer 4 Destination Port Number Field</p> <p>When the L4PEN0 bit is reset and the L4DPM0 bit is set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with the TCP Destination Port Number field in the IPv4 or IPv6 packets.</p> <p>When the L4PEN0 and L4DPM0 bits are set in MAC_L3_L4_Control0 register, this field contains the value to be matched with the UDP Destination Port Number field in the IPv4 or IPv6 packets.</p> <p>Value After Reset: 0x0</p>
15:0	L4SP0	R/W	<p>Layer 4 Source Port Number Field</p> <p>When the L4PEN0 bit is reset and the L4SPM0 bit is set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with the TCP Source Port Number field in the IPv4 or IPv6 packets.</p> <p>When the L4PEN0 and L4SPM0 bits are set in MAC_L3_L4_Control0 register, this field contains the value to be matched with the UDP Source Port Number field in the IPv4 or IPv6 packets.</p> <p>Value After Reset: 0x0</p>

MAC_Layer3_Addr0_Reg(#i) (for i = 0; i <= 3)

Description: For IPv4 packets, the Layer 3 Address 0 Register 0 register contains the 32-bit IP Source Address field. For IPv6 packets, it contains Bits[31:0] of the 128-bit IP Source Address or Destination Address field.

Size: 32 bits

Offset: (0x0030*i)+0x0910



Fields for Register: MAC_Layer3_Addr0_Reg(#i) (for i = 0; i <= 3)

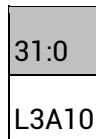
Bits	Name	Memory Access	Description
31:0	L3A00	R/W	<p>Layer 3 Address 0 Field</p> <p>When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits[31:0] of the IP Source Address field in the IPv6 packets.</p> <p>When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits[31:0] of the IP Destination Address field in the IPv6 packets.</p> <p>When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with the IP Source Address field in the IPv4 packets.</p> <p>Value After Reset: 0x0</p>

MAC_Layer3_Addr1_Reg(#i) (for i = 0; i <= 3)

Description: For IPv4 packets, the Layer 3 Address 1 Register 0 register contains the 32-bit IP Destination Address field. For IPv6 packets, it contains Bits[63:32] of the 128-bit IP Source Address or Destination Address field.

Size: 32 bits

Offset: (0x0030*i)+0x0914



Fields for Register: MAC_Layer3_Addr1_Reg(#i) (for i = 0; i <= 3)

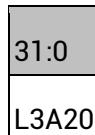
Bits	Name	Memory Access	Description
31:0	L3A10	R/W	<p>Layer 3 Address 1 Field</p> <p>When the L3PEN0 and L3SAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits[63:32] of the IP Source Address field in the IPv6 packets.</p> <p>When the L3PEN0 and L3DAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits[63:32] of the IP Destination Address field in the IPv6 packets.</p> <p>When the L3PEN0 bit is reset and the L3SAM0 bit is set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with the IP Destination Address field in the IPv4 packets.</p> <p>Value After Reset: 0x0</p>

MAC_Layer3_Addr2_Reg(#i) (for i = 0; i <= 3)

Description: The Layer 3 Address 2 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[95:64] of 128-bit IP Source Address or Destination Address field.

Size: 32 bits

Offset: (0x0030*i)+0x0918

**Fields for Register: MAC_Layer3_Addr2_Reg(#i) (for i = 0; i <= 3)**

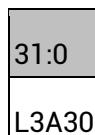
Bits	Name	Memory Access	Description
31:0	L3A20	R/W	<p>Layer 3 Address 2 Field</p> <p>When the L3PENO and L3SAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits[95:64] of the IP Source Address field in the IPv6 packets.</p> <p>When the L3PENO and L3DAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits[95:64] of the IP Destination Address field in the IPv6 packets.</p> <p>When the L3PENO bit is reset in the MAC_L3_L4_Control0 register, this field is not used.</p> <p>Value After Reset: 0x0</p>

MAC_Layer3_Addr3_Reg(#i) (for i = 0; i <= 3)

Description: The Layer 3 Address 3 Register 0 register is reserved for IPv4 packets. For IPv6 packets, it contains Bits[127:96] of 128-bit IP Source Address or Destination Address field.

Size: 32 bits

Offset: (0x0030*i)+0x091C

**Fields for Register: MAC_Layer3_Addr3_Reg(#i) (for i = 0; i <= 3)**

Bits	Name	Memory Access	Description
31:0	L3A30	R/W	<p>Layer 3 Address 3 Field</p> <p>When the L3PENO and L3SAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits[127:96] of the IP Source Address field in the IPv6 packets.</p> <p>When the L3PENO and L3DAM0 bits are set in the MAC_L3_L4_Control0 register, this field contains the value to be matched with Bits[127:96] of the IP Destination Address field in the IPv6 packets.</p>

Bits	Name	Memory Access	Description							
			When the L3PEN0 bit is reset in the MAC_L3_L4_Control0 register, this field is not used. Value After Reset: 0x0							

MAC_Indir_Access_Ctrl

Description: This register provides the Indirect Access control and status for MAC_<MSEL>_<AOFF> registers.

Size: 32 bits

Offset: 0xa70

31:20	19:16	15:8	7:6	5	4:2	1	0
Reserved_31_20	MSEL	AOFF	Reserved_7_6	AUTO	Reserved_4_2	COM	OB

Fields for Register: MAC_Indir_Access_Ctrl

Bits	Name	Memory Access	Description							
31:20	Reserved_31_20	R	Reserved. Value After Reset: 0x0							
19:16	MSEL	R/W	Mode Select This field is used in indirect access of MAC_<MSEL>_<AOFF>. This field must be set along with initiation of read/write to MAC_<MSEL>_<AOFF> and should not be changed until the OB is reset. Values: 0000:Typ_RXQ_ (Type based RXQ mapping) 0001-1111: Reserved for future use. Value After Reset: 0x0							
15:8	AOFF	R/W	Address Offset This field is used in indirect access of MAC_<MSEL>_<AOFF>. This field must be set along with initiation of read/write to MAC_<MSEL>_<AOFF> and should not be changed until the OB is reset. Values: 00000000: IndReg0(Indirect register 0) 00000001: IndReg1(Indirect register 1) ...							

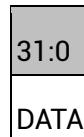
Bits	Name	Memory Access	Description
			<p>00000111: IndReg7(Indirect register 7)</p> <p>00001000-11111111: Reserved for future use.</p> <p>Value After Reset: 0x0</p>
7:6	Reserved_7_6	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
5	AUTO	R/W	<p>Auto increment</p> <p>1: AOFF is incremented by 1. Software should ensure not to cause a wrap condition. Byte wise read/write is not supported when auto increment is enabled.</p> <p>0: AOFF is not incremented automatically. Software should program the correct Address Offset for each access.</p> <p>Value After Reset: 0x0</p>
4:2	Reserved_4_2	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
1	COM	R/W	<p>Command type</p> <p>This bit indicates the register access type.</p> <p>1: Indicates a read operation.</p> <p>0: Indicates a write operation.</p> <p>Values:</p> <p>0x0 (WRITE): Write operation</p> <p>0x1 (READ): Read operation</p> <p>Value After Reset: 0x0</p>
0	OB	R/W	<p>Operation Busy.</p> <p>This bit is set along with a read or write command for initiating the indirect access to MAC_<MSEL>_<AOFF> register. This bit is reset when the read or write command to MAC_<MSEL>_<AOFF> register is complete. The next indirect register access can be initiated only after this bit is reset. During a write operation, the bit is reset only after the data has been written into MAC_<MSEL>_<AOFF> register. During a read operation, the data should be read from MAC_Indir_Access_Data register only after this bit is reset.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MAC_Indir_Access_Data

Description: This register holds the read/write data for Indirect Access of MAC_<MSEL_<AOFF> registers.

Size: 32 bits

Offset: 0xa74

**Fields for Register: MAC_Indir_Access_Data**

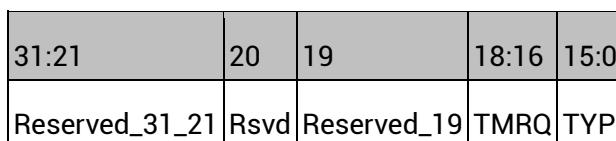
Bits	Name	Memory Access	Description
31:0	DATA	R/W	<p>This field contains data to read/write for Indirect address access associated with MAC_Indir_Access_Ctrl register.</p> <p>Value After Reset: 0x0</p>

MAC_TMRQ_Regs(#i) (for i = 0; i <= 7)

Description: This register contains the type, associated queue number and paket type (Preemption/EXpress) related to Type based RXQ mapping.

Size: 32 bits

Offset: 0xa74

**Fields for Register: MAC_TMRQ_Regs(#i) (for i = 0; i <= 7)**

Bits	Name	Memory Access	Description
31:21	Reserved_31_21	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
20			Reserved Field: Yes
19	Reserved_19	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
18:16	TMRQ	R/W	<p>Type Match Rx Queue Number</p> <p>Indicates the receive queue number to which the packet needs to be forwarded on detecting a type match.</p> <p>Value After Reset: 0x0</p>
15:0	TYP	R/W	Type field Value Indicates the type value of packet that needs to be

Bits	Name	Memory Access	Description
			<p>compared with received Ethernet packet. This field is valid when programmed to a value greater than or equal to 0x0600.</p> <p>Value After Reset: 0x0</p>

MTL_Operation_Mode

Description: The Operation Mode register establishes the Transmit and Receive operating modes and commands.

Size: 32 bits

Offset: 0xc00

31:16	15:1 4	13:10	9	8	7	6:5	4:3	2	1	0
Reserved_31_16	Rsvd	Reserved_13_10	CNTCLR	CNTPRST	Reserved_7	SCHA LG	Reserved_4_3	RA A	DTXS TS	Reserved_0

Fields for Register: MTL_Operation_Mode

Bits	Name	Memory Access	Description
31:16	Reserved_31_16	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
15:14			Reserved Field: Yes
13:10	Reserved_13_10	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
9	CNTCLR	R/W	<p>Counters Reset</p> <p>When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle.</p> <p>If this bit is set along with CNT_PRESET bit, CNT_PRESET has precedence.</p> <p>Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Counters are not reset 0x1 (ENABLE): All counters are reset <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
8	CNTPRST	R/W	Counters Preset

Bits	Name	Memory Access	Description
			<p>When this bit is set, MTL_TxQ[0-7]_Underflow register is initialized/preset to 12'h7F0.</p> <p>Missed Packet and Overflow Packet counters in MTL_RxQ[0-7]_Missed_Packet_Overflow_Cnt register is initialized/preset to 12'h7F0.</p> <p>Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Counters Preset is disabled 0x1 (ENABLE): Counters Preset is enabled <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
7	Reserved_7	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
6:5	SCHALG	R/W	<p>Tx Scheduling Algorithm</p> <p>This field indicates the algorithm for Tx scheduling:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (WRR): WRR algorithm 0x1 (WFQ): WFQ algorithm when DCB feature is selected.Otherwise, Reserved 0x2 (DWRR): DWRR algorithm when DCB feature is selected.Otherwise, Reserved 0x3 (SP): Strict priority algorithm <p>Value After Reset: 0x0</p>
4:3	Reserved_4_3	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
2	RAA	R/W	<p>Receive Arbitration Algorithm</p> <p>This field is used to select the arbitration algorithm for the Rx side.</p> <ul style="list-style-type: none"> 0: Strict priority (SP). Queue 0 has the lowest priority and the last queue has the highest priority. 1: Weighted Strict Priority (WSP) <p>Values:</p> <ul style="list-style-type: none"> 0x0 (SP): Strict priority (SP)

Bits	Name	Memory Access	Description
			<p>0x1 (WSP): Weighted Strict Priority (WSP)</p> <p>Value After Reset: 0x0</p>
1	DTXSTS	R/W	<p>Drop Transmit Status</p> <p>1: Tx packet status received from the MAC is dropped in the MTL</p> <p>0: Tx packet status received from the MAC is forwarded to the application</p> <p>Values:</p> <p>0x0 (DISABLE): Drop Transmit Status is disabled</p> <p>0x1 (ENABLE): Drop Transmit Status is enabled</p> <p>Value After Reset: 0x0</p>
0	Reserved_0	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>

MTL_Interrupt_Status

Description: The software driver (application) reads this register during interrupt service routine or polling to determine the interrupt status of MTL queues and the MAC.

Size: 32 bits

Offset: 0xc20

31:24	23	22:19	18:16	15:8	7:4	3	2	1	0
Reserved_31_24	Rsvd	Reserved_22_19	Rsvd	Reserved_15_8	Rsvd	Q3IS	Q2IS	Q1IS	Q0IS

Fields for Register: MTL_Interrupt_Status

Bits	Name	Memory Access	Description
31:24	Reserved_31_24	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
23			Reserved Field: Yes
22:19	Reserved_22_19	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
18:16			Reserved Field: Yes
15:8	Reserved_15_8	R	Reserved.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
7:4			Reserved Field: Yes
3	Q3IS	R	<p>Queue 3 Interrupt status</p> <p>This bit indicates that there is an interrupt from Queue 3. To reset this bit, the application must read the MTL_Q3_Interrupt_Control_Status register to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Queue 3 Interrupt status not detected 0x1 (ACTIVE): Queue 3 Interrupt status detected <p>Value After Reset: 0x0</p>
2	Q2IS	R	<p>Queue 2 Interrupt status</p> <p>This bit indicates that there is an interrupt from Queue 2. To reset this bit, the application must read the MTL_Q2_Interrupt_Control_Status register to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Queue 2 Interrupt status not detected 0x1 (ACTIVE): Queue 2 Interrupt status detected <p>Value After Reset: 0x0</p>
1	Q1IS	R	<p>Queue 1 Interrupt status</p> <p>This bit indicates that there is an interrupt from Queue 1. To reset this bit, the application must read the MTL_Q1_Interrupt_Control_Status register to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Queue 1 Interrupt status not detected 0x1 (ACTIVE): Queue 1 Interrupt status detected <p>Value After Reset: 0x0</p>
0	Q0IS	R	<p>Queue 0 Interrupt status</p> <p>This bit indicates that there is an interrupt from Queue 0. To reset this bit, the application must read Queue 0 Interrupt Control and Status register to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Queue 0 Interrupt status not detected

Bits	Name	Memory Access	Description
			<p>0x1 (ACTIVE): Queue 0 Interrupt status detected Value After Reset: 0x0</p>

MTL_RxQ_DMA_Map0

Description: The Receive Queue and DMA Channel Mapping 0 register is reserved in EQOS-CORE and EQOS-MTL configurations.

Size: 32 bits

Offset: 0xc30

31:29	28	27:26	n:24	23:21	20	19:18	17:16	15:13	12	11:10	9:8	7:5	4	3:2	1:0
Reserved_31_29	Q3DDMACH	Reserved_27_y	Q3MDMACH	Reserved_23_21	Q2DDMACH	Reserved_19_y	Q2MDMACH	Reserved_15_13	Q1DDMACH	Reserved_11_y	Q1MDMACH	Reserved_7_5	Q0DDMACH	Reserved_3_y	Q0MDMACH

Fields for Register: MTL_RxQ_DMA_Map0

Bits	Name	Memory Access	Description
31:29	Reserved_31_29	R	<p>Reserved. Value After Reset: 0x0</p>
28	Q3DDMACH	R/W	<p>Queue 3 Enabled for Dynamic (per packet) DMA Channel Selection When set, this bit indicates that the packets received in Queue 3 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address. When reset, this bit indicates that the packets received in Queue 3 are routed to the DMA Channel programmed in the Q3MDMACH field (Bits[26:24]). Values: 0x0 (DISABLE): Queue 3 disabled for DA-based DMA Channel Selection 0x1 (ENABLE): Queue 3 enabled for DA-based DMA Channel Selection Value After Reset: 0x0</p>
27:26	Reserved_27_y	R	<p>Reserved. Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
25:24	Q3MDMACH	R/W	<p>Queue 3 Mapped to DMA Channel</p> <p>This field controls the routing of the received packet in Queue 3 to the DMA channel:</p> <ul style="list-style-type: none"> 000: DMA Channel 0 001: DMA Channel 1 010: DMA Channel 2 011: DMA Channel 3 100: DMA Channel 4 101: DMA Channel 5 110: DMA Channel 6 111: DMA Channel 7 <p>This field is valid when the Q3DDMACH field is reset.</p> <p>Note: The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the others are reserved</p> <p>Value After Reset: 0x0</p>
23:21	Reserved_23_21	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
20	Q2DDMACH	R/W	<p>Queue 2 Enabled for DA-based DMA Channel Selection</p> <p>When set, this bit indicates that the packets received in Queue 2 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address.</p> <p>When reset, this bit indicates that the packets received in Queue 2 are routed to the DMA Channel programmed in the Q2MDMACH field (Bits[18:16]).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Queue 2 disabled for DA-based DMA Channel Selection 0x1 (ENABLE): Queue 2 enabled for DA-based DMA Channel Selection <p>Value After Reset: 0x0</p>
19:18	Reserved_19_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
17:16	Q2MDMACH	R/W	Queue 2 Mapped to DMA Channel

Bits	Name	Memory Access	Description
			<p>This field controls the routing of the received packet in Queue 2 to the DMA channel:</p> <ul style="list-style-type: none"> 000: DMA Channel 0 001: DMA Channel 1 010: DMA Channel 2 011: DMA Channel 3 100: DMA Channel 4 101: DMA Channel 5 110: DMA Channel 6 111: DMA Channel 7 <p>This field is valid when the Q2DDMACH field is reset.</p> <p>Value After Reset: 0x0</p>
15:13	Reserved_15_13	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
12	Q1DDMACH	R/W	<p>Queue 1 Enabled for DA-based DMA Channel Selection</p> <p>When set, this bit indicates that the packets received in Queue 1 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address.</p> <p>When reset, this bit indicates that the packets received in Queue 1 are routed to the DMA Channel programmed in the Q1MDMACH field (Bits[10:8]).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Queue 1 disabled for DA-based DMA Channel Selection 0x1 (ENABLE): Queue 1 enabled for DA-based DMA Channel Selection <p>Value After Reset: 0x0</p>
11:10	Reserved_11_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
9:8	Q1MDMACH	R/W	<p>Queue 1 Mapped to DMA Channel</p> <p>This field controls the routing of the received packet in Queue 1 to the DMA channel:</p> <ul style="list-style-type: none"> 000: DMA Channel 0 001: DMA Channel 1

Bits	Name	Memory Access	Description
			<p>010: DMA Channel 2 011: DMA Channel 3 100: DMA Channel 4 101: DMA Channel 5 110: DMA Channel 6 111: DMA Channel 7</p> <p>This field is valid when the Q1DDMACH field is reset.</p> <p>The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.</p> <p>Value After Reset: 0x0</p>
7:5	Reserved_7_5	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
4	Q0DDMACH	R/W	<p>Queue 0 Enabled for DA-based DMA Channel Selection</p> <p>When set, this bit indicates that the packets received in Queue 0 are routed to a particular DMA channel as decided in the MAC Receiver based on the DMA channel number programmed in the L3-L4 filter registers, or the Ethernet DA address.</p> <p>When reset, this bit indicates that the packets received in Queue 0 are routed to the DMA Channel programmed in the Q0MDMACH field.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Queue 0 disabled for DA-based DMA Channel Selection 0x1 (ENABLE): Queue 0 enabled for DA-based DMA Channel Selection <p>Value After Reset: 0x0</p>
3:2	Reserved_3_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
1:0	Q0MDMACH	R/W	<p>Queue 0 Mapped to DMA Channel</p> <p>This field controls the routing of the packet received in Queue 0 to the DMA channel:</p> <ul style="list-style-type: none"> 000: DMA Channel 0 001: DMA Channel 1 010: DMA Channel 2

Bits	Name	Memory Access	Description
			<p>011: DMA Channel 3 100: DMA Channel 4 101: DMA Channel 5 110: DMA Channel 6 111: DMA Channel 7</p> <p>This field is valid when the Q0DDMACH field is reset.</p> <p>The width of this field depends on the number of RX DMA channels and not all the values might be valid in some configurations. For example, if the number of RX DMA channels selected is 2, only 000 and 001 are valid, the other bits are reserved.</p> <p>Value After Reset: 0x0</p>

MTL_TxQ0_Operation_Mode

Description: The Queue 0 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

Size: 32 bits

Offset: 0xd00

31:y	19:16	15:7	6:4	3:2	1	0
Reserved_31_y	TQS	Reserved_15_7	TTC	TXQEN	TSF	FTQ

Fields for Register: MTL_TxQ0_Operation_Mode

Bits	Name	Memory Access	Description
31:20	Reserved_31_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
19:16	TQS	R/W	<p>Transmit Queue Size</p> <p>This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. This means that value of 0x0 = 256 bytes, 0x1 = 512 bytes and so on. So, program TQS [5:0] = 6'b001111 to allocate queue size of 4096 (4K) bytes. In general, the size of the Queue = (TQS+1)*256 bytes.</p> <p>When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value.</p> <p>The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of</p>

Bits	Name	Memory Access	Description
			<p>this field is 3 bits: $\text{LOG2}(2048/256) = \text{LOG2}(8) = 3 \text{ bits}$</p> <p>Value After Reset: 0x0</p>
15:7	Reserved_15_7	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
6:4	TTC	R/W	<p>Transmit Threshold Control</p> <p>These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_32BYTES): 32 0x1 (M_64BYTES): 64 0x2 (M_96BYTES): 96 0x3 (M_128BYTES): 128 0x4 (M_192BYTES): 192 0x5 (M_256BYTES): 256 0x6 (M_384BYTES): 384 0x7 (M_512BYTES): 512 <p>Value After Reset: 0x0</p>
3:2	TXQEN	R/W	<p>Transmit Queue Enable</p> <p>This field is used to enable/disable the transmit queue 0.</p> <ul style="list-style-type: none"> 2'b00: Not enabled 2'b01: Reserved 2'b10: Enabled 2'b11: Reserved <p>This field is Read Only in Single Queue configurations and Read Write in Multiple Queue configurations.</p> <p>Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Not enabled 0x1 (EN_IF_AV): Enable in AV mode (Reserved in non-AV) 0x2 (ENABLE): Enabled

Bits	Name	Memory Access	Description
			<p>0x3 (RSVD2): Reserved</p> <p>Value After Reset: 0x0</p>
1	TSF	R/W	<p>Transmit Store and Forward</p> <p>When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Transmit Store and Forward is disabled 0x1 (ENABLE): Transmit Store and Forward is enabled <p>Value After Reset: 0x0</p>
0	FTQ	R/W	<p>Flush Transmit Queue</p> <p>When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TxQ1_Operation_Mode register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission.</p> <p>Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active.</p> <p>Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Flush Transmit Queue is disabled 0x1 (ENABLE): Flush Transmit Queue is enabled <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MTL_TxQ0_Underflow

Description: The Queue 0 Underflow Counter register contains the counter for packets aborted because of Transmit queue underflow and packets missed because of Receive queue packet flush

Size: 32 bits

Offset: 0xd04

31:12	11	10:0
Reserved_31_12	UFCNTOVF	UFFRMCNT

Fields for Register: MTL_TxQ0_Underflow

Bits	Name	Memory Access	Description
31:12	Reserved_31_12	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
11	UFCNTOVF	R	<p>Overflow Bit for Underflow Packet Counter</p> <p>This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Overflow not detected for Underflow Packet Counter 0x1 (ACTIVE): Overflow detected for Underflow Packet Counter <p>Value After Reset: 0x0</p>
10:0	UFFRMCNT	R	<p>Underflow Packet Counter</p> <p>This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Value After Reset: 0x0</p>

MTL_TxQ0_Debug

Description: The Queue 0 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

Size: 32 bits

Offset: 0xd08

31:23	22:20	19	18:1 6	15:6	5	4	3	2:1	0
Reserved_31_23	STXST SF	Reserved_19	PTX Q	Reserved_15_6	TXSTSFS TS	TXQSTS	TWCSTS	TRCSTS	TXQPAUSED

Fields for Register: MTL_TxQ0_Debug

Bits	Name	Memory Access	Description
31:23	Reserved_31_23	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
22:20	STXSTSF	R	<p>Number of Status Words in Tx Status FIFO of Queue</p> <p>This field indicates the current number of status in the Tx Status FIFO of this queue.</p> <p>When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.</p> <p>Value After Reset: 0x0</p>
19	Reserved_19	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
18:16	PTXQ	R	<p>Number of Packets in the Transmit Queue</p> <p>This field indicates the current number of packets in the Tx Queue.</p> <p>When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of packets in the Transmit queue.</p> <p>Value After Reset: 0x0</p>
15:6	Reserved_15_6	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
5	TXSTSFSTS	R	<p>MTL Tx Status FIFO Full Status</p> <p>When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MTL Tx Status FIFO Full status is not detected 0x1 (ACTIVE): MTL Tx Status FIFO Full status is detected <p>Value After Reset: 0x0</p>
4	TXQSTS	R	<p>MTL Tx Queue Not Empty Status</p> <p>When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MTL Tx Queue Not Empty status is not detected 0x1 (ACTIVE): MTL Tx Queue Not Empty status is detected

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
3	TWCSTS	R	<p>MTL Tx Queue Write Controller Status</p> <p>When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MTL Tx Queue Write Controller status is not detected 0x1 (ACTIVE): MTL Tx Queue Write Controller status is detected <p>Value After Reset: 0x0</p>
2:1	TRCSTS	R	<p>MTL Tx Queue Read Controller Status</p> <p>This field indicates the state of the Tx Queue Read Controller:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (IDLE): Idle state 0x1 (READ): Read state (transferring data to the MAC transmitter) 0x2 (WAIT): Waiting for pending Tx Status from the MAC transmitter 0x3 (FLUSH): Flushing the Tx queue because of the Packet Abort request from the MAC <p>Value After Reset: 0x0</p>
0	TXQPAUSED	R	<p>Transmit Queue in Pause</p> <p>When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following:</p> <ul style="list-style-type: none"> Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled Reception of 802.3x Pause packet when PFC is disabled <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Transmit Queue in Pause status is not detected 0x1 (ACTIVE): Transmit Queue in Pause status is detected <p>Value After Reset: 0x0</p>

MTL_TxQ0_ETS_Status

Description: The Queue 0 ETS Status register provides the average traffic transmitted in Queue 0.

Size: 32 bits**Offset:** 0xd14

31:24	23:0
Reserved_31_24	ABS

Fields for Register: MTL_TxQ0_ETS_Status

Bits	Name	Memory Access	Description
31:24	Reserved_31_24	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
23:0	ABS	R	<p>Average Bits per Slot</p> <p>This field contains the average transmitted bits per slot.</p> <p>When the DCB operation is enabled for Queue 0, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.</p> <p>Value After Reset: 0x0</p>

MTL_TxQ0_Quantum_Weight

Description: The Queue 0 Quantum or Weights register contains the quantum value for Deficit Weighted Round Robin (DWRR), weights for the Weighted Round Robin (WRR), and Weighted Fair Queuing (WFQ) for Queue 0.

Size: 32 bits**Offset:** 0xd18

31:21	20:0
Reserved_31_21	ISCQW

Fields for Register: MTL_TxQ0_Quantum_Weight

Bits	Name	Memory Access	Description
31:21	Reserved_31_21	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
20:0	ISCQW	R/W	<p>Quantum or Weights</p> <p>When the DCB operation is enabled with DWRR algorithm for Queue 0 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes.</p> <p>When DCB operation is enabled with WFQ algorithm for Queue 0</p>

Bits	Name	Memory Access	Description
			<p>traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. The higher the programmed weights lesser the bandwidth allocated for the particular Transmit Queue. This is because the weights are used to compute the packet finish time (weights * packet_size). Lesser the finish time, higher the probability of the packet getting scheduled first and using more bandwidth.</p> <p>When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 0 traffic, this field contains the weight for this queue. The maximum value is 0x64.</p> <p>Bits [20:7] must be written to zero.</p> <p>Value After Reset: 0x0</p>

MTL_Q0_Interrupt_Control_Status

Description: This register contains the interrupt enable and status bits for the queue 0 interrupts.

Size: 32 bits

Offset: 0xd2c

31:25	24	23:17	16	15:10	9	8	7:2	1	0
Reserved_31_25	RXOIE	Reserved_23_17	RXOVFIS	Reserved_15_10	ABPSI_E	TXUI_E	Reserved_7_2	ABPSI_S	TXUNFIS

Fields for Register: MTL_Q0_Interrupt_Control_Status

Bits	Name	Memory Access	Description
31:25	Reserved_31_25	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
24	RXOIE	R/W	<p>Receive Queue Overflow Interrupt Enable</p> <p>When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Queue Overflow Interrupt is disabled 0x1 (ENABLE): Receive Queue Overflow Interrupt is enabled <p>Value After Reset: 0x0</p>
23:17	Reserved_23_17	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
16	RXOVFIS	R/W	<p>Receive Queue Overflow Interrupt Status</p> <p>This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Receive Queue Overflow Interrupt Status not detected 0x1 (ACTIVE): Receive Queue Overflow Interrupt Status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
15:10	Reserved_15_10	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
9	ABPSIE	R/W	<p>Average Bits Per Slot Interrupt Enable</p> <p>When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated.</p> <p>When this bit is cleared, the interrupt is not asserted for such an event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Average Bits Per Slot Interrupt is disabled 0x1 (ENABLE): Average Bits Per Slot Interrupt is enabled <p>Value After Reset: 0x0</p>
8	TXUIE	R/W	<p>Transmit Queue Underflow Interrupt Enable</p> <p>When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Transmit Queue Underflow Interrupt Status is disabled 0x1 (ENABLE): Transmit Queue Underflow Interrupt Status is enabled <p>Value After Reset: 0x0</p>
7:2	Reserved_7_2	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
1	ABPSIS	R/W	<p>Average Bits Per Slot Interrupt Status</p> <p>When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Average Bits Per Slot Interrupt Status not detected 0x1 (ACTIVE): Average Bits Per Slot Interrupt Status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
0	TXUNFIS	R/W	<p>Transmit Queue Underflow Interrupt Status</p> <p>This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Transmit Queue Underflow Interrupt Status not detected 0x1 (ACTIVE): Transmit Queue Underflow Interrupt Status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MTL_RxQ0_Operation_Mode

Description: The Queue 0 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

Size: 32 bits

Offset: 0xd30

31:y	23:2 0	19:17	16:1 4	13:11	10: 8	7	6	5	4	3	2	1:0
Reserved_31 -y	RQS	Reserved_19 -y	RFD	Reserved_13 -y	RF A	EHF C	DIS_TCP_EF	RS F	FE P	FU P	Reserved _2	RT C

Fields for Register: MTL_RxQ0_Operation_Mode

Bits	Name	Memory Access	Description
31:24	Reserved_31_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
23:20	RQS	R/W	<p>Receive Queue Size</p> <p>This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. This means that value of 0x0 = 256 bytes, 0x1 = 512 bytes and so on. So, program RQS [5:0] = 6'b001111 to allocate queue size of 4096 (4K) bytes. In general, the size of the Queue = (RQS+1)*256 bytes.</p> <p>When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value.</p> <p>The width of this field depends on the Rx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits:</p> $\text{LOG2}(2048/256) = \text{LOG2}(8) = 3 \text{ bits}$ <p>Value After Reset: 0x0</p>
19:17	Reserved_19_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
16:14	RFD	R/W	<p>Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes)</p> <p>These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation:</p> <ul style="list-style-type: none"> 0: Full minus 1 KB, that is, FULL 1 KB 1: Full minus 1.5 KB, that is, FULL 1.5 KB 2: Full minus 2 KB, that is, FULL 2 KB 3: Full minus 2.5 KB, that is, FULL 2.5 KB ... 6: Full minus 4 KB, that is, FULL 4 KB 7: Full minus 4.5 KB, that is, FULL 4.5 KB <p>The de-assertion is effective only after flow control is asserted.</p> <p>Note: The value must be programmed in such a way to make sure that the threshold is a positive number.</p> <p>When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB.</p>

Bits	Name	Memory Access	Description
			<p>For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register.</p> <p>The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.</p> <p>Value After Reset: 0x0</p>
13:11	Reserved_13_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
10:8	RFA	R/W	<p>Threshold for Activating Flow Control (in half-duplex and full-duplex)</p> <p>These bits control the threshold (fill-level of Rx queue) at which the flow control is activated:</p> <p>For more information on encoding for this field, see RFD.</p> <p>Value After Reset: 0x0</p>
7	EHFC	R/W	<p>Enable Hardware Flow Control</p> <p>When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Hardware Flow Control is disabled 0x1 (ENABLE): Hardware Flow Control is enabled <p>Value After Reset: 0x0</p>
6	DIS_TCP_EF	R/W	<p>Disable Dropping of TCP/IP Checksum Error Packets</p> <p>When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC.</p> <p>When this bit is reset, all error packets are dropped if the FEP bit is reset.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): Dropping of TCP/IP Checksum Error Packets is enabled 0x1 (DISABLE): Dropping of TCP/IP Checksum Error Packets is disabled <p>Value After Reset: 0x0</p>
5	RSF	R/W	Receive Queue Store and Forward

Bits	Name	Memory Access	Description
			<p>When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Queue Store and Forward is disabled 0x1 (ENABLE): Receive Queue Store and Forward is enabled <p>Value After Reset: 0x0</p>
4	FEP	R/W	<p>Forward Error Packets</p> <p>When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped.</p> <p>When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet might be forwarded to the application or DMA.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Forward Error Packets is disabled 0x1 (ENABLE): Forward Error Packets is enabled <p>Value After Reset: 0x0</p>
3	FUP	R/W	<p>Forward Undersized Good Packets</p> <p>When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Forward Undersized Good Packets is disabled 0x1 (ENABLE): Forward Undersized Good Packets is enabled <p>Value After Reset: 0x0</p>
2	Reserved_2	R	Reserved.
1:0	RTC	R/W	Receive Queue Threshold Control

Bits	Name	Memory Access	Description
			<p>These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred.</p> <p>This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_64BYTE): 64 0x1 (M_32BYTE): 32 0x2 (M_96BYTE): 96 0x3 (M_128BYTE): 128 <p>Value After Reset: 0x0</p>

MTL_RxQ0_Missed_Packet_Overflow_Cnt

Description: The Queue 0 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

Size: 32 bits

Offset: 0xd34

31:28	27	26:16	15:12	11	10:0
Reserved_31_28	MISCNTOVF	MISPKTCNT	Reserved_15_12	OVFCNTOVF	OVFPKTCNT

Fields for Register: MTL_RxQ0_Missed_Packet_Overflow_Cnt

Bits	Name	Memory Access	Description
31:28	Reserved_31_28	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
27	MISCNTOVF	R	<p>Missed Packet Counter Overflow Bit</p> <p>When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Missed Packet Counter overflow not detected 0x1 (ACTIVE): Missed Packet Counter overflow detected

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
26:16	MISPKTCNT	R	<p>Missed Packet Counter</p> <p>This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1'b1.</p> <p>This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Value After Reset: 0x0</p>
15:12	Reserved_15_12	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
11	OVFCNTOVF	R	<p>Overflow Counter Overflow Bit</p> <p>When set, this bit indicates that the Rx Queue Overflow Packet Counter field crossed the maximum limit.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Overflow Counter overflow not detected 0x1 (ACTIVE): Overflow Counter overflow detected <p>Value After Reset: 0x0</p>
10:0	OVFPKTCNT	R	<p>Overflow Packet Counter</p> <p>This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Value After Reset: 0x0</p>

MTL_RxQ0_Debug

Description: The Queue 0 Receive Debug register gives the debug status of various blocks related to the Receive queue.

Size: 32 bits

Offset: 0xd38

31:30	29:16	15:6	5:4	3	2:1	0
Reserved_31_30	PRXQ	Reserved_15_6	RXQSTS	Reserved_3	RRCTS	RWCSTS

Fields for Register: MTL_RxQ0_Debug

Bits	Name	Memory Access	Description
31:30	Reserved_31_30	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
29:16	PRXQ	R	<p>Number of Packets in Receive Queue</p> <p>This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.</p> <p>Value After Reset: 0x0</p>
15:6	Reserved_15_6	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
5:4	RXQSTS	R	<p>MTL Rx Queue Fill-Level Status</p> <p>This field gives the status of the fill-level of the Rx Queue:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (EMPTY): Rx Queue empty 0x1 (BLWTHR): Rx Queue fill-level below flow-control deactivate threshold 0x2 (ABVTHR): Rx Queue fill-level above flow-control activate threshold 0x3 (FULL): Rx Queue full <p>Value After Reset: 0x0</p>
3	Reserved_3	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
2:1	RRCTS	R	<p>MTL Rx Queue Read Controller State</p> <p>This field gives the state of the Rx queue Read controller:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (IDLE): Idle state 0x1 (READ_DATA): Reading packet data 0x2 (READ_STS): Reading packet status (or timestamp) 0x3 (FLUSH): Flushing the packet data and status <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
0	RWCSTS	R	<p>MTL Rx Queue Write Controller Active Status</p> <p>When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MTL Rx Queue Write Controller Active Status not detected 0x1 (ACTIVE): MTL Rx Queue Write Controller Active Status detected <p>Value After Reset: 0x0</p>

MTL_RxQ0_Control

Description: The Queue Receive Control register controls the receive arbitration and passing of received packets to the application.

Size: 32 bits

Offset: 0xd3c

31:4	3	2:0
Reserved_31_4	RXQ_FRM_ARBIT	RXQ_WEGT

Fields for Register: MTL_RxQ0_Control

Bits	Name	Memory Access	Description
31:4	Reserved_31_4	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
3	RXQ_FRM_ARBIT	R/W	<p>Receive Queue Packet Arbitration</p> <p>When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue.</p> <p>When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue:</p> <ul style="list-style-type: none"> PBL amount of data (indicated by ari_qN_pbl_i[]) or Complete data of a packet <p>The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-</p>

Bits	Name	Memory Access	Description
			<p>and-forward mode) or the last PBL request for the packet (in Threshold mode).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Queue Packet Arbitration is disabled 0x1 (ENABLE): Receive Queue Packet Arbitration is enabled <p>Value After Reset: 0x0</p>
2:0	RXQ_WEGT	R/W	<p>Receive Queue Weight</p> <p>This field indicates the weight assigned to the Rx Queue 0. This field needs to be programmed with one value less than the required weight, that is, reset value of 0 indicates weight of 1, value of 1 indicates weight of 2, and so on. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.</p> <p>Note: The change in value of RXQ_WEGT takes effect only after the completion of current service round or when there is change from RAA=SP to RAA=WSP algorithm. This approach is taken so that there is smooth transition. For the RXQ_WEGT value to take effect at the start, the MTL_RxQ(#i)_Control registers must be programmed before the MTL_Operation_Mode register.</p> <p>Value After Reset: 0x0</p>

MTL_TxQ(#i)_Operation_Mode (for i = 1; i <= 3)

Description: The Queue 1 Transmit Operation Mode register establishes the Transmit queue operating modes and commands.

Size: 32 bits

Offset: (0x0040*i)+0x0D00

31:y	19:16	15:7	6:4	3:2	1	0
Reserved_31_y	TQS	Reserved_15_7	TTC	TXQEN	TSF	FTQ

Fields for Register: MTL_TxQ(#i)_Operation_Mode (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:20	Reserved_31_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
19:16	TQS	R/W	<p>Transmit Queue Size</p> <p>This field indicates the size of the allocated Transmit queues in blocks of 256 bytes. The TQS field is read-write only if the number of Tx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. This means that value of 0x0 = 256 bytes, 0x1 = 512 bytes and</p>

Bits	Name	Memory Access	Description
			<p>so on. So, program TQS [5:0] = 6'b001111 to allocate queue size of 4096 (4K) bytes. In general, the size of the Queue = (TQS+1)*256 bytes.</p> <p>When the number of Tx Queues is one, the field is read-only and the configured TX FIFO size in blocks of 256 bytes is reflected in the reset value.</p> <p>The width of this field depends on the Tx memory size selected in your configuration. For example, if the memory size is 2048, the width of this field is 3 bits:</p> $\text{LOG2}(2048/256) = \text{LOG2}(8) = 3 \text{ bits}$ <p>Value After Reset: 0x0</p>
15:7	Reserved_15_7	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
6:4	TTC	R/W	<p>Transmit Threshold Control</p> <p>These bits control the threshold level of the MTL Tx Queue. The transmission starts when the packet size within the MTL Tx Queue is larger than the threshold. In addition, full packets with length less than the threshold are also transmitted. These bits are used only when the TSF bit is reset.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_32BYTES): 32 0x1 (M_64BYTES): 64 0x2 (M_96BYTES): 96 0x3 (M_128BYTES): 128 0x4 (M_192BYTES): 192 0x5 (M_256BYTES): 256 0x6 (M_384BYTES): 384 0x7 (M_512BYTES): 512 <p>Value After Reset: 0x0</p>
3:2	TXQEN	R/W	<p>Transmit Queue Enable</p> <p>This field is used to enable/disable the transmit queue 0.</p> <ul style="list-style-type: none"> 2'b00: Not enabled 2'b01: Reserved 2'b10: Enabled 2'b11: Reserved <p>Note: In multiple Tx queues configuration, all the queues are disabled by default. Enable the Tx queue by programming this field.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Not enabled 0x1 (EN_IF_AV): Enable in AV mode (Reserved in non-AV) 0x2 (ENABLE): Enabled 0x3 (RSVD2): Reserved <p>Value After Reset: 0x0</p>
1	TSF	R/W	<p>Transmit Store and Forward</p> <p>When this bit is set, the transmission starts when a full packet resides in the MTL Tx queue. When this bit is set, the TTC values specified in Bits[6:4] of this register are ignored. This bit should be changed only when the transmission is stopped.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Transmit Store and Forward is disabled 0x1 (ENABLE): Transmit Store and Forward is enabled <p>Value After Reset: 0x0</p>
0	FTQ	R/W	<p>Flush Transmit Queue</p> <p>When this bit is set, the Tx queue controller logic is reset to its default values. Therefore, all the data in the Tx queue is lost or flushed. This bit is internally reset when the flushing operation is complete. Until this bit is reset, you should not write to the MTL_TxQ1_Operation_Mode register. The data which is already accepted by the MAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt packet transmission.</p> <p>Note: The flush operation is complete only when the Tx queue is empty and the application has accepted the pending Tx Status of all transmitted packets. To complete this flush operation, the PHY Tx clock (clk_tx_i) should be active.</p> <p>Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Flush Transmit Queue is disabled 0x1 (ENABLE): Flush Transmit Queue is enabled <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MTL_TxQ(#i)_Underflow (for i = 1; i <= 3)

Description: The Queue 1 Underflow Counter register contains the counter for packets aborted

because of Transmit queue underflow and packets missed because of Receive queue packet flush

Size: 32 bits

Offset: (0x0040*i)+0xD04

31:12	11	10:0
Reserved_31_12	UFCNTOVF	UFFRMCNT

Fields for Register: MTL_TxQ(#i)_Underflow (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:12	Reserved_31_12	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
11	UFCNTOVF	R	<p>Overflow Bit for Underflow Packet Counter</p> <p>This bit is set every time the Tx queue Underflow Packet Counter field overflows, that is, it has crossed the maximum count. In such a scenario, the overflow packet counter is reset to all-zeros and this bit indicates that the rollover happened.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Overflow not detected for Underflow Packet Counter 0x1 (ACTIVE): Overflow detected for Underflow Packet Counter <p>Value After Reset: 0x0</p>
10:0	UFFRMCNT	R	<p>Underflow Packet Counter</p> <p>This field indicates the number of packets aborted by the controller because of Tx Queue Underflow. This counter is incremented each time the MAC aborts outgoing packet because of underflow. The counter is cleared when this register is read with mci_be_i[0] at 1'b1.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Value After Reset: 0x0</p>

MTL_TxQ(#i)_Debug (for i = 1; i <= 3)

Description: The Queue 1 Transmit Debug register gives the debug status of various blocks related to the Transmit queue.

Size: 32 bits

Offset: (0x0040*i)+0xD08

31:23	22:20	19	18:1 6	15:6	5	4	3	2:1	0
Reserved_31_23	STXST SF	Reserved_19	PTX Q	Reserved_15_6	TXSTSFS TS	TXQSTS	TWCSTS	TRCSTS	TXQPAUSED

Fields for Register: MTL_TxQ(#i)_Debug (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:23	Reserved_31_23	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
22:20	STXSTSFS	R	<p>Number of Status Words in Tx Status FIFO of Queue</p> <p>This field indicates the current number of status in the Tx Status FIFO of this queue.</p> <p>When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of status words in Tx Status FIFO.</p> <p>Value After Reset: 0x0</p>
19	Reserved_19	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
18:16	PTXQ	R	<p>Number of Packets in the Transmit Queue</p> <p>This field indicates the current number of packets in the Tx Queue.</p> <p>When the DTXSTS bit of MTL_Operation_Mode register is set to 1, this field does not reflect the number of packets in the Transmit queue.</p> <p>Value After Reset: 0x0</p>
15:6	Reserved_15_6	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
5	TXSTSFS	R	<p>MTL Tx Status FIFO Full Status</p> <p>When high, this bit indicates that the MTL Tx Status FIFO is full. Therefore, the MTL cannot accept any more packets for transmission.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MTL Tx Status FIFO Full status is not detected 0x1 (ACTIVE): MTL Tx Status FIFO Full status is detected <p>Value After Reset: 0x0</p>
4	TXQSTS	R	<p>MTL Tx Queue Not Empty Status</p> <p>When this bit is high, it indicates that the MTL Tx Queue is not empty and some data is left for transmission.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x0 (INACTIVE): MTL Tx Queue Not Empty status is not detected</p> <p>0x1 (ACTIVE): MTL Tx Queue Not Empty status is detected</p> <p>Value After Reset: 0x0</p>
3	TWCSTS	R	<p>MTL Tx Queue Write Controller Status</p> <p>When high, this bit indicates that the MTL Tx Queue Write Controller is active, and it is transferring the data to the Tx Queue.</p> <p>Values:</p> <p>0x0 (INACTIVE): MTL Tx Queue Write Controller status is not detected</p> <p>0x1 (ACTIVE): MTL Tx Queue Write Controller status is detected</p> <p>Value After Reset: 0x0</p>
2:1	TRCSTS	R	<p>MTL Tx Queue Read Controller Status</p> <p>This field indicates the state of the Tx Queue Read Controller:</p> <p>Values:</p> <p>0x0 (IDLE): Idle state</p> <p>0x1 (READ): Read state (transferring data to the MAC transmitter)</p> <p>0x2 (WAIT): Waiting for pending Tx Status from the MAC transmitter</p> <p>0x3 (FLUSH): Flushing the Tx queue because of the Packet Abort request from the MAC</p> <p>Value After Reset: 0x0</p>
0	TXQPAUSED	R	<p>Transmit Queue in Pause</p> <p>When this bit is high and the Rx flow control is enabled, it indicates that the Tx Queue is in the Pause condition (in the full-duplex only mode) because of the following:</p> <ul style="list-style-type: none"> Reception of the PFC packet for the priorities assigned to the Tx Queue when PFC is enabled Reception of 802.3x Pause packet when PFC is disabled <p>Values:</p> <p>0x0 (INACTIVE): Transmit Queue in Pause status is not detected</p> <p>0x1 (ACTIVE): Transmit Queue in Pause status is detected</p> <p>Value After Reset: 0x0</p>

MTL_TxQ(#i)_ETS_Status (for i = 1; i <= 3)

Description: The Queue 1 ETS Status register provides the average traffic transmitted in Queue 1.

Size: 32 bits

Offset: (0x0040*i)+0x0D14

31:24	23:0
Reserved_31_24	ABS

Fields for Register: MTL_TxQ(#i)_ETS_Status (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:24	Reserved_31_24	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
23:0	ABS	R	<p>Average Bits per Slot</p> <p>This field contains the average transmitted bits per slot.</p> <p>If AV operation is enabled, this field is computed over number of slots, programmed in the SLC field of MTL_TxQ(#i)_ETS_CONTROL register. The maximum value of this field is 0x6_4000 in 100 Mbps, 0x3E_8000 in 1000 Mbps and 9C_4000 in 2500 Mbps mode respectively.</p> <p>When the DCB operation is enabled for Queue, this field is computed over every 10 million bit times slot (4 ms in 2500 Mbps; 10 ms in 1000 Mbps; 100 ms in 100 Mbps). The maximum value is 0x989680.</p> <p>Value After Reset: 0x0</p>

MTL_TxQ(#i)_Quantum_Weight (for i = 1; i <= 3)

Description: The Queue 1 idleSlopeCredit, Quantum or Weights register provides the average traffic transmitted in Queue 1.

Size: 32 bits

Offset: (0x0040*i)+0x0D18

31:21	20:0
Reserved_31_21	ISCQW

Fields for Register: MTL_TxQ(#i)_Quantum_Weight (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:21	Reserved_31_21	R	Reserved.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
20:0	ISCQW	R/W	<p>idleSlopeCredit, Quantum or Weights</p> <p>idleSlopeCredit</p> <p>When AV feature is enabled, this field contains the idleSlopeCredit value required for the credit-based shaper algorithm for Queue 1. This is the rate of change of credit in bits per cycle (40 ns for 100 Mbps; 8 ns for 1000 Mbps; 3.2 ns for 2500 Mbps) when the credit is increasing. The software should program this field with computed credit in bits per cycle scaled by 1,024. The maximum value is portTransmitRate, that is, 0x2000 in 1000/2500 Mbps mode and 0x1000 in 100 Mbps mode. Bits[20:14] must be written to zero.</p> <p>Quantum</p> <p>When the DCB operation is enabled with DWRR algorithm for Queue 1 traffic, this field contains the quantum value in bytes to be added to credit during every queue scanning cycle. The maximum value is 0x1312D0 bytes.</p> <p>Weights</p> <p>When DCB operation is enabled with WFQ algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x3FFF where weight of 0 indicates 100% bandwidth. Bits[20:14] must be written to zero. When DCB operation or generic queuing operation is enabled with WRR algorithm for Queue 1 traffic, this field contains the weight for this queue. The maximum value is 0x64. Bits [20:7] must be written to zero.</p> <p>Note 1: In multiple Queue configuration this field in respective per queue register must be programmed to some non-zero value when multiple queues are enabled or single queue other than Q0 is enabled. This field need not be programmed when only Q0 is enabled. In general, when WRR algorithm is selected a non-zero value must be programmed on both Receive and Transmit. In Receive, the register is MTL_Operation_Mode register.</p> <p>Note 2: For WFQ algorithm, higher the programmed weights lesser the bandwidth allocated for that Transmit Queue. The finish time is not a function of particular packet alone but it is as per the formula: (previous_finish_time of particular Transmit Queue + (weights*packet_size))</p> <p>Note 3: The weights programmed do not correspond to the number of packets but the fraction of bandwidth or time allocated for particular queue w.r.t. total BW or time.</p> <p>Value After Reset: 0x0</p>

MTL_Q(#i)_Interrupt_Control_Status) (for i = 1; i <= 3)

Description: This register contains the interrupt enable and status bits for the queue 1 interrupts.

Size: 32 bits**Offset:** (0x0040*i)+0x0D2C

31:25	24	23:17	16	15:10	9	8	7:2	1	0
Reserved_31_25	RXOI_E	Reserved_23_17	RXOVFI_S	Reserved_15_10	ABPSI_E	TXUI_E	Reserved_7_2	ABPSI_S	TXUNFI_S

Fields for Register: MTL_Q(#i)_Interrupt_Control_Status) (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:25	Reserved_31_25	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
24	RXOIE	R/W	<p>Receive Queue Overflow Interrupt Enable</p> <p>When this bit is set, the Receive Queue Overflow interrupt is enabled. When this bit is reset, the Receive Queue Overflow interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Queue Overflow Interrupt is disabled 0x1 (ENABLE): Receive Queue Overflow Interrupt is enabled <p>Value After Reset: 0x0</p>
23:17	Reserved_23_17	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
16	RXOVFIS	R/W	<p>Receive Queue Overflow Interrupt Status</p> <p>This bit indicates that the Receive Queue had an overflow while receiving the packet. If a partial packet is transferred to the application, the overflow status is set in RDES3[21]. This bit is cleared when the application writes 1 to this bit.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Receive Queue Overflow Interrupt Status not detected 0x1 (ACTIVE): Receive Queue Overflow Interrupt Status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
15:10	Reserved_15_10	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
9	ABPSIE	R/W	Average Bits Per Slot Interrupt Enable

Bits	Name	Memory Access	Description
			<p>When this bit is set, the MAC asserts the sbd_intr_o or mci_intr_o interrupt when the average bits per slot status is updated.</p> <p>When this bit is cleared, the interrupt is not asserted for such an event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Average Bits Per Slot Interrupt is disabled 0x1 (ENABLE): Average Bits Per Slot Interrupt is enabled <p>Value After Reset: 0x0</p>
8	TXUIE	R/W	<p>Transmit Queue Underflow Interrupt Enable</p> <p>When this bit is set, the Transmit Queue Underflow interrupt is enabled. When this bit is reset, the Transmit Queue Underflow interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Transmit Queue Underflow Interrupt Status is disabled 0x1 (ENABLE): Transmit Queue Underflow Interrupt Status is enabled <p>Value After Reset: 0x0</p>
7:2	Reserved_7_2	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
1	ABPSIS	R/W	<p>Average Bits Per Slot Interrupt Status</p> <p>When set, this bit indicates that the MAC has updated the ABS value. This bit is cleared when the application writes 1 to this bit.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Average Bits Per Slot Interrupt Status not detected 0x1 (ACTIVE): Average Bits Per Slot Interrupt Status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
0	TXUNFIS	R/W	<p>Transmit Queue Underflow Interrupt Status</p> <p>This bit indicates that the Transmit Queue had an underflow while transmitting the packet. Transmission is suspended and an Underflow Error TDES3[2] is set. This bit is cleared when the application writes 1 to this bit.</p>

Bits	Name	Memory Access	Description
			<p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Transmit Queue Underflow Interrupt Status not detected 0x1 (ACTIVE): Transmit Queue Underflow Interrupt Status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

MTL_RxQ(#i)_Operation_Mode (for i = 1; i <= 3)

Description: The Queue 1 Receive Operation Mode register establishes the Receive queue operating modes and command.

The RFA and RFD fields are not backward compatible with the RFA and RFD fields of 4.00a release

Size: 32 bits

Offset: (0x0040*i)+0x0D30

31:y	23:2 0	19:17	16:1 4	13:11	10: 8	7	6	5	4	3	2	1:0
Reserved_31_y	RQS	Reserved_19_y	RFD	Reserved_13_y	RF_A	EHF_C	DIS_TCP_EF	RS_F	FE_P	FU_P	Reserved_2	RT_C

Fields for Register: MTL_RxQ(#i)_Operation_Mode (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:24	Reserved_31_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
23:20	RQS	R/W	<p>Receive Queue Size</p> <p>This field indicates the size of the allocated Receive queues in blocks of 256 bytes. The RQS field is read-write only if the number of Rx Queues more than one, the reset value is 0x0 and indicates size of 256 bytes. This means that value of 0x0 = 256 bytes, 0x1 = 512 bytes and so on. So, program RQS [5:0] = 6'b001111 to allocate queue size of 4096 (4K) bytes. In general, the size of the Queue = (RQS+1)*256 bytes.</p> <p>When the number of Rx Queues is one, the field is read-only and the configured RX FIFO size in blocks of 256 bytes is reflected in the reset value.</p> <p>The width of this field depends on the Rx memory size selected in your</p>

Bits	Name	Memory Access	Description
			<p>configuration. For example, if the memory size is 2048, the width of this field is 3 bits:</p> $\text{LOG2}(2048/256) = \text{LOG2}(8) = 3 \text{ bits}$ <p>Value After Reset: 0x0</p>
19:17	Reserved_19_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
16:14	RFD	R/W	<p>Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes)</p> <p>These bits control the threshold (fill-level of Rx queue) at which the flow control is de-asserted after activation:</p> <ul style="list-style-type: none"> 0: Full minus 1 KB, that is, FULL 1 KB 1: Full minus 1.5 KB, that is, FULL 1.5 KB 2: Full minus 2 KB, that is, FULL 2 KB 3: Full minus 2.5 KB, that is, FULL 2.5 KB ... 6: Full minus 4 KB, that is, FULL 4 KB 7: Full minus 4.5 KB, that is, FULL 4.5 KB <p>The de-assertion is effective only after flow control is asserted.</p> <p>Note: The value must be programmed in such a way to make sure that the threshold is a positive number.</p> <p>When the EHFC is set high, these values are applicable only when the Rx queue size determined by the RQS field of this register, is equal to or greater than 4 KB.</p> <p>For a given queue size, the values ranges between 0 and the encoding for FULL minus (QSIZE - 0.5 KB) and all other values are illegal. Here the term FULL and QSIZE refers to the queue size determined by the RQS field of this register.</p> <p>The width of this field depends on RX FIFO size selected during the configuration. Remaining bits are reserved and read only.</p> <p>Value After Reset: 0x0</p>
13:11	Reserved_13_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
10:8	RFA	R/W	<p>Threshold for Activating Flow Control (in half-duplex and full-duplex)</p> <p>These bits control the threshold (fill-level of Rx queue) at which the flow control is activated:</p> <p>For more information on encoding for this field, see RFD.</p>

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
7	EHFC	R/W	<p>Enable Hardware Flow Control</p> <p>When this bit is set, the flow control signal operation, based on the fill-level of Rx queue, is enabled. When reset, the flow control operation is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Hardware Flow Control is disabled 0x1 (ENABLE): Hardware Flow Control is enabled <p>Value After Reset: 0x0</p>
6	DIS_TCP_EF	R/W	<p>Disable Dropping of TCP/IP Checksum Error Packets</p> <p>When this bit is set, the MAC does not drop the packets which only have the errors detected by the Receive Checksum Offload engine. Such packets have errors only in the encapsulated payload. There are no errors (including FCS error) in the Ethernet packet received by the MAC.</p> <p>When this bit is reset, all error packets are dropped if the FEP bit is reset.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ENABLE): Dropping of TCP/IP Checksum Error Packets is enabled 0x1 (DISABLE): Dropping of TCP/IP Checksum Error Packets is disabled <p>Value After Reset: 0x0</p>
5	RSF	R/W	<p>Receive Queue Store and Forward</p> <p>When this bit is set, the DWC_ether_qos reads a packet from the Rx queue only after the complete packet has been written to it, ignoring the RTC field of this register. When this bit is reset, the Rx queue operates in the Threshold (cut-through) mode, subject to the threshold specified by the RTC field of this register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Queue Store and Forward is disabled 0x1 (ENABLE): Receive Queue Store and Forward is enabled <p>Value After Reset: 0x0</p>
4	FEP	R/W	<p>Forward Error Packets</p> <p>When this bit is reset, the Rx queue drops packets with error status (CRC error, GMII_ER, watchdog timeout, or overflow). However, if the start byte (write) pointer of a packet is already transferred to the read controller side (in Threshold mode), the packet is not dropped.</p>

Bits	Name	Memory Access	Description
			<p>When this bit is set, all packets except the runt error packets are forwarded to the application or DMA. If the RSF bit is set and the Rx queue overflows when a partial packet is written, the packet is dropped irrespective of the setting of this bit. However, if the RSF bit is reset and the Rx queue overflows when a partial packet is written, a partial packet might be forwarded to the application or DMA.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Forward Error Packets is disabled 0x1 (ENABLE): Forward Error Packets is enabled <p>Value After Reset: 0x0</p>
3	FUP	R/W	<p>Forward Undersized Good Packets</p> <p>When this bit is set, the Rx queue forwards the undersized good packets (packets with no error and length less than 64 bytes), including pad-bytes and CRC. When this bit is reset, the Rx queue drops all packets of less than 64 bytes, unless a packet is already transferred because of the lower value of Rx Threshold, for example, RTC = 01.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Forward Undersized Good Packets is disabled 0x1 (ENABLE): Forward Undersized Good Packets is enabled <p>Value After Reset: 0x0</p>
2	Reserved_2	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
1:0	RTC	R/W	<p>Receive Queue Threshold Control</p> <p>These bits control the threshold level of the MTL Rx queue (in bytes): The received packet is transferred to the application or DMA when the packet size within the MTL Rx queue is larger than the threshold. In addition, full packets with length less than the threshold are automatically transferred.</p> <p>This field is valid only when the RSF bit is zero. This field is ignored when the RSF bit is set to 1.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (M_64BYTE): 64 0x1 (M_32BYTE): 32 0x2 (M_96BYTE): 96 0x3 (M_128BYTE): 128 <p>Value After Reset: 0x0</p>

MTL_RxQ(#i)_Missed_Packet_Overflow_Cnt (for i = 1; i <= 3)

Description: The Queue 1 Missed Packet and Overflow Counter register contains the counter for packets missed because of Receive queue packet flush and packets discarded because of Receive queue overflow.

Size: 32 bits

Offset: (0x0040*i)+0x0D34

31:28	27	26:16	15:12	11	10:0
Reserved_31_28	MISCNTOVF	MISPKTCNT	Reserved_15_12	OVFCNTOVF	OVFPKTCNT

Fields for Register: MTL_RxQ(#i)_Missed_Packet_Overflow_Cnt (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:28	Reserved_31_28	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
27	MISCNTOVF	R	<p>Missed Packet Counter Overflow Bit</p> <p>When set, this bit indicates that the Rx Queue Missed Packet Counter crossed the maximum limit.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Missed Packet Counter overflow not detected 0x1 (ACTIVE): Missed Packet Counter overflow detected <p>Value After Reset: 0x0</p>
26:16	MISPKTCNT	R	<p>Missed Packet Counter</p> <p>This field indicates the number of packets missed by the DWC_ether_qos because the application asserted ari_pkt_flush_i[] for this queue. This counter is reset when this register is read with mci_be_i[0] at 1b1.</p> <p>This counter is incremented by 1 when the DMA discards the packet because of buffer unavailability.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Value After Reset: 0x0</p>
15:12	Reserved_15_12	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
11	OVFCNTOVF	R	<p>Overflow Counter Overflow Bit</p> <p>When set, this bit indicates that the Rx Queue Overflow Packet</p>

Bits	Name	Memory Access	Description
			<p>Counter field crossed the maximum limit.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Overflow Counter overflow not detected 0x1 (ACTIVE): Overflow Counter overflow detected <p>Value After Reset: 0x0</p>
10:0	OVFPKTCNT	R	<p>Overflow Packet Counter</p> <p>This field indicates the number of packets discarded by the DWC_ether_qos because of Receive queue overflow. This counter is incremented each time the DWC_ether_qos discards an incoming packet because of overflow. This counter is reset when this register is read with mci_be_i[0] at 1'b1.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Value After Reset: 0x0</p>

MTL_RxQ(#i)_Debug (for i = 1; i <= 3)

Description: The Queue 1 Receive Debug register gives the debug status of various blocks related to the Receive queue.

Size: 32 bits

Offset: (0x0040*i)+0x0D38

31:30	29:16	15:6	5:4	3	2:1	0
Reserved_31_30	PRXQ	Reserved_15_6	RXQSTS	Reserved_3	RRCSTS	RWCSTS

Fields for Register: MTL_RxQ(#i)_Debug (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:30	Reserved_31_30	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
29:16	PRXQ	R	<p>Number of Packets in Receive Queue</p> <p>This field indicates the current number of packets in the Rx Queue. The theoretical maximum value for this field is 256KB/16B = 16K Packets, that is, Max_Queue_Size/Min_Packet_Size.</p> <p>Value After Reset: 0x0</p>
15:6	Reserved_15_6	R	Reserved.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
5:4	RXQSTS	R	<p>MTL Rx Queue Fill-Level Status</p> <p>This field gives the status of the fill-level of the Rx Queue:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (EMPTY): Rx Queue empty 0x1 (BLW_THR): Rx Queue fill-level below flow-control deactivate threshold 0x2 (ABV_THR): Rx Queue fill-level above flow-control activate threshold 0x3 (FULL): Rx Queue full <p>Value After Reset: 0x0</p>
3	Reserved_3	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
2:1	RRCSTS	R	<p>MTL Rx Queue Read Controller State</p> <p>This field gives the state of the Rx queue Read controller:</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (IDLE): Idle state 0x1 (READ_DATA): Reading packet data 0x2 (READ_STS): Reading packet status (or timestamp) 0x3 (FLUSH): Flushing the packet data and status <p>Value After Reset: 0x0</p>
0	RWCSTS	R	<p>MTL Rx Queue Write Controller Active Status</p> <p>When high, this bit indicates that the MTL Rx queue Write controller is active, and it is transferring a received packet to the Rx Queue.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MTL Rx Queue Write Controller Active Status not detected 0x1 (ACTIVE): MTL Rx Queue Write Controller Active Status detected <p>Value After Reset: 0x0</p>

MTL_RxQ(#i)_Control (for i = 1; i <= 3)

Description: The Queue Receive Control register controls the receive arbitration and passing of

received packets to the application.

Size: 32 bits

Offset: (0x0040*i)+0x0D3C

31:4	3	2:0
Reserved_31_4	RXQ_FRM_ARBIT	RXQ_WEGT

Fields for Register: MTL_RxQ(#i)_Control (for i = 1; i <= 3)

Bits	Name	Memory Access	Description
31:4	Reserved_31_4	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
3	RXQ_FRM_ARBIT	R/W	<p>Receive Queue Packet Arbitration</p> <p>When this bit is set, the DWC_ether_qos drives the packet data to the ARI interface such that the entire packet data of currently-selected queue is transmitted before switching to other queue.</p> <p>When this bit is reset, the DWC_ether_qos drives the packet data to the ARI interface such that the following amount of data of currently-selected queue is transmitted before switching to other queue:</p> <ul style="list-style-type: none"> PBL amount of data (indicated by ari_qN_pbl_i[]) or Complete data of a packet <p>The status and the timestamp are not a part of the PBL data. Therefore, the DWC_ether_qos drives the complete status (including timestamp status) during first PBL request for the packet (in store-and-forward mode) or the last PBL request for the packet (in Threshold mode).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Queue Packet Arbitration is disabled 0x1 (ENABLE): Receive Queue Packet Arbitration is enabled <p>Value After Reset: 0x0</p>
2:0	RXQ_WEGT	R/W	<p>Receive Queue Weight</p> <p>This field indicates the weight assigned to the Rx Queue 0. This field needs to be programmed with one value less than the required weight, that is, reset value of 0 indicates weight of 1, value of 1 indicates weight of 2, and so on. The weight is used as the number of continuous PBL or packets requests (depending on the RXQ_FRM_ARBIT) allocated to the queue in one arbitration cycle.</p> <p>Note: The change in value of RXQ_WEGT takes effect only after the completion of current service round or when there is change from RAA=SP to RAA=WSP algorithm. This approach is taken so that there is smooth transition. For the RXQ_WEGT value to take effect at the start, the MTL_RxQ(#i)_Control registers must be programmed</p>

Bits	Name	Memory Access	Description
			before the MTL_Operation_Mode register. Value After Reset: 0x0

DMA_Mode

Description: The Bus Mode register establishes the bus operating modes for the DMA.

Size: 32 bits

Offset: 0x1000

31:24	23:2 RN DF	21:2 TN DF	19 DC HE	18 Reserved _18	17:1 6 INT M	15 Reserved _15	14:1 2 Rsv d	11 TXP R	10: 9 Rsv d	8 DSP W	7:5 Reserved _7_5	4:2 TA A	1 Rsv d	0 SW R
Reserved_31_24														

Fields for Register: DMA_Mode

Bits	Name	Memory Access	Description
31:24	Reserved_31_24	R	Reserved. Value After Reset: 0x0
23:22	RNDF	R/W	Rx DMA's Maximum Number of Descriptors to be fetched in a burst 00: DWC_EQOS_DCRX_DEP 01: DWC_EQOS_DCRX_DEP/2 10: DWC_EQOS_DCRX_DEP/4 11: DWC_EQOS_DCRX_DEP/8 (Reserved when DWC_EQOS_DCRX_DEP = 4) Value After Reset: 0x0
21:20	TNDF	R/W	Tx DMA's Maximum Number of Descriptors to be fetched in a burst 00: DWC_EQOS_DCTX_DEP 01: DWC_EQOS_DCTX_DEP/2 10: DWC_EQOS_DCTX_DEP/4 11: DWC_EQOS_DCTX_DEP/8 (Reserved when DWC_EQOS_DCTX_DEP = 4) Value After Reset: 0x0
19	DCHE	R/W	Descriptor Cache Enable When set enables prefetching of descriptors to the Descriptor Cache. When reset descriptor cache feature is disabled. Values:

Bits	Name	Memory Access	Description
			<p>0x0 (DISABLE): Descriptor Cache Support is disabled 0x1 (ENABLE): Descriptor Cache Support is enabled Value After Reset: 0x0</p>
18	Reserved_18	R	<p>Reserved. Value After Reset: 0x0</p>
17:16	INTM	R/W	<p>Interrupt Mode This field defines the interrupt mode of DWC_ether_qos.</p> <p>The behavior of the following outputs change depending on the following settings:</p> <ul style="list-style-type: none"> sbd_perch_tx_intr_o[] (Transmit Per Channel Interrupt) sbd_perch_rx_intr_o[] (Receive Per Channel Interrupt) sbd_intr_o (Common Interrupt) <p>It also changes the behavior of the RI/TI bits in the DMA_CH0_Status.</p> <p>00: sbd_perch_* are pulse signals for each TX/RX packet transfer completion events (irrespective of whether corresponding interrupts are enabled) for which IOC bits are enabled in descriptor. sbd_intr_o is also asserted when corresponding interrupts are enabled and cleared only when software clears the corresponding RI/TI status bits.</p> <p>01: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events.</p> <p>10: sbd_perch_* are level signals asserted on TX/RX packet transfer completion event when corresponding interrupts are enabled and de-asserted when the software clears the corresponding RI/TI status bits. However, the signal is asserted again if the same event occurred again before it was cleared. The sbd_intr_o is not asserted for these TX/RX packet transfer completion events.</p> <p>11: Reserved</p> <p>For more details, see Table "DWC_ether_qos Transfer Complete Interrupt Behavior".</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (MODE0): See above description 0x1 (MODE1): See above description 0x2 (MODE2): See above description 0x3 (RSVD): Reserved

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
15	Reserved_15	R	Reserved. Value After Reset: 0x0
14:12			Reserved Field: Yes
11	TXPR	R/W	Transmit Priority When set, this bit indicates that the Tx DMA has higher priority than the Rx DMA during arbitration for the system-side bus or Descriptor reads from DCACHE memory when DWC_EQOS_DCEXT is enabled Values: 0x0 (DISABLE): Transmit Priority is disabled 0x1 (ENABLE): Transmit Priority is enabled Value After Reset: 0x0
10:9			Reserved Field: Yes
8	DSPW	R/W	Descriptor Posted Write When this bit is set to 0: The descriptor writes are always non-posted. 1: The descriptor writes are non-posted only when IOC (Interrupt on completion) is set in last descriptor. Otherwise, the descriptor writes are always posted. Values: 0x0 (DISABLE): Descriptor Posted Write is disabled 0x1 (ENABLE): Descriptor Posted Write is enabled Value After Reset: 0x0
7:5	Reserved_7_5	R	Reserved. Value After Reset: 0x0
4:2	TAA	R/W	Transmit Arbitration Algorithm This field is used to select the arbitration algorithm for the Transmit side when multiple Tx DMAs are selected. Values: 0x0 (FP): Fixed priority (Channel 0 has the lowest priority and the last channel has the highest priority) 0x1 (WSP): Weighted Strict Priority (WSP) 0x2 (WRR): Weighted Round-Robin (WRR)

Bits	Name	Memory Access	Description
			<p>0x3 (RSVD): Reserved (for 3'b011 to 3'b111)</p> <p>Value After Reset: 0x0</p>
1			<p>Reserved Field: Yes</p>
0	SWR	R/W	<p>Software Reset</p> <p>When this bit is set, the MAC and the DMA controller reset the logic and all internal registers of the DMA, MTL, and MAC. This bit is automatically cleared after the reset operation is complete in all DWC_ether_qos clock domains. Before reprogramming any DWC_ether_qos register, a value of zero should be read in this bit.</p> <p>This bit must be read at least 4 CSR clock cycles after it is written to 1.</p> <p>Note: The reset operation is complete only when all resets in all active clock domains are de-asserted. Therefore, it is essential that all PHY inputs clocks (applicable for the selected PHY interface) are present for software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock.</p> <p>Access restriction applies. Setting 1 sets. Self-cleared. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Software Reset is disabled 0x1 (ENABLE): Software Reset is enabled <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

DMA_SysBus_Mode

Description: The System Bus mode register controls the behavior of the AHB or AXI master. It mainly controls burst splitting and number of outstanding requests.

Size: 32 bits

Offset: 0x1004

31	30	29:27	26:24	23:19	18:16	15:14	13	12	11	10	9:8	7:4	3	2	1	0
EN_LPI	Rsvd	Reserved_29_y	WR_OSR_LMT	Reserved_23_y	RD_OSR_LMT	Rsvd	ONEKBBE	AAL	EAME	AALE	Reserved_9_8	Rsvd	BLEN16	BLEN8	BLEN4	FB

Fields for Register: DMA_SysBus_Mode

Bits	Name	Memory Access	Description
31	EN_LPI	R/W	<p>Enable Low Power Interface (LPI)</p> <p>When set to</p> <ul style="list-style-type: none"> 1: This bit enables the LPI mode supported by the EQOS-AXI configuration and accepts the LPI request from the AXI System Clock controller. 0: This bit disables the LPI mode. Denies the LPI request from the AXI System Clock controller. <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Low Power Interface (LPI) is disabled 0x1 (ENABLE): Low Power Interface (LPI) is enabled <p>Value After Reset: 0x0</p>
30			Reserved Field: Yes
29:27	Reserved_29_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
26:24	WR_OSR_LMT	R/W	<p>AXI Maximum Write Outstanding Request Limit</p> <p>This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT + 1</p> <p>Note:</p> <ul style="list-style-type: none"> Bit 26 is reserved if DWC_Ether_QoS_AXI_Max_Wr_Req = 4 Bit 27 is reserved if DWC_Ether_QoS_AXI_Max_Wr_Req!= 16 <p>Value After Reset: 0x1</p>
23:19	Reserved_23_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
18:16	RD_OSR_LMT	R/W	<p>AXI Maximum Read Outstanding Request Limit</p> <p>This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT + 1</p> <p>Note:</p> <ul style="list-style-type: none"> Bit 18 is reserved if parameter DWC_Ether_QoS_AXI_Max_Rd_Req = 4 Bit 19 is reserved if parameter DWC_Ether_QoS_AXI_Max_Rd_Req!= 16 <p>Value After Reset: 0x1</p>
15:14			Reserved Field: Yes
13	ONEKBEE	R/W	1 KB Boundary Crossing Enable for the EQOS-AXI Master

Bits	Name	Memory Access	Description
			<p>1: The burst transfers performed by the EQOS-AXI master do not cross 1 KB boundary.</p> <p>0: The burst transfers performed by the EQOS-AXI master do not cross 4 KB boundary.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): 1 KB Boundary Crossing for the EQOS-AXI Master Beats is disabled 0x1 (ENABLE): 1 KB Boundary Crossing for the EQOS-AXI Master Beats is enabled <p>Value After Reset: 0x0</p>
12	AAL	R/W	<p>Address-Aligned Beats</p> <p>When this bit is set to 1, the EQOS-AXI or EQOS-AHB master performs address-aligned burst transfers on Read and Write channels.</p> <p>When this bit is set to 0, the EQOS-AXI or EQOS-AHB master performs burst transfers on Read and Write channels without aligning to address boundaries.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Address-Aligned Beats is disabled 0x1 (ENABLE): Address-Aligned Beats is enabled <p>Value After Reset: 0x0</p>
11	EAME	R/W	<p>Enhanced Address Mode Enable.</p> <p>When set to</p> <p>1: the DMA master enables the enhanced address mode (40-bit or 48-bit addressing mode). In this mode, the DMA engine uses either the 40- or 48-bit address, depending on the configuration.</p> <p>0: the DMA master enables the normal address mode (32-bit).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Enhanced Address Mode is disabled 0x1 (ENABLE): Enhanced Address Mode is enabled <p>Value After Reset: 0x0</p>
10	AALE	R/W	<p>Automatic AXI LPI enable</p> <p>When set to 1, enables the AXI master to enter into LPI state when there is no activity in the DWC_ether_qos for number of system clock cycles programmed in the LPIEI field of AXI_LPI_Entry_Interval register.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (DISABLE): Automatic AXI LPI is disabled 0x1 (ENABLE): Automatic AXI LPI is enabled Value After Reset: 0x0</p>
9:8	Reserved_9_8	R	<p>Reserved. Value After Reset: 0x0</p>
7:4			<p>Reserved Field: Yes</p>
3	BLEN16	R/W	<p>AXI Burst Length 16 When this bit is set to 1 or the FB bit is set to 1, the EQOS-AXI master can select a burst length of 16 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. Values: 0x0 (DISABLE): No effect 0x1 (ENABLE): AXI Burst Length 16 Value After Reset: 0x0</p>
2	BLEN8	R/W	<p>AXI Burst Length 8 When this bit is set to 1 or the FB bit is set to 1, the EQOS-AXI master can select a burst length of 8 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. Values: 0x0 (DISABLE): No effect 0x1 (ENABLE): AXI Burst Length 8 Value After Reset: 0x0</p>
1	BLEN4	R/W	<p>AXI Burst Length 4 When this bit is set to 1 or the FB bit is set to 1, the EQOS-AXI master can select a burst length of 4 on the AXI interface. When the FB bit is set to 0, setting this bit has no effect. Values: 0x0 (DISABLE): No effect 0x1 (ENABLE): AXI Burst Length 4 Value After Reset: 0x0</p>
0	FB	R/W	<p>Fixed Burst Length 1: the EQOS-AXI master initiates burst transfers of specified lengths:</p>

Bits	Name	Memory Access	Description
			<p>Burst transfers of fixed burst lengths as indicated by the BLEN256, BLEN128, BLEN64, BLEN32, BLEN16, BLEN8, or BLEN4 field</p> <p>Burst transfers of length 1</p> <p>0: EQOS-AXI master initiates burst transfers that are equal to or less than the maximum allowed burst length programmed in Bits[7:4].</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Fixed Burst Length is disabled 0x1 (ENABLE): Fixed Burst Length is enabled <p>Value After Reset: 0x0</p>

DMA_Interrupt_Status

Description: The application reads this Interrupt Status register during interrupt service routine or polling to determine the interrupt status of DMA channels, MTL queues, and the MAC.

Size: 32 bits

Offset: 0x1008

31:18	17	16	15:8	7:4	3	2	1	0
Reserved_31_18	MACIS	MTLIS	Reserved_15_8	Rsvd	DC3IS	DC2IS	DC1IS	DC0IS

Fields for Register: DMA_Interrupt_Status

Bits	Name	Memory Access	Description
31:18	Reserved_31_18	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
17	MACIS	R	<p>MAC Interrupt Status</p> <p>This bit indicates an interrupt event in the MAC. To reset this bit to 1'b0, the software must read the corresponding register in the MAC to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MAC Interrupt Status not detected 0x1 (ACTIVE): MAC Interrupt Status detected <p>Value After Reset: 0x0</p>
16	MTLIS	R	<p>MTL Interrupt Status</p> <p>This bit indicates an interrupt event in the MTL. To reset this bit to</p>

Bits	Name	Memory Access	Description
			<p>1'b0, the software must read the corresponding register in the MTL to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): MTL Interrupt Status not detected 0x1 (ACTIVE): MTL Interrupt Status detected <p>Value After Reset: 0x0</p>
15:8	Reserved_15_8	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
7:4			<p>Reserved Field: Yes</p>
3	DC3IS	R	<p>DMA Channel 3 Interrupt Status</p> <p>This bit indicates an interrupt event in DMA Channel 3. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 3 to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): DMA Channel 3 Interrupt Status not detected 0x1 (ACTIVE): DMA Channel 3 Interrupt Status detected <p>Value After Reset: 0x0</p>
2	DC2IS	R	<p>DMA Channel 2 Interrupt Status</p> <p>This bit indicates an interrupt event in DMA Channel 2. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 2 to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): DMA Channel 2 Interrupt Status not detected 0x1 (ACTIVE): DMA Channel 2 Interrupt Status detected <p>Value After Reset: 0x0</p>
1	DC1IS	R	<p>DMA Channel 1 Interrupt Status</p> <p>This bit indicates an interrupt event in DMA Channel 1. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 1 to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): DMA Channel 1 Interrupt Status not detected 0x1 (ACTIVE): DMA Channel 1 Interrupt Status detected <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
0	DC0IS	R	<p>DMA Channel 0 Interrupt Status</p> <p>This bit indicates an interrupt event in DMA Channel 0. To reset this bit to 1'b0, the software must read the corresponding register in DMA Channel 0 to get the exact cause of the interrupt and clear its source.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): DMA Channel 0 Interrupt Status not detected 0x1 (ACTIVE): DMA Channel 0 Interrupt Status detected <p>Value After Reset: 0x0</p>

DMA_Debug_Status0

Description: The Debug Status 0 register gives the Receive and Transmit process status for DMA Channel 0-Channel 2 for debugging purpose.

