ADM5120

Network Processor

V1.16

COM CPE

Never stop thinking.

The information in this document is subject to change without notice.

Edition 2005-03-30

Published by Infineon Technologies AG,
St.-Martin-Strasse 53,
81669 München, Germany

Attention please!

The information herein is given to describe certain components and shall not be considered as a guarantee of characteristics.

Terms of delivery and rights to technical change reserved.

We hereby disclaim any and all warranties, including but not limited to warranties of non-infringement, regarding circuits, descriptions and charts stated herein.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may only be used in life-support devices or systems with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

Revision History: 2005-03-30, Rev. 1.1

Previous Version:

Page	Subjects (major changes since last revision)

Trademarks

Table of Contents

List of Figures

List of Tables

1 Product Overview

The following chapter gives an overview of the ADM5120.

1.1 Overview

ADM5120 is a high performance, highly integrated, and highly flexible SOC (System-On-Chip) that facilitates the functionalities of SOHO/SME Gateway, NAT Router, Print Server, WLAN Access Point and VPN Gateway. ADM5120 enables the sharing of IP-based broadband services throughout the home/office using wired/wireless computers, entertainment equipment, printers, and other intelligent devices.

Internally, the ADM5120 ASIC consists of a high performance (227 MIPS) embedded MIPS CPU, an embedded switch engine, 10/100M PHY, an embedded PCI bridge, an embedded USB host, and interfaces for UART, SDRAM and Flash. The following diagram illustrates a system configuration that uses the supported functionalities/facilities of ADM5120.

1.2 Features

The following describes the features of the ADM5120.

1.2.1 ASIC Features

Description of ASIC features:

1.2.1.1 Processor

Processor features:

- MIPS 4Kc CPU
- Embedded cache, 8 Kbyte I-cache, 8 Kbyte D-cache
- Embedded memory management unit (MMU) – 32-entry TLB, organized as 16 entry pairs
- 175 MHz/227 MIPS

1.2.1.2 Networking

Networking features:

- 6 ports
- IEEE 802.3 Fast Ethernet
- 5 auto-MDIX (auto-crossover) twisted paired LAN interfaces, embedded 10/100M PHY
- 1 GMII[1)]/MII interface
- Flexible WAN port selection
- Embedded switch engine
- Embedded Data-buffer/Address-look-up table
- Look-up table read/write-able
- MAC layer security
- MAC clone solution
- Multicast grouping (IGMP)
- MAC filtering, Bandwidth control
- Class of Services (CoS) with two priority levels
- Shared dynamic data buffer management, embedded SSRAM
- Port grouping VLAN (overlap-able)
- TCP/IP accelerator

1) Available in BGA only, not PQFP

1.2.1.3 Memory Interface

Memory features:

- SDRAM
- Two bank support (2 chip select pins)
- Each bank can support -- 1M x 32 up to 32M x 32bit (128M-byte)
- Flash
- NAND Flash boot[1)]
- NOR Flash boot: Two bank support (2 chip select pins)
- NOR Flash boot: Each bank can support – 1M x 8-bit, up to 1M x 32-bit (4M-byte)

1.2.1.4 System

- UART interface (support MODEM interface)
- PCI bridge that supports 3 master devices[1)]
- GPIO[1)]
- USB 1.1 host
- Clock source
- 25 MHz crystal for 10/100
- 48 MHz crystal for USB
- 0.18 µ CMOS process
- 1.8 V/3.3 V dual power
- BGA/PQFP

1.2.2 Software Features

Description of software features:

- Linux/Nucleus Real-Time OS
- Linux-based and Nucleus-based turn key support
- Telnet
- IEEE 802.3 Ethernet Driver
- IEEE 802.11 WLAN Driver
- RS232 Driver for Console User Interface
- DHCP Server/Client
- PPP over Ethernet (PPPoE)
- Network Address Translation (NAT) for IP Address Mapping/Sharing/Security
- DNS Proxy
- Simple Network Time Protocol (SNTP)
- Firewall
- Web-Based Configuration: WEB and HTTP
- TFTP upload/download

1.2.3 Typical Applications

The typical applications of the ADM5120 are:

- IEEE 802.3 SOHO/SME Gateway
- NAT Router
- Single band 802.11g Access Point (through PCI bus: 5120+802.11g NIC)
- Multiple band 802.11a/b/g Access Point (through PCI bus: 5120+802.11a/b/g NIC)
- Print Server (through USB1.1)
- 12-port SME Gateway (through GMII: 5120+6999U)

1) PQFP has 4 GPIO pins v.s. BGA has 8 pins.