Size: 32 bits

Offset: 0x100c

31:28	27:24	23:20	19:16	15:12	11:8	7:2	1	0
TPS2	RPS2	TPS1	RPS1	TPS0	RPS0	Reserved_7_2	AXRHSTS	AXWHSTS

Fields for Register: DMA_Debug_Status0

Bits	Name	Memory Access	Description
31:28	TPS2	R	<p>DMA Channel 2 Transmit Process State</p> <p>This field indicates the Tx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (STOP): Stopped (Reset or Stop Transmit Command issued) 0x1 (RUN_FTTD): Running (Fetching Tx Transfer Descriptor) 0x2 (RUN_WS): Running (Waiting for status) 0x3 (RUN_RDS): Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4 (TSTMP_WS): Timestamp write state 0x5 (RSVD): Reserved for future use 0x6 (SUSPND): Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7 (RUN_CTD): Running (Closing Tx Descriptor)

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
27:24	RPS2	R	<p>DMA Channel 2 Receive Process State</p> <p>This field indicates the Rx DMA FSM state for Channel 2. The MSB of this field always returns 0. This field does not generate an interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (STOP): Stopped (Reset or Stop Receive Command issued) 0x1 (RUN_FRTD): Running (Fetching Rx Transfer Descriptor) 0x2 (RSVD): Reserved for future use 0x3 (RUN_WRP): Running (Waiting for Rx packet) 0x4 (SUSPND): Suspended (Rx Descriptor Unavailable) 0x5 (RUN_CRD): Running (Closing the Rx Descriptor) 0x6 (TSTMP): Timestamp write state 0x7 (RUN_TRP): Running (Transferring the received packet data from the Rx buffer to the system memory) <p>Value After Reset: 0x0</p>
23:20	TPS1	R	<p>DMA Channel 1 Transmit Process State</p> <p>This field indicates the Tx DMA FSM state for Channel 1. The MSB of this field always returns 0. This field does not generate an interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (STOP): Stopped (Reset or Stop Transmit Command issued) 0x1 (RUN_FTTD): Running (Fetching Tx Transfer Descriptor) 0x2 (RUN_WS): Running (Waiting for status) 0x3 (RUN_RDS): Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4 (TSTMP_WS): Timestamp write state 0x5 (RSVD): Reserved for future use 0x6 (SUSPND): Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7 (RUN_CTD): Running (Closing Tx Descriptor) <p>Value After Reset: 0x0</p>
19:16	RPS1	R	<p>DMA Channel 1 Receive Process State</p> <p>This field indicates the Rx DMA FSM state for Channel 1. The MSB of this field always returns 0. This field does not generate an interrupt.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (STOP): Stopped (Reset or Stop Receive Command issued)</p> <p>0x1 (RUN_FRTD): Running (Fetching Rx Transfer Descriptor)</p> <p>0x2 (RSVD): Reserved for future use</p> <p>0x3 (RUN_WRP): Running (Waiting for Rx packet)</p> <p>0x4 (SUSPND): Suspended (Rx Descriptor Unavailable)</p> <p>0x5 (RUN_CRD): Running (Closing the Rx Descriptor)</p> <p>0x6 (TSTMP): Timestamp write state</p> <p>0x7 (RUN_TRP): Running (Transferring the received packet data from the Rx buffer to the system memory)</p> <p>Value After Reset: 0x0</p>
15:12	TPS0	R	<p>DMA Channel 0 Transmit Process State</p> <p>This field indicates the Tx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (STOP): Stopped (Reset or Stop Transmit Command issued) 0x1 (RUN_FTTD): Running (Fetching Tx Transfer Descriptor) 0x2 (RUN_WS): Running (Waiting for status) 0x3 (RUN_RDS): Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4 (TSTMP_WS): Timestamp write state 0x5 (RSVD): Reserved for future use 0x6 (SUSPND): Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7 (RUN_CTD): Running (Closing Tx Descriptor) <p>Value After Reset: 0x0</p>
11:8	RPS0	R	<p>DMA Channel 0 Receive Process State</p> <p>This field indicates the Rx DMA FSM state for Channel 0. The MSB of this field always returns 0. This field does not generate an interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (STOP): Stopped (Reset or Stop Receive Command issued) 0x1 (RUN_FRTD): Running (Fetching Rx Transfer Descriptor) 0x2 (RSVD): Reserved for future use 0x3 (RUN_WRP): Running (Waiting for Rx packet) 0x4 (SUSPND): Suspended (Rx Descriptor Unavailable)

Bits	Name	Memory Access	Description
			<p>0x5 (RUN_CRD): Running (Closing the Rx Descriptor) 0x6 (TSTMP): Timestamp write state 0x7 (RUN_TRP): Running (Transferring the received packet data from the Rx buffer to the system memory)</p> <p>Value After Reset: 0x0</p>
7:2	Reserved_7_2	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
1	AXRHSTS	R	<p>AXI Master Read Channel Status When high, this bit indicates that the read channel of the AXI master is active, and it is transferring the data.</p> <p>Values:</p> <p>0x0 (INACTIVE): AXI Master Read Channel Status not detected 0x1 (ACTIVE): AXI Master Read Channel Status detected</p> <p>Value After Reset: 0x0</p>
0	AXWHSTS	R	<p>AXI Master Write Channel When high, this bit indicates that the write channel of the AXI master is active, and it is transferring data.</p> <p>Values:</p> <p>0x0 (INACTIVE): AXI Master Write Channel or AHB Master Status not detected 0x1 (ACTIVE): AXI Master Write Channel or AHB Master Status detected</p> <p>Value After Reset: 0x0</p>

DMA_Debug_Status1

Description: The Debug Status1 register gives the Receive and Transmit process status for DMA Channel 3-Channel 6.

Size: 32 bits

Offset: 0x1010

31:8	7:4	3:0
Rsvd	TPS3	RPS3

Fields for Register: DMA_Debug_Status1

Bits	Name	Memory Access	Description
31:8			Reserved Field: Yes
7:4	TPS3	R	<p>DMA Channel 3 Transmit Process State</p> <p>This field indicates the Tx DMA FSM state for Channel 3. The MSB of this field always returns 0. This field does not generate an interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (STOP): Stopped (Reset or Stop Transmit Command issued) 0x1 (RUN_FTTD): Running (Fetching Tx Transfer Descriptor) 0x2 (RUN_WS): Running (Waiting for status) 0x3 (RUN_RDS): Running (Reading Data from system memory buffer and queuing it to the Tx buffer (Tx FIFO)) 0x4 (TSTMP_WS): Timestamp write state 0x5 (RSVD): Reserved for future use 0x6 (SUSPND): Suspended (Tx Descriptor Unavailable or Tx Buffer Underflow) 0x7 (RUN_CTD): Running (Closing Tx Descriptor) <p>Value After Reset: 0x0</p>
3:0	RPS3	R	<p>DMA Channel 3 Receive Process State</p> <p>This field indicates the Rx DMA FSM state for Channel 3. The MSB of this field always returns 0. This field does not generate an interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (STOP): Stopped (Reset or Stop Receive Command issued) 0x1 (RUN_FRTD): Running (Fetching Rx Transfer Descriptor) 0x2 (RSVD): Reserved for future use 0x3 (RUN_WRP): Running (Waiting for Rx packet) 0x4 (SUSPND): Suspended (Rx Descriptor Unavailable) 0x5 (RUN_CRD): Running (Closing the Rx Descriptor) 0x6 (TSTMP): Timestamp write state 0x7 (RUN_TRP): Running (Transferring the received packet data from the Rx buffer to the system memory) <p>Value After Reset: 0x0</p>

AXI_LPI_Entry_Interval

Description: This register is used to control the AXI LPI entry interval.

Size: 32 bits**Offset:** 0x1040

31:4	3:0
Reserved_31_4	LPIEI

Fields for Register: AXI_LPI_Entry_Interval

Bits	Name	Memory Access	Description
31:4	Reserved_31_4	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
3:0	LPIEI	R/W	<p>LPI Entry Interval</p> <p>Contains the number of system clock cycles, multiplied by 64, to wait for an activity in the DWC_ether_qos to enter into the AXI low power state</p> <p>0 indicates 64 clock cycles</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_Control (for i = 0; i <= 3)

Description: The DMA Channel/Control register specifies the MSS value for segmentation, length to skip between two descriptors, and also the features such as header splitting and 8xPBL mode.

Size: 32 bits**Offset:** (0x0080*i)+0x1100

31:25	24	23:21	20:18	17	16	15:14	13:0
Reserved_31_25	Rsvd	Reserved_23_21	DSL	Reserved_17	PBLx8	Reserved_15_14	Rsvd

Fields for Register: DMA_CH(#i)_Control (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:25	Reserved_31_25	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
24			Reserved Field: Yes
23:21	Reserved_23_21	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
20:18	DSL	R/W	<p>Descriptor Skip Length</p> <p>This bit specifies the Word, Dword, or Lword number (depending on the 32-bit, 64-bit, or 128-bit bus) to skip between two unchained descriptors. The address skipping starts from the end of the current</p>

Bits	Name	Memory Access	Description
			<p>descriptor to the start of the next descriptor.</p> <p>When the DSL value is equal to zero, the DMA takes the descriptor table as contiguous.</p> <p>Value After Reset: 0x0</p>
17	Reserved_17	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
16	PBLx8	R/W	<p>8xPBL mode</p> <p>When this bit is set, the PBL value programmed in Bits[21:16] in DMA_CH(#i)_Tx_Control and Bits[21:16] in DMA_CH(#i)_Rx_Control is multiplied by eight times. Therefore, the DMA transfers the data in 8, 16, 32, 64, 128, and 256 beats depending on the PBL value.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): 8xPBL mode is disabled 0x1 (ENABLE): 8xPBL mode is enabled <p>Value After Reset: 0x0</p>
15:14	Reserved_15_14	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
13:0			Reserved Field: Yes

DMA_CH(#i)_Tx_Control (for i = 0; i <= 3)

Description: The DMA Channel/Transmit Control register controls the Tx features such as PBL, TCP segmentation, and Tx Channel weights.

Size: 32 bits

Offset: (0x0080*i)+0x1104

31	30:24	23	22	21:16	15	14:12	11:5	4	3:1	0
Reserved_31	Rsvd	Reserved_23	Rsvd	TxPBL	IPBL	Rsvd	Reserved_11_5	OSF	TCW	ST

Fields for Register: DMA_CH(#i)_Tx_Control (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31	Reserved_31	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
30:24			Reserved Field: Yes

Bits	Name	Memory Access	Description
23	Reserved_23	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
22			Reserved Field: Yes
21:16	TxPBL	R/W	<p>Transmit Programmable Burst Length</p> <p>These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior.</p> <p>To transfer more than 32 beats, perform the following steps:</p> <ol style="list-style-type: none"> 1. Set the 8xPBL mode in DMA_CH0_Control register. 2. Set the TxPBL. <p>Note: The maximum value of TxPBL must be less than or equal to half the Tx Queue size (TQS field of MTL_TxQ(#i)_Operation_Mode register) in terms of beats. This is required so that the Tx Queue has space to store at least another Tx PBL worth of data while the MTL Tx Queue Controller is transferring data to MAC. For example, in 64-bit data width configurations the total locations in Tx Queue of size 512 bytes is 64, TxPBL and 8xPBL needs to be programmed to less than or equal to 32.</p> <p>Value After Reset: 0x0</p>
15	IPBL	R/W	<p>Ignore PBL Requirement</p> <p>When this bit is set, the DMA does not check for PBL number of locations in the MTL before initiating a transfer. If space is not available, the MTL might use handshaking to slow the DMA.</p> <p>Note: This bit/mode must not be used when multiple Transmit DMA Channels are enabled as it might block other Transmit and Receive DMA Channels from accessing the Read Data Channel of AXI bus until space is available in Transmit Queue for current transfer.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Ignore PBL Requirement is disabled 0x1 (ENABLE): Ignore PBL Requirement is enabled <p>Value After Reset: 0x0</p>
14:12			Reserved Field: Yes
11:5	Reserved_11_5	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
4	OSF	R/W	Operate on Second Packet

Bits	Name	Memory Access	Description
			<p>When this bit is set, it instructs the DMA to process the second packet of the Transmit data even before the status for the first packet is obtained.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Operate on Second Packet disabled 0x1 (ENABLE): Operate on Second Packet enabled <p>Value After Reset: 0x0</p>
3:1	TCW	R/W	<p>Transmit Channel Weight</p> <p>This field indicates the weight assigned to the corresponding Transmit channel. When reset is complete, this field is set to 0 for all channels by default, resulting in equal weights to all channels.</p> <p>Value After Reset: 0x0</p>
0	ST	R/W	<p>Start or Stop Transmission Command</p> <p>When this bit is set, transmission is placed in the Running state. The DMA checks the Transmit list at the current position for a packet to be transmitted.</p> <p>The DMA tries to acquire descriptor from either of the following positions:</p> <ul style="list-style-type: none"> The current position in the list <p>This is the base address of the Transmit list set by the DMA_CH0_TxDesc_List_Address register.</p> <ul style="list-style-type: none"> The position at which the transmission was previously stopped <p>If the DMA does not own the current descriptor, the transmission enters the Suspended state and the TBU bit of the DMA_CH0_Status register is set. The Start Transmission command is effective only when the transmission is stopped. If the command is issued before setting the DMA_CH0_TxDesc_List_Address register, the DMA behavior is unpredictable.</p> <p>When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current packet. The Next Descriptor position in the Transmit list is saved, and it becomes the current position when the transmission is restarted. To change the list address, you need to program DMA_CH0_TxDesc_List_Address register with a new value when this bit is reset. The new value is considered when this bit is set again. The stop transmission command is effective only when the transmission of the current packet is complete or the transmission is in the Suspended state.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (STOP): Stop Transmission Command 0x1 (START): Start Transmission Command

Bits	Name	Memory Access	Description
			Value After Reset: 0x0 Testable: untestable

DMA_CH(#i)_Rx_Control (for i = 0; i <= 3)

Description: The DMA Channel/Receive Control register controls the Rx features such as PBL, buffer size, and extended status.

Size: 32 bits

Offset: (0x0080*i)+0x1108

31	30:28	27:24	23	22	21:16	15	14:3	2:1	0
RPF	Reserved_30_28	Rsvd	Reserved_23	Rsvd	RxPBL	Reserved_15	RBSZ_13_y	RBSZ_x_0	SR

Fields for Register: DMA_CH(#i)_Rx_Control (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31	RPF	R/W	<p>Rx Packet Flush.</p> <p>1: DWC_ether_qos automatically flushes the packet from the Rx Queues destined to this DMA Rx Channel, when it is stopped. When this bit remains set and the DMA is re-started by the software driver, the packets residing in the Rx Queues that were received when this RxDMA was stopped, are flushed out. The packets that are received by the MAC after the RxDMA is re-started are routed to the RxDMA. The flushing is done on the Read side of the Rx Queue.</p> <p>0: DWC_ether_qos does not flush the packet in the Rx Queue destined to this RxDMA Channel when it is in STOP state. This might cause head-of-line blocking in the corresponding RxQueue.</p> <p>Note: The stopping of packet flow from a Rx DMA Channel to the application by setting RPF works only when there is one-to-one mapping of Rx Queue to Rx DMA channels. In Dynamic mapping mode, setting RPF bit in any DMA_CH(#i)_Rx_Control register might flush packets from unintended Rx Queues which are destined to the stopped Rx DMA Channel.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Rx Packet Flush is disabled 0x1 (ENABLE): Rx Packet Flush is enabled <p>Value After Reset: 0x0</p>
30:28	Reserved_30_28	R	Reserved.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
27:24			Reserved Field: Yes
23	Reserved_23	R	Reserved. Value After Reset: 0x0
22			Reserved Field: Yes
21:16	RxPBL	R/W	<p>Receive Programmable Burst Length</p> <p>These bits indicate the maximum number of beats to be transferred in one DMA block data transfer. The DMA always attempts max burst as specified in PBL each time it starts a burst transfer on the application bus. You can program PBL with any of the following values: 1, 2, 4, 8, 16, or 32. Any other value results in undefined behavior.</p> <p>To transfer more than 32 beats, perform the following steps:</p> <ol style="list-style-type: none"> 1. Set the 8xPBL mode in the DMA_CH0_Control register. 2. Set the RxPBL. <p>Note: The maximum value of RxPBL must be less than or equal to half the Rx Queue size (RQS field of MTL_RxQ(#i)_Operation_Mode register) in terms of beats. This is required so that the Rx Queue has space to store at least another Rx PBL worth of data while the Rx DMA is transferring a block of data. For example, in 64-bit data width configurations the total locations in Rx Queue of size 512 bytes is 64, so RxPBL and 8xPBL needs to be programmed to less than or equal to 32.</p> <p>Value After Reset: 0x0</p>
15	Reserved_15	R	Reserved. Value After Reset: 0x0
14:3	RBSZ_13_y	R/W	<p>Receive Buffer size High</p> <p>RBSZ[13:0] is split into two fields higher RBSZ_13_y and lower RBSZ_x_0. The RBSZ[13:0] field indicates the size of the Rx buffers specified in bytes. The maximum buffer size is limited to 16K bytes. The buffer size is applicable to payload buffers when split headers are enabled.</p> <p>Note: The buffer size must be a multiple of 4, 8, or 16 depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). This is required even if the value of buffer address pointer is not aligned to data bus width. Hence the lower RBSZ_x_0 bits are read-only and the value is considered as all-zero. Thus the RBSZ_13_y indicates the buffer size in terms of locations (with the width same as bus-width).</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
2:1	RBSZ_x_0	R	<p>Receive Buffer size Low</p> <p>RBSZ[13:0] is split into two fields RBSZ_13_y and RBSZ_x_0. The RBSZ_x_0 is the lower field whose width is based on data bus width of the configuration.</p> <p>This field is of width 2, 3, or 4 bits for 32-bit, 64-bit, or 128-bit data bus width respectively. This field is read-only (RO).</p> <p>Value After Reset: 0x0</p>
0	SR	R/W	<p>Start or Stop Receive</p> <p>When this bit is set, the DMA tries to acquire the descriptor from the Receive list and processes the incoming packets.</p> <p>The DMA tries to acquire descriptor from either of the following positions:</p> <ul style="list-style-type: none"> The current position in the list <p>This is the address set by the DMA_CH0_RxDesc_List_Address register.</p> <ul style="list-style-type: none"> The position at which the Rx process was previously stopped <p>If the DMA does not own the current descriptor, the reception is suspended and the RBU bit of the DMA_CH0_Status register is set. The Start Receive command is effective only when the reception is stopped. If the command is issued before setting the DMA_CH0_RxDesc_List_Address register, the DMA behavior is unpredictable.</p> <p>When this bit is reset, the Rx DMA operation is stopped after the transfer of the current packet. The next descriptor position in the Receive list is saved, and it becomes the current position after the Rx process is restarted. The Stop Receive command is effective only when the Rx process is in the Running (waiting for Rx packet) or Suspended state.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (STOP): Stop Receive 0x1 (START): Start Receive <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

DMA_CH(#i)_TxDesc_List_HAddress (for i = 0; i <= 3)

Description: The Channel/Tx Descriptor List HAddress register has the higher 8 or 16 bits of the start address of the Transmit descriptor list.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in DMA_CH/Tx_Control register. When stopped, this register can be written with a new descriptor list

address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier. This register must be programmed with desired the higher address before programming the DMA_CH/TxDesc_List_Address register.

Size: 32 bits

Offset: (0x0080*i)+0x1110

31:y	15:0
Reserved_31_y	TDESHA

Fields for Register: DMA_CH(#i)_TxDesc_List_HAddress (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:16	Reserved_31_y	R	Reserved. Value After Reset: 0x0
15:0	TDESHA	R/W	Start of Transmit List This field contains the most-significant 8 or 16 bits of the 40- or 48-bit base address of the first descriptor in the Transmit descriptor list. Value After Reset: 0x0

DMA_CH(#i)_TxDesc_List_Address (for i = 0; i <= 3)

Description: The Channel/Tx Descriptor List Address register points the DMA to the start of Transmit descriptor list. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB to low.

You can write to this register only when the Tx DMA has stopped, that is, the ST bit is set to zero in DMA_CH0_Tx_Control register. When stopped, this register can be written with a new descriptor list address. When you set the ST bit to 1, the DMA takes the newly-programmed descriptor base address. If this register is not changed when the ST bit is set to 0, the DMA takes the descriptor address where it was stopped earlier.

Size: 32 bits

Offset: (0x0080*i)+0x1114

31:2	1:0
TDESLA	Reserved_LSb

Fields for Register: DMA_CH(#i)_TxDesc_List_Address (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:2	TDESLA	R/W	Start of Transmit List This field contains the base address of the first descriptor in the

Bits	Name	Memory Access	Description
			<p>Transmit descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO).</p> <p>The width of this field depends on the configuration:</p> <ul style="list-style-type: none"> 31:2 for 32-bit configuration 31:3 for 64-bit configuration 31:4 for 128-bit configuration <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
1:0	Reserved_LSb	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_RxDesc_List_HAddress (for i = 0; i <= 3)

Description: The Channel/Rx Descriptor List HAddress register has the higher 8 or 16 bits of the start address of the Receive descriptor list.

Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in the DMA_CH/_RxDesc_List_Address register

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

Size: 32 bits

Offset: (0x0080*i)+0x1118

31:y	15:0
Reserved_31_y	RDESHA

Fields for Register: DMA_CH(#i)_RxDesc_List_HAddress (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:16	Reserved_31_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
15:0	RDESHA	R/W	<p>Start of Receive List</p> <p>This field contains the most-significant 8 or 16 bits of the 40-bit or 48-bit base address of the first descriptor in the Rx Descriptor list.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_RxDesc_List_Address (for i = 0; i <= 3)

Description: The Channel/Rx Descriptor List Address register points the DMA to the start of Receive descriptor list.

This register points to the start of the Receive Descriptor List. The descriptor lists reside in the physical memory space of the application and must be Word, Dword, or Lword-aligned (for 32-bit, 64-bit, or 128-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given. You can write to this register only when Rx DMA has stopped, that is, SR bit is set to zero in DMA_CH0_Rx_Control register. When stopped, this register can be written with a new descriptor list address.

When you set the SR bit to 1, the DMA takes the newly programmed descriptor base address.

Size: 32 bits

Offset: (0x0080*i)+0x111C

31:2	1:0
RDESLA	Reserved_LSb

Fields for Register: DMA_CH(#i)_RxDesc_List_Address (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:2	RDESLA	R/W	<p>Start of Receive List</p> <p>This field contains the base address of the first descriptor in the Rx Descriptor list. The DMA ignores the LSB bits (1:0, 2:0, or 3:0) for 32-bit, 64-bit, or 128-bit bus width and internally takes these bits as all-zero. Therefore, these LSB bits are read-only (RO).</p> <p>The width of this field depends on the configuration:</p> <ul style="list-style-type: none"> 31:2 for 32-bit configuration 31:3 for 64-bit configuration 31:4 for 128-bit configuration <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
1:0	Reserved_LSb	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_TxDesc_Tail_Pointer (for i = 0; i <= 3)

Description: The Channel/Tx Descriptor Tail Pointer register points to an offset from the base and indicates the location of the last valid descriptor.

Size: 32 bits

Offset: (0x0080*i)+0x1120

31:2	1:0
------	-----

TDTP	Reserved_LSb
------	--------------

Fields for Register: DMA_CH(#i)_TxDesc_Tail_Pointer (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:2	TDTP	R/W	<p>Transmit Descriptor Tail Pointer</p> <p>This field contains the tail pointer for the Tx descriptor ring. The software writes the tail pointer to add more descriptors to the Tx channel. The hardware tries to transmit all packets referenced by the descriptors between the head and the tail pointer registers.</p> <p>The width of this field depends on the configuration:</p> <ul style="list-style-type: none"> 31:2 for 32-bit configuration 31:3 for 64-bit configuration 31:4 for 128-bit configuration <p>Value After Reset: 0x0</p>
1:0	Reserved_LSb	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_RxDesc_Tail_Pointer (for i = 0; i <= 3)

Description: The Channel/Rx Descriptor Tail Pointer Points to an offset from the base and indicates the location of the last valid descriptor.

Size: 32 bits

Offset: (0x0080*i)+0x1128

31:2	1:0
RDTP	Reserved_LSb

Fields for Register: DMA_CH(#i)_RxDesc_Tail_Pointer (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:2	RDTP	R/W	<p>Receive Descriptor Tail Pointer</p> <p>This field contains the tail pointer for the Rx descriptor ring. The software writes the tail pointer to add more descriptors to the Rx channel. The hardware tries to write all received packets to the descriptors referenced between the head and the tail pointer registers.</p> <p>The width of this field depends on the configuration:</p> <ul style="list-style-type: none"> 31:2 for 32-bit configuration 31:3 for 64-bit configuration 31:4 for 128-bit configuration

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
1:0	Reserved_LSb	R	Reserved. Value After Reset: 0x0

DMA_CH(#i)_TxDesc_Ring_Length (for i = 0; i <= 3)

Description: The Tx Descriptor Ring Length register contains the length of the Transmit descriptor ring.

Size: 32 bits

Offset: (0x0080*i)+0x112C

31:10	9:0
Reserved_31_10	TDRL

Fields for Register: DMA_CH(#i)_TxDesc_Ring_Length (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:10	Reserved_31_10	R	Reserved. Value After Reset: 0x0
9:0	TDRL	R/W	Transmit Descriptor Ring Length This field sets the maximum number of Tx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors. ESWIN recommends a minimum ring descriptor length of 4. For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9. Value After Reset: 0x0

DMA_CH(#i)_Rx_Control2 (for i = 0; i <= 3)

Description: The Channel/Receive Control register controls the Rx features such as Rx Descriptor Ring Length and Alternate Rx Buffer Size.

Size: 32 bits

Offset: (0x0080*i)+0x1130

31:24	23:16	15:10	9:0
Reserved_31_24	ARBS	Reserved_x_10	RDRL

Fields for Register: DMA_CH(#i)_Rx_Control2 (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:24	Reserved_31_24	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
23:16	ARBS	R/W	<p>Alternate Receive Buffer Size</p> <p>Indicates size in bytes for Buffer 1 when ARBS is programmed to a non-zero value (when split header feature is not enabled). When split header feature is enabled, ARBS indicates the buffer size for header data. The maximum alternate buffer is limited to 1020, 1016 or 1008-bytes depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively). When ARBS=0, Rx Buffer1 and Rx Buffer2 sizes are based on RBSZ field of DMA_CH(#i)_Rx_Control. Width of ARBS field is 8, 7 or 6-bits depending on the data bus widths (32-bit, 64-bit, or 128-bit respectively)</p> <p>Value After Reset: 0x0</p>
15:10	Reserved_x_10	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
9:0	RDRL	R/W	<p>Receive Descriptor Ring Length</p> <p>This register sets the maximum number of Rx descriptors in the circular descriptor ring. The maximum number of descriptors is limited to 1K descriptors.</p> <p>For example, You can program any value up to 0x3FF in this field. This field is 10 bits wide, if you program 0x3FF, you can have 1024 descriptors. If you want to have 10 descriptors, program it to a value of 0x9.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_Interrupt_Enable (for i = 0; i <= 3)

Description: The Channel/ Interrupt Enable register enables the interrupts reported by the Status register.

Size: 32 bits

Offset: (0x0080*i)+0x1134

31:16	15	14	13	12	11	10	9	8	7	6	5:3	2	1	0
Reserved_31_16	NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Reserved_5_3	TBUE	TXSE	TIE

Fields for Register: DMA_CH(#i)_Interrupt_Enable (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:16	Reserved_31_16	R	Reserved.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
15	NIE	R/W	<p>Normal Interrupt Summary Enable</p> <p>When this bit is set, the normal interrupt summary is enabled. This bit enables the following interrupts in the DMA_CH0_Status register:</p> <ul style="list-style-type: none"> Bit 0: Transmit Interrupt Bit 2: Transmit Buffer Unavailable Bit 6: Receive Interrupt Bit 11: Early Receive Interrupt <p>When this bit is reset, the normal interrupt summary is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Normal Interrupt Summary is disabled 0x1 (ENABLE): Normal Interrupt Summary is enabled <p>Value After Reset: 0x0</p>
14	AIE	R/W	<p>Abnormal Interrupt Summary Enable</p> <p>When this bit is set, the abnormal interrupt summary is enabled. This bit enables the following interrupts in the DMA_CH0_Status register:</p> <ul style="list-style-type: none"> Bit 1: Transmit Process Stopped Bit 7: Rx Buffer Unavailable Bit 8: Receive Process Stopped Bit 9: Receive Watchdog Timeout Bit 10: Early Transmit Interrupt Bit 12: Fatal Bus Error Bit 13: Context Descriptor Error <p>When this bit is reset, the abnormal interrupt summary is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Abnormal Interrupt Summary is disabled 0x1 (ENABLE): Abnormal Interrupt Summary is enabled <p>Value After Reset: 0x0</p>
13	CDEE	R/W	<p>Context Descriptor Error Enable</p> <p>When this bit is set along with the AIE bit, the Descriptor error interrupt is enabled. When this bit is reset, the Descriptor error interrupt is disabled.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x0 (DISABLE): Context Descriptor Error is disabled 0x1 (ENABLE): Context Descriptor Error is enabled</p> <p>Value After Reset: 0x0</p>
12	FBEE	R/W	<p>Fatal Bus Error Enable</p> <p>When this bit is set along with the AIE bit, the Fatal Bus error interrupt is enabled. When this bit is reset, the Fatal Bus Error error interrupt is disabled.</p> <p>Values:</p> <p>0x0 (DISABLE): Fatal Bus Error is disabled 0x1 (ENABLE): Fatal Bus Error is enabled</p> <p>Value After Reset: 0x0</p>
11	ERIE	R/W	<p>Early Receive Interrupt Enable</p> <p>When this bit is set along with the NIE bit, the Early Receive interrupt is enabled. When this bit is reset, the Early Receive interrupt is disabled.</p> <p>Values:</p> <p>0x0 (DISABLE): Early Receive Interrupt is disabled 0x1 (ENABLE): Early Receive Interrupt is enabled</p> <p>Value After Reset: 0x0</p>
10	ETIE	R/W	<p>Early Transmit Interrupt Enable</p> <p>When this bit is set along with the AIE bit, the Early Transmit interrupt is enabled. When this bit is reset, the Early Transmit interrupt is disabled.</p> <p>Values:</p> <p>0x0 (DISABLE): Early Transmit Interrupt is disabled 0x1 (ENABLE): Early Transmit Interrupt is enabled</p> <p>Value After Reset: 0x0</p>
9	RWTE	R/W	<p>Receive Watchdog Timeout Enable</p> <p>When this bit is set along with the AIE bit, the Receive Watchdog Timeout interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout interrupt is disabled.</p> <p>Values:</p> <p>0x0 (DISABLE): Receive Watchdog Timeout is disabled 0x1 (ENABLE): Receive Watchdog Timeout is enabled</p>

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
8	RSE	R/W	<p>Receive Stopped Enable</p> <p>When this bit is set along with the AIE bit, the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Stopped is disabled 0x1 (ENABLE): Receive Stopped is enabled <p>Value After Reset: 0x0</p>
7	RBUE	R/W	<p>Receive Buffer Unavailable Enable</p> <p>When this bit is set along with the AIE bit, the Receive Buffer Unavailable interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Buffer Unavailable is disabled 0x1 (ENABLE): Receive Buffer Unavailable is enabled <p>Value After Reset: 0x0</p>
6	RIE	R/W	<p>Receive Interrupt Enable</p> <p>When this bit is set along with the NIE bit, the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive Interrupt is disabled 0x1 (ENABLE): Receive Interrupt is enabled <p>Value After Reset: 0x0</p>
5:3	Reserved_5_3	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
2	TBUE	R/W	<p>Transmit Buffer Unavailable Enable</p> <p>When this bit is set along with the NIE bit, the Transmit Buffer Unavailable interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Transmit Buffer Unavailable is disabled 0x1 (ENABLE): Transmit Buffer Unavailable is enabled <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
1	TXSE	R/W	<p>Transmit Stopped Enable</p> <p>When this bit is set along with the AIE bit, the Transmission Stopped interrupt is enabled. When this bit is reset, the Transmission Stopped interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Transmit Stopped is disabled 0x1 (ENABLE): Transmit Stopped is enabled <p>Value After Reset: 0x0</p>
0	TIE	R/W	<p>Transmit Interrupt Enable</p> <p>When this bit is set along with the NIE bit, the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Transmit Interrupt is disabled 0x1 (ENABLE): Transmit Interrupt is enabled <p>Value After Reset: 0x0</p>

DMA_CH(#i)_Rx_Interrupt_Watchdog_Timer (for i = 0; i <= 3)

Description: The Receive Interrupt Watchdog Timer register indicates the watchdog timeout for Receive Interrupt (RI) from the DMA. When this register is written with a non-zero value, it enables the watchdog timer for the RI bit of the DMA_CH_Status register.

Size: 32 bits

Offset: (0x0080*i)+0x1138

31:18	17:16	15:8	7:0
Reserved_31_18	RWTU	Reserved_15_8	RWT

Fields for Register: DMA_CH(#i)_Rx_Interrupt_Watchdog_Timer (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:18	Reserved_31_18	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
17:16	RWTU	R/W	<p>Receive Interrupt Watchdog Timer Count Units</p> <p>This field indicates the number of system clock cycles corresponding to one unit in RWT field.</p> <p>2'b00: 256</p>

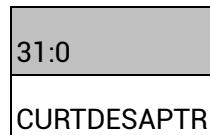
Bits	Name	Memory Access	Description
			<p>2'b01: 512 2'b10: 1024 2'b11: 2048</p> <p>For example, when RWT=2 and RWTU=1, the watchdog timer is set for $2 \times 512 = 1024$ system clock cycles.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
15:8	Reserved_15_8	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
7:0	RWT	R/W	<p>Receive Interrupt Watchdog Timer Count</p> <p>This field indicates the number of system clock cycles, multiplied by factor indicated in RWTU field, for which the watchdog timer is set.</p> <p>The watchdog timer is triggered with the programmed value after the Rx DMA completes the transfer of a packet for which the RI bit is not set in the DMA_CH(#i)_Status register, because of the setting of Interrupt Enable bit in the corresponding descriptor RDES3[30].</p> <p>When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per the Interrupt Enable bit RDES3[30] of any received packet.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

DMA_CH(#i)_Current_App_TxDesc (for i = 0; i <= 3)

Description: The Channel/ Current Application Transmit Descriptor register points to the current Transmit descriptor read by the DMA.

Size: 32 bits

Offset: (0x0080*i)+0x1144



Fields for Register: DMA_CH(#i)_Current_App_TxDesc (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:0	CURTDESAPTR	R	Application Transmit Descriptor Address Pointer

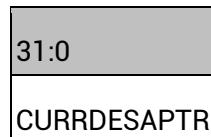
Bits	Name	Memory Access	Description
			<p>The DMA updates this pointer during Tx operation. This pointer is cleared on reset.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_Current_App_RxDesc (for i = 0; i <= 3)

Description: The Channel/ Current Application Receive Descriptor register points to the current Receive descriptor read by the DMA.

Size: 32 bits

Offset: (0x0080*i)+0x114C

**Fields for Register: DMA_CH(#i)_Current_App_RxDesc (for i = 0; i <= 3)**

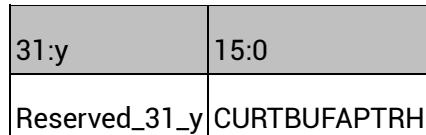
Bits	Name	Memory Access	Description
31:0	CURRDESAPTR	R	<p>Application Receive Descriptor Address Pointer</p> <p>The DMA updates this pointer during Rx operation. This pointer is cleared on reset.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_Current_App_TxBuffer_H (for i = 0; i <= 3)

Description: The Channel/Current Application Transmit Buffer Address High register has the higher 8 or 16 bits of the current address of the Transmit buffer address read by the DMA.

Size: 32 bits

Offset: (0x0080*i)+0x1150

**Fields for Register: DMA_CH(#i)_Current_App_TxBuffer_H (for i = 0; i <= 3)**

Bits	Name	Memory Access	Description
31:16	Reserved_31_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>

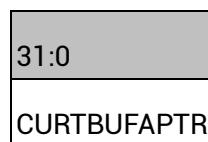
Bits	Name	Memory Access	Description
15:0	CURTBUFAPTRH	R	<p>Application Transmit Buffer Address Pointer</p> <p>The DMA updates this pointer during Tx operation. This pointer is cleared on reset.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_Current_App_TxBuffer (for i = 0; i <= 3)

Description: The Channel/Current Application Transmit Buffer Address register points to the current Tx buffer address read by the DMA.

Size: 32 bits

Offset: (0x0080*i)+0x1154

**Fields for Register: DMA_CH(#i)_Current_App_TxBuffer (for i = 0; i <= 3)**

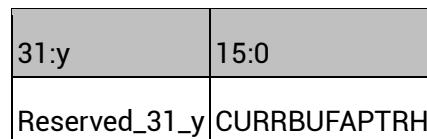
Bits	Name	Memory Access	Description
31:0	CURTBUFAPTR	R	<p>Application Transmit Buffer Address Pointer</p> <p>The DMA updates this pointer during Tx operation. This pointer is cleared on reset.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_Current_App_RxBuffer_H (for i = 0; i <= 3)

Description: The Channel/Current Application Receive Buffer Address High register has the higher 8 or 16 bits of the current address of the Receive buffer address read by the DMA.

Size: 32 bits

Offset: (0x0080*i)+0x1158

**Fields for Register: DMA_CH(#i)_Current_App_RxBuffer_H (for i = 0; i <= 3)**

Bits	Name	Memory Access	Description
31:16	Reserved_31_y	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>

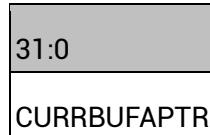
Bits	Name	Memory Access	Description
15:0	CURRBUFAPTRH	R	<p>Application Receive Buffer Address Pointer</p> <p>The DMA updates this pointer during Rx operation. This pointer is cleared on reset.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_Current_App_RxBuffer (for i = 0; i <= 3)

Description: The Channel 0 Current Application Receive Buffer Address register points to the current Rx buffer address read by the DMA.

Size: 32 bits

Offset: (0x0080*i)+0x115C

**Fields for Register: DMA_CH(#i)_Current_App_RxBuffer (for i = 0; i <= 3)**

Bits	Name	Memory Access	Description
31:0	CURRBUFAPTR	R	<p>Application Receive Buffer Address Pointer</p> <p>The DMA updates this pointer during Rx operation. This pointer is cleared on reset.</p> <p>Value After Reset: 0x0</p>

DMA_CH(#i)_Status (for i = 0; i <= 3)

Description: The software driver (application) reads the Status register during interrupt service routine or polling to determine the status of the DMA.

Note: The number of DMA_CH(#i)_Status register in the configuration is the higher of number of Rx DMA Channels and Tx DMA Channels.

Size: 32 bits

Offset: (0x0080*i)+0x1160

31:22	21:1 9	18:1 6	15	14	13	12	11	10	9	8	7	6	5:3	2	1	0
Reserved_31_2	REB	TEB	NI S	AI S	CD E	FB E	ER I	ET I	RW T	RP S	RB U	R I	Reserved_5_ 3	TB U	TP S	T I

Fields for Register: DMA_CH(#i)_Status (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:22	Reserved_31_22	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
21:19	REB	R	<p>Rx DMA Error Bits</p> <p>This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface.</p> <p>Bit 21</p> <ul style="list-style-type: none"> 1'b1: Error during data transfer by Rx DMA 1'b0: No Error during data transfer by Rx DMA <p>Bit 20</p> <ul style="list-style-type: none"> 1'b1: Error during descriptor access 1'b0: Error during data buffer access <p>Bit 19</p> <ul style="list-style-type: none"> 1'b1: Error during read transfer 1'b0: Error during write transfer <p>This field is valid only when the FBE bit is set. This field does not generate an interrupt.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
18:16	TEB	R	<p>Tx DMA Error Bits</p> <p>This field indicates the type of error that caused a Bus Error. For example, error response on the AHB or AXI interface.</p> <p>Bit 18</p> <ul style="list-style-type: none"> 1'b1: Error during data transfer by Tx DMA 1'b0: No Error during data transfer by Tx DMA <p>Bit 17</p> <ul style="list-style-type: none"> 1'b1: Error during descriptor access 1'b0: Error during data buffer access <p>Bit 16</p> <ul style="list-style-type: none"> 1'b1: Error during read transfer 1'b0: Error during write transfer <p>This field is valid only when the FBE bit is set. This field does not generate an interrupt.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

Bits	Name	Memory Access	Description
15	NIS	R/W	<p>Normal Interrupt Summary</p> <p>Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in the DMA_CH0_Interrupt_Enable register:</p> <ul style="list-style-type: none"> Bit 0: Transmit Interrupt Bit 2: Transmit Buffer Unavailable Bit 6: Receive Interrupt Bit 11: Early Receive Interrupt <p>Only unmasked bits (interrupts for which interrupt enable is set in DMA_CH0_Interrupt_Enable register) affect the Normal Interrupt Summary bit.</p> <p>This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit which causes NIS to be set is cleared.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Normal Interrupt Summary status not detected 0x1 (ACTIVE): Normal Interrupt Summary status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
14	AIS	R/W	<p>Abnormal Interrupt Summary</p> <p>Abnormal Interrupt Summary bit value is the logical OR of the following when the corresponding interrupt bits are enabled in the DMA_CH0_Interrupt_Enable register:</p> <ul style="list-style-type: none"> Bit 1: Transmit Process Stopped Bit 7: Receive Buffer Unavailable Bit 8: Receive Process Stopped Bit 10: Early Transmit Interrupt Bit 12: Fatal Bus Error Bit 13: Context Descriptor Error <p>Only unmasked bits affect the Abnormal Interrupt Summary bit.</p> <p>This is a sticky bit. You must clear this bit (by writing 1 to this bit) each time a corresponding bit, which causes AIS to be set, is cleared.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (INACTIVE): Abnormal Interrupt Summary status not detected 0x1 (ACTIVE): Abnormal Interrupt Summary status detected</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
13	CDE	R/W	<p>Context Descriptor Error</p> <p>This bit indicates that the DMA Tx/Rx engine received a descriptor error, which indicates invalid context in the middle of packet flow (intermediate descriptor) or all one's descriptor in Tx case and on Rx side it indicates DMA has read a descriptor with either of the buffer address as ones which is considered to be invalid.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <p>0x0 (INACTIVE): Context Descriptor Error status not detected 0x1 (ACTIVE): Context Descriptor Error status detected</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
12	FBE	R/W	<p>Fatal Bus Error</p> <p>This bit indicates that a bus error occurred (as described in the EB field). When this bit is set, the corresponding DMA channel engine disables all bus accesses.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <p>0x0 (INACTIVE): Fatal Bus Error status not detected 0x1 (ACTIVE): Fatal Bus Error status detected</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
11	ERI	R/W	<p>Early Receive Interrupt</p> <p>This bit when set indicates that the RxDMA has completed the transfer of packet data to the memory.</p> <p>In configurations supporting ERIC,</p> <p>ERIC = 0: This bit is set only after the Rx DMA has completely filled up a receive buffer with packet data.</p> <p>ERIC = 1: This bit is set after every burst transfer of data from the Rx DMA to the buffer.</p>

Bits	Name	Memory Access	Description
			<p>The setting of RI bit automatically clears this bit.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Early Receive Interrupt status not detected 0x1 (ACTIVE): Early Receive Interrupt status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
10	ETI	R/W	<p>Early Transmit Interrupt</p> <p>This bit when set indicates that the TxDMA has completed the transfer of packet data to the MTL TXFIFO memory.</p> <p>In configurations supporting ERIC:</p> <ul style="list-style-type: none"> ETIC = 0: This bit is set only after the Tx DMA has transferred a complete packet to MTL. ETIC = 1: This bit is set after completion of (partial) packet data transfer from buffers in the Transmit descriptor in which IOC = 1. <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Early Transmit Interrupt status not detected 0x1 (ACTIVE): Early Transmit Interrupt status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
9	RWT	R/W	<p>Receive Watchdog Timeout</p> <p>This bit is asserted when a packet with length greater than 2,048 bytes (10,240 bytes when Jumbo Packet mode is enabled) is received.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Receive Watchdog Timeout status not detected 0x1 (ACTIVE): Receive Watchdog Timeout status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
8	RPS	R/W	<p>Receive Process Stopped</p> <p>This bit is asserted when the Rx process enters the Stopped state.</p>

Bits	Name	Memory Access	Description
			<p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Receive Process Stopped status not detected 0x1 (ACTIVE): Receive Process Stopped status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
7	RBU	R/W	<p>Receive Buffer Unavailable</p> <p>This bit indicates that the application owns the next descriptor in the Receive list, and the DMA cannot acquire it. The Rx process is suspended. To resume processing Rx descriptors, the application should change the ownership of the descriptor and issue a Receive Poll Demand command. If this command is not issued, the Rx process resumes when the next recognized incoming packet is received. In ring mode, the application should advance the Receive Descriptor Tail Pointer register of a channel. This bit is set only when the DMA owns the previous Rx descriptor.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Receive Buffer Unavailable status not detected 0x1 (ACTIVE): Receive Buffer Unavailable status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
6	RI	R/W	<p>Receive Interrupt</p> <p>This bit indicates that the packet reception is complete. When packet reception is complete, Bit 31 of RDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor.</p> <p>The reception remains in the Running state.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Receive Interrupt status not detected 0x1 (ACTIVE): Receive Interrupt status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

Bits	Name	Memory Access	Description
5:3	Reserved_5_3	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
2	TBU	R/W	<p>Transmit Buffer Unavailable</p> <p>This bit indicates that the application owns the next descriptor in the Transmit list, and the DMA cannot acquire it. Transmission is suspended. The TPS0 field of the DMA_Debug_Status0 register explains the Transmit Process state transitions.</p> <p>To resume processing the Transmit descriptors, the application should do the following:</p> <ul style="list-style-type: none"> Change the ownership of the descriptor by setting Bit 31 of TDES3. Issue a Transmit Poll Demand command. <p>For ring mode, the application should advance the Transmit Descriptor Tail Pointer register of a channel.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Transmit Buffer Unavailable status not detected 0x1 (ACTIVE): Transmit Buffer Unavailable status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
1	TPS	R/W	<p>Transmit Process Stopped</p> <p>This bit is set when the transmission is stopped.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Transmit Process Stopped status not detected 0x1 (ACTIVE): Transmit Process Stopped status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>
0	TI	R/W	<p>Transmit Interrupt</p> <p>This bit indicates that the packet transmission is complete. When transmission is complete, Bit 31 of TDES3 is reset in the last descriptor, and the specific packet status information is updated in the descriptor.</p> <p>Access restriction applies. Self-set to 1 on internal event. Setting 1 clears. Setting 0 has no effect.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Transmit Interrupt status not detected 0x1 (ACTIVE): Transmit Interrupt status detected <p>Value After Reset: 0x0</p> <p>Testable: untestable</p>

DMA_CH(#i)_Miss_Frame_Cnt (for i = 0; i <= 3)

Description: This register has the number of packet counter that got dropped by the DMA either due to Bus Error or due to programming RPF field in DMA_CH\${i}_Rx_Control register.

Size: 32 bits

Offset: (0x0080*i)+0x1164

31:16	15	14:11	10:0
Reserved_31_16	MFCO	Reserved_14_11	MFC

Fields for Register: DMA_CH(#i)_Miss_Frame_Cnt (for i = 0; i <= 3)

Bits	Name	Memory Access	Description
31:16	Reserved_31_16	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
15	MFCO	R	<p>Overflow status of the MFC Counter</p> <p>When this bit is set then the MFC counter does not get incremented further. The bit gets cleared when this register is read.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Miss Frame Counter overflow not occurred 0x1 (ACTIVE): Miss Frame Counter overflow occurred <p>Value After Reset: 0x0</p>
14:11	Reserved_14_11	R	<p>Reserved.</p> <p>Value After Reset: 0x0</p>
10:0	MFC	R	<p>Dropped Packet Counters</p> <p>This counter indicates the number of packet counters that are dropped by the DMA either because of bus error or because of programming RPF field in DMA_CH\${i}_Rx_Control register. The</p>

Bits	Name	Memory Access	Description
			<p>counter gets cleared when this register is read.</p> <p>Access restriction applies. Clears on read. Self-set to 1 on internal event.</p> <p>Value After Reset: 0x0</p>

10.3 USB

10.3.1 overview

The chip contains two identical usb,supports Super-speed(5Gb/s)、High-speed(480Mb/s)、Full-speed(12Mb/s) and Low-speed(1.5Mb/s) data transfer.USB controller supports standard USB2.0/3.0 protocol and DRD mode(Dual Role Device), can be configured as Host or Devcie during initialization. The main features are as followed.

10.3.1.1 General Feature

- Universal Serial Bus 3.0 Specification, Revision 1.0
- Universal Serial Bus Specification, Revision 2.0
- LPM protocol in USB2.0 and U0/U1/U2/U3 states for USB3.0
- Dynamic FIFO memory allocation fore endpoints
- Keep-Alive feature in LS mode and (micro-)SOFs in HS/FS modes
- Software controlled standard USB commands
- Interrupt moderation

10.3.1.2 USB3.0 Device Feature

- Up to 16 single directional endpoints; max of 8 active in endpoints
- Flexible endpoint configuration
- Simultaneous IN and OUT transfer support for multiple applications
- Stream-based bulk endpoints with controller automatically initiating data movement
- Isochronous endpoints with isochronous data in data buffers or external hardware FIFOs

10.3.1.3 USB Class-Specific Device Feature

- Stream support for UASP application
- Gathering of scattered packet to support Ethernet over USB
- Scheduling of multiple Ethernet packets without interrupt
- Variable FIFO buffer allocation
- For isochronous applications, scheduling of variable-length payloads for each microframe
- Microframe precise scheduling for isochronous applications
- Configurable endpoint type selection and dynamic FIFO allocation to facilitate multi-function/composite device implementation

10.3.1.4 xHCI Host Feature

- Support 64 devices
- Support 1 interrupter
- Support 1 USB2.0 port and 1 SuperSpeed port
- xHCI1.1 compatible
- Standard or open-source xHCI and class drivers
- Concurrent IN and OUT transfers to get the full 8 Gbps duplex throughput

10.3.2 Block Diagram

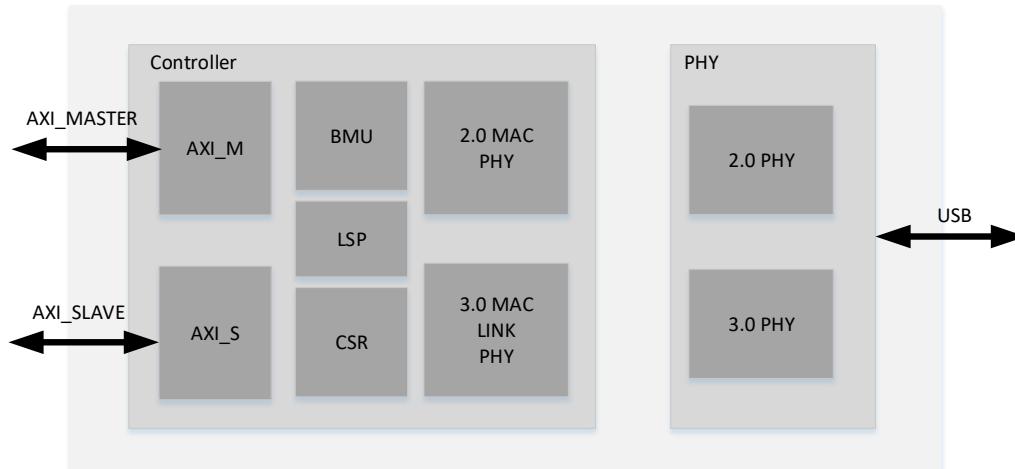


Figure 10-14 USB Diagram

The USB controller interacts with the system through the AXI bus, and mainly contains the following modules:

- BMU handles the buffering requirements between the system bus and USB bus
- LSP manages interaction between the USB bus, the system bus, and the driver
- MAC/LINK/PHY implements the standard USB protocol

10.3.3 Initialization

- Clock gating

USB0 bus clock can be disabled by setting USB0_CTRL [28] to 0. USB1 corresponding register is USB1_CTRL.

- Reset control

After power up, controller and phy is in reset mode by default, set USB0_CTRL [24] to 1 to release

controller reset, set USB0_CTRL [25] to 0 to release USB0 PHY reset.USB1 corresponding register is USB1_CTRL.

- Device mode initialization related registers

Register	Description
DCTL	Set the CSftRst field to '1' and wait for a read to return '0'. This resets the device controller.
GSBUSCFG0/1	Leave the default values if the correct power-on values were selected during configuration.
GTXTHRFCFG/ GRXTHRCFG	This is required only if you are planning to enable thresholding. Leave the default values if the correct power-on values were selected during configuration.
GUID	Optionally, the software can program the User ID GUID register.
GUSB2PHYCFG	Program the following PHY configuration fields: USBTrdTim, FSIntf, PHYIf, TOUTCal, or leave the default values if the correct power-on values were selected during configuration. Note: The PHY must not be enabled for auto-resume in device mode. Therefore, the field GUSB2PHYCFG[15] (ULPIAutoRes) must be written with '0' during the power-on initialization in case the reset value is '1'.
GUSB3PIPECTL	Program the following PHY configuration fields: DatWidth, PrtOpDir, or leave the default values if the correct power-on values were selected during configuration.
GTXFIFOSIZn	Write these registers to allocate prefetch buffers for each Tx endpoint. Unless the packet sizes of the endpoints are application-specific, it is recommended to use the default value.
GRXFIFOSIZ0	Write this register to allocate the receive buffer for all endpoints. Unless the packet sizes of the endpoints are application-specific, it is recommended to use the default value.
GEVNTADRx/ GEVNTSIZn/ GEVNTCOUNTn	Depending on the number of interrupts allocated, program the Event Buffer Address and Size registers to point to the Event Buffer locations in system memory, the sizes of the buffers, and unmask the interrupt. Note: USB operation stops if the Event Buffer memory is insufficient, because the controller stops receiving/transmitting packets.
GCTL	Program this register to override scaledown, RAM clock select, and clock gating parameters
DCFG	Program device speed and periodic frame interval
DEVTEN	At a minimum, enable USB Reset, Connection Done, and USB/Link State Change events.
DEPCMD0	Issue a DEPSTARTCFG command with DEPCMD0.XferRscldx set to 0 and CmdIOC set to 0 to initialize the transfer resource allocation. Poll CmdAct for completion.
DEPCMD0/ DEPCMD1	Issue a DEPCFG command for physical endpoints 0 & 1 with the following characteristics, and poll CmdAct for completions: <ul style="list-style-type: none"> ■ USB Endpoint Number = 0 or 1 (for physical endpoint 0 or 1) ■ FIFONum= 0 ■ XferNRdyEn and XferCmplEn = 1 ■ Maximum Packet Size = 512 ■ Burst Size = 0 ■ EPType = 2'b00 (Control) Note: The command has to be issued for EP0 first, followed by EP1.
DEPCMD0/ DEPCMD1	Issue a DEPXFERCFG command for physical endpoints 0 & 1 with DEPCMDPAR0_0/1 set to 1, and poll CmdAct for completions Note: The command has to be issued for EP0 first, followed by EP1.
DEPCMD0	Prepare a buffer for a setup packet, initialize a setup TRB, and issue a DEPSTRTRXFER command for physical endpoint 0, pointing to the setup TRB. Poll CmdAct for completion. Note: The controller attempts to fetch the setup TRB via the master interface after this command completes.

Register	Description
DALEPENA	Enable physical endpoints 0 & 1 by writing 0x3 to this register.
DCTL	Set DCTL.RunStop to '1' to allow the device to attach to the host. At this point, the device is ready to receive SOF packets, respond to control transfers on control endpoint 0, and generate events.

- To initialize the controller as host, the application must perform the steps described in the xHCI specification. If intermediate firmware needs to modify default settings, it can write to the following specific registers at power-on.

Register	Description
GSBUSCFG0/1	Leave the default values if the correct power-on values were selected during configuration.
GTXTHRCFG/ GRXTHRCFG	This is required only if you are planning to enable thresholding. Leave the default values if the correct power-on values were selected during configuration.
GUID	Optionally, the software can program the User ID GUID register.
GUSB2PHYCFG	Program the following PHY configuration fields: USBTrdTim, FSIntf, PHYIf, TOUTCal, or leave the default values if the correct power-on values were selected during configuration.
GUSB3PIPECTL	Program the following PHY configuration fields: DatWidth, PrtOpDir, or leave the default values if the correct power-on values were selected during configuration.
GTXFIFOSIZn/ GRXFIFOSIZn	Program these registers based on the speed used for the FIFO. Unless the packet sizes of the endpoints are application-specific, it is recommended that you use the default value.
GCTL	Program this register to override scaledown, RAM clock select, and clock gating parameters. It is recommended that you set this to 32'h2001004.
GUCTL	If you want to improve the interoperability with different devices, program this register to override the behavior of the controller in Host mode.
DCTL	If you want to improve interoperability as a Debug Capability Target, program the InitU1/U2Ena and AcceptU1/U2Ena fields to '0' to disable U1/U2 entry.

10.3.4 Register Description

10.3.4.1 Register Overview

Local Register

Base_addr:5044_0000

CSR_NAME	OFFSET	ACCESS	RESET VALUE	SIGNALS
USB0_CTRL	0x800	R/W	32'h12010200	[28]:usb0 bus clk enable 1'b1:enable bus clk 1'b0:disable bus clk [25]:usb0 phy reset, high active [24]:usb0 vcc reset, low active [8]:usb0 axi slave endian sel, 1 for big endian, 0 for little endian [2]:Bypass the filter for utmisrp_bvalid

CSR_NAME	OFFSET	ACCESS	RESET VALUE	SIGNALS
				[1]: Bypass the filter for pipe3_PowerPresent all U3 ports [0]: Bypass the filter for utmiotg_vbusvalid all U2 ports 1'b0:bus filter enable 1'b1:bus filter bypass
USB0_AXI_LP_CTRL	0x808	R/W	32'h1_0001	[31:17]:Reserved. [16]:usb0_xs_csysreq, Not supported.This must be driven to 1. [15:1]:Reserved [0]: usb0_xm_csysreq, Not supported.This must be driven to 1.
USB0_FSEL	0x83c	R/W	32'h27	[24]:not use, set to 1'b0 [16]:not use, set to 1'b0 [5:0]:frequency parameter sel according to phy reference clk, this shall be set to 6'b101010 Ref_clk fsel 24M 101010 25M 000010 100M 100111
USB0_MPLL_CTRL	0x840	R/W	32'h0	[6:0]:mpll frequency multiplier control, this should be set to 7'b0000_0000
USB0_REF_SSC_CTRL	0x844	R/W	32'h1_0000	[31:29]:Reserved [28]:not use, set to 1'b0 [27:25]:Reserved [24]:not use, set to 1'b0 [23:21]:Reserved [20]:usb0_ref_ssp_en,must set to 1'b1 after reference clk is stable and then de-assert phy reset. [19:17]:Reserved [16]:usb0_ssc_en,must set to 1'b1 [15]:Reserved [14:12]:not use, set to 3'b0 [11:9]:Reserved [8:0]:not use, set to 9'b0
USB1_CTRL	0x900	R/W	32'h12010200	[28]:usb1 bus clk enable 1'b1:enable bus clk

CSR_NAME	OFFSET	ACCESS	RESET VALUE	SIGNALS
				<p>1'b0:disable bus clk [25]:usb1 phy reset, high active [24]:usb1 vcc reset, low active [8]:usb1 slave endian sel, 1 for big endian, 0 for little endian [2]:Bypass the filter for utmisrp_bvalid [1]: Bypass the filter for pipe3_PowerPresent all U3 ports [0]: Bypass the filter for utmiotg_vbusvalid all U2 ports 1'b0:bus filter enable 1'b1:bus filter bypass</p>
USB1_AXI_LP_CTRL	0x908	R/W	32'h1_0001	<p>[31:17]:Reserved. [16]:usb1_xs_csysreq, Not supported.This must be driven to 1. [15:1]:Reserved [0]: usb1_xm_csysreq, Not supported.This must be driven to 1.</p>
USB1_FSEL	0x93c	R/W	32'h27	<p>[24]:not use, set to 1'b0 [16]:not use, set to 1'b0 [5:0]:frequency parameter sel according to phy reference clk, this shall be set to 6'b101010 Ref_clk fsel 24M 101010 25M 000010 100M 100111</p>
USB1_MPLL_CTRL	0x940	R/W	32'h0	[6:0]:mpll frequency multiplier control, this should be set to 7'b0000_0000
USB1_REF_SSC_CTRL	0x944	R/W	32'h1_0000	<p>[31:29]:Reserved [28]:not use, set to 1'b0 [27:25]:Reserved [24]:not use, set to 1'b0 [23:21]:Reserved [20]:usb1_ref_ssp_en,must set to 1'b1 after reference clk is stable and then de-assert phy reset. [19:17]:Reserved</p>

CSR_NAME	OFFSET	ACCESS	RESET VALUE	SIGNALS
				[16]:usb1_ssc_en,must set to 1'b1 [15]:Reserved [14:12]:not use, set to 3'b0 [11:9]:Reserved [8:0]:not use, set to 9'b0
USB0_QOS	0x1018	R/W	32'h0	[31:8]:Reserved [7:4]:usb0_sec_awqos, usb0 axi master awqos configuration [3:0]:usb0_sec_arqos, usb0 axi master arqos configuration
USB1_QOS	0x101c	R/W	32'h0	[31:8]:Reserved [7:4]:usb1_sec_awqos, usb1 axi master awqos configuration [3:0]:usb1_sec_arqos, usb1 axi master arqos configuration
USB0_MMU_CFG	0x1044	R/W	32'h0	[31:24]:usb0_awmmusid [23:16]:usb0_awmmussid [15:8]:usb0_armmuid [7:0]:usb0_armmussid
USB1_MMU_CFG	0x1048	R/W	32'h0	[31:24]:usb1_awmmusid [23:16]:usb1_awmmussid [15:8]:usb1_armmuid [7:0]:usb1_armmussid

USB Global Register

The two USB controller registers are identical and only need to be accessed according to different base address.

CSR_NAME	OFFSET	DESCRIPTION
USB_GSBUSCFG0	0xc100	configures system bus DMA options for the master bus
USB_GSBUSCFG1	0xc104	configures system bus DMA options for the master bus
USB_GTXTHRCFG	0xc108	Tx Threshold Control Register
USB_GRXTHRCFG	0xc10c	Rx Threshold Control Register
USB_GCTL	0xc110	Core Control Register
USB_GPMSTS	0xc114	Power Management Status Register for debug
USB_GSTS	0xc118	global status register

CSR_NAME	OFFSET	DESCRIPTION
USB_GUCTL1	0xc11c	Global User Control Register 1
USB_GUID	0xc128	Global User ID
USB_GUCTL	0xc12c	Global User Control Register
USB_GBUSEERRADDRLO	0xc130	SoC Bus Error Address Register - Low
USB_GBUSEERRADDRHI	0xc134	SoC Bus Error Address Register - High
USB_GPRTBIMAPLO	0xc138	SS Port to Bus Instance Mapping Register - Low
USB_GPRTBIMAPHI	0xc13c	SS Port to Bus Instance Mapping Register - High
USB_GPRTBIMAPHSLO	0xc180	High-Speed Port to Bus Instance Mapping Register - Low
USB_GPRTBIMAPHSHI	0xc184	High-Speed Port to Bus Instance Mapping Register - High
USB_GPRTBIMAPFSLO	0xc188	Full-Speed Port to Bus Instance Mapping Register - Low
USB_GPRTBIMAPFSHI	0xc18c	Full-Speed Port to Bus Instance Mapping Register - High
USB_GUCTL2	0xc19c	Global User Control Register 2
USB_GUSB2PHYCFG	0xc200	USB2.0 PHY Configuration Register
USB_GUSB3PIPECTL	0xc2c0	USB 3.0 PHY and PIPE Control Register
USB_GTXFIFOSIZ	0xc300	This register specifies the RAM start address and depth for each implemented TxFIFO
USB_GRXFIFOSIZ	0xc380	This register specifies the RAM start address and depth for each implemented RxFIFO device mode requires only one RxFIFO.
USB_GEVNTADRLO	0xc400	Event Buffer Address (Low) Register
USB_GEVNTADRHI	0xc404	Event Buffer Address (High) Register
USB_GEVNTSIZ	0xc408	Event Buffer Size and the Event Interrupt Mask Register
USB_GEVNTCOUNT	0xc40c	Event Buffer Count Register
USB_GUCTL3	0xc60c	Global User Control Register 3
USB_GTXFIFOPRIDEV	0xc610	Device TX FIFO DMA Priority Register the device mode uses only one RXFIFO, there is no Device RXFIFO DMA Priority Register
USB_GTXFIFOPRIHST	0xc618	Host TX FIFO DMA Priority Register
USB_GRXFIFOPRIHST	0xc61c	Host RX FIFO DMA Priority Register
USB_GDMAHLRATIO	0xc624	Host FIFO DMA High-Low Priority Ratio Register
USB_GFLADJ	0xc630	Frame Length Adjustment Register
USB_GUSB2RHBCTL	0xc640	USB 2.0 Root Hub Control Register
USB_DCFG	0xc700	configures the controller in Device mode after power-on or after certain control commands or enumeration
USB_DCTL	0xc704	Device Control Register

CSR_NAME	OFFSET	DESCRIPTION
USB_DEVTEN	0xc708	Device Event Enable Register
USB_DSTS	0xc70c	Device Status Register
USB_DGCMDPAR	0xc710	Device Generic Command Parameter Register
USB_DGCMD	0xc714	Device Generic Command Register
USB_DALEPENA	0xc720	Device Active USB Endpoint Enable Register
USB_DEPCMDPAR2[0:15]	0xc800+(i*0x10)	Device Physical Endpoint-n Command Parameter 2 Register i=0~15
USB_DEPCMDPAR1[0:15]	0xc804+(i*0x10)	Device Physical Endpoint-n Command Parameter 1 Register i=0~15
USB_DEPCMDPAR0[0:15]	0xc808+(i*0x10)	Device Physical Endpoint-n Command Parameter 0 Register i=0~15
USB_DEPCMD[0:15]	0xc80c+(i*0x10)	Device Physical Endpoint-n Command Register i=0~15
USB_DEV_IMOD	0xca00	Device Interrupt Moderation Register

10.3.4.2 Register Detail

USB_GSBUSCFG0

- **Description:** Global SoC Bus Configuration Register 0. This register configures system bus DMA options for the master bus(AXI). Options include burst length and cache type (bufferable/posted, cacheable/snoop, and so on). The application can program this register upon power-on, or a change in mode of operation after the DMA engine is halted.
- **xHCI Register Power-On Value:** If you are using a standard xHCI host driver, make sure to set the register's power-on value because the standard xHCI driver does not access this register.
- **Size:** 32 bits
- **Offset:** 0xc100

Bits	Name	Memory Access	Description
31:28	DATRDREQINFO	R/W	<p>DATRDREQINFO</p> <p>AXI-cache ReqInfo for Data Read (DatRdReqInfo)</p> <p>Input to BUS-GM.</p> <p>Value After Reset: 0x0</p>
27:24	DESRDREQINFO	R/W	<p>DESRDREQINFO</p> <p>AXI-cache ReqInfo for Descriptor Read (DesRdReqInfo).</p> <p>Input to BUS-GM.</p> <p>Value After Reset: 0x0</p>
23:20	DATWRREQINFO	R/W	<p>DATWRREQINFO</p> <p>AXI-cache ReqInfo for Data Write (DatWrReqInfo).</p> <p>Input to BUS-GM.</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
19:16	DESWRREQINFO	R/W	<p>DESWRREQINFO</p> <p>AXI-cache ReqInfo for Descriptor Write (DesWrReqInfo)</p> <p>Input to BUS-GM.</p> <p>Value After Reset: 0x0</p>
15:12	reserved_15_12	R	<p>Reserved for future use</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xf</p> <p>WriteConstraint: writeAsRead</p>
11	DATBIGEND	R/W	<p>Data Access is Big Endian</p> <p>This bit controls the endian mode for data accesses.</p> <ul style="list-style-type: none"> • Little-endian (default); • Big-endian; <p>In big-endian mode, DMA access (both read and write) for packet data a Byte Invariant Big-Endian mode.</p> <p>Note: Since AXI requires byte invariant endianness, setting DescBigend and DatBigEnd to one causes an address invariant transform to be applied, which is not appropriate. See section 9.3 and 9.4 of the <i>AMBA AXI Specification</i>. Hence for an AXI master, this bit must be set to zero.</p> <p>Value After Reset: 0x0</p>
10	DESBIGEND	R/W	<p>Descriptor Access is Big Endian</p> <p>This bit controls the endian mode for descriptor accesses.</p> <ul style="list-style-type: none"> • Little-endian (default) • Big-endian <p>In big-endian mode, DMA access (both read and write) for descriptors uses a Byte Invariant Big-Endian mode (see "Little-Endian and Big-Endian" section in the User Guide).</p> <p>Data is considered as 'embedded data' in the descriptors in the following cases:</p> <ul style="list-style-type: none"> • Device mode: The buffer pointer of a Setup TRB points to the Setup TRB itself. • Host mode: The Immediate Data (IDT) bit in a Transfer TRB is set to 1. <p>In device mode, if the system uses different endian modes for descriptor and data, software must not use 'embedded' data.</p> <p>In host mode, if the system uses different endian modes for data and descriptors, the controller treats 'embedded data' as descriptor (not as data) in terms of endian mode handling. If this is not the expectation of the system, the software must manipulate the 'embedded data' accordingly.</p> <p>Note: Since AXI requires byte invariant endianness, setting DescBigend and DatBigEnd to one causes an address invariant transform to be applied, which is not appropriate. See section 9.3 and 9.4 of the AMBA AXI Specification. Hence</p>

Bits	Name	Memory Access	Description
			<p>for an AXI master, this bit must be set to zero.</p> <p>Value After Reset: 0x0</p>
9:8	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x3</p> <p>WriteConstraint: writeAsRead</p>
7	INCR256BRSTENA	R/W	<p>INCR256 Burst Type Enable</p> <p>Input to BUS-GM.</p> <p>For the AXI configuration, if software set this bit to 1, the AXI master uses INCR to do the 256-beat burst.</p> <p>Value After Reset: 0x0</p>
6	INCR128BRSTENA	R/W	<p>INCR128 Burst Type Enable</p> <p>Input to BUS-GM;</p> <p>For the AXI configuration, if software set this bit to 1, the AXI master uses INCR to do the 128-beat burst.</p> <p>Value After Reset: 0x0</p>
5	INCR64BRSTENA	R/W	<p>INCR64 Burst Type Enable</p> <p>Input to BUS-GM;</p> <p>For the AXI configuration, if software set this bit to 1, the AXI master uses INCR to do the 64-beat burst.</p> <p>Value After Reset: 0x0</p>
4	INCR32BRSTENA	R/W	<p>INCR32 Burst Type Enable</p> <p>Input to BUS-GM; For the AXI configuration, if software set this bit to 1, the AXI master uses INCR to do the 32-beat burst.</p> <p>Value After Reset: 0x0</p>
3	INCR16BRSTENA	R/W	<p>INCR16 Burst Type Enable</p> <p>Input to BUS-GM. For the AXI configuration, if software set this bit to '1', the AXI master uses INCR to do the 16-beat burst.</p> <p>Value After Reset: 0x0</p>
2	INCR8BRSTENA	R/W	<p>INCR8 Burst Type Enable</p> <p>Input to BUS-GM; For the AXI configuration, if software set this bit to "1", the AXI master uses INCR to do the 8-beat burst.</p> <p>Value After Reset: 0x0</p>
1	INCR4BRSTENA	R/W	<p>INCR4 Burst Type Enable</p> <p>Input to BUS-GM; For the AXI configuration, when this bit is enabled the controller is allowed to do bursts of beat length 1, 2, and 4. It is highly recommended that this bit is enabled to prevent descriptor reads and writes</p>

Bits	Name	Memory Access	Description
			<p>from being broken up into separate transfers.</p> <p>Value After Reset: 0x0</p>
0	INCRBRSTENA	R/W	<p>Undefined Length INCR Burst Type Enable (INCRBrstEna)</p> <p>Input to BUS-GM; This bit determines the set of burst lengths the master interface uses. It works in conjunction with the USB_GSBUSCFG0[7:1] enables (INCR256/128/64/32/16/8/4).</p> <p><i>0: INCRX burst mode</i></p> <p>ARLEN/AWLEN do not use INCR except in case of non-aligned burst transfers. In the case of address-aligned transfers, they use only the following burst lengths:</p> <ul style="list-style-type: none"> • 1 • 2, 4 (if GSBUSCFG0.INCR4BrstEna = 1) • 8 (if GSBUSCFG0.INCR8BrstEna = 1) • 16 (if GSBUSCFG0.INCR16BrstEna = 1) • 32 (if GSBUSCFG0.INCR32BrstEna = 1) • 64 (if GSBUSCFG0.INCR64BrstEna = 1) • 128 (if GSBUSCFG0.INCR128BrstEna = 1) • 256 (if GSBUSCFG0.INCR256BrstEna = 1) <p>Note:</p> <ul style="list-style-type: none"> • In case of non-address-aligned transfers, INCR may get generated at the beginning and end of the transfers to align the address boundaries, even though INCR is disabled. • In AHB mode, if INCRX burst mode is enabled, but none of the supported INCRx bursts bits are enabled, then the controller will perform (undefined length) INCR bursts. <p><i>1: INCR (undefined length) burst mode</i></p> <ul style="list-style-type: none"> • AHB configurations: HBURST uses SINGLE or INCR of any length with handling 1KB boundary breakup. • AXI configurations: ARLEN/AWLEN uses any length less than or equal to the largest-enabled burst length of INCR32/64/128/256. <p>For cache line-aligned applications, this bit is typically set to 0 to ensure that the master interface uses only power-of-2 burst lengths (as enabled via USB_GSBUSCFG0[7:0]).</p> <p>Value After Reset: 0x1</p>

USB_CSBUSCFG1

- **Description:** Global SoC Bus Configuration Register 1
- xHCI Register Power-On Value:
- If you are using a standard xHCI host driver, make sure to set the register's power-on value because the standard xHCI driver does not access this register.
- **Size:** 32 bits
- **Offset:** 0xc104

Bits	Name	Memory Access	Description
31:13	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x7fff</p> <p>WriteConstraint: writeAsRead</p>
12	EN1KPAGE	R/W	<p>1k Page Boundary Enable</p> <p>By default (this bit is disabled) the AXI breaks transfers at the 4k page boundary. When this bit is enabled, the AXI master (DMA data) breaks transfers at the 1k page boundary.</p> <p>Value After Reset: 0x0</p>
11:8	PipeTransLimit	R/W	<p>AXI Pipelined Transfers Burst Request Limit</p> <p>The field controls the number of outstanding pipelined transfer requests the AXI master pushes to the AXI slave.</p> <p>When the AXI master reaches this limit, it does not make any more requests on the AXI ARADDR and AWADDR buses until the associated data phases complete.</p> <p>This field is encoded as follows:</p> <ul style="list-style-type: none"> • 'h0: 1 request • 'h1: 2 requests • 'h2: 3 requests • 'h3: 4 requests • ... • 'hF: 16 requests <p>Value After Reset: 0x3</p>
7:0	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xff</p> <p>WriteConstraint: writeAsRead</p>

USB_GTXTHRCFG

- **Description:** Global Tx Threshold Control Register
- **Size:** 32 bits
- **Offset:** 0xc108

Bits	Name	Memory Access	Description
31:30	reserved_31	R	<p>Reserved</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
			<p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
29	UsbTxPktCntSel	R/W	<p>USB Transmit Packet Count Enable</p> <p>This field enables/disables the USB transmission multi-packet thresholding:</p> <ul style="list-style-type: none"> • 0: USB transmission multi-packet thresholding is disabled; the controller can start transmission on the USB after the entire (one full) packet has been fetched into the corresponding TXFIFO. • 1: USB transmission multi-packet thresholding is enabled. The controller can only start transmission on the USB after USB Transmit Packet Count amount of packets for the USB transaction (burst) are already in the corresponding TXFIFO. This mode is valid in both host and device modes. It is only used for SuperSpeed operation. <p>Value After Reset: 0x0</p> <p>Testable: unconstrained</p>
28	reserved_28	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
27:24	UsbTxPktCnt	R/W	<p>USB Transmit Packet Count</p> <p>This field specifies the number of packets that must be in the TXFIFO before the controller can start transmission for the corresponding USB transaction (burst). This field is only valid when the USB Transmit Packet Count Enable field is set to one. Valid values are from 1 to 15.</p> <p>Note:</p> <ul style="list-style-type: none"> • In device mode, if device controller does not have the TRBs for the number of packets or if it cannot fetch the TRBs because of high latency or switching between other endpoints, then it does not wait for the threshold number of packets. The threshold number of packets will be honored only when the TRBs are available in the controller for the number of packets before it starts the data fetch. • This field must be less than or equal to the USB Maximum TX Burst Size field. <p>Value After Reset: 0x0</p> <p>Testable: unconstrained</p>
23:16	UsbMaxTxBurstSize	R/W	<p>USB Maximum TX Burst Size</p> <p>When UsbTxPktCntSel is one, this field specifies the Maximum Bulk OUT burst the controller can do. When the system bus is slower than the USB, TX FIFO can underrun during a long burst. User can program a smaller value to this field to limit the TX burst size that the controller can do.</p> <p>Host mode: It only applies to SS Bulk, Isochronous, and Interrupt OUT</p>

Bits	Name	Memory Access	Description
			<p>endpoints.</p> <p>Device mode: This value is not used in device mode, but users need to program a value when using the TX threshold feature to make sure that the value programmed in UsbTxPktCnt is less than this value.</p> <p>Valid values are from 1 to 16.</p> <p>Value After Reset: 0x0</p> <p>Testable: unconstrained</p>
15	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
14	reserved	R	<p>Reserved1(Rsvd/Rs)</p> <p>Register field must write only 0 by the application. The read value must be treated as X (unknown).</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
13:11	reserved	R	<p>Reserved (Rsvd/Rs)</p> <p>The register field must write only 0 by the application. The read value must be treated as X (unknown).</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x7</p> <p>WriteConstraint: writeAsRead</p>
10:0	reserved	R	<p>Reserved for future use</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x7ff</p> <p>WriteConstraint: writeAsRead</p>

USB_GRXTHRCFG

- **Description:** Global Rx Threshold Control Register. In a normal case, a Tx burst starts as soon as one packet is prefetched; an Rx burst starts as soon as 1-packet space is available. This works well as long as the system bus is faster than the USB 3.0 bus (a 1024-bytes packet takes ~2.2 microseconds on the USB bus in SS mode). If the system bus latency is larger than 2.2 microseconds to access a 1024-byte packet, then starting a burst on 1-packet condition leads to an early abort of the burst causing unnecessary performance reduction. To avoid underrun and overrun during the burst, in a high-latency bus system (like USB), threshold and burst size

control is provided through USB_GTXTHRCFG and USB_GRXTHRCFG registers. Bit [29] of the USB_GTXTHRCFG and USB_GRXTHRCFG registers enables this feature.

- Note:
- There is an issue when ACK TP with NumP=0 followed by ACK TP with NumP=1 without ERDY TP sent by the device controller during a burst bulk OUT transfer. This may cause third-party USB 3.0 host controllers to keep waiting for the ERDY TP.
- The USB 3.0 specification states that "When an endpoint is not in a flow control condition, it shall not send an ERDY TP unless the endpoint is a Bulk endpoint that supports streams." In this case, after the device sent the ACK TP (numP=1), the endpoint was not in the flow control, so it did not send an ERDY.
- The device would have sent ERDY if the next OUT packet was not received. When the next OUT packet was received, at that time there was enough buffer space to accept it, so the device accepted the packet by informing host that it is no longer in the flow control. The Host should wait for the responses for all the OUT packets to return and then decide if the endpoint is still in flow control or not.
- The USB 3.1 specification supersedes all the USB 3.0 specification. The errata states that "If the host continues, or resumes, transactions to an endpoint, the endpoint shall re-evaluate its flow control state and respond appropriately." However, there are no ECNs on the USB 3.0 for this issue.
- To work around this issue, the Global Rx Threshold mode must be disable by setting USB_GRXTHRCFG.UsbRxPktCntSel=0. Instead, software can program the USB_DCFG.NUMP mode (where fixed NUMP is transmitted always) instead of the RX threshold based numP mode to prevent the device from sending ACK TP with NumP=0. The NUMP in the ACK TP is the minimum value of (USB_DCFG.NUMP, bMaxBurstSize) for each endpoint.
- **Size:** 32 bits
- **Offset:** 0xc10c

Bits	Name	Memory Access	Description
31:30	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x3</p> <p>WriteConstraint: writeAsRead</p>
29	UsbRxPktCntSel	R/W	<p>USB Receive Packet Count Enable</p> <p>This field enables/disables the USB reception multi-packet thresholding:</p> <ul style="list-style-type: none"> ● 0: The controller can only start reception on the USB when the RX FIFO has space for at least one packet. ● 1: The controller can only start reception on the USB when the RX FIFO has space for at least UsbRxPktCnt amount of packets. This mode is valid in both host and device mode. It is only used for SuperSpeed. <p>In device mode,</p> <ul style="list-style-type: none"> ● Setting this bit to 1 also enables the functionality of reporting NUMP in the ACK TP based on the RX FIFO space instead of reporting a fixed NUMP derived from DCFG.NUMP for non-control endpoints. ● If you are using external buffer control (EBC) feature, disable this mode by setting UsbRxPktCntSel to 0. <p>Value After Reset: 0x0</p>
28	reserved	R	Reserved

Bits	Name	Memory Access	Description
			<p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
27:24	UsbRxPktCnt	R/W	<p>USB Receive Packet Count</p> <p>In host mode, this field specifies the space (in terms of the number of packets) that must be available in the RX FIFO before the controller can start the corresponding USB RX transaction (burst).</p> <p>In device mode, this field specifies the space (in terms of the number of packets) that must be available in the RX FIFO before the controller can send ERDY for a flow-controlled endpoint.</p> <p>This field is valid only when the USB Receive Packet Count Enable field is set to 1. The valid values for this field are from 1 to 15.</p> <p>Note: This field must be less than or equal to the USB Maximum Receive Burst Size field.</p> <p>Value After Reset: 0x0</p>
23:19	UsbMaxRxBurstSize	R/W	<p>USB Maximum Receive Burst Size</p> <p>In host mode, this field specifies the Maximum Bulk IN burst the usb controller can perform.</p> <p>When the system bus is slower than the USB, RX FIFO can overrun during a long burst.</p> <p>You can program a smaller value to this field to limit the RX burst size that the controller can perform. It only applies to SS Bulk, Isochronous, and Interrupt IN endpoints in the host mode.</p> <p>In device mode, this field specifies the NUMP value that is sent in ERDY for an OUT endpoint. The programmed value should not exceed the RXFIFO size.</p> <p>This field is valid only when UsbRxPktCntSel is one. The valid values for this field are from 1 to 16.</p> <p>Value After Reset: 0x0</p>
18:13	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x3f</p> <p>WriteConstraint: writeAsRead</p>
12:0	ResvISOCOUTSpc	R/W	<p>Space reserved in Rx FIFO for ISOC OUT</p> <p>In host mode, this field is not applicable and must be programmed to 0.</p> <p>In device mode, this value represents the amount of space to be reserved for ISOC OUT packets.</p> <p>The value to be programmed should be chosen so as to ensure that non ISOC packets are not completely dropped.</p> <p>If no space needs to be reserved for ISOC OUT packets, program this field to 0.</p>

Bits	Name	Memory Access	Description
			<p>This field is valid only in device mode. The maximum configurable depth of RX FIFO is 8192. Therefore, this field is 13 bits wide.</p> <p>The value of space reserved is in terms of 32.</p> <p>For SS, the space reservation is always rounded off to the nearest packet boundary. Therefore, it is always recommended to program a value corresponding to MPS or its multiples.</p> <p>For HS/FS, the space reservation is the actual value.</p> <p>Note: For SS, reserve space for ISOC when the Rx FIFO space can accommodate two MPS or more. Otherwise, this may result in degraded performance for non-ISOC packets. If the space is entirely allocated for ISOC, the non-ISOC packets will be completely dropped. To help you decide during the time of configuring the controller, refer to the "Device-Mode Receive Path" section in the Databook.</p> <p>Value After Reset: 0x0</p>

USB_GCTL

- **Description:** Global Core Control Register
- **Size:** 32 bits
- **Offset:** 0xc110

Bits	Name	Memory Access	Description
31:19	PWRDNSCALE	R/W	<p>Power Down Scale (PwrDnScale)</p> <p>The USB3 suspend_clk input replaces pipe3_rx_pclk as a clock source to a small part of the USB3 controller that operates when the SS PHY is in its lowest power (P3) state, and therefore does not provide a clock.</p> <p>The Power Down Scale field specifies how many suspend_clk periods fit into a 16 kHz clock period. When performing the division, round up the remainder.</p> <p>For example, when using an 32-bit PHY and 25-MHz Suspend clock,</p> <p>Power Down Scale = $25000 \text{ kHz} / 16 \text{ kHz} = 13'd1563$ (rounder up)</p> <p>Note:</p> <ul style="list-style-type: none"> • Minimum Suspend clock frequency is 32 kHz • Maximum Suspend clock frequency is 125 MHz <p>The LTSSM uses Suspend clock for 12-ms and 100-ms timers during suspend mode. According to the USB 3.0 specification, the accuracy on these timers is 0% to +50%.</p> <ul style="list-style-type: none"> • $12 \text{ ms} + 0\text{~}+50\% \text{ accuracy} = 18 \text{ ms}$ (Range is 12 ms - 18 ms) • $100 \text{ ms} + 0\text{~}+50\% \text{ accuracy} = 150 \text{ ms}$ (Range is 100 ms - 150 ms). <p>The suspend clock accuracy requirement is:</p> <ul style="list-style-type: none"> • $(12,000/62.5) * (\text{GCTL}[31:19]) * \text{actual suspend_clk_period}$ must be between 12,000 and 18,000 • $(100,000/62.5) * (\text{GCTL}[31:19]) * \text{actual suspend_clk_period}$ must be between 100,000 and 150,000 <p>For example, if your suspend_clk frequency varies from 7.5 MHz to 10.5MHz,</p>

Bits	Name	Memory Access	Description
			<p>then the value needs to programmed is:</p> <p>Power Down Scale = $10500/16 = 657$ (rounded up; and fastest frequency used).</p> <p>Value After Reset: 0x3</p>
18	MASTERFILTBYPASS	R/W	<p>Master Filter Bypass</p> <p>When this bit is set to 1'b1, irrespective of the parameter `DWC_USB3_EN_BUS_FILTERS chosen, all the filters in the DWC_usb3_filter module are bypassed.</p> <p>The double synchronizers to mac_clk preceding the filters are also bypassed. For enabling the filters, this bit must be 1'b0.</p> <p>Value After Reset: 0x0</p>
17	BYPSETADDR	R/W	<p>Bypass SetAddress in Device Mode.</p> <p>When BYPSETADDR bit is set, the device controller uses the value in the DCFG[DevAddr] bits directly for comparing the device address in the tokens.</p> <p>For simulation, you can use this feature to avoid sending an actual SET ADDRESS control transfer on the USB, and make the device controller respond to a new address.</p> <p>When the xHCI Debug capability is enabled and this bit is set, the Debug Target immediately enters the configured state without requiring the Debug Host to send a SetAddress or SetConfig request.</p> <p>Note: You can set this bit for simulation purposes only. In the actual hardware, this bit must be set to 1'b0.</p> <p>Value After Reset: 0x0</p>
16	U2RSTECN	R/W	<p>U2RSTECN</p> <p>If the SuperSpeed connection fails during POLL or LMP exchange, the device connects at non-SS mode.</p> <p>If this bit is set, then device attempts three more times to connect at SS, even if it previously failed to operate in SS mode. For each attempt, the device checks receiver termination eight times.</p> <p>From 2.60a release, this bit controls whether to check for Rx.Detect eight times or one time for every attempt. Device controller on USB 2.0 reset checks for receiver termination eight times per attempt if this bit is set to zero, or only once per attempt if the bit is set to one.</p> <p>Note: This bit is applicable only in device mode.</p> <p>Value After Reset: 0x1</p>
15:14	FRMSCLDW	R/W	<p>FRMSCLDW</p> <p>This field scales down device view of a SOF/USOF/ITP duration.</p> <p>For SS/HS mode:</p> <ul style="list-style-type: none"> • Value of 2'h3 implements interval to be 15.625 us • Value of 2'h2 implements interval to be 31.25 us • Value of 2'h1 implements interval to be 62.5 us • Value of 2'h0 implements interval to be 125us

Bits	Name	Memory Access	Description
			<p>For FS mode, the scale-down value is multiplied by 8.</p> <p>When xHCI Debug Capability is enabled, this field also scales down the MaxPacketSize of the IN and OUT bulk endpoint to allow more traffic during simulation. It can only be changed from a non-zero value during simulation.</p> <ul style="list-style-type: none"> • 2'h0: 1024 bytes • 2'h1: 512 bytes • 2'h2: 256 bytes • 2'h3: 128 bytes <p>Value After Reset: 0x0</p>
13:12	PRTCAPDIR	R/W	<p>PRTCAPDIR: Port Capability Direction (PrtCapDir)</p> <ul style="list-style-type: none"> • 2'b01: for Host configurations • 2'b10: for Device configurations <p>Note: For static Host-only/Device-only applications, use DRD Host or DRD Device mode. The combination of GCTL.PrtCapDir=2'b11 with SRP and HNP/RSP disabled is not recommended for these applications.</p> <p>The sequence for switching modes in DRD configuration is as follows:</p> <p><i>Switching from Device to Host:</i></p> <ol style="list-style-type: none"> 1. Reset the controller using GCTL[11] (CoreSoftReset). 2. Set GCTL[13:12] (PrtCapDir) to 2'b01 (Host mode). 3. Reset the host using USBCMD.HCRESET. 4. Follow the steps in "Initializing Host Registers" section of the Programming Guide. <p><i>Switching from Host to Device:</i></p> <ol style="list-style-type: none"> 1. Reset the controller using GCTL[11] (CoreSoftReset). 2. Set GCTL[13:12] (PrtCapDir) to 2'b10 (Device mode). 3. Reset the device by setting DCTL[30] (CSftRst). 4. Follow the steps in "Register Initialization" section of the Programming Guide. <p>Programming this field with random data causes the controller to keep toggling between the host mode and the device mode . Bit Bash register testing is not recommended.</p> <p>Value After Reset: 0x2</p> <p>Testable: writeAsRead</p>
11	CORESOFTRESET	R/W	<p>Core Soft Reset (CoreSoftReset)</p> <ul style="list-style-type: none"> • 1'b0 - No soft reset • 1'b1 - Soft reset to controller <p>Clears the interrupts and all the CSRs except the following registers:</p> <ul style="list-style-type: none"> • GCTL

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> • GUCTL • GSTS • GSNPSID • GGPIO • GUID • GUSB2PHYCFGn registers • GUSB3PIPECTLn registers • DCFG • DCTL • DEVTEM • DSTS <p>When you reset PHYs (using GUSB3PHYCFG or GUSB3PIPECTL registers), you must keep the controller in reset state until PHY clocks are stable. This controls the bus, ram, and mac domain resets.</p> <p>Note: This bit is for debug purposes only. Use USBCMD.HCRESET in xHCI Mode and DCTL.SoftReset in device mode for soft reset.</p> <p>Programming this field with random data will reset the internal logic of the host controller. Due to this side effect Bit Bash register testing is not recommended.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p>
10	SOFITPSYNC	R/W	<p>Not Use.</p> <p>Program this bit to 0 if the controller is intended to be operated in USB 3.0 mode.</p> <p>Value After Reset: 0x0</p>
9	U1U2TimerScale	R/W	<p>Disable U1/U2 timer Scaledown (U1U2TimerScale).</p> <p>If set to '1' along with GCTL[5:4] (ScaleDown) = 2'bX1, disables the scale down of U1/U2 inactive timer values. This is for simulation mode only.</p> <p>Value After Reset: 0x0</p>
8	DEBUGATTACH	R/W	<p>Debug Attach</p> <p>When this bit is set,</p> <ul style="list-style-type: none"> • SS Link proceeds directly to the Polling link state (after RUN/STOP in the DCTL register is asserted) without checking remote termination; • Link LFPS polling timeout is infinite; • Polling timeout during TS1 is infinite (in case link is waiting for TXEQ to finish). <p>Value After Reset: 0x0</p>
7:6	RAMCLKSEL	R/W	RAM Clock Select (RAMClkSel)

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> • 2'b00: bus clock • 2'b01: pipe clock (Only used in device mode) • 2'b10: In device mode , pipe/2 clock.In Host mode, controller switches ram_clk between pipe/2 clock, mac2_clk and bus_clk based on the status of the U2/U3 ports • 2'b11: In device mode, selects mac2_clk as ram_clk (when 8-bit UTMI or ULPI used. Not supported in 16-bit UTMI mode) In Host mode, controller switches ram_clk between pipe_clk, mac2_clk and bus_clk based on the status of the U2/U3 ports. <p>In device mode, upon a USB reset and USB disconnect, the hardware clears these bits to 2'b00.</p> <p>Note:</p> <ul style="list-style-type: none"> • In device mode, if you set RAMClkSel to 2'b11 (mac2_clk), the controller internally switches the ram_clk to bus_clk when the link state changes to Suspend (L2 or L3), and switches the ram_clk back to mac2_clk when the link state changes to resume or U2. • In host mode, if a value of 2/3 is chosen, then controller switches ram_clk between bus_clk, mac2_clk and pipe_clk, pipe_clk/2, based on the state of the U2/U3 ports. For example, if only the U2 port is active and the U3 ports are suspended, then the ram_clk is switched to mac2_clk. When only the U3 ports are active and the U2 ports are suspended, the controller internally switches the ram_clk to pipe3 clock and when all U2 and U3 ports are suspended, it switches the ram_clk to bus_clk. This allows decoupling the ram_clk from the bus_clk, and depending on the bandwidth requirement allows the bus_clk to be run at a lower frequency than the ram_clk requirements. The bus_clk frequency still cannot be less than 60MHz in host mode, and this is not verified. <p>A value of 2 can be chosen only if the pipe data width is 8 or 16 bits. In this case the when the ram_clk is switched to pipe_clk, it uses pipe_clk/2 instead of pipe_clk. If a value of 3 is chosen for RAMClkSel, then when ram_clk is switched to pipe_clk, then pipe_clk is used without any divider.</p> <ul style="list-style-type: none"> • In device mode, when RAMClkSel != 2'b00, the bus_clk_early frequency can be a minimum of 1 MHz. This is tested in simulation and also in hardware with Linux, Microsoft Windows 8, and MCCI Windows7 host drivers. Only control and non periodic transfers are supported when bus_clk is 1 MHz. For periodic applications, the bus_clk_early minimum frequency is higher depending on your application and SoC bus. Even though 1 MHz has been tested with standard host drivers, ESWIN recommends 5 MHz minimum for ASIC designs to provide a margin or at least have a backup option to increase the bus_clk frequency to 5 MHz if needed. <p>Programming this field with random data will cause side effect. Bit Bash register testing is not recommended.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p>
5:4	SCALEDOWN	R/W	<p>Scale-Down Mode (ScaleDown)</p> <p>When Scale-Down mode is enabled for simulation, the controller uses scaled-down timing values, resulting in faster simulations.</p>

Bits	Name	Memory Access	Description
			<p>When Scale-Down mode is disabled, actual timing values are used. This is required for hardware operation.</p> <p><i>HS/FS/LS Modes</i></p> <ul style="list-style-type: none"> • 2'b00: Disables all scale-downs. Actual timing values are used. • 2'b01: Enables scale-down of all timing values except Device mode suspend and resume. These include Speed enumeration, HNP/SRP, and Host mode suspend and resume • 2'b10: Enables scale-down of Device mode suspend and resume timing values only. • 2'b11: Enables bit 0 and bit 1 scale-down timing values. <p><i>SS Mode</i></p> <ul style="list-style-type: none"> • 2'b00: Disables all scale-downs. Actual timing values are used. • 2'b01: Enables scaled down SS timing and repeat values including: (1) Number of TxEq training sequences reduce to 8; (2) LFPS polling burst time reduce to 256 ns; (3) LFPS warm reset receive reduce to 30 uS. Refer to the <code>rtl_vip_scaledown_mapping.xls</code> file under <workspace>/sim/SoC_sim directory for the complete list. • 2'b10: No TxEq training sequences are sent. Overrides Bit 4. • 2'b11: Enables bit 0 and bit 1 scale-down timing values. <p>Value After Reset: 0x0</p>
3	DISSCRAMBLE	R/W	<p>Disable Scrambling (DisScramble)</p> <p>Transmit request to Link Partner on next transition to Recovery or Polling.</p> <p>Value After Reset: 0x0</p>
2	U2EXIT_LFPS	R/W	<p>U2EXIT_LFPS</p> <p>If this bit is,</p> <ul style="list-style-type: none"> • 0: the link treats 248ns LFPS as a valid U2 exit. • 1: the link waits for 8us of LFPS before it detects a valid U2 exit. <p>This bit is added to improve interoperability with a third-party host/device controller. This host/device controller in U2 state while performing receiver detection generates an LFPS glitch of about 4ms duration. This causes the host/device to exit from U2 state because the LFPS filter value is 248ns. With the new functionality enabled, the host/device can stay in U2 while ignoring this glitch from the host/device controller.</p> <p>This bit is applicable for both host and device controller. This bit is added to improve interoperability with a third party host controller. This host controller in U2 state while performing receiver detection generates an LFPS glitch of about 4ms duration. This causes the device to exit from U2 state because the LFPS filter value is 248ns. With the new functionality enabled, the device can stay in U2 while ignoring this glitch from the host controller.</p> <p>Value After Reset: 0x1</p>
1	GblHibernationEn	R/W	<p>GblHibernationEn</p> <p>Not Use.</p>

Bits	Name	Memory Access	Description
			<p>Value After Reset: 0x0</p> <p>WriteConstraint: writeAsRead</p>
0	DSBLCLKGTNG	R/W	<p>Disable Clock Gating (DsbIClkGtng)</p> <p>This bit is set to 1 and the controller is in Low Power mode, internal clock gating is disabled.</p> <p>You can set this bit to 1'b1 after Power On Reset.</p> <p>Value After Reset: 0x0</p>

USB_GSTS

- **Description:** Global Status Register
- **Size:** 32 bits
- **Offset:** 0xc118

Bits	Name	Memory Access	Description
31:20	CBELT	R	<p>Current BELT Value</p> <p>In Host mode, this field indicates the minimum value of all received device BELT values and the BELT value that is set by the Set Latency Tolerance Value command.</p> <p>Value After Reset: 0x7e8</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
19:12	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xff</p> <p>WriteConstraint: writeAsRead</p>
11	SSIC_IP	R	<p>This field is not used.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
10	OTG_IP	R	<p>This field is not used.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
9	BC_IP	R	<p>Battery Charger Interrupt Pending</p> <p>This field indicates that there is a pending interrupt pertaining to BC in BCEVT register.</p>

Bits	Name	Memory Access	Description
			<p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
8	ADP_IP	R	<p>This field is not used.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
7	Host_IP	R	<p>Host Interrupt Pending:</p> <p>This field indicates that there is a pending interrupt pertaining to xHC in the Host event queue.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
6	Device_IP	R	<p>Device Interrupt Pending</p> <p>This field indicates that there is a pending interrupt pertaining to peripheral (device) operation in the Device event queue.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
5	CSRTimeout	R/W1C	<p>CSR Timeout</p> <p>When this bit is 1'b1, it indicates that the software performed a write or read to a controller register that could not be completed within `DWC_USB3_CSR_ACCESS_TIMEOUT bus clock cycles (default: h1FFFF).</p> <p>Value After Reset: 0x0</p> <p>Reset Mask: 0x1</p> <p>Volatile: true</p>
4	BUSERRADDRVLD	R/W1C	<p>Bus Error Address Valid (BusErrAddrVld)</p> <p>Indicates that the GBUSERRADDR register is valid and reports the first bus address that encounters a bus error.</p> <p>Note: Only supported in AHB and AXI configurations.</p> <p>Value After Reset: 0x0</p> <p>Reset Mask: 0x1</p> <p>Volatile: true</p>
3:2	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p>

Bits	Name	Memory Access	Description
			<p>Reset Mask: 0x3</p> <p>WriteConstraint: writeAsRead</p>
1:0	CURMOD	R	<p>Current Mode of Operation (CurMod)</p> <p>Indicates the current mode of operation:</p> <ul style="list-style-type: none"> • 2'b00: Device mode • 2'b01: Host mode <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>

USB_GUCTL1

- **Description:** Global User Control Register 1
- **Size:** 32 bits
- **Offset:** 0xc11c

Bits	Name	Memory Access	Description
31	DEV_DECOUPLE_L1L2_EVT	R/W	<p>DEV_DECOUPLE_L1L2_EVT</p> <ul style="list-style-type: none"> • 0: Default behavior, no change in device events L1/L2U3 events are not decoupled (old behavior of v2.90a and before) • 1: Feature enabled, L1 and L2 events are separated when operating in 2.0 mode. Separate event enable bits for L1 suspend and wake events. <p>This bit is applicable for device mode only. If this feature is enabled, L1 suspend and wake events have individual controls to enable/mask them. Enable this feature if you want to get L1 (LPM) events separately and not combined with L2 events when operating in 2.0 speeds.</p>
30	DS_RXDET_MAX_TOUT_CTRL	R/W	<p>DS_RXDET_MAX_TOUT_CTRL</p> <p>This bit is used to control the tRxDetectTimeoutDFP timer for the SuperSpeed link.</p> <ul style="list-style-type: none"> • 0: Default behavior; 12ms is used as tRxDetectTimeoutDFP. • 1: 120ms is used as the tRxDetectTimeoutDFP. <p>This bit is used only in host mode. For more details, refer to ECN020 for USB 3.0 Specification.</p> <p>Value After Reset: 0x0</p>
29	FILTER_SE0_FSLS_EOP	R/W	<p>FILTER_SE0_FSLS_EOP</p> <ul style="list-style-type: none"> • 0: Default behavior, no change in Linestate check for SE0 detection in FS/LS • 1: Feature enabled, FS/LS SE0 is filtered for 2 clocks for detecting EOP

Bits	Name	Memory Access	Description
			<p>This bit is applicable for FS/LS operation. If this feature is enabled, then SEO on the linestate is validated for 2 consecutive utmi/ulpI clock edges for EOP detection. This feature is applicable only in FS in device mode and FS/LS mode of operation in host mode.</p> <p><i>Device mode:</i> FS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then for device LPM handshake, the controller will ignore single SEO glitch on the linestate during transmit. Only 2 or more SEO is considered as a valid EOP on FS.</p> <p><i>Host mode:</i> FS/LS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then the controller will ignore single SEO glitch on the linestate during transmit. Only 2 or more SEO is considered as a valid EOP on FS/LS port.</p> <p>Enable this feature if the LineState has SEO glitches during transmission. This bit is quasi-static, that is, it must not be changed during device operation.</p> <p>Value After Reset: 0x0</p>
28	TX_IPGAP_LINECHECK_DIS	R/W	<p>TX_IPGAP_LINECHECK_DIS</p> <ul style="list-style-type: none"> • 0: Default behavior, no change in Linestate check • 1: Feature enabled, 2.0 MAC disables Linestate check during HS transmit <p>This bit is applicable for HS operation of u2mac. If this feature is enabled, then the 2.0 mac operating in HS ignores the UTMI/ULPI Linestate during the transmit of a token (during token-to-token and token-to-data IPGAP). When enabled, the controller implements a fixed 40-bit TxEndDelay after the packet is given on UTMI and ignores the Linestate during this time. This feature is applicable only in HS mode of operation.</p> <p><i>Device mode:</i> If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then for device LPM handshake, the controller will ignore the linestate after TX and wait for fixed clocks (40 bit times equivalent) after transmitting ACK on utmi.</p> <p><i>Host mode:</i> If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then the ipgap between (tkn to tkn/data) is added by 40 bit times of TXENDDELAY, and linestate is ignored during this 40 bit times delay.</p> <p>Enable this bit if the LineState will not reflect the expected line state (J) during transmission. This bit is quasi-static, that is, it must not be changed during device operation.</p> <p>Value After Reset: 0x0</p>
27	DEV_TRB_OUT_SPR_IND	R/W	<p>DEV_TRB_OUT_SPR_IND</p> <ul style="list-style-type: none"> • 0: Default behavior, no change in TRB status dword • 1: Feature enabled, OUT TRB status indicates Short Packet <p>This bit is applicable for device mode only (and ignored in host mode). If the device application (software/hardware) wants to know if a short packet was received for an OUT in the TRB status itself, then this feature can be enabled, so that a bit is set in the TRB writeback in the buf_size dword. Bit[26] - SPR of the {trbstatus, RSVD, SPR, PCM1, bufsize} dword will be set during an OUT transfer TRB write back if this is the last TRB used for that transfer descriptor. This bit is quasi-static, that is, it must not be changed</p>

Bits	Name	Memory Access	Description
			<p>during device operation.</p> <p>Value After Reset: 0x0</p>
26	DEV_FORCE_20_CLK_FOR_30_CLK	R/W	<p>Not Use.</p> <p>Value After Reset: 0x0</p>
25	P3_IN_U2	R/W	<p>P3_IN_U2</p> <ul style="list-style-type: none"> • 0: Default behavior, When SuperSpeed link is in U2, PowerState P2 is attempted on the PIPE Interface. • 1: When SuperSpeed link is in U2, PowerState P3 is attempted if GUSB3PIPECTL[17] is set. <p>Setting this bit enables P3 Power State when the SuperSpeed link is in U2. Another Power Saving option. Check with your PHY vendor before enabling this option. When setting this bit to 1 to enable P3 in P2, GUSB3PIPECTL[27] should be set to 0 to make sure that the U2 exit is attempted in P0. This bit should be set only when GCTL.SOFITPSYNC=1 or GFLADJ.GFLADJ_REFCLK_LPM_SEL=1.</p> <p>Value After Reset: 0x0</p>
24	DEV_L1_EXIT_BY_HW	R/W	<p>DEV_L1_EXIT_BY_HW</p> <ul style="list-style-type: none"> • 0: Default behavior, disables device L1 hardware exit logic • 1: feature enabled <p>This bit is applicable for device mode (2.0) only. This field enables device controller sending remote wakeup for L1 if the device becomes ready for sending/accepting data when in L1 state. If the host expects the device to send remote wakeup signaling to resume after going into L1 in flow controlled state, then this bit can be set to send the remote wake signal automatically when the device controller becomes ready. This hardware remote wake feature is applicable only to bulk and interrupt transfers, and not for Isoch/Control</p> <ul style="list-style-type: none"> • When control transfers are in progress, the LPM will be rejected (NYET response). Only after control transfers are completed (either with ACK/STALL), LPM will be accepted • For Isoch transfers, the host needs to do the wake-up and start the transfer. Device controller will not do remote-wakeup when Isoch endpoints get ready. The device SW needs to keep the GUSB2PHYCFG[EnbISlpM] reset in order to keep the PHY clock to be running for keeping track of SOF intervals. • When L1 hibernation is enabled, the controller will not do automatic exit for hibernation requests thru L1. <p>This bit is quasi-static, that is, it must not be changed during device operation.</p> <p>Value After Reset: 0x0</p>
23:21	IP_GAP_ADD_ON	R/W	<p>This register field is used to add on to the default inter packet gap setting in the USB 2.0 MAC. This should be programmed to a non zero value only in case where you need to increase the default inter packet delay calculations in the USB 2.0 MAC module DWC_usb3_u2mac.v</p>

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
20	DEV_LSP_TAIL_LOCK_DIS	R/W	<p>DEV_LSP_TAIL_LOCK_DIS</p> <ul style="list-style-type: none"> • 0: Default behavior, enables device lsp lock logic for tail TRB update • 1: Fix disabled <p>This is a bug fix for STAR 9000716195 that affects the CSP mode for OUT endpoints in device mode. The issue is that tail TRB index is not synchronized with the cache Scratchpad bytecount update. If the fast-forward request comes in-between the bytecount update on a newly fetched TRB and the tail-index write update in TPF, the RDP works on an incorrect tail index and misses the byte count decrement for the newly fetched TRB in the fast-forwarding process. This fix needs to be present all the times.</p> <p>Value After Reset: 0x0</p>
19	NAK_PER_ENH_FS	R/W	<p>NAK_PER_ENH_FS</p> <ul style="list-style-type: none"> • 1: Enables performance enhancement for FS async endpoints in the presence of NAKs • 0: Enhancement not applied <p>If a periodic endpoint is present , and if a bulk endpoint which is also active is being NAKed by the device, then this could result in a decrease in performance of other Full Speed bulk endpoint which is ACKed by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your FullSpeed application. Setting this bit will only control, and is only required for Full Speed transfers.</p> <p>Value After Reset: 0x0</p>
18	NAK_PER_ENH_HS	R/W	<p>NAK_PER_ENH_HS</p> <ul style="list-style-type: none"> • 1: Enables performance enhancement for HS async endpoints in the presence of NAKs • 0: Enhancement not applied <p>If a periodic endpoint is present, and if a bulk endpoint which is also active is being NAKed by the device, then this could result in decrease in performance of other High Speed bulk endpoint which is ACKed by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your HighSpeed application. Setting this bit will only control, and is only required for High Speed transfers.</p> <p>Value After Reset: 0x1</p>
17	PARKMODE_DISABLE_SS	R/W	<p>PARKMODE_DISABLE_SS</p> <p>This bit is used only in host mode, and is for debug purpose only.</p>

Bits	Name	Memory Access	Description
			<p>When this bit is set to '1' all SS bus instances in park mode are disabled.</p> <p>Value After Reset: 0x0</p>
16	PARKMODE_DISABLE_HS	R/W	<p>PARKMODE_DISABLE_HS</p> <p>This bit is used only in host mode.</p> <p>When this bit is set to '1' all HS bus instances park mode are disabled.</p> <p>To improve performance in park mode, the xHCI scheduler queues in three requests of 4 packets each for High Speed asynchronous endpoints in a micro-frame. But if a device is slow and if it NAKs more than 3 times, then it is rescheduled only in the next micro-frame. This could decrease the performance of a slow device even further.</p> <p>In a few high speed devices (such as Sandisk Cruzer Blade 4GB VID:1921, PID:21863 and Flex Drive VID:3744, PID:8552) when an IN request is sent within 900ns of the ACK of the previous packet, these devices send a NAK. When connected to these devices, if required, the software can disable the park mode if you see performance drop in your system. When park mode is disabled, pipelining of multiple packet is disabled and instead one packet at a time is requested by the scheduler. This allows up to 12 NAKs in a micro-frame and improves performance of these slow devices.</p> <p>Value After Reset: 0x0</p>
15	PARKMODE_DISABLE_FSL S	R/W	<p>PARKMODE_DISABLE_FSL S</p> <p>This bit is used only in host mode, and is for debug purpose only.</p> <p>When this bit is set to '1' all FS/LS bus instances in park mode disabled.</p> <p>Value After Reset: 0x0</p>
14:13	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x3</p> <p>WriteConstraint: writeAsRead</p>
12	DisUSB2RefClkGtng	R	<p>Disable ref_clk gating for 2.0 PHY (DisUSB2RefClkGtng)</p> <p>If ref_clk gating is disabled, then the ref_clk input cannot be turned off to the USB 2.0 PHY and controller. This is independent of the GCTL[DisClkGtng] setting.</p> <ul style="list-style-type: none"> • 1'b0: ref_clk gating enabled for USB 2.0 PHY • 1'b1: ref_clk gating disabled for USB 2.0 PHY <p>Value After Reset: 0x1</p> <p>WriteConstraint: writeAsRead</p>
11	DisRefClkGtng	R	Disable ref_clk gating (DisRefClkGtng)

Bits	Name	Memory Access	Description
			<p>If the ref_clk gating is disabled then input ref_clk cannot be turned off to SSPHY and controller. This is independent of GCTL[DisClkGtng] setting.</p> <ul style="list-style-type: none"> • 1'b0: ref_clk gating Enabled for SSPHY • 1'b1: ref_clk gating Disabled for SSPHY <p>Value After Reset: 0x1</p> <p>WriteConstraint: writeAsRead</p>
10	RESUME_OPMODE_HS_HOST	R/W	<p>RESUME_OPMODE_HS_HOST</p> <p>This bit is used only in host mode, and is for USB 2.0 opmode behavior in HS Resume.</p> <ul style="list-style-type: none"> • When this bit is set to '1', the UTMI/ULPI opmode will be changed to "normal" along with HS terminations after EOR. This option is to support certain legacy UTMI/ULPI PHYs. • When this bit is set to '0', the UTMI/ULPI opmode will be changed to "normal" 2us after HS terminations change after EOR. This is the default behavior. <p>Value After Reset: 0x0</p>
9	DEV_HS_NYET_BULK_SPR	R/W	<p>DEV_HS_NYET_BULK_SPR</p> <ul style="list-style-type: none"> • 0: Default behavior, no change in device response • 1: Feature enabled, HS bulk OUT short packet gets NYET response <p>This bit is applicable for device mode only (and ignored in host mode) to be used in 2.0 operation.</p> <p>If this bit is set, the device controller sends NYET response instead of ACK response for a successfully received bulk OUT short packet. If NYET is sent after receiving short packet, then the host would PING before sending the next OUT; this improves the performance as well as clears up the buffer/cache on the host side. Internal to the device controller, short packet (SPR=1) processing takes some time, and during this time, the USB is flow controlled. With NYET response instead of ACK on short packet, the host does not send another OUT-DATA without pinging in HS mode.</p> <p>This bit is quasi-static, that is, it must not be changed during device operation.</p> <p>Value After Reset: 0x0</p>
8	L1_SUSP_THRLD_EN_FOR_HOST	R/W	<p>L1_SUSP_THRLD_EN_FOR_HOST</p> <p>This bit is used only in host mode.</p> <p>The host controller asserts the utmi_l1_suspend_n and utmi_sleep_n output signals (see "LPM Interface Signals" table in the Databook) as follows:</p> <p>The controller asserts the utmi_l1_suspend_n signal to put the PHY into deep low-power mode in L1 when both of the following are true:</p> <ul style="list-style-type: none"> • The HIRD/BESL value used is greater than or equal to the value in L1_SUSP_THRLD_FOR_HOST field.

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b1. <p>The controller asserts utmi_sleep_n on L1 when one of the following is true:</p> <ul style="list-style-type: none"> The HIRD/BESL value used is less than the value in L1_SUSP_THRLD_FOR_HOST field. The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b0. <p>Value After Reset: 0x1</p>
7:4	L1_SUSP_THRLD_FOR_HOST	R/W	<p>L1_SUSP_THRLD_FOR_HOST</p> <p>This field is effective only when the L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1. For more details, refer to the description of the L1_SUSP_THRLD_EN_FOR_HOST bit.</p> <p>Value After Reset: 0x8</p>
3	HC_ERRATA_ENABLE	R/W	<p>Host ELD Enable (HELDEn)</p> <p>When this bit is set to 1, it enables the Exit Latency Delta (ELD) support defined in the xHCl 1.0 Errata.</p> <p>This bit is used only in the host mode. This bit has to be set to 1 in Host mode.</p> <p>Value After Reset: 0x1</p>
2	HC_PARCHK_DISABLE	R/W	<p>Host Parameter Check Disable (HParChkDisable)</p> <p>When this bit is set to '0' (by default), the xHC checks that the input slot/EP context fields comply to the xHCl Specification. Upon detection of a parameter error during command execution, the xHC generates an event TRB with completion code indicating 'PARAMETER ERROR'.</p> <p>When the bit is set to '1', the xHC does not perform parameter checks and does not generate 'PARAMETER ERROR' completion code.</p> <p>Value After Reset: 0x0</p>
1	OVRLD_L1_SUSP_COM	R/W	<p>OVRLD_L1_SUSP_COM</p> <p>If this bit is set, the utmi_l1_suspend_com_n is overloaded with the utmi_sleep_n signal. This bit is usually set if the PHY stops the port clock during L1 sleep condition.</p> <p>Note: The recommended connection for the SUSPENDM/SLEEPM signals to the PHY with respect to this bit is as follows.</p> <p>For non-zero ports:</p> <p>Connect:</p> <ul style="list-style-type: none"> utmi_sleep_n[n] to SLEEPM[n] (utmi_suspend_n[n] & utmi_l1_suspend_n[n]) to SUSPENDM[n] USB2 PHYCLK[n] to utmi_clk[n] <p>GUCTL1.OVRLD_L1_SUSP_COM impacts only Port0.</p> <p>For Port0:</p>

Bits	Name	Memory Access	Description
			<p><i>For ESWIN PHY,</i></p> <p>GUSB2PHYCFGn.U2_FREECLK_EXISTS=1; With this connection, the PHY keeps PLL active so that FREECLK is always available irrespective of suspend/sleep.</p> <ul style="list-style-type: none"> • Connect USB2 PHY COMMONONN to 0. • Connect utmi_sleep_n[0] to SLEEPM[0]. • Connect (utmi_suspend_n[0] & utmi_l1_suspend_n[0]) to SUSPENDM[0]. • Connect USB2 PHY FREECLK to utmi_clk[0]. • Leave utmi_suspend_com_n, utmi_l1_suspend_com_n unconnected. • GUCTL1.OVRLD_L1_SUSP_COM can be set to any value. <p><i>For Third Party PHY,</i></p> <p>GUSB2PHYCFGn.U2_FREECLK_EXISTS=0; With this connection the PHY can shut off all the clocks when the required conditions are met (like, GUSB2PHYCFGn[8,6], GUCTL1[1], GFLADJ[23], GCTL[10], Suspend condition, HW LPM enable etc).</p> <ul style="list-style-type: none"> • Connect -utmi_suspend_com_n to SUSPENDM[0] (or equivalent). • Connect -utmi_l1_suspend_com_n to SLEEPM[0] (or equivalent). • Connect PHYCLK0 (first port clock) to utmi_clk[0]. • Leave utmi_suspend_n[0], utmi_l1_suspend_n[0], utmi_sleep_n[0] unconnected. • Set GUCTL1.OVRLD_L1_SUSP_COM to 1'b1. <p>Value After Reset: 0x1</p>
0	LOA_FILTER_EN	R/W	<p>LOA_FILTER_EN</p> <p>If this bit is set, the USB 2.0 port babble is checked at least three consecutive times before the port is disabled. This prevents false triggering of the babble condition when using low quality cables.</p> <p>Note: This bit is valid only in host mode.</p> <p>Value After Reset: 0x0</p>

USB_GUID

- **Description:** Global User ID Register
- This is a read/write register containing the User ID. The power-on value for this register is specified as the User Identification Register. This register can be used in the following ways: To store the version or revision of your system; To store hardware configurations that are outside the controller; As a scratch register.
- **Size:** 32 bits
- **Offset:** 0xc128

Bits	Name	Memory Access	Description
31:0	USERID	R/W	<p>USERID Application-programmable ID field.</p> <p>Value After Reset: 0x12345678</p>

USB_GUCTL

- **Description:** Global User Control Register
- This register provides a few options for the software to control the controller behavior in the Host mode. Most of the options are used to improve host inter-operability with different devices.
- **Size:** 32 bits
- **Offset:** 0xc12c

Bits	Name	Memory Access	Description
31:22	REFCLKPER	R/W	<p>Not Use.</p> <p>This field must not be set to '0' at any time. If you never plan to use this feature, then set this field to 'h8, the default value.</p> <p>Value After Reset: 0x14</p>
21	NoExtrDI	R/W	<p>No Extra Delay Between SOF and the First Packet(NoExtrDI)</p> <p>Some HS devices misbehave when the host sends a packet immediately after a SOF. However, adding an extra delay between a SOF and the first packet can reduce the USB data rate and performance.</p> <p>This bit is used to control whether the host must wait for 2 microseconds before it sends the first packet after a SOF, or not. User can set this bit to one to improve the performance if those problematic devices are not a concern in the user's host environment.</p> <ul style="list-style-type: none"> • 1'b0: Host waits for 2 microseconds after a SOF before it sends the first USB packet. • 1'b1: Host doesn't wait after a SOF before it sends the first USB packet. <p>Value After Reset: 0x0</p>
20:18	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x7</p> <p>WriteConstraint: writeAsRead</p>
17	SprsCtrlTransEn	R/W	<p>Sparse Control Transaction Enable</p> <p>Some devices are slow in responding to Control transfers. Scheduling multiple transactions in one microframe/frame can cause these devices to misbehave.</p> <p>If this bit is set to 1'b1, the host controller schedules transactions for a Control transfer in different microframes/frames.</p>

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
16	ResBwHSEPS	R/W	<p>Reserving 85% Bandwidth for HS Periodic EPs (ResBwHSEPS)</p> <p>By default, HC reserves 80% of the bandwidth for periodic EPs. If this bit is set, the bandwidth is relaxed to 85% to accommodate two high speed, high bandwidth ISOC EPs.</p> <p>USB 2.0 required 80% bandwidth allocated for ISOC traffic. If two High-bandwidth ISOC devices (HD Webcams) are connected, and if each requires 1024-bytes X 3 packets per Micro-Frame, then the bandwidth required is around 82%. If this bit is set, then it is possible to connect two Webcams of 1024bytes X 3 payload per Micro-Frame each. Otherwise, you may have to reduce the resolution of the Webcams.</p> <p>This bit is valid in Host and DRD configuration and is used in host mode operation only. Ignore this bit in device mode.</p> <p>Value After Reset: 0x0</p>
15	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p>
14	USBHstInAutoRetryEn	R/W	<p>Host IN Auto Retry (USBHstInAutoRetryEn)</p> <p>When set, this field enables the Auto Retry feature. For IN transfers (non-isochronous) that encounter data packets with CRC errors or internal overrun scenarios, the auto retry feature causes the Host controller to reply to the device with a non-terminating retry ACK (that is, an ACK transaction packet with Retry = 1 and NumP != 0).</p> <p>If the Auto Retry feature is disabled (default), the controller will respond with a terminating retry ACK (that is, an ACK transaction packet with Retry = 1 and NumP = 0).</p> <ul style="list-style-type: none"> • 1'b0: Auto Retry Disabled • 1'b1: Auto Retry Enabled <p>Note: When enabling Auto Retry feature, if the system latency is large enough to cause the internal PSQ full (PSQ can be full as the result of messages not being processed because of pending fetches before flushing the TxQ due to NRDY/ERDY conditions), then the host controller can generate a transaction error.</p> <p>Value After Reset: 0x0</p>
13	EnOverlapChk	R/W	<p>Enable Check for LFPS Overlap During Remote Ux Exit:</p> <p>If this bit is set to,</p> <ul style="list-style-type: none"> • 1'b1: The SuperSpeed link when exiting U1/U2/U3 waits for either the remote link LFPS or TS1/TS2 training symbols before it confirms that the LFPS handshake is complete. This is done to handle the case where the LFPS glitch causes the link to start exiting from the low power state. Looking for the

Bits	Name	Memory Access	Description
			<p>LFPS overlap makes sure that the link partner also sees the LFPS.</p> <ul style="list-style-type: none"> • 1'b0: When the link exists U1/U2/U3 because of a remote exit, it does not look for an LFPS overlap. <p>Value After Reset: 0x0</p>
12	ExtCapSupptEN	R/W	<p>External Extended Capability Support Enable (ExtCapSuptEN)</p> <p>When set, this field enables extended capabilities to be implemented outside the controller.</p> <p>When the ExtCapSupEN is set and the Debug Capability is enabled, the Next Capability pointer in "Debug Capability" returns 16.</p> <p>A read to the first DWORD of the last internal extended capability (the "xHCI Supported Protocol Capability for USB 3.0" when the Debug Capability is not enabled) returns a value of 4 in the Next Capability Pointer field.</p> <p>This indicates to software that there is another capability four DWORDs after this capability (for example, at address N+16 where N is the address of this DWORD). If enabled, an external address decoder that snoops the xHC slave interface must be implemented. If it sees an access to N+16 or greater, the slave access is re-routed to a piece of hardware which returns the external capability pointer register of the new capability and also handles reads/writes to this new capability and the side effects.</p> <p>If disabled, a read to the first DWORD of the last internal extended capability returns 0 in the 'Next Capability Pointer' field. This indicates there are no more capabilities.</p> <p>Value After Reset: 0x0</p>
11	InsrtExtrFSBODI	R/W	<p>Insert Extra Delay Between FS Bulk OUT Transactions (InsrtExtrFSBODI).</p> <p>Some FS devices are slow to receive Bulk OUT data and can get stuck when there are consecutive Bulk OUT transactions with short inter-transaction delays. This bit is used to control whether the host inserts extra delay between consecutive Bulk OUT transactions to a FS Endpoint.</p> <ul style="list-style-type: none"> • 1'b0: Host doesn't insert extra delay between consecutive Bulk OUT transactions to a FS Endpoint. • 1'b1: Host inserts about 12us extra delay between consecutive Bulk OUT transactions to a FS Endpoint to work around the device issue. <p>Note: Setting this bit to one will reduce the Bulk OUT transfer performance for most of the FS devices.</p> <p>Value After Reset: 0x0</p>
10:9	DTCT	R/W	<p>Device Timeout Coarse Tuning (DTCT)</p> <p>This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout.</p> <p>The controller first checks the DTCT value. If it is 0, then the</p>

Bits	Name	Memory Access	Description
			<p>timeout value is defined by the DTFT. If it is non-zero, then it uses the following timeout values:</p> <ul style="list-style-type: none"> • 2'b00: 0 usec -> use DTFT value instead • 2'b01: 500 usec • 2'b10: 1.5 msec • 2'b11: 6.5 msec <p>Note: When the system latency is larger than the programmed DTCT/DTFT value, if the host controller is not able to accept certain transactions on the bus (because of system bus delays), the controller may not release header credits which in turn can cause the host to report a transaction error. Therefore, program this value to be larger than your system delay.</p> <p>Value After Reset: 0x0</p>
8:0	DTFT	R/W	<p>Device Timeout Fine Tuning (DTFT)</p> <p>This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout.</p> <p>For the DTFT field to take effect, DTCT must be set to 2'b00.</p> <p>The DTFT value is the number of 125 MHz clocks * 256 to count before considering a device timeout.</p> <p>The minimum value of DTFT is 2.</p> <p>For example, if the mac3_clk is 125 MHz clk (8 ns period), this is calculated as follows:</p> $(DTFT \text{ value}) * 256 * (8 \text{ ns})$ <p>Quick Reference:</p> <ul style="list-style-type: none"> • if DTFT = 0x2, $2*256*8 = 4\text{usec}$ timeout • if DTFT = 0x5, $5*256*8 = 10\text{usec}$ timeout • if DTFT = 0xA, $10*256*8 = 20\text{usec}$ timeout • if DTFT = 0x10, $16*256*8 = 32\text{usec}$ timeout • if DTFT = 0x19, $25*256*8 = 51\text{usec}$ timeout • if DTFT = 0x31, $49*256*8 = 100\text{usec}$ timeout • if DTFT = 0x62, $98*256*8 = 200\text{usec}$ timeout <p>Note: When the system latency is larger than the programmed DTCT/DTFT value, if the host controller is not able to accept certain transactions on the bus (because of system bus delays), the controller may not release header credits which in turn can cause the host to report a transaction error. Therefore, program this value to be larger than your system delay.</p> <p>Value After Reset: 0x10</p>

USB_GBUSERRADDRLO

- **Description:** Gobal SoC Bus Error Address Register – Low
- **Size:** 32 bits
- **Offset:** 0xc130

Bits	Name	Memory Access	Description
31:0	BUSERRADDR	R	<p>Bus Address - Low (BusAddrLo) This register contains the lower 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the controller.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>

USB_GBUSERRADDRHI

- **Description:** Gobal SoC Bus Error Address Register - High
- **Size:** 32 bits
- **Offset:** 0xc134

Bits	Name	Memory Access	Description
31:0	BUSERRADDR	R	<p>Bus Address - High (BusAddrHi) This register contains the higher 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the controller.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>

USB_GPRTBIMAPLO

- **Description:** Global SS Port to Bus Instance Mapping Register – Low. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields. For a configuration with number of USB 3.0 ports same as number of SS Bus Instances, do not remap during debug session. If you remap for some reason, then the debug host must be connected to a port which has a dedicated SS Bus Instance. For example, if USB3_NUM_U3_ROOT_PORTS =3 and USB3_NUM_SS_USB_INSTANCES=3, and software maps the first SS port to the first SS BI and the second/third port to the second BI, then the debug host can be connected to the first port only.
- **Size:** 32 bits
- **Offset:** 0xc138

Bits	Name	Memory Access	Description
31:28	BINUM8	R/W	<p>BINUM8: SS USB Instance Number for Port 8.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
27:24	BINUM7	R/W	BINUM7: SS USB Instance Number for Port 7.

Bits	Name	Memory Access	Description
			<p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
23:20	BINUM6	R/W	<p>BINUM6: SS USB Instance Number for Port 6.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
19:16	BINUM5	R/W	<p>BINUM5: SS USB Instance Number for Port 5.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
15:12	BINUM4	R/W	<p>BINUM4: SS USB Instance Number for Port 4.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
11:8	BINUM3	R/W	<p>BINUM3: SS USB Instance Number for Port 3.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
7:4	BINUM2	R/W	<p>BINUM2: SS USB Instance Number for Port 2.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
3:0	BINUM1	R/W	<p>BINUM1: SS USB Instance Number for Port 1.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>

USB_GPRTBIMAPHI

- **Description:** Global SS Port to Bus Instance Mapping Register – High. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields. For a configuration with number of USB 3.0 ports same as number of SS Bus Instances, do not remap during debug session. If you remap for some reason, then the debug host must be connected to a port which has a dedicated SS Bus Instance. For example, if USB3_NUM_U3_ROOT_PORTS=3 and USB3_NUM_SS_USB_INSTANCES=3, and software maps the first SS port to the first SS BI and the second/third port to the second BI, then the debug host can be connected to the first port only.
- **Size:** 32 bits
- **Offset:** 0xc13c

Bits	Name	Memory Access	Description
31:28	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xf</p> <p>WriteConstraint: writeAsRead</p>
27:24	BINUM15	R/W	<p>BINUM15: SS USB Instance Number for Port 15.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
23:20	BINUM14	R/W	<p>BINUM14: SS USB Instance Number for Port 14.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
19:16	BINUM13	R/W	<p>BINUM13: SS USB Instance Number for Port 13.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
15:12	BINUM12	R/W	<p>BINUM12: SS USB Instance Number for Port 12.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
11:8	BINUM11	R/W	<p>BINUM11: SS USB Instance Number for Port 11.</p> <p>Application-programmable ID field.</p>

Bits	Name	Memory Access	Description
			Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead
7:4	BINUM10	R/W	BINUM10: SS USB Instance Number for Port 10. Application-programmable ID field. Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead
3:0	BINUM9	R/W	BINUM9: SS USB Instance Number for Port 9. Application-programmable ID field. Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead

USB_GPRTBIMAPHSLO

- **Description:** Global High-Speed Port to Bus Instance Mapping Register – Low. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.
- **Size:** 32 bits
- **Offset:** 0xc180

Bits	Name	Memory Access	Description
31:28	BINUM8	R/W	BINUM8: HS USB Instance Number for Port 8. Application-programmable ID field. Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead
27:24	BINUM7	R/W	BINUM7: HS USB Instance Number for Port 7. Application-programmable ID field. Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead
23:20	BINUM6	R/W	BINUM6 USB Instance Number for Port 6. Application-programmable ID field. Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead
19:16	BINUM5	R/W	BINUM5: HS USB Instance Number for Port 5.

Bits	Name	Memory Access	Description
			<p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
15:12	BINUM4	R/W	<p>BINUM4: HS USB Instance Number for Port 4.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
11:8	BINUM3	R/W	<p>BINUM3: HS USB Instance Number for Port 3.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
7:4	BINUM2	R/W	<p>BINUM2: HS USB Instance Number for Port 2.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
3:0	BINUM1	R/W	<p>BINUM1: HS USB Instance Number for Port 1.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>

USB_GPRTBIMAPHSHI

- **Description:** Global High-Speed Port to Bus Instance Mapping Register – High. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.
- **Size:** 32 bits
- **Offset:** 0xc184

Bits	Name	Memory Access	Description
31:28	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xf</p> <p>WriteConstraint: writeAsRead</p>

Bits	Name	Memory Access	Description
27:24	BINUM15	R/W	<p>BINUM15: HS USB Instance Number for Port 15.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
23:20	BINUM14	R/W	<p>BINUM14: HS USB Instance Number for Port 14. Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
19:16	BINUM13	R/W	<p>BINUM13: HS USB Instance Number for Port 13.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
15:12	BINUM12	R/W	<p>BINUM12: HS USB Instance Number for Port 12.</p> <p>SApplication-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
11:8	BINUM11	R/W	<p>BINUM11: HS USB Instance Number for 11.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
7:4	BINUM10	R/W	<p>BINUM10: HS USB Instance Number for Port 10.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
3:0	BINUM9	R/W	<p>BINUM9: HS USB Instance Number for Port 9.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>

USB_GPRTBIMAPFSLO

- **Description:** Global Full-Speed Port to Bus Instance Mapping Register – Low. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.
- **Size:** 32 bits
- **Offset:** 0xc188

Bits	Name	Memory Access	Description
31:28	BINUM8	R/W	<p>BINUM8: FS USB Instance Number for Port 8. Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
27:24	BINUM7	R/W	<p>BINUM7: FS USB Instance Number for Port 7. Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
23:20	BINUM6	R/W	<p>BINUM6: FS USB Instance Number for Port 6. Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
19:16	BINUM5	R/W	<p>BINUM5: FS USB Instance Number for Port 5. Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
15:12	BINUM4	R/W	<p>BINUM4: FS USB Instance Number for Port 4. Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
11:8	BINUM3	R/W	<p>BINUM3: FS USB Instance Number for Port 3. Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>

Bits	Name	Memory Access	Description
7:4	BINUM2	R/W	<p>BINUM2: FS USB Instance Number for Port 2.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
3:0	BINUM1	R/W	<p>BINUM1: FS USB Instance Number for Port 1.</p> <p>Application-programmable ID field.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>

USB_GPRTBIMAPFSHI

- **Description:** Global Full-Speed Port to Bus Instance Mapping Register – High. Register fields are read-write with respect to number of port instantiated. writeAsRead constraint is added to limit side effects for unused fields.
- **Size:** 32 bits
- **Offset:** 0xc18c

Bits	Name	Memory Access	Description
31:28	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xf</p> <p>WriteConstraint: writeAsRead</p>
27:24	BINUM15	R/W	<p>BINUM15: FS USB Instance Number for Port 15.</p> <p>Application-programmable ID field</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
23:20	BINUM14	R/W	<p>BINUM14: FS USB Instance Number for Port 14.</p> <p>Application-programmable ID field</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
19:16	BINUM13	R/W	<p>BINUM13: FS USB Instance Number for Port 13.</p> <p>Application-programmable ID field</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p>

Bits	Name	Memory Access	Description
			WriteConstraint: writeAsRead
15:12	BINUM12	R/W	BINUM12: FS USB Instance Number for Port 12. Application-programmable ID field Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead
11:8	BINUM11	R/W	BINUM11: FS USB Instance Number for Port 11. Application-programmable ID field Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead
7:4	BINUM10	R/W	BINUM10: FS USB Instance Number for Port 10. Application-programmable ID field Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead
3:0	BINUM9	R/W	BINUM9: FS USB Instance Number for Port 9. Application-programmable ID field. Value After Reset: 0x0 Testable: writeAsRead WriteConstraint: writeAsRead

USB_GUCTL2

- **Description:** Global User Control Register 2. This register provides a few options for the software to control the controller behavior in the Host and device mode. Most of the options are used to improve inter-operability with different hosts and devices.
- **Size:** 32 bits
- **Offset:** 0xc19c

Bits	Name	Memory Access	Description
31:26	reserved	R/W	Reserved Value After Reset: 0x0 Testable: writeAsRead Reset Mask: 0xff WriteConstraint: writeAsRead
25:19	EN_HP_PM_TIMER	R/W	This register field is used to set new HP and PM timers. <ul style="list-style-type: none"> • To enable PM timer, set GUCTL2[19] bit as 1.

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> To enable HP timer, set GUCTL2[20] bit as 1. Default value of HP timer is 4us when HP PM timer is not enabled; when new HP timer is enabled default value is 12us. <p>Use GUCTL2[25:21] to specify HP timer value in microseconds.</p> <p>Value After Reset: 0x33</p>
18:15	NOLOWPWRDUR	R/W	<p>No Low Power Duration (NOLOWPWRDUR)</p> <p>This bit is applicable for device mode only and is ignored in host mode.</p> <p>After starting a transfer on a SS ISOC endpoint, the application must program these bits to prevent the device to lose frame synchronization over a period of time. Based on this count-down counter, the device will wake itself from U1/U2 low power states. After entering to U0 state and receiving two ITPs (which will sync-up the host and the device), U1/U2 low power entry is allowed.</p> <p>Each count represents the duration in terms of milliseconds. For example, a value of 3 represents 3ms.</p> <p>Note:</p> <ul style="list-style-type: none"> To disable this feature, set this field to 4'b0. These bits are applicable only in device mode and ignored in host mode. Some xHCI hosts do not send ITPs when performing ISOC transfers when the link enters U1/U2 low power states. This causes the device to lose frame synchronization over a period of time resulting in ISOC packets being dropped. <p>Value After Reset: 0x0</p>
14	Rst_actbitlater	R/W	<p>Enable clearing of the command active bit for the ENDXFER command after the command execution is completed.</p> <p>This bit is valid in device mode only.</p> <p>Value After Reset: 0x0</p>
13	reserved	R	<p>Reserved for future use</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
12	EnableEpCacheEvict	R/W	<p>Enable Evicting Endpoint cache after Flow Control for bulk endpoints.</p> <p>In 3.00a release, a performance enhancement was done to keep the non-stream capable bulk IN endpoint in cache after flow control. Setting this bit will disable this enhancement. This should be set only for debug purpose.</p> <p>Value After Reset: 0x0</p>
11	DisableCFC	R/W	<p>Disable xHCI Errata Feature Contiguous Frame ID Capability</p> <p>This field controls the xHCI Errata feature Contiguous FrameID capability. When set, the xHCI HCCPARAMS1 bit 11 will be set to 0 indicating that CFC is not supported. Disable this feature only if your application cannot tolerate</p>

Bits	Name	Memory Access	Description
			<p>Missed Service Error events for Isochronous transfers, and your system latencies are large to cause Missed Service errors even if the software is following the Isochronous Thresholding rules.</p> <p>Value After Reset: 0x0</p>
10:5	RxPingDuration	R/W	<p>Receive Ping Maximum Duration</p> <p>This field is relevant to Host mode and controls the maximum duration of received LFPS to be treated as a Ping LFPS. The Max duration of the Ping LFPS is controlled by programming this value and is in terms of 8 ns granularity. Eg: A value of 32 indicates 256 ns.</p> <p>Value After Reset: 0x20</p>
4:0	TxPingDuration	R/W	<p>Transmit Ping Maximum Duration</p> <p>This field is relevant to Device mode and controls the maximum duration for which the controller should instruct the PHY to transmit a Ping LFPS. The duration of the Ping LFPS is controlled by programming this value and is in terms of 8 ns granularity. Eg: A value of 13 indicates 104 ns.</p> <p>Value After Reset: 0xd</p>

USB_GUSB2PHYCFG

- **Description:** Global USB2 PHY Configuration Register. The application must program this register before starting any transactions on either the SoC bus or the USB.
- **Size:** 32 bits
- **Offset:** 0xc200

Bits	Name	Memory Access	Description
31	PHYSOFTRST	R/W	<p>UTMI PHY Soft Reset (PHYSoftrst)</p> <p>Causes the usb2phy_reset signal to be asserted to reset a UTMI PHY. Not applicable to ULPI because ULPI PHYs are reset via their FunctionControl.Reset register, and the controller automatically writes to this register when the controller is reset (vcc_reset_n, USBCMD.HCRST, DCTL.SoftReset, or GCTL.SoftReset)</p> <p>Value After Reset: 0x0</p>
30	U2_FREECLK_EXISTS	R/W	<p>U2_FREECLK_EXISTS</p> <p>Specifies whether your USB 2.0 PHY provides a free-running PHY clock, which is active when the clock control input is active.</p> <p>If your USB 2.0 PHY provides a free-running PHY clock, it must be connected to the utmi_clk[0] input. The remaining utmi_clk[n] must be connected to the respective port clocks. The controller uses the Port-0 clock for generating the internal mac2 clock.</p> <ul style="list-style-type: none"> • 1'b0: USB 2.0 free clock does not exist • 1'b1: USB 2.0 free clock exists <p>Note: When the controller is configured as device-only (DWC_USB3_MODE = 0), do not set this bit to 1.</p> <p>Value After Reset: 0x1</p>
29	ULPI_LPM_WITH_OPMODE	R/W	ULPI_LPM_WITH_OPMODE_CHK

Bits	Name	Memory Access	Description
	_CHK		<p>Support the LPM over ULPI without NOPID token to the ULPI PHY.</p> <p>If this bit is set, the ULPI PHY is expected to qualify the EXT PID with OPMODE=2'b00 for LPM and not treat it as a NOPID. Check with your PHY vendor about your PHY behavior. This bit is valid only when the DWC_USB3_HSPHY_INTERFACE parameter is 2 or 3.</p> <ul style="list-style-type: none"> • 1'b0: A NOPID is sent before sending an EXTPID for LPM; • 1'b1: An EXTPID is sent without previously sending a NOPID; <p>Note: This bit is valid only in host mode. This bit should be '0' for ESWIN PHY.</p> <p>Value After Reset: 0x0</p>
28:27	HSIC_CON_WIDTH_ADJ	R	<p>HSIC_CON_WIDTH_ADJ</p> <p>This bit is used in the HSIC device mode of operation. By default, the connect duration for the HSIC device controller is thrice the strobe period. You can change this duration to 4, 5, or 6 times the strobe period by setting the value of this field to 1, 2, or 3. This value is added to the default connect duration.</p> <p>Value After Reset: 0x0</p> <p>WriteConstraint: writeAsRead</p>
26	INV_SEL_HSIC	R	<p>INV_SEL_HSIC</p> <p>The application driver uses this bit to control the HSIC enable/disable function. When set to '1', this bit overrides and functionally inverts the "if_select_hsic" input signal. If !(INV_SEL_HSIC, if_select_hsic) is:</p> <ul style="list-style-type: none"> • 00: HSIC Capability is disabled. • 01: HSIC Capability is enabled. • 10: HSIC Capability is enabled. • 11: HSIC Capability is disabled. <p>If the controller operates as non-HSIC-capable, it can only connect to non-HSIC-capable PHYs. If it operates as HSIC-capable, it can connect to HSIC-capable PHYs.</p> <p>This bit is reserved if the DWC_USB3_ENABLE_HSIC parameter is set to '0'. When selecting the HSIC feature, set the host side to HSIC mode first, then set the device mode side. If the device side is set to HSIC mode first and if the host does not see a connection in HSIC mode, then you must de-select the device HSIC mode and select it again using the if_select_hsic setting or register bit GUSB2PHYCFGn[26] to ensure that the device can connect to the host.</p> <p>Value After Reset: 0x0</p> <p>WriteConstraint: writeAsRead</p>
25	OVRD_FSLS_DISC_TIME	R/W	<p>Overriding the FS/LS disconnect time to 32us.</p> <ul style="list-style-type: none"> • If this value is 0, the FS/LS disconnect time is set to 2.5us as per the USB specification.

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> If this value is non-0, the disconnect detection time is set to 32us. <p>Normally, this value is set to 0. However, if the USB 2.0 PHYs introduce noise on the UTMI linestate and cause SE0 glitches longer than 2.5us, then a false disconnect condition may get triggered. To avoid interoperability issues with these PHYs, this bit can be set to 1.</p> <p>Value After Reset: 0x0</p>
24:22	LSTRD	R/W	<p>LS Turnaround Time (LSTRDTIM)</p> <p>This field indicates the value of the Rx-to-Tx packet gap for LS devices. The encoding is as follows:</p> <ul style="list-style-type: none"> 0: 2 bit times 1: 2.5 bit times 2: 3 bit times 3: 3.5 bit times 4: 4 bit times 5: 4.5 bit times 6: 5 bit times 7: 5.5 bit times <p>Note:</p> <ul style="list-style-type: none"> This field is applicable only in Host mode. For normal operation (to work with most LS devices), set the default value of this field to 3'h0 (2 bit times). The programmable LS device inter-packet gap and turnaround delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the Open LS mouse requires 3 bit times of inter-packet gap to work correctly. Include your PHY delays when programming the LSIPD/LSTRDTIM values. For example, if your PHY's TxEndDelay in LS mode is 30 UTMI/ULPI CLKs, then subtract this delay (~1 LS bit time) from the device's delay requirement. <p>Value After Reset: 0x0</p>
21:19	LSIPD	R/W	<p>LS Inter-Packet Time (LSIPD)</p> <p>This field indicates the value of Tx-to-Tx packet gap for LS devices. The encoding is as follows:</p> <ul style="list-style-type: none"> 0: 2 bit times 1: 2.5 bit times 2: 3 bit times 3: 3.5 bit times

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> • 4: 4 bit times • 5: 4.5 bit times • 6: 5 bit times • 7: 5.5 bit times <p>Note:</p> <ul style="list-style-type: none"> • This field is applicable only in Host mode. • For normal operation (to work with most LS devices), set the default value of this field to 3'h2 (3 bit times). • The programmable LS device inter-packet gap and turnaround delays are provided to support some legacy LS devices that might require different delays than the default/fixed ones. For instance, the AOpen LS mouse requires 3 bit times of inter-packet gap to work correctly. • Include your PHY delays when programming the LSIPD/LSTRDTIM values. For example, if your PHY's TxEndDelay in LS mode is 30 UTMII/ULPI CLKs, then subtract this delay (~1 LS bit time) from the device's delay requirement. <p>Value After Reset: 0x2</p>
18	ULPIEXTVBUSINDIATOR	R/W	<p>ULPI External VBUS Indicator (ULPIExtVbusIndicator)</p> <p>Indicates the ULPI PHY VBUS over-current indicator.</p> <ul style="list-style-type: none"> • 1'b0: PHY uses an internal VBUS valid comparator. • 1'b1: PHY uses an external VBUS valid comparator. <p>Valid only when RTL parameter DWC_USB3_HSPHY_INTERFACE = 2 or 3</p> <p>Value After Reset: 0x0</p>
17	ULPIEXTVBUSDRV	R/W	<p>ULPI External VBUS Drive (ULPIExtVbusDrv)</p> <p>Selects supply source to drive 5V on VBUS, in the ULPI PHY.</p> <ul style="list-style-type: none"> • 1'b0: PHY drives VBUS with internal charge pump (default). • 1'b1: PHY drives VBUS with an external supply. <p>(Only when RTL parameter DWC_USB3_HSPHY_INTERFACE = 2 or 3)</p> <p>Value After Reset: 0x0</p>
16	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
15	ULPIAUTORES	R/W	ULPI Auto Resume (ULPIAutoRes)

Bits	Name	Memory Access	Description
			<p>Sets the AutoResume bit in Interface Control register on the ULPI PHY.</p> <ul style="list-style-type: none"> • 1'b0: PHY does not use the AutoResume feature. • 1'b1: PHY uses the AutoResume feature. <p>Valid only when RTL parameter DWC_USB3_HSPHY_INTERFACE = 2 or 3</p> <p>Value After Reset: 0x0</p>
14	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
13:10	USBTRDTIM	R/W	<p>USB 2.0 Turnaround Time (USBTrdTim)</p> <p>Sets the turnaround time in PHY clocks.</p> <p>Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM).</p> <p>The following are the required values for the minimum SoC bus frequency of 60 MHz. USB turnaround time is a critical certification criteria when using long cables and five hub levels.</p> <p>The required values for this field:</p> <ul style="list-style-type: none"> • 4'h5: When the MAC interface is 16-bit UTMI+. • 4'h9: When the MAC interface is 8-bit UTMI+/ULPI. <p>If SoC bus clock is less than 60 MHz, and USB turnaround time is not critical, this field can be set to a larger value.</p> <p>Note: This field is valid only in device mode.</p> <p>Value After Reset: 0x9</p>
9	XCVRDLY	R/W	<p>Transceiver Delay:</p> <p>Enables a delay between the assertion of the UTMI/ULPI Transceiver Select signal (for HS) and the assertion of the TxValid signal during a HS Chirp.</p> <p>When this bit is set to 1, a delay (of approximately 2.5 us) is introduced from the time when the Transceiver Select is set to 2'b00 (HS) to the time the TxValid is driven to 0 for sending the chirp-K. This delay is required for some UTMI/ULPI PHYs.</p> <p>Note:</p> <ul style="list-style-type: none"> • If you enable the hibernation feature when the device controller comes out of power-off, you must re-initialize this bit with the appropriate value because the controller does not save and restore this bit value during hibernation. • This bit is valid only in device mode.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
8	ENBLSLPM	R/W	<p>Enable utmi_sleep_n and utmi_l1_suspend_n (EnbISlpM)</p> <p>The application uses this bit to control utmi_sleep_n and utmi_l1_suspend_n assertion to the PHY in the L1 state.</p> <ul style="list-style-type: none"> • 1'b0: utmi_sleep_n and utmi_l1_suspend_n assertion from the controller is not transferred to the external PHY. • 1'b1: utmi_sleep_n and utmi_l1_suspend_n assertion from the controller is transferred to the external PHY. <p>Note: This bit must be set high for Port0 if ESWIN PHY is used.</p> <p>Note: In Device mode - Before issuing any device endpoint command when operating in 2.0 speeds, disable this bit and enable it after the command completes. Without disabling this bit, if a command is issued when the device is in L1 state and if mac2_clk (utmi_clk/ulpi_clk) is gated off, the command will not get completed.</p> <p>Value After Reset: 0x0</p>
7	PHYSEL	W	<p>USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select</p> <p>The application uses this bit to select a high-speed PHY or a full-speed transceiver.</p> <ul style="list-style-type: none"> • 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY. This bit is always 0, with Write Only access. • 1'b1: USB 1.1 full-speed serial transceiver. This bit is always 1, with Write Only access. <p>If both interface types are selected in coreConsultant (that is, parameters' values are not zero), the application uses this bit to select the active interface is active, with Read-Write bit access.</p> <p>Note: USB 1.1 full-serial transceiver is not supported. This bit always reads as 1'b0.</p> <p>Value After Reset: 0x0</p>
6	SUSPENDUSB20	R/W	<p>Suspend USB2.0 HS/FS/LS PHY (SusPHY)</p> <p>When set, USB2.0 PHY enters Suspend mode if Suspend conditions are valid.</p> <p>For DRD configurations, it is recommended that this bit is set to 0 during coreConsultant configuration. If it is set to 1, then the application must clear this bit after power-on reset. Application needs to set it to 1 after the controller initialization completes.</p> <p>For all other configurations, this bit can be set to 1 during controller configuration.</p> <p>Note:</p> <ul style="list-style-type: none"> • In host mode, on reset, this bit is set to 1. Software can override this bit after reset. • In device mode, before issuing any device endpoint command when operating in 2.0 speeds, disable this bit and enable it after the command completes. If you issue a

Bits	Name	Memory Access	Description
			<p>command without disabling this bit when the device is in L2 state and if mac2_clk (utmi_clk/ulpi_clk) is gated off, the command will not get completed.</p> <p>Value After Reset: 0x0</p>
5	FSINTF	R	<p>Full-Speed Serial Interface Select (FSIntf)</p> <p>The application uses this bit to select a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface.</p> <ul style="list-style-type: none"> • 1'b0: 6-pin unidirectional full-speed serial interface. This bit is set to 0 with Read Only access. • 1'b1: 3-pin bidirectional full-speed serial interface. This bit is set to 0 with Read Only access. <p>Note: USB 1.1 full-speed serial interface is not supported. This bit always reads as 1'b0.</p> <p>Value After Reset: 0x0</p> <p>WriteConstraint: writeAsRead</p>
4	ULPI_UTMI_Sel	R/W	<p>ULPI or UTMI+ Select (ULPI_UTMI_Sel)</p> <p>The application uses this bit to select a UTMI+ or ULPI Interface.</p> <ul style="list-style-type: none"> • 1'b0: UTMI+ Interface • 1'b1: ULPI Interface <p>This bit is writable only if UTMI+ and ULPI is specified for High-Speed PHY Interface(s) in coreConsultant configuration (DWC_USB3_HSPHY_INTERFACE = 3).</p> <p>Otherwise, this bit is read-only and the value depends on the interface selected through DWC_USB3_HSPHY_INTERFACE.</p> <p>Value After Reset: 0x0</p> <p>WriteConstraint: writeAsRead</p>
3	PHYIF	R/W	<p>PHY Interface (PHYIf)</p> <p>If UTMI+ is selected, the application uses this bit to configure the controller to support a UTMI+ PHY with an 8- or 16-bit interface.</p> <ul style="list-style-type: none"> • 1'b0: 8 bits • 1'b1: 16 bits <p>ULPI Mode: 1'b0</p> <p>Note:</p> <ul style="list-style-type: none"> • All the enabled 2.0 ports must have the same clock frequency as Port0 clock frequency (utmi_clk[0]). • The UTMI 8-bit and 16-bit modes cannot be used together for different ports at the same time (that is, all the ports must be in 8-bit mode, or all of them must be in 16-bit mode, at a time). • If any of the USB 2.0 ports is selected as ULPI port for operation, then all the USB 2.0 ports must be operating at

Bits	Name	Memory Access	Description
			<p>60 MHz.</p> <p>Value After Reset: 0x0</p>
2:0	TOutCal	R/W	<p>HS/FS Timeout Calibration (TOutCal)</p> <p>The number of PHY clocks, as indicated by the application in this field, is multiplied by a bit-time factor; this factor is added to the high-speed/full-speed interpacket timeout duration in the controller to account for additional delays introduced by the PHY. This may be required, since the delay introduced by the PHY in generating the linestate condition may vary among PHYs.</p> <p>The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of connection. The number of bit times added per PHY clock are:</p> <p>High-speed operation:</p> <ul style="list-style-type: none"> • One 30-MHz PHY clock = 16 bit times • One 60-MHz PHY clock = 8 bit times <p>Full-speed operation:</p> <ul style="list-style-type: none"> • One 30-MHz PHY clock = 0.4 bit times • One 60-MHz PHY clock = 0.2 bit times • One 48-MHz PHY clock = 0.25 bit times <p>Value After Reset: 0x0</p>

USB_GUSB3PIPECTL

- **Description:** Global USB 3.0 PIPE Control Register. The application uses this register to configure the USB3 PHY and PIPE interface.
- **Size:** 32 bits
- **Offset:** 0xc2c0

Bits	Name	Memory Access	Description
31	PHYSoftRst	R/W	<p>USB3 PHY Soft Reset</p> <p>After setting this bit to '1', the software needs to clear this bit.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p>
30	HstPrtCmpl	R/W	<p>HstPrtCmpl</p> <p>This feature tests the PIPE PHY compliance patterns without having to have a test fixture on the USB 3.0 cable.</p> <p>This bit enables placing the SS port link into a compliance state. By default, this bit must be set to 1'b0.</p> <p>In compliance lab testing, the SS port link enters compliance after failing the first polling sequence after power on. Set this bit to 0, when you run compliance tests.</p>

Bits	Name	Memory Access	Description
			<p>The sequence for using this functionality is as follows:</p> <ul style="list-style-type: none"> • 1. Disconnect any plugged in devices. • 2. Perform USBCMD.HCRST or power-on-chip reset. • 3. Set PORTSC.PLS=0xA. • 4. Set PORTSC.PP=0. • 5. Set GUSB3PIPECTL. HstPrtCmpl=1. This places the link into compliance state. <p>To advance the compliance pattern, follow this sequence (toggle the set GUSB3PIPECTL. HstPrtCmpl):</p> <ul style="list-style-type: none"> • 1. Set GUSB3PIPECTL.HstPrtCmpl=0. • 2. Set GUSB3PIPECTL.HstPrtCmpl=1. <p>This advances the link to the next compliance pattern.</p> <p>To exit from the compliance state perform USBCMD.HCRST or power-on-chip reset.</p> <p>Value After Reset: 0x0</p>
29	U2P3ok	R/W	<p>P3 OK for SSInactive (SSIP3ok)</p> <ul style="list-style-type: none"> • 0: During link state SS.Inactive, put PHY in P2 (Default) • 1: During link state SS.Inactive, put PHY in P3. <p>Value After Reset: 0x0</p>
28	DisRxDetP3	R/W	<p>Disabled receiver detection in P3 (DisRxDetP3)</p> <ul style="list-style-type: none"> • 0: If PHY is in P3 and controller needs to perform receiver detection, The controller performs receiver detection in P3. (Default) • 1: If PHY is in P3 and controller needs to perform receiver detection, The controller changes the PHY power state to P2 and then performs receiver detection. After receiver detection, the cores changes PHY power state to P3. <p>Value After Reset: 0x0</p>
27	Ux_exit_in_Px	R/W	<p>Ux Exit in Px (Ux_exit_in_Px)</p> <ul style="list-style-type: none"> • 0: The controller does U1/U2/U3 exit in PHY power state P0 (default behavior). • 1: The controller does U1/U2/U3 exit in PHY power state P1/P2/P3 respectively. <p>Note:It must be set to '0'.</p> <p>Value After Reset: 0x0</p>
26	ping_enhancement_en	R/W	<p>Ping Enhancement Enable (ping_enhancement_en)</p> <p>When set, the Downstream port U1 ping receive timeout becomes 500 ms instead of 300 ms. Minimum Ping.LFPS receive duration is 8 ns (one mac3_clk). This field is valid for the downstream port only.</p>

Bits	Name	Memory Access	Description
			<p>Note: It must be set to '0'.</p> <p>Value After Reset: 0x0</p>
25	u1u2exitfail_to_recov	R/W	<p>U1U2exitfail to Recovery (u1u2exitfail_to_recov)</p> <p>When set, and U1/U2 LFPS handshake fails, the LTSSM transitions from U1/U2 to Recovery instead of SS Inactive. If Recovery fails, then the LTSSM can enter SS.Inactive. This is an enhancement only. It prevents interoperability issue if the remote link does not do proper handshake.</p> <p>Value After Reset: 0x0</p>
24	request_p1p2p3	R/W	<p>Always Request P1/P2/P3 for U1/U2/U3 (request_p1p2p3)</p> <p>When set, the controller always requests PHY power change from P0 to P1/P2/P3 during U0 to U1/U2/U3 transition.</p> <p>If this bit is 0, and immediate Ux exit (remotely initiated, or locally initiated) happens, the controller does not request P1/P2/P3 power state change.</p> <p>Note: This bit must be set to '1'.</p> <p>Value After Reset: 0x1</p>
23	StartRxDetU3RxDet	W	<p>Start Receiver Detection in U3/Rx.Detect (StartRxdetU3RxDet)</p> <p>If DWC_USB3_GUSB3PIPECTL_INIT[22] is set, and the link is in either U3 or Rx.Detect state, the controller starts receiver detection on the rising edge of this bit. This can only be used for Downstream ports. This bit must be set to '0' for Upstream ports. This feature must not be enabled for normal operation.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p>
22	DisRxDetU3RxDet	R/W	<p>Disable Receiver Detection in U3/Rx.Det</p> <p>When set, the controller does not handle receiver detection in either U3 or Rx.Detect states. DWC_USB3_GUSB3PIPECTL_INIT[23] must be used to start receiver detection manually. This bit can only be used for the downstream port. This bit must be set to "0" for Upstream ports. This feature must not be enabled for normal operation.</p> <p>Value After Reset: 0x0</p>
21:19	DelayP1P2P3	R/W	<p>Delay P1P2P3</p> <p>Delay P0 to P1/P2/P3 request when entering U1/U2/U3 until (DWC_USB3_GUSB3PIPECTL_INIT[21:19]*8) 8B10B error occurs, or Pipe3_RxValid drops to 0.</p> <p>DWC_USB3_GUSB3PIPECTL_INIT[18] must be 1 to enable this functionality.</p> <p>Value After Reset: 0x1</p>
18	DELAYP1TRANS	R/W	<p>Delay PHY power change from P0 to P1/P2/P3 when link state changing from U0 to U1/U2/U3 respectively.</p> <ul style="list-style-type: none"> • 1'b1: When entering U1/U2/U3, delay the transition to P1/P2/P3 until the pipe3 signals, Pipe3_RxEleIdle is 1 and pipe3_RxValid is 0 • 1'b0: When entering U1/U2/U3, transition to P1/P2/P3 without checking for Pipe3_RxEleIdle and pipe3_RxValid.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
17	SUSPENDENABLE	R/W	<p>Suspend USB3.0 SS PHY (Suspend_en)</p> <p>When set, and if Suspend conditions are valid, the USB 3.0 PHY enters Suspend mode.</p> <p>For DRD configurations, it is recommended that this bit is set to '0' during coreConsultant configuration. If it is set to '1', then the application must clear this bit after power-on reset. Application needs to set it to '1' after the controller initialization is completed.</p> <p>For all other configurations, this bit can be set to '1' during controller configuration.</p> <p>Value After Reset: 0x0</p>
16:15	DATWIDTH	R	<p>PIPE Data Width (DatWidth)</p> <ul style="list-style-type: none"> 2'b00: 32 bits <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
14	AbortRxDetInU2	R/W	<p>Abort Rx Detect in U2 (AbortRxDetInU2)</p> <p>When set and the link state is U2, the controller will abort receiver detection if it receives U2 exit LFPS from the remote link partner. This bit is for the downstream port only.</p> <p>Note: It must be set to '0'.</p> <p>Value After Reset: 0x0</p>
13	SkipRxDet	R/W	<p>Skip Rx Detect:</p> <p>When set, the controller skips Rx Detection if pipe3_RxEleidle is low.</p> <p>Skip is defined as waiting for the appropriate timeout, then repeating the operation.</p> <p>Value After Reset: 0x0</p>
12	LFPSPOAlign	R/W	<p>LFPS P0 Align.</p> <p>Not Use.</p> <p>Value After Reset: 0x0</p>
11	P3P2TranOK	R/W	<p>P3 P2 Transitions OK (P3P2TranOK)</p> <p>When set, the controller transitions directly from Phy power state P2 to P3 or from state P3 to P2. When not set, P0 is always entered as an intermediate state during transitions between P2 and P3, as defined in the <i>PIPE3 Specification</i>.</p> <p>According to the <i>PIPE3 Specification</i>, any direct transition between P3 and P2 is illegal.</p> <p>Note: It must be set to '0'.</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
10	P3ExSigP2	R/W	<p>P3 Exit Signal in P2 (P3ExSigP2)</p> <p>When this bit is set, the controller always changes the PHY power state to P2, before attempting a U3 exit handshake.</p> <p>Note: It must be set to '0'.</p> <p>Value After Reset: 0x0</p>
9	LFPSFILTER	R/W	<p>LFPS Filter (LFPSFilt)</p> <p>When set, filter LFPS reception with pipe3_RxValid in PHY power state P0, that is, ignore LFPS reception from the PHY unless both pipe3_Rxelecidle and pipe3_RxValid are deasserted.</p> <p>Value After Reset: 0x0</p>
8	RX_DETECT_to_Polling_LFPS_Control	R/W	<p>RX_DETECT to Polling.LFPS Control</p> <ul style="list-style-type: none"> • 1'b0 (Default): Enables a 400us delay to start Polling LFPS after RX_DETECT. This allows VCM offset to settle to a proper level. • 1'b1: Disables the 400us delay to start Polling LFPS after RX_DETECT. <p>During controller certification with third party PHY it is observed that the PHY is not able to meet the Tx AC common mode voltage active (VTX-CM-ACPP_ACTIVE <100mv) if the link starts polling within 80us from the time rx.detect is performed.</p> <p>To meet this VTX-CM-ACPP_ACTIVE specification, the polling must be delayed further. If the PHY does not have issue then they can set this bit to 1 which allows polling to start within 80us.</p> <p>Value After Reset: 0x0</p>
7	SSICEn	R/W	<p>This field is not used.</p> <p>Value After Reset: 0x0</p>
6	TX_SWING	R/W	<p>Tx Swing (TxSwing)</p> <p>Refer to the <i>PIPE3 specification</i>.</p> <p>Value After Reset: 0x0</p>
5:3	TX_MARGIN	R/W	<p>Tx Margin[2:0] (TxMargin)</p> <p>Refer to Table 5-3 of the <i>PIPE3 Specification</i>.</p> <p>Value After Reset: 0x0</p>
2:1	SS_TX_DE_EMPHASIS	R/W	<p>Tx Deemphasis (TxDeemphasis)</p> <p>The value driven to the PHY is controlled by the LTSSM during USB3 Compliance mode.</p> <p>(Refer to Table 5-3 of the <i>PIPE3 specification</i>.)</p> <p>Value After Reset: 0x1</p>
0	ELASTIC_BUFFER_MODE	R/W	<p>Elastic Buffer Mode (ElasticBufferMode)</p> <p>(Refer to Table 5-3 of the <i>PIPE3 specification</i>.)</p> <p>Value After Reset: 0x0</p>

USB_GTXFIFOSIZ

- **Description:** Global Transmit FIFO Size Register
- This register specifies the RAM start address and depth (both in 32bit) for each implemented TxFIFO. The number of TxFIFOs depends on the configuration parameters including the number of Device IN Endpoints, number of Host Bus Instances, and presence of Debug Capability. The register default values for each mode are based on the maximum packet size, number of packets to be buffered, speed of host bus instance, bus latency, and mode of operation (host, device, or, DBC). Upon reset and mode transitions, hardware automatically programs these registers to the default values. Consequently, there is typically no need for the software to modify the pre-defined default values.
- **Size:** 32 bits
- **Offset:** 0xc300 + ($i \times 0x40$), where $i = 0..7$

Bits	Name	Memory Access	Description
31:16	TXFSTADDR_N	R/W	<p>Transmit FIFOOn RAM Start Address</p> <p>This field contains the memory start address for TxFIFOn in 32bit.</p> <p>Value After Reset: 0x0</p>
15:0	TXFDEP_N	R/W	<p>TxFIFO Depth</p> <p>This field contains the depth of TxFIFOn in 32bit.</p> <ul style="list-style-type: none"> • Minimum value: 32 • Maximum value: 32,768 <p>Value After Reset: 0x82</p>

USB_GRXFIFOSIZ

- **Description:** Global Receive FIFO Size Register
- This register specifies the RAM start address and depth (both in 32bit) for each implemented Rx FIFO. The number of Rx FIFOs depends on the configuration parameters including the number of Host Bus Instances and presence of Debug Capability; device mode requires only one Rx FIFO. The register default values for each mode are based on the maximum packet size, number of packets to be buffered, speed of the host bus instance, bus latency, and mode of operation (host, device, or DBC). Upon reset and mode transitions, hardware automatically programs these registers to the default values. Consequently, there is typically no need for the software to modify the pre-defined default values.
- **Size:** 32 bits
- **Offset:** 0xc380 + ($i \times 0x40$), where $i = 0..2$

Bits	Name	Memory Access	Description
31:16	RXFSTADDR_N	R/W	<p>RxFIFOOn RAM Start Address (RxFStAddr_n)</p> <p>This field contains the memory start address for RxFIFOn in 32bit.</p> <p>Value After Reset: 0x0</p>
15:0	RXFDEP_N	R/W	<p>RxFIFO Depth (RxFDep_n)</p> <p>This field contains the depth of RxFIFOn in 32bit .</p> <ul style="list-style-type: none"> • Minimum value: 32 • Maximum value: 16,384 <p>Value After Reset: 0x30a</p>

USB_GEVNTADRLO

- **Description:** Global Event Buffer Address (Low) Register
- **Size:** 32 bits
- **Offset:** 0xc400

Bits	Name	Memory Access	Description
31:0	EVNTADRLO	R/W	<p>Event Buffer Address (EvntAdrLo)</p> <p>Holds the lower 32 bits of start address of the external memory for the Event Buffer. During operation, hardware does not update this address.</p> <p>Value After Reset: 0x0</p> <p>Reset Mask: 0xffffffff</p> <p>Volatile: true</p>

USB_GEVNTADRHI

- **Description:** Global Event Buffer Address (High) Register
- **Size:** 32 bits
- **Offset:** 0xc404

Bits	Name	Memory Access	Description
31:0	EVNTADRHI	R/W	<p>Event Buffer Address (EvntAdrHi)</p> <p>Holds the higher 32 bits of start address of the external memory for the Event Buffer. During operation, hardware does not update this address.</p> <p>Value After Reset: 0x0</p> <p>Reset Mask: 0xffffffff</p> <p>Volatile: true</p>

USB_GEVNTSIZ

- **Description:** Global Event Buffer Size Register
- This register holds the Event Buffer Size and the Event Interrupt Mask bit. During power-on initialization, software must initialize the size with the number of bytes allocated for the Event Buffer. The Event Interrupt Mask will mask the interrupt, but events are still queued. After configuration, software must preserve the Event Buffer Size value when changing the Event Interrupt Mask.
- **Size:** 32 bits
- **Offset:** 0xc408

Bits	Name	Memory Access	Description
31	EVNTINTRPTMASK	R/W	<p>Event Interrupt Mask (EvntIntMask).</p> <p>When set to '1', this prevents the interrupt from being generated. However, even when the mask is set, the events are queued.</p> <p>Value After Reset: 0x0</p>
30:16	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p>

Bits	Name	Memory Access	Description
			<p>Reset Mask: 0x7fff</p> <p>WriteConstraint: writeAsRead</p>
15:0	EVENTSIZ	R/W	<p>Event Buffer Size in bytes (EVNTSiz)</p> <p>Holds the size of the Event Buffer in bytes; must be a multiple of four. This is programmed by software once during initialization. The minimum size of the event buffer is 32 bytes.</p> <p>Value After Reset: 0x0</p>

USB_GEVNTCOUNT

- **Description:** Global Event Buffer Count Register
- This register holds the number of valid bytes in the Event Buffer. During initialization, software must initialize the count by writing 0 to the Event Count field. Each time the hardware writes a new event to the Event Buffer, it increments this count. Most events are four bytes, but some events may span over multiple four byte entries. Whenever the count is greater than zero and if enabled, conditions for interrupt moderation are satisfied, the hardware raises the corresponding interrupt line (depending on the EvntIntMask bit in the USB_GEVNTSIZn register). On an interrupt, software processes one or more events out of the Event Buffer. Afterwards, software must write the Event Count field with the number of bytes it processed. If Interrupt Moderation is enabled, then software needs to clear EVNT_HANDLER_BUSY bit. Clock crossing delays may result in the continuous assertion of the interrupt after software acknowledges the last event. Therefore, when the interrupt line is asserted, software must read the USB_GEVNTCOUNT register and only process events if the USB_GEVNTCOUNT is greater than 0.
- **Size:** 32 bits
- **Offset:** 0xc40c

Bits	Name	Memory Access	Description
31	EVNT_HANDLER_BUSY	R/W	<p>Event Handler Busy</p> <p>Device software event handler busy indication. The controller sets this bit when the interrupt line is asserted due to pending events. Software clears this bit (with 1'b1) when it has finished processing the events (along with updating the EVNTCOUNT in this register). The controller does not raise the interrupt line for a new event unless this bit is cleared.</p> <p>Note: When Interrupt moderation is disabled (that is, DEVICE_IMODI = 0), this bit is ignored.</p> <p>Value After Reset: 0x0</p> <p>Reset Mask: 0x1</p> <p>Volatile: true</p>
30:16	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x7fff</p> <p>WriteConstraint: writeAsRead</p>
15:0	EVNTCOUNT	R/W	Event Count (EVNTCount)

Bits	Name	Memory Access	Description
			<p>When read, returns the number of valid events in the Event Buffer (in bytes). When written, hardware decrements the count by the value written.</p> <p>When Interrupt moderation is enabled (that is, DEVICE_IMODI!= 0), the interrupt line gets de-asserted when the first write happens on this register to decrement the count. When Interrupt moderation is disabled (that is, DEVICE_IMODI = 0), the Interrupt line continues to get asserted until the event count becomes zero (no-moderation behavior).</p> <p>Value After Reset: 0x0</p> <p>Reset Mask: 0xffff</p> <p>Volatile: true</p>

USB_GUCTL3

- **Description:** Global User Control Register 3
- This register provides a few options for the software to control the controller behavior in the Host mode. Most of the options are used to improve host inter-operability with different devices.
- **Size:** 32 bits
- **Offset:** 0xc60c

Bits	Name	Memory Access	Description
31:17	reserved	R/W	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xff</p> <p>WriteConstraint: writeAsRead</p>
16	Sch_Ping_early	R/W	<p>Enable SuperSpeed Ping Transaction Packet scheduling early in the microframe.</p> <p>This bit is valid in Host mode only.</p> <p>Value After Reset: 0x1</p>
15:0	reserved	R/W	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xff</p> <p>WriteConstraint: writeAsRead</p>

USB_GTXFIFOPRIDEV

- **Description:** Global Device TX FIFO DMA Priority Register
- This register specifies the relative DMA priority level among the Device TXFIFOs (one per IN endpoint). Each register bit[n] controls the priority (1: high, 0: low) of each TXFIFO[n]. When multiple TXFIFOs compete for DMA service at a given time (that is, multiple TXQs contain TX DMA requests and their corresponding TXFIFOs have space available), the TX DMA arbiter grants access on a packet-basis in the following manner:
 - 1. High-priority TXFIFOs are granted access using round-robin arbitration
 - 2. Low-priority TXFIFOs are granted access using round-robin arbitration only after the high-

priority TXFIFOs have no further processing to do (that is, either the TXQs are empty or the corresponding TXFIFOs are full).

- For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed.
- When configuring periodic IN endpoints, software must set register bit[n]=1, where n is the TXFIFO assignment. This ensures that the DMA for isochronous or interrupt IN endpoints are prioritized over bulk or control IN endpoints.
- This register is present only when the controller is configured to operate in the device mode (includes DRD mode). The register size corresponds to the number of Device IN endpoints.
- Note: Since the device mode uses only one RXFIFO, there is no Device RXFIFO DMA Priority Register.
- Size:** 32 bits
- Offset:** 0xc610

Bits	Name	Memory Access	Description
31:8	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xffff</p> <p>WriteConstraint: writeAsRead</p>
7:0	gtxfifo pri dev	R/W	<p>Device TxFIFO priority</p> <p>Value After Reset: 0x0</p>

USB_GTXFIFOPRIHST

- Description:** Global Host TX FIFO DMA Priority Register
- This register specifies the relative DMA priority level among the Host TXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FSLS). Each register bit[n] controls the priority (1: high, 0: low) of TXFIFO[n] within a speed group. When multiple TXFIFOs compete for DMA service at a given time (i.e., multiple TXQs contain TX DMA requests and their corresponding TXFIFOs have space available), the TX DMA arbiter grants access on a packet-basis in the following manner:
 - 1. Among the FIFOs in the same speed group (SS or HS/FSLS):
 - a. High-priority TXFIFOs are granted access using round-robin arbitration
 - b. Low-priority TXFIFOs are granted access using round-robin arbitration only after the high-priority TXFIFOs have no further processing to do (that is, either the TXQs are empty or the corresponding TXFIFOs are full).
 - 2. The TX DMA arbiter prioritizes the SS speed group or HS/FSLS speed group according to the ratio programmed in the USB_GDMAHLRATIO register.
- For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed.
- This register is present only when the controller is configured to operate in the host mode (includes DRD mode). The register size corresponds to the number of configured USB bus instances; for example, in the default configuration, there are 3 USB bus instances (1 SS, 1 HS, and 1 FSLS).
- Size:** 32 bits
- Offset:** 0xc618

Bits	Name	Memory Access	Description
31:3	reserved	R	Reserved

Bits	Name	Memory Access	Description
			Value After Reset: 0x0 Testable: writeAsRead Reset Mask: 0xffff WriteConstraint: writeAsRead
2:0	gtxfifoprihst	R/W	Host TxFIFO priority Value After Reset: 0x0

USB_GRXFIFOPRIHST

- **Description:** Global Host RX FIFO DMA Priority Register
- This register specifies the relative DMA priority level among the Host RXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FSLS). Each register bit[n] controls the priority (1: high, 0: low) of RXFIFO[n] within a speed group. When multiple RXFIFOs compete for DMA service at a given time (i.e., multiple RXQs contain RX DMA requests and their corresponding RXFIFOs have data available), the RX DMA arbiter grants access on a packet-basis in the following manner:
 - 1. Among the FIFOs in the same speed group (SS or HS/FSLS):
 - a. High-priority RXFIFOs are granted access using round-robin arbitration
 - b. Low-priority RXFIFOs are granted access using round-robin arbitration only after high-priority RXFIFOs have no further processing to do (that is, either the RXQs are empty or the corresponding RXFIFOs do not have the required data).
 - 2. The RX DMA arbiter prioritizes the SS speed group or HS/FSLS speed group according to the ratio programmed in the USB_GDMAHLRATIO register.
 - For scatter-gather packets, the arbiter grants successive DMA requests to the same FIFO until the entire packet is completed.
- This register is present only when the controller is configured to operate in the host mode (includes DRD mode). The register size corresponds to the number of configured USB bus instances; for example, in the default configuration, there are 3 USB bus instances (1 SS, 1 HS, and 1 FSLS).
- **Size:** 32 bits
- **Offset:** 0xc61c

Bits	Name	Memory Access	Description
31:3	reserved	R	Reserved Value After Reset: 0x0 Testable: writeAsRead Reset Mask: 0xffff WriteConstraint: writeAsRead
2:0	grxfifoprihst	R/W	Host RxFIFO priority Value After Reset: 0x0

USB_GDMAHLRATIO

- **Description:** Global Host FIFO DMA High-Low Priority Ratio Register. This register specifies the relative priority of the SS FIFOs with respect to the HS/FSLS FIFOs. The DMA arbiter prioritizes the HS/FSLS round-robin arbiter group every DMA High-Low Priority Ratio grants as indicated in the register separately for TX and RX.

- To illustrate, consider that all FIFOs are requesting access simultaneously, and the ratio is 4. SS gets priority for 4 packets, HS/FSLS gets priority for 1 packet, SS gets priority for 4 packets, HS/FSLS gets priority for 1 packet, and so on.
- If FIFOs from both speed groups are not requesting access simultaneously then, if SS got grants 4 out of the last 4 times, then HS/FSLS get the priority on any future request. if HS/FSLS got the grant last time, SS gets the priority on the next request. if there is a valid request on either SS or HS/FSLS, a grant is always awarded; there is no idle.
- This register is present if the controller is configured to operate in host mode (includes DRD).
- **Size:** 32 bits
- **Offset:** 0xc624

Bits	Name	Memory Access	Description
31:13	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x7ffff</p> <p>WriteConstraint: writeAsRead</p>
12:8	hstrxfifo	R/W	<p>Host RXFIFO DMA High-Low Priority</p> <p>Value After Reset: 0x8</p>
7:5	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x7</p> <p>WriteConstraint: writeAsRead</p>
4:0	hsttxfifo	R/W	<p>Host TXFIFO DMA High-Low Priority</p> <p>Value After Reset: 0x8</p>

USB_GFLADJ

- **Description:** Global Frame Length Adjustment Register. This register provides options for the software to control the controller behavior with respect to SOF (Start of Frame) and ITP (Isochronous Timestamp Packet) timers and frame timer functionality. It provides an option to override the fladj_30mhz_reg sideband signal. In addition, it enables running SOF or ITP frame timer counters completely from the ref_clk. This facilitates hardware LPM in host mode with the SOF or ITP counters being run from the ref_clk signal.
- **Size:** 32 bits
- **Offset:** 0xc630

Bits	Name	Memory Access	Description
31	GFLADJ_REFCLK_240MHZDECR_PLS1	R/W	<p>GFLADJ_REFCLK_240MHZDECR_PLS1</p> <p>This field indicates that the decrement value that the controller applies for each ref_clk must be GFLADJ_REFCLK_240MHZ_DECR and GFLADJ_REFCLK_240MHZ_DECR +1 alternatively on each ref_clk.</p> <p>Set this bit to a '1' only if GFLADJ_REFCLK_LPM_SEL is set to '1' and the fractional component of 240/ref_frequency is greater than or equal to 0.5.</p>

Bits	Name	Memory Access	Description
			<p>Examples:</p> <p>If the ref_clk is 19.2 MHz then</p> <ul style="list-style-type: none"> • GUCTL.REF_CLK_PERIOD = 52 • GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = $(240/19.2) = 12.5$ • GFLADJ.GFLADJ_REFCLK_240MHZDECR_PLS1 = 1 <p>If the ref_clk is 24 MHz then</p> <ul style="list-style-type: none"> • GUCTL.REF_CLK_PERIOD = 41 • GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = $(240/24) = 10$ • GFLADJ.GFLADJ_REFCLK_240MHZDECR_PLS1 = 0 <p>Value After Reset: 0x0</p>
30:24	GFLADJ_REFCLK_240MHZ_DECR	R/W	<p>This field indicates the decrement value that the controller applies for each ref_clk in order to derive a frame timer in terms of a 240-MHz clock.</p> <p>This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to '1'.</p> <p>The value is derived as follows:</p> $\text{GFLADJ_REFCLK_240MHZ_DECR} = 240/\text{ref_clk_frequency}$ <p>Examples: If the ref_clk is 24 MHz then</p> <ul style="list-style-type: none"> • GUCTL.REF_CLK_PERIOD = 41 • GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = $240/24 = 10$ <p>If the ref_clk is 48 MHz then</p> <ul style="list-style-type: none"> • GUCTL.REF_CLK_PERIOD = 20 • GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = $240/48 = 5$ <p>If the ref_clk is 17 MHz then</p> <ul style="list-style-type: none"> • GUCTL.REF_CLK_PERIOD = 58 • GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = $240/17 = 14$ <p>Value After Reset: 0x0</p>
23	GFLADJ_REFCLK_LPM_SEL	R/W	<p>Not Use.</p> <p>This bit enables the functionality of running SOF/ITP counters on the ref_clk.</p> <p>This bit must not be set to '1' if GCTL.SOFITPSYNC bit is set to '1'.</p> <p>Similarly, if GFLADJ_REFCLK_LPM_SEL set to '1', GCTL.SOFITPSYNC must not be set to '1'.</p> <p>In device mode, setting this bit to '1' enables SOF tracking using ref_clk.</p> <p>When GFLADJ_REFCLK_LPM_SEL is set to '1' the overloading of the suspend control of the USB 2.0 first port PHY (UTMI/ULPI) with USB 3.0 port states is removed.</p> <p>Note that the ref_clk frequencies supported in this mode are</p>

Bits	Name	Memory Access	Description
			<p>16/17/19.2/20/24/39.7/40 MHz. The utmi_clk[0] signal of the controller must be connected to the FREECLK of the PHY.</p> <p>Note: If you set this bit to '1', the GUSB2PHYCFG.U2_FREECLK_EXISTS bit must be set to '0'.</p> <p>Value After Reset: 0x0</p>
22	reserved	R	<p>Reserved for future use</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
21:8	GFLADJ_REFCLK_FLADJ	R/W	<p>This field indicates the frame length adjustment to be applied when SOF/ITP counter is running on the ref_clk.</p> <p>This register value is used to adjust the ITP interval when GCTL[SOFITPSYNC] is set to '1'; SOF and ITP interval when GLADJ.GFLADJ_REFCLK_LPM_SEL is set to '1'.</p> <p>This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to '1' or GCTL.SOFITPSYNC is set to '1'.</p> <p>The value is derived as follows:</p> $\text{FLADJ_REF_CLK_FLADJ} = ((125000/\text{ref_clk_period_integer}) - (125000/\text{ref_clk_period})) * \text{ref_clk_period}$ <ul style="list-style-type: none"> the ref_clk_period_integer is the integer value of the ref_clk period got by truncating the decimal (fractional) value that is programmed in the GUCTL.REF_CLK_PERIOD field. the ref_clk_period is the ref_clk period including the fractional value. <p>Examples: If the ref_clk is 24 MHz then</p> <ul style="list-style-type: none"> GUCTL.REF_CLK_PERIOD = 41 GFLADJ.GLADJ_REFCLK_FLADJ = $((125000/41) - (125000/41.6666)) * 41.6666 = 2032$ (ignoring the fractional value) <p>If the ref_clk is 48 MHz then</p> <ul style="list-style-type: none"> GUCTL.REF_CLK_PERIOD = 20 GFLADJ.GLADJ_REFCLK_FLADJ = $((125000/20) - (125000/20.8333)) * 20.8333 = 5208$ (ignoring the fractional value) <p>Value After Reset: 0x0</p>
7	GFLADJ_30MHZ_SD_BND_SEL	R/W	<p>GFLADJ_30MHZ_SDBND_SEL</p> <p>This field selects whether to use the input signal fladj_30mhz_reg or the GFLADJ.GFLADJ_30MHZ to adjust the frame length for the SOF/ITP. When this bit is set to,</p> <ul style="list-style-type: none"> 1, the controller uses the register field GFLADJ.GFLADJ_30MHZ value

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> • 0, the controller uses the input signal fladj_30mhz_reg value <p>Value After Reset: 0x0</p>
6	reserved	R	<p>Reserved for future use</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
5:0	GFLADJ_30MHZ	R/W	<p>GFLADJ_30MHZ</p> <p>This field indicates the value that is used for frame length adjustment instead of considering from the sideband input signal fladj_30mhz_reg.</p> <p>This enables post-silicon frame length adjustment in case the input signal fladj_30mhz_reg is connected to a wrong value or is not valid.</p> <p>For details on how to set this value, refer to section 5.2.4, "Frame Length Adjustment Register (FLADJ)," of the <i>xHCI Specification</i>.</p> <p>Value After Reset: 0x0</p>

USB_GUSB2RHBCTL

- **Description:** Global USB 2.0 Root Hub Control Register. The application must program this register before starting any transactions on the USB if a non-default value is desired.
- **Size:** 32 bits
- **Offset:** 0xc640

Bits	Name	Memory Access	Description
31:4	Reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x0</p>
3:0	OVRD_L1TIMEOUT	R/W	<p>Overriding the driver programmed L1TIMEOUT value.</p> <p>If this value is 0, the L1 Timeout value is taken from the xHCI PORTLHLMC register. If this value is non-0, then this will override the L1 Timeout value programmed in the xHCI PORTLHLMC register. In that case the actual L1 Timeout would be $2^{<\text{OVRD_L1TIMEOUT-1}>} * 8\text{us}$. (1=8us, 2=16us, 3=32us etc)</p> <p>Value After Reset: 0x0</p>

USB_DCFG

- **Description:** Device Configuration Register. This register configures the controller in Device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.
- **Size:** 32 bits
- **Offset:** 0xc700

Bits	Name	Memory Access	Description
31:25	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x3f</p> <p>WriteConstraint: writeAsRead</p>
24	reserved	R	<p>Reserved Field: Yes</p> <p>Value After Reset: 0x0</p>
23	IgnStrmPP	R/W	<p>IgnoreStreamPP This bit only affects stream-capable bulk endpoints.</p> <p>When this bit is set to '0' and the controller receives a Data Packet with the Packet Pending (PP) bit set to 0 for OUT endpoints, or it receives an ACK with the NumP field set to 0 and PP set to 0 for IN endpoints, the controller attempts to search for another stream (CStream) to initiate to the host. However, there are two situations where this behavior is not optimal:</p> <ul style="list-style-type: none"> • When the host is setting PP=0 even though it has not finished the stream, or • When the endpoint on the device is configured with one transfer resource and therefore does not have any other streams to initiate to the host. <p>When this bit is set to '1', the controller ignores the Packet Pending bit for the purposes of stream selection and does not search for another stream when it receives DP(PP=0) or ACK(NumP=0, PP=0). This can enhance the performance when the device system bus bandwidth is low or the host responds to the controller's ERDY transmission very quickly.</p> <p>Value After Reset: 0x0</p>
22	LPMCAP	R/W	<p>LPM Capable</p> <p>The application uses this bit to control the LPM capabilities of the DWC_usb3 controller. If the controller operates as a non-LPM-capable device, it cannot respond to LPM transactions.</p> <ul style="list-style-type: none"> • 1'b0: LPM capability is not enabled. • 1'b1: LPM capability is enabled. <p>Value After Reset: 0x0</p>
21:17	NUMP	R/W	<p>Number of Receive Buffers.</p> <p>This bit indicates the number of receive buffers to be reported in the ACK TP.</p> <p>The DWC_usb3 controller uses this field for non-control endpoints if GRXTHRCFG.UsbRxPktCntSel is set to '0'. The application can program this value based on RxFIFO size, buffer sizes programmed in descriptors, and system latency.</p> <p>For an OUT endpoint, this field controls the number of receive buffers reported in the NumP field of the ACK TP transmitted by the controller.</p> <p>Note: This bit is used in host mode when Debug Capability is enabled.</p> <p>Value After Reset: 0x4</p>
16:12	INTRNUM	R/W	<p>Interrupt number</p> <p>Indicates interrupt/EventQ number on which non-endpoint-specific device-related interrupts (see DEVT) are generated.</p>

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
11:10	reserved	R	Reserved Value After Reset: 0x2 Testable: writeAsRead Reset Mask: 0x3 WriteConstraint: writeAsRead
9:3	DEVADDR	R/W	Device Address. The application must perform the following: <ul style="list-style-type: none"> Program this field after every SetAddress request. Reset this field to zero after USB reset. Value After Reset: 0x0
2:0	DEVSPD	R/W	Device Speed. Indicates the speed at which the application requires the controller to connect, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the controller is connected. <ul style="list-style-type: none"> 3'b100: SuperSpeed (USB 3.0 PHY clock is 125 MHz or 250 MHz) 3'b000: High-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 3'b001: Full-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) Values: <ul style="list-style-type: none"> 0x4 (SuperSpeed): SuperSpeed (USB 3.0 PHY clock is 125 MHz or 250 MHz) 0x0 (HighSpeed): High-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 0x1 (FullSpeed): Full-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) Value After Reset: 0x4 Testable: unconstrained

USB_DCTL

- Description:** Device Control Register. **Note:** When Hibernation is not enabled using GCTL.GblHibernationEn field, you can write any value to CSS, CRS, L1HibernationEn, and KeepConnect fields. L1HibernationEn, and KeepConnect fields always return 0 when read in this hibernation-disabled state.
- Size:** 32 bits
- Offset:** 0xc704

Bits	Name	Memory Access	Description
31	RUN_STOP	R/W	Run/Stop The software writes 1 to this bit to start the device controller operation.

Bits	Name	Memory Access	Description
			<p>To stop the device controller operation, the software must remove any active transfers and write 0 to this bit. When the controller is stopped, it sets the DSTS.DevCtrlHlt bit when the controller is idle and the lower layer finishes the disconnect process.</p> <p>The Run/Stop bit must be used in following cases as specified:</p> <ul style="list-style-type: none"> After power-on reset and CSR initialization, the software must write 1 to this bit to start the device controller. The controller does not signal connect to the host until this bit is set. The software uses this bit to control the device controller to perform a soft disconnect. When the software writes 0 to this bit, the host does not see that the device is connected. The device controller stays in the disconnected state until the software writes 1 to this bit. The minimum duration of keeping this bit cleared is specified in the Note below. If the software attempts a connect after the soft disconnect or detects a disconnect event, it must set DCTL[8:5] to 5 before reasserting the Run/Stop bit. When the USB or Link is in a lower power state and the Two Power Rails configuration is selected, software writes 0 to this bit to indicate that it is going to turn off the Core Power Rail. After the software turns on the Core Power Rail again and re-initializes the device controller, it must set this bit to start the device controller. For more details, see "Low Power Operation" section of the Databook. <p>Note: The following is the minimum duration under various conditions for which the soft disconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect:</p> <p><i>30ms:</i></p> <ul style="list-style-type: none"> For SuperSpeed, when the device state is Suspended, Idle, Transmit, or Receive. <p><i>10ms:</i></p> <ul style="list-style-type: none"> For high-speed, when the device state is Suspended, Idle, or not Idle/Suspended (performing transactions) For full-speed/low-speed, when the device state is Suspended, Idle, or not Idle/Suspended (performing transactions) <p>To accommodate clock jitter, it is recommended that the application add extra delay to the specified minimum duration.</p> <p>Value After Reset: 0x0</p>
30	CSFTRST	R/W1S	<p>Core Soft Reset</p> <p>Resets the all clock domains as follows:</p> <ul style="list-style-type: none"> This bit clears the interrupts and all the CSRs except GSTS, GSNSID, GPIO, GUID, GUSB2PHYCFGn registers, GUSB3PIPECTLn registers, DCFG, DCTL, DEVREN, and DSTS registers. All module state machines (except the SoC Bus Slave Unit) are reset to the IDLE state, and all the TxFIFOs and the RxFIFO are flushed. Any transactions on the SoC bus Master are terminated as soon as possible, after gracefully completing the last data phase of a SoC bus transfer. Any transactions on the USB are terminated immediately. <p>The application can write this bit at any time to reset the controller. This is a self-</p>

Bits	Name	Memory Access	Description
			<p>clearing bit; the controller clears this bit after all necessary logic is reset in the controller, which may take several clocks depending on the current state of the controller. Once this bit is cleared, the software must wait at least 3 PHY clocks before accessing the PHY domain (synchronization delay). Typically, software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain must be reset for proper operation.</p> <p>Note: Programming this field with random data causes side effect . Bit Bash register testing is not recommended.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Volatile: true</p>
29	reserved	R	<p>Reserved1</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
28:24	HIRDTHRES	R/W	<p>HIRD Threshold (HIRD_Thres)</p> <p>The controller asserts output signals utmi_l1_suspend_n and utmi_sleep_n (see "LPM Interface Signals" table in the Databook) on the basis of this signal:</p> <p>The controller asserts utmi_l1_suspend_n to put the PHY into Deep Low-Power mode in L1 when both of the following are true:</p> <ul style="list-style-type: none"> • HIRD value is greater than or equal to the value in DCTL.HIRD_Thres[3:0] • HIRD_Thres[4] is set to 1'b1. <p>The controller asserts utmi_sleep_n on L1 when one of the following is true:</p> <ul style="list-style-type: none"> • If the HIRD value is less than HIRD_Thres[3:0] or • HIRD_Thres[4] is set to 1'b0. <p>Note: This field must be set to '0' during SuperSpeed mode of operation.</p> <p>Value After Reset: 0x0</p>
23:20	LPM_NYET_thres	R/W	<p>LPM NYET Threshold</p> <p>When LPM Errata is enabled:</p> <p>Bits [23:20]: LPM NYET Response Threshold (LPM_NYET_thres)</p> <p>Handshake response to LPM token specified by device application. Response depends on DCFG.LPMCap.</p> <ul style="list-style-type: none"> • DCFG.LPMCap is 1'b0 - The controller always responds with Timeout (that is, no response). • DCFG.LPMCap is 1'b1 - The controller responds with an ACK on successful LPM transaction, which requires that all of the following are satisfied:

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> There are no PID or CRC5 errors in both the EXT token and the LPM token (if not true, inactivity results in a timeout ERROR). No data is pending in the TxFIFO and RxFIFO is empty (else NYET). The BESL value in the LPM token is less than or equal to LPM_NYET_thres[3:0] <p>Value After Reset: 0xf</p>
19	KeepConnect	R/W	<p>Keep Connect</p> <p>When '1', this bit enables the save and restore programming model by preventing the controller from disconnecting from the host when DCTL.RunStop is set to '0'. It also enables the Hibernation Request Event to be generated when the link goes to U3 or L2.</p> <p>The device controller disconnects from the host when DCTL.RunStop is set to '0'. This bit indicates whether to preserve this behavior ('0'), or if the controller must not disconnect when RunStop is set to 0 ('1').</p> <p>This bit also prevents the LTSSM from automatically going to U0/L0 when the host requests resume from U3/L2.</p> <p>Note: If Hibernation is disabled, that is, GCTL[1].GblHibernationEn = 0, this bit is tied to zero.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
18	L1HibernationEn	R/W	<p>L1HibernationEn</p> <p>When this bit is set along with KeepConnect, the device controller generates a Hibernation Request Event if L1 is enabled and the HIRD value in the LPM token is larger than the threshold programmed in DCTL.HIRD_Thres.</p> <p>The controller does not exit the LPM L1 state until software writes Recovery into the DCTL.ULStChngReq field.</p> <p>This prevents corner cases where the device is entering hibernation at the same time the host is attempting to exit L1.</p> <p>Note: If Hibernation is disabled, that is, GCTL[1].GblHibernationEn = 0, this bit is tied to zero.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
17	CRS	R/W	<p>Controller Restore State (CRS)</p> <p>This command is similar to the USBCMD.CRS bit in host mode and initiates the restore process. When software sets this bit to '1', the controller immediately sets DSTS.RSS to '1'. When the controller has finished the restore process, it sets DSTS.RSS to '0'.</p> <p>Note: When read, this field always returns '0'.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>

Bits	Name	Memory Access	Description
16	CSS	R/W	<p>Controller Save State (CSS)</p> <p>This command is similar to the USBCMD.CSS bit in host mode and initiates the save process. When software sets this bit to '1', the controller immediately sets DSTS.SSS to '1'. When the controller has finished the save process, it sets DSTS.SSS to '0'.</p> <p>Note: When read, this field always returns '0'.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
15:13	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x7</p> <p>WriteConstraint: writeAsRead</p>
12	INITU2ENA	R/W	<p>Initiate U2 Enable</p> <ul style="list-style-type: none"> • 1'b0: May not initiate U2 (default) • 1'b1: May initiate U2 <p>On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U2_ENABLE), and clears this bit when ClearFeature(U2_ENABLE) is received.</p> <p>If DCTL[11] (AcceptU2Ena) is 0, the link immediately exits U2 state.</p> <p>Value After Reset: 0x0</p>
11	ACCEPTU2ENA	R/W	<p>Accept U2 Enable</p> <ul style="list-style-type: none"> • 1'b0: Reject U2 except when Force_LinkPM_Accept bit is set (default) • 1'b1: Controller accepts transition to U2 state if nothing is pending on the application side. <p>On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command.</p> <p>Value After Reset: 0x0</p>
10	INITU1ENA	R/W	<p>Initiate U1 Enable</p> <ul style="list-style-type: none"> • 1'b0: May not initiate U1 (default); • 1'b1: May initiate U1. <p>On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U1_ENABLE), and clears this bit when ClearFeature(U1_ENABLE) is received.</p> <p>If DCTL[9] (AcceptU1Ena) is 0, the link immediately exits U1 state.</p> <p>Value After Reset: 0x0</p>
9	ACCEPTU1ENA	R/W	Accept U1 Enable

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> • 1'b0: Controller rejects U1 except when Force_LinkPM_Accept bit is set (default) • 1'b1: Controller accepts transition to U1 state if nothing is pending on the application side. <p>On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command.</p> <p>Value After Reset: 0x0</p>
8:5	ULSTCHNGREQ	W	<p>ULSTCHNGREQ</p> <p>Software writes this field to issue a USB/Link state change request. A change in this field indicates a new request to the controller.</p> <p>If software wants to issue the same request back-to-back, it must write a 0 to this field between the two requests. The result of the state change request is reflected in the USB/Link State in DSTS. These bits are self-cleared on the MAC Layer exiting suspended state.</p> <p>If software is updating other fields of the DCTL register and not intending to force any link state change, then it must write a 0 to this field.</p> <p>SS Compliance mode is normally entered and controlled by the remote link partner. Refer to the USB 3.0 specification.</p> <p>Alternatively, you can force the local link directly into compliance mode, by resetting the SS link with the RUN/STOP bit set to zero.</p> <p>If you then write '10' to the USB/Link State Change field and '1' to RUN/STOP, the link goes to compliance mode.</p> <p>Once you are in compliance, you may alternately write zero and '10' to this field to advance the compliance pattern.</p> <p>In SS mode:</p> <ul style="list-style-type: none"> • Value Requested Link State Transition/Action • 0 No Action • 4 SS.Disabled • 5 Rx.Detect • 6 SS.Inactive • 8 U3 exit request • 10 Compliance • Others: Reserved <p>In HS/FS/LS mode:</p> <ul style="list-style-type: none"> • ValueRequested USB state transition • 8 Remote wakeup request • Others: Reserved <p>The Remote wakeup request must be issued 2us after the device goes into suspend state (DSTS[21:18] is 3 - refer to Table "Fields for Register: DSTS").</p> <p>Note: After coming out of hibernation, software must write 8 (Recovery) into this</p>

Bits	Name	Memory Access	Description
			<p>field to confirm exit from the suspended state.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
4:1	TSTCTL	R/W	<p>Test Control</p> <ul style="list-style-type: none"> • 4'b000: Test mode disabled • 4'b001: Test_J mode • 4'b010: Test_K mode • 4'b011: Test_SE0_NAK mode • 4'b100: Test_Packet mode • 4'b101: Test_Force_Enable • Others: Reserved <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p>
0	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>

USB_DEVEN

- **Description:** Device Event Enable Register
- This register controls the generation of device-specific events. If an enable bit is set to 0, the event will not be generated.
- **Size:** 32 bits
- **Offset:** 0xc708

Bits	Name	Memory Access	Description
31:17	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x7fff</p> <p>WriteConstraint: writeAsRead</p>
16	ECCERREN	R	<p>ECC Error Enable.</p> <p>If this bit is set to 1, the controller reports an ECC error to the software when an uncorrectable ECC occurs internally.</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
			<p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
15	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
14	L1WKUPEVTEN	R/W	<p>L1 Resume Detected Event Enable.</p> <p>Note: If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, then this bit is for L1 Resume Detected Event Enable.</p> <p>Value After Reset: 0x0</p>
13	reserved	R	<p>Reserved Field: Yes</p> <p>Value After Reset: 0x0</p>
12	VENDEVTSTRCVDEN	R/W	<p>Vendor Device Test LMP Received Event (VndrDevTstRcvedEn)</p> <p>Value After Reset: 0x0</p>
11	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
10	reserved_	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
9	ERRTICERREVREN	R/W	<p>Erratic Error Event Enable</p> <p>Value After Reset: 0x0</p>
8	L1SUSPEN	R/W	<p>L1 Suspend Event Enable</p> <p>Note: Only if GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, this bit is for L1 Suspend Event Enable.</p> <p>Value After Reset: 0x0</p>
7	SOFTEVTEN	R/W	<p>Start of (u)frame</p> <p>Value After Reset: 0x0</p>
6	U3L2L1SuspEn	R/W	<p>U3/L2 or U3/L2L1 Suspend Event Enable.</p> <p>Note:</p>

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> If GUCTL1[DEV_DECUPLE_L1L2_EVT] is enabled, then this bit is for U3/L2 Suspend Event Enable. If GUCTL1[DEV_DECUPLE_L1L2_EVT] is not enabled, then this bit is for U3/L2L1 Suspend Event Enable. <p>Value After Reset: 0x0</p>
5	HibernationReqEvtEn	R/W	<p>This bit enables/disables the generation of the Hibernation Request Event.</p> <p>Value After Reset: 0x0</p>
4	WKUPEVTEN	R/W	<p>U3/L2 or U3/L2L1 Resume Detected Event Enable.</p> <p>Note:</p> <ul style="list-style-type: none"> If GUCTL1[DEV_DECUPLE_L1L2_EVT] is enabled, then this bit is for U3/L2 Resume Detected Event Enable. If GUCTL1[DEV_DECUPLE_L1L2_EVT] is not enabled, then this bit is for U3/L2L1 Resume Detected Event Enable. <p>Value After Reset: 0x0</p>
3	ULSTCNGEN	R/W	<p>USB/Link State Change Event Enable</p> <p>Value After Reset: 0x0</p>
2	CONNECTDONEEVTCN	R/W	<p>Connection Done Enable</p> <p>Value After Reset: 0x0</p>
1	USBRSTEVTCN	R/W	<p>USB Reset Enable</p> <p>Value After Reset: 0x0</p>
0	DISSCONNNEVTCN	R/W	<p>Disconnect Detected Event Enable</p> <p>Value After Reset: 0x0</p>

USB_DSTS

- Description:** Device Status Register. This register indicates the status of the device controller with respect to USB-related events.
- Note:** When Hibernation is not enabled, RSS and SSS fields always return 0 when read.
- Size:** 32 bits
- Offset:** 0xc70c

Bits	Name	Memory Access	Description
31:30	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x3</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>

Bits	Name	Memory Access	Description
29	DCNRD	R	<p>Device Controller Not Ready</p> <p>The bit indicates that the controller is in the process of completing the state transitions after exiting from hibernation.</p> <p>To complete the state transitions, it takes 256 bus clock cycles from the time DCTL[31].Run/Stop is set. During hibernation, if the UTMII/ULPI PHY is in suspended state, then the 256-bus clock cycle delay starts after the PHY exited suspended state. Software must set DCTL[31].Run/Stop to '1' and wait for this bit to be deasserted to zero before processing DSTS.USBLnkSt.</p> <p>This bit is valid only when DWC_USB3_EN_PWRROPT is set to 2 and GCTL[1].GblHibernationEn =1.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
28	SRE	R/W1C	<p>Save Restore Error. Currently not supported.</p> <p>Value After Reset: 0x0</p> <p>Reset Mask: 0x1</p> <p>Volatile: true</p>
27:26	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x3</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
25	RSS	R	<p>RSS Restore State Status</p> <p>This bit is similar to the USBSTS.RSS in host mode.</p> <p>When the controller finishes the restore process, it completes the command by setting DSTS.RSS to '0'.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
24	SSS	R	<p>SSS Save State Status</p> <p>This bit is similar to the USBSTS.SSS in host mode.</p> <p>When the controller has finished the save process, it completes the command by setting DSTS.SSS to '0'.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p>

Bits	Name	Memory Access	Description
			<p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
23	COREIDLE	R	<p>Core Idle</p> <p>The bit indicates that the controller finished transferring all RxFIFO data to system memory, writing out all completed descriptors, and all Event Counts are zero.</p> <p>Note: While testing for Reset values, mask out the read value. This bit represents the changing state of the controller and does not hold a static value.</p> <p>Value After Reset: 0x1</p> <p>Testable: untestable</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
22	DEVCTRLHLT	R	<p>Device Controller Halted</p> <p>This bit is set to 0 when the Run/Stop bit in the DCTL register is set to 1.</p> <p>The controller sets this bit to 1 when, after SW sets Run/Stop to 0, the controller is idle and the lower layer finishes the disconnect process.</p> <p>When Halted=1, the controller does not generate Device events.</p> <p>Note:</p> <ul style="list-style-type: none"> • The controller does not set this bit to 1 if GEVNTCOUNTn has some valid value. Software needs to acknowledge the events that are generated (by writing to GEVNTCOUNTn) while it is waiting for this bit to be set to 1. • When Interrupt Moderation is enabled, there could be delay in raising the interrupt line when the event count is non-zero. Software should read the GEVNTCOUNT register directly and acknowledge them. <p>Value After Reset: 0x1</p> <p>Testable: writeAsRead</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
21:18	USBLNKST	R	<p>USBLNKST.</p> <p>USB/Link State</p> <p><i>In SS mode:</i> LTSSM State</p> <ul style="list-style-type: none"> • 4'h0: U0 • 4'h1: U1 • 4'h2: U2 • 4'h3: U3 • 4'h4: SS_DIS • 4'h5: RX_DET • 4'h6: SS_INACT

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> • 4'h7: POLL • 4'h8: RECOV • 4'h9: HRESET • 4'ha: CMPLY • 4'hb: LPBK • 4'hf: Resume/Reset <p><i>In HS/FS/LS mode:</i></p> <ul style="list-style-type: none"> • 4'h0: On state • 4'h2: Sleep (L1) state • 4'h3: Suspend (L2) state • 4'h4: Disconnected state (Default state) • 4'h5: Early Suspend state (valid only when Hibernation is disabled, GCTL[1].GblHibernationEn = 0) • 4'he: Reset (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) • 4'hf: Resume (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) <p>The link state Resume/Reset indicates that the controller received a resume or USB reset request from the host while the link was in hibernation. Software must write '8' (Recovery) to the DCTL.ULStChngReq field to acknowledge the resume/reset request.</p> <p>When Hibernation is enabled, GCTL[1].GblHibernationEn = 1, this field USBLnkSt is valid only when DCTL[31].Run/Stop set to '1' and DSTS[29].DCNRD = 0.</p> <p>The Early Suspend link state is an early indication of device suspend in HS/FS. The link state changes to Early Suspend after detecting bus idle for 3ms.</p> <ul style="list-style-type: none"> • In HS operation, this is an indication that the USB bus (that is, LineState) has been in idle (SE0) for 3ms. However, it does not confirm whether the next process is Suspend or Reset. The device checks the bus again after pull up enable delay and if the line state indicates Suspend (full speed J), then the device waits for an additional time (~3ms) to indicate the actual Suspend state. • In FS operation, this is an indication that the USB bus (that is, LineState) has been in idle (J) for 3ms. The device waits for an additional time (~3ms of Idle) to indicate the actual Suspend state. <p>Value After Reset: 0x4</p> <p>Testable: writeAsRead</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
17	RXFIFOEMPTY	R	<p>RxFIFO Empty.</p> <p>Value After Reset: 0x1</p> <p>Testable: writeAsRead</p>

Bits	Name	Memory Access	Description
			<p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
16:3	SOFFN	R	<p>Frame/Microframe Number of the Received SOF.</p> <p>When the controller is operating at SuperSpeed,</p> <ul style="list-style-type: none"> • [16:3] indicates the uframe/ITP number <p>When the controller is operating at high-speed,</p> <ul style="list-style-type: none"> • [16:6] indicates the frame number • [5:3] indicates the microframe number <p>When the controller is operating at full-speed,</p> <ul style="list-style-type: none"> • [16:14] is not used. Software can ignore these 3 bits • [13:3] indicates the frame number <p>Note: After power-on reset, the controller generates the microframe number internally for every 125us if the USB host has not issued SOF/ITP yet. During P3 state, the duration of SOFFN is based on the suspend_clk frequency.</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>
2:0	CONNECTSPD	R	<p>Connected Speed (ConnectSpd)</p> <p>Indicates the speed at which the DWC_usb3 controller has come up after speed detection through a chirp sequence.</p> <ul style="list-style-type: none"> • 3'b100: SuperSpeed (PHY clock is running at 125 or 250 MHz) • 3'b000: High-speed (PHY clock is running at 30 or 60 MHz) • 3'b001: Full-speed (PHY clock is running at 30 or 60 MHz) <p>Low-speed is not supported for devices using a UTMI+ PHY.</p> <p>Values:</p> <ul style="list-style-type: none"> • 0x4 (SuperSpeed): SuperSpeed (PHY clock is running at 125 or 250 MHz) • 0x0 (HighSpeed): High-speed (PHY clock is running at 30 or 60 MHz) • 0x1 (FullSpeed): Full-speed (PHY clock is running at 30 or 60 MHz) <p>Value After Reset: 0x4</p> <p>Testable: writeAsRead</p> <p>Volatile: true</p> <p>WriteConstraint: writeAsRead</p>

USB_DGCMMPAR

- **Description:** Device Generic Command Parameter Register. This register indicates the device command parameter. This must be programmed before or along with the device command. The

available device commands are listed in USB_DGCMD register.

- **Size:** 32 bits
- **Offset:** 0xc710

Bits	Name	Memory Access	Description
31:16	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0xffff</p> <p>WriteConstraint: writeAsRead</p>
15:12	CMDSTATUS	R	<p>Command Status</p> <ul style="list-style-type: none"> • 1: CmdErr: Indicates that the device controller encountered an error while processing the command. • 0: Indicates command success <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>WriteConstraint: writeAsRead</p>
11	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
10	CMDACT	R/W1S	<p>Command Active</p> <p>The software sets this bit to 1 to enable the device controller to execute the generic command.</p> <p>The device controller sets this bit to 0 after executing the command.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
9	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: writeAsRead</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
8	CMDIOC	R/W	<p>Command Interrupt on Complete</p> <p>When this bit is set, the device controller issues a Generic Command Completion event after executing the command.</p> <p>Note that this interrupt is mapped to DCFG.IntrNum.</p> <p>Note: This field must not set to '1' if the DCTL.RunStop field is '0'.</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
7:0	CMDTYP	R/W	<p>CMDTYP Generic Command Type Specifies the type of generic command the software driver is requesting the controller to perform.</p> <ul style="list-style-type: none"> • 02h: Set Periodic Parameters • 04h: Set Scratchpad Buffer Array Address Lo • 05h: Set Scratchpad Buffer Array Address Hi • 07h: Transmit Device Notification • 09h: Selected FIFO Flush • 0Ah: All FIFO Flush • 0Ch: Set Endpoint NRDY • 10h: Run SoC Bus LoopBack Test • 11h: Restart After Disconnect <p>All other values are reserved.</p> <p>Value After Reset: 0x0</p>

USB_DALEPENA

- **Description:** Device Active USB Endpoint Enable Register. This register indicates whether a USB endpoint is active in a given configuration or interface.
- **Size:** 32 bits
- **Offset:** 0xc720

Bits	Name	Memory Access	Description
31:0	USBACTEP	R/W	<p>USBACTEP USB Active Endpoints (USBActEP)</p> <p>This field indicates if a USB endpoint is active in the current configuration and interface. It applies to USB IN endpoints 0.15 and OUT endpoints 0.15, with one bit for each of the 32 possible endpoints. Even numbers are for USB OUT endpoints, and odd numbers are for USB IN endpoints, as follows:</p> <ul style="list-style-type: none"> • Bit[0]: USB EP0-OUT • Bit[1]: USB EP0-IN • Bit[2]: USB EP1-OUT • Bit[3]: USB EP1-IN <p>The entity programming this register must set bits 0 and 1 because they enable control endpoints that map to physical endpoints (resources) after USBReset.</p> <p>Hardware clears these bits for all endpoints (other than EP0-OUT and EP0-IN) after detecting a USB reset event. After receiving SetConfiguration and SetInterface requests, the application must program endpoint registers accordingly and set these bits.</p> <p>Value After Reset: 0x0</p>

USB_DEPCMAPAR2[0:15]

- **Description:** Device Physical Endpoint-n Command Parameter 2 Register (DEPCMDPAR2n)
- This register indicates the physical endpoint command Parameter 2. It must be programmed before issuing the command.
- **Size:** 32 bits
- **Offset:** 0xc800 + (*i* * 0x10), where *i* = 0..15

Bits	Name	Memory Access	Description
31:0	PARAMETER	R/W	<p>PARAMETER</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0xffffffff</p> <p>WriteConstraint: writeAsRead</p>

USB_DEPCMDPAR1[0:15]

- **Description:** Device Physical Endpoint-n Command Parameter 1 Register (DEPCMDPAR1n)
- **Size:** 32 bits
- **Offset:** 0xc804 + (*i* * 0x10), where *i* = 0..15

Bits	Name	Memory Access	Description
31:0	PARAMETER	R/W	<p>PARAMETER</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0xffffffff</p> <p>WriteConstraint: writeAsRead</p>

USB_DEPCMDPAR0[0:15]

- **Description:** Device Physical Endpoint-n Command Parameter 0 Register (DEPCMDPAR0n)
- **Size:** 32 bits
- **Offset:** 0xc808 + (*i* * 0x10), where *i* = 0..15

Bits	Name	Memory Access	Description
31:0	PARAMETER	R/W	<p>PARAMETER</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0xffffffff</p> <p>WriteConstraint: writeAsRead</p>

USB_DEPCMD[0:15]

- **Description:** Device Physical Endpoint-n Command Register
- This register enables software to issue physical endpoint-specific commands. This register contains command, control, and status fields relevant to the current generic command, while the USB_DEPCMDPAR[2:0]n registers provide command parameters and return status information. Several fields (including Command Type) are write-only, so their read values are undefined. After power-on, prior to issuing the first endpoint command, the read value of this register is undefined.

In particular, the CmdAct bit may be set after power-on. In this case, it is safe to issue an endpoint command.

- **Size:** 32 bits
- **Offset:** 0xc80c + (*i* * 0x10), where *i* = 0..15

Bits	Name	Memory Access	Description
31:16	COMMANDPARAM	R/W	<p>Command Parameters or Event Parameters</p> <p>Command Parameters (CommandParam), when this register is written:</p> <p>For Start Transfer command:</p> <ul style="list-style-type: none"> ● [31:16]: StreamID. The USB StreamID assigned to this transfer <p>For Start Transfer command applied to an isochronous endpoint</p> <ul style="list-style-type: none"> ● [31:16]: StartMicroFramNum: Indicates the (micro)frame number to which the first TRB applies. <p>For Update Transfer, End Transfer, and Start New Configuration commands</p> <ul style="list-style-type: none"> ● [22:16]: Transfer Resource Index (XferRscldx). The hardware-assigned transfer resource index for the transfer, which was returned in response to the Start Transfer command. The application software-assigned transfer resource index for a Start New Configuration command. <p>Event Parameters (EventParam), when this register is read. Refer to bits [31:16] in Table "Device Endpoint-n Events: DEPEVT".</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0xffff</p> <p>WriteConstraint: writeAsRead</p>
15:12	CMDSTATUS	R/W	<p>Command Completion Status (CmdStatus)</p> <p>Additional information about the completion of this command is available in this field. The information is in the same format as bits 15:12 of the Endpoint Command Complete event, see "Device Endpoint-n Events: DEPEVT" table in the Programming Guide.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0xf</p> <p>WriteConstraint: writeAsRead</p>
11	HIPRI_FORCERM	R/W	<p>HighPriority/ForceRM (HiPri_ForceRM)</p> <ul style="list-style-type: none"> ● HighPriority: Only valid for Start Transfer command ● ForceRM: Only valid for End Transfer command ● ClearPendIN: Only valid for Clear Stall command . Software sets this bit to clear any pending IN transaction (on that endpoint) stuck at the lower layers when a Clear Stall command is issued. <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0x1</p>

Bits	Name	Memory Access	Description
			<p>WriteConstraint: writeAsRead</p>
10	CMDACT	R/W	<p>Command Active (CmdAct)</p> <p>Software sets this bit to 1 to enable the device endpoint controller to execute the generic command.</p> <p>The device controller sets this bit to 0 when the CmdStatus field is valid and the endpoint is ready to accept another command. This does not imply that all the effects of the previously-issued command have taken place.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
9	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
8	CMDIOC	R/W	<p>CMDIOC</p> <p>Command Interrupt on Complete (CmdIOC)</p> <p>When this bit is set, the device controller issues a generic Endpoint Command Complete event after executing the command.</p> <p>Note that this interrupt is mapped to DEPCFG.IntrNum.</p> <p>When the DEPCFG command is executed, the command interrupt on completion goes to the interrupt pointed by the DEPCFG.IntrNum in the current command.</p> <p>Note: This field must not set to 1 if the DCTL.RunStop field is 0.</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0x1</p> <p>WriteConstraint: writeAsRead</p>
7:4	reserved	R	<p>Reserved</p> <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0xf</p> <p>WriteConstraint: writeAsRead</p>
3:0	CMDTYP	R/W	<p>Command Type</p> <p>Specifies the type of command the software driver is requesting the controller to perform.</p> <ul style="list-style-type: none"> • 00h: Reserved

Bits	Name	Memory Access	Description
			<ul style="list-style-type: none"> • 01h: Set Endpoint Configuration - -64 or 96-bit Parameter • 02h: Set Endpoint Transfer Resource Configuration - 32-bit Parameter • 03h: Get Endpoint State - No Parameter Needed • 04h: Set Stall - No Parameter Needed • 05h: Clear Stall (see Set Stall) - No Parameter Needed • 06h: Start Transfer - 64-bit Parameter • 07h: Update Transfer - No Parameter Needed • 08h: End Transfer - No Parameter Needed • 09h: Start New Configuration - No Parameter Needed <p>Value After Reset: 0x0</p> <p>Testable: untestable</p> <p>Reset Mask: 0xf</p> <p>WriteConstraint: writeAsRead</p>

USB_DEV_IMOD

- **Description:** Device Interrupt Moderation Register (DEV_IMOD). This register controls the Interrupt Moderation feature that allows the device software to throttle the interrupt rate.
- Key Functions:
- Interrupt Moderation is enabled only when the IMOD Interval is programmed to a non-zero value.
- Interrupt is asserted whenever the IMOD (down) counter is 0, EVNT_HANDLER_BUSY is 0, and there are pending events (that is, event count is non-zero)
- USB_GEVNTCOUNT[EVNT_HANDLER_BUSY] is set by hardware when interrupt is asserted, and cleared by software when interrupt processing is completed.
- The Interrupt line is de-asserted after the first write to the event count.
- IMOD counter is loaded with IMOD interval whenever the Interrupt line is de-asserted.
- **Size:** 32 bits
- **Offset:** 0xca00

Bits	Name	Memory Access	Description
31:16	DEVICE_IMODC	R/W	<p>Interrupt Moderation Down Counter</p> <p>Loaded with the DEVICE_IMODI value, whenever the hardware interrupt(n) line is de-asserted from the asserted state, counts down to 0, and stops.</p> <p>The interrupt(n) is signaled whenever this counter is 0, EVNT_HANDLER_BUSY is 0, and there are pending events (that is, event count is non-zero).</p> <p>This counter may be directly written by software at any time to alter the interrupt rate.</p> <p>Value After Reset: 0x0</p> <p>Reset Mask: 0xffff</p> <p>Volatile: true</p>
15:0	DEVICE_IMODI	R/W	Moderation Interval (DEVICE_IMODI)

Bits	Name	Memory Access	Description
			<p>This field holds the minimum inter-interrupt interval between events. The interval is specified in terms of 250ns increments.</p> <p>A value of 0 disables the interrupt throttling logic and interrupts are generated immediately if event count becomes non-zero.</p> <p>In scaledown simulation mode, 4 ram clocks are used to time 250ns.</p> <p>Value After Reset: 0x0</p> <p>Reset Mask: 0xffff</p> <p>Volatile: true</p>

11 system security

Please get in touch with ESWIN to request more detailed technical information.

ibu_techsupport@eswincomputing.com

12 Peripheral Devices

EIC7700X includes the following interfaces:

- 12×I2C, supports speeds up to 1000 kbit/s
- 5×UART, uart1 supports automatic flow control
- 2×SPI, supports up to 50 MHz and only supports master mode
- 1×PWM, three programmable timers are internally instantiated, supporting 0%~100% duty cycle
- 4×TIMER, each TIMER internally instantiates 8 programmable timers, supporting 0%~100% duty cycle
- 112×GPIO, the 0~31 GPIO supports interrupt mode
- 3×JTAG, supports up to 20 MHz
- 3×I2S, supports up to 192 kbps
- 1 fan control interface (FAN_CTRL) to receive fan speed information

I2C/UART/SPI/PWM/FAN_CTRL all have soft reset and gating. Before enable the peripheral, you need to release the reset and disable clock gating. For some peripheral, you need to configure the corresponding IO pad. The reset and clock of the GPIO module are the same as those of the CLMM.

The base address of the low-speed peripheral is as follows:

Table 12-1 Peripheral address space.

Name	Address space	Size	Remarks
wdt0	0x5080_0000~0x5080_3FFF	16k	LSP APB2
wdt1	0x5080_4000~0x5080_7FFF	16k	LSP APB2
wdt2	0x5080_8000~0x5080_BFFF	16k	LSP APB2
wdt3	0x5080_C000~0x5080_FFFF	16k	LSP APB2
spi0	0x5081_0000~0x5081_3FFF	16k	LSP APB2
spi1	0x5081_4000~0x5081_7FFF	16k	LSP APB2
pwm	0x5081_8000~0x5081_BFFF	16k	LSP APB2
uart0	0x5090_0000~0x5090_FFFF	64k	LSP APB3
uart1	0x5091_0000~0x5091_FFFF	64k	LSP APB3
uart2	0x5092_0000~0x5092_FFFF	64k	LSP APB3
uart3	0x5093_0000~0x5093_FFFF	64k	LSP APB3
uart4	0x5094_0000~0x5094_FFFF	64k	LSP APB3
i2c0	0x5095_0000~0x5095_FFFF	64k	LSP APB3
i2c1	0x5096_0000~0x5096_FFFF	64k	LSP APB3

Name	Address space	Size	Remarks
i2c2	0x5097_0000~0x5097_FFFF	64k	LSP APB3
i2c3	0x5098_0000~0x5098_FFFF	64k	LSP APB3
i2c4	0x5099_0000~0x5099_FFFF	64k	LSP APB3
i2c5	0x509A_0000~0x509A_FFFF	64k	LSP APB3
i2c6	0x509B_0000~0x509B_FFFF	64k	LSP APB3
i2c7	0x509C_0000~0x509C_FFFF	64k	LSP APB3
i2c8	0x509D_0000~0x509D_FFFF	64k	LSP APB3
i2c9	0x509E_0000~0x509E_FFFF	64k	LSP APB3
GPIO	0x5160_0000~0x5160_007F	128B	
CLMM control	0x5160_0000~0x517F_FFFF	2M-128B	
i2c10	0x5183_0000~0x5183_7FFF	32k	
i2c11	0x5183_8000~0x5183_FFFF	32k	
TIMER0	0x5184_0000~0x5184_7FFF	32k	
TIMER1	0x5184_8000~0x5184_FFFF	32k	
TIMER2	0x5185_0000~0x5185_7FFF	32k	
TIMER3	0x5185_8000~0x5185_FFFF	32k	
I2S0	0x5020_0000~0x5020_FFFF	64k	
I2S1	0x5021_0000~0x5021_FFFF	64k	
I2S2	0x5022_0000~0x5022_FFFF	64k	

12.1 I/O Control (CLMM)

12.1.1 Overview

CLMM (Chip-Level Mode Mux) provides a pad reuse mechanism for many functional modules that require IO ports for internal and external data/control interaction.

Supports the following functions:

- Supports apb2.0. The data bit width is 32bit
- Support I/O input enable control
- Support I/O driving strength setting
- Control I/O multiplexing through function sel
- Support Schmidt trigger
- Support RGMII pad, support 1.8V, 3.3V voltage
- Support pull-up/pull-down setting
- Clock input through OSC pad
- Only need to set the function selection, pull up and down, etc. It is not recommended to set the

driving strength

12.1.2 Register Description

12.1.2.1 Register Specification

The CLMM provides three types of I/O pads.

General io register control is as follows:

No.	Name	remarks
1	DS3-DS0	Drive strength, {DS3,DS2,DS1,DS0}=pwdata[6:3]
2	IE	Input enable, pwdata[0]
3	PU	PU:pull up enable, PD: pull down enable; pwdata[2:1] = 2'b01, PU=1, pull up
4	PD	pwdata[2:1] = 2'b10, PD=1, pull down
5	ST	Schmidt trigger, pwdata[7]
6	func_sel	function selection, pwdata[18:16]

RGMII io register is controlled as follows:

No.	Name	remarks
1	ST	pwdata[7], Schmidt trigger
2	IE	pwdata[0], input enable
3	PU	PU:pull up enable, PD: pull down enable; pwdata[2:1] = 2'b01, PU=1, pull up
4	PD	pwdata[2:1] = 2'b10, PD=1, pull down
5	DS3-DS0	Drive strength, {DS3,DS2,DS1,DS0}=pwdata[6:3]
6	func_sel	function selection, pwdata[18:16]

RGMII IO Voltage Control:

No.	Name	remarks
1	MS1	Voltage control signal, {MS2,MS1} = pwdata[1:0], 2'b11: 1.8 V 2'b00: 3.3 V
2	MS2	It is forbidden to set other values

OSC IO Register:

No.	Name	remarks
1	DS3-DS0	Drive strength, {DS3,DS2,DS1,DS0}=pwdata[7:4]
2	RD0	Damping resistor setting, {RD1,RD0}=pwdata[3:2]
3	RD1	
4	REF0	Feedback resistor settings, {REF1,REF0}=pwdata[1:0]

No.	Name	remarks				
5	REF1					

12.1.2.2 Register Detail Description

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
CHIP_MODE	0x80					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	R	7	7	1
		DS	R	6	3	0
		PD	R	2	2	1
		PU	R	1	1	0
		IE	R	0	0	1
MODE_SET0	0x84					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	1
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
MODE_SET1	0x88					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	1
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
MODE_SET2	0x8c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	1
		DS	RW	6	3	1

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
MODE_SET3	0x90					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	1
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
XIN	0x94					
		reserved2	R	31	16	0
		reserved1	R	15	8	0
		DS	RW	7	4	12
		FRS	RW	3	2	2
		RD	RW	1	0	0
RESERVED	0x98					
		reserved2	R	31	16	0
		reserved1	R	15	8	0
		DS	RW	7	4	12
		FRS	RW	3	2	2
		RD	RW	1	0	0
RST_OUT_N	0x9c					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	R	0	0	0
KEY_RESET_N	0xa0					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		SMT	R	7	7	1
		DS	R	6	3	0
		PD	R	2	2	0
		PU	R	1	1	1
		IE	R	0	0	1
RESERVED	0xa4					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	R	7	7	1
		DS	R	6	3	0
		PD	R	2	2	0
		PU	R	1	1	1
		IE	R	0	0	1
RESERVED	0xa8					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	R	7	7	1
		DS	R	6	3	0
		PD	R	2	2	0
		PU	R	1	1	1
		IE	R	0	0	1
RESERVED	0xac					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	R	7	7	0
		DS	RW	6	3	1
		PD	R	2	2	0
		PU	R	1	1	1
		IE	R	0	0	0
GPIO0	0xb0					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
POR_SEL	0xb4					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	R	7	7	0
		DS	R	6	3	0
		PD	R	2	2	0
		PU	R	1	1	0
		IE	R	0	0	1
JTAG0_TCK	0xb8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	1
		PU	RW	1	1	0
		IE	RW	0	0	1
JTAG0_TMS	0xbc					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	1
JTAG0_TDI	0xc0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	1
JTAG0_TDO	0xc4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
JTAG0_TRST	0xc8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
SPI2_CS0_N	0xcc					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
JTAG1_TCK	0xd0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	1

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		PU	RW	1	1	0
		IE	RW	0	0	1
JTAG1_TMS	0xd4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	1
JTAG1_TDI	0xd8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	1
JTAG1_TDO	0xdc					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
JTAG1_TRST	0xe0					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		IE	RW	0	0	0
SPI2_CS1_N	0xe4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
PCIE_CLKREQ_N	0xe8					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
PCIE_WAKE_N	0xec					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	1
PCIE_PERST_N	0xf0					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
HDMI_SCL	0xf4					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
HDMI_SDA	0xf8					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
HDMI_CEC	0xfc					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	0
JTAG2_TRST	0x100					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	1
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	1
RGMII0_CLK_125	0x104					

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII0_TXEN	0x108					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII0_TXCLK	0x10c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII0_TXD0	0x110					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII0_TXD1	0x114					
		reserved2	R	31	19	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII0_TXD2	0x118					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII0_TXD3	0x11c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2S0_BCLK	0x120					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2S0_WCLK	0x124					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2S0_SDI	0x128					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2S0_SDO	0x12c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2S_MCLK	0x130					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII0_RXCLK	0x134					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII0_RXDV	0x138					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII0_RXD0	0x13c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII0_RXD1	0x140					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII0_RXD2	0x144					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII0_RXD3	0x148					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2S2_BCLK	0x14c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2S2_WCLK	0x150					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2S2_SDI	0x154					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2S2_SDO	0x158					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
GPIO27	0x15c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
GPIO28	0x160					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
GPIO29	0x164					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII0_MDC	0x168					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII0_MDIO	0x16c					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII0_INTB	0x170					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII1_CLK_125	0x174					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		IE	RW	0	0	1
RGMII1_TXEN	0x178					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII1_TXCLK	0x17c					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII1_TXD0	0x180					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII1_TXD1	0x184					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
RGMII1_TXD2	0x188					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII1_TXD3	0x18c					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2S1_BCLK	0x190					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2S1_WCLK	0x194					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2S1_SDI	0x198					

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2S1_SDO	0x19c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
GPIO34	0x1a0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII1_RXCLK	0x1a4					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII1_RXDV	0x1a8					
		reserved2	R	31	19	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII1_RXD0	0x1ac					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII1_RXD1	0x1b0					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII1_RXD2	0x1b4					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII1_RXD3	0x1b8					
		reserved2	R	31	19	0
		func_sel	R	18	16	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
SPI1_CS0_N	0x1bc					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
SPI1_CLK	0x1c0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
SPI1_D0	0x1c4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
SPI1_D1	0x1c8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
SPI1_D2	0x1cc					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
SPI1_D3	0x1d0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
SPI1_CS1_N	0x1d4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII1_MDC	0x1d8					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RGMII1_MDIO	0x1dc					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RGMII1_INTB	0x1e0					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	2
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
USB0_PWREN	0x1e4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
USB1_PWREN	0x1e8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
I2C0_SCL	0x1ec					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C0_SDA	0x1f0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C1_SCL	0x1f4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C1_SDA	0x1f8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C2_SCL	0x1fc					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C2_SDA	0x200					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C3_SCL	0x204					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C3_SDA	0x208					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		IE	RW	0	0	1
I2C4_SCL	0x20c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C4_SDA	0x210					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C5_SCL	0x214					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C5_SDA	0x218					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
UART0_TX	0x21c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
UART0_RX	0x220					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
UART1_TX	0x224					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
UART1_RX	0x228					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
UART1_CTS	0x22c					

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
UART1_RTS	0x230					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
UART2_TX	0x234					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
UART2_RX	0x238					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
JTAG2_TCK	0x23c					
		reserved2	R	31	19	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	1
		PU	RW	1	1	0
		IE	RW	0	0	1
JTAG2_TMS	0x240					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	1
JTAG2_TDI	0x244					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	1
JTAG2_TDO	0x248					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
FAN_PWM	0x24c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
FAN_TACH	0x250					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
MIPI_CSI0_XVS	0x254					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI0_XHS	0x258					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI0_MCLK	0x25c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI1_XVS	0x260					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI1_XHS	0x264					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI1_MCLK	0x268					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI2_XVS	0x26c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI2_XHS	0x270					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI2_MCLK	0x274					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI3_XVS	0x278					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI3_XHS	0x27c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI3_MCLK	0x280					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI4_XVS	0x284					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI4_XHS	0x288					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI4_MCLK	0x28c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI5_XVS	0x290					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI5_XHS	0x294					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
MIPI_CSI5_MCLK	0x298					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
SPI3_CS_N	0x29c					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		IE	RW	0	0	1
SPI3_CLK	0x2a0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
SPI3_DI	0x2a4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
SPI3_DO	0x2a8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
GPIO92	0x2ac					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
GPIO93	0x2b0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
S_MODE	0x2b4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
GPIO95	0x2b8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
SPI0_CS_N	0x2bc					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	0
SPI0_CLK	0x2c0					

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
SPI0_D0	0x2c4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
SPI0_D1	0x2c8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
SPI0_D2	0x2cc					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
SPI0_D3	0x2d0					
		reserved2	R	31	19	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C10_SCL	0x2d4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C10_SDA	0x2d8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C11_SCL	0x2dc					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
I2C11_SDA	0x2e0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
GPIO106	0x2e4					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
BOOT_SEL0	0x2e8					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
BOOT_SEL1	0x2ec					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
BOOT_SEL2	0x2f0					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
BOOT_SEL3	0x2f4					
		reserved2	R	31	19	0
		func_sel	RW	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
GPIO111	0x2f8					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RESERVED	0x2fc					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
RESERVED	0x300					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	0
RESERVED	0x304					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	1
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	1
RESERVED	0x308					
		reserved2	R	31	19	0
		func_sel	R	18	16	0
		reserved1	R	15	8	0
		SMT	RW	7	7	0
		DS	RW	6	3	1
		PD	RW	2	2	0
		PU	RW	1	1	1
		IE	RW	0	0	0
LPDDR_REF_CLK	0x30c					
		reserved1	R	31	10	0
		MS2	RW	9	9	1
		MS1	RW	8	8	1
		SMT	RW	7	7	0
		DS	RW	6	3	0
		PD	RW	2	2	0
		PU	RW	1	1	0
		IE	RW	0	0	1
ADDR_RGMII0_SEL_MODE	0x310					
		reserved2	R	31	16	0
		reserved1	R	15	2	0
		RGM0_ms1	RW	1	1	0
		RGM0_ms2	RW	0	0	0

NAME	OFFSET	FILED	ACCESS	MSB	LSB	DEFAULT
ADDR_RGMII1_SEL_MODE	0x314					
		reserved2	R	31	16	0
		reserved1	R	15	2	0
		RGM1_ms1	RW	1	1	0
		RGM1_ms2	RW	0	0	0

12.2 I2C

12.2.1 Overview

I2C is a configurable, programmable control bus that supports the communication between devices. It is a simple two-wire bus protocols for system control for temperature sensors and voltage level translators, general-purpose I/O, A/D and D/A converters, codecs and many types of microprocessors.

The following features are supported:

- Support APB4.0, bus data bit width 32bit
- Master or slave I2C operation
- Three speeds:
 - Standard mode: 0~100kb/s
 - Fast mode: ≤400kb/s
 - fast mode plus: ≤1000kb/s
- 7-bit address
- Bulk transmit mode
- Interrupt or polled-mode operation
- Support DMA transmission
- Receive/send data fifo depth 32
- Handles Bit and Byte waiting at all bus speeds
- Programmable SDA hold time (tHD;DAT)
- SMBus/PMBus support
- Bus clear feature

12.2.2 Block Diagram

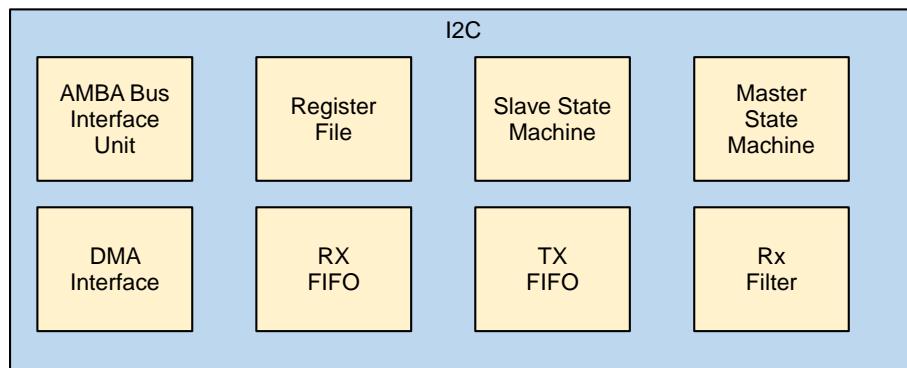


Figure 12-1 I2C block diagram

12.2.3 Function Description

12.2.3.1 Initialize

In addition to configuring the I2C itself, you also need to configure:

- Release the I2C apb bus reset, that is, the value of sw_i2c_rst_n register is 1
- Configure the CLMM func_sel to enable IE for pads
- Configure the APB clock

12.2.3.2 TXFIFO Data Format

IC_DATA_CMD	Restart	Stop	CMD	DATA
-------------	---------	------	-----	------

DATA –Read/Write field; data retrieved from slave is read from this field; data to be sent to slave is written to this field.

CMD –Write-only field; this bit determines whether transfer to be carried out is Read (CMD=1) or Write (CMD=0)

Figure 12-2 TX fifo Data format

12.2.3.3 Spike Suppression

This logic is based on counters that monitor the input signals (SCL and SDA), checking if they remain stable for a predetermined amount of ic_clk cycles before they are sampled internally. There is one separate counter for each signal (SCL and SDA). The number of ic_clk cycles can be programmed and should be calculated taking into account the frequency of apb clk and the relevant spike length specification.

Each counter is started whenever its input signal changes its value. Depending on the behavior of the input signal, one of the following scenarios occurs:

- The input signal remains unchanged until the counter reaches its count limit value. When this happens, the internal version of the signal is updated with the input value, and the counter is reset and stopped. The counter is not restarted until a new change on the input signal is detected.
- The input signal changes again before the counter reaches its count limit value. When this happens, the counter is reset and stopped, but the internal version of the signal is not updated. The counter remains stopped until a new change on the input signal is detected.

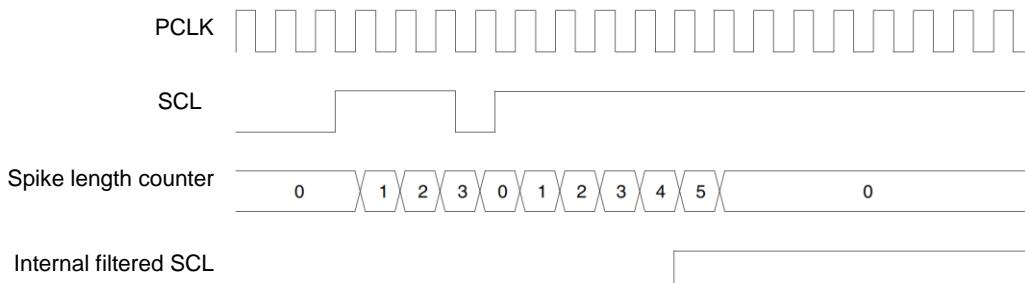


Figure 12-3 Spike Suppression Example

The spike suppression length in SS and FS modes is controlled by IC_FS_SPKLEN registers. The registers are 8 bits wide and can be read and written through the APB interface, however, they can only be written when i2c is disabled. The minimum value that can be programmed to these registers is 1; values less than 1 are treated as 1.

12.2.3.4 SCL Rate Calculation

When i2c is running as an I2C master, in send and receive transmissions:

- IC_SS_SCL_LCNT/IC_FS_SCL_LCNT should bigger than IC_FS_SPKLEN + 7
- IC_SS_SCL_HCNT/IC_FS_SCL_HCNT should bigger than IC_FS_SPKLEN + 5

Regardless of SCL_Rise_time and SCL_Fall_time, the following is an example:

- The data rate of fast mode is 400kb/s; Indicates that the SCL period is $1/400\text{kHz} = 2.5\mu\text{s}$
- Minimum SCL high and low time:
 - MIN_SCL_LOWtime_FS = 1300ns
 - MIN_SCL_HIGHTime_FS = 600ns

- The APB clock takes 200MHz as an example, that is, the period is 5ns

SPKLEN does not directly affect the SCL frequency, but the master will check whether the SCL of the bus is pulled low by the slave when the master outputs SCL high. That is to say, the SCL input will be sent to the internal after SPKLEN, and then the master will decide whether to continue transmission according to the input SCL, so that SPLKEN will reduce the SCL frequency. In practice, it is also affected by the rise time and fall time of SCL. Theoretical calculated value after connecting the slave:

$$\text{scl_low_time_fs} = (\text{IC_FS_SCL_LCNT} + 1) * \text{apb_clk_period}$$

$$\text{scl_high_time_fs} = (\text{IC_FS_SCL_HCNT} + \text{IC_FS_SPKLEN} + 7) * \text{apb_clk_period}$$

if low time = 1600ns(bigger than 1300ns), high time = 900 ns(bigger than 600ns), IC_FS_SPKLEN = 10:

$$1600 = (\text{IC_FS_SCL_LCNT} + 1) * 5$$

$$900 = (\text{IC_FS_SCL_HCNT} + 10 + 7) * 5$$

Then:

$$\text{IC_FS_SCL_LCNT} = 0x1F3$$

$$\text{IC_FS_SCL_HCNT} = 0xA3$$

12.2.3.5 Master Mode

1. Disable the I2C by writing 0 to bit 0 of the IC_ENABLE register.
2. Write to the IC_CON register to set the maximum speed mode supported for slave operation (bits 2:1) and to specify whether the I2C starts its transfers in 7 bit addressing mode when the device is a slave (bit 3)
3. Write to the IC_TAR register the address of the I2C device to be addressed. It also indicates whether a General Call or a START BYTE command is going to be performed by I2C
4. Enable the I2C by writing a 1 to bit 0 of the IC_ENABLE register
5. Now write the transfer direction and data to be sent to the IC_DATA_CMD register

12.2.3.6 Master Transmit And Master Receive

The i2c supports switching back and forth between reading and writing dynamically. To transmit data, write the data to be written to the lower byte of the I2C Rx/Tx Data Buffer and Command Register (IC_DATA_CMD). The CMD bit [8] should be written to 0 for I2C write operations. Subsequently, a read command may be issued by writing "don't cares" to the lower byte of the IC_DATA_CMD register, and a 1 should be written to the CMD bit. The i2c master continues to initiate transfers as long as there are commands present in the transmit FIFO. If the transmit FIFO becomes empty it checks to see if IC_DATA_CMD[9] is set to 1.

- If set to 1, it issues a STOP condition after completing the current transfer.
- If set to 0, it holds SCL low until next command is written to the transmit FIFO

12.2.3.7 Slave Mode

To use the i2c as a slave, perform the following steps:

1. Disable the i2c by writing a '0' to bit 0 of the IC_ENABLE register.
2. Write to the IC_SAR register (bits 9:0) to set the slave address. This is the address to which the i2c responds.
3. Put the i2c in slave-only mode by writing '0' to the 6th bit (IC_SLAVE_DISABLE) of the IC_CON and '0' to the 0th bit (MASTER mode)..
4. Enable the i2c by writing a '1' in bit 0 of the IC_ENABLE register.

12.2.3.8 SDA Hold Time

The I2C protocol specification requires 300ns of hold time on the SDA signal ($t_{HD;DAT}$) in standard mode and fast mode, and a hold time long enough to bridge the undefined part between logic 1 and logic 0 of the falling edge of SCL in high speed mode and fast mode plus.

The i2c contains a software programmable register (IC_SDA_HOLD) to enable dynamic adjustment of the SDA hold-time.

The bits [15:0] 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] are used to extend the SDA transition (if any) whenever SCL is HIGH in the receiver (in either master or slave mode).

IC_SDA_HOLD register:

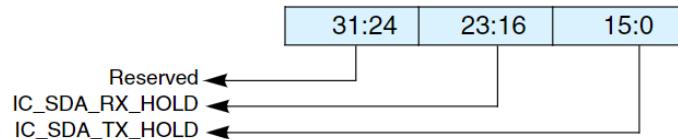


Figure 12-4 IC_SDA_HOLD Register

1. RX hold

When i2c acts as a receiver, according to the I2C protocol, the device should internally hold the SDA line to bridge undefined gap between logic 1 and logic 0 of SCL. IC_SDA_RX_HOLD can be used to change the internal hold time that I2C applies to an inbound SDA line. Each value in the IC_SDA_RX_HOLD register represents a unit of ic_clk period. The minimum value of the IC_SDA_RX_HOLD is 0. This hold time only applies when the SCL is HIGH. After changing to LOW inside the SCL, the receiver does not extend the SDA.

When the IC_SDA_RX_HOLD is equal to or greater than 3, the waveform is shown below

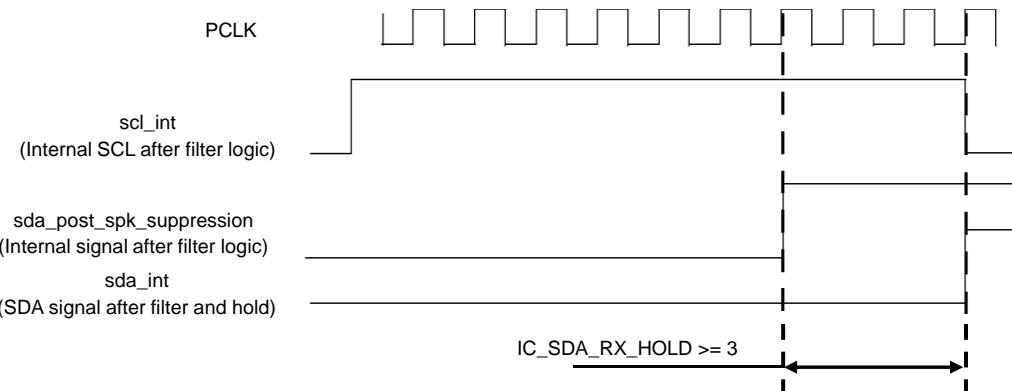


Figure 12-5 RX Hold time diagram

2. TX hold

Note: The hold time of tx cannot exceed $scl_lcnt - 2$

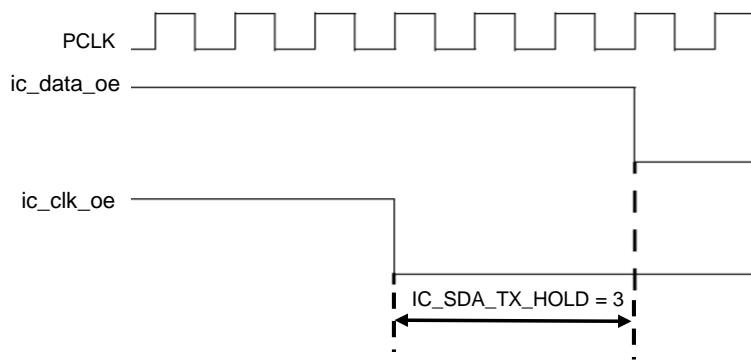


Figure 12-6 TX Hold time diagram.

12.2.4 Register Description

12.2.4.1 Register Overview

Register	Offset	Description
IC_CON	0x0	I2C Control Register
IC_TAR	0x4	I2C Target Address Register
IC_SAR	0x8	I2C Slave Address Register
IC_DATA_CMD	0x10	I2C Rx/Tx Data Buffer and Command Register
IC_SS_SCL_HCNT	0x14	Standard Speed I2C Clock SCL High Count Register
IC_SS_SCL_LCNT	0x18	Standard Speed I2C Clock SCL Low Count Register
IC_FS_SCL_HCNT	0x1c	Fast Mode or Fast Mode Plus I2C Clock SCL High Count Register
IC_FS_SCL_LCNT	0x20	Fast Mode or Fast Mode Plus I2C Clock SCL Low Count Register
IC_INTR_STAT	0x2c	I2C Interrupt Status Register
IC_INTR_MASK	0x30	I2C Interrupt Mask Register
IC_RAW_INTR_STAT	0x34	I2C Raw Interrupt Status Register
IC_RX_TL	0x38	I2C Receive FIFO Threshold Register
IC_TX_TL	0x3c	I2C Transmit FIFO Threshold Register
IC_CLR_INTR	0x40	Clear Combined and Individual Interrupt Register
IC_CLR_RX_UNDER	0x44	Clear RX_UNDER Interrupt Register
IC_CLR_RX_OVER	0x48	Clear RX_OVER Interrupt Register
IC_CLR_TX_OVER	0x4c	Clear TX_OVER Interrupt Register
IC_CLR_RD_REQ	0x50	Clear RD_REQ Interrupt Register

Register	Offset	Description
IC_CLR_TX_ABRT	0x54	Clear TX_ABRT Interrupt Register
IC_CLR_RX_DONE	0x58	Clear RX_DONE Interrupt Register
IC_CLR_ACTIVITY	0x5c	Clear ACTIVITY Interrupt Register
IC_CLR_STOP_DET	0x60	Clear STOP_DET Interrupt Register
IC_CLR_START_DET	0x64	Clear START_DET Interrupt Register
IC_CLR_GEN_CALL	0x68	Clear GEN_CALL Interrupt Register
IC_ENABLE	0x6c	I2C ENABLE Register
IC_STATUS	0x70	I2C STATUS Register
IC_TXFLR	0x74	I2C Transmit FIFO Level Register
IC_RXFLR	0x78	I2C Receive FIFO Level Register
IC_SDA_HOLD	0x7c	I2C SDA Hold Time Length Register
IC_TX_ABRT_SOURCE	0x80	I2C Transmit Abort Source Register
IC_SLV_DATA_NACK_ONLY	0x84	Generate Slave Data NACK Register
IC_DMA_CR	0x88	DMA Control Register
IC_DMA_TDLR	0x8c	DMA Transmit Data Level Register
IC_DMA_RDLR	0x90	DMA Receive Data Level Register
IC_SDA_SETUP	0x94	I2C SDA Setup Register
IC_ACK_GENERAL_CALL	0x98	I2C ACK General Call Register
IC_ENABLE_STATUS	0x9c	I2C Enable Status Register
IC_FS_SPKLEN	0xa0	I2C SS, FS or FM+ spike suppression limit
IC_CLR_RESTART_DET	0xa8	Clear RESTART_DET Interrupt Register
IC_SCL_STUCK_AT_LOW_TIMEOUT	0xac	I2C SCL Stuck at Low Timeout register
IC_SDA_STUCK_AT_LOW_TIMEOUT	0xb0	I2C SDA Stuck at Low Timeout register
IC_CLR_SCL_STUCK_DET	0xb4	Clear SCL Stuck at Low Detect interrupt Register
IC_SMBUS_CLK_LOW_SEXT	0xbc	SMBus Slave Clock Extend Timeout register
IC_SMBUS_CLK_LOW_MEXT	0xc0	SMBus Master Clock Extend Timeout register
IC_SMBUS_THIGH_MAX_IDLE_COUNT	0xc4	SMBus Master THigh MAX Bus-idle count Register
IC_SMBUS_INTR_STAT	0xc8	SMBus Interrupt Status Register

Register	Offset	Description
IC_SMBUS_INTR_MASK	0xcc	SMBus Interrupt Mask Register
IC_SMBUS_RAW_INTR_STAT	0xd0	SMBus Raw Interrupt Status Register
IC_CLR_SMBUS_INTR	0xd4	Clear SMBus Interrupt Register
IC_SMBUS_UDID_WORD0	0xdc	SMBUS ARP UDID WORD0 Register
IC_SMBUS_UDID_WORD1	0xe0	SMBUS ARP UDID WORD1 Register
IC_SMBUS_UDID_WORD2	0xe4	SMBUS ARP UDID WORD2 Register
IC_SMBUS_UDID_WORD3	0xe8	SMBUS ARP UDID WORD3 Register
REG_TIMEOUT_RST	0xf0	Register timeout counter reset value
IC_COMP_PARAM_1	0xf4	Component Parameter Register 1
IC_COMP_VERSION	0xf8	I2C Component Version Register
IC_COMP_TYPE	0xfc	I2C Component Type Register

12.2.4.2 Register Detail Description

IC_CON

Name: I2C Control Register

Description: I2C Control Register.

This register can be written only when the i2c is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.

Read/Write Access:

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

Size: 32 bits

Offset: 0x0

Bits	Name	Memory Access	Description
31:26	RSVD_IC_CON_2	R	IC_CON_2 Reserved bits - Read Only

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
25	RSVD_IC_SAR4_SMBUS_ARP_EN	R	IC_SAR4_SMBUS_ARP_EN Reserved bits - Read Only Value After Reset: 0x0
24	RSVD_IC_SAR3_SMBUS_ARP_EN	R	IC_SAR3_SMBUS_ARP_EN Reserved bits - Read Only Value After Reset: 0x0
23	RSVD_IC_SAR2_SMBUS_ARP_EN	R	IC_SAR2_SMBUS_ARP_EN Reserved bits - Read Only Value After Reset: 0x0
22:20			Reserved Field: Yes
19	SMBUS_PERSISTENT_SLV_ADDR_EN	R/W	<p>The bit controls to enable i2c slave as persistent or non persistent slave.</p> <p>If the slave is non-PSA then 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.</p> <p>This bit is applicable only in Slave mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): SMBus Persistent Slave address control is enabled. 0x0 (DISABLED): SMBus Persistent Slave address control is disabled. <p>Value After Reset: 0x0</p>
18	SMBUS_ARP_EN	R/W	<p>This bit controls whether i2c should enable Address Resolution Logic in SMBus Mode. The Slave mode will decode the Address Resolution Protocol commands and respond to it. The i2c slave also includes the generation/validity of PEC byte for Address Resolution Protocol commands. This bit is applicable only in Slave mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): SMBus ARP control is enabled. 0x0 (DISABLED): SMBus ARP control is disabled.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
17	SMBUS_SLAVE_QUICK_EN	R/W	<p>If this bit is set to 1, i2c slave only receives Quick commands in SMBus Mode.</p> <p>If this bit is set to 0, i2c slave receives all bus protocols but not Quick commands.</p> <p>This bit is applicable only in slave mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): SMBus Slave is enabled to receive Quick command. 0x0 (DISABLED): SMBus Slave is disabled to receive Quick command. <p>Value After Reset: 0x0</p>
16	RSVD_OPTIONAL_SAR_CTRL	R	<p>OPTIONAL_SAR_CTRL Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
15:12	RSVD_IC_CON_1	R	<p>IC_CON_1 Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
11	BUS_CLEAR_FEATURE_CTRL	R/W	<p>This bit controls whether the i2c enables the bus clear feature. For more information, refer to sections "Bus Clear Feature" and "SMBus/PMBus".</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): Bus Clear Feature is enabled. 0x0 (DISABLED): Bus Clear Feature is disabled. <p>Value After Reset: 0x0</p>
10	STOP_DET_IF_MASTER_ACTIVE	R/W	<p>In Master mode:</p> <p>1'b1: issues the STOP_DET interrupt only when master is active.</p> <p>1'b0: issues the STOP_DET irrespective of whether master is active or not.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): Master issues the STOP_DET interrupt only when master is active 0x0 (DISABLED): Master issues the STOP_DET interrupt irrespective of whether master is

Bits	Name	Memory Access	Description
			active or not Value After Reset: 0x0
9	RX_FIFO_FULL_HLD_CTRL	R/W	<p>This bit controls whether 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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): Hold bus when RX_FIFO is full 0x0 (DISABLED): Overflow when RX_FIFO is full <p>Value After Reset: 0x0</p>
8	TX_EMPTY_CTRL	R/W	<p>This bit controls the generation of the TX_EMPTY interrupt, as described in the IC_RAW_INTR_STAT register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): Controlled generation of TX_EMPTY interrupt 0x0 (DISABLED): Default behaviour of TX_EMPTY interrupt <p>Value After Reset: 0x0</p>
7	STOP_DET_IFADDRESSED	R/W	<p>In slave mode:</p> <p>1'b1: issues the STOP_DET interrupt only when it is addressed.</p> <p>1'b0: issues the STOP_DET irrespective of whether it's addressed or not.</p> <p>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).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): slave issues STOP_DET intr only if addressed 0x0 (DISABLED): slave issues STOP_DET intr always <p>Value After Reset: 0x0</p>
6	IC_SLAVE_DISABLE	R/W	This bit controls whether I2C has its slave

Bits	Name	Memory Access	Description
			<p>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.</p> <p>If this bit is set (slave is disabled), i2c functions only as a master and does not perform any action that requires a slave.</p> <p>Note: Software should ensure that if this bit is written with 0, then bit 0 should also be written with a 0.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (SLAVE_DISABLED): Slave mode is disabled 0x0 (SLAVE_ENABLED): Slave mode is enabled <p>Value After Reset: 0x1</p>
5	IC_RESTART_EN	R/W	<p>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 i2c operations. When RESTART is disabled, the master is prohibited from performing the following functions:</p> <ul style="list-style-type: none"> Sending a START BYTE Performing any high-speed mode operation High-speed mode operation Performing direction changes in combined format mode Performing a read operation with a 10-bit address <p>By replacing RESTART condition followed by a STOP and a subsequent START condition, split operations are broken down into multiple i2c transfers. If the above operations are performed, it will result in setting bit 6 (TX_ABRT) of the IC_RAW_INTR_STAT register.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x1 (ENABLED): Master restart enabled 0x0 (DISABLED): Master restart disabled Value After Reset: 0x1</p>
4	IC_10BITADDR_MASTER_rd_only	R	<p>If the I2C_DYNAMIC_TAR_UPDATE configuration parameter is set to 'No' (0), this bit is named IC_10BITADDR_MASTER and controls whether the 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.</p> <p>0: 7-bit addressing 1: 10-bit addressing Values: 0x1 (ADDR_10BITS): Master 10Bit addressing mode 0x0 (ADDR_7BITS): Master 7Bit addressing mode Value After Reset: 0x0</p>
3	IC_10BITADDR_SLAVE	R/W	<p>When acting as a slave, this bit controls whether the i2c responds to 7- or 10-bit addresses.</p> <p>0: 7-bit addressing. The i2c ignores transactions that involve 10-bit addressing; for 7-bit addressing, only the lower 7 bits of the IC_SAR register are compared.</p> <p>1: 10-bit addressing. The i2c responds to only 10-bit addressing transfers that match the full 10 bits of the IC_SAR register.</p> <p>Values: 0x1 (ADDR_10BITS): Slave 10Bit addressing 0x0 (ADDR_7BITS): Slave 7Bit addressing Value After Reset: 0x0</p>
2:1	SPEED	R/W	<p>These bits control at which speed the i2c operates; its setting is relevant only if one is operating the i2c in master mode. Hardware protects against illegal values being programmed by software. These bits must be</p>

Bits	Name	Memory Access	Description
			<p>programmed appropriately for slave mode also, as it is used to capture correct value of spike filter as per the speed mode.</p> <p>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.</p> <p>1: standard mode (100 kbit/s) 2: fast mode (<=400 kbit/s) or fast mode plus (<=1000Kbit/s) 3: high speed mode (3.4 Mbit/s)</p> <p>Note: This field is not applicable when IC_ULTRA_FAST_MODE=1</p> <p>Values:</p> <p>0x1 (STANDARD): Standard Speed mode of operation 0x2 (FAST): Fast or Fast Plus mode of operation 0x3 (HIGH): High Speed mode of operation</p> <p>Value After Reset: 0x2</p>
0	MASTER_MODE	R/W	<p>This bit controls whether the i2c master is enabled.</p> <p>Note: Software should ensure that if this bit is written with '1' then bit 6 should also be written with a '1'.</p> <p>Values:</p> <p>0x1 (ENABLED): Master mode is enabled 0x0 (DISABLED): Master mode is disabled</p> <p>Value After Reset: 0x1</p>

IC_TAR

Name: I2C Target Address Register

Description: I2C Target Address Register

If the configuration parameter I2C_DYNAMIC_TAR_UPDATE is set to 'No' (0), this register is 12 bits wide, and bits 31:12 are reserved. This register can be written to only when IC_ENABLE[0] is set to 0.

However, if I2C_DYNAMIC_TAR_UPDATE = 1, then the register becomes 13 bits wide. In this case, writes to IC_TAR succeed when one of the following conditions are true:

- i2c is NOT enabled (IC_ENABLE[0] is set to 0); or
- i2c is enabled (IC_ENABLE[0]=1); AND i2c is NOT engaged in any Master (tx, rx) operation (IC_STATUS[5]=0); AND i2c is enabled to operate in Master mode (IC_CON[0]=1); AND there are NO entries in the TX FIFO (IC_STATUS[2]=1)

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

- i2c is enabled (IC_ENABLE[0]=1); AND IC_EMPTYFIFO_HOLD_MASTER_EN configuration parameter is set to 1; AND i2c is enabled to operate in Master mode (IC_CON[0]=1); AND there are NO entries in the Tx FIFO and the master is in HOLD state (IC_INTR_STAT[13]=1).

Note: If the software or application is aware that the i2c is not using the TAR address for the pending commands in the Tx FIFO, then it is possible to update the TAR address even while the Tx FIFO has entries (IC_STATUS[2]=0).

- It is not necessary to perform any write to this register if i2c is enabled as an I2C slave only.

Size: 32 bits

Offset: 0x4

Bits	Name	Memory Access	Description
31:17	RSVD_IC_TAR_2	R	IC_TAR_2 Reserved bits - Read Only Value After Reset: 0x0
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 i2c. Values: 0x1 (ENABLED): Enables programming of QUICK-CMD transmission 0x0 (DISABLED): Disables programming of QUICK-CMD transmission Value After Reset: 0x0
15:14	RSVD_IC_TAR_1	R	IC_TAR_1 Reserved bits - Read Only Value After Reset: 0x0
13	RSVD_DEVICE_ID	R	DEVICE_ID Reserved bits - Read Only Value After Reset: 0x0
12	IC_10BITADDR_MASTER	R/W	This bit controls whether the i2c starts its transfers in 7- or 10-bit addressing mode when acting as a master. Values: 0x1 (ADDR_10BITS): Address 10Bit transmission format 0x0 (ADDR_7BITS): Address 7Bit transmission format Value After Reset: 0x0

Bits	Name	Memory Access	Description
11	SPECIAL	R/W	<p>This bit indicates whether software performs a Device-ID or General Call or START BYTE command.</p> <p>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</p> <p>Values:</p> <p>0x1 (ENABLED): Enables programming of GENERAL_CALL or START_BYTE transmission 0x0 (DISABLED): Disables programming of GENERAL_CALL or START_BYTE transmission</p> <p>Value After Reset: 0x0</p>
10	GC_OR_START	R/W	<p>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 i2c.</p> <p>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 i2c remains in General Call mode until the SPECIAL bit value (bit 11) is cleared.</p> <p>1: START BYTE</p> <p>Values:</p> <p>0x1 (START_BYTE): START byte transmission 0x0 (GENERAL_CALL): GENERAL_CALL byte transmission</p> <p>Value After Reset: 0x0</p>
9:0	IC_TAR	R/W	<p>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.</p> <p>If the IC_TAR and IC_SAR are the same, loopback exists but the FIFOs are shared between master and slave, so full loopback is not feasible. A master cannot transmit to itself; it can transmit to only a slave.</p> <p>Value After Reset: 0x55</p>

IC_SAR**Name:** I2C Slave Address Register**Description:** I2C Slave Address Register**Size:** 32 bits

Offset: 0x8

Bits	Name	Memory Access	Description
31:10	RSVD_IC_SAR	R	<p>IC_SAR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
9:0	IC_SAR	R/W	<p>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.</p> <p>This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.</p> <p>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. Refer to Table "I2C/SMBus Definition of Bits in First Byte" for a complete list of these reserved values.</p> <p>Value After Reset: 0x55</p> <p>Volatile: true</p>

IC_DATA_CMD

Name: I2C Rx/Tx Data Buffer and Command Register

Description: I2C Rx/Tx Data Buffer and Command Register; this is the register the CPU writes to when filling the TX FIFO and the CPU reads from when retrieving bytes from RX FIFO.

The size of the register changes as follows:

Write:

- 11 bits when IC_EMPTYFIFO_HOLD_MASTER_EN=1
- 9 bits when IC_EMPTYFIFO_HOLD_MASTER_EN=0

Read:

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

Note: In order for the i2c to continue acknowledging reads, a read command should be written for every byte that is to be received; otherwise the i2c will stop acknowledging.

Size: 32 bits

Offset: 0x10

Bits	Name	Memory Access	Description
31:12	RSVD_IC_DATA_CMD	R	<p>IC_DATA_CMD Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
11	FIRST_DATA_BYTE	R	<p>Indicates the first data byte received after the address phase for receive transfer in Master receiver or Slave receiver mode.</p> <p>Note: In case of APB_DATA_WIDTH=8,</p> <p>The user has to perform two APB Reads to IC_DATA_CMD in order to get status on 11 bit.</p> <p>In order to read the 11 bit, the user has to perform the first data byte read [7:0] (offset 0x10) and then perform the second read[15:8](offset 0x11) in order to know the status of 11 bit (whether the data received in previous read is a first data byte or not).</p> <p>The 11th bit is an optional read field, user can ignore 2nd byte read [15:8] (offset 0x11) if not interested in FIRST_DATA_BYTE status.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Non sequential data byte received 0x0 (INACTIVE): Sequential data byte received <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
10	RESTART	W	<p>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.</p> <p>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.</p> <p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLE): Issue RESTART before this command 0x0 (DISABLE): Do not Issue RESTART before this command <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

Bits	Name	Memory Access	Description
9	STOP	W	<p>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.</p> <p>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.</p> <p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLE): Issue STOP after this command 0x0 (DISABLE): Do not Issue STOP after this command <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
8	CMD	W	<p>This bit controls whether a read or a write is performed. This bit does not control the direction when the i2c acts as a slave. It controls only the direction when it acts as a master.</p> <p>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.</p> <p>When programming this bit, you should remember the following:</p> <p>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.</p> <p>If a "1" is written to this bit after receiving a RD_REQ interrupt, then a TX_ABRT interrupt occurs.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (READ): Master Read Command 0x0 (WRITE): Master Write Command <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
7:0	DAT	R/W	This register contains the data to be transmitted or received

Bits	Name	Memory Access	Description
			<p>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 i2c. However, when you read this register, these bits return the value of data received on the i2c interface.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

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

Bits	Name	Memory Access	Description
31:16	RSVD_IC_SS_SCL_HIGH_COUNT	R	<p>IC_SS_SCL_HCNT Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
15:0	IC_SS_SCL_HCNT	R/W	<p>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. For more information, refer to "IC_CLK Frequency Configuration".</p> <p>This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.</p> <p>The minimum valid value is derived through below formula:</p> $((\text{IC_ULTRA_FAST_MODE} == 1) ? 3 : ((\text{IC_CLK_FREQ_OPTIMIZATION} == 1) ? 1 : 6))$ <p>Hardware prevents values less than this being written, and if attempted results in minimum valid value being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of the i2c. The lower byte must be programmed first. Then the upper byte is programmed.</p> <p>When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.</p> <p>Note: This register must not be programmed to a</p>

Bits	Name	Memory Access	Description
			<p>value higher than 65525, because 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.</p> <p>Value After Reset: 0x320</p>

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

Bits	Name	Memory Access	Description
31:16	RSVD_IC_SS_SCL_LOW_COUNT	R	<p>RSVD_IC_SS_SCL_LOW_COUNT Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
15:0	IC_SS_SCL_LCNT	R/W	<p>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. For more information, refer to "IC_CLK Frequency Configuration"</p> <p>This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.</p> <p>The minimum valid value is derived through below formula:</p> $((\text{IC_ULTRA_FAST_MODE} == 1) ? 5 : ((\text{IC_CLK_FREQ_OPTIMIZATION} == 1) ? 6 : 8))$ <p>Hardware prevents values less than this being written, and if attempted, results in minimum valid value being set. For designs with APB_DATA_WIDTH = 8, the order of programming is important to ensure the correct operation of i2c. The lower byte must be programmed first, and then the upper byte is programmed.</p> <p>When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.</p> <p>Value After Reset: 0x3ac</p>

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

Bits	Name	Memory Access	Description
31:16	RSVD_IC_FS_SCL_HCNT	R	IC_FS_SCL_HCNT Reserved bits - Read Only Value After Reset: 0x0
15:0	IC_FS_SCL_HCNT	R/W	<p>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. For more information, refer to "IC_CLK Frequency Configuration".</p> <p>This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard.</p> <p>This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.</p> <p>The minimum valid value is derived through below formula:</p> $((\text{IC_ULTRA_FAST_MODE} == 1) ? 3 : ((\text{IC_CLK_FREQ_OPTIMIZATION} == 1) ? 1 : 6))$ <p>Hardware prevents values less than this being written, and if attempted results in minimum valid value being set. For designs with APB_DATA_WIDTH == 8 the order of programming is important to ensure the correct operation of the i2c. The lower byte must be programmed first. Then the upper byte is programmed.</p> <p>Value After Reset: 0x78</p>

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

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

Bits	Name	Memory Access	Description
15:0	IC_FS_SCL_LCNT	R/W	<p>This register must be set before any I2C bus transaction can take place to ensure proper I/O timing. This register sets the SCL clock low period count for fast speed. It is used in high-speed mode to send the Master Code and START BYTE or General CALL. For more information, refer to "IC_CLK Frequency Configuration".</p> <p>This register goes away and becomes read-only returning 0s if IC_MAX_SPEED_MODE = standard.</p> <p>This register can be written only when the I2C interface is disabled, which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.</p> <p>The minimum valid value is derived through below formula:</p> $((\text{IC_ULTRA_FAST_MODE} == 1) ? 5 : ((\text{IC_CLK_FREQ_OPTIMIZATION} == 1) ? 6 : 8))$ <p>Hardware prevents values less than this being written, and if attempted results in minimum valid value being set. For designs with APB_DATA_WIDTH = 8 the order of programming is important to ensure the correct operation of the i2c. The lower byte must be programmed first. Then the upper byte is programmed. If the value is less than minimum valid value then the count value gets changed to minimum valid value.</p> <p>When the configuration parameter IC_HC_COUNT_VALUES is set to 1, this register is read only.</p> <p>Value After Reset: 0x104</p>

IC_INTR_STAT

Name: I2C Interrupt Status Register

Description: I2C Interrupt Status Register.

Each bit in this register has a corresponding mask bit in the IC_INTR_MASK register. These bits are cleared by reading the matching interrupt clear register. The unmasked raw versions of these bits are available in the IC_RAW_INTR_STAT register.

Size: 32 bits

Offset: 0x2c

Bits	Name	Memory Access	Description
31:20	RSVD_IC_INTR_STAT	R	<p>IC_INTR_STAT Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

Bits	Name	Memory Access	Description
19	RSVD_R_SLV_ADDR4_TAG	R	R_SLV_ADDR4_TAG Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
18	RSVD_R_SLV_ADDR3_TAG	R	R_SLV_ADDR3_TAG Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
17	RSVD_R_SLV_ADDR2_TAG	R	R_SLV_ADDR2_TAG Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
16	RSVD_R_SLV_ADDR1_TAG	R	R_SLV_ADDR1_TAG Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
15	RSVD_R_WR_REQ	R	R_WR_REQ Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
14	R_SCL_STUCK_AT_LOW	R	See IC_RAW_INTR_STAT for a detailed description of R_SCL_STUCK_AT_LOW bit. Values: 0x1 (ACTIVE): R_SCL_STUCK_AT_LOW interrupt is active 0x0 (INACTIVE): R_SCL_STUCK_AT_LOW interrupt is inactive Value After Reset: 0x0 Volatile: true
13	R_MASTER_ON_HOLD	R	See IC_RAW_INTR_STAT for a detailed description of R_MASTER_ON_HOLD bit. Values: 0x1 (ACTIVE): R_MASTER_ON_HOLD interrupt is active 0x0 (INACTIVE): R_MASTER_ON_HOLD interrupt is

Bits	Name	Memory Access	Description
			<p>inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
12	R_RESTART_DET	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_RESTART_DET bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): R_RESTART_DET interrupt is active</p> <p>0x0 (INACTIVE): R_RESTART_DET interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
11	R_GEN_CALL	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_GEN_CALL bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): R_GEN_CALL interrupt is active</p> <p>0x0 (INACTIVE): R_GEN_CALL interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
10	R_START_DET	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_START_DET bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): R_START_DET interrupt is active</p> <p>0x0 (INACTIVE): R_START_DET interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
9	R_STOP_DET	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_STOP_DET bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): R_STOP_DET interrupt is active</p> <p>0x0 (INACTIVE): R_STOP_DET interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
8	R_ACTIVITY	R	See IC_RAW_INTR_STAT for a detailed description of R_ACTIVITY bit.

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x1 (ACTIVE): R_ACTIVITY interrupt is active</p> <p>0x0 (INACTIVE): R_ACTIVITY interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
7	R_RX_DONE	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_RX_DONE bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): R_RX_DONE interrupt is active</p> <p>0x0 (INACTIVE): R_RX_DONE interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
6	R_TX_ABRT	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_TX_ABRT bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): R_TX_ABRT interrupt is active</p> <p>0x0 (INACTIVE): R_TX_ABRT interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
5	R_RD_REQ	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_RD_REQ bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): R_RD_REQ interrupt is active</p> <p>0x0 (INACTIVE): R_RD_REQ interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
4	R_TX_EMPTY	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_TX_EMPTY bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): R_TX_EMPTY interrupt is active</p> <p>0x0 (INACTIVE): R_TX_EMPTY interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

Bits	Name	Memory Access	Description
3	R_TX_OVER	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_TX_OVER bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): R_TX_OVER interrupt is active 0x0 (INACTIVE): R_TX_OVER interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
2	R_RX_FULL	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_RX_FULL bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): R_RX_FULL interrupt is active 0x0 (INACTIVE): R_RX_FULL interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
1	R_RX_OVER	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_RX_OVER bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): R_RX_OVER interrupt is active 0x0 (INACTIVE): R_RX_OVER interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	R_RX_UNDER	R	<p>See IC_RAW_INTR_STAT for a detailed description of R_RX_UNDER bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): RX_UNDER interrupt is active 0x0 (INACTIVE): RX_UNDER interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

IC_INTR_MASK

Name: I2C Interrupt Mask Register

Description: I2C Interrupt Mask Register.

These bits mask their corresponding interrupt status bits. This register is active low; a value of 0 masks

the interrupt, whereas a value of 1 unmasks the interrupt.

Size: 32 bits

Offset: 0x30

Bits	Name	Memory Access	Description
31:20	RSVD_IC_INTR_STAT	R	IC_INTR_STAT Reserved bits - Read Only Value After Reset: 0x0
19	RSVD_M_SLV_ADDR4_TAG	R	M_SLV_ADDR4_TAG Register field Reserved bits - Read Only Value After Reset: 0x0
18	RSVD_M_SLV_ADDR3_TAG	R	M_SLV_ADDR3_TAG Register field Reserved bits - Read Only Value After Reset: 0x0
17	RSVD_M_SLV_ADDR2_TAG	R	M_SLV_ADDR2_TAG Register field Reserved bits - Read Only Value After Reset: 0x0
16	RSVD_M_SLV_ADDR1_TAG	R	M_SLV_ADDR1_TAG Register field Reserved bits - Read Only Value After Reset: 0x0
15	RSVD_M_WR_REQ	R	M_WR_REQ Register field Reserved bits - Read Only Value After Reset: 0x0
14	M_SCL_STUCK_AT_LOW	R/W	This bit masks the R_SCL_STUCK_AT_LOW interrupt in IC_INTR_STAT register. Values: 0x1 (DISABLED): SCL_STUCK_AT_LOW interrupt is unmasked 0x0 (ENABLED): SCL_STUCK_AT_LOW interrupt is masked Value After Reset: 0x1
13	M_MASTER_ON_HOLD	R/W	This bit masks the R_MASTER_ON_HOLD interrupt in IC_INTR_STAT register. Values: 0x1 (DISABLED): MASTER_ON_HOLD interrupt is unmasked 0x0 (ENABLED): MASTER_ON_HOLD interrupt is masked

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
12	M_RESTART_DET	R/W	<p>This bit masks the R_RESTART_DET interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): RESTART_DET interrupt is unmasked</p> <p>0x0 (ENABLED): RESTART_DET interrupt is masked</p> <p>Value After Reset: 0x0</p>
11	M_GEN_CALL	R/W	<p>This bit masks the R_GEN_CALL interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): GEN_CALL interrupt is unmasked</p> <p>0x0 (ENABLED): GEN_CALL interrupt is masked</p> <p>Value After Reset: 0x1</p>
10	M_START_DET	R/W	<p>This bit masks the R_START_DET interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): START_DET interrupt is unmasked</p> <p>0x0 (ENABLED): START_DET interrupt is masked</p> <p>Value After Reset: 0x0</p>
9	M_STOP_DET	R/W	<p>This bit masks the R_STOP_DET interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): STOP_DET interrupt is unmasked</p> <p>0x0 (ENABLED): STOP_DET interrupt is masked</p> <p>Value After Reset: 0x0</p>
8	M_ACTIVITY	R/W	<p>This bit masks the R_ACTIVITY interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): ACTIVITY interrupt is unmasked</p> <p>0x0 (ENABLED): ACTIVITY interrupt is masked</p> <p>Value After Reset: 0x0</p>
7	M_RX_DONE	R/W	This bit masks the R_RX_DONE interrupt in

Bits	Name	Memory Access	Description
			<p>IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): RX_DONE interrupt is unmasked</p> <p>0x0 (ENABLED): RX_DONE interrupt is masked</p> <p>Value After Reset: 0x1</p>
6	M_TX_ABRT	R/W	<p>This bit masks the R_TX_ABRT interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): TX_ABORT interrupt is unmasked</p> <p>0x0 (ENABLED): TX_ABORT interrupt is masked</p> <p>Value After Reset: 0x1</p>
5	M_RD_REQ	R/W	<p>This bit masks the R_RD_REQ interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): RD_REQ interrupt is unmasked</p> <p>0x0 (ENABLED): RD_REQ interrupt is masked</p> <p>Value After Reset: 0x1</p>
4	M_TX_EMPTY	R/W	<p>This bit masks the R_TX_EMPTY interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): TX_EMPTY interrupt is unmasked</p> <p>0x0 (ENABLED): TX_EMPTY interrupt is masked</p> <p>Value After Reset: 0x1</p>
3	M_TX_OVER	R/W	<p>This bit masks the R_TX_OVER interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): TX_OVER interrupt is unmasked</p> <p>0x0 (ENABLED): TX_OVER interrupt is masked</p> <p>Value After Reset: 0x1</p>
2	M_RX_FULL	R/W	<p>This bit masks the R_RX_FULL interrupt in IC_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): RX_FULL interrupt is unmasked</p>

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

IC_RAW_INTR_STAT

Name: I2C Raw Interrupt Status Register

Description: I2C Raw Interrupt Status Register.

Unlike the IC_INTR_STAT register, these bits are not masked so they always show the true status of the i2c.

Size: 32 bits

Offset: 0x34

Bits	Name	Memory Access	Description
31:20	RSVD_IC_RAW_INTR_STAT	R	IC_RAW_INTR_STAT Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
19	RSVD_SLV_ADDR4_TAG	R	SLV_ADDR4_TAG Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
18	RSVD_SLV_ADDR3_TAG	R	SLV_ADDR3_TAG Register field Reserved bits - Read Only

Bits	Name	Memory Access	Description
			<p>Value After Reset: 0x0</p> <p>Volatile: true</p>
17	RSVD_SLV_ADDR2_TAG	R	<p>SLV_ADDR2_TAG Register field Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
16	RSVD_SLV_ADDR1_TAG	R	<p>SLV_ADDR1_TAG Register field Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
15	RSVD_WR_REQ	R	<p>WR_REQ Register field Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
14	SCL_STUCK_AT_LOW	R	<p>Indicates whether the SCL Line is stuck at low for the IC_SCL_STUCK_LOW_TIMEOUT number of ic_clk periods.</p> <p>Enabled only when IC_BUS_CLEAR_FEATURE=1 and IC_ULTRA_FAST_MODE=0.</p> <p>Values:</p> <p>0x1 (ACTIVE): SCL_STUCK_AT_LOW interrupt is active</p> <p>0x0 (INACTIVE): SCL_STUCK_AT_LOW interrupt is inactive.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
13	MASTER_ON_HOLD	R	<p>Indicates whether master is holding the bus and TX FIFO is empty. Enabled only when I2C_DYNAMIC_TAR_UPDATE=1 and IC_EMPTYFIFO_HOLD_MASTER_EN=1.</p> <p>Values:</p> <p>0x1 (ACTIVE): MASTER_ON_HOLD interrupt is active</p> <p>0x0 (INACTIVE): MASTER_ON_HOLD interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
12	RESTART_DET	R	Indicates whether a RESTART condition has occurred on

Bits	Name	Memory Access	Description
			<p>the I2C interface when i2c is operating in Slave mode and the slave is being addressed.</p> <p>Enabled only when IC_SLV_RESTART_DET_EN=1.</p> <p>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 i2c does not generate the RESTART_DET interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): RESTART_DET interrupt is active 0x0 (INACTIVE): RESTART_DET interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
11	GEN_CALL	R	<p>Set only when a General Call address is received and it is acknowledged. It stays set until it is cleared either by disabling i2c or when the CPU reads bit 0 of the IC_CLR_GEN_CALL register. i2c stores the received data in the Rx buffer.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): GEN_CALL interrupt is active 0x0 (INACTIVE): GEN_CALL interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
10	START_DET	R	<p>Indicates whether a START or RESTART condition has occurred on the I2C interface regardless of whether i2c is operating in slave or master mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): START_DET interrupt is active 0x0 (INACTIVE): START_DET interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
9	STOP_DET	R	<p>Indicates whether a STOP condition has occurred on the I2C interface regardless of whether i2c is operating in slave or master mode.</p> <p>In Slave Mode:</p> <p>If IC_CON[7]=1'b1 (STOP_DET_IFADDRESSED), the STOP_DET interrupt will be issued only if slave is</p>

Bits	Name	Memory Access	Description
			<p>addressed.</p> <p>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).</p> <p>If IC_CON[7]=1'b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt is issued irrespective of whether it is being addressed.</p> <p>In Master Mode:</p> <p>If IC_CON[10]=1'b1 (STOP_DET_IF_MASTER_ACTIVE), the STOP_DET interrupt will be issued only if Master is active.</p> <p>If IC_CON[10]=1'b0 (STOP_DET_IFADDRESSED), the STOP_DET interrupt will be issued irrespective of whether master is active or not.</p> <p>Values:</p> <p>0x1 (ACTIVE): STOP_DET interrupt is active</p> <p>0x0 (INACTIVE): STOP_DET interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
8	ACTIVITY	R	<p>This bit captures i2c activity and stays set until it is cleared. There are four ways to clear it:</p> <ul style="list-style-type: none"> Disabling the i2c Reading the IC_CLR_ACTIVITY register Reading the IC_CLR_INTR register System reset <p>Once this bit is set, it stays set unless one of the four methods is used to clear it. Even if the i2c module is idle, this bit remains set until cleared, indicating that there was activity on the bus.</p> <p>Values:</p> <p>0x1 (ACTIVE): RAW_INTR_ACTIVITY interrupt is active</p> <p>0x0 (INACTIVE): RAW_INTR_ACTIVITY interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

Bits	Name	Memory Access	Description
7	RX_DONE	R	<p>When the 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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): RX_DONE interrupt is active 0x0 (INACTIVE): RX_DONE interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
6	TX_ABRT	R	<p>This bit indicates if 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.</p> <p>Note: The 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 flush because of TX_ABRT is controlled by the coreConsultant parameter IC_AVOID_RX_FIFO_FLUSH_ON_TX_ABRT.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): TX_ABRT interrupt is active 0x0 (INACTIVE): TX_ABRT interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
5	RD_REQ	R	<p>This bit is set to 1 when i2c is acting as a slave and another I2C master is attempting to read data from i2c. The 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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): RD_REQ interrupt is active

Bits	Name	Memory Access	Description
			<p>0x0 (INACTIVE): RD_REQ interrupt is inactive Value After Reset: 0x0 Volatile: true</p>
4	TX_EMPTY	R	<p>The behavior of the TX_EMPTY interrupt status differs based on the TX_EMPTY_CTRL selection in the IC_CON register.</p> <p>When TX_EMPTY_CTRL = 0:</p> <p>This bit is set to 1 when the transmit buffer is at or below the threshold value set in the IC_TX_TL register.</p> <p>When TX_EMPTY_CTRL = 1:</p> <p>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.</p> <p>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.</p> <p>Values:</p> <p>0x1 (ACTIVE): TX_EMPTY interrupt is active 0x0 (INACTIVE): TX_EMPTY interrupt is inactive Value After Reset: 0x0 Volatile: true</p>
3	TX_OVER	R	<p>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.</p> <p>Values:</p> <p>0x1 (ACTIVE): TX_OVER interrupt is active 0x0 (INACTIVE): TX_OVER interrupt is inactive Value After Reset: 0x0 Volatile: true</p>

Bits	Name	Memory Access	Description
2	RX_FULL	R	<p>Set when the receive buffer reaches or goes above the RX_TL threshold in the IC_RX_TL register. It is automatically cleared by hardware when buffer level goes below the threshold. If the module is disabled (IC_ENABLE[0]=0), the RX FIFO is flushed and held in reset; therefore the RX FIFO is not full. So this bit is cleared once the IC_ENABLE bit 0 is programmed with a 0, regardless of the activity that continues.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): RX_FULL interrupt is active 0x0 (INACTIVE): RX_FULL interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
1	RX_OVER	R	<p>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 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.</p> <p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): RX_OVER interrupt is active 0x0 (INACTIVE): RX_OVER interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	RX_UNDER	R	<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): RX_UNDER interrupt is active 0x0 (INACTIVE): RX_UNDER interrupt is inactive <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
			Volatile: true

IC_RX_TL**Name:** I2C Receive FIFO Threshold Register**Description:** I2C Receive FIFO Threshold Register**Size:** 32 bits**Offset:** 0x38

Bits	Name	Memory Access	Description
31:8	RSVD_IC_RX_TL	R	IC_RX_TL Reserved bits - Read Only Value After Reset: 0x0
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. Value After Reset: 0x0

IC_TX_TL**Name:** I2C Transmit FIFO Threshold Register**Description:** I2C Transmit FIFO Threshold Register**Size:** 32 bits**Offset:** 0x3c

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

Bits	Name	Memory Access	Description
			<p>255, with the additional restriction that it may not be set to value larger than the depth of the buffer. If an attempt is made to do that, the actual value set will be the maximum depth of the buffer. A value of 0 sets the threshold for 0 entries, and a value of 255 sets the threshold for 255 entries.</p> <p>Value After Reset: 0x0</p>

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

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_INTR	R	<p>CLR_INTR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	CLR_INTR	R	<p>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.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

IC_CLR_RX_UNDER**Name:** Clear RX_UNDER Interrupt Register**Description:** Clear RX_UNDER Interrupt Register**Size:** 32 bits**Offset:** 0x44

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_RX_UNDER	R	<p>IC_CLR_RX_UNDER Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

Bits	Name	Memory Access	Description
0	CLR_RX_UNDER	R	<p>Read this register to clear the RX_UNDER interrupt (bit 0) of the IC_RAW_INTR_STAT register.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

IC_CLR_RX_OVER**Name:** Clear RX_OVER Interrupt Register**Description:** Clear RX_OVER Interrupt Register**Size:** 32 bits**Offset:** 0x48

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

IC_CLR_TX_OVER**Name:** Clear TX_OVER Interrupt Register**Description:** Clear TX_OVER Interrupt Register**Size:** 32 bits**Offset:** 0x4c

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

Bits	Name	Memory Access	Description
			Value After Reset: 0x0 Volatile: true

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

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

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

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_TX_ABRT	R	IC_CLR_TX_ABRT Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
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

Bits	Name	Memory Access	Description
			<p>the TX FIFO. Refer to Bit 9 of the IC_TX_ABRT_SOURCE register for an exception to clearing IC_TX_ABRT_SOURCE.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

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

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

IC_CLR_ACTIVITY**Name:** Clear ACTIVITY Interrupt Register**Description:** Clear ACTIVITY Interrupt Register**Size:** 32 bits**Offset:** 0x5c

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_ACTIVITY	R	<p>IC_CLR_ACTIVITY Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	CLR_ACTIVITY	R	<p>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</p>

Bits	Name	Memory Access	Description
			<p>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.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

IC_CLR_STOP_DET**Name:** Clear STOP_DET Interrupt Register**Description:** Clear STOP_DET Interrupt Register**Size:** 32 bits**Offset:** 0x60

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

IC_CLR_START_DET**Name:** Clear START_DET Interrupt Register**Description:** Clear START_DET Interrupt Register**Size:** 32 bits**Offset:** 0x64

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_START_DET	R	<p>IC_CLR_START_DET Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	CLR_START_DET	R	Read this register to clear the START_DET interrupt (bit

Bits	Name	Memory Access	Description
			10) of the IC_RAW_INTR_STAT register. Value After Reset: 0x0 Volatile: true

IC_CLR_GEN_CALL**Name:** Clear GEN_CALL Interrupt Register**Description:** Clear GEN_CALL Interrupt Register**Size:** 32 bits**Offset:** 0x68

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

IC_ENABLE**Name:** I2C ENABLE Register**Description:** I2C Enable Register**Size:** 32 bits**Offset:** 0x6c

Bits	Name	Memory Access	Description
31:23	RSVD_IC_ENABLE_2	R	IC_ENABLE Reserved bits - Read Only Value After Reset: 0x0
22	RSVD_IC_SAR4_EN	R	IC_SAR4_EN Register field Reserved bits - Read Only Value After Reset: 0x0

Bits	Name	Memory Access	Description
21	RSVD_IC_SAR3_EN	R	IC_SAR3_EN Register field Reserved bits - Read Only Value After Reset: 0x0
20	RSVD_IC_SAR2_EN	R	IC_SAR2_EN Register field Reserved bits - Read Only Value After Reset: 0x0
19	RSVD_IC_SAR_EN	R	IC_SAR_EN Register field Reserved bits - Read Only Value After Reset: 0x0
18	RSVD_SMBUS_ALERT_EN	R	SMBUS_ALERT_EN Register field Reserved bits - Read Only Value After Reset: 0x0
17	RSVD_SMBUS_SUSPEND_EN	R	SMBUS_SUSPEND_EN Register field Reserved bits - Read Only Value After Reset: 0x0
16	SMBUS_CLK_RESET	R/W	This bit is used in SMBus Host mode to initiate the SMBus Master Clock Reset. This bit should be enabled only when Master is in idle. Whenever this bit is enabled, the SMBCLK is held low for the IC_SCL_STUCK_TIMEOUT ic_clk cycles to reset the SMBus slave devices. Values: 0x1 (ENABLED): Master initiates the SMBUS Clock Reset Mechanism. 0x0 (DISABLED): Master will not initiate SMBUS Clock Reset Mechanism. Value After Reset: 0x0
15:4	RSVD_IC_ENABLE_1	R	RSVD_IC_ENABLE_1 Reserved bits - Read Only Value After Reset: 0x0
3	SDA_STUCK_RECOVERY_ENABLE	R/W	If SDA is stuck at low indicated through the TX_ABORT interrupt (IC_TX_ABRT_SOURCE[17]), then this bit is used as a control knob to initiate the SDA Recovery Mechanism (that is, send at most 9 SCL clocks and STOP to release the SDA line) and then this bit gets auto clear Values: 0x1 (SDA_STUCK_RECOVERY_ENABLED): Master

Bits	Name	Memory Access	Description
			<p>initiates the SDA stuck at low recovery mechanism.</p> <p>0x0 (SDA_STUCK_RECOVERY_DISABLED): Master disabled the SDA stuck at low recovery mechanism.</p> <p>Value After Reset: 0x0</p>
2	TX_CMD_BLOCK	R/W	<p>In Master mode:</p> <p>1'b1: Blocks the transmission of data on I2C bus even if Tx FIFO has data to transmit.</p> <p>1'b0: The transmission of data starts on I2C bus automatically, as soon as the first data is available in the Tx FIFO.</p> <p>Note: To block the execution of Master commands, set the TX_CMD_BLOCK bit only when Tx FIFO is empty (IC_STATUS[2]==1) and Master is in Idle state (IC_STATUS[5] == 0). Any further commands put in the Tx FIFO are not executed until TX_CMD_BLOCK bit is unset.</p> <p>Values:</p> <p>0x1 (BLOCKED): Tx Command execution blocked</p> <p>0x0 (NOT_BLOCKED): Tx Command execution not blocked</p> <p>Value After Reset: 0x0</p>
1	ABORT	R/W	<p>When set, the controller initiates the transfer abort.</p> <p>0: ABORT not initiated or ABORT done</p> <p>1: ABORT operation in progress</p> <p>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.</p> <p>For a detailed description on how to abort I2C transfers, refer to "Aborting I2C Transfers".</p> <p>Values:</p> <p>0x1 (ENABLED): ABORT operation in progress</p>

Bits	Name	Memory Access	Description
			<p>0x0 (DISABLE): ABORT operation not in progress Value After Reset: 0x0</p>
0	ENABLE	R/W	<p>Controls whether the i2c is enabled. 0: Disables i2c (TX and RX FIFOs are held in an erased state) 1: Enables i2c</p> <p>Software can disable i2c while it is active. However, it is important that care be taken to ensure that i2c is disabled properly. A recommended procedure is described in "Disabling i2c".</p> <p>When i2c is disabled, the following occurs:</p> <ul style="list-style-type: none"> The TX FIFO and RX FIFO get flushed. Status bits in the IC_INTR_STAT register are still active until 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 i2c stops the current transfer at the end of the current transfer and does not acknowledge the transfer. In systems with asynchronous pclk and ic_clk when IC_CLK_TYPE parameter set to asynchronous (1), there is a two ic_clk delay when enabling or disabling the i2c. For a detailed description on how to disable i2c, refer to "Disabling i2c" <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): I2C is enabled 0x0 (DISABLED): I2C is disabled <p>Value After Reset: 0x0</p>

IC_STATUS

Name: I2C STATUS Register

Description: I2C Status Register.

This is a read-only register used to indicate the current transfer status and FIFO status. The status register may be read at any time. None of the bits in this register request an interrupt.

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

- Bits 1 and 2 are set to 1

- Bits 3 and 10 are set to 0

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

- Bits 5 and 6 are set to 0

Size: 32 bits

Offset: 0x70

Bits	Name	Memory Access	Description
31:27	RSVD_IC_STATUS_2	R	IC_STATUS Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
26:21			Reserved Field: Yes
20	RSVD_SMBUS_ALERT_STATUS	R	SMBUS_ALERT_STATUS Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
19	RSVD_SMBUS_SUSPEND_STATUS	R	SMBUS_SUSPEND_STATUS Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
18	SMBUS_SLAVE_ADDR_RESOLVED	R	This bit indicates whether the slave address (ic_sar) is resolved by the ARP Master. Values: 0x1 (ACTIVE): SMBUS Slave Address is Resolved. 0x0 (INACTIVE): SMBUS Slave Address is not Resolved. Value After Reset: 0x0 Volatile: true
17	SMBUS_SLAVE_ADDR_VALID	R	This bit indicates whether the slave address (ic_sar) is valid or not. Values: 0x1 (ACTIVE): SMBUS Slave Address is Valid. 0x0 (INACTIVE): SMBUS Slave Address is not valid.

Bits	Name	Memory Access	Description
			<p>Value After Reset: 0x0</p> <p>Volatile: true</p>
16	SMBUS_QUICK_CMD_BIT	R	<p>This bit indicates the R/W bit of the Quick command received. This bit will be cleared after the user has read this bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): SMBUS QUICK CMD Read/write is set to 1.</p> <p>0x0 (INACTIVE): SMBUS QUICK CMD Read/write is set to 0.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
15:13	RSVD_IC_STATUS_1	R	<p>RSVD_IC_STATUS_1 Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
12	RSVD_SLV_ISO_SAR_DATA_CLK_STRETCH	R	<p>SLV_ISO_SAR_DATA_CLK_STRETCH Register field Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
11	SDA_STUCK_NOT_RECOVERED	R	<p>This bit indicates that SDA stuck at low is not recovered after the recovery mechanism. In Slave mode, this register bit is not applicable.</p> <p>Values:</p> <p>0x1 (ACTIVE): SDA Stuck at low is not recovered after recovery mechanism.</p> <p>0x0 (INACTIVE): SDA Stuck at low is recovered after recovery mechanism.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
10	SLV_HOLD_RX_FIFO_FULL	R	<p>This bit indicates the BUS Hold in Slave mode due to Rx FIFO is Full and an additional byte has been received (This kind of Bus hold is applicable if IC_RX_FULL_HLD_BUS_EN is set to 1 and bit 9 of the IC_CON register</p>

Bits	Name	Memory Access	Description
			<p>(RX_FIFO_FULL_HLD_CTRL) is programmed to HIGH).</p> <p>Values:</p> <p>0x1 (ACTIVE): Slave holds the bus due to Rx FIFO is full</p> <p>0x0 (INACTIVE): Slave is not holding the bus or Bus hold is not due to Rx FIFO is full</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
9	SLV_HOLD_TX_FIFO_EMPTY	R	<p>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.</p> <p>Values:</p> <p>0x1 (ACTIVE): Slave holds the bus due to Tx FIFO is empty</p> <p>0x0 (INACTIVE): Slave is not holding the bus or Bus hold is not due to Tx FIFO is empty</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
8	MST_HOLD_RX_FIFO_FULL	R	<p>This bit indicates the BUS Hold in Master mode due to Rx FIFO is Full and additional byte has been received (This kind of Bus hold is applicable if IC_RX_FULL_HLD_BUS_EN is set to 1 and bit 9 of the IC_CON register (RX_FIFO_FULL_HLD_CTRL) is programmed to HIGH).</p> <p>Values:</p> <p>0x1 (ACTIVE): Master holds the bus due to Rx FIFO is full</p> <p>0x0 (INACTIVE): Master is not holding the bus or Bus hold is not due to Rx FIFO is full</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
7	MST_HOLD_TX_FIFO_EMPTY	R	If the

Bits	Name	Memory Access	Description
			<p>IC_EMPTYFIFO_HOLD_MASTER_EN parameter is set to 1, the 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 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).</p> <p>Values:</p> <p>0x1 (ACTIVE): Master holds the bus due to Tx FIFO is empty</p> <p>0x0 (INACTIVE): Master is not holding the bus or Bus hold is not due to Tx FIFO is empty</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
6	SLV_ACTIVITY	R	<p>Slave FSM Activity Status. When the Slave Finite State Machine (FSM) is not in the IDLE state, this bit is set.</p> <p>Values:</p> <p>0x1 (ACTIVE): Slave not idle</p> <p>0x0 (IDLE): Slave is idle</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
5	MST_ACTIVITY	R	<p>Master FSM Activity Status. When the Master Finite State Machine (FSM) is not in the IDLE state, this bit is set.</p> <p>Note: IC_STATUS[0]-that is, ACTIVITY bit-is the OR of SLV_ACTIVITY and MST_ACTIVITY bits.</p> <p>Values:</p> <p>0x1 (ACTIVE): Master not idle</p> <p>0x0 (IDLE): Master is idle</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
4	RFF	R	Receive FIFO Completely Full. When the

Bits	Name	Memory Access	Description
			<p>receive FIFO is completely full, this bit is set. When the receive FIFO contains one or more empty location, this bit is cleared.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (FULL): Rx FIFO is full 0x0 (NOT_FULL): Rx FIFO not full <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
3	RFNE	R	<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (NOT_EMPTY): Rx FIFO not empty 0x0 (EMPTY): Rx FIFO is empty <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
2	TFE	R	<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (EMPTY): Tx FIFO is empty 0x0 (NON_EMPTY): Tx FIFO not empty <p>Value After Reset: 0x1</p> <p>Volatile: true</p>
1	TFNF	R	<p>Transmit FIFO Not Full. Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (NOT_FULL): Tx FIFO not full 0x0 (FULL): Tx FIFO is full <p>Value After Reset: 0x1</p>

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

IC_TXFLR

Name: I2C Transmit FIFO Level Register

Description: I2C Transmit FIFO Level Register.

This register contains the number of valid data entries in the transmit FIFO buffer. It is cleared whenever:

- The I2C is disabled
- There is a transmit abort - that is, TX_ABRT bit is set in the IC_RAW_INTR_STAT register
- The slave bulk transmit mode is aborted

The register increments whenever data is placed into the transmit FIFO and decrements when data is taken from the transmit FIFO.

Size: 32 bits

Offset: 0x74

Bits	Name	Memory Access	Description
31:6	RSVD_TXFLR	R	TXFLR Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
5:0	TXFLR	R	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. Value After Reset: 0x0 Volatile: true

IC_RXFLR

Name: I2C Receive FIFO Level Register

Description: I2C Receive FIFO Level Register.

This register contains the number of valid data entries in the receive FIFO buffer. It is cleared whenever:

- The I2C is disabled
- Whenever there is a transmit abort caused by any of the events tracked in IC_TX_ABRT_SOURCE

The register increments whenever data is placed into the receive FIFO and decrements when data is taken from the receive FIFO.

Size: 32 bits

Offset: 0x78

Bits	Name	Memory Access	Description
31:6	RSVD_RXFLR	R	<p>RXFLR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
5:0	RXFLR	R	<p>Receive FIFO Level. Contains the number of valid data entries in the receive FIFO.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

IC_SDA_HOLD

Name: I2C SDA Hold Time Length Register

Description: I2C SDA Hold Time Length Register.

The bits [15:0] of this register are used to control the hold time of SDA during transmit in both slave and master mode (after SCL goes from HIGH to LOW).

The bits [23:16] of this register are used to extend the SDA transition (if any) whenever SCL is HIGH in the receiver in either master or slave mode.

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

The values in this register are in units of ic_clk period. The value programmed in IC_SDA_TX_HOLD must be greater than the minimum hold time in each mode one cycle in master mode, seven cycles in slave mode for the value to be implemented.

The programmed SDA hold time during transmit (IC_SDA_TX_HOLD) cannot exceed at any time the duration of the low part of scl. Therefore the programmed value cannot be larger than N_SCL_LOW-2, where N_SCL_LOW is the duration of the low part of the scl period measured in ic_clk cycles.

Size: 32 bits

Offset: 0x7c

Bits	Name	Memory Access	Description
31:24	RSVD_IC_SDA_HOLD	R	<p>IC_SDA_HOLD Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
23:16	IC_SDA_RX_HOLD	R/W	Sets the required SDA hold time in units of ic_clk period, when i2c acts as a receiver. Value After Reset: 0x0
15:0	IC_SDA_TX_HOLD	R/W	Sets the required SDA hold time in units of ic_clk period, when i2c acts as a transmitter. Value After Reset: 0x1

IC_TX_ABRT_SOURCE**Name:** I2C Transmit Abort Source Register**Description:** I2C Transmit Abort Source Register.

This register has 32 bits that indicate the source of the TX_ABRT bit. Except for Bit 9, this register is cleared whenever the IC_CLR_TX_ABRT register or the IC_CLR_INTR register is read. To clear bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; RESTART must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]).

Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, Bit 9 clears for one cycle and is then re-asserted.

Size: 32 bits**Offset:** 0x80

Bits	Name	Memory Access	Description
31:23	TX_FLUSH_CNT	R	This field indicates the number of Tx FIFO Data Commands which are flushed due to TX_ABRT interrupt. It is cleared whenever I2C is disabled. Role of i2c: Master-Transmitter or Slave-Transmitter Value After Reset: 0x0 Volatile: true
22:21	RSVD_IC_TX_ABRT_SOURCE	R	IC_TX_ABRT_SOURCE Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
20:18	RSVD_ABRT_DEVICE_WRITE	R	ABRT_DEVICE_WRITE Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
17	ABRT_SDA_STUCK_AT_LOW	R	This is a master-mode-only bit. Master detects the

Bits	Name	Memory Access	Description
			<p>SDA Stuck at low for the IC_SDA_STUCK_AT_LOW_TIMEOUT value of ic_clks.</p> <p>Role of i2c: Master</p> <p>Values:</p> <p>0x1 (ACTIVE): This abort is generated because of Sda stuck at low for IC_SDA_STUCK_AT_LOW_TIMEOUT value of ic_clks</p> <p>0x0 (INACTIVE): This abort is not generated</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
16	ABRT_USER_ABRT	R	<p>This is a master-mode-only bit. Master has detected the transfer abort (IC_ENABLE[1])</p> <p>Role of i2c: Master-Transmitter</p> <p>Values:</p> <p>0x1 (ABRT_USER_ABRT_GENERATED): Transfer abort detected by master</p> <p>0x0 (ABRT_USER_ABRT_VOID): Transfer abort detected by master- scenario not present</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
15	ABRT_SLVRD_INTX	R	<p>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.</p> <p>Role of i2c: Slave-Transmitter</p> <p>Values:</p> <p>0x1 (ABRT_SLVRD_INTX_GENERATED): Slave trying to transmit to remote master in read mode</p> <p>0x0 (ABRT_SLVRD_INTX_VOID): Slave trying to transmit to remote master in read mode- scenario not present</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
14	ABRT_SLV_ARBLOST	R	<p>This field indicates that a Slave has lost the bus while transmitting data to a remote master. IC_TX_ABRT_SOURCE[12] is set at the same time.</p> <p>Note: Even though the slave never 'owns' the bus, something could go wrong on the bus. This is a fail safe</p>

Bits	Name	Memory Access	Description
			<p>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 i2c no longer own the bus.</p> <p>Role of i2c: Slave-Transmitter</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_SLV_ARBLOST_GENERATED): Slave lost arbitration to remote master 0x0 (ABRT_SLV_ARBLOST_VOID): Slave lost arbitration to remote master- scenario not present <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
13	ABRT_SLVFLUSH_TXFIFO	R	<p>This field specifies that the Slave has received a read command and some data exists in the TX FIFO, so the slave issues a TX_ABRT interrupt to flush old data in TX FIFO.</p> <p>Role of i2c: Slave-Transmitter</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_SLVFLUSH_TXFIFO_GENERATED): Slave flushes existing data in TX-FIFO upon getting read command 0x0 (ABRT_SLVFLUSH_TXFIFO_VOID): Slave flushes existing data in TX-FIFO upon getting read command- scenario not present <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
12	ARB_LOST	R	<p>This field specifies that the Master has lost arbitration, or if IC_TX_ABRT_SOURCE[14] is also set, then the slave transmitter has lost arbitration.</p> <p>Role of i2c: Master-Transmitter or Slave-Transmitter</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_LOST_GENERATED): Master or Slave-Transmitter lost arbitration 0x0 (ABRT_LOST_VOID): Master or Slave-Transmitter lost arbitration- scenario not present <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
11	ABRT_MASTER_DIS	R	This field indicates that the User tries to initiate a

Bits	Name	Memory Access	Description
			<p>Master operation with the Master mode disabled.</p> <p>Role of i2c: Master-Transmitter or Master-Receiver</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_MASTER_DIS_GENERATED): User initiating master operation when MASTER disabled 0x0 (ABRT_MASTER_DIS_VOID): User initiating master operation when MASTER disabled- scenario not present <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
10	ABRT_10B_RD_NORSTRT	R	<p>This field indicates that the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the master sends a read command in 10-bit addressing mode.</p> <p>Role of i2c: Master-Receiver</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_10B_RD_GENERATED): Master trying to read in 10Bit addressing mode when RESTART disabled 0x0 (ABRT_10B_RD_VOID): Master not trying to read in 10Bit addressing mode when RESTART disabled <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
9	ABRT_SBYTE_NORSTRT	R	<p>To clear Bit 9, the source of the ABRT_SBYTE_NORSTRT must be fixed first; restart must be enabled (IC_CON[5]=1), the SPECIAL bit must be cleared (IC_TAR[11]), or the GC_OR_START bit must be cleared (IC_TAR[10]). Once the source of the ABRT_SBYTE_NORSTRT is fixed, then this bit can be cleared in the same manner as other bits in this register. If the source of the ABRT_SBYTE_NORSTRT is not fixed before attempting to clear this bit, bit 9 clears for one cycle and then gets reasserted. When this field is set to 1, the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to send a START Byte.</p> <p>Role of i2c: Master</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_SBYTE_NORSTRT_GENERATED): User trying to send START byte when RESTART disabled 0x0 (ABRT_SBYTE_NORSTRT_VOID): User trying to send START byte when RESTART disabled- scenario

Bits	Name	Memory Access	Description
			<p>not present</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
8	ABRT_HS_NORSTRT	R	<p>This field indicates that the restart is disabled (IC_RESTART_EN bit (IC_CON[5]) =0) and the user is trying to use the master to transfer data in High Speed mode.</p> <p>Role of i2c: Master-Transmitter or Master-Receiver</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_HS_NORSTRT_GENERATED): User trying to switch Master to HS mode when RESTART disabled 0x0 (ABRT_HS_NORSTRT_VOID): User trying to switch Master to HS mode when RESTART disabled- scenario not present <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
7	ABRT_SBYTE_ACKDET	R	<p>This field indicates that the Master has sent a START Byte and the START Byte was acknowledged (wrong behavior).</p> <p>Role of i2c: Master</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_SBYTE_ACKDET_GENERATED): ACK detected for START byte 0x0 (ABRT_SBYTE_ACKDET_VOID): ACK detected for START byte- scenario not present <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
6	ABRT_HS_ACKDET	R	<p>This field indicates that the Master is in High Speed mode and the High Speed Master code was acknowledged (wrong behavior).</p> <p>Role of i2c: Master</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_HS_ACK_GENERATED): HS Master code ACKed in HS Mode 0x0 (ABRT_HS_ACK_VOID): HS Master code ACKed in HS Mode- scenario not present <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
			Volatile: true
5	ABRT_GCALL_READ	R	<p>This field indicates that i2c in the master mode has sent a General Call but the user programmed the byte following the General Call to be a read from the bus (IC_DATA_CMD[9] is set to 1).</p> <p>Role of i2c: Master-Transmitter</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_GCALL_READ_GENERATED): GCALL is followed by read from bus 0x0 (ABRT_GCALL_READ_VOID): GCALL is followed by read from bus-scenario not present <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
4	ABRT_GCALL_NOACK	R	<p>This field indicates that i2c in master mode has sent a General Call and no slave on the bus acknowledged the General Call.</p> <p>Role of i2c: Master-Transmitter</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_GCALL_NOACK_GENERATED): GCALL not ACKed by any slave 0x0 (ABRT_GCALL_NOACK_VOID): GCALL not ACKed by any slave-scenario not present <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
3	ABRT_TXDATA_NOACK	R	<p>This field indicates the master-mode only bit. When the master receives an acknowledgement for the address, but when it sends data byte(s) following the address, it did not receive an acknowledge from the remote slave(s).</p> <p>Role of i2c: Master-Transmitter</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ABRT_TXDATA_NOACK_GENERATED): Transmitted data not ACKed by addressed slave 0x0 (ABRT_TXDATA_NOACK_VOID): Transmitted data non-ACKed by addressed slave-scenario not present <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

Bits	Name	Memory Access	Description
2	ABRT_10ADDR2_NOACK	R	<p>This field indicates that the Master is in 10-bit address mode and that the second address byte of the 10-bit address was not acknowledged by any slave.</p> <p>Role of i2c: Master-Transmitter or Master-Receiver</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Byte 2 of 10Bit Address not ACKed by any slave 0x0 (INACTIVE): This abort is not generated <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
1	ABRT_10ADDR1_NOACK	R	<p>This field indicates that the Master is in 10-bit address mode and the first 10-bit address byte was not acknowledged by any slave.</p> <p>Role of i2c: Master-Transmitter or Master-Receiver</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Byte 1 of 10Bit Address not ACKed by any slave 0x0 (INACTIVE): This abort is not generated <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	ABRT_7B_ADDR_NOACK	R	<p>This field indicates that the Master is in 7-bit addressing mode and the address sent was not acknowledged by any slave.</p> <p>Role of i2c: Master-Transmitter or Master-Receiver</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): This abort is generated because of NOACK for 7-bit address 0x0 (INACTIVE): This abort is not generated <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

IC_SLV_DATA_NACK_ONLY**Name:** Generate Slave Data NACK Register**Description:** Generate Slave Data NACK Register

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

- i2c is disabled (IC_ENABLE[0] = 0)
- Slave part is inactive (IC_STATUS[6] = 0)

Note: The IC_STATUS[6] is a register read-back location for the internal slv_activity signal; the user should poll this before writing the ic_slv_data_nack_only bit.

Size: 32 bits

Offset: 0x84

Bits	Name	Memory Access	Description
31:1	RSVD_IC_SLV_DATA_NACK_ONLY	R	<p>IC_SLV_DATA_NACK_ONLY Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
0	NACK	R/W	<p>Generate NACK. This NACK generation only occurs when 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.</p> <p>When the register is set to a value of 0, it generates NACK/ACK, depending on normal criteria.</p> <p>1: generate NACK after data byte received 0: generate NACK/ACK normally</p> <p>Values:</p> <p>0x1 (ENABLED): Slave receiver generates NACK upon data reception only</p> <p>0x0 (DISABLED): Slave receiver generates NACK normally</p> <p>Value After Reset: 0x0</p>

IC_DMA_CR

Name: DMA Control Register

Description: DMA Control Register.

This register is only valid when i2c is configured with a set of DMA Controller interface signals (IC_HAS_DMA = 1). When i2c is not configured for DMA operation, this register does not exist and writing to the register's address has no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation. There is a separate bit for transmit and receive. This can be programmed regardless of the state of IC_ENABLE.

Size: 32 bits

Offset: 0x88

Bits	Name	Memory Access	Description
31:2	RSVD_IC_DMA_CR_2_31	R	RSVD_IC_DMA_CR_2_31 Reserved bits - Read Only Value After Reset: 0x0
1	TDMAE	R/W	Transmit DMA Enable. This bit enables/disables the transmit FIFO DMA channel. Values: 0x1 (ENABLED): Transmit FIFO DMA channel enabled 0x0 (DISABLED): transmit FIFO DMA channel disabled Value After Reset: 0x0
0	RDMAE	R/W	Receive DMA Enable. This bit enables/disables the receive FIFO DMA channel. Values: 0x1 (ENABLED): Receive FIFO DMA channel enabled 0x0 (DISABLED): Receive FIFO DMA channel disabled Value After Reset: 0x0

IC_DMA_TDLR

Name: DMA Transmit Data Level Register

Description: DMA Transmit Data Level Register.

This register is only valid when the i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Size: 32 bits

Offset: 0x8c

Bits	Name	Memory Access	Description
31:5	RSVD_DMA_TDLR	R	DMA_TDLR Reserved bits - Read Only Value After Reset: 0x0
4:0	DMATDL	R/W	Transmit Data Level. This bit field controls the level at which a DMA request is made by the transmit logic. It is equal to the watermark level; that is, the dma_tx_req signal is generated when the number of valid data entries in the transmit FIFO is equal to or below this field value, and TDMAE = 1. Value After Reset: 0x0

IC_DMA_RDLR**Name:** DMA Receive Data Level Register**Description:** I2C Receive Data Level Register.

This register is only valid when i2c is configured with a set of DMA interface signals (IC_HAS_DMA = 1). When i2c is not configured for DMA operation, this register does not exist; writing to its address has no effect; reading from its address returns zero.

Size: 32 bits**Offset:** 0x90

Bits	Name	Memory Access	Description
31:5	RSVD_DMA_RDLR	R	DMA_RDLR Reserved bits - Read Only Value After Reset: 0x0
4:0	DMARDL	R/W	Receive Data Level. This bit field controls the level at which a DMA request is made by the receive logic. The watermark level = DMARDL+1; that is, dma_rx_req is generated when the number of valid data entries in the receive FIFO is equal to or more than this field value + 1, and RDMAE =1. For instance, when DMARDL is 0, then dma_rx_req is asserted when 1 or more data entries are present in the receive FIFO. Value After Reset: 0x0

IC_SDA_SETUP**Name:** I2C SDA Setup Register**Description:** I2C SDA Setup Register.

This register controls the amount of time delay (in terms of number of ic_clk clock periods) introduced in the rising edge of SCL - relative to SDA changing - when i2c services a read request in a slave-transmitter operation. The relevant I2C requirement is tSU:DAT as detailed in the I2C Bus Specification. This register must be programmed with a value greater than or equal to one ic_clock_period of time.

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

Note: The length of setup time is calculated using below formula:

- If IC_CLK_FREQ_OPTIMIZATION = 0 : [(IC_SDA_SETUP - 1) * (ic_clock_period)]
- If IC_CLK_FREQ_OPTIMIZATION = 1 : [(IC_SDA_SETUP + 1) * (ic_clock_period)]

The IC_SDA_SETUP register is only used by the i2c when operating as a slave transmitter.

Size: 32 bits**Offset:** 0x94

Bits	Name	Memory Access	Description
31:8	RSVD_IC_SDA_SETUP	R	IC_SDA_SETUP Reserved bits - Read Only

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
7:0	SDA_SETUP	R/W	<p>SDA Setup (tSU:DAT).</p> <p>It is recommended that if the required delay is 1000ns, then for an ic_clk frequency of 10 MHz:</p> <ul style="list-style-type: none"> - If IC_CLK_FREQ_OPTIMIZATION = 0 : IC_SDA_SETUP should be programmed to a value of 11. - If IC_CLK_FREQ_OPTIMIZATION = 1 : IC_SDA_SETUP should be programmed to a value of 9. <p>Value After Reset: 0x64</p>

IC_ACK_GENERAL_CALL

Name: I2C ACK General Call Register

Description: I2C ACK General Call Register.

The register controls whether i2c responds with a ACK or NACK when it receives an I2C General Call address.

This register is applicable only when the i2c is in slave mode.

Size: 32 bits

Offset: 0x98

Bits	Name	Memory Access	Description
31:1	RSVD_IC_ACK_GEN_1_31	R	<p>RSVD_IC_ACK_GEN_1_31 Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
0	ACK_GEN_CALL	R/W	<p>ACK General Call. When set to 1, i2c responds with a ACK (by asserting ic_data_oe) when it receives a General Call. Otherwise, i2c responds with a NACK (by negating ic_data_oe).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): Generate ACK for a General Call 0x0 (DISABLED): Generate NACK for General Call <p>Value After Reset: 0x1</p>

IC_ENABLE_STATUS

Name: I2C Enable Status Register

Description: I2C Enable Status Register.

The register is used to report the i2c hardware status when the IC_ENABLE[0] register is set from 1 to 0; that is, when 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 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 i2c depends on I2C bus activities.

Size: 32 bits

Offset: 0x9c

Bits	Name	Memory Access	Description
31:3	RSVD_IC_ENABLE_STATUS	R	<p>IC_ENABLE_STATUS Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
2	SLV_RX_DATA_LOST	R	<p>Slave Received Data Lost. This bit indicates if a Slave-Receiver operation has been aborted with at least one data byte received from an I2C transfer due to the setting bit 0 of IC_ENABLE from 1 to 0. When read as 1, 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.</p> <p>Note: If the remote I2C master terminates the transfer with a STOP condition before the 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.</p> <p>When read as 0, i2c is deemed to have been disabled without being actively involved in the data phase of a Slave-Receiver transfer.</p> <p>Note: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Slave RX Data is lost 0x0 (INACTIVE): Slave RX Data is not lost <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
1	SLV_DISABLED_WHILE_BUSY	R	<p>Slave Disabled While Busy (Transmit, Receive). This bit indicates if a potential or active Slave operation has been aborted due to the setting bit 0 of the IC_ENABLE register from 1 to 0. This bit is set when the CPU writes a 0 to the IC_ENABLE register while:</p> <ul style="list-style-type: none"> (a) i2c is receiving the address byte of the Slave-Transmitter operation from a remote master;

Bits	Name	Memory Access	Description
			<p>OR,</p> <p>(b) address and data bytes of the Slave-Receiver operation from a remote master.</p> <p>When read as 1, 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 i2c (IC_SAR register) OR if the transfer is completed before IC_ENABLE is set to 0 but has not taken effect.</p> <p>Note: If the remote I2C master terminates the transfer with a STOP condition before the i2c has a chance to NACK a transfer, and IC_ENABLE[0] has been set to 0, then this bit will also be set to 1.</p> <p>When read as 0, i2c is deemed to have been disabled when there is master activity, or when the I2C bus is idle.</p> <p>Note: The CPU can safely read this bit when IC_EN (bit 0) is read as 0.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Slave is disabled when it is active 0x0 (INACTIVE): Slave is disabled when it is idle <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	IC_EN	R	<p>IC Enable Status. This bit always reflects the value driven on the output port ic_en.</p> <p>When read as 1, i2c is deemed to be in an enabled state.</p> <p>When read as 0, i2c is deemed completely inactive.</p> <p>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).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): I2C enabled 0x0 (DISABLED): I2C disabled <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

IC_FS_SPKLEN

Name: I2C SS, FS or FM+ spike suppression limit

Description: I2C SS, FS or FM+ Spike Suppression Limit.

This register is used to store the duration, measured in ic_clk cycles, of the longest spike that is filtered out by the spike suppression logic when the component is operating in SS, FS or FM+ modes. The relevant I2C requirement is tSP (table 4) as detailed in the I2C Bus Specification. This register must be programmed with a minimum value of 1.

Size: 32 bits

Offset: 0xa0

Bits	Name	Memory Access	Description
31:8	RSVD_IC_FS_SPKLEN	R	IC_FS_SPKLEN Reserved bits - Read Only Value After Reset: 0x0
7:0	IC_FS_SPKLEN	R/W	This register must be set before any I2C bus transaction can take place to ensure stable operation. This register sets the duration, measured in ic_clk cycles, of the longest spike in the SCL or SDA lines that will be filtered out by the spike suppression logic. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect. The minimum valid value is 1; hardware prevents values less than this being written, and if attempted results in 1 being set. For more information, refer to "Spike Suppression". Value After Reset: 0xa

IC_CLR_RESTART_DET

Name: Clear RESTART_DET Interrupt Register

Description: Clear RESTART_DET Interrupt Register

Size: 32 bits

Offset: 0xa8

Bits	Name	Memory Access	Description
31:1	RSVD_IC_CLR_RESTART_DET	R	IC_CLR_RESTART_DET Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
0	CLR_RESTART_DET	R	Read this register to clear the RESTART_DET interrupt (bit 12) of IC_RAW_INTR_STAT register. Value After Reset: 0x0 Volatile: true

IC_SCL_STUCK_AT_LOW_TIMEOUT**Name:** I2C SCL Stuck at Low Timeout register**Description:** I2C SCL Stuck at Low Timeout.

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

Size: 32 bits**Offset:** 0xac

Bits	Name	Memory Access	Description
31:0	IC_SCL_STUCK_LOW_TIMEOUT	R/W	<p>i2c generate the interrupt to indicate SCL stuck at low (ic_scl_stuck_at_low_intr/ic_scl_stuck_at_low_intr_n) if it detects the SCL stuck at low for the IC_SCL_STUCK_LOW_TIMEOUT in units of ic_clk period. This register can be written only when the I2C interface is disabled which corresponds to the IC_ENABLE[0] register being set to 0. Writes at other times have no effect.</p> <p>Value After Reset: 0xffffffff</p>

IC_SDA_STUCK_AT_LOW_TIMEOUT**Name:** I2C SDA Stuck at Low Timeout register**Description:** I2C SDA Stuck at Low Timeout.

This register is used to store the duration, measured in ic_clk cycles, used to Recover the Data (SDA) line through sending SCL pulses if SDA is held low for the mentioned duration.

Size: 32 bits**Offset:** 0xb0

Bits	Name	Memory Access	Description
31:0	IC_SDA_STUCK_LOW_TIMEOUT	R/W	<p>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.</p> <p>Value After Reset: 0xffffffff</p>

IC_CLR_SCL_STUCK_DET**Name:** Clear SCL Stuck at Low Detect interrupt Register**Description:** Clear SCL Stuck at Low Detect Interrupt Register**Size:** 32 bits

Offset: 0xb4

Bits	Name	Memory Access	Description
31:1	RSVD_CLR_SCL_STUCK_DET	R	CLR_SCL_STUCK_DET Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
0	CLR_SCL_STUCK_DET	R	Read this register to clear the SCL_STUCT_AT_LOW interrupt (bit 15) of the IC_RAW_INTR_STAT register. Value After Reset: 0x0 Volatile: true

IC_SMBUS_CLK_LOW_SEXT**Name:** SMBus Slave Clock Extend Timeout register**Description:** SMBus Slave Clock Extend Timeout Register.

This Register contains the Timeout value used to determine the Slave Clock Extend Timeout in one transfer (from START to STOP).

This Register can be written only when the i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS is set to 1. This register is used to store the duration, measured in ic_clk cycles, used to detect the slave clock extend timeout if slave extends the clock (SCL) for the mentioned duration.

Size: 32 bits**Offset:** 0xbc

Bits	Name	Memory Access	Description
31:0	SMBUS_CLK_LOW_SEXT_TIMEOUT	R/W	This field is used to detect the Slave Clock Extend timeout (tLOW:SEXT) in master mode extended by the slave device in one message from the initial START to the STOP. The values in this register are in units of ic_clk period. Value After Reset: 0xffffffff

IC_SMBUS_CLK_LOW_MEXT**Name:** SMBus Master Clock Extend Timeout register**Description:** SMBus Master Clock Extend Timeout Register.

This Register contains the Timeout value used to determine the Master Clock Extend Timeout in one byte of transfer.

This register can be written only when the i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS is set to 1. This register is used to store the duration, measured in ic_clk cycles, used to detect the Master clock extend timeout if Master

extends the clock (SCL) for the mentioned duration.

Size: 32 bits

Offset: 0xc0

Bits	Name	Memory Access	Description
31:0	SMBUS_CLK_LOW_MEXT_TIMEOUT	R/W	<p>This field is used to detect the Master extend SMBus clock (SCLK) timeout defined from START-to-ACK, ACK-to-ACK, or ACK-to-STOP in Master mode. The values in this register are in units of ic_clk period.</p> <p>Value After Reset: 0xffffffff</p>

IC_SMBUS_THIGH_MAX_IDLE_COUNT

Name: SMBus Master THigh MAX Bus-idle count Register

Description: SMBus Master THigh MAX Bus-idle count Register.

This register programs the Bus-idle time period used when a master has been dynamically added to the bus or when a master has generated a clock reset on the bus. This register is used to store the duration, measured in ic_clk cycles, used to detect the Bus Idle condition if SCL and SDA are held high for the mentioned duration. This register can be written only when the i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS is set to 1.

Size: 32 bits

Offset: 0xc4

Bits	Name	Memory Access	Description
31:16	RSVD_SMBUS_THIGH_MAX_BUS_IDLE_CNT	R	<p>SMBUS_THIGH_MAX_BUS_IDLE_CNT Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
15:0	SMBUS_THIGH_MAX_BUS_IDLE_CNT	R/W	<p>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.</p> <p>In this case, the master must wait long enough to ensure that a transfer is not currently in progress. The values in this register are in units of ic_clk period.</p> <p>Value After Reset: 0xffff</p>

IC_SMBUS_INTR_STAT**Name:** SMBus Interrupt Status Register**Description:** SMBUS Interrupt Status Register.

Each bit in this register has a corresponding mask bit in the IC_SMBUS_INTR_MASK register. These bits are cleared by writing the matching SMBus interrupt clear register (IC_CLR_SMBUS_INTR) bits. The unmasked raw versions of these bits are available in the IC_SMBUS_RAW_INTR_STAT register.

Size: 32 bits**Offset:** 0xc8

Bits	Name	Memory Access	Description
31:11	RSVD_IC_SMBUS_INTR_STAT	R	IC_SMBUS_INTR_STAT Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
10	RSVD_R_SMBUS_ALERT_DET	R	R_SMBUS_ALERT_DET Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
9	RSVD_R_SMBUS_SUSPEND_DET	R	R_SMBUS_SUSPEND_DET Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
8	R_SLV_RX_PEC_NACK	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_SLV_RX_PEC_NACK bit. Values: 0x1 (ACTIVE): SLV_RX_PEC_NACK interrupt is active 0x0 (INACTIVE): SLV_RX_PEC_NACK interrupt is inactive Value After Reset: 0x0 Volatile: true
7	R_ARP_ASSGN_ADDR_CMD_DET	R	See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_ARP_ASSGN_ADDR_CMD_DET bit. Values: 0x1 (ACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_ASSGN_ADDR_CMD_DET

Bits	Name	Memory Access	Description
			<p>interrupt is inactive Value After Reset: 0x0 Volatile: true</p>
6	R_ARP_GET_UDID_CMD_DET	R	<p>See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_ARP_GET_UDID_CMD_DET bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): ARP_GET_UDID_CMD_DET interrupt is active</p> <p>0x0 (INACTIVE): ARP_GET_UDID_CMD_DET interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
5	R_ARP_RST_CMD_DET	R	<p>See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_ARP_RST_CMD_DET bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): ARP_RST_CMD_DET interrupt is active</p> <p>0x0 (INACTIVE): ARP_RST_CMD_DET interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
4	R_ARP_PREPARE_CMD_DET	R	<p>See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_ARP_PREPARE_CMD_DET bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): ARP_PREPARE_CMD_DET interrupt is active</p> <p>0x0 (INACTIVE): ARP_PREPARE_CMD_DET interrupt is inactive</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
3	R_HOST_NOTIFY_MST_DET	R	<p>See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_HOST_NOTIFY_MST_DET bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): HOST_NOTIFY_MST_DET interrupt is active</p> <p>0x0 (INACTIVE): HOST_NOTIFY_MST_DET</p>

Bits	Name	Memory Access	Description
			<p>interrupt is inactive Value After Reset: 0x0 Volatile: true</p>
2	R_QUICK_CMD_DET	R	<p>See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_QUICK_CMD_DET bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): QUICK_CMD_DET interrupt is active 0x0 (INACTIVE): QUICK_CMD_DET interrupt is inactive</p> <p>Value After Reset: 0x0 Volatile: true</p>
1	R_MST_CLOCK_EXTND_TIMEOUT	R	<p>See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_MST_CLOCK_EXTND_TIMEOUT bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): MST_CLOCK_EXTND_TIMEOUT interrupt is active 0x0 (INACTIVE): MST_CLOCK_EXTND_TIMEOUT interrupt is inactive</p> <p>Value After Reset: 0x0 Volatile: true</p>
0	R_SLV_CLOCK_EXTND_TIMEOUT	R	<p>See IC_SMBUS_RAW_INTR_STAT for a detailed description of R_SLV_CLOCK_EXTND_TIMEOUT bit.</p> <p>Values:</p> <p>0x1 (ACTIVE): SLV_CLOCK_EXTND_TIMEOUT interrupt is active 0x0 (INACTIVE): SLV_CLOCK_EXTND_TIMEOUT interrupt is inactive</p> <p>Value After Reset: 0x0 Volatile: true</p>

IC_SMBUS_INTR_MASK**Name:** SMBus Interrupt Mask Register**Description:** SMBus Interrupt Mask Register

Size: 32 bits**Offset:** 0xcc

Bits	Name	Memory Access	Description
31:11	RSVD_IC_SMBUS_INTR_MASK	R	IC_SMBUS_INTR_MASK Reserved bits - Read Only Value After Reset: 0x0
10:9	RSVD_IC_SMBUS_INTR_MASK_9_10	R	RSVD_IC_SMBUS_INTR_MASK_9_10 Register field Reserved bits. Value After Reset: 0x0
8	M_SLV_RX_PEC_NACK	R/W	This bit masks the R_SLV_RX_PEC_NACK interrupt in IC_SMBUS_INTR_STAT register. Values: 0x1 (DISABLED): SLV_RX_PEC_NACK interrupt is unmasked 0x0 (ENABLED): SLV_RX_PEC_NACK interrupt is masked Value After Reset: 0x1
7	M_ARP_ASSGN_ADDR_CMD_DET	R/W	This bit masks the R_ARP_ASSGN_ADDR_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. Values: 0x1 (DISABLED): ARP_ASSGN_ADDR_CMD_DET interrupt is unmasked 0x0 (ENABLED): ARP_ASSGN_ADDR_CMD_DET interrupt is masked Value After Reset: 0x1
6	M_ARP_GET_UDID_CMD_DET	R/W	This bit masks the R_ARP_GET_UDID_CMD_DET interrupt in IC_SMBUS_INTR_STAT register. Values: 0x1 (DISABLED): ARP_GET_UDID_CMD_DET interrupt is unmasked 0x0 (ENABLED): ARP_GET_UDID_CMD_DET interrupt is masked Value After Reset: 0x1

Bits	Name	Memory Access	Description
5	M_ARP_RST_CMD_DET	R/W	<p>This bit masks the R_ARP_RST_CMD_DET interrupt in IC_SMBUS_INTR_STAT register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (DISABLED): ARP_RST_CMD_DET interrupt is unmasked 0x0 (ENABLED): ARP_RST_CMD_DET interrupt is masked <p>Value After Reset: 0x1</p>
4	M_ARP_PREPARE_CMD_DET	R/W	<p>This bit masks the R_ARP_PREPARE_CMD_DET interrupt in IC_SMBUS_INTR_STAT register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (DISABLED): ARP_PREPARE_CMD_DET interrupt is unmasked 0x0 (ENABLED): ARP_PREPARE_CMD_DET interrupt is masked <p>Value After Reset: 0x1</p>
3	M_HOST_NOTIFY_MST_DET	R/W	<p>This bit masks the R_HOST_NOTIFY_MST_DET interrupt in IC_SMBUS_INTR_STAT register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (DISABLED): HOST_NOTIFY_MST_DET interrupt is unmasked 0x0 (ENABLED): HOST_NOTIFY_MST_DET interrupt is masked <p>Value After Reset: 0x1</p>
2	M_QUICK_CMD_DET	R/W	<p>This bit masks the R_QUICK_CMD_DET interrupt in IC_SMBUS_INTR_STAT register.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (DISABLED): QUICK_CMD_DET interrupt is unmasked 0x0 (ENABLED): QUICK_CMD_DET interrupt is masked <p>Value After Reset: 0x1</p>
1	M_MST_CLOCK_EXTND_TIMEOUT	R/W	<p>This bit masks the R_MST_CLOCK_EXTND_TIMEOUT interrupt in IC_SMBUS_INTR_STAT register.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x1 (DISABLED): MST_CLOCK_EXTND_TIMEOUT interrupt is unmasked</p> <p>0x0 (ENABLED): MST_CLOCK_EXTND_TIMEOUT interrupt is masked</p> <p>Value After Reset: 0x1</p>
0	M_SLV_CLOCK_EXTND_TIMEOUT	R/W	<p>This bit masks the R_SLV_CLOCK_EXTND_TIMEOUT interrupt in IC_SMBUS_INTR_STAT register.</p> <p>Values:</p> <p>0x1 (DISABLED): SLV_CLOCK_EXTND_TIMEOUT interrupt is unmasked</p> <p>0x0 (ENABLED): SLV_CLOCK_EXTND_TIMEOUT interrupt is masked</p> <p>Value After Reset: 0x1</p>

IC_SMBUS_RAW_INTR_STAT**Name:** SMBus Raw Interrupt Status Register**Description:** SMBus Raw Interrupt Status Register.

Unlike the IC_SMBUS_INTR_STAT register, these bits are not masked so they always show the true status of the i2c.

Size: 32 bits**Offset:** 0xd0

Bits	Name	Memory Access	Description
31:11	RSVD_IC_SMBUS_RAW_INTR_STAT	R	<p>IC_SMBUS_RAW_INTR_STAT Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
10	RSVD_SMBUS_ALERT_DET	R	<p>SMBUS_ALERT_DET Register field Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
			Volatile: true
9	RSVD_SMBUS_SUSPEND_DET	R	SMBUS_SUSPEND_DET Register field Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
8	SLV_RX_PEC_NACK	R	Indicates whether a NACK has been sent due to PEC mismatch while working as ARP slave. Values: 0x1 (ACTIVE): SLV_RX_PEC_NACK interrupt is active 0x0 (INACTIVE): SLV_RX_PEC_NACK interrupt is inactive Value After Reset: 0x0 Volatile: true
7	ARP_ASSGN_ADDR_CMD_DET	R	Indicates whether an Assign Address ARP command has been received. Values: 0x1 (ACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_ASSGN_ADDR_CMD_DET interrupt is inactive Value After Reset: 0x0 Volatile: true
6	ARP_GET_UDID_CMD_DET	R	Indicates whether a Get UDID ARP command has been received. Values: 0x1 (ACTIVE): ARP_GET_UDID_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_GET_UDID_CMD_DET interrupt is inactive Value After Reset: 0x0 Volatile: true
5	ARP_RST_CMD_DET	R	Indicates whether a General or Directed Reset ARP command has been received. Values:

Bits	Name	Memory Access	Description
			<p>0x1 (ACTIVE): ARP_RST_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_RST_CMD_DET interrupt is inactive Value After Reset: 0x0 Volatile: true</p>
4	ARP_PREPARE_CMD_DET	R	<p>Indicates whether a prepare to ARP command has been received. Values: 0x1 (ACTIVE): ARP_PREPARE_CMD_DET interrupt is active 0x0 (INACTIVE): ARP_PREPARE_CMD_DET interrupt is inactive Value After Reset: 0x0 Volatile: true</p>
3	HOST_NTFY_MST_DET	R	<p>Indicates whether a Notify ARP Master ARP command has been received. Values: 0x1 (ACTIVE): HOST_NTFY_MST_DET interrupt is active 0x0 (INACTIVE): HOST_NTFY_MST_DET interrupt is inactive Value After Reset: 0x0 Volatile: true</p>
2	QUICK_CMD_DET	R	<p>Indicates whether a Quick command has been received on the SMBus interface regardless of whether i2c is operating in slave or master mode. Enabled only when IC_SMBUS=1 is set to 1. Values: 0x1 (ACTIVE): Quick Command interrupt is active 0x0 (INACTIVE): Quick Command interrupt is inactive Value After Reset: 0x0 Volatile: true</p>
1	MST_CLOCK_EXTND_TIMEOUT	R	Indicates whether the Master device

Bits	Name	Memory Access	Description
			<p>transaction (START-to-ACK, ACK-to-ACK, or ACK-to-STOP) from START to STOP exceeds IC_SMBUS_CLOCK_LOW_MEXT time within each byte of message.</p> <p>This bit is enabled only when:</p> <ul style="list-style-type: none"> IC_SMBUS=1 IC_CON[0]=1 IC_EMPTYFIFO_HOLD_MASTER_EN=1 or IC_RX_FULL_HLD_BUS_EN=1 <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Master Clock Extend Timeout interrupt is active 0x0 (INACTIVE): Master Clock Extend Timeout interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	SLV_CLOCK_EXTND_TIMEOUT	R	<p>Indicates whether the transaction from Slave (i.e. from START to STOP) exceeds IC_SMBUS_CLK_LOW_SEXT time.</p> <p>This bit is enabled only when:</p> <ul style="list-style-type: none"> IC_SMBUS=1 IC_CON[0]=1 <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Slave Clock Extend Timeout interrupt is active 0x0 (INACTIVE): Slave Clock Extend Timeout interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

IC_CLR_SMBUS_INTR**Name:** Clear SMBus Interrupt Register**Description:** SMBus Clear Interrupt Register**Size:** 32 bits**Offset:** 0xd4

Bits	Name	Memory Access	Description
31:11	RSVD_IC_CLR_SMBUS_INTR	W	IC_CLR_SMBUS_INTR Reserved bits - Read Only Value After Reset: 0x0
10	RSVD_CLR_SMBUS_ALERT_DET	W	CLR_SMBUS_ALERT_DET Register field Reserved bits - Read Only Value After Reset: 0x0
9	RSVD_CLR_SMBUS_SUSPEND_DET	W	CLR_SMBUS_SUSPEND_DET Register field Reserved bits - Read Only Value After Reset: 0x0
8	CLR_SLV_RX_PEC_NACK	W	Write this register bit to clear the SLV_RX_PEC_NACK interrupt (bit 8) of the IC_SMBUS_RAW_INTR_STAT register. Value After Reset: 0x0
7	CLR_ARP_ASSGN_ADDR_CMD_DET	W	Write this register bit to clear the ARP_ASSGN_ADDR_CMD_DET interrupt (bit 7) of the IC_SMBUS_RAW_INTR_STAT register. Value After Reset: 0x0
6	CLR_ARP_GET_UDID_CMD_DET	W	Write this register bit to clear the ARP_GET_UDID_CMD_DET interrupt (bit 6) of the IC_SMBUS_RAW_INTR_STAT register. Value After Reset: 0x0
5	CLR_ARP_RST_CMD_DET	W	Write this register bit to clear the ARP_RST_CMD_DET interrupt (bit 5) of the IC_SMBUS_RAW_INTR_STAT register. Value After Reset: 0x0
4	CLR_ARP_PREPARE_CMD_DET	W	Write this register bit to clear the ARP_PREPARE_CMD_DET interrupt (bit 4) of the IC_SMBUS_RAW_INTR_STAT register. Value After Reset: 0x0
3	CLR_HOST_NOTIFY_MST_DET	W	Write this register bit to clear the HOST_NOTIFY_MST_DET interrupt (bit 3) of the IC_SMBUS_RAW_INTR_STAT register. Value After Reset: 0x0
2	CLR_QUICK_CMD_DET	W	Write this register bit to clear the QUICK_CMD_DET interrupt (bit 2) of the IC_SMBUS_RAW_INTR_STAT register.

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
1	CLR_MST_CLOCK_EXTND_TIMEOUT	W	<p>Write this register bit to clear the MST_CLOCK_EXTND_TIMEOUT interrupt (bit 1) of the IC_SMBUS_RAW_INTR_STAT register.</p> <p>Value After Reset: 0x0</p>
0	CLR_SLV_CLOCK_EXTND_TIMEOUT	W	<p>Write this register bit to clear the SLV_CLOCK_EXTND_TIMEOUT interrupt (bit 0) of the IC_SMBUS_RAW_INTR_STAT register.</p> <p>Value After Reset: 0x0</p>

IC_SMBUS_UDID_WORD0**Name:** SMBUS ARP UDID WORD0 Register**Description:** SMBUS UDID WORD0 Register

This Register can be written only when the i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS_UDID_HC is set to 0. This register is used to store the Lower 32 bit value of Slave UDID register i.e. UDID[31:0] used in Address Resolution Protocol of SMBus.

Size: 32 bits**Offset:** 0xdc

Bits	Name	Memory Access	Description
31:0	SMBUS_UDID_WORD0	R/W	<p>This field is used to store the Lower 32 bit value of slave unique device identifier used in Address Resolution Protocol.</p> <p>Value After Reset: 0xffffffff</p>

IC_SMBUS_UDID_WORD1**Name:** SMBUS ARP UDID WORD1 Register**Description:** SMBUS UDID WORD1 Register

This Register can be written only when the i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS_UDID_HC is set to 0. This register is used to store the Middle-Lower 32 bit value of Slave UDID register i.e. UDID[63:32] used in Address Resolution Protocol of SMBus.

Size: 32 bits**Offset:** 0xe0

Bits	Name	Memory Access	Description
31:0	SMBUS_UDID_WORD1	R/W	<p>This field is used to store the Middle-Lower 32 bit value of slave unique device identifier used in Address Resolution Protocol.</p> <p>Value After Reset: 0x0</p>

IC_SMBUS_UDID_WORD2**Name:** SMBUS ARP UDID WORD2 Register**Description:** SMBUS UDID WORD2 Register

This Register can be written only when the i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS_UDID_HC is set to 0. This register is used to store the Middle-Upper 32 bit value of Slave UDID register i.e. UDID[95:64] used in Address Resolution Protocol of SMBus.

Size: 32 bits**Offset:** 0xe4

Bits	Name	Memory Access	Description
31:0	SMBUS_UDID_WORD2	R/W	<p>This field is used to store the Middle-Upper 32 bit value of slave unique device identifier used in Address Resolution Protocol.</p> <p>Value After Reset: 0x0</p>

IC_SMBUS_UDID_WORD3**Name:** SMBUS ARP UDID WORD3 Register**Description:** SMBUS UDID WORD3 Register

This Register can be written only when the i2c is disabled, which corresponds to IC_ENABLE[0] being set to 0. This register is present only if configuration parameter IC_SMBUS_UDID_HC is set to 0. This register is used to store the Upper 32 bit value of Slave UDID register i.e. UDID[127:96] used in Address Resolution Protocol of SMBus.

Size: 32 bits**Offset:** 0xe8

Bits	Name	Memory Access	Description
31:0	SMBUS_UDID_WORD3	R/W	<p>This field is used to store the Upper 32 bit value of slave unique device identifier used in Address Resolution Protocol.</p> <p>Value After Reset: 0x0</p>

REG_TIMEOUT_RST**Name:** Register timeout counter reset value

Description: **Name:** Register timeout counter reset register **Size:** REG_TIMEOUT_WIDTH bits **Address:** 0xF0 **Read/Write Access:** Read/Write This register keeps the timeout value of register timer counter. The reset value of the register is REG_TIMEOUT_VALUE. The default reset value can be further modified if HC_REG_TIMEOUT_VALUE = 0. The final programmed value (or the default reset value if not programmed) determines from what value the register timeout counter starts counting down. A zero on this counter will break the waited transaction with PSLVERR as high.

Size: 32 bits**Offset:** 0xf0

Bits	Name	Memory Access	Description
31:4	RSVD_REG_TIMEOUT_RST	R	Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
3:0	REG_TIMEOUT_RST_rw	R/W	This field holds reset value of REG_TIMEOUT counter register. Value After Reset: 0x8 Volatile: true

IC_COMP_PARAM_1**Name:** Component Parameter Register 1**Description:** Component Parameter Register 1

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

Size: 32 bits**Offset:** 0xf4

Bits	Name	Memory Access	Description
31:24	RSVD_IC_COMP_PARAM_1	R	IC_COMP_PARAM_1 Reserved bits - Read Only Value After Reset: 0x0
23:16	TX_BUFFER_DEPTH	R	The value of this register is derived from the IC_TX_BUFFER_DEPTH coreConsultant parameter. 0x00 = Reserved

Bits	Name	Memory Access	Description
			<p>0x01 = 2 0x02 = 3 ... 0xFF = 256 Value After Reset: 0x1f</p>
15:8	RX_BUFFER_DEPTH	R	<p>The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter.</p> <p>0x00: Reserved 0x01: 2 0x02: 3 ... 0xFF: 256 Value After Reset: 0x1f</p>
7	ADD_ENCODED_PARAMS	R	<p>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.</p> <p>Values:</p> <p>0x1 (ENABLED): Enables capability of reading encoded parameters 0x0 (DISABLED): Disables capability of reading encoded parameters Value After Reset: 0x1</p>
6	HAS_DMA	R	<p>The value of this register is derived from the IC_HAS_DMA coreConsultant parameter.</p> <p>Values:</p> <p>0x1 (ENABLED): DMA handshaking signals are enabled 0x0 (DISABLED): DMA handshaking signals are disabled Value After Reset: 0x1</p>
5	INTR_IO	R	<p>The value of this register is derived from the IC_INTR_IO coreConsultant parameter.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x1 (COMBINED): COMBINED Interrupt outputs 0x0 (INDIVIDUAL): INDIVIDUAL Interrupt outputs Value After Reset: 0x1</p>
4	HC_COUNT_VALUES	R	<p>The value of this register is derived from the IC_HC_COUNT_VALUES coreConsultant parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ENABLED): Hard code the count values for each mode. 0x0 (DISABLED): Programmable count values for each mode. <p>Value After Reset: 0x0</p>
3:2	MAX_SPEED_MODE	R	<p>The value of this register is derived from the IC_MAX_SPEED_MODE coreConsultant parameter.</p> <p>0x0: Reserved 0x1: Standard 0x2: Fast 0x3: High</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (STANDARD): IC MAX SPEED is STANDARD MODE 0x2 (FAST): IC MAX SPEED is FAST MODE 0x3 (HIGH): IC MAX SPEED is HIGH MODE <p>Value After Reset: 0x2</p>
1:0	APB_DATA_WIDTH	R	<p>The value of this register is derived from the APB_DATA_WIDTH coreConsultant parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (APB_08BITS): APB data bus width is 08 bits 0x1 (APB_16BITS): APB data bus width is 16 bits 0x2 (APB_32BITS): APB data bus width is 32 bits 0x3 (RESERVED): Reserved bits <p>Value After Reset: 0x2</p>

IC_COMP_VERSION

Name: I2C Component Version Register

Description: I2C Component Version Register

Size: 32 bits

Offset: 0xf8

Bits	Name	Memory Access	Description
31:0	IC_COMP_VERSION	R	Specific values for this register are described in the Releases Table in the i2c Release Notes Value After Reset: 0x3230332a

IC_COMP_TYPE

Name: I2C Component Type Register

Description: I2C Component Type Register

Size: 32 bits

Offset: 0xfc

Bits	Name	Memory Access	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. Value After Reset: 0x44570140

12.2.5 Programming Example

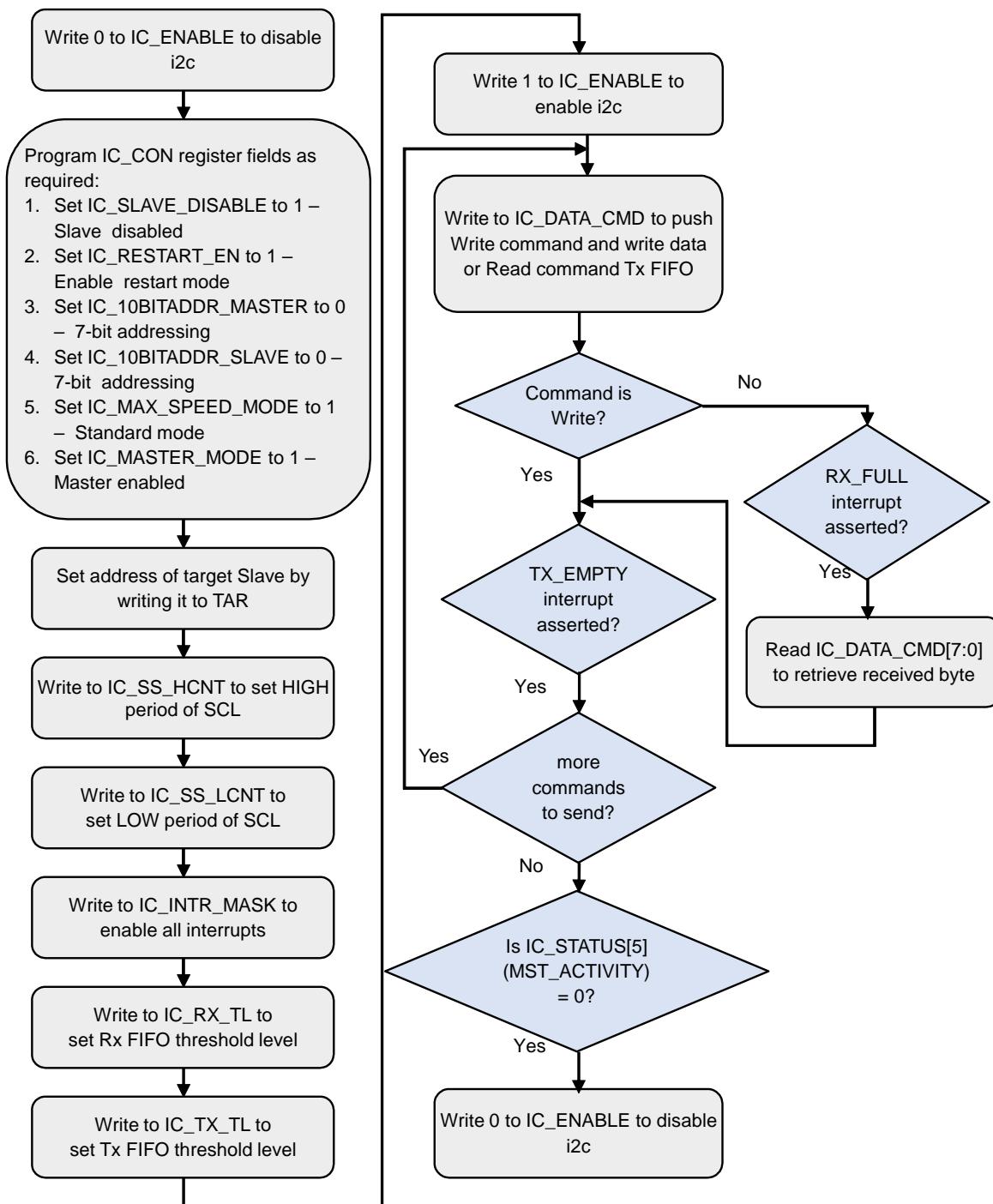


Figure 12-7 I2C MASTER programming example

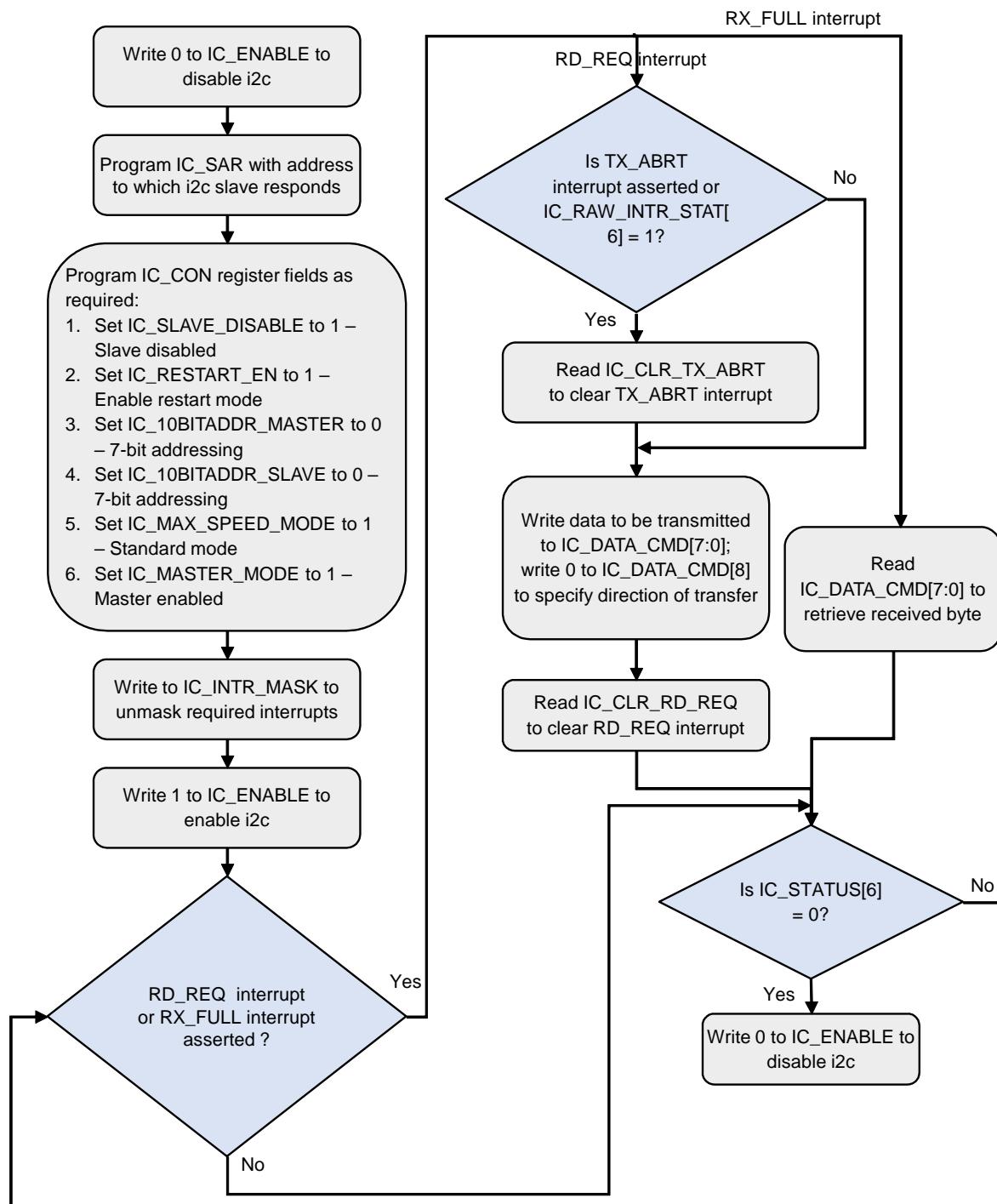


Figure 12-8 I2C slave programming example

12.2.6 Timing Sequence

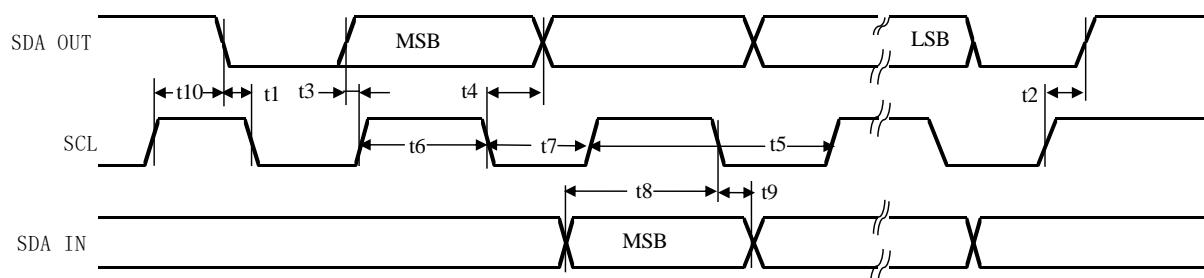


Figure 12-9 I2C timing sequence

12.3 I2S

12.3.1 Overview

The I2S bus (Inter-IC sound bus) is a serial link used for the transmission of digital audio data between devices in a system. Commonly used I2S bus devices include ADC, DAC, DSP, CPU, etc. With the I2S interface, we can connect audio devices and embedded SoC platforms together to provide an audio interface solution for the system.

The following features are supported:

- APB2.0 bus, 32-bit data bit width
- I2S master mode
- Supports I2S transmitter and receiver
- Dual-channel audio transmission is supported
- Audio data resolution 12, 14, 16, 20, 24, 32 bits can be matched
- Support for SCLK gating
- Audio data transmission FIFO depth 16
- DMA transmission is supported

12.3.2 Block Diagram

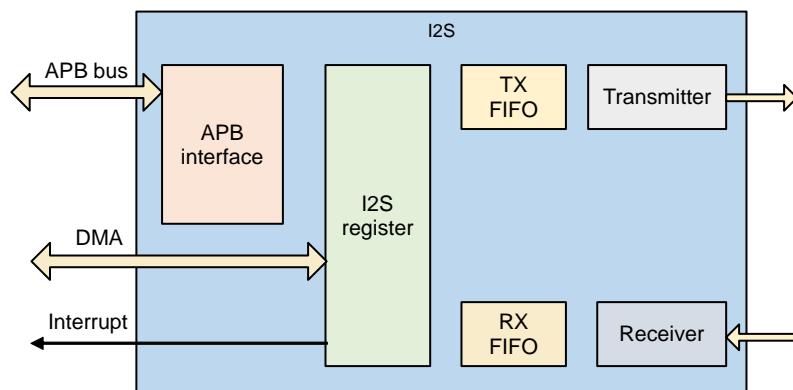


Figure 12-10 I2S block diagram

12.3.3 Function Description

12.3.3.1 Enable I2S

Before receiving or sending any data to the fifo, the IEN bit of the I2S Enable Register must be set to 1 - Enable I2s. Set "IER[0]" to "0" to disable "i2s". When disabled, the following events occur:

- TX and RX FIFOs are cleared, and read/write pointers are reset
- Any data in the process of being transmitted or received is lost
- All other programmable enables (such as transmitter/receiver block enables and individual TX/RX channel enables) in the component are overridden
- Generation of master mode clock signals sclk_en, ws_out and sclk_gate are disabled (for instance, they are held low)

When I2S is enabled and configured as the master device, the device always starts at the left stereo data cycle (WS = 0) and then one SCLK cycle transitions to the right stereo data cycle (WS = 1).

12.3.3.2 Write Send Channel

The stereo data pairs to be transmitted by a TX channel are written to the TX FIFOs through the Left Transmit Holding Register (LTHRx, where x is the channel number) and the Right Transmit Holding Register (RTHRx, where x is the channel number). All stereo data pairs must be written using the

following two-stage process:

1. Write left stereo data to LTHR_x
2. Write right stereo data to RTHR_x

You must write stereo data to the device in this order, otherwise, the interrupt and status lines values are invalid, and the left/right stereo pairs might be transmitted out of sync.

When TX DMA (I2S_TX_DMA = 1) is enabled, data for the TX channel is written to the TX fifo through the TXDMA registers instead of through LTHR_x and RTHR_x.

12.3.3.3 Reading From A Receive Channel

The stereo data pairs received by a RX channel are written to the left and right RX FIFOs. These FIFOs can be read through the Left Receive Buffer Register (LRBR_x, where x is the channel number) and the Right Receive Buffer Register (RRBR_x, where x is the channel number). All stereo data pairs must be read using the following two-stage process:

1. Read the left stereo data from LRBR_x.
2. Read the right stereo data from RRBR_x.

When RX DMA (I2S_RX_DMA=1) is enabled, data can be read from the RX fifo via the RXDMA registers instead of through LRBR_x and RRBR_x.

12.3.3.4 Clock Generation

The Clock Generation Enable (CLKEN) bit of the Clock Enable Register (CER) enables and disables the master mode clock signals: ws_out, sclk_en, and sclk_gate. To enable these signals, set CER[0] to 1; to disable them, set this bit to 0, in which case ws_out is held low (ws_out = 0).

When the CLKEN bit is disabled, any incoming or outgoing data is lost. However, data already in the RX and TX FIFOs are preserved. After this bit is enabled, transmission recommences at the start of the next stereo frame.

12.3.4 Register Description

12.3.4.1 Register Overview

Register	Offset	Description
IER	0x0	i2s Enable Register
IRER	0x4	I2S Receiver Block Enable Register
ITER	0x8	I2S Transmitter Block Enable Register
CER	0xc	Clock Enable Register
CCR	0x10	Clock Configuration Register
RXFFR	0x14	Receiver Block FIFO Reset Register
TXFFR	0x18	Transmitter Block FIFO Reset Register
SR	0x1c	Status Register
LRBR _x	0x020 + 0x40*x	Left Receive Buffer Register x
LTHR _x	0x020 + 0x40*x	Left Transmit Holding Register x
RRBR _x	0x024 + 0x40*x	Right Transmit Holding Register x
RTHR _x	0x024 + 0x40*x	This specifies the Right Transmit Holding Register.
RER _x	0x028 + 0x40*x	Receive Enable Register x

Register	Offset	Description
TERx	0x02C + 0x40*x	Transmit Enable Register x
RCRx	0x030 + 0x40*x	Receive Configuration Register x
TCRx	0x034 + 0x40*x	Transmit Configuration Register x
ISRx	0x038 + 0x40*x	Interrupt status Register x
IMRx	0x03C + 0x40*x	Interrupt Mask Register x
RORx	0x040 + 0x40*x	Receive Overrun Register x
TORx	0x044 + 0x40*x	Transmit Overrun Register x
RFCRx	0x048 + 0x40*x	Receive FIFO Configuration Register x
TFCRx	0x04C + 0x40*x	Transmit FIFO Configuration Register x
RFFx	0x050 + 0x40*x	Receive FIFO Flush Register x
TFFx	0x054 + 0x40*x	Transmit FIFO Flush Register x
RXDMA	0x1c0	Receiver Block DMA Register.
RRXDMA	0x1c4	Reset Receiver Block DMA Register.
TXDMA	0x1c8	Transmitter Block DMA Register
RTXDMA	0x1cc	Reset Transmitter Block DMA Register
I2S_COMP_PARAM_2	0x1f0	Component Parameter Register 2
I2S_COMP_PARAM_1	0x1f4	Component Parameter Register 1
I2S_COMP_VERSION	0x1f8	I2S Component Version Register
I2S_COMP_TYPE	0x1fc	I2S Component Type Register
DMACR	0x200	DMA Control Register
RXDMA_CHx	0x204 + 0x4*x	Receiver Block DMA Register
TXDMA_CHx	0x214 + 0x4*x	Receiver Block DMA Register

12.3.4.2 Register Detail Description

Note: I2S_RX_CHANNELS = 1

IER

Name: i2s Enable Register

Description: This register acts as a global enable/disable for i2s.

Size: 32 bits

Offset: 0x0

Bits	Name	Memory Access	Description
31:12	RSVD_IER	R	RSVD_IER Reserved bits - Read Only Value After Reset: 0x0

11:8	TDM_SLOTS	R	RSVD
7:6	RSVD_6to7	R	RSVD_6to7 Reserved bits - Read Only Value After Reset: 0x0
5	FRAME_OFF	R/W	Frame Offset register. This register specifies the TDM Interface Frame Offset. The i2s must be disabled prior to any changes in this field (that is, IER[0] = 0). Values: 0x0 (SCLK_0): 0 sclk cycle 0x1 (SCLK_1): 1 sclk cycle Value After Reset: 0x0
4:2	RSVD_2to4	R	RSVD_2to4 Reserved bits - Read Only Value After Reset: 0x0
1	INTF_TYPE	R	Reserved bits - Read Only
0	IEN	R/W	i2s enable. This bit enables or disables i2s. A disable on this bit overrides any other block or channel enables and flushes all FIFOs. For more information about how this register affects the other i2s blocks, refer to "i2s Enable". Values: 0x0 (DISABLED): i2s disabled. 0x1 (ENABLED): i2s enabled Value After Reset: 0x0

IRER

Name: I2S Receiver Block Enable Register

Description: This register acts as an enable/disable for the i2s Receiver block.

Size: 32 bits

Offset: 0x4

Bits	Name	Memory Access	Description
31:1	RSVD_IRER	R	RSVD_IRER Reserved bits - Read Only Value After Reset: 0x0
0	RXEN	R/W	Receiver block enable. This bit enables or disables the receiver. A disable on this

Bits	Name	Memory Access	Description
			<p>bit overrides any individual receive channel enables. For more information about the receiver block, refer to "i2s as Receiver".</p> <p>Note: When the i2s is configured with TDM Interface support (I2S_HAS_TDM = 1), the number of channels are always fixed to one and it is always enabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Receiver disabled 0x1 (ENABLED): Receiver enabled <p>Value After Reset: 0x0</p>

ITER

Name: I2S Transmitter Block Enable Register

Description: This register acts as an enable/disable for the i2s Transmitter block.

Size: 32 bits

Offset: 0x8

Bits	Name	Memory Access	Description
31:1	RSVD_ITER	R	<p>RSVD_ITER Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
0	TXEN	R/W	<p>Transmitter block enable.</p> <p>This bit enables or disables the transmitter. A disable on this bit overrides any individual transmit channel enables. For more information about the transmitter block, refer to "i2s as Transmitter".</p> <p>Note: When the i2s is configured with TDM Interface support (I2S_HAS_TDM = 1), the number of channels are always fixed to one and it is always enabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Transmitter disabled 0x1 (ENABLED): Transmitter enabled <p>Value After Reset: 0x0</p>

CER

Name: Clock Enable Register

Description: This register acts as an enable/disable for the i2s Clock Generation block,

which is only relevant in master mode (I2S_MODE_EN = 1). When this block is enabled, the clock signals sclk_en, ws_out, and sclk_gate appear on the interface.

Size: 32 bits**Offset:** 0xc

Bits	Name	Memory Access	Description
31:1	RSVD_CER	R	<p>RSVD_CER Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
0	CLKEN	R/W	<p>Clock generation enable/disable.</p> <p>This bit enables/disables the clock generation signals when i2s is a master: sclk_en, ws_out, and sclk_gate. For more information about clock generation, refer to "Clock Generation (Master Mode)".</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Clock generation disabled 0x1 (ENABLED): Clock generation enabled <p>Value After Reset: 0x0</p>

CCR

Name: Clock Configuration Register**Description:** This register configures the ws_out and sclk_gate signals when i2s is a master.**Size:** 32 bits**Offset:** 0x10**I2S_WS_LENGTH:** 0x2**I2S_SCLK_GATE :** 0x0

Bits	Name	Memory Access	Description
31:5	RSVD_CCR	R	<p>RSVD_CCR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
4:3	WSS	R/W	<p>These bits are used to program the number of sclk cycles for which the word select line (ws_out) stays in the left or right sample mode. The I2S Clock Generation block must be disabled (CER[0] = 0) prior to any changes in this value.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (CLOCK_CYCLES_16): 16 sclk cycles 0x1 (CLOCK_CYCLES_24): 24 sclk cycles 0x2 (CLOCK_CYCLES_32): 32 sclk cycles <p>Value After Reset: I2S_WS_LENGTH</p>

Bits	Name	Memory Access	Description
2:0	SCLKG	R/W	<p>These bits are used to program the gating of sclk. The programmed gating value must be less than or equal to the largest configured/programmed audio resolution to prevent the truncating of RX/TX data. The I2S Clock Generation block must be disabled (CER[0] = 0) before making any changes in this value.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_CLOCK_GATING): Clock gating is disabled 0x1 (CLOCK_CYCLES_12): Gating after 12 sclk cycles 0x2 (CLOCK_CYCLES_16): Gating after 16 sclk cycles 0x3 (CLOCK_CYCLES_20): Gating after 20 sclk cycles 0x4 (CLOCK_CYCLES_24): Gating after 24 sclk cycles <p>Value After Reset: I2S_SCLK_GATE</p>

RXFFR

Name: Receiver Block FIFO Reset Register

Description: This register specifies the Receiver Block FIFO Reset Register.

Size: 32 bits

Offset: 0x14

Bits	Name	Memory Access	Description
31:1	RSVD_RXFFR	W	<p>RSVD_RXFFR Reserved bits - Write Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	RXFFR	W	<p>Receiver FIFO Reset.</p> <p>Writing a 1 to this register flushes all the RX FIFOs (this is a self clearing bit). The Receiver Block must be disabled before writing to this bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_FLUSH): Does not flush the RX FIFO 0x1 (FLUSH): Flushes the RX FIFO <p>Value After Reset: 0x0</p>

			Volatile: true
--	--	--	-----------------------

TXFFR**Name:** Transmitter Block FIFO Reset Register**Description:** This register specifies the Transmitter Block FIFO Reset Register.**Size:** 32 bits**Offset:** 0x18

Bits	Name	Memory Access	Description
31:1	RSVD_TXFFR	W	<p>RSVD_TXFFR Reserved bits - Write Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	TXFFR	W	<p>Transmitter FIFO Reset.</p> <p>Writing a 1 to this register flushes all the TX FIFOs (this is a self clearing bit). The Transmitter Block must be disabled prior to writing this bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_FLUSH): Does not flush the TX FIFO 0x1 (FLUSH): Flushes the TX FIFO <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

SR**Name:** Status Register**Description:** This register specifies the Transmit FIFO empty status.**Size:** 32 bits**Offset:** 0x1c

Bits	Name	Memory Access	Description
31:1	RSVD_SR	R	<p>RSVD_SR Reserved bits - Write Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	TFE	R	<p>Transmit FIFO Empty.</p> <p>When Left and Right transmit FIFO are completely empty, this bit is set. When the Left or Right transmit FIFO contains one or more valid entries, this bit is cleared.</p> <p>Note: When the i2s is transmitting the last I2S frame, this bit is set</p>

Bits	Name	Memory Access	Description
			<p>along with the right data transmission in that frame - as the data is popped with it. Hence, you must wait for one word select length (to ensure the right data is transmitted), to disable the interface.</p> <p>This bit field does not request an interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NOT_EMPTY): Transmit FIFO is not empty 0x1 (EMPTY): Transmit FIFO is empty <p>Value After Reset: 0x1</p> <p>Volatile: true</p>

LRBRx (for x = 0; x <= I2S_RX_CHANNELS-1)**Name:** Left Receive Buffer Register x**Description:** This specifies the Left Receive Buffer Register.**Size:** 32 bits**Offset:** 0x020 + 0x40*x

Bits	Name	Memory Access	Description
31:y	RSVD_LRBx	R	<p>RSVD_LRBx Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
x:0	LRBRx	R	<p>The left stereo data received serially from the receive channel input (sdix). If the RX FIFO is full and the two-stage read operation (for instance, a read from LRBRx followed by a read from RRBRx) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (data already in the RX FIFO is preserved.)</p> <p>Note: Before reading this register again, the right stereo data must be read from RRBRx or the status/interrupts will not be valid.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

LTHR_x (for x = 0; x <= I2S_RX_CHANNELS-1)**Name:** Left Transmit Holding Register x**Description:** This specifies the Left Transmit Holding Register.**Size:** 32 bits**Offset:** 0x020 + 0x40*x

I2S_RX_WORDSIZE_0: 32

Bits	Name	Memory Access	Description
31:y	RSVD_RRBRx	R	<p>RSVD_RRBRx Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[y]: I2S_RX_WORDSIZE_0</p>
x:0	RRBRx	R	<p>The right stereo data received serially from the receive channel input (sdix) is read through this register. If the RX FIFO is full and the two-stage read operation (for instance, read from LRBRx followed by a read from RRBRx) is not performed before the start of the next stereo pair, then the new data is lost and an overrun interrupt occurs. (Data already in the RX FIFO is preserved.)</p> <p>Note: Prior to reading this register, the left stereo data MUST be read from LRBRx, or the status/interrupts will not be valid.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[x]: I2S_RX_WORDSIZE_0 - 1</p>

RTHRx (for x = 0; x <= I2S_RX_CHANNELS-1)

Description: This specifies the Right Transmit Holding Register.

Size: 32 bits

Offset: 0x024 + 0x40*x

I2S_TX_WORDSIZE_0: 32

Bits	Name	Memory Access	Description
31:y	RSVD_RTHRx	W	<p>RSVD_RTHRx Reserved bits - Write Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[y]: I2S_TX_WORDSIZE_0</p>
x:0	RTHRx	W	<p>The right stereo data to be transmitted serially through the transmit channel output (sdox) is written through this register.</p> <p>Writing is a two-stage process:</p>

Bits	Name	Memory Access	Description
			<p>1. A left stereo sample MUST be written to the LTHR_x register.</p> <p>2. A write to this register passes the right stereo sample to the transmitter.</p> <p>Data should only be written to the FIFO when it is not full. Any attempt to write to a full FIFO results in that data being lost and an overrun interrupt being generated.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[x]: I2S_TX_WORDSIZE_0 - 1</p>

RER_x (for x = 0; x <= I2S_RX_CHANNELS-1)**Name:** Receive Enable Register x**Description:** This specifies the Receive Enable Register.

Note: When the i2s is configured with TDM Interface support (I2S_HAS_TDM = 1), the number of channels are always fixed to one and it is always enabled. Hence, with TDM Interface configuration, the receiver block must be disabled prior to any changes in this register value (that is, IRER[0] = 0).

Size: 32 bits**Offset:** 0x028 + 0x40*x

Bits	Name	Memory Access	Description
31:1	RSVD_RER _x	R	<p>RSVD_RER_x Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
0	RXCHE _N x	R/W	<p>Receive channel enable. This bit enables/disables a receive channel, independently of all other channels.</p> <p>On enable, the channel begins receiving on the next left stereo cycle.</p> <p>A global disable of i2s (IER[0] = 0) or the Receiver block (IRER[0] = 0) overrides this value.</p> <p>When the i2s is configured with TDM Interface support (I2S_HAS_TDM = 1), the number of channels are always fixed to one and it is always enabled.</p> <p>Values:</p> <p>0x0 (DISABLED): Receive Channel Disable</p>

Bits	Name	Memory Access	Description
			0x1 (ENABLED): Receive Channel Enable Value After Reset: 0x1

TERx (for x = 0; x <= I2S_TX_CHANNELS-1)**Name:** Transmit Enable Register x**Description:** This specifies the Transmit Enable Register.**Size:** 32 bits**Offset:** 0x02C + 0x40*x

Bits	Name	Memory Access	Description
31:1	RSVD_TERx	R	RSVD_TERx Reserved bits - Read Only Value After Reset: 0x0
0	TXCHENx	RW	Transmit channel enable. This bit enables/disables a transmit channel, independently of all other channels. On enable, the channel begins transmitting on the next left stereo cycle. A global disable of i2s (IER[0] = 0) or Transmitter block (ITER[0] = 0) overrides this value. Values: 0x0 (DISABLED): Transmit Channel Disable 0x1 (ENABLED): Transmit Channel Enable Value After Reset: 0x1

RCRx (for x = 0; x <= I2S_RX_CHANNELS-1)**Name:** Receive Configuration Register x**Description:** This specifies the Receive Configuration Register.**Size:** 32 bits**Offset:** 0x030 + 0x40*x

Bits	Name	Memory Access	Description
31:3	RSVD_RCRx	R	RSVD_RCRx Reserved bits - Read Only Value After Reset: 0x0
2:0	WLEN	R/W	These bits are used to program the desired data resolution of the

Bits	Name	Memory Access	Description
			<p>receiver and enables the LSB of the incoming left (or right) word to be placed in the LSB of the LRBRx (or RRB Rx) register.</p> <p>Programmed data resolution must be less than or equal to I2S_RX_WORDSIZE_x. If the selected resolution is greater than the I2S_RX_WORDSIZE_x, the receive channel defaults back to I2S_RX_WORDSIZE_RESET_VALUE_x.</p> <p>The channel must be disabled prior to any changes in this value (RERO[0] = 0).</p> <p>When the i2s is configured with TDM Interface support (I2S_HAS_TDM = 1), the number of channels are always fixed to one and it is always enabled. Hence, with TDM Interface configuration, the receiver block must be disabled prior to any changes in this field's value (that is, IRER[0] = 0).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (IGNORE_WORD_LENGTH): Ignore the word length 0x1 (RESOLUTION_12_BIT): 12-bit data resolution of the receiver. 0x2 (RESOLUTION_16_BIT): 16-bit data resolution of the receiver. 0x3 (RESOLUTION_20_BIT): 20-bit data resolution of the receiver. 0x4 (RESOLUTION_24_BIT): 24-bit data resolution of the receiver. 0x5 (RESOLUTION_32_BIT): 32-bit data resolution of the receiver. <p>Value After Reset: 0x5</p>

TCRx (for x = 0; x <= I2S_TX_CHANNELS-1)**Name:** Transmit Configuration Register x**Description:** This specifies the Transmit Configuration Register.**Size:** 32 bits**Offset:** 0x034 + 0x40*x

Bits	Name	Memory Access	Description
31:3	RSVD_TCRx	R	<p>RSVD_TCRx Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
2:0	WLEN	R/W	<p>These bits are used to program the data resolution of the transmitter and ensures the MSB of the data is transmitted first.</p> <p>Programmed resolution must be less than or equal to I2S_TX_WORDSIZE_x. If the selected resolution is greater than I2S_TX_WORDSIZE_x, the transmit channel defaults back to I2S_TX_WORDSIZE_RESET_VALUE_x value.</p>

Bits	Name	Memory Access	Description
			<p>The channel must be disabled prior to any changes in this value (TER0[0] = 0).</p> <p>When the i2s is configured with TDM Interface support (I2S_HAS_TDM = 1), the number of channels are always fixed to one and it is always enabled. Hence, with TDM Interface configuration, the transmitter block must be disabled prior to any changes in this field's value (that is, ITER[0] = 0).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (IGNORE_WORD_LENGTH): Ignore the word length 0x1 (RESOLUTION_12_BIT): 12-bit data resolution of the transmitter. 0x2 (RESOLUTION_16_BIT): 16-bit data resolution of the transmitter. 0x3 (RESOLUTION_20_BIT): 20-bit data resolution of the transmitter. 0x4 (RESOLUTION_24_BIT): 24-bit data resolution of the transmitter. 0x5 (RESOLUTION_32_BIT): 32-bit data resolution of the transmitter. <p>Value After Reset: 0x5</p>

ISR_x (for x = 0; x <= I2S_TX_CHANNELS-1)**Name:** Interrupt status Register x**Description:** This specifies the Interrupt Status Register.**Size:** 32 bits**Offset:** 0x038 + 0x40*x

Bits	Name	Memory Access	Description
31:6	RSVD31_6	R	<p>RSVD31_6 Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
5	TXFO	R	<p>Status of Data Overrun interrupt for the TX channel.</p> <p>This bit specifies whether the TX FIFO write is valid or an overrun. Attempt to write to full TX FIFO.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (WRITE_VALID): TX FIFO write valid 0x1 (WRITE_OVERRUN): TX FIFO write overrun Value After Reset: 0x0 Volatile: true</p>
4	TXFE	R	<p>Status of Transmit Empty Trigger interrupt. This bit specifies whether the TX FIFO trigger level has reached or not. TX FIFO is empty. Values: 0x0 (REACHED_TRIGGER_LEVEL): TX FIFO trigger level is reached 0x1 (NOT_REACHED): TX FIFO trigger level is not reached Value After Reset: 0x1 Volatile: true</p>
3:2	RSVD3_2	R	<p>RSVD3_2 Reserved bits - Read Only Value After Reset: 0x0 Volatile: true</p>
1	RXFO	R	<p>Status of Data Overrun interrupt for the RX channel. Incoming data lost due to a full RX FIFO. Values: 0x0 (WRITE_VALID): RX FIFO write valid 0x1 (WRITE_OVERRUN): RX FIFO write overrun Value After Reset: 0x0 Volatile: true</p>
0	RXDA	R	<p>Status of Receive Data Available interrupt. This bit denotes the status of the RX FIFO trigger level. Values: 0x1 (REACHED_TRIGGER_LEVEL): RX FIFO trigger level is reached 0x0 (NOT_REACHED): RX FIFO trigger level is not reached Value After Reset: 0x0 Volatile: true</p>

IMRx (for x = 0; x <= I2S_TX_CHANNELS-1)**Name:** Interrupt Mask Register x**Description:** This specifies the Interrupt Mask Register.**Size:** 32 bits**Offset:** 0x03C + 0x40*x

Bits	Name	Memory Access	Description
31:6	RSVD_IMR0_6_31	R	RSVD_IMR0_6_31 Reserved bits - Read Only Value After Reset: 0x0
5	TXFOM	R/W	Mask TX FIFO Overrun interrupt. This bit masks or unmasks a TX FIFO overrun interrupt. Values: 0x1 (MASK_INTERRUPT): Masks TX FIFO Overrun interrupt 0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Overrun interrupt Value After Reset: 0x1
4	TXFEM	R/W	Mask TX FIFO Empty interrupt. This bit masks or unmasks a TX FIFO Empty interrupt. Values: 0x1 (MASK_INTERRUPT): Masks TX FIFO Empty interrupt 0x0 (UNMASK_INTERRUPT): Unmasks TX FIFO Empty interrupt Value After Reset: 0x1
3:2	RSVD_IMR0_2_3	R	RSVD_IMR0_2_3 Reserved bits - Read Only Value After Reset: 0x0
1	RXFOM	R/W	Mask RX FIFO Overrun interrupt. This bit masks or unmasks an RX FIFO Overrun interrupt. Values: 0x1 (MASK_INTERRUPT): Masks RX FIFO Overrun interrupt 0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO Overrun interrupt Value After Reset: 0x1
0	RXDAM	R/W	Mask RX FIFO Data Available interrupt.

Bits	Name	Memory Access	Description
			<p>This bit masks or unmasks an RX FIFO Data Available interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (MASK_INTERRUPT): Masks RX FIFO data available interrupt 0x0 (UNMASK_INTERRUPT): Unmasks RX FIFO data available interrupt <p>Value After Reset: 0x1</p>

RORx (for x = 0; x <= I2S_RX_CHANNELS-1)**Name:** Receive Overrun Register x**Description:** This specifies the Receive Overrun Register.**Size:** 32 bits**Offset:** 0x040 + 0x40*x

Bits	Name	Memory Access	Description
31:1	RSVD_RORx	R	<p>RSVD_RORx Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	RXCHO	R	<p>Read this bit to clear the RX FIFO Data Overrun interrupt. Values:</p> <ul style="list-style-type: none"> 0x0 (WRITE_VALID): RX FIFO write valid 0x1 (WRITE_OVERRUN): RX FIFO write overrun <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

TORx (for x = 0; x <= I2S_TX_CHANNELS-1)**Name:** Transmit Overrun Register x**Description:** This specifies the Transmit Overrun Register.**Size:** 32 bits**Offset:** 0x044 + 0x40*x

Bits	Name	Memory Access	Description
31:1	RSVD_TORx	R	<p>RSVD_TORx Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>

			Volatile: true
0	TXCHO	R	<p>Read this bit to clear the TX FIFO Data Overrun interrupt.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (WRITE_VALID): TX FIFO write valid 0x1 (WRITE_OVERRUN): TX FIFO write overrun <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

RFCRx (for x = 0; x <= I2S_RX_CHANNELS-1)**Name:** Receive FIFO Configuration Register x**Description:** This specifies the Receive FIFO Configuration Register.**Size:** 32 bits**Offset:** 0x048 + 0x40*x

Bits	Name	Memory Access	Description
31:4	RSVD_RFCRx	R	<p>RSVD_RFCRx Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
3:0	RXCHDT	R/W	<p>These bits program the trigger level in the RX FIFO at which the Received Data Available interrupt and DMA request is generated.</p> <p>Trigger Level = Programmed Value + 1 (for example, 1 to I2S_RX_FIFO_DEPTH_0)</p> <p>Valid RXCHDT values: 0 to (I2S_RX_FIFO_0 - 1)</p> <p>If an illegal value is programmed, these bits saturate to (I2S_RX_FIFO_0 - 1).</p> <p>The channel must be disabled prior to any changes in this value (that is, RERx[0] = 0).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (TRIGGER_LEVEL_1): Interrupt trigger and DMA request asserted when FIFO level is 1. 0x1 (TRIGGER_LEVEL_2): Interrupt trigger and DMA request asserted when FIFO level is 2. 0x2 (TRIGGER_LEVEL_3): Interrupt trigger and DMA request asserted when FIFO level is 3. 0x3 (TRIGGER_LEVEL_4): Interrupt trigger and DMA request asserted when FIFO level is 4.

		<p>0x4 (TRIGGER_LEVEL_5): Interrupt trigger and DMA request asserted when FIFO level is 5.</p> <p>0x5 (TRIGGER_LEVEL_6): Interrupt trigger and DMA request asserted when FIFO level is 6.</p> <p>0x6 (TRIGGER_LEVEL_7): Interrupt trigger and DMA request asserted when FIFO level is 7.</p> <p>0x7 (TRIGGER_LEVEL_8): Interrupt trigger and DMA request asserted when FIFO level is 8.</p> <p>0x8 (TRIGGER_LEVEL_9): Interrupt trigger and DMA request asserted when FIFO level is 9.</p> <p>0x9 (TRIGGER_LEVEL_10): Interrupt trigger and DMA request asserted when FIFO level is 10.</p> <p>0xa (TRIGGER_LEVEL_11): Interrupt trigger and DMA request asserted when FIFO level is 11.</p> <p>0xb (TRIGGER_LEVEL_12): Interrupt trigger and DMA request asserted when FIFO level is 12.</p> <p>0xc (TRIGGER_LEVEL_13): Interrupt trigger and DMA request asserted when FIFO level is 13.</p> <p>0xd (TRIGGER_LEVEL_14): Interrupt trigger and DMA request asserted when FIFO level is 14.</p> <p>0xe (TRIGGER_LEVEL_15): Interrupt trigger and DMA request asserted when FIFO level is 15.</p> <p>0xf (TRIGGER_LEVEL_16): Interrupt trigger and DMA request asserted when FIFO level is 16.</p> <p>Value After Reset: 0x3</p>
--	--	--

TFCRx (for x = 0; x <= I2S_TX_CHANNELS-1)**Name:** Transmit FIFO Configuration Register x**Description:** This specifies the Transmit FIFO Configuration Register.**Size:** 32 bits**Offset:** 0x04C + 0x40*x

Bits	Name	Memory Access	Description
31:4	RSVD_TFCRx	R	<p>RSVD_TFCRx Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
3:0	TXCHET	R/W	<p>These bits program the trigger level in the TX FIFO at which the Empty Threshold Reached Interrupt and DMA request is generated.</p> <p>Trigger Level = TXCHET</p> <p>TXCHET values: 0 to (I2S_TX_FIFO_0 - 1)</p>

Bits	Name	Memory Access	Description
			<p>If an illegal value is programmed, these bits saturate to (I2S_TX_FIFO_0 - 1). The channel must be disabled prior to any changes in this value (that is, TER0[0] = 0).</p> <p>When the i2s is configured with TDM Interface support (I2S_HAS_TDM = 1), the number of channels are always fixed to one and it is always enabled. Hence, with TDM Interface configuration, the transmitter block must be disabled prior to any changes in this field's value (that is, ITER[0] = 0).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (TRIGGER_LEVEL_1): Interrupt trigger and DMA request asserted when FIFO level is 1. 0x1 (TRIGGER_LEVEL_2): Interrupt trigger and DMA request asserted when FIFO level is 2. 0x2 (TRIGGER_LEVEL_3): Interrupt trigger and DMA request asserted when FIFO level is 3. 0x3 (TRIGGER_LEVEL_4): Interrupt trigger and DMA request asserted when FIFO level is 4. 0x4 (TRIGGER_LEVEL_5): Interrupt trigger and DMA request asserted when FIFO level is 5. 0x5 (TRIGGER_LEVEL_6): Interrupt trigger and DMA request asserted when FIFO level is 6. 0x6 (TRIGGER_LEVEL_7): Interrupt trigger and DMA request asserted when FIFO level is 7. 0x7 (TRIGGER_LEVEL_8): Interrupt trigger and DMA request asserted when FIFO level is 8. 0x8 (TRIGGER_LEVEL_9): Interrupt trigger and DMA request asserted when FIFO level is 9. 0x9 (TRIGGER_LEVEL_10): Interrupt trigger and DMA request asserted when FIFO level is 10. 0xa (TRIGGER_LEVEL_11): Interrupt trigger and DMA request asserted when FIFO level is 11. 0xb (TRIGGER_LEVEL_12): Interrupt trigger and DMA request asserted when FIFO level is 12. 0xc (TRIGGER_LEVEL_13): Interrupt trigger and DMA request asserted when FIFO level is 13. 0xd (TRIGGER_LEVEL_14): Interrupt trigger and DMA request asserted when FIFO level is 14. 0xe (TRIGGER_LEVEL_15): Interrupt trigger and DMA request asserted when FIFO level is 15.

Bits	Name	Memory Access	Description
			0xf (TRIGGER_LEVEL_16): Interrupt trigger and DMA request asserted when FIFO level is 16. Value After Reset: 0x3

RFFx (for x = 0; x <= I2S_RX_CHANNELS-1)**Name:** Receive FIFO Flush Register x**Description:** This specifies the Receive FIFO Flush Register.**Size:** 32 bits**Offset:** 0x050 + 0x40*x

Bits	Name	Memory Access	Description
31:1	RSVD_RFFx	W	RSVD_RFFx Reserved bits - Write Only Value After Reset: 0x0 Volatile: true
0	RXCHFR	W	Receive Channel FIFO Reset. Writing a 1 to this register flushes an individual RX FIFO (This is a self clearing bit.). A Rx channel or block must be disabled prior to writing to this bit. In TDM mode, writing a 1 to this register flushes all Slot FIFOs. Values: 0x0 (NO_FLUSH): Does not flush an individual RX FIFO 0x1 (FLUSH): Flushes an individual RX FIFO Value After Reset: 0x0 Volatile: true

TFFx (for x = 0; x <= I2S_TX_CHANNELS-1)**Name:** Transmit FIFO Flush Register x**Description:** This specifies the Transmit FIFO Flush Register.**Size:** 32 bits**Offset:** 0x054 + 0x40*x

Bits	Name	Memory Access	Description
31:1	RSVD_TFFx	W	RSVD_TFFx Reserved bits - Write Only Value After Reset: 0x0

			Volatile: true
0	TXCHFR	W	<p>Transmit Channel FIFO Reset.</p> <p>Writing a 1 to this register flushes channel's TX FIFO (This is a self clearing bit.). The TX channel or block must be disabled prior to writing to this bit. In TDM mode, writing a 1 to this register flushes all Slot FIFOs.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_FLUSH): Do not flush TX FIFO of the channel. 0x1 (FLUSH): Flushes TX FIFO of the channel. <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

RXDMA

Name: Receiver Block DMA Register.

Description:

Standard I2S Interface Mode, IER[1] = 0 :

The RXDMA register allows access to all enabled Receive channels via a single point rather than through the LRBRx and RRBRx registers. The Receive channels are targeted in a cyclical fashion (starting at the lowest numbered enabled channel) and takes two reads (left and right stereo data) before the component points to the next channel.

The following example describes the behavior of this register for a component that has been configured with four Receive channels, where Channels 0 and 3 are enabled:

Order of returned read data:

1. Ch0 - Left Data
2. Ch0 - Right Data
3. Ch3 - Left Data
4. Ch3 - Right Data
5. Ch0 - Left Data
6. Ch0 - Right Data, and so on

Size: 32 bits

Offset: 0x1c0

APB_DATA_WIDTH: 32

Bits	Name	Memory Access	Description
31:y	RSVD_RXDMA	R	<p>RSVD_RXDMA Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[y]: APB_DATA_WIDTH</p>
x:0	RXDMA	R	<p>Receiver Block DMA Register.</p> <p>In Standard I2S Interface Mode these bits are used to cycle repeatedly through the enabled receive channels (from lowest numbered to highest), reading stereo data pairs.</p> <p>In TDM Interface Mode, These bits are used to cycle repeatedly read through the enabled receive slots (from lowest numbered to highest).</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[x]: APB_DATA_WIDTH - 1</p>

RRXDMA

Name: Reset Receiver Block DMA Register.

Description: Standard I2S Interface Mode, IER[1] = 0 :

The RXDMA can be reset to the lowest enabled Channel via the RRXDMA register. The RRXDMA register can be written to at any stage of the RXDMA's read cycle, however, it has no effect when the component is in the middle of a stereo pair read. The following example describes the operation of this register for a system with four Receive channels, where channels 0, 1, 2, and 3 are enabled.

Order of returned read data:

1. Ch0 - Left Data
2. Ch0 - Right Data
3. RRXDMA Reset
4. Ch0 - Left Data
5. Ch0 - Right Data
6. Ch1 - Left Data
7. RRXDMA Reset - No effect (read not complete)
8. Ch1 - Right Data, etc.
9. Ch2 - Left Data
10. Ch2 - Right Data
11. RRXDMA Reset

12. Ch0 - Left Data

13. Ch0 - Right Data

Size: 32 bits

Offset: 0x1c4

Bits	Name	Memory Access	Description
31:1	RSVD_RRXDMA	W	<p>RSVD_RRXDMA Reserved bits - Write Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	RRXDMA	W	<p>Reset Receiver Block DMA Register.</p> <p>Writing a 1 to this self-clearing register resets the RXDMA register mid-cycle to point to the lowest enabled Receive channel.</p> <p>Note: Writing to this register has no effect if the component is performing a stereo pair read (such as, when left stereo data has been read but not right stereo data).</p> <p>When the i2s is programmed for TDM support, Writing a 1 to this self-clearing register resets the RXDMA register mid-cycle to point to the lowest enabled Receive slot.</p> <p>Values:</p> <p>0x1 (RESET)</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

TXDMA

Name: Transmitter Block DMA Register

Description: The TXDMA register functions similar to the RXDMA register and allows write accesses to all of the enabled Transmit channels via a single point rather than through the LTHR_x and RTHR_x registers.

Note: There is no write coherency logic, the APB_DATA_WIDTH must be greater than or equal to the largest Transmit channel word size to ensure all half data pairs can be written using a single write.

Channels can be enabled or disabled during the write cycles; however, i2s does not support disabling a channel in the middle of a stereo pair.

Size: 32 bits

Offset: 0x1c8

APB_DATA_WIDTH: 32

Bits	Name	Memory Access	Description
31:y	RSVD_TXDMA	W	<p>RSVD_TXDMA Reserved bits - Write Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[y]: APB_DATA_WIDTH</p>
x:0	TXDMA	W	<p>Transmitter Block DMA Register.</p> <p>The register bits can be used to cycle repeatedly through the enabled Transmit channels (from lowest numbered to highest) to allow writing of stereo data pairs.</p> <p>In TDM Interface mode, These bits are used to cycle repeatedly write through the enabled transmit slots (from lowest numbered to highest).</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[x]: APB_DATA_WIDTH - 1</p>

RTXDMA

Name: Reset Transmitter Block DMA Register

Description: This register provides the same functionality as the RRXDMA register but targets TXDMA instead.

Size: 32 bits

Offset: 0x1cc

Bits	Name	Memory Access	Description
31:1	RSVD_RTXDMA	W	<p>RSVD_RTXDMA Reserved bits - Write Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	RTXDMA	W	<p>Reset Transmitter Block DMA Register.</p> <p>Writing a 1 to this self-clearing register resets the TXDMA register mid-cycle to point to the lowest enabled Transmit channel.</p> <p>Note: This register has no effect in the middle of a stereo pair write (such as, when left stereo data has been written but not right stereo data).</p>

			<p>When the i2s is programmed for TDM support, Writing a 1 to this self-clearing register resets the TXDMA register mid-cycle to point to the lowest enabled transmit slot.</p> <p>Values:</p> <p>0x1 (RESET): Reset Transmitter Block DMA Register</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
--	--	--	--

I2S_COMP_PARAM_2**Name:** Component Parameter Register 2**Description:** This specifies bits for Component Parameter Register 2.**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).**Size:** 32 bits**Offset:** 0x1f0

Bits	Name	Memory Access	Description
31:13	RSVD_31_13	R	<p>RSVD_31_13 Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
12:10	I2S_RX_WORDSIZE_3	R	<p>These bits specify the RX resolution for WORDSIZE_3.</p> <p>Values:</p> <p>0x0 (RESOLUTION_12_BIT): 12-bit Resolution 0x1 (RESOLUTION_16_BIT): 16-bit Resolution 0x2 (RESOLUTION_20_BIT): 20-bit Resolution 0x3 (RESOLUTION_24_BIT): 24-bit Resolution 0x4 (RESOLUTION_32_BIT): 32-bit Resolution</p> <p>Value After Reset: 0x1</p>
9:7	I2S_RX_WORDSIZE_2	R	<p>These bits specify the RX resolution for WORDSIZE_2.</p> <p>Values:</p> <p>0x0 (RESOLUTION_12_BIT): 12-bit Resolution 0x1 (RESOLUTION_16_BIT): 16-bit Resolution 0x2 (RESOLUTION_20_BIT): 20-bit Resolution 0x3 (RESOLUTION_24_BIT): 24-bit Resolution</p>

Bits	Name	Memory Access	Description
			0x4 (RESOLUTION_32_BIT): 32-bit Resolution Value After Reset: 0x1
6	RSVD_I2S_COMP_PARAM_2_6	R	RSVD_I2S_COMP_PARAM_2_6 Reserved bits - Read Only Value After Reset: 0x0
5:3	I2S_RX_WORDSIZE_1	R	These bits specify the RX resolution for WORDSIZE_1. Values: 0x0 (RESOLUTION_12_BIT): 12-bit Resolution 0x1 (RESOLUTION_16_BIT): 16-bit Resolution 0x2 (RESOLUTION_20_BIT): 20-bit Resolution 0x3 (RESOLUTION_24_BIT): 24-bit Resolution 0x4 (RESOLUTION_32_BIT): 32-bit Resolution Value After Reset: 0x1
2:0	I2S_RX_WORDSIZE_0	R	These bits specify the RX resolution for WORDSIZE_0. Values: 0x0 (RESOLUTION_12_BIT): 12-bit Resolution 0x1 (RESOLUTION_16_BIT): 16-bit Resolution 0x2 (RESOLUTION_20_BIT): 20-bit Resolution 0x3 (RESOLUTION_24_BIT): 24-bit Resolution 0x4 (RESOLUTION_32_BIT): 32-bit Resolution Value After Reset: 0x04

2S_COMP_PARAM_1

Name: Component Parameter Register 1

Description: This specifies bits for Component Parameter Register 1.

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).

Size: 32 bits

Offset: 0x1f4

Bits	Name	Memory Access	Description
31:28	RSVD_PARAM_1_28_31	R	<p>RSVD_I2S_COMP_PARAM_1_28_31 Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
27:25	I2S_TX_WORDSIZE_3	R	<p>These bits specify the TX resolution for WORDSIZE_3.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (RESOLUTION_12_BIT): 12-bit Resolution 0x1 (RESOLUTION_16_BIT): 16-bit Resolution 0x2 (RESOLUTION_20_BIT): 20-bit Resolution 0x3 (RESOLUTION_24_BIT): 24-bit Resolution 0x4 (RESOLUTION_32_BIT): 32-bit Resolution <p>Value After Reset: 0x1</p>
24:22	I2S_TX_WORDSIZE_2	R	<p>These bits specify the TX resolution for WORDSIZE_2.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (RESOLUTION_12_BIT): 12-bit Resolution 0x1 (RESOLUTION_16_BIT): 16-bit Resolution 0x2 (RESOLUTION_20_BIT): 20-bit Resolution 0x3 (RESOLUTION_24_BIT): 24-bit Resolution 0x4 (RESOLUTION_32_BIT): 32-bit Resolution <p>Value After Reset: 0x1</p>
21:19	I2S_TX_WORDSIZE_1	R	<p>These bits specify the TX resolution for WORDSIZE_1.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (RESOLUTION_12_BIT): 12-bit Resolution 0x1 (RESOLUTION_16_BIT): 16-bit Resolution 0x2 (RESOLUTION_20_BIT): 20-bit Resolution 0x3 (RESOLUTION_24_BIT): 24-bit Resolution 0x4 (RESOLUTION_32_BIT): 32-bit Resolution <p>Value After Reset: ENCODED_I2S_TX_WORDSIZE_1</p>
18:16	I2S_TX_WORDSIZE_0	R	<p>These bits specify the TX resolution for WORDSIZE_0.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (RESOLUTION_12_BIT): 12-bit Resolution 0x1 (RESOLUTION_16_BIT): 16-bit Resolution

Bits	Name	Memory Access	Description
			<p>0x2 (RESOLUTION_20_BIT): 20-bit Resolution 0x3 (RESOLUTION_24_BIT): 24-bit Resolution 0x4 (RESOLUTION_32_BIT): 32-bit Resolution</p> <p>Value After Reset: ENCODED_I2S_TX_WORDSIZE_0</p>
15:11	RSVD_PARAM_1_11_15	R	<p>RSVD_I2S_COMP_PARAM_1_11_15 Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
10:9	I2S_TX_CHANNELS	R	<p>These bits specify the number of TX channels.</p> <p>Values:</p> <p>0x0 (TX_CHANNEL_1): 1 Transmit Channel 0x1 (TX_CHANNEL_2): 2 Transmit Channels 0x2 (TX_CHANNEL_3): 3 Transmit Channels 0x3 (TX_CHANNEL_4): 4 Transmit Channels</p> <p>Value After Reset: ENCODED_I2S_TX_CHANNELS</p>
8:7	I2S_RX_CHANNELS	R	<p>These bits specify the number of RX channels.</p> <p>Values:</p> <p>0x0 (RX_CHANNEL_1): 1 Receive Channel 0x1 (RX_CHANNEL_2): 2 Receive Channels 0x2 (RX_CHANNEL_3): 3 Receive Channels 0x3 (RX_CHANNEL_4): 4 Receive Channels</p> <p>Value After Reset: ENCODED_I2S_RX_CHANNELS</p>
6	I2S_RECEIVER_BLOCK	R	<p>This bit specifies whether the receiver block is enabled or not.</p> <p>Values:</p> <p>0x0 (FALSE): Receiver block is disabled 0x1 (TRUE): Receiver block is enabled</p> <p>Value After Reset: I2S_RECEIVER_BLOCK</p>
5	I2S_TRANSMITTER_BLOCK	R	<p>This bit specifies whether the transmitter block is enabled or not.</p> <p>Values:</p> <p>0x0 (FALSE): Transmitter block is disabled 0x1 (TRUE): Transmitter block is enabled</p> <p>Value After Reset: I2S_TRANSMITTER_BLOCK</p>
4	I2S_MODE_EN	R	This bit specifies whether the master mode is

Bits	Name	Memory Access	Description
			<p>enabled or not. Values:</p> <p>0x0 (FALSE): Master mode is disabled 0x1 (TRUE): Master mode is enabled</p> <p>Value After Reset: I2S_MODE_EN</p>
3:2	I2S_FIFO_DEPTH_GLOBAL	R	<p>These bits specify the FIFO depth for TX and RX channels.</p> <p>Values:</p> <p>0x0 (FIFO_DEPTH_2): FIFO depth is equals to 2 for TX and RX channels 0x1 (FIFO_DEPTH_4): FIFO depth is equals to 4 for TX and RX channels 0x2 (FIFO_DEPTH_8): FIFO depth is equals to 8 for TX and RX channels 0x3 (FIFO_DEPTH_16): FIFO depth is equals to 16 for TX and RX channels</p> <p>Value After Reset: ENCODED_I2S_FIFO_DEPTH_GLOBAL</p>
1:0	APB_DATA_WIDTH	R	<p>These bits specify the APB data width.</p> <p>Values:</p> <p>0x0 (BITS_8): 8 bits APB data width 0x1 (BITS_16): 16 bits APB data width 0x2 (BITS_32): 32 bits APB data width</p> <p>Value After Reset: ENCODED_APB_DATA_WIDTH</p>

I2S_COMP_VERSION

Name: I2S Component Version Register

Description: This register specifies the I2S Component Version.

Size: 32 bits

Offset: 0x1f8

Bits	Name	Memory Access	Description
31:0	I2S_COMP_VERSION	R	<p>These bits specify the I2S component version.</p> <p>Value After Reset: I2S_COMP_VERSION</p>

I2S_COMP_TYPE**Name:** I2S Component Type Register**Description:** This register specifies the I2S Component Type.**Size:** 32 bits**Offset:** 0x1fc

Bits	Name	Memory Access	Description
31:0	I2S_COMP_TYPE	R	<p>DesignWare Component Type number = 0x445701a0.</p> <p>This unique hexadecimal value is constant and is derived from the two ASCII letters 'DW' followed by a 16-bit unsigned number.</p> <p>Value After Reset: I2S_COMP_TYPE</p>

DMACR**Name:** DMA Control Register

Description: This register is only valid when i2s is configured with a set of DMA Controller interface signals (I2S_HAS_DMA_INTERFACE = 1). When i2s is not configured with DMA handshake interface, this register will not exist and writing to the register's address will have no effect and reading from this register address will return zero. The register is used to enable the DMA Controller interface operation.

Size: 32 bits**Offset:** 0x200

Bits	Name	Memory Access	Description
31:18	RSVD_DMACR	R	<p>DMACR Reserved bits - Read Only.</p> <p>Value After Reset: 0x0</p>
17	DMAEN_TXBLOCK	R/W	<p>DMA Enable for transmit block. The corresponding bits of this field enables/disables the DMA handshake logic for transmitter block.</p> <p>Values:</p> <p>0x0 (DISABLED): DMA disabled for transmit block</p> <p>0x1 (ENABLED): DMA enabled for transmit block</p> <p>Value After Reset: 0x0</p>
16	DMAEN_RXBLOCK	R/W	<p>DMA Enable for receive block. The corresponding bits of this field enables/disables the DMA handshake logic for receiver block.</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
			<p>receiver block</p> <p>Values:</p> <p>0x0 (DISABLED): DMA disabled for receiver block</p> <p>0x1 (ENABLED): DMA enabled for receiver block</p> <p>Value After Reset: 0x0</p>
15:19	RSVD_DMAEN_TXCH	R	<p>Reserved bits for transmit channel DMA Enable - Read Only.</p> <p>Value After Reset: 0x0</p>
8	DMAEN_TXCH_0	R/W	<p>DMA Enable for transmit channel 0. The corresponding bits of this field enables/disables the transmit FIFO DMA for channel 0.</p> <p>Values:</p> <p>0x0 (DISABLED): DMA disabled for transmit channel 0</p> <p>0x1 (ENABLED): DMA enabled for transmit channel 0</p> <p>Value After Reset: 0x0</p>
7:1	RSVD_DMAEN_RXCH	R	<p>Reserved bits for receive channel DMA Enable - Read Only.</p> <p>Value After Reset: 0x0</p>
0	DMAEN_RXCH_0	R/W	<p>DMA Enable for receive channel 0. The corresponding bits of this field enables/disables the receive FIFO DMA for channel 0.</p> <p>Values:</p> <p>0x0 (DISABLED): DMA disabled for receive channel 0</p> <p>0x1 (ENABLED): DMA enabled for receive channel 0</p> <p>Value After Reset: 0x0</p>

RXDMA_CHx (for x = 0; x <= I2S_RX_CHANNELS-1)

Name: Receiver Block DMA Register

Description: The RXDMA_CHx register allows access to enabled Receive channel x via a single point rather than through the LRBRx and RRBRx registers. This register is available only if I2S has dedicated DMA handshaking interface enabled for channel x.

Note: There is no read coherency logic; hence, the APB_DATA_WIDTH must be greater than or equal to the largest Receive channel word size to ensure all half data pairs can be accessed using a single read.

Channels can be enabled or disabled during the read cycles; however, it is recommended not to disable a channel in the middle of a stereo pair.

Size: 32 bits

Offset: 0x204 + 0x4*x

Bits	Name	Memory Access	Description
31:y	RSVD_RXDMA_CHx	R	RXDMA_CHx Reserved bits - Read Only Value After Reset: 0x0 Volatile: true Range Variable[y]: APB_DATA_WIDTH
x:0	RXDMA_CHx	R	Receiver Block DMA Register for channel x. These bits are used for reading stereo data pairs. Value After Reset: 0x0 Volatile: true Range Variable[x]: APB_DATA_WIDTH - 1

TXDMA_CHx (for x = 0; x <= I2S_TX_CHANNELS-1)

Name: Receiver Block DMA Register

Description: The TXDMA_CHx register allows access to enabled Transmit channel x via a single point rather than through the LTHR_x and RTHR_x registers. This register is available only if I2S has dedicated DMA handshaking interface enabled for channel x.

Note: There is no read coherency logic; hence, the APB_DATA_WIDTH must be greater than or equal to the largest Transmit channel word size to ensure all half data pairs can be accessed using a single read.

Channels can be enabled or disabled during the read cycles; however, it is recommended not to disable a channel in the middle of a stereo pair.

Size: 32 bits

Offset: 0x214 + 0x4*x

Bits	Name	Memory Access	Description
31:y	RSVD_TXDMA_CHx	W	<p>TXDMA_CHx Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[y]: APB_DATA_WIDTH</p>
x:0	TXDMA_CHx	W	<p>Receiver Block DMA Register for channel x. These bits are used for reading stereo data pairs.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p> <p>Range Variable[x]: APB_DATA_WIDTH - 1</p>

12.3.5 Programming Example

12.3.5.1 I2S As Transmitter (Master Mode)

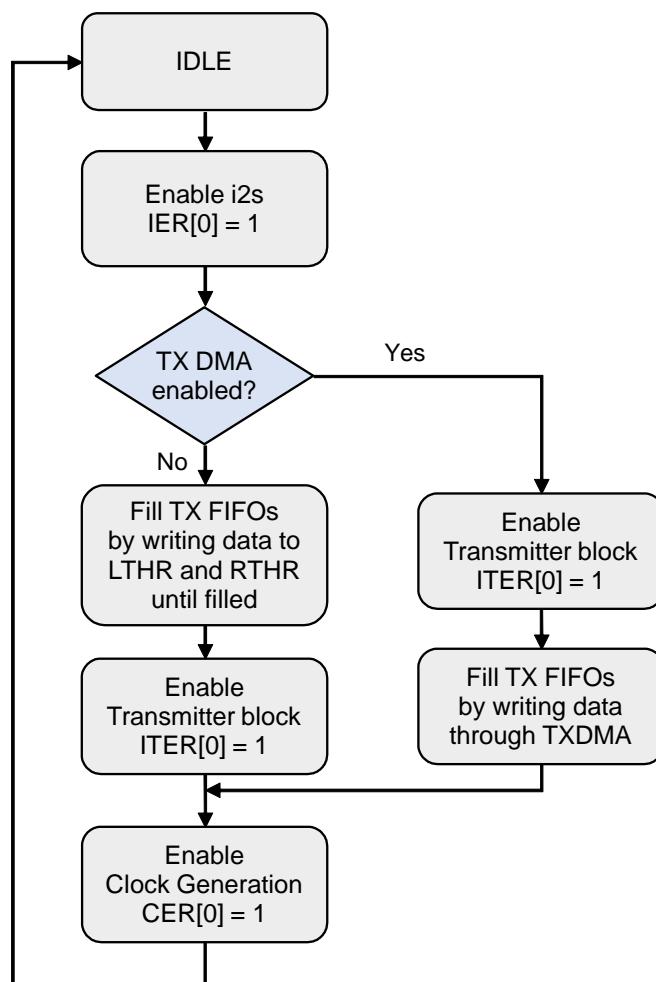


Figure 12-11 master transmitting process

12.3.5.2 I2S As Receiver (Master Mode)

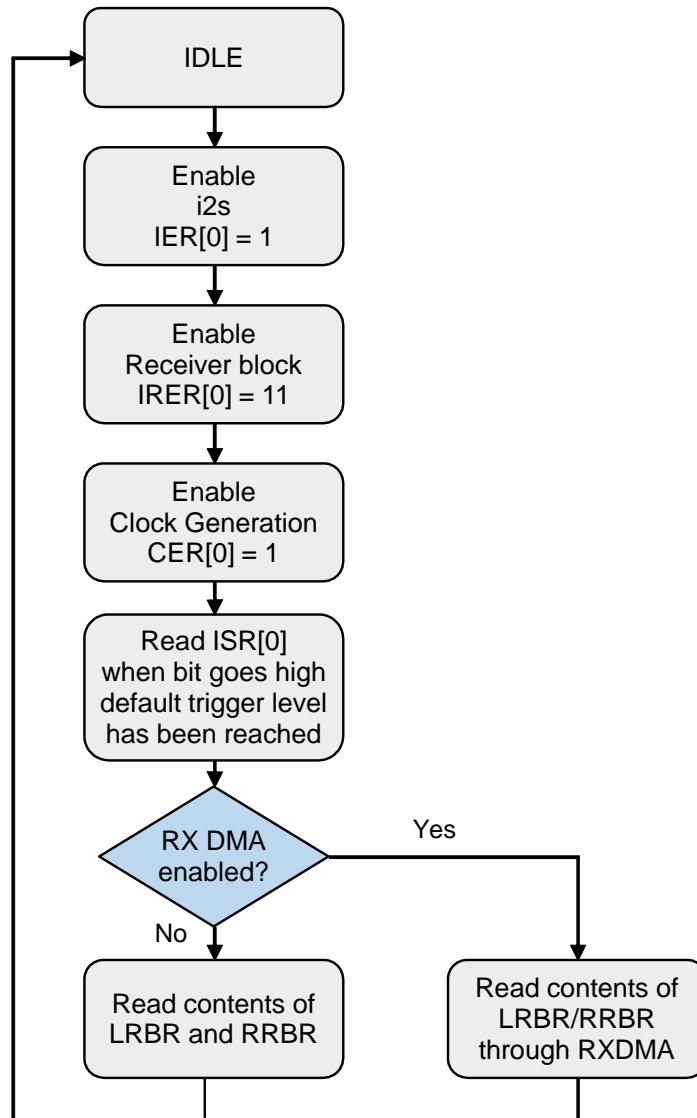


Figure 12-12 I2S MASTER receiving flow chart

12.3.6 Timing Sequence

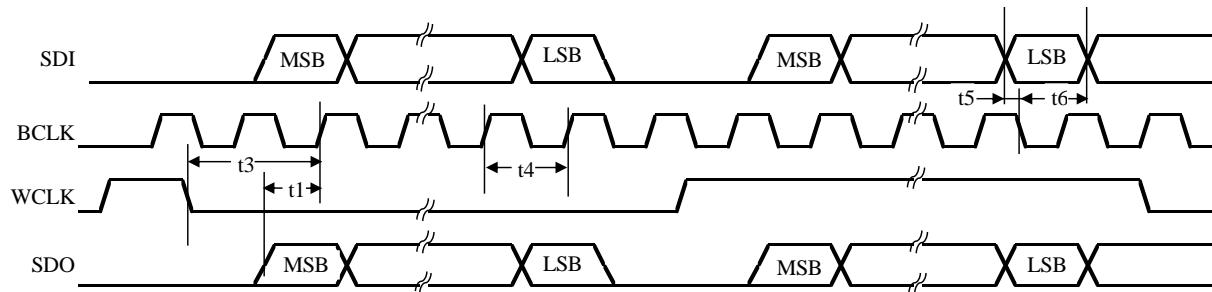


Figure 12-13 I2S timing sequence

12.4 UART

12.4.1 Overview

Data can be written from the host (CPU) to the UART via the APB bus, and then converted to serial form and transmitted to the target device. Serial data is also received by the UART and stored for the host (CPU) to read back.

Supported features:

- Support APB4.0, bus data bit width 32 bits
- Support DMA mode, automatic flow control mode
- Transmit and receive FIFO depth 32
- Functionality based on the 16550 industry standard
 - Number of data bits per character (5-8)
 - Optional parity bit (with odd, even select or Stick Parity)
 - Number of stop bits (1, 1.5 or 2)
 - Line break generation and detection
 - DMA signaling with two programmable modes
 - Prioritized interrupt identification
- Programmable decimal baud rate is supported

12.4.2 Block Diagram

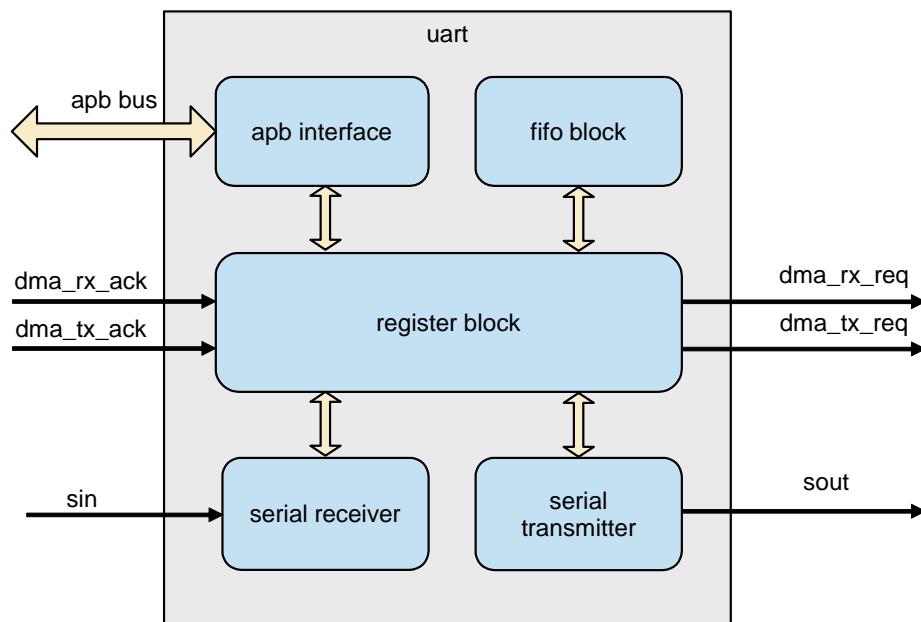


Figure 12-14 uart block diagram

12.4.3 Function Description

12.4.3.1 UART(RS232)

Serial communication between UART devices is asynchronous, with the start and stop bits indicating the start and end in the serial data. These bits are used to synchronize the two devices.

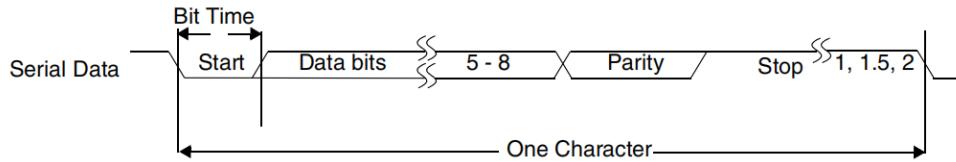


Figure 12-15 Serial Data Format

12.4.3.2 Baudrate

UART supports fractional Baud Rate:

$$\text{Baud Rate Divisor} = \frac{\text{Serial Clock Frequency}}{16 \times \text{Required Baud Rate}} = \text{BRDI} + \text{BRDF}$$

BRDI - Integer part of the divisor

BRDF - Fractional part of the divisor, $\text{DLF}/2^{\text{DLF_SIZE}}$

Serial Clock Frequency – apb clk frequency

Consider the following parameters:

Target baud rate(RBR) = 38400

APB clk frequency(PCLK)= 200 MHz

$\text{DLF_SIZE} = 4$

Then:

$$\text{BRDI} + \text{BRDF} = 200 \text{ MHz} / (16 \times 38400) = 325.520833$$

$$\text{BRDI} = 325, \quad \text{BRDF} = 0.520833$$

Therefore, the baud rate divisor fractional value (DLF) is:

$$\text{DLF} = 0.520833 \times 2^{\text{DLF_SIZE}} = 8.3 = 8(\text{Rounded})$$

The actual baud rate (GDR) is:

$$\text{GBR} = \frac{200 \text{ MHz}}{16 \times (\frac{8}{2^{\text{DLF_SIZE}}} + \text{BRDI})} = 38,402.457$$

Percentage ERROR:

$$\text{Error} = \frac{\text{GBR} - \text{RBR}}{\text{RBR}} = \frac{38,402.457 - 38400}{38400} = 0.006\%$$

12.4.3.3 Auto Flow Control

The uart can be configured to have a 16750-compatible Auto RTS and Auto CTS serial data flow control mode available; if FIFOs are not implemented, this mode cannot be selected. When Auto Flow Control is not selected, none of the corresponding logic is implemented and the mode cannot be enabled, reducing overall gate counts. When Auto Flow Control mode is selected, it can be enabled with the Modem Control Register (MCR[5]).

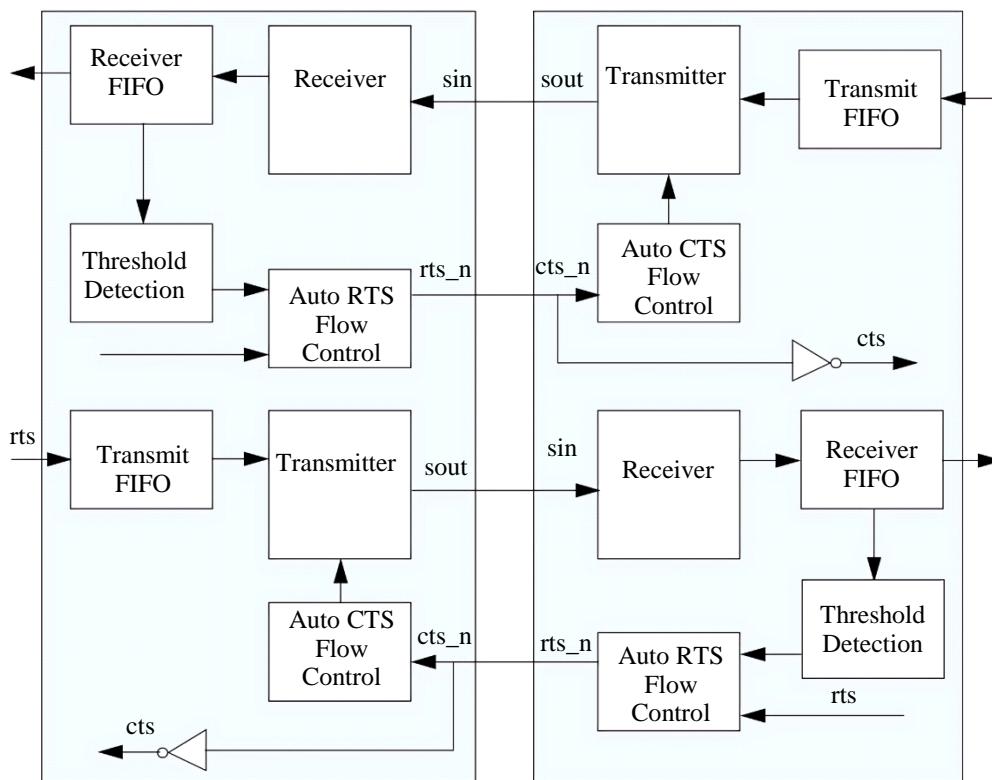


Figure 12-16 uart Auto Flow Control

Auto RTS – Becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- RTS (MCR[1] bit and MCR[5]bit are both set)
- FIFOs are enabled (FCR[0]) bit is set)
- SIR mode is disabled (MCR[6] bit is not set)

When Auto RTS is enabled, the rts_n output is forced inactive (high) when the receiver FIFO level reaches the threshold set by FCR[7:6]. When rts_n is connected to the cts_n input of another UART device, the other UART stops sending serial data until the receiver FIFO has available space.

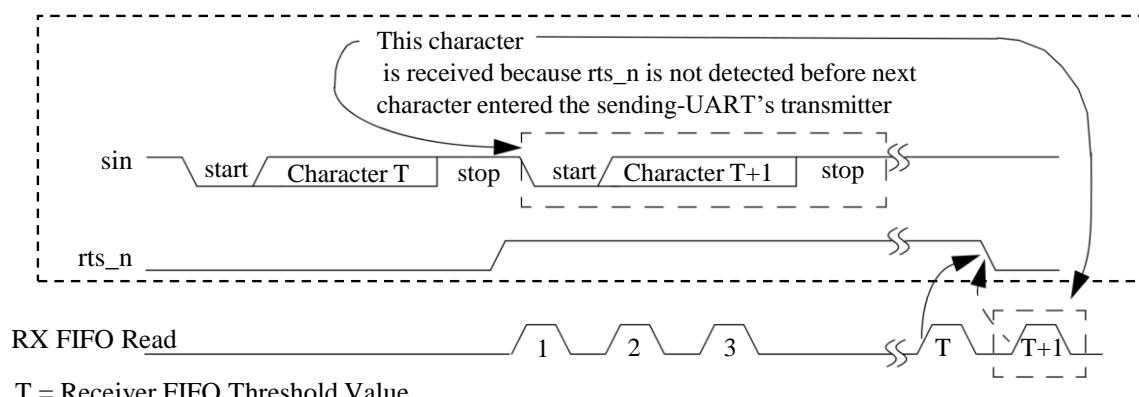


Figure 12-17 Auto CTS timing

Auto CTS – becomes active when the following occurs:

- Auto Flow Control is selected during configuration
- FIFOs are implemented
- AFCE (MCR[5] bit = 1)
- FIFOs are enabled through FIFO Control Register FCR[0] bit

- SIR mode is disabled (MCR[6] bit = 0)

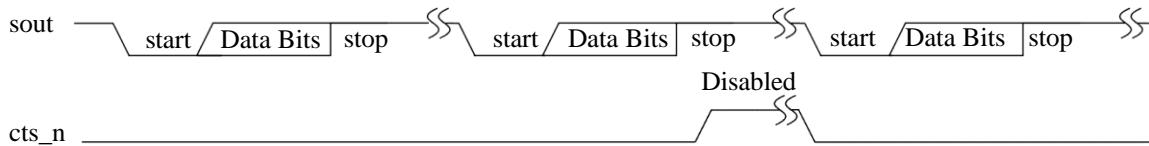


Figure 12-18 Auto CTS Timing

When Auto CTS is enabled (active), the UART transmitter is disabled whenever the *cts_n* input becomes inactive (high); this prevents overflowing the FIFO of the receiving UART.

When using the "FIFO full" status, software can poll this before each write to the Transmitter FIFO; for details, see "Programmable THRE Interrupt". When the *cts_n* input becomes active (low) again, transmission resumes.

12.4.3.4 Programmable THRE Interrupt

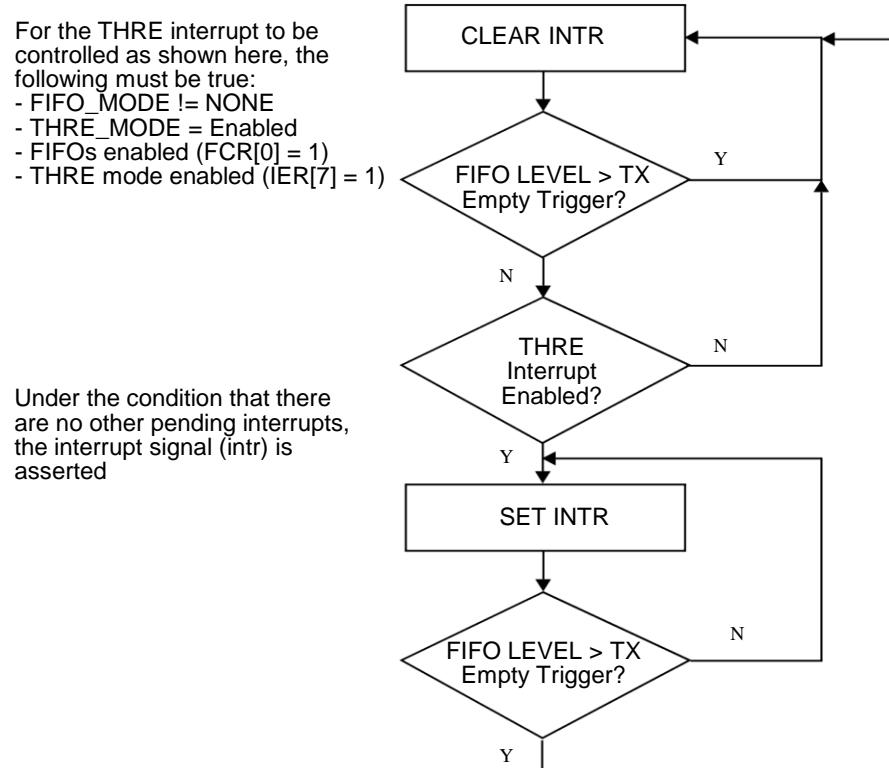


Figure 12-19 Flowchart of Interrupt Generation for Programmable THRE Interrupt Mode

12.4.3.5 Interrupt

The UART interrupt output signal (INTR) is a combined interrupt that is pulled up when one or more of the following are in effect. Specific interrupt information can be read in the IIR registers:

- Receiver Error
- Receiver Data Available
- Character Timeout (in FIFO mode only)
- Transmitter Holding Register Empty at/below threshold (in Programmable THRE interrupt mode)
- Modem Status
- Busy Detect Indication

Table 12-2 UART 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	<p>For Overrun/parity/framing/break interrupt reset control, the behavior is as follows:</p> <ul style="list-style-type: none"> ■ If LSR_STATUS_CLEAR=0 (RBR Read or LSR Read), then the status is cleared on: <ul style="list-style-type: none"> - Reading the line status register Or - In addition to an LSR read, the Receiver line status is also cleared when RX_FIFO is read. <ul style="list-style-type: none"> ■ If LSR_STATUS_CLEAR=1 (LSR Read), the status is cleared only on: <ul style="list-style-type: none"> - Reading the line status register. ■ For address received interrupt, the status is cleared on: <ul style="list-style-type: none"> - Reading the line status register
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 at or below threshold (Prog. THRE Mode)	Reading the IIR register (if source of interrupt); or, writing into THR (FIFOs or THRE Mode not selected)

Interrupt ID				interrupt set and reset functions			
Bit 3	Bit 2	Bit 1	Bit 0	Priority Level	Interrupt type	Interrupt Source	Interrupt Reset Control
						enabled)	or disabled) or XMIT FIFO above threshold (FIFOs and THRE Mode selected and enabled).
0	0	0	0	Fourth	Modem status	Clear to send or data set ready or ring indicator or data carrier detect. Note that if auto flow control mode is enabled, a change in CTS (that is, DCTS set) does not cause an interrupt.	Reading the Modem status register
0	1	1	1	Fifth	Busy detect indication	UART_16550_COMPATIBLE = NO and master has tried to write to the Line Control Register while the uart is busy (USR[0] is set to 1)	Reading the UART status register

12.4.4 Register Description

12.4.4.1 Register Overview

Register	Offset	Description
RBR	0x0	Receive Buffer Register
THR	0x0	Transmit Holding Register
DLL	0x0	Divisor Latch (Low)
IER	0x4	Interrupt Enable Register
DLH	0x4	Divisor Latch High
IIR	0x8	Interrupt Identification Register
FCR	0x8	FIFO Control Register
LCR	0xc	Line Control Register
MCR	0x10	Modem Control Register
LSR	0x14	Line Status Register
MSR	0x18	Modem Status Register
SCR	0x1c	Scratchpad Register

Register	Offset	Description
FAR	0x70	FIFO Access Register
TFR	0x74	Transmit FIFO Read
RFW	0x78	Receive FIFO Write
USR	0x7c	UART Status register
HTX	0xa4	Halt TX
DMASA	0xa8	DMA Software Acknowledge Register
TCR	0xac	Transceiver Control Register
DE_EN	0xb0	Driver Output Enable Register
RE_EN	0xb4	Receiver Output Enable Register
DET	0xb8	Driver Output Enable Timing Register
TAT	0xbc	TurnAround Timing Register
DLF	0xc0	Divisor Latch Fraction Register
UART_PROT_LEVEL	0xd0	UART Protection level
REG_TIMEOUT_RST	0xd4	Register timeout counter reset value

12.4.4.2 Register Detail Description

RBR

Name: Receive Buffer Register

Description: Receive Buffer Register.

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

Size: 32 bits

Offset: 0x0

Bits	Name	Memory Access	Description
31:8	RSVD_RBR	R	RBR 31to9or8 Reserved bits and read as zero (0). Value After Reset: 0x0
7:0	RBR	R	Receive Buffer Register. This register contains the data byte received on the serial input port (sin)

		<p>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.</p> <p>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 will be overwritten, resulting in an over-run error.</p> <p>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 will be preserved but any incoming data will be lost and an over-run error occurs.</p> <p>Note:</p> <p>When UART_9BIT_DATA_EN=0, this field width is 8.</p> <p>When UART_9BIT_DATA_EN=1, this field width is 9.</p> <p>Value After Reset: 0x0</p>
--	--	--

THR

Name: Transmit Holding Register

Description: Transmit Holding Register.

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

Size: 32 bits

Offset: 0x0

Bits	Name	Memory Access	Description
31:8	RSVD_THR	R	<p>THR 31to9or8 Reserved bits and read as zero (0).</p> <p>Value After Reset: 0x0</p>
7:0	THR	W	<p>Transmit Holding Register.</p> <p>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.</p> <p>If in non-FIFO mode or FIFO's are disabled (FCR[0] set to zero) 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</p>

		<p>data to be overwritten.</p> <p>If in FIFO mode and FIFO's are enabled (FCR[0] set to one) 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 is set during configuration. Any attempt to write data when the FIFO is full results in the write data being lost.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. When UART_9BIT_DATA_EN=0, this field width is 8. 2. When UART_9BIT_DATA_EN=1, this field width is 9. The 9th bit is applicable only when LCR_EXT[3]=1. <p>Value After Reset: 0x0</p>
--	--	---

DLL

Name: Divisor Latch (Low)

Description: Divisor Latch (Low).

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.

Size: 32 bits

Offset: 0x0

Bits	Name	Memory Access	Description
31:8	RSVD_DLL_31to8	R	<p>DLL 31to8 Reserved bits and read as zero (0).</p> <p>Value After Reset: 0x0</p>
7:0	DLL	R/W	<p>Divisor Latch (Low).</p> <p>This register makes up the lower 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.</p> <p>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).</p> <p>Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications will occur. Also, once the DLL is set, at least 8 clock cycles of the</p>

			slowest uart clock should be allowed to pass before transmitting or receiving data. Value After Reset: 0x0
--	--	--	--

IER**Name:** Interrupt Enable Register**Description:** Interrupt Enable Register.

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

Size: 32 bits**Offset:** 0x4

Bits	Name	Memory Access	Description
31:8	RSVD_IER_31to8	R	IER 31to8 Reserved bits and read as zero (0). Value After Reset: 0x0
7	PTIME	R/W	Programmable THRE Interrupt Mode Enable. Writeable only when THRE_MODE_USER == Enabled, always readable. This is used to enable/disable the generation of THRE Interrupt. Values: 0x0 (DISABLED): Disable Programmable THRE Interrupt Mode 0x1 (ENABLED): Enable Programmable THRE Interrupt Mode Value After Reset: 0x0
6:5	RSVD_IER_6to5	R	IER 6to5 Reserved bits read as zero (0). Value After Reset: 0x0
4	ELCOLR	R	Interrupt Enable Register: ELCOLR, this bit controls the method for clearing the status in the LSR register. This is applicable only for Overrun Error, Parity Error, Framing Error, and Break Interrupt status bits. 0: LSR status bits are cleared either on reading Rx FIFO (RBR Read) or On reading LSR register. 1: LSR status bits are cleared only on reading LSR register. Writeable only when LSR_STATUS_CLEAR = 1, otherwise always

Bits	Name	Memory Access	Description
			<p>readable.</p> <p>Values:</p> <p>0x0 (DISABLED): Disable ALC</p> <p>0x1 (ENABLED): Enable ALC</p> <p>Value After Reset: 0x0</p>
3	EDSSI	R/W	<p>Enable Modem Status Interrupt. This is used to enable/disable the generation of Modem Status Interrupt. This is the fourth highest priority interrupt.</p> <p>Values:</p> <p>0x0 (DISABLED): Disable Modem Status Interrupt</p> <p>0x1 (ENABLED): Enable Modem Status Interrupt</p> <p>Value After Reset: 0x0</p>
2	ELSI	R/W	<p>Enable Receiver Line Status Interrupt. This is used to enable/disable the generation of Receiver Line Status Interrupt. This is the highest priority interrupt.</p> <p>Values:</p> <p>0x0 (DISABLED): Disable Receiver Line Status Interrupt</p> <p>0x1 (ENABLED): Enable Receiver Line Status Interrupt</p> <p>Value After Reset: 0x0</p>
1	ETBEI	R/W	<p>Enable Transmit Holding Register Empty Interrupt. This is used to enable/disable the generation of Transmitter Holding Register Empty Interrupt. This is the third highest priority interrupt.</p> <p>Values:</p> <p>0x0 (DISABLED): Disable Transmit empty interrupt</p> <p>0x1 (ENABLED): Enable Transmit empty interrupt</p> <p>Value After Reset: 0x0</p>
0	ERBFI	R/W	<p>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 FIFO's enabled). These are the second highest priority interrupts.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x0 (DISABLED): Disable Receive data Interrupt</p> <p>0x1 (ENABLED): Enable Receive data Interrupt</p> <p>Value After Reset: 0x0</p>

DLH**Name:** Divisor Latch High**Description:** Divisor Latch High (DLH) Register.

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.

Size: 32 bits**Offset:** 0x4

Bits	Name	Memory Access	Description
31:8	RSVD_DLH	R	<p>DLH 31to8 Reserved bits and read as zero (0).</p> <p>Value After Reset: 0x0</p>
7:0	dlh	R/W	<p>Upper 8-bits of a 16-bit, read/write, Divisor Latch register that contains the baud rate divisor for the UART.</p> <p>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).</p> <p>Note that with the Divisor Latch Registers (DLL and DLH) set to zero, the baud clock is disabled and no serial communications will 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.</p> <p>Value After Reset: 0x0</p>

IIR**Name:** Interrupt Identification Register

Description: Interrupt Identification Register**Size:** 32 bits**Offset:** 0x8

Bits	Name	Memory Access	Description
31:8	RSVD_IIR_31to8	R	IIR 31to8 Reserved bits and read as 0. Value After Reset: 0x0
7:6	FIFOSE	R	FIFOs Enabled. This is used to indicate whether the FIFOs are enabled or disabled. Values: 0x0 (DISABLED): FIFOs are disabled 0x3 (ENABLED): FIFOs are enabled Value After Reset: 0x0
5:4	RSVD_IIR_5to4	R	IIR 5to4 Reserved bits read as 0. Value After Reset: 0x0
3:0	IID	R	Interrupt ID (or IID). This indicates the highest priority pending interrupt which can be one of the following types specified in Values. For information on several levels into which the interrupt priorities are split into, see the 'Interrupts' section in the uart Databook. Note: An interrupt of type 0111 (busy detect) will never get 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. Values: 0x0 (MODEM_STATUS): modem status 0x1 (NO_INTERRUPT_PENDING): no interrupt pending 0x2 (THR_EMPTY): THR empty 0x4 (RECEIVED_DATA_AVAILABLE): received data available 0x6 (RECEIVER_LINE_STATUS): receiver line status

Bits	Name	Memory Access	Description
			<p>0x7 (BUSY_DETECT): busy detect 0xc (CHARACTER_TIMEOUT): character timeout Value After Reset: 0x1</p>

FCR**Name:** FIFO Control Register

Description: This register is only valid when the uart is configured to have FIFO's implemented (FIFO_MODE != NONE). If FIFO's are not implemented, this register does not exist and writing to this register address will have no effect.

Size: 32 bits**Offset:** 0x8

Bits	Name	Memory Access	Description
31:8	RSVD_FCR_31to8	R	<p>FCR 31to8 Reserved bits and read as 0. Value After Reset: 0x0</p>
7:6	RT	W	<p>RCVR Trigger (or RT). This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt will be generated. In auto flow control mode, it is used to determine when the rts_n signal will be de-asserted only when RTC_FCT is disabled. It also determines when the dma_rx_req_n signal will be asserted when in certain modes of operation. For details on DMA support, refer to 'DMA Support' section of data book.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (FIFO_CHAR_1): 1 character in FIFO 0x1 (FIFO_QUARTER_FULL): FIFO 1/4 full 0x2 (FIFO_HALF_FULL): FIFO 1/2 full 0x3 (FIFO_FULL_2): FIFO 2 less than full <p>Value After Reset: 0x0</p>
5:4	TET	W	<p>TX Empty Trigger (or TET). Writes will have no effect when THRE_MODE_USER == Disabled. This is used to select the empty threshold level at which the THRE Interrupts will be generated when the mode is active. It also determines when the</p>

Bits	Name	Memory Access	Description
			<p>dma_tx_req_n signal will be asserted when in certain modes of operation. For details on DMA support, refer to 'DMA Support' section of data book.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (FIFO_EMPTY): FIFO Empty 0x1 (FIFO_CHAR_2): 2 characters in FIFO 0x2 (FIFO_QUARTER_FULL): FIFO 1/4 full 0x3 (FIFO_HALF_FULL): FIFO 1/2 full <p>Value After Reset: 0x0</p>
3	DMAM	W	<p>DMA Mode (or DMAM). 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). For details on DMA support, refer to 'DMA Support' section of data book.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (MODE0): Mode 0 0x1 (MODE1): Mode 1 <p>Value After Reset: 0x0</p>
2	XFIFOR	W	<p>XMIT FIFO Reset (or XFIFOR). This resets the control portion of the transmit FIFO and treats the FIFO as empty. This will also de-assert the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing' and it is not necessary to clear this bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (RESET): Transmit FIFO reset <p>Value After Reset: 0x0</p>
1	RFIFOR	W	<p>RCVR FIFO Reset (or RFIFOR). This resets the control portion of the receive FIFO and treats the FIFO as empty. This will also de-assert the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing' and it is not necessary to clear this bit.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x1 (RESET): Receive FIFO reset</p> <p>Value After Reset: 0x0</p>
0	FIFOE	W	<p>FIFO Enable (or FIFOE). 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.</p> <p>Values:</p> <p>0x0 (DISABLED): FIFO disabled</p> <p>0x1 (ENABLED): FIFO enabled</p> <p>Value After Reset: 0x0</p>

LCR**Name:** Line Control Register**Description:** Line Control Register**Size:** 32 bits**Offset:** 0xc

Bits	Name	Memory Access	Description
31:8	RSVD_LCR_31to8	R	<p>LCR 31to8 Reserved bits and read as 0.</p> <p>Value After Reset: 0x0</p>
7	DLAB	R/W	<p>Divisor Latch Access Bit.</p> <p>If <code>UART_16550_COMPATIBLE == NO</code> then, writeable only when UART is not busy (<code>USR[0]</code> is zero), otherwise always writable and 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.</p> <p>Values:</p> <p>0x0 (DISABLED): Divisor Latch register is writable only when UART Not BUSY</p> <p>0x1 (ENABLED): Divisor Latch register is always readable and</p>

Bits	Name	Memory Access	Description
			writable Value After Reset: 0x0
6	BC	R/W	<p>Break Control Bit.</p> <p>This is used to cause a break condition to be transmitted to the receiving device. If set to one 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 one) 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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Serial output is released for data transmission 0x1 (ENABLED): Serial output is forced to spacing state Value After Reset: 0x0
5	SP	R/W	<p>Stick Parity.</p> <p>If UART_16550_COMPATIBLE = NO, then writeable only when UART is not busy (USR[0] is 0); otherwise always writable and 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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Stick parity disabled 0x1 (ENABLED): Stick parity enabled Value After Reset: 0x0
4	EPS	R/W	<p>Even Parity Select.</p> <p>If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable and always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic '1's is transmitted or checked. If set to zero, an odd number of logic '1's is transmitted or checked.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x0 (ODD_PARITY): an odd parity is transmitted or checked</p> <p>0x1 (EVEN_PARITY): an even parity is transmitted or checked</p> <p>Value After Reset: 0x0</p>
3	PEN	R/W	<p>Parity Enable.</p> <p>If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable and always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively.</p> <p>Values:</p> <p>0x0 (DISABLED): disable parity</p> <p>0x1 (ENABLED): enable parity</p> <p>Value After Reset: 0x0</p>
2	STOP	R/W	<p>Number of stop bits.</p> <p>If UART_16550_COMPATIBLE == NO then, writeable only when UART is not busy (USR[0] is zero), otherwise always writable and always readable. This is used to select the number of stop bits per character that the peripheral will transmit and receive. If set to zero, one stop bit is transmitted in the serial data.</p> <p>If set to one and the data bits are set to 5 (LCR[1:0] set to zero) 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 will only check the first stop bit.</p> <p>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; for details on idle time between transmitted transfers, refer to 'Back-to-Back Character Stream Transmission' section in data book.</p> <p>Values:</p> <p>0x0 (STOP_1BIT): 1 stop bit</p> <p>0x1 (STOP_1_5BIT_OR_2BIT): 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
1:0	DLS	R/W	<p>Data Length Select (or CLS as used in legacy).</p> <p>If <code>UART_16550_COMPATIBLE == NO</code> then, writeable only when <code>UART</code> is not busy (<code>USR[0]</code> is zero), otherwise always writable and always readable. When <code>DLS_E</code> in <code>LCR_EXT</code> is set to 0, this register is used to select the number of data bits per character that the peripheral will transmit and receive.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (CHAR_5BITS): 5 data bits per character 0x1 (CHAR_6BITS): 6 data bits per character 0x2 (CHAR_7BITS): 7 data bits per character 0x3 (CHAR_8BITS): 8 data bits per character <p>Value After Reset: 0x0</p>

MCR

Name: Modem Control Register

Description: Modem Control Register

Size: 32 bits

Offset: 0x10

Bits	Name	Memory Access	Description
31:7	RSVD_MCR_31to7	R	<p>MCR 31to7 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
6	SIRE	R	<p>SIR Mode Enable</p> <p>Writeable only when <code>SIR_MODE == Enabled</code>, always readable. This is used to enable/ disable the IrDA SIR Mode features as described in section 'IrDA 1.0 SIR Protocol' in the databook.</p> <p>Note: To enable SIR mode, write the appropriate value to the MCR register before writing to the LCR register. For details of the recommended programming sequence, refer to 'Programming Examples' section of data book.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x0 (DISABLED): IrDA SIR Mode disabled</p> <p>0x1 (ENABLED): IrDA SIR Mode enabled</p> <p>Value After Reset: 0x0</p>
5	AFCE	R/W	<p>Auto Flow Control Enable</p> <p>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 as described in section 'Auto Flow Control' in data book.</p> <p>Values:</p> <p>0x0 (DISABLED): Auto Flow Control Mode disabled</p> <p>0x1 (ENABLED): Auto Flow Control Mode enabled</p> <p>Value After Reset: 0x0</p>
4	LoopBack	R/W	<p>LoopBack Bit</p> <p>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 zero), 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.</p> <p>If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line.</p> <p>Values:</p> <p>0x0 (DISABLED): Loopback mode disabled</p> <p>0x1 (ENABLED): Loopback mode enabled</p> <p>Value After Reset: 0x0</p>
3	OUT2	R/W	<p>OUT2</p> <p>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. Note that in Loopback mode (MCR[4] set to</p>

Bits	Name	Memory Access	Description
			<p>one), the out2_n output is held inactive high while the value of this location is internally looped back to an input.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (OUT2_0): out2_n de-asserted (logic 1) 0x1 (OUT2_1): out2_n asserted (logic 0) <p>Value After Reset: 0x0</p>
2	OUT1	R/W	<p>OUT1</p> <p>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. Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (OUT1_0): out1_n de-asserted (logic 1) 0x1 (OUT1_1): out1_n asserted (logic 0) <p>Value After Reset: 0x0</p>
1	RTS	R/W	<p>Request to Send.</p> <p>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.</p> <p>When Auto RTS Flow Control is not enabled (MCR[5] set to zero), 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 one) and FIFO's enable (FCR[0] set to one), 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). The rts_n signal will be de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Request to Send rts_n de-asserted (logic 1) 0x1 (ACTIVE): Request to Send rts_n asserted (logic 0) <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
0	DTR	R/W	<p>Data Terminal Ready.</p> <p>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.</p> <p>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 one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): dtr_n de-asserted (logic1) 0x1 (ACTIVE): dtr_n asserted (logic 0) <p>Value After Reset: 0x0</p>

LSR

Name: Line Status Register

Description: Line Status Register

Size: 32 bits

Offset: 0x14

Bits	Name	Memory Access	Description
31:9	RSVD_LSR_31to9	R	<p>LSR 31to9 Reserved bits read as zero.</p> <p>Value After Reset: 0x0</p>
8	RSVD_ADDR_RCVD	R	<p>ISR 8 Reserved bit for UART_9BIT_DATA_EN</p> <p>Value After Reset: 0x0</p>
7	RFE	R	<p>Receiver FIFO Error bit.</p> <p>This bit is only relevant when FIFO_MODE != NONE AND FIFO's are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO.</p>

Bits	Name	Memory Access	Description
			<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_RX_FIFO_ERROR): No error in RX FIFO 0x1 (RX_FIFO_ERROR): Error in RX FIFO <p>Value After Reset: 0x0</p>
6	TEM _T	R	<p>Transmitter Empty bit.</p> <p>If in FIFO mode (FIFO_MODE != NONE) and FIFO's enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in the non-FIFO mode or FIFO's are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Transmitter not empty 0x1 (ENABLED): Transmitter empty <p>Value After Reset: 0x1</p>
5	THRE	R	<p>Transmit Holding Register Empty bit.</p> <p>If THRE_MODE_USER = Disabled or THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty.</p> <p>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 one and FCR[0] set to one 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. Programmable THRE interrupt mode operation is described in detail in section 'Programmable THRE Interrupt' in data book.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): THRE interrupt control is disabled

Bits	Name	Memory Access	Description
			<p>0x1 (ENABLED): THRE interrupt control is enabled</p> <p>Value After Reset: 0x1</p>
4	BI	R	<p>Break Interrupt bit.</p> <p>This is used to indicate the detection of a break sequence on the serial input data.</p> <p>If in UART mode 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.</p> <p>If in infrared mode 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 zeros, to be received by the UART.</p> <p>In the 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.</p> <p>If configuration parameter LSR_STATUS_CLEAR = 0:</p> <p>Reading the LSR or RBR register clears the BI bit.</p> <p>If configuration parameter LSR_STATUS_CLEAR = 1:</p> <p>If register field IER.ELCOLR = 0: Reading the LSR or RBR register clears the BI bit.</p> <p>If register field IER.ELCOLR = 1: Reading the LSR register clears the BI bit.</p> <p>In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read.</p> <p>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.</p> <p>Values:</p> <p>0x0 (NO_BREAK): No break sequence detected</p> <p>0x1 (BREAK): Break sequence detected</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
3	FE	R	<p>Framing Error bit.</p> <p>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.</p> <p>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 will try 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 i.e. data, and/or parity and stop.</p> <p>It should be noted that the Framing Error (FE) bit (LSR[3]) will be 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.</p> <p>If configuration parameter LSR_STATUS_CLEAR = 0:</p> <p>Reading the LSR or RBR register clears the FE bit.</p> <p>If configuration parameter LSR_STATUS_CLEAR = 1:</p> <p>If register field IER.ELCOLR = 0: Reading the LSR or RBR register clears the FE bit.</p> <p>If register field IER.ELCOLR = 1: Reading the LSR register clears the FE bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_FRAMING_ERROR): no framing error 0x1 (FRAMING_ERROR): framing error <p>Value After Reset: 0x0</p>
2	PE	R	<p>Parity Error bit.</p> <p>This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set.</p> <p>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.</p> <p>It should be noted that the Parity Error (PE) bit (LSR[2]) will be set if a break interrupt has occurred, as indicated by Break Interrupt</p>

Bits	Name	Memory Access	Description
			<p>(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).</p> <p>If configuration parameter LSR_STATUS_CLEAR = 0:</p> <p>Reading the LSR or RBR register clears the PE bit.</p> <p>If configuration parameter LSR_STATUS_CLEAR = 1:</p> <p>If register field IER.ELCOLR = 0: Reading the LSR or RBR register clears the PE bit.</p> <p>If register field IER.ELCOLR = 1: Reading the LSR register clears the PE bit.</p> <p>Values:</p> <p>0x0 (NO_PARITY_ERROR): no parity error</p> <p>0x1 (PARITY_ERROR): parity error</p> <p>Value After Reset: 0x0</p>
1	OE	R	<p>Overrun error bit.</p> <p>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.</p> <p>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.</p> <p>If configuration parameter LSR_STATUS_CLEAR = 0:</p> <p>Reading the LSR register clears the OE bit.</p> <p>Note: In this case the OE bit is only cleared on LSR register read to avoid loss of OE interrupt status information by DMA operation (RBR read). However, with respect to PE, FE and BI interrupt status bits, these information are available with the received data in RX FIFO.</p> <p>the read of RBR register. If configuration parameter LSR_STATUS_CLEAR = 1:</p> <p>If register field IER.ELCOLR = 0: Reading the LSR or RBR register</p>

Bits	Name	Memory Access	Description
			<p>clears the OE bit.</p> <p>If register field IER.ELCOLR = 1: Reading the LSR register clears the OE bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_OVER_RUN_ERROR): no overrun error 0x1 (OVER_RUN_ERROR): overrun error <p>Value After Reset: 0x0</p>
0	DR	R	<p>Data Ready bit.</p> <p>This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. This bit is cleared when the RBR is read in the non-FIFO mode, or when the receiver FIFO is empty, in the FIFO mode.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NOT_READY): data not ready 0x1 (READY): data ready <p>Value After Reset: 0x0</p>

MSR

Name: Modem Status Register

Description: Whenever bits 0, 1, 2 or 3 is set to logic one, to indicate a change on the modem control inputs, a modem status interrupt will be generated if enabled via the IER regardless of when the change occurred. The bits (bits 0, 1, 3) 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.

Size: 32 bits

Offset: 0x18

Bits	Name	Memory Access	Description
31:8	RSVD_MSR_31to8	R	<p>MSR 31to8 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
7	DCD	R	<p>Data Carrier Detect.</p> <p>This is used to indicate the current state of the modem control line dcd_n. That is this bit is the complement 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.</p> <p>In Loopback Mode (MCR[4] set to one), DCD is the same as MCR[3] (Out2).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DEASSERTED): dcd_n input is de-asserted (logic 1) 0x1 (ASSERTED): dcd_n input is asserted (logic 0) <p>Value After Reset: 0x0</p>
6	RI	R	<p>Ring Indicator.</p> <p>This is used to indicate the current state of the modem control line ri_n. That is this bit is the complement 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.</p> <p>In Loopback Mode (MCR[4] set to one), RI is the same as MCR[2] (Out1).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DEASSERTED): ri_n input is de-asserted (logic 1) 0x1 (ASSERTED): ri_n input is asserted (logic 0) <p>Value After Reset: 0x0</p>
5	DSR	R	<p>Data Set Ready.</p> <p>This is used to indicate the current state of the modem control line dsr_n. That is this bit is the complement 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.</p> <p>In Loopback Mode (MCR[4] set to one), DSR is the same as MCR[0] (DTR).</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DEASSERTED): dsr_n input is de-asserted (logic 1)

Bits	Name	Memory Access	Description
			<p>0x1 (ASSERTED): dsr_n input is asserted (logic 0)</p> <p>Value After Reset: 0x0</p>
4	CTS	R	<p>Clear to Send.</p> <p>This is used to indicate the current state of the modem control line cts_n. That is, this bit is the complement 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.</p> <p>In Loopback Mode (MCR[4] set to one), CTS is the same as MCR[1] (RTS).</p> <p>Values:</p> <p>0x0 (DEASSERTED): cts_n input is de-asserted (logic 1)</p> <p>0x1 (ASSERTED): cts_n input is asserted (logic 0)</p> <p>Value After Reset: 0x0</p>
3	DDCD	R	<p>Delta Data Carrier Detect.</p> <p>This is used to indicate that the modem control line dcd_n has changed since the last time the MSR was read.</p> <p>Reading the MSR clears the DDCD bit. In Loopback Mode (MCR[4] set to one), DDCD reflects changes on MCR[3] (Out2).</p> <p>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 will get set when the reset is removed if the dcd_n signal remains asserted.</p> <p>Values:</p> <p>0x0 (NO_CHANGE): No change on dcd_n since last read of MSR</p> <p>0x1 (CHANGE): change on dcd_n since last read of MSR</p> <p>Value After Reset: 0x0</p>
2	TERI	R	<p>Trailing Edge of Ring Indicator.</p> <p>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.</p> <p>Reading the MSR clears the TERI bit. In Loopback Mode (MCR[4] set to one), TERI reflects when MCR[2] (Out1) has changed state</p>

Bits	Name	Memory Access	Description
			<p>from a high to a low.</p> <p>Values:</p> <p>0x0 (NO_CHANGE): no change on ri_n since last read of MSR</p> <p>0x1 (CHANGE): change on ri_n since last read of MSR</p> <p>Value After Reset: 0x0</p>
1	DDSR	R	<p>Delta Data Set Ready.</p> <p>This is used to indicate that the modem control line dsr_n has changed since the last time the MSR was read.</p> <p>Reading the MSR clears the DDSR bit. In Loopback Mode (MCR[4] set to one), DDSR reflects changes on MCR[0] (DTR).</p> <p>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 will get set when the reset is removed if the dsr_n signal remains asserted.</p> <p>Values:</p> <p>0x0 (NO_CHANGE): no change on dsr_n since last read of MSR</p> <p>0x1 (CHANGE): change on dsr_n since last read of MSR</p> <p>Value After Reset: 0x0</p>
0	DCTS	R	<p>Delta Clear to Send.</p> <p>This is used to indicate that the modem control line cts_n has changed since the last time the MSR was read.</p> <p>Reading the MSR clears the DCTS bit. In Loopback Mode (MCR[4] set to one), DCTS reflects changes on MCR[1] (RTS).</p> <p>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 will get set when the reset is removed if the cts_n signal remains asserted.</p> <p>Values:</p> <p>0x0 (NO_CHANGE): no change on cts_n since last read of MSR</p> <p>0x1 (CHANGE): change on cts_n since last read of MSR</p> <p>Value After Reset: 0x0</p>

SCR**Name:** Scratchpad Register**Description:** Scratchpad Register**Size:** 32 bits**Offset:** 0x1c

Bits	Name	Memory Access	Description
31:8	RSVD_SCR_31to8	R	<p>SCR 31to8 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
7:0	SCR	R/W	<p>This register is for programmers to use as a temporary storage space. It has no defined purpose in the uart.</p> <p>Value After Reset: 0x0</p>

FAR**Name:** FIFO Access Register**Description:** FIFO Access Register**Size:** 32 bits**Offset:** 0x70

Bits	Name	Memory Access	Description
31:1	RSVD_FAR_31to1	R	<p>FAR 31to1 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
0	FAR	R/W	<p>Writes will 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 FIFO's 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.</p> <p>Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and</p>

			<p>the FIFO's are treated as empty.</p> <p>Values:</p> <p>0x0 (DISABLED): FIFO access mode disabled</p> <p>0x1 (ENABLED): FIFO access mode enabled</p> <p>Value After Reset: 0x0</p>
--	--	--	---

TFR

Name: Transmit FIFO Read

Description: 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.

Size: 32 bits

Offset: 0x74

Bits	Name	Memory Access	Description
31:8	RSVD_TFR_31to8	R	<p>TFR 31to8 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
7:0	TFR	R	<p>Transmit FIFO Read.</p> <p>These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one).</p> <p>When FIFO's 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.</p> <p>When FIFO's are not implemented or not enabled, reading this register gives the data in the THR.</p> <p>Value After Reset: 0x0</p>

RFW

Name: Receive FIFO Write

Description: 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.

Size: 32 bits**Offset:** 0x78

Bits	Name	Memory Access	Description
31:10	RSVD_RFW_31to10	R	<p>RFW 31to10 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
9	RFFE	W	<p>Receive FIFO Framing Error.</p> <p>These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFO's are implemented and enabled, this bit is used to write framing error detection information to the receive FIFO. When FIFO's are not implemented or not enabled, this bit is used to write framing error detection information to the RBR.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Frame error disabled 0x1 (ENABLED): Frame error enabled <p>Value After Reset: 0x0</p>
8	RFPE	W	<p>Receive FIFO Parity Error.</p> <p>These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFO's are implemented and enabled, this bit is used to write parity error detection information to the receive FIFO. When FIFO's are not implemented or not enabled, this bit is used to write parity error detection information to the RBR.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Parity error disabled 0x1 (ENABLED): Parity error enabled <p>Value After Reset: 0x0</p>
7:0	RFWD	W	<p>Receive FIFO Write Data.</p> <p>These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFO's 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 FIFO's are not implemented or not enabled, the data that is written to the</p>

Bits	Name	Memory Access	Description
			<p>RFWD is pushed into the RBR.</p> <p>Value After Reset: 0x0</p>

USR**Name:** UART Status register**Description:** UART Status register.**Size:** 32 bits**Offset:** 0x7c

Bits	Name	Memory Access	Description
31:5	RSVD_USR_31to5	R	<p>USR 31to5 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
4	RSVD_RFF	R	<p>Receive FIFO Full.</p> <p>This bit is only valid when FIFO_STAT == NO. This is used to indicate that the receive FIFO is completely full. That is: This bit is cleared when the RX FIFO is no longer full.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NOT_FULL): Receive FIFO not full 0x1 (FULL): Receive FIFO full <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
3	RSVD_RFNE	R	<p>Receive FIFO Not Empty.</p> <p>This bit is only valid when FIFO_STAT == NO. This is used to indicate that the receive FIFO contains one or more entries. This bit is cleared when the RX FIFO is empty.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (EMPTY): Receive FIFO is empty 0x1 (NOT_EMPTY): Receive FIFO is not empty</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
2	RSVD_TFE	R	<p>Transmit FIFO Empty.</p> <p>This bit is only valid when FIFO_STAT == NO. This is used to indicate that the transmit FIFO is completely empty. This bit is cleared when the TX FIFO is no longer empty.</p> <p>Values:</p> <p>0x0 (NOT_EMPTY): Transmit FIFO is not empty 0x1 (EMPTY): Transmit FIFO is empty</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
1	RSVD_TFNF	R	<p>Transmit FIFO Not Full.</p> <p>This bit is only valid when FIFO_STAT == NO. This is used to indicate that the transmit FIFO is not full. This bit is cleared when the TX FIFO is full.</p> <p>Values:</p> <p>0x0 (FULL): Transmit FIFO is full 0x1 (NOT_FULL): Transmit FIFO is not full</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	BUSY	R	<p>UART Busy.</p> <p>This bit is only valid when UART_16550_COMPATIBLE == NO. This indicates that a serial transfer is in progress, when cleared indicates that the uart is idle or inactive.</p> <p>This bit will be set to 1 (busy) under any of the following conditions:</p> <ul style="list-style-type: none"> - Transmission in progress on serial interface - Transmit data present in THR, when FIFO access mode is not

Bits	Name	Memory Access	Description
			<p>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)</p> <ul style="list-style-type: none"> - Reception in progress on the interface - Receive data present in RBR, when FIFO access mode is not being used (FAR = 0) <p>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 the 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 will also be delayed by several cycles of the slower clock.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (IDLE): uart is idle or inactive 0x1 (BUSY): uart is busy (actively transferring data) <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

HTX

Name: Halt TX

Description: Halt TX

Size: 32 bits

Offset: 0xa4

Bits	Name	Memory Access	Description
31:1	RSVD_HTX_31to1	R	<p>HTX 31to1 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
0	HTX	R/W	Halt TX.

		<p>Writes will have no effect when FIFO_MODE == NONE, always readable. This register is used to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFO's are implemented and enabled.</p> <p>Note, if FIFO's are implemented and not enabled the setting of the halt TX register will have no effect on operation.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Halt Transmission disabled 0x1 (ENABLED): Halt Transmission enabled <p>Value After Reset: 0x0</p>
--	--	--

DMASA

Name: DMA Software Acknowledge Register

Description: DMA Software Acknowledge Register

Size: 32 bits

Offset: 0xa8

Bits	Name	Memory Access	Description
31:1	RSVD_DMASA_31to1	R	<p>DMASA 31to1 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
0	DMASA	W	<p>DMA Software Acknowledge.</p> <p>Writes will have no effect when DMA_EXTRA == No. This register is used to perform 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 will cause the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing' and it is not necessary to clear this bit.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (SOFT_ACK): DMA software acknowledge <p>Value After Reset: 0x0</p>

TCR**Name:** Transceiver Control Register

Description: 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.

Size: 32 bits**Offset:** 0xac

Bits	Name	Memory Access	Description
31:5	RSVD_TCR_31to5	R	<p>TCR 31to5 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
4:3	XFER_MODE	R/W	<p>Transfer Mode.</p> <p>Values:</p> <p>0x0 (XFER_MODE_0): In this mode, transmit and receive can happen simultaneously. The user can enable DE_EN, RE_EN at any point of time.</p> <p>Turn around timing as programmed in the TAT register is not applicable in this mode.</p> <p>0x1 (XFER_MODE_1): In this mode, DE and RE are mutually exclusive. Either DE or RE only one of them is expected to be enabled through programming.</p> <p>Hardware will consider the Turn Around timings which are programmed in the TAT register while switching from RE to DE or DE to RE. For transmission Hardware will wait if it is in middle of receiving any transfer, before it starts transmitting.</p> <p>0x2 (XFER_MODE_2): In this mode, DE and RE are mutually exclusive. Once DE_EN/RE_EN is programmed - by default 're' will be enabled and uart controller will be ready to receive. If the user programs the TX FIFO with the data then uart, after ensuring no receive is in progress, disable 're' and enable 'de' signal.</p> <p>Once the TX FIFO becomes empty, 're' signal gets enabled and 'de' signal will be disabled. In this mode of operation hardware will consider the Turn Around timings which are programmed in the TAT register while switching from RE to DE or DE to RE. In this mode, 'de' and 're' signals are strictly complementary to each other.</p>

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
2	DE_POL	R/W	<p>Driver Enable Polarity.</p> <p>Values:</p> <p>0x0 (LOW): DE signal is active low</p> <p>0x1 (HIGH): DE signal is active high</p> <p>Value After Reset: 0x1</p>
1	RE_POL	R/W	<p>Receiver Enable Polarity.</p> <p>Values:</p> <p>0x0 (LOW): RE signal is active low</p> <p>0x1 (HIGH): RE signal is active high</p> <p>Value After Reset: 0x1</p>
0	RS485_EN	R/W	<p>RS485 Transfer Enable.</p> <p>0 : In this mode, the transfers are still in the RS232 mode. All other fields in this register are reserved and register DE_EN/RE_EN/TAT are also reserved.</p> <p>1 : In this mode, the transfers will happen in RS485 mode. All other fields of this register are applicable.</p> <p>Values:</p> <p>0x0 (DISABLE): RS232 Mode</p> <p>0x1 (ENABLE): RS485 Mode</p> <p>Value After Reset: 0x0</p>

DE_EN

Name: Driver Output Enable Register

Description: 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.

Size: 32 bits**Offset:** 0xb0

Bits	Name	Memory Access	Description
31:1	RSVD_DE_EN_31to1	R	<p>DE_EN 31to1 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
0	DE_Enable	R/W	<p>DE Enable control.</p> <p>The 'DE Enable' register bit is used to control assertion and de-assertion of 'de' signal.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DE_ASSERT): De-assert 'de' signal 0x1 (ASSERT): Assert 'de' signal <p>Value After Reset: 0x0</p>

RE_EN

Name: Receiver Output Enable Register**Description:** 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.

Size: 32 bits**Offset:** 0xb4

Bits	Name	Memory Access	Description
31:1	RSVD_RE_EN_31to1	R	<p>RE_EN 31to1 Reserved bits read as 0.</p> <p>Value After Reset: 0x0</p>
0	RE_Enable	R/W	<p>RE Enable control.</p> <p>The 'RE Enable' register bit is used to control assertion and de-assertion of 're' signal.</p>

		Values: 0x0 (DE_ASSERT): De-assert 're' signal 0x1 (ASSERT): Assert 're' signal Value After Reset: 0x0
--	--	--

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

Size: 32 bits**Offset:** 0xb8

Bits	Name	Memory Access	Description
31:24	RSVD_DE_DEAT_31to24	R	DET 31to24 Reserved bits read as 0. Value After Reset: 0x0
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 sout to the falling edge of Driver output enable signal. Value After Reset: 0x0
15:8	RSVD_DE_AT_15to8	R	DET 15to8 Reserved bits read as 0. Value After Reset: 0x0
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. Value After Reset: 0x0

TAT**Name:** TurnAround Timing Register**Description:** 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.

Size: 32 bits**Offset:** 0xbc

Bits	Name	Memory Access	Description
31:16	RE_to_DE	R/W	<p>Receiver Enable to Driver Enable TurnAround time.</p> <p>Turnaround time (in terms of serial clock) for RE De-assertion to DE assertion.</p> <p>Note:</p> <ul style="list-style-type: none"> - 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. <p>Value After Reset: 0x0</p>
15:0	DE_to_RE	R/W	<p>Driver Enable to Receiver Enable TurnAround time.</p> <p>Turnaround time (in terms of serial clock) for DE De-assertion to RE assertion.</p> <p>Note: The actual time is the programmed value + 1.</p> <p>Value After Reset: 0x0</p>

DLF**Name:** Divisor Latch Fraction Register**Description:** 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.**Size:** 32 bits

Offset: 0xc0

Bits	Name	Memory Access	Description
31:4	RSVD_DLF	R	DLF 31 to DLF_SIZE Reserved bits read as 0. Value After Reset: 0x0
3:0	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). For information on DLF values to be programmed for DLF_SIZE=4, see the 'Fractional Baud Rate Support' section in the uart Databook. Value After Reset: 0x0

UART_PROT_LEVEL**Name:** UART Protection level**Description:** UART Protection level register**Size:** 32 bits**Offset:** 0xd0

Bits	Name	Memory Access	Description
31:3	RSVD_UART_PROT_LEVEL	R	UART_PROT_LEVEL[31:29] Reserved field-read-only Value After Reset: 0x0
2:0	UART_PROT_LEVEL	R/W	Protection level register. Enabling protection on any of its three bits would require a greater or equal privilege on the PPROT signal to gain access to protected registers. Value After Reset: 0x2

REG_TIMEOUT_RST**Name:** Register timeout counter reset value**Description:** Register timeout counter reset register This register keeps the reset value of reg_timer counter register. The reset value of the register is REG_TIMEOUT_DEFAULT The default reset value can be further modified if HC_REG_TIMEOUT_VALUE = 0. The final programmed value (or the default reset

value if not programmed) determines what value the reg_timeout counter register starts counting down from. A zero on the counter will break the hung transaction with PSLVERR high

Size: 32 bits

Offset: 0xd4

Bits	Name	Memory Access	Description
31:4	RSVD_REG_TIMEOUT_RST	R	<p>Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
3:0	REG_TIMEOUT_RST	R/W	<p>This field holds reset value of REG_TIMEOUT counter register.</p> <p>Value After Reset: 0x8</p> <p>Volatile: true</p>

12.4.5 Programming Example

12.4.5.1 Transmission Flow

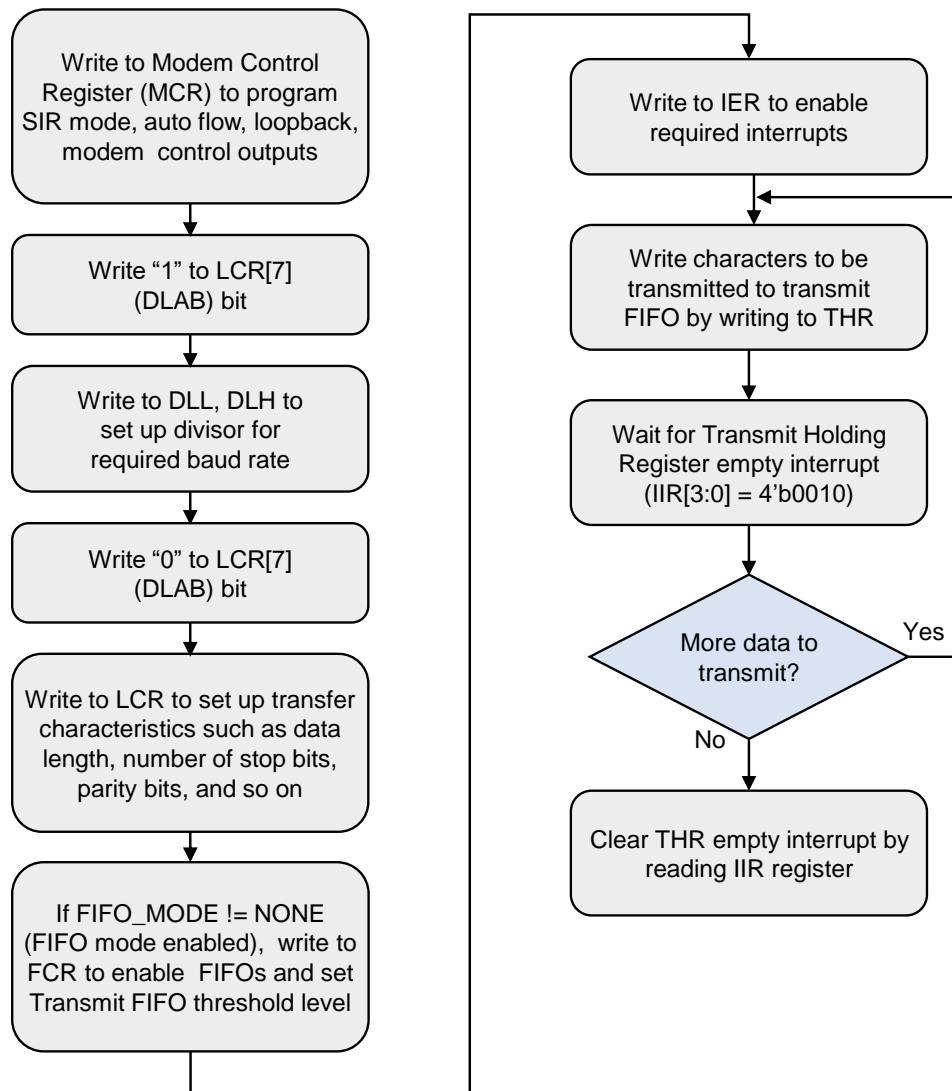


Figure 12-20 Flowchart for uart Transmit Programming Example

12.4.5.2 Receive Flow

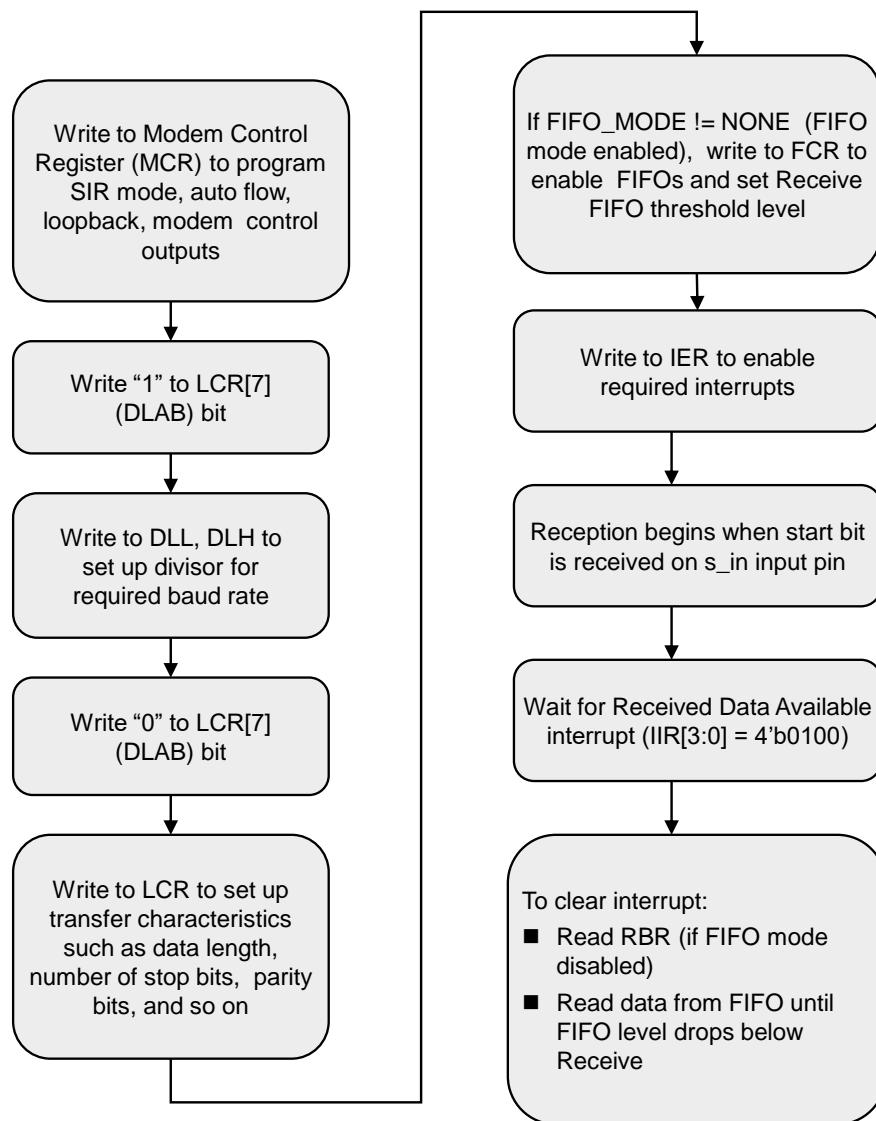


Figure 12-21 Flowchart for uart Receive Programming Example

12.5 SPI

12.5.1 Overview

The following features are supported:

- Support APB4.0, bus data bit width 32
- Configurable and programmable Dual/Quad SPI support in Master Mode
- Support standard SPI transmission protocol
- Support DMA mode
- An high validity interrupt
- Configurable features:
 - Serial interface operation – Choice of Motorola SPI or Texas Instruments Synchronous Serial Protocol.
 - Clock bit-rate – Dynamic control of the serial bit rate of the data transfer; used only in serial-master mode of operation.
 - Data Item size (4 to 32 bits) – Item size of each data transfer under the control of the programmer.
- FIFO depth 32

12.5.2 Block Diagram

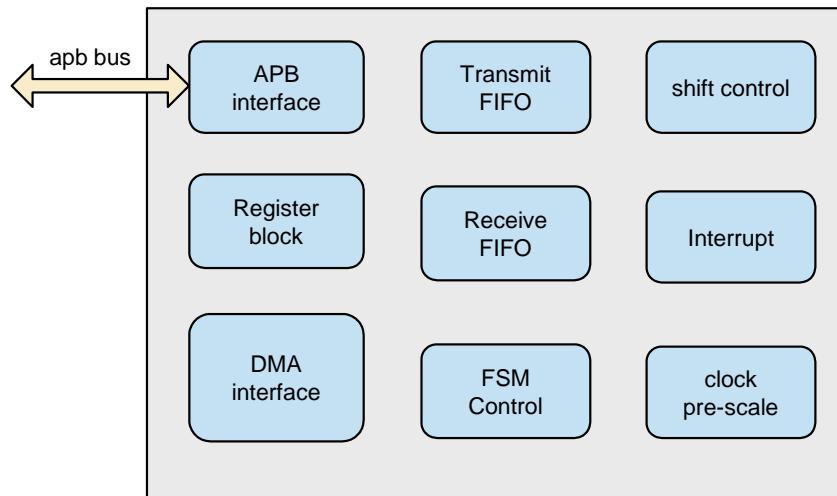


Figure 12-22 SPI block diagram

12.5.3 Function Description

12.5.3.1 Transfer Modes

Transmit and Receive

When $\text{TMOD} = 2'b00$, both transmit and receive logic are valid. The data transfer occurs as normal according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame.

Transmit Only

When $\text{TMOD} = 2'b01$, the receive data are invalid and should not be stored in the receive FIFO. The data transfer occurs as normal, according to the selected frame format (serial protocol). Transmit data are popped from the transmit FIFO and sent through the txd line to the target device, which replies with data on the rxd line. At the end of the data frame, the receive shift register does not load its newly received data into the receive FIFO. The data in the receive shift register is overwritten by the next transfer. You should mask interrupts originating from the receive logic when this mode is entered.

Receive Only

When $\text{TMOD} = 2'b10$, the transmit data are invalid. When configured as a slave, the transmit FIFO is never popped in Receive Only mode. The txd output remains at a constant logic level during the transmission. The data transfer occurs as normal according to the selected frame format (serial protocol). The receive data from the target device is moved from the receive shift register into the receive FIFO at the end of each data frame. You should mask interrupts originating from the transmit logic when this mode is entered.

EEPROM Read

When $\text{TMOD} = 2'b11$, the transmit data is used to transmit an opcode and/or an address to the EEPROM device. Typically, this requires three data frames (8-bit opcode followed by an 8-bit upper address and 8-bit lower address). During the transmission of opcodes and addresses, the receive logic does not capture any data (as long as the SPI host transmits data on its txd line, the data on the rxd line is ignored). The SPI master continues to transmit data until the sending FIFO is empty. Therefore, you should only have enough data frames in the transmit FIFO to provide opcodes and addresses to the EEPROM. If there are more data frames in the sending FIFO than needed, the read data is lost.

When the transmit FIFO becomes empty (all control information has been sent), the data on the receive

line (rxd) is valid and stored in the receive FIFO, and the TXD output remains at a constant logic level. The txd output is held at a constant logic level. The serial transfer continues until the number of data frames received by the SPI master matches the value of the NDF field in the CTRLR1 register + 1.

12.5.3.2 Compatible Interfaces

12.5.3.2.1 Motorola Serial Peripheral Interface (SPI)

With the SPI, the clock polarity (SCPOL) configuration parameter determines whether the inactive state of the serial clock is high or low. To transmit data, both SPI peripherals must have identical serial clock phase (SCPH) and clock polarity (SCPOL) values. The data frame can be 4 to 32-bits in length.

When the configuration parameter SCPH = 0, data transmission begins on the falling edge of the slave select signal.

(1) Serial Format Continuous Transfers (SCPH = 0) when CTRLR0. SSTE = 1

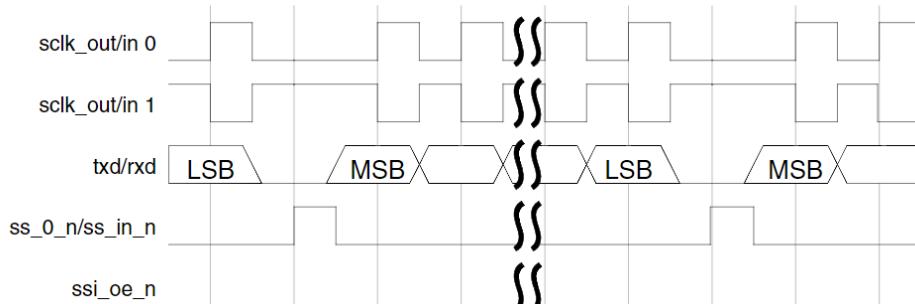


Figure 12-23 Motorola Serial Peripheral Interface(SCPH = 0, SSTE = 1)

(2) Serial Format Continuous Transfers (SCPH=0) when CTRLR0. SSTE = 0

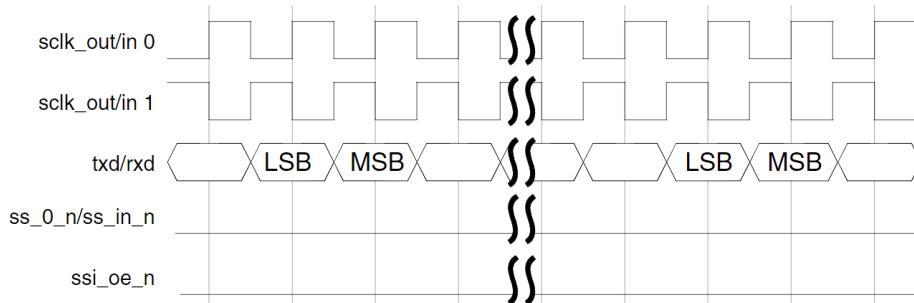


Figure 12-24 Motorola Serial Peripheral Interface(SCPH = 0, SSTE = 0)

When the configuration parameter SCPH = 1, both master and slave peripherals begin transmitting data on the first serial clock edge after the slave select line is activated. The first data bit is captured on the second (trailing) serial clock edge. Data are propagated by the master and slave peripherals on the leading edge of the serial clock. During continuous data frame transfers, the slave select line may be held active-low until the last bit of the last frame has been captured.

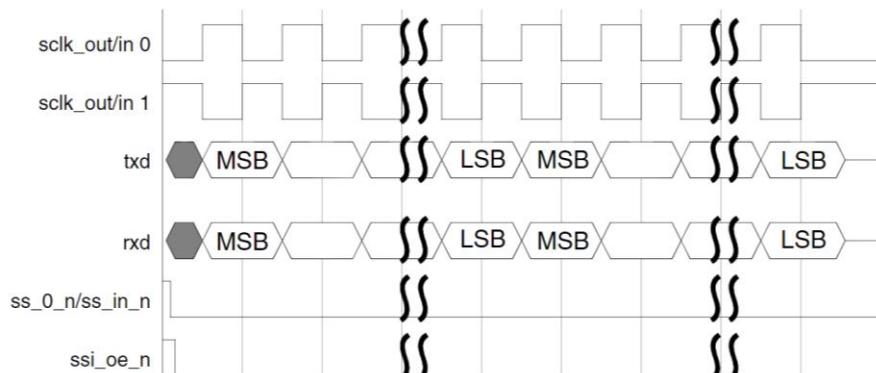


Figure 12-25 SPI Serial Format Continuous Transfer (SCPH = 1)

12.5.3.2.2 National Semiconductor Microwire

When the SPI is configured as a serial master, data transmission begins with the falling edge of the slave-select signal (ss_0_n). One-half serial clock (sclk_out) period later, the first bit of the control word is sent out on the txd line. The length of the control word can be in the range 1 to 16 bits and is set by writing bit field CFS (bits 15:12) in CTRLR0. The remainder of the control word is transmitted (propagated on the falling edge of sclk_out) by the SPI serial master. During this transmission, no data are present (high impedance) on the serial master's rxd line.

The direction of the data word is controlled by the MDD bit field (bit 1) in the Microwire Control Register (MWCR). When MDD=0, this indicates that the SPI serial master receives data from the external serial slave. One clock cycle after the LSB of the control word is transmitted, the slave peripheral responds with a dummy 0 bit, followed by the data frame, which can be 4 to 32-bits in length. Data are propagated on the falling edge of the serial clock and captured on the rising edge.

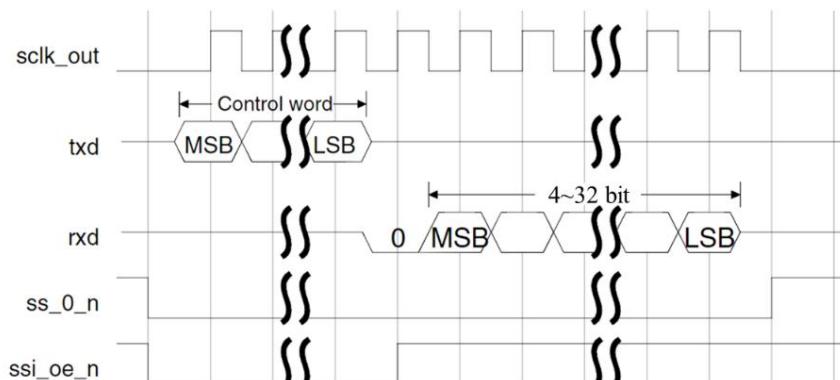


Figure 12-26 Single Master Microwire Serial Transfer (MDD=0)

12.5.3.2.3 Enhanced SPI Modes

You can choose the dual/quad modes of SPI using the SSI_SPI_MODE(SSI_SPI_MODE=2) configuration parameter. The possible values for this parameter are Standard, Dual SPI, Quad SPI modes. When dual, quad mode is selected for this parameter, the width of txd, rxd and ssi_oe_n signals change to 2, 4, or 8, respectively. Hence, the data is shifted out/in on more than one line, increasing the overall throughput. All four combinations of the serial clock's polarity and phase are valid in this mode and it works same as in normal SPI mode. The mode of operation (write/read) can be selected using the CTRLR0.TMOD field.

The following register fields are used for a write operation:

- CTRLR0.SPI_FRF - Specifies the format in which the transmission happens for the frame.
- SPI_CTRLR0 (Control Register 0 register) – Specifies length of instruction, address, and data.
- SPI_CTRLR0.INST_L – Specifies length of an instruction (possible values for an instruction are 0, 4, 8, or 16 bits.)
- SPI_CTRLR0.ADDR_L – Specifies address length

- CTRLR0.DFS or CTRLR0.DFS_32 – Specifies data length.

The instruction, address and data can be programmed to send in dual/quad mode, which can be selected from the SPI_CTRLR0.TRANS_TYPE and CTRLR0.SPI_FRF fields. As shown below, Instruction transmitted in standard and address transmitted in Enhanced SPI format.

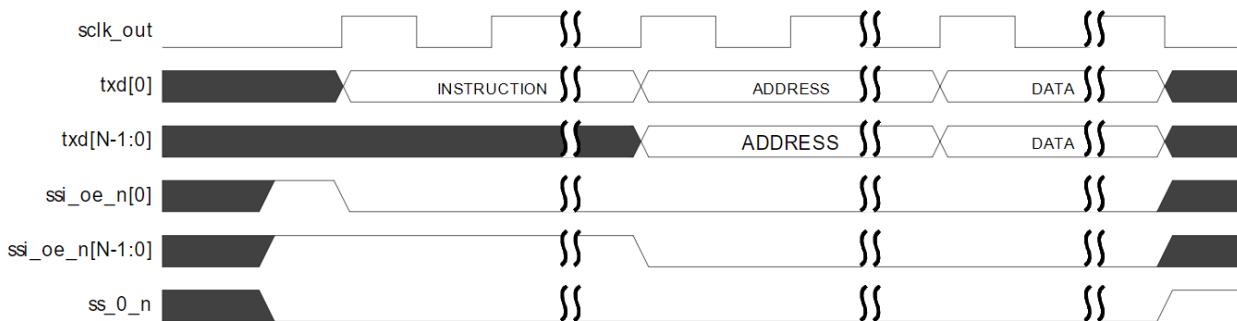


Figure 12-27 Instruction Transmitted in Standard and Address Transmitted in Enhanced SPI Format

For Reading Operation, wait Cycles can be programmed using SPI_CTRLR0.WAIT_CYCLES field. The value programmed into SPI_CTRLR0.WAIT_CYCLES is mapped directly to sclk_out times. For example, WAIT_CYCLES=0 indicates no Wait, WAIT_CYCLES=1, indicates 1 wait cycle and so on. The wait cycles are introduced for target slave to change their mode from input to output and the wait cycles can vary for different devices.

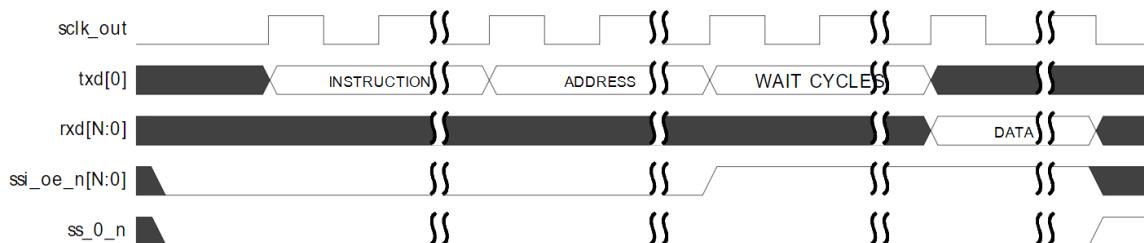


Figure 12-28 Typical Read Operation in Enhanced SPI Mode

12.5.3.2.4 RXD Sample Delay

Due to the data delay, the data from slave may not reach the host within one clock cycle, and the sampling delay needs to be set. Specifically, set the RX_SAMPLE_DLY to increment from 0 until the data is received normally, and record the value n set at this time, and continue to increase the RX_SAMPLE_DLY until the data cannot be received, and record the delay value m set at this time. Then the RX_SAMPLE_DLY is round((m+n-1)/2) which is the optimal delay value that can receive data stably.

12.5.3.3 Interrupts

SPI supports combined interrupt requests, and each interrupt request can be masked. The merged interrupt request is the result of all other SPI interrupts after masking. All SPI interrupts are active-high level interrupts.

The SPI interrupts are described as follows:

- Transmit FIFO Empty Interrupt (ssi_txe_intr) – Set when the transmit FIFO is equal to or below its threshold value and requires service to prevent an under-run. The threshold value, set through a software-programmable register, determines the level of transmit FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data are written into the transmit FIFO buffer, bringing it over the threshold level.
- Transmit FIFO Overflow Interrupt (ssi_txo_intr) – Set when an APB access attempts to write into the transmit FIFO after it has been completely filled. When set, data written from the APB is discarded. This interrupt remains set until you read the transmit FIFO overflow interrupt clear

register (TXOICR).

- Receive FIFO Full Interrupt (ssi_rxf_intr) – Set when the receive FIFO is equal to or above its threshold value plus 1 and requires service to prevent an overflow. The threshold value, set through a software-programmable register, determines the level of receive FIFO entries at which an interrupt is generated. This interrupt is cleared by hardware when data are read from the receive FIFO buffer, bringing it below the threshold level.
- Receive FIFO Overflow Interrupt (ssi_rxo_intr) – Set when the receive logic attempts to place data into the receive FIFO after it has been completely filled. When set, newly received data are discarded. This interrupt remains set until you read the receive FIFO overflow interrupt clear register (RXOICR).
- Receive FIFO Underflow Interrupt (ssi_rxu_intr) – Set when an APB access attempts to read from the receive FIFO when it is empty. When set, zeros are read back from the receive FIFO. This interrupt remains set until you read the receive FIFO underflow interrupt clear register (RXUICR).

12.5.3.4 SCLK Clock Ratios

Output frequency:

$$F_{sclk_out} = \frac{F_{ssi_clk}}{SCKDV}$$

ssi_clk is the apb clock

sclk_out is the output clock, with a maximum support of 50MHz

12.5.4 Register Description

12.5.4.1 Register Overview

Register	Offset	Description
CTRLR0	0x0	Control Register 0
CTRLR1	0x4	Control Register 1
SSIENR	0x8	SSI Enable Register
MWCR	0xc	Microwire Control Register
SER	0x10	Slave Enable Register
BAUDR	0x14	Baud Rate Select
TXFTLR	0x18	Transmit FIFO Threshold Level
RXFTLR	0x1c	Receive FIFO Threshold Level
TXFLR	0x20	Transmit FIFO Level Register
RXFLR	0x24	Receive FIFO Level Register
SR	0x28	Status Register
IMR	0x2c	Interrupt Mask Register
ISR	0x30	Interrupt Status Register
RISR	0x34	Raw Interrupt Status Register

Register	Offset	Description
TXOICR	0x38	Transmit FIFO Overflow Interrupt Clear Register
RXOICR	0x3c	Receive FIFO Overflow Interrupt Clear Register
RXUICR	0x40	Receive FIFO Underflow Interrupt Clear Register
MSTICR	0x44	Multi-Master Interrupt Clear Register
ICR	0x48	Interrupt Clear Register
DMACR	0x4c	DMA Control Register
DMATDLR	0x50	DMA Transmit Data Level
DMARDLR	0x54	DMA Receive Data Level
IDR	0x58	Identification Register
SSI_VERSION_ID	0x5c	coreKit version ID Register
DR0	0x60	Data Register x
DR1	0x64	Data Register x
DR2	0x68	Data Register x
DR3	0x6c	Data Register x
DR4	0x70	Data Register x
DR5	0x74	Data Register x
DR6	0x78	Data Register x
DR7	0x7c	Data Register x
DR8	0x80	Data Register x
DR9	0x84	Data Register x
DR10	0x88	Data Register x
DR11	0x8c	Data Register x
DR12	0x90	Data Register x
DR13	0x94	Data Register x
DR14	0x98	Data Register x
DR15	0x9c	Data Register x
DR16	0xa0	Data Register x

Register	Offset	Description
DR17	0xa4	Data Register x
DR18	0xa8	Data Register x
DR19	0xac	Data Register x
DR20	0xb0	Data Register x
DR21	0xb4	Data Register x
DR22	0xb8	Data Register x
DR23	0xbc	Data Register x
DR24	0xc0	Data Register x
DR25	0xc4	Data Register x
DR26	0xc8	Data Register x
DR27	0xcc	Data Register x
DR28	0xd0	Data Register x
DR29	0xd4	Data Register x
DR30	0xd8	Data Register x
DR31	0xdc	Data Register x
DR32	0xe0	Data Register x
DR33	0xe4	Data Register x
DR34	0xe8	Data Register x
DR35	0xec	Data Register x
RX_SAMPLE_DLY	0xf0	RX Sample Delay Register
SPI_CTRLR0	0xf4	SPI Control Register
RSVD	0xfc	RSVD - Reserved address location

12.5.4.2 Register Detail Description

CTRLR0

Name: Control Register 0

Description: This register controls the serial data transfer. It is impossible to write to this register when the SPI is enabled. The SPI is enabled and disabled by writing to the SSIENR register.

Reset Value: SSI_CTRLR0_RST

Size: 32 bits

Offset: 0x0

Bits	Name	Memory Access	Description
31:26	RSVD_CTRLR0	R	SSTE Reserved bits - Read Only Value After Reset: 0x0
25	SECONV	R/W	Set the Endianness for XIP and data register reads. Values: 0x0 (DISABLED): Endian Conversion disabled. 0x1 (ENABLED): Endian Conversion enabled. Value After Reset: 0x0
24	SSTE	R/W	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 will stay low and sclk will run continuously for the duration of the transfer. Note: This register is only valid when SSI_SCPH0_SSTOGGLE is set to 1. Value After Reset: 0x1
23	RSVD_CTRLR0_23	R	CTRLR0_23 Reserved bits - Read Only Value After Reset: 0x0
22:21	SPI_FRF	R/W	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

Bits	Name	Memory Access	Description
20:16	DFS_32	R/W	<p>Data Frame Size in 32-bit transfer size mode. Used to select the data frame size in 32-bit transfer mode. These bits are only 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.</p> <p>Note: When SSI_SPI_MODE is either set to "Dual" or "Quad" or "Octal" mode and SPI_FRF is not set to 2'b00.</p> <p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 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

Bits	Name	Memory Access	Description
			<p>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: 0x7</p>
15:12	CFS	R/W	<p>Control Frame Size. Selects the length of the control word for the Microwire frame format.</p> <p>Values:</p> <ul style="list-style-type: none"> 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 <p>Value After Reset: 0x0</p>
11	SRL	R/W	<p>Shift Register Loop.</p> <p>Used for testing purposes only. When internally active, connects the transmit shift register output to the receive shift register input.</p>

Bits	Name	Memory Access	Description
			<p>Can be used in both serial-slave and serial-master modes.</p> <p>When the SPI 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</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (TESTING_MODE): Test mode: Tx & Rx shift reg connected 0x0 (NORMAL_MODE): Normal mode operation <p>Value After Reset: 0x0</p>
10	SLV_OE	R	<p>Slave Output Enable. Relevant only when the SPI 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 SPI 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.</p> <p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (DISABLED): Slave Output is disabled 0x0 (ENABLED): Slave Output is enabled <p>Value After Reset: 0x0</p>
9:8	TMOD	R/W	<p>Transfer Mode.</p> <p>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.</p> <p>In transmit-only mode, data received from the external device is not valid and is not stored in the receive FIFO memory; it is overwritten on the next transfer.</p> <p>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.</p> <p>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.</p> <p>In eeprom-read mode, receive data is not valid while control data</p>

Bits	Name	Memory Access	Description
			<p>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 SPI is configured as master device.</p> <p>00 - Transmit & Receive 01 - Transmit Only 10 - Receive Only 11 - EEPROM Read</p> <p>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:</p> <p>10 - Read 01 - Write</p> <p>Values:</p> <p>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</p> <p>Value After Reset: 0x0</p>
7	SCPOL	R/W	<p>Serial Clock Polarity.</p> <p>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 SPI master is not actively transferring data on the serial bus.</p> <p>Values:</p> <p>0x0 (SCLK_LOW): Inactive state of serial clock is low 0x1 (SCLK_HIGH): Inactive state of serial clock is high</p> <p>Value After Reset: 0x0</p>
6	SCPH	R/W	<p>Serial Clock Phase.</p> <p>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.</p> <p>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.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (SCPH_MIDDLE): Serial clock toggles in middle of first data bit</p> <p>0x1 (SCPH_START): Serial clock toggles at start of first data bit</p> <p>Value After Reset: 0x0</p>
5:4	FRF	R/W	<p>Frame Format. Selects which serial protocol transfers the data.</p> <p>Values:</p> <ul style="list-style-type: none"> 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 <p>Value After Reset: 0x0</p>
3:0	DFS	R	<p>Data Frame Size.</p> <p>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.</p> <p>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.</p> <p>You must right-justify transmit data before writing into the transmit FIFO. The transmit logic ignores the upper unused bits when transmitting the data.</p> <p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 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

Bits	Name	Memory Access	Description
			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: 0x0

CTRLR1

Name: Control Register 1

Description: This register exists only when the SPI is configured as a master device. When the SPI 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 SPI is enabled. The SPI is enabled and disabled by writing to the SSIENR register.

Reset Value: 0x0

Size: 32 bits

Offset: 0x4

Bits	Name	Memory Access	Description
31:16	RSVD_CTRLR1	R	CTRLR1 Reserved bits - Read Only Value After Reset: 0x0
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 SPI. The SPI 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 SPI 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 SPI is configured as a serial slave. Value After Reset: 0x0

SSIENR

Name: SSI Enable Register

Description: This register enables and disables the SPI.

Reset Value: 0x0

Size: 32 bits**Offset:** 0x8

Bits	Name	Memory Access	Description
31:1	RSVD_SSIENR	R	<p>SSIENR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
0	SSI_EN	R/W	<p>SSI Enable.</p> <p>Enables and disables all SPI 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 SPI control registers when enabled. When disabled, the <code>ssi_sleep</code> output is set (after delay) to inform the system that it is safe to remove the <code>ssi_clk</code>, thus saving power consumption in the system.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Disables Serial Transfer 0x1 (ENABLED): Enables Serial Transfer <p>Value After Reset: 0x0</p>

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 SPI is enabled. The SPI is enabled and disabled by writing to the SSIENR register.

Reset Value: 0x0**Size:** 32 bits**Offset:** 0xc

Bits	Name	Memory Access	Description
31:3	RSVD_MWCR	R	<p>MWCR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
2	MHS	R/W	<p>Microwire Handshaking.</p> <p>Relevant only when the SPI 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 SPI 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.</p> <p>Values:</p>

Bits	Name	Memory Access	Description
			<p>0x0 (DISABLE): Handshaking interface is disabled 0x1 (ENABLED): Handshaking interface is enabled Value After Reset: 0x0</p>
1	MDD	R/W	<p>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 SPI MacroCell from the external serial device. When this bit is set to 1, the data word is transmitted from the SPI MacroCell to the external serial device. Values: 0x0 (RECEIVE): SSI receives data 0x1 (TRANSMIT): SSI transmits data Value After Reset: 0x0</p>
0	MWMOD	R/W	<p>Microwire Transfer Mode. Defines whether the Microwire transfer is sequential or non-sequential. When sequential mode is used, only one control word is needed to transmit or receive a block of data words. When non-sequential mode is used, there must be a control word for each data word that is transmitted or received. Values: 0x0 (NON_SEQUENTIAL): Non-Sequential Microwire Transfer 0x1 (SEQUENTIAL): Sequential Microwire Transfer Value After Reset: 0x0</p>

SER

Name: Slave Enable Register

Description: This register is valid only when the SPI is configured as a master device. When the SPI 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 SPI master. Up to 16 slave-select output pins are available on the SPI 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

Bits	Name	Memory	Description

		Access	
31:2	RSVD_SER	R	SER Reserved bits - Read Only Value After Reset: 0x0
1:0	SER	R/W	Slave Select Enable Flag. Each bit in this register corresponds to a slave select line (ss_x_n) from the SPI 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

BAUDR**Name:** Baud Rate Select

Description: This register is valid only when the SPI is configured as a master device. When the SPI 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 SPI is enabled. The SPI is enabled and disabled by writing to the SSIENR register.

Reset Value: 0x0**Size:** 32 bits**Offset:** 0x14

Bits	Name	Memory Access	Description
31:16	RSVD_BAUDR	R	BAUDR Reserved bits - Read Only Value After Reset: 0x0
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: $F_{sclk_out} = F_{ssi_clk}/SCKDV$ 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 Value After Reset: 0x0
--	--	--	---

TXFTLR**Name:** Transmit FIFO Threshold Level**Description:** This register controls the threshold value for the transmit FIFO memory. The SPI is enabled and disabled by writing to the SSIENR register.**Reset Value:** 0x0**Size:** 32 bits**Offset:** 0x18

Bits	Name	Memory Access	Description
31:5	RSVD_TXFTLR	R	TXFTLR Reserved bits - Read Only Value After Reset: 0x0
4:0	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 SPI Databook. ssi_txe_intr is asserted when TFT or less data entries are present in transmit FIFO Value After Reset: 0x0

RXFTLR**Name:** Receive FIFO Threshold Level**Description:** This register controls the threshold value for the receive FIFO memory. The SPI is enabled and disabled by writing to the SSIENR register.**Reset Value:** 0x0**Size:** 32 bits**Offset:** 0x1c

Bits	Name	Memory Access	Description
------	------	---------------	-------------

31:5	RSVD_RXFTLR	R	RXFTLR Reserved bits - Read Only Value After Reset: 0x0
4:0	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 SPI Databook. <code>ssi_rxf_intr</code> is asserted when RFT or more data entries are present in receive FIFO. Value After Reset: 0x0

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

Bits	Name	Memory Access	Description
31:6	RSVD_TXFLR	R	TXFLR Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
5:0	TXTFL	R	Transmit FIFO Level. Contains the number of valid data entries in the transmit FIFO. Value After Reset: 0x0 Volatile: true

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

Bits	Name	Memory Access	Description
31:6	RSVD_RXFLR	R	<p>RXFLR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
5:0	RXTFL	R	<p>Receive FIFO Level.</p> <p>Contains the number of valid data entries in the receive FIFO.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

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

Bits	Name	Memory Access	Description
31:7	RSVD_SR	R	<p>SR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
6	DCOL	R	<p>Data Collision Error.</p> <p>Relevant only when the SPI is configured as a master device. This bit will be set if ss_in_n input is asserted by other master, when the SPI master is in the middle of the transfer. This informs the processor that the last data transfer was halted before completion. This bit is cleared when read.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NO_ERROR_CONDITION): No Error 0x1 (TX_COLLISION_ERROR): Transmit Data Collision Error <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
5	RSVD_TXE	R	<p>TXE Reserved field - read-only</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
			Volatile: true
4	RFF	R	<p>Receive FIFO Full.</p> <p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NOT_FULL): Receive FIFO is not full 0x1 (FULL): Receive FIFO is full <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
3	RFNE	R	<p>Receive FIFO Not Empty.</p> <p>Set when the receive FIFO contains one or more entries and is cleared when the receive FIFO is empty. This bit can be polled by software to completely empty the receive FIFO.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (EMPTY): Receive FIFO is empty 0x1 (NOT_EMPTY): Receive FIFO is not empty <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
2	TFE	R	<p>Transmit FIFO Empty.</p> <p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NOT_EMPTY): Transmit FIFO is not empty 0x1 (EMPTY): Transmit FIFO is empty <p>Value After Reset: 0x1</p> <p>Volatile: true</p>
1	TFNF	R	<p>Transmit FIFO Not Full.</p> <p>Set when the transmit FIFO contains one or more empty locations, and is cleared when the FIFO is full.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (FULL): Transmit FIFO is full 0x1 (NOT_FULL): Transmit FIFO is not Full

Bits	Name	Memory Access	Description
			<p>Value After Reset: 0x1</p> <p>Volatile: true</p>
0	BUSY	R	<p>SSI Busy Flag.</p> <p>When set, indicates that a serial transfer is in progress; when cleared indicates that the SPI is idle or disabled.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): SPI is idle or disabled 0x1 (ACTIVE): SPI is actively transferring data <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

IMR**Name:** Interrupt Mask Register

Description: This read/write register masks or enables all interrupts generated by the SPI. When the SPI 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

Bits	Name	Memory Access	Description
31:6	RSVD_IMR	R	<p>IMR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
5	MSTIM	R/W	<p>Multi-Master Contention Interrupt Mask. This bit field is not present if the SPI is configured as a serial-slave device.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (MASKED): ssi_mst_intr interrupt is masked 0x1 (UNMASKED): ssi_mst_intr interrupt is not masked <p>Value After Reset: 0x1</p>
4	RXFIM	R/W	<p>Receive FIFO Full Interrupt Mask</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (MASKED): ssi_rxf_intr interrupt is masked 0x1 (UNMASKED): ssi_rxf_intr interrupt is not masked

Bits	Name	Memory Access	Description
			Value After Reset: 0x1
3	RXOIM	R/W	<p>Receive FIFO Overflow Interrupt Mask</p> <p>Values:</p> <p>0x0 (MASKED): ssi_rxo_intr interrupt is masked</p> <p>0x1 (UNMASKED): ssi_rxo_intr interrupt is not masked</p> <p>Value After Reset: 0x1</p>
2	RXUIM	R/W	<p>Receive FIFO Underflow Interrupt Mask</p> <p>Values:</p> <p>0x0 (MASKED): ssi_rxu_intr interrupt is masked</p> <p>0x1 (UNMASKED): ssi_rxu_intr interrupt is not masked</p> <p>Value After Reset: 0x1</p>
1	TXOIM	R/W	<p>Transmit FIFO Overflow Interrupt Mask</p> <p>Values:</p> <p>0x0 (MASKED): ssi_txo_intr interrupt is masked</p> <p>0x1 (UNMASKED): ssi_txo_intr interrupt is not masked</p> <p>Value After Reset: 0x1</p>
0	TXEIM	R/W	<p>Transmit FIFO Empty Interrupt Mask</p> <p>Values:</p> <p>0x0 (MASKED): ssi_txe_intr interrupt is masked</p> <p>0x1 (UNMASKED): ssi_txe_intr interrupt is not masked</p> <p>Value After Reset: 0x1</p>

ISR

Name: Interrupt Status Register

Description: This register reports the status of the SPI interrupts after they have been masked.

Reset Value: 0x0

Size: 32 bits

Offset: 0x30

Bits	Name	Memory Access	Description
31:6	RSVD_ISR	R	ISR Reserved bits - Read Only

Bits	Name	Memory Access	Description
			<p>Value After Reset: 0x0</p> <p>Volatile: true</p>
5	MSTIS	R	<p>Multi-Master Contention Interrupt Status. This bit field is not present if the SPI is configured as a serial-slave device.</p> <p>Values:</p> <p>0x0 (INACTIVE): ssi_mst_intr interrupt not active after masking</p> <p>0x1 (ACTIVE): ssi_mst_intr interrupt is active after masking</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
4	RXFIS	R	<p>Receive FIFO Full Interrupt Status</p> <p>Values:</p> <p>0x0 (INACTIVE): ssi_rxf_intr interrupt is not active after masking</p> <p>0x1 (ACTIVE): ssi_rxf_intr interrupt is full after masking</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
3	RXOIS	R	<p>Receive FIFO Overflow Interrupt Status</p> <p>Values:</p> <p>0x0 (INACTIVE): ssi_rxo_intr interrupt is not active after masking</p> <p>0x1 (ACTIVE): ssi_rxo_intr interrupt is active after masking</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
2	RXUIS	R	<p>Receive FIFO Underflow Interrupt Status</p> <p>Values:</p> <p>0x0 (INACTIVE): ssi_rxu_intr interrupt is not active after masking</p> <p>0x1 (ACTIVE): ssi_rxu_intr interrupt is active after masking</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
1	TXOIS	R	<p>Transmit FIFO Overflow Interrupt Status</p> <p>Values:</p> <p>0x0 (INACTIVE): ssi_txo_intr interrupt is not active after masking</p> <p>0x1 (ACTIVE): ssi_txo_intr interrupt is active after masking</p>

Bits	Name	Memory Access	Description
			Value After Reset: 0x0 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 Volatile: true

RISR**Name:** Raw Interrupt Status Register**Description:** This read-only register reports the status of the SPI interrupts prior to masking.**Reset Value:** 0x0**Size:** 32 bits**Offset:** 0x34

Bits	Name	Memory Access	Description
31:6	RSVD_RISR	R	RISR Reserved bits - Read Only Value After Reset: 0x0 Volatile: true
5	MSTIR	R	Multi-Master Contention Raw Interrupt Status. This bit field is not present if the SPI 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 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

Bits	Name	Memory Access	Description
			Value After Reset: 0x0 Volatile: true
3	RXOIR	R	Receive FIFO Overflow Raw Interrupt Status Values: 0x1 (ACTIVE): ssi_rxo_intr interrupt is not active prior to masking 0x0 (INACTIVE): ssi_rxo_intr interrupt is active prior masking Value After Reset: 0x0 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 Volatile: true
1	TXOIR	R	Transmit FIFO Overflow Raw Interrupt Status Values: 0x0 (INACTIVE): ssi_txo_intr interrupt is not active prior to masking 0x1 (ACTIVE): ssi_txo_intr interrupt is active prior masking Value After Reset: 0x0 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 masking Value After Reset: 0x0 Volatile: true

TXOICR

Name: Transmit FIFO Overflow Interrupt Clear Register

Description: Transmit FIFO Overflow Interrupt Clear Register.

Reset Value: 0x0

Size: 32 bits**Offset:** 0x38

Bits	Name	Memory Access	Description
31:1	RSVD_TXOICR	R	<p>TXOICR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	TXOICR	R	<p>Clear Transmit FIFO Overflow Interrupt.</p> <p>This register reflects the status of the interrupt. A read from this register clears the ssi_txo_intr interrupt; writing has no effect.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

RXOICR

Name: Receive FIFO Overflow Interrupt Clear Register**Description:** Receive FIFO Overflow Interrupt Clear Register.**Reset Value:** 0x0**Size:** 32 bits**Offset:** 0x3c

Bits	Name	Memory Access	Description
31:1	RSVD_RXOICR	R	<p>RXOICR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	RXOICR	R	<p>Clear Receive FIFO Overflow Interrupt.</p> <p>This register reflects the status of the interrupt. A read from this register clears the ssi_rxo_intr interrupt; writing has no effect.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

RXUICR

Name: Receive FIFO Underflow Interrupt Clear Register**Description:** Receive FIFO Underflow Interrupt Clear Register.**Reset Value:** 0x0**Size:** 32 bits

Offset: 0x40

Bits	Name	Memory Access	Description
31:1	RSVD_RXUICR	R	<p>RXUICR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	RXUICR	R	<p>Clear Receive FIFO Underflow Interrupt.</p> <p>This register reflects the status of the interrupt. A read from this register clears the ssi_rxu_intr interrupt; writing has no effect.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

MSTICR**Name:** Multi-Master Interrupt Clear Register**Description:** Multi-Master Interrupt Clear Register.**Reset Value:** 0x0**Size:** 32 bits**Offset:** 0x44

Bits	Name	Memory Access	Description
31:1	RSVD_MSTICR	R	<p>MSTICR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	MSTICR	R	<p>Clear Multi-Master Contention Interrupt.</p> <p>This register reflects the status of the interrupt. A read from this register clears the ssi_mst_intr interrupt; writing has no effect.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

ICR**Name:** Interrupt Clear Register**Description:** Interrupt Clear Register.**Reset Value:** 0x0**Size:** 32 bits**Offset:** 0x48

Bits	Name	Memory Access	Description
31:1	RSVD_ICR	R	<p>ICR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	ICR	R	<p>Clear Interrupts.</p> <p>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.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

DMACR

Name: DMA Control Register

Description: This register is only valid when SPI is configured with a set of DMA Controller interface signals (SSI_HAS_DMA = 1). When SPI 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

Bits	Name	Memory Access	Description
31:2	RSVD_DMACR	R	<p>DMACR Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
1	TDMAE	R/W	<p>Transmit DMA Enable.</p> <p>This bit enables/disables the transmit FIFO DMA channel.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Transmit DMA disabled 0x1 (ENABLED): Transmit DMA enabled <p>Value After Reset: 0x0</p>
0	RDMAE	R/W	<p>Receive DMA Enable.</p> <p>This bit enables/disables the receive FIFO DMA channel</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLE): Receive DMA disabled 0x1 (ENABLED): Receive DMA enabled

Bits	Name	Memory Access	Description
			Value After Reset: 0x0

DMATDLR**Name:** DMA Transmit Data Level

Description: This register is only valid when the SPI is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When SPI 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

Bits	Name	Memory Access	Description
31:5	RSVD_DMATDLR	R	DMATDLR Reserved bits - Read Only Value After Reset: 0x0
4:0	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 SPI Databook. dma_tx_req is asserted when DMATDL or less data entries are present in the transmit FIFO Value After Reset: 0x0

DMARDLR**Name:** DMA Receive Data Level

Description: This register is only valid when SPI is configured with a set of DMA interface signals (SSI_HAS_DMA = 1). When SPI 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

Bits	Name	Memory Access	Description
31:5	RSVD_DMARDLR	R	DMARDLR Reserved bits - Read Only

Bits	Name	Memory Access	Description
			Value After Reset: 0x0
4:0	DMARDL	R/W	<p>Receive Data Level.</p> <p>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 SPI Databook.</p> <p>dma_rx_req is asserted when DMARDL or more valid data entries are present in the receive FIFO.</p> <p>Value After Reset: 0x0</p>

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

Bits	Name	Memory Access	Description
31:0	IDCODE	R	<p>Identification code.</p> <p>The register contains the peripheral's identification code, which is written into the register at configuration time using CoreConsultant.</p> <p>Value After Reset: 0xffffffff</p>

SSI_VERSION_ID**Name:** coreKit version ID Register**Description:** This read-only register stores the specific SPI component version.**Reset Value:** SSI_VERSION_ID**Size:** 32 bits**Offset:** 0x5c

Bits	Name	Memory Access	Description
31:0	SSI_COMP_VERSION	R	Contains the hex representation of the ESWIN component

Bits	Name	Memory Access	Description
			<p>version. Consists of ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01*.</p> <p>Value After Reset: 0x3430332a</p>

DRx(for x = 0; x <= 35)**Name:** Data Register x

Description: The SPI data register is a 32-bit 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. 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 SPI 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 SPI are not addressable.

Reset Value: 0x0**Size:** 32 bits**Offset:** 0x60 + 0x04*x

Bits	Name	Memory Access	Description
31:0	DR	R/W	<p>Data Register. When writing to this register, you must right-justify the data. Read data are automatically right-justified. Read = Receive FIFO buffer Write = Transmit FIFO buffer.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

RX_SAMPLE_DLY**Name:** RX Sample Delay Register

Description: This register is only valid when the SPI is configured with rxd sample delay logic (SSI_HAS_RX_SAMPLE_DELAY==1). When the SPI 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 SPI is enabled. The SPI is enabled and disabled by writing to the SSIENR register.

Reset Value: 0x0**Size:** 32 bits**Offset:** 0xf0

Bits	Name	Memory Access	Description
31:8	RSVD_RX_SAMPLE_DLY	R	SAMPLE_DLY Reserved bits - Read Only Value After Reset: 0x0
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

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 SPI is enabled (SSI_EN=1). The SPI is enabled and disabled by writing to the SSIENR register.

Reset Value: 0x00000200**Size:** 32 bits**Offset:** 0xf4

Bits	Name	Memory Access	Description
31:19	RSVD_SPI_CTRLR0	R	SPI_CTRLR0 Reserved bits - Read Only Value After Reset: 0x0
18	SPI_RXDS_EN	R	Read data strobe enable bit. Once this bit is set to 1 SPI will use Read data strobe (rxds) to capture read data in DDR mode. Value After Reset: 0x0
17	INST_DDR_EN	R	Instruction DDR Enable bit. This will enable Dual-data rate transfer for Instruction phase. Value After Reset: 0x0
16	SPI_DDR_EN	R	SPI DDR Enable bit. This will enable Dual-data rate transfers in

Bits	Name	Memory Access	Description
			<p>Dual/Quad/Octal frame formats of SPI.</p> <p>Value After Reset: 0x0</p>
15:11	WAIT_CYCLES	R/W	<p>Wait cycles</p> <p>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 SPI Databook.</p> <p>Value After Reset: 0x0</p>
10	RSVD_SPI_CTRLR0_10	R	<p>CTRLR0_10 Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
9:8	INST_L	R/W	<p>Instruction Length</p> <p>Dual/Quad/Octal mode instruction length in bits.</p> <p>Values:</p> <ul style="list-style-type: none"> 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 <p>Value After Reset: 0x2</p>
7:6	RSVD_SPI_CTRLR0_6_7	R	<p>CTRLR0_6_7 Reserved bits - Read Only</p> <p>Value After Reset: 0x0</p>
5:2	ADDR_L	R/W	<p>Address Length.</p> <p>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_L decode value, see "Read Operation in Enhanced SPI Modes" section in the SPI Databook.</p> <p>Values:</p> <ul style="list-style-type: none"> 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

Bits	Name	Memory Access	Description
			<p>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</p> <p>Value After Reset: 0x0</p>
1:0	TRANS_TYPE	R/W	<p>Address and instruction transfer format.</p> <p>Selects whether SPI 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.</p> <p>01 - Instruction will be sent in Standard SPI Mode and Address will be sent in the mode specified by CTRLR0.SPI_FRF.</p> <p>10 - Both Instruction and Address will be sent in the mode specified by SPI_FRF. 11 - Reserved.</p> <p>Value After Reset: 0x0</p>

RSVD

Name: RSVD - Reserved address location

Description: RSVD - Reserved address location.

Size: 32 bits

Offset: 0xfc

Bits	Name	Memory Access	Description
31:0	RSVD	R	<p>RSVD 31to0 Reserved address location</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

12.5.5 Programming example

12.5.5.1 Master SPI Transfer Flow

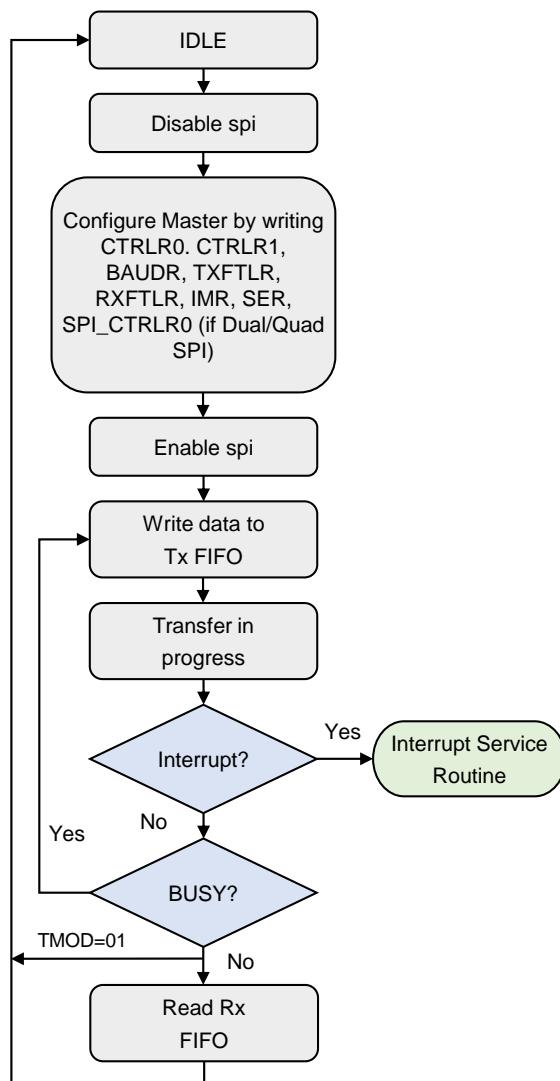


Figure 12-29 Master SPI Transfer Flow

12.5.5.2 Master Microwire Transfer Flow

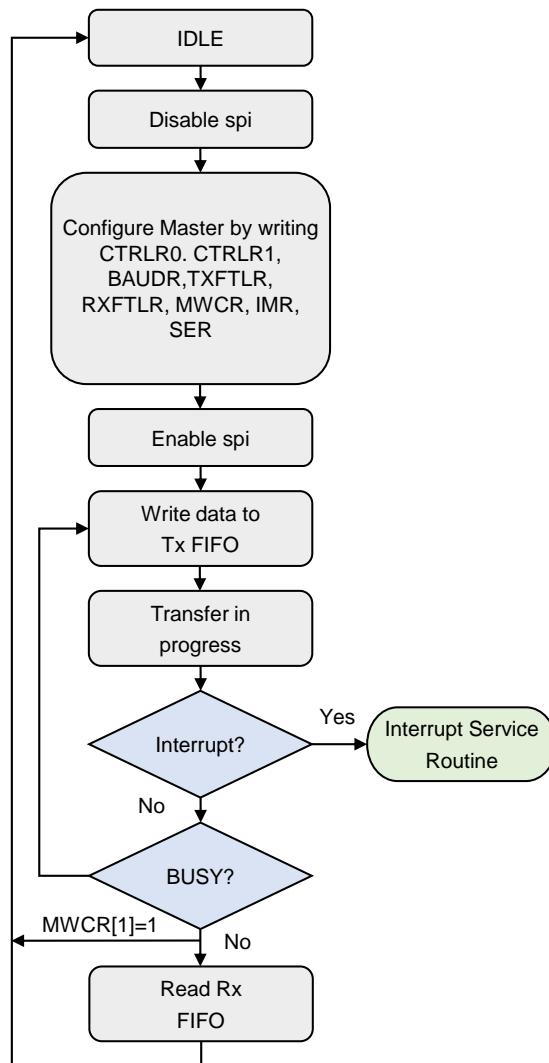


Figure 12-30 Master Microwire Transfer Flow

12.5.6 Timing Sequence

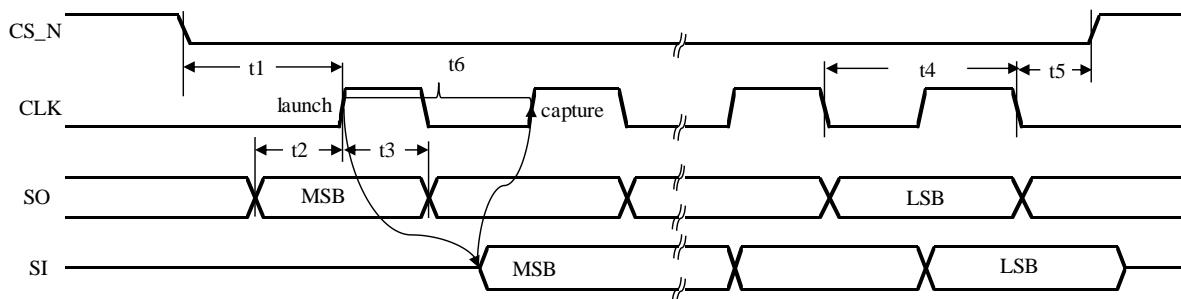


Figure 12-31 SPI timing sequence

12.6 GPIO

12.6.1 Overview

GPIO is a programmable, general-purpose I/O peripheral. This component is an APB slave. The GPIO controls the output data and direction of the external I/O. It can also use memory-mapped registers to read back external data.

The following features are supported:

- 112 independently configurable signals
- A/B/C/D port. This 4 ports map to 112 GPIO
- Each signal has its own data register and data direction register
- Configurable interrupt mode for port A
- Debounce logic can be configured to debounce interrupt the signal
- Configurable synchronization of interrupt signals

12.6.2 Block Diagram

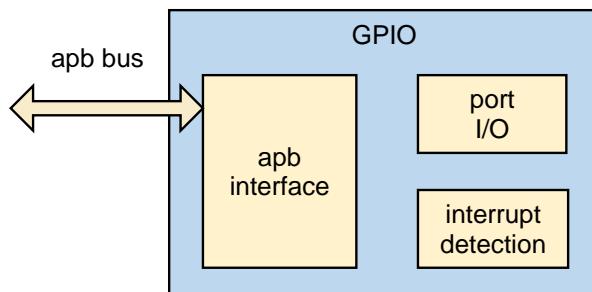


Figure 12-32 GPIO block diagram

12.6.3 Function Description

12.6.3.1 Software Control

The data and direction control of the signal come from the data register (`gpio_swportx_dr`) and the output enable register (`gpio_swportx_ddr`), respectively, where x is A, B, C, or D. When used as input, the IE corresponding to the IO must be enable, and the external input value is read out by the register `gpio_ext_portx`.

12.6.3.2 Interrupt Function(PortA)

GPIO 0~31:

- Active-high and level
- Active-low and level
- Rising edge
- Falling edge

For edge-detected interrupts, you can clear the interrupt by writing a 1 to the `gpio_porta_eoi` register for the corresponding bit to disable the interrupt.

12.6.3.3 Debounce

When the porta is working in interrupt mode, it can be configured whether or not to debounce to remove any glitch that is less than one cycle of the debounce clock.

12.6.4 Register Description

12.6.4.1 Register Overview

Register	Offset	Description
GPIO_SWPORTA_DR	0x0	Port A data register
GPIO_SWPORTA_DDR	0x4	Port A Data Direction Register
GPIO_INTEN	0x30	Interrupt enable register
GPIO_INTMASK	0x34	Interrupt mask register
GPIO_INTTYPE_LEVEL	0x38	Interrupt level
GPIO_INT_POLARITY	0x3c	Interrupt polarity
GPIO_INTSTATUS	0x40	Interrupt status
GPIO_RAW_INTSTATUS	0x44	Raw interrupt status
GPIO_DEBOUNCE	0x48	Debounce enable register
GPIO_PORTA_EOI	0x4c	Port A clear interrupt register
GPIO_EXT_PORTA	0x50	External port A register
GPIO_LS_SYNC	0x60	Synchronization level
GPIO_ID_CODE	0x64	GPIO ID code
GPIO_VER_ID_CODE	0x6c	GPIO Component Version
GPIO_CONFIG_REG2	0x70	GPIO Configuration Register 2
GPIO_CONFIG_REG1	0x74	GPIO Configuration Register 1

12.6.4.2 Register Detail Description

GPIO_SWPORTA_DR

Name: Port A data register

Description: Port A data register

Size: 32 bits

Offset: 0x0

Bits	Name	Memory Access	Description
31:0	GPIO_SWPORTA_DR	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

Bits	Name	Memory Access	Description
			<p>set to Software mode. The value read back is equal to the last value written to this register.</p> <p>Value After Reset: 0x0</p>

GPIO_SWPORTA_DDR**Name:** Port A Data Direction Register**Description:** Port A Data Direction Register**Size:** 32 bits**Offset:** 0x4

Bits	Name	Memory Access	Description
31:0	GPIO_SWPORTA_DDR	R/W	<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (IN): Input Direction 0x1 (OUT): Output Direction <p>Value After Reset: 0x0</p>

GPIO_INTEN**Name:** Interrupt enable register**Description:** Interrupt enable register

Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).

Size: 32 bits**Offset:** 0x30

Bits	Name	Memory Access	Description

31:0	GPIO_INTEN	R/W	<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Interrupt is disabled 0x1 (ENABLED): Interrupt is enabled <p>Value After Reset: 0x0</p>
------	------------	-----	--

GPIO_INTMASK

Name: Interrupt mask register

Description: Interrupt mask register

Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).

Size: 32 bits

Offset: 0x34

Bits	Name	Memory Access	Description
31:0	GPIO_INTMASK	R/W	<p>Controls whether an interrupt on Port A can create an interrupt for the interrupt controller by not masking it. By default, all interrupt 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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Interrupt bits are unmasked 0x1 (ENABLED): Mask interrupt <p>Value After Reset: 0x0</p>

GPIO_INTPOLARITY

Name: Interrupt polarity

Description: Interrupt level

Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).

Size: 32 bits

Offset: 0x38

Bits	Name	Memory Access	Description
31:0	GPIO_INTTYPE_LEVEL	R/W	<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (LEVEL_SENSITIVE): Interrupt is level sensitive 0x1 (EDGE_SENSITIVE): Interrupt is edge sensitive <p>Value After Reset: 0x0</p>

GPIO_INT_POLARITY

Name: Interrupt polarity

Description: Interrupt polarity

Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).

Size: 32 bits

Offset: 0x3c

Bits	Name	Memory Access	Description
31:0	GPIO_INT_POLARITY	R/W	<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (ACTIVE_LOW): Active Low polarity

Bits	Name	Memory Access	Description
			0x1 (ACTIVE_HIGH): Active High polarity Value After Reset: 0x0

GPIO_INTSTATUS**Name:** Interrupt status**Description:** Interrupt status

Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).

Size: 32 bits**Offset:** 0x40

Bits	Name	Memory Access	Description
31:0	GPIO_INTSTATUS	R	Interrupt status of Port A. Values: 0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active Value After Reset: 0x0 Volatile: true

GPIO_RAW_INTSTATUS**Name:** Raw interrupt status**Description:** Raw interrupt status

Note: This register is available only if Port A is configured to generate interrupts (GPIO_PORTA_INTR = Include (1)).

Size: 32 bits**Offset:** 0x44

Bits	Name	Memory Access	Description
31:0	GPIO_RAW_INTSTATUS	R	<p>Raw interrupt of status of Port A (premasking bits)</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (INACTIVE): Inactive 0x1 (ACTIVE): Active <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

GPIO_DEBOUNCE

Name: Debounce enable register

Description: Debounce enable register

Note: 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)).

Size: 32 bits

Offset: 0x48

Bits	Name	Memory Access	Description
31:0	GPIO_DEBOUNCE	R/W	<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): No debounce 0x1 (ENABLED): Enable debounce <p>Value After Reset: 0x0</p>

GPIO_PORTA_EOI

Name: Port A clear interrupt register

Description: Port A clear interrupt register

Note: 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)).

Size: 32 bits

Offset: 0x4c

Bits	Name	Memory Access	Description
31:0	GPIO_PORTA_EOI	W	<p>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.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): No interrupt clear 0x1 (ENABLED): Clear Interrupt <p>Value After Reset: 0x0</p>

GPIO_EXT_PORTA

Name: External port A register

Description: External port A register

Size: 32 bits

Offset: 0x50

Bits	Name	Memory Access	Description
31:0	GPIO_EXT_PORTA	R	<p>This register always reflects the signals value on the External Port A.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

GPIO_LS_SYNC

Name: Synchronization level

Description: Synchronization level

Size: 32 bits

Offset: 0x60

Bits	Name	Memory Access	Description
31:1	RSVD_GPIO_LS_SYNC	R	<p>RSVD_GPIO_LS_SYNC Reserved bits - read as zero</p> <p>Value After Reset: 0x0</p>
0	GPIO_LS_SYNC	R/W	<p>Writing a 1 to this register results in all level-sensitive interrupts being synchronized to pclk_intr.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): No synchronization to pclk_int (default) 0x1 (ENABLED): Synchronize to pclk_intr <p>Value After Reset: 0x0</p>

GPIO_ID_CODE**Name:** GPIO ID code**Description:** GPIO ID code**Size:** 32 bits**Offset:** 0x64

Bits	Name	Memory Access	Description
31:0	GPIO_ID_CODE	R	<p>This is a user-specified code that a system can read. It can be used for chip identification, and so on.</p> <p>Value After Reset: 0x0</p>

GPIO_VER_ID_CODE**Name:** GPIO Component Version**Description:** GPIO Component Version**Size:** 32 bits**Offset:** 0x6c

Bits	Name	Memory Access	Description
31:0	GPIO_VER_ID_CODE	R	<p>ASCII value for each number in the version, followed by *. For example 32_31_32_2A represents the version 2.12*.</p> <p>Value After Reset: 0x3231342a</p>

GPIO_CONFIG_REG2**Name:** GPIO Configuration Register 2**Description:** GPIO Configuration Register 2

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.

Size: 32 bits**Offset:** 0x70

Bits	Name	Memory Access	Description
31:20	RSVD_GPIO_CONFIG_REG2	R	<p>RSVD_GPIO_CONFIG_REG2 Reserved bits - read as zero</p> <p>Value After Reset: 0x0</p>
19:15	ENCODED_ID_PWIDTH_D	R	<p>The value of this register is equal to GPIO_PWIDTH_D-1.</p> <p>Value After Reset: 0x7</p>
14:10	ENCODED_ID_PWIDTH_C	R	<p>The value of this register is equal to GPIO_PWIDTH_C-1.</p> <p>Value After Reset: 0x7</p>
9:5	ENCODED_ID_PWIDTH_B	R	<p>The value of this register is equal to GPIO_PWIDTH_B-1.</p> <p>Value After Reset: 0x7</p>
4:0	ENCODED_ID_PWIDTH_A	R	<p>The value of this register is equal to GPIO_PWIDTH_A-1.</p> <p>Value After Reset: 0x1f</p>

GPIO_CONFIG_REG1**Name:** GPIO Configuration Register 1**Description:** GPIO Configuration Register 1

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).

Size: 32 bits**Offset:** 0x74

Bits	Name	Memory Access	Description
31:22	RSVD_GPIO_CONFIG_REG1	R	<p>RSVD_GPIO_CONFIG_REG1 Reserved bits - read as zero</p> <p>Value After Reset: 0x0</p>
21	INTERRUPT_BOTH_EDGE_TYPE	R	<p>The value of this register is derived from the GPIO_INT_BOTH_EDGE configuration parameter</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Interrupt generation on rising or falling edge 0x1 (ENABLED): Interrupt generation on both rising and falling edge <p>Value After Reset: 0x0</p>
20:16	ENCODED_ID_WIDTH	R	<p>The value of this register is derived from the GPIO_ID_WIDTH configuration parameter.</p> <p>Value After Reset: 0x1f</p>
15	GPIO_ID	R	<p>The value of this register is derived from the GPIO_ID configuration parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): GPIO_ID not included 0x1 (ENABLED): GPIO_ID is included <p>Value After Reset: 0x1</p>
14	ADD_ENCODED_PARAMS	R	<p>The value of this register is derived from the GPIO_ADD_ENCODED_PARAMS configuration parameter.</p>

Bits	Name	Memory Access	Description
			<p>Values:</p> <p>0x0 (DISABLED): Encoded parameters not added</p> <p>0x1 (ENABLED): Encoded parameters added</p> <p>Value After Reset: 0x1</p>
13	DEBOUNCE	R	<p>The value of this register is derived from the GPIO_DEBOUNCE configuration parameter.</p> <p>Values:</p> <p>0x0 (DISABLED): Exclude debounce capability</p> <p>0x1 (ENABLED): Include debounce capability</p> <p>Value After Reset: 0x1</p>
12	PORTA_INTR	R	<p>The value of this register is derived from the GPIO_PORTA_INTR configuration parameter.</p> <p>Values:</p> <p>0x0 (DISABLED): PORT A is not used as an interrupt source</p> <p>0x1 (ENABLED): PORT A is required to be used as an interrupt source</p> <p>Value After Reset: 0x1</p>
11	HW_PORTD	R	<p>The value of this register is derived from the GPIO_HW_PORTD configuration parameter.</p> <p>Values:</p> <p>0x0 (DISABLED): Port D has external, auxiliary hardware signals excluded</p> <p>0x1 (ENABLED): Port D has external, auxiliary hardware signals included</p> <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
10	HW_PORTC	R	<p>The value of this register is derived from the GPIO_HW_PORTC configuration parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Port C has external, auxiliary hardware signals excluded 0x1 (ENABLED): Port C has external, auxiliary hardware signals included <p>Value After Reset: 0x0</p>
9	HW_PORTB	R	<p>The value of this register is derived from the GPIO_HW_PORTB configuration parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Port B has external, auxiliary hardware signals excluded 0x1 (ENABLED): Port B has external, auxiliary hardware signals included <p>Value After Reset: 0x0</p>
8	HW_PORTA	R	<p>The value of this register is derived from the GPIO_HW_PORTA configuration parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): Port A has external, auxiliary hardware signals excluded 0x1 (ENABLED): Port A has external, auxiliary hardware signals included <p>Value After Reset: 0x0</p>

Bits	Name	Memory Access	Description
7	PORTD_SINGLE_CTL	R	<p>The value of this register is derived from the GPIO_PORTD_SINGLE_CTL configuration parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): PORTD is not controlled from a single source 0x1 (ENABLED): PORTD is controlled from a single source <p>Value After Reset: 0x1</p>
6	PORTC_SINGLE_CTL	R	<p>The value of this register is derived from the GPIO_PORTC_SINGLE_CTL configuration parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): PORTC is not controlled from a single source 0x1 (ENABLED): PORTC is controlled from a single source <p>Value After Reset: 0x1</p>
5	PORTB_SINGLE_CTL	R	<p>The value of this register is derived from the GPIO_PORTB_SINGLE_CTL configuration parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): PORTB is not controlled from a single source 0x1 (ENABLED): PORTB is controlled from a single source <p>Value After Reset: 0x1</p>
4	PORTA_SINGLE_CTL	R	<p>The value of this register is derived from the GPIO_PORTA_SINGLE_CTL configuration parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (DISABLED): PORTA is not controlled from a single source

Bits	Name	Memory Access	Description
			<p>0x1 (ENABLED): PORTA is controlled from a single source</p> <p>Value After Reset: 0x1</p>
3:2	NUM_PORTS	R	<p>The value of this register is derived from the GPIO_NUM_PORT configuration parameter.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (NUM_PORTS_1): Number of ports is 1 0x1 (NUM_PORTS_2): Number of ports is 2 0x2 (NUM_PORTS_3): Number of ports is 3 0x3 (NUM_PORTS_4): Number of ports is 4 <p>Value After Reset: 0x0</p>
1:0	APB_DATA_WIDTH	R	<p>The value of this register is derived from the GPIO_APB_DATA_WIDTH configuration parameter.</p> <p>Note: 0x3 = Reserved</p> <p>Values:</p> <ul style="list-style-type: none"> 0x0 (APB_8BITS): APB DATA WIDTH is 8 bits 0x1 (APB_16BITS): APB DATA WIDTH is 16 bits 0x2 (APB_32BITS): APB DATA WIDTH is 32 bits <p>Value After Reset: 0x2</p>

12.6.5 Programming example

GPIO Features:

1. Select the corresponding pad
2. Disable PortA interrupt function
3. Assert Output Enable(gpio_swportx_ddr)
4. Configure the level of the output signal(gpio_swportx_dr)

12.7 JTAG

12.7.1 JTAG Chain

There are three groups of JTAG interfaces:

- JTAG0 chain: MCPU -> LPCPU -> NPU -> reserved, no JTAG reset
- JTAG1 chain: SCPU, no JTAG reset
- JTAG2 chain: DSP

JTAG bypass control is controlled by register `jtag_chain_ctrl`, and 0~4bit controls LPCPU, MCPU, NPU, reserved and DSP respectively.

12.7.2 Timing Sequence

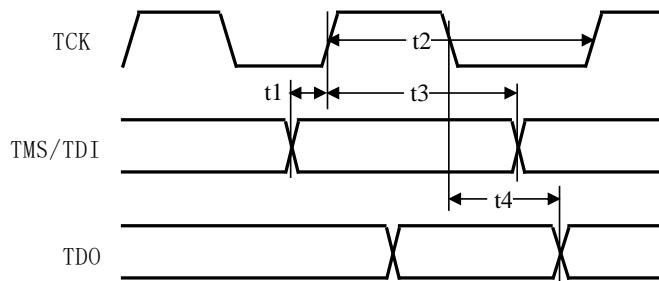


Figure 12-33 JTAG timing sequence

12.8 PWM/TIMER

12.8.1 Overview

TITIME and PWM have the same characteristics. Unless otherwise specified, the feature is both.

The following features are supported:

- APB4.0 bus interface, 32-bit data bit width
- Internally instantiate N programmable timers
 - 3 timers are instantiated inside the PWM module
 - 8 timers are instantiated inside the TIMER module
- Support two modes of operation:
 - Free-running
 - User-defined
- Each timer generates a separate high active interrupt signal
- The duty cycle of the output signal is 0%~100% adjustable
- The TIMER module supports the pause function, but the PWM module does not support the pause function

12.8.2 Block Diagram

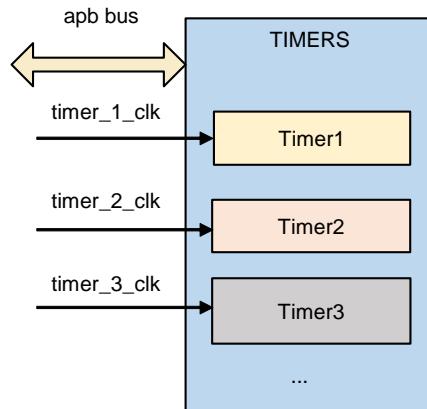


Figure 12-34 PWM/TIMER block diagram

12.8.3 Function Description

12.8.3.1 The Working Mode

When the timer is enabled after being reset or disable, the count value is loaded from the TimerNLoadCount register; this happens in both Free-running and User-defined count modes.

When a timer counts down to 0, it loads one of two values, depending on the timer operating mode:

- User-defined count mode – Timer loads the current value of the TimerNLoadCount register. Use this mode if you want a fixed, timed interrupt. Designate this mode by writing a "1" to bit 1 of TimerNControlReg.
- Free-running mode – Timer loads the maximum value, which is dependent on the timer width; that is, the TimerNLoadCount register is comprised of $2^{\text{TIMER_WIDTH_N}-1}$ bits, all of which are loaded with 1s. The timer counter wrapping to its maximum value allows time to reprogram or disable the timer before another interrupt occurs. Use this mode if you want a single timed interrupt. Designate this mode by writing a "0" to bit 1 of TimerNControlReg.

12.8.3.2 Duty Cycle Calculation

Each timer can be pulse-width modulated, when register bit TimerNControlReg[4] (TIMER_PWM bit) is set to 1, the HIGH and LOW periods of the toggle outputs can be controlled separately by programming the TimerNLoadCount2 and TimerNLoadCount registers.

The pulse widths of the toggle outputs are controlled as follows:

- Width of timer_N_toggle HIGH period = (TimerNLoadCount2 + 1) * timer_N_clk clock period
- Width of timer_N_toggle LOW period = (TimerNLoadCount + 1) * timer_N_clk clock period

12.8.3.3 0%~100% Duty Cycle

PWM/TIMER supports the programming for 0% and 100% duty cycle pulse width modulation of toggle outputs (timer_N_toggle) through the TimerNLoadCount and TimerNLoadCount2 registers, when 0% and 100% duty cycle mode is enable. You can enable the duty cycle mode either by setting the TimerNControlReg [4] register.

- 0% duty cycle
 - TimerNLoadCount = Do not care
 - TimerNLoadCount2 = 0
- 100% duty cycle
 - TimerNLoadCount = 0
 - TimerNLoadCount2 = Do not care
- Other duty cycle – When 0% and 100% duty cycle mode is enabled (with timer PWM mode and the user-defined count mode is enabled), the definition of the toggle high and low period changes as follows for a duty cycle other than 0% or 100%:

- Width of timer_N_toggle HIGH period = TimerNLoadCount2 * timer_N_clk clock period
- Width of timer_N_toggle LOW period = TimerNLoadCount * timer_N_clk clock period

12.8.3.4 Interrupt

When the counter reaches 0, an interrupt is generated. Interrupts can be cleared and masked by register.

12.8.3.5 Pause Function (Supported By TIMER Only)

The TIMER pause function is controlled by the system control register timerN_pause, 0~7bit controls the 1~8 internal timer of TIMERN respectively.

12.8.4 Input Clock

timer8 of TIMER3 is controlled by a timer3_clk8 clock; The timerN_clk of TIMER0, TIMER1, TIMER2, and TIMER3 (except timer8) are gated by timer_clk[N], and the timerN_clk of PWM is the same as timer_pclk.

12.8.5 Register Description

12.8.5.1 Register Overview

Register	Offset	Description
TimerNLoadCount	0x0 + (N-1)*0x14	Timer N Load Count Register
TlmerNCurrentVal	0x4 + (N-1)*0x14	Current value of Timer N
TimerNControlReg	0x8 + (N-1)*0x14	Timer N Control Register
TimerNEOI	0xc + (N-1)*0x14	Timer N End-of-Interrupt Register
TimerNIntStatus	0x10 + (N-1)*0x14	Timer N Interrupt Status Register
TimersIntStatus	0xa0	Timers Interrupt Status Register
TimersEOI	0xa4	Timers End-of-Interrupt Register
TimersRawIntStatus	0xa8	Timers Raw Interrupt Status Register
TIMERS_COMP_VERSION	0xac	Timers Component Version
TimerNLoadCount2	0xb0 + (N-1)*0x14	Timer N Load Count2 Register
TIMER_N_PROT_LEVEL L	0xd0 + (N-1)*0x14	Timer_N Protection level

12.8.5.2 Register Detail Description

Note:

PWM: NUM_TIMERS = 3

TIMER: NUM_TIMERS = 8

TimerNLoadCount(for N = 1; N <= NUM_TIMERS)

Name: Timer N Load Count Register

Description: Value to be loaded into Timer N

Size: 32 bits

Offset: $0x0 + (N-1)*0x14$

Bits	Name	Memory Access	Description
31:0	TIMER1LOADCOUNT	R/W	<p>Value to be loaded into Timer 1. This is the value from which counting commences. Any value written to this register is loaded into the associated timer.</p> <p>Note:</p> <p>The field name for Timer 2 will be TIMER2LOADCOUNT.</p> <p>The field name for Timer 3 will be TIMER3LOADCOUNT.</p> <p>The field name for Timer 4 will be TIMER4LOADCOUNT.</p> <p>The field name for Timer 5 will be TIMER5LOADCOUNT.</p> <p>The field name for Timer 6 will be TIMER6LOADCOUNT.</p> <p>The field name for Timer 7 will be TIMER7LOADCOUNT.</p> <p>The field name for Timer 8 will be TIMER8LOADCOUNT.</p> <p>Value After Reset: 0x0</p>

TlmerNCurrentVal (For N = 1; N <= NUM_TIMERS)

Description: Current value of Timer N

Size: 32 bits

Offset: $0x4 + (N-1)*0x14$

Bits	Name	Memory Access	Description
31:0	TIMER1CURRENTVAL	R	<p>Current Value of Timer 1. When TIM_NEWMODE=0, This register is supported only when timer_1_clk is synchronous to pclk. Reading this register when using independent clocks results in an undefined value. When TIM_NEWMODE=1, no restrictions apply.</p> <p>Note:</p>

Bits	Name	Memory Access	Description
			<p>The field name for Timer 2 will be TIMER2CURRENTVAL.</p> <p>The field name for Timer 3 will be TIMER3CURRENTVAL.</p> <p>The field name for Timer 4 will be TIMER4CURRENTVAL.</p> <p>The field name for Timer 5 will be TIMER5CURRENTVAL.</p> <p>The field name for Timer 6 will be TIMER6CURRENTVAL.</p> <p>The field name for Timer 7 will be TIMER7CURRENTVAL.</p> <p>The field name for Timer 8 will be TIMER8CURRENTVAL.</p> <p>Note: TIMERNCURRENTVALUE is synchronized from timer clock domain to the APB clock domain. When TIM_NEWMODE=1, reading register TIMERNCURRENTVALUE immediately after reset returns all zeros. Reading this register after synchronization depth number of clock cycles returns the synchronized value which is TIM_RST_CURRENTVAL_N.</p> <p>Value After Reset: 0xffffffff</p> <p>Volatile: true</p>

TimerNControlReg (for N = 1; N <= NUM_TIMERS)**Name:** Timer N Control Register**Description:** Control Register for Timer N. This register controls enabling, operating mode (free-running or defined-count), and interrupt mask of Timer N. You can program each Timer1ControlReg to enable or disable a specific timer and to control its mode of operation.**Size:** 32 bits**Offset:** 0x8 + (N-1)*0x14

Bits	Name	Memory Access	Description
31:5	RSVD_TimerNControlReg	R	<p>TimerNControlReg 31to5 Reserved field</p> <p>Value After Reset: 0x0</p>
4	TIMER_ON100PWM_EN	((@TIM_ON100_PWM_MODE==1) AND (@TIMER_HAS_TOGGLE_N==1) AND (@TIMER_ON100_PWM_H	Optional. Allows user to enable or disable the usage of Timer 0% and 100% mode feature. This bit is present only when (TIM_ON100_PWM_MODE=1 and

Bits	Name	Memory Access	Description
		C_EN==0)) ? read-write : read-only	<p>TIMER_HAS_TOGGLE_N=1). Otherwise reserved.</p> <p>Values:</p> <p>0x1 (ENABLED): Timer 0% and 100% PWM duty cycle mode is enabled</p> <p>0x0 (DISABLE): Timer 0% and 100% PWM duty cycle mode is disabled</p> <p>Value After Reset: 0x0</p>
3	TIMER_PWM	R/W	<p>Pulse Width Modulation of timer_N_toggle output. This field is only present when TIM_NEWMODE is enabled</p> <p>Values:</p> <p>0x1 (ENABLED): PWM for timer_N_toggle o/p is enabled</p> <p>0x0 (DISABLE): PWM for timer_N_toggle o/p is disabled</p> <p>Value After Reset: 0x0</p>
2	TIMER_INTERRUPT_MASK	R/W	<p>Timer interrupt mask for Timer N.</p> <p>Values:</p> <p>0x1 (MASKED): Timer N interrupt is masked</p> <p>0x0 (UNMASKED): Timer N interrupt is unmasked</p> <p>Value After Reset: 0x0</p>
1	TIMER_MODE	R/W	<p>Timer mode for Timer N.</p> <p>Note: You must set the Timer1LoadCount register to all 1s before enabling the timer in free-running mode.</p> <p>Values:</p> <p>0x1 (USER_DEFINED): User-Defined mode of operation</p>

Bits	Name	Memory Access	Description
			<p>0x0 (FREE_RUNNING): Free Running mode of operation</p> <p>Value After Reset: 0x0</p>
0	TIMER_ENABLE	R/W	<p>Timer enable bit for Timer N.</p> <p>Values:</p> <p>0x1 (ENABLED): Timer N is enabled</p> <p>0x0 (DISABLE): Timer N is disabled</p> <p>Value After Reset: 0x0</p>

TimerNEOI (for N = 1; N <= NUM_TIMERS)**Name:** Timer N End-of-Interrupt Register**Description:** Clears the interrupt from Timer N**Size:** 32 bits**Offset:** 0xc + (N-1)*0x14

Bits	Name	Memory Access	Description
31:1	RSVD_Timer1EOI	R	<p>Timer1EOI 31to1 Reserved field.</p> <p>Note:</p> <p>The reserved field name for Timer 2 will be RSVD_Timer2EOI.</p> <p>The reserved field name for Timer 3 will be RSVD_Timer3EOI.</p> <p>The reserved field name for Timer 4 will be RSVD_Timer4EOI.</p> <p>The reserved field name for Timer 5 will be RSVD_Timer5EOI.</p> <p>The reserved field name for Timer 6 will be RSVD_Timer6EOI.</p> <p>The reserved field name for Timer 7 will be RSVD_Timer7EOI.</p> <p>The reserved field name for Timer 8 will be RSVD_Timer8EOI.</p> <p>Value After Reset: 0x0</p>
0	Timer1EOI	R	Reading from this register returns all zeroes (0) and clears the

Bits	Name	Memory Access	Description
			<p>interrupt from Timer 1.</p> <p>Note:</p> <p>The field name for Timer 2 will be Timer2EOI.</p> <p>The field name for Timer 3 will be Timer3EOI.</p> <p>The field name for Timer 4 will be Timer4EOI.</p> <p>The field name for Timer 5 will be Timer5EOI.</p> <p>The field name for Timer 6 will be Timer6EOI.</p> <p>The field name for Timer 7 will be Timer7EOI.</p> <p>The field name for Timer 8 will be Timer8EOI.</p> <p>Value After Reset: 0x0</p>

TimerNIntStatus (for N = 1; N <= NUM_TIMERS)**Name:** Timer N Interrupt Status Register**Description:** Contains the interrupt status for Timer N**Size:** 32 bits**Offset:** 0x10 + (N-1)*0x14

Bits	Name	Memory Access	Description
31:1	RSVD_TIMER1INTSTAT	R	<p>Timer1IntStatus 31to1 Reserved field</p> <p>Note:</p> <p>The reserved field name for Timer 2 will be RSVD_TIMER2INTSTAT.</p> <p>The reserved field name for Timer 3 will be RSVD_TIMER3INTSTAT.</p> <p>The reserved field name for Timer 4 will be RSVD_TIMER4INTSTAT.</p> <p>The reserved field name for Timer 5 will be RSVD_TIMER5INTSTAT.</p>

Bits	Name	Memory Access	Description
			<p>The reserved field name for Timer 6 will be RSVD_TIMER6INTSTAT.</p> <p>The reserved field name for Timer 7 will be RSVD_TIMER7INTSTAT.</p> <p>The reserved field name for Timer 8 will be RSVD_TIMER8INTSTAT.</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
0	TIMER1INTSTAT	R	<p>Contains the interrupt status for Timer 1.</p> <p>Note:</p> <ul style="list-style-type: none"> The field name for Timer 2 will be TIMER2INTSTAT. The field name for Timer 3 will be TIMER3INTSTAT. The field name for Timer 4 will be TIMER4INTSTAT. The field name for Timer 5 will be TIMER5INTSTAT. The field name for Timer 6 will be TIMER6INTSTAT. The field name for Timer 7 will be TIMER7INTSTAT. The field name for Timer 8 will be TIMER8INTSTAT. <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Timer N Interrupt is active 0x0 (INACTIVE): Timer N Interrupt is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

TimersIntStatus

Name: Timers Interrupt Status Register

Description: Contains the interrupt status of all timers in the component.

Size: 32 bits

Offset: 0xa0

Bits	Name	Memory Access	Description
31:3	RSVD_TimersIntStatus	R	<p>TimersIntStatus 31toNUM_TIMERS Reserved field</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
2:0	TimersIntStatus	R	<p>Contains the interrupt status of all timers in the component. If a bit of this register is 0, then the corresponding timer interrupt is not active and the corresponding interrupt could be on either the timer_intr bus or the timer_intr_n bus, depending on the interrupt polarity you have chosen. Similarly, if a bit of this register is 1, then the corresponding interrupt bit has been set in the relevant interrupt bus. In both cases, the status reported is the status after the interrupt mask has been applied. Reading from this register does not clear any active interrupts.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Timer_intr(_n) is active 0x0 (INACTIVE): Timer_intr(_n) is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

TimersEOI

Name: Timers End-of-Interrupt Register

Description: Returns all zeroes (0) and clears all active interrupts.

Size: 32 bits

Offset: 0xa4

Bits	Name	Memory Access	Description
31:3	RSVD_TIMERSEOI	R	<p>TimersEOI 31toNUM_TIMERS Reserved field</p> <p>Value After Reset: 0x0</p>
2:0	TIMERSEOI	R	<p>Reading this register returns all zeroes (0) and clears all active interrupts.</p> <p>Value After Reset: 0x0</p>

TimersRawIntStatus**Name:** Timers Raw Interrupt Status Register**Description:** Contains the unmasked interrupt status of all timers in the component.**Size:** 32 bits**Offset:** 0xa8

Bits	Name	Memory Access	Description
31:3	RSVD_TIMERSRAWINTSTAT	R	<p>TimersRawIntStatus 31toNUM_TIMERS Reserved field</p> <p>Value After Reset: 0x0</p> <p>Volatile: true</p>
2:0	TIMERSRAWINTSTAT	R	<p>The register contains the unmasked interrupt status of all timers in the component.</p> <p>Values:</p> <ul style="list-style-type: none"> 0x1 (ACTIVE): Raw Timer_intr(_n) is active 0x0 (INACTIVE): Raw Timer_intr(_n) is inactive <p>Value After Reset: 0x0</p> <p>Volatile: true</p>

TIMERS_COMP_VERSION**Name:** Timers Component Version**Description:** Current revision number of the timers component.**Size:** 32 bits**Offset:** 0xac

Bits	Name	Memory Access	Description
31:0	TIMERSCOMPVERSION	R	Current revision number of the timers component. For the value, see the releases table in the AMBA 2 release notes

Bits	Name	Memory Access	Description
			Value After Reset: 0x3231332a

TimerNLoadCount2 (for N = 1; N <= NUM_TIMERS)**Name:** Timer N Load Count2 Register**Description:** Value to be loaded into Timer N when toggle output changes from 0 to 1**Size:** 32 bits**Offset:** 0xb0 + (N-1)*0x14

Bits	Name	Memory Access	Description
31:0	TIMERNLOADCOUNT2	R/W	<p>Value to be loaded into Timer N when timer_N_toggle output changes from 0 to 1. This value determines the width of the HIGH period of the timer_N_toggle output.</p> <p>Value After Reset: 0x0</p>

TIMER_N_PROT_LEVEL (for N = 1; N <= NUM_TIMERS)**Name:** Timer_N Protection level**Description:** Timer_N Protection level register

Read/Write Access:

R/W if HC_PROT_LEVEL=0, else R

Enabling protection on any of its three bits would require a greater or equal privilege on the input PPROT signal to gain access to protected registers of the timer.

Size: 32 bits**Offset:** 0xd0 + (N-1)*0x14

Bits	Name	Memory Access	Description
31:3	RsvdTimer_N_ProtLevel	R	<p>TIMER_N_PROT_LEVEL 31to3 Reserved field- read-only</p> <p>Value After Reset: 0x0</p>
2:0	Timer_N_ProtLevelField	R/W	This field holds protection value of

Bits	Name	Memory Access	Description
			TIMER_N_PROT_LEVEL register. Value After Reset: 0x2

12.8.6 Programming Example

You can use the following programming flow to enable the 0% and 100% duty cycle mode:

1. Disable the timer enable bit in the TimerNControlReg register.
2. Program the TimerNLoadCount and TimerNLoadCount2 registers with appropriate values.
3. Enable the 0% and 100% duty cycle mode bit, the Pulse width modulation bit and set the Timer mode to user-defined count mode in the TimerNControlReg register.
4. Set the timer enable bit in the TimerNControlReg register such that the toggle output is 100% (high) or 0% (low).

When the 0% and 100% duty cycle mode is enabled, internal timer is disabled. The internal timers can be enabled again by switching to Normal toggle output mode or to Pulse width modulation toggle output mode.

12.9 FAN_CTRL

12.9.1 Overview

FAN_CTRL for fan speed monitoring, supporting:

- APB3.0 bus, data bit width 32bit
- Monitor external fan speed

12.9.2 Register description

Reg Name	Offset	Field name	Bit	Access	Description
REG_FAN_INT	0x0	fan_en	[0]	R/W	FAN_CTRL enable
		fan_intr	[1]	RO	FAN_CTRL interrupt. The interrupt is set to 1 when the blade is locked or when the data is detected normally
		Reserved	[2]	RO	Value: 0
		blade_lock	[3]	RO	blade is locked
		Reserved	[31:4]	RO	Value: 0
REG_FAN_RPM	0x04	rpm_cnt	[31:0]	RO	The value represents the number of APB clock cycles in a PWM cycle, and the software calculates the actual rotational speed value

12.9.3 Programming Example

The FAN_CTRL operation process is as follows:

1. Set the REG_FAN_INT [0] value to 32'h1 to enable speed detection
2. Wait for the REG_FAN_INT[1] interrupt

3. Read the REG_FAN_RPM, configurate REG_FAN_INT [2] to 1, and clear the interrupt
4. Repeat the process for the next read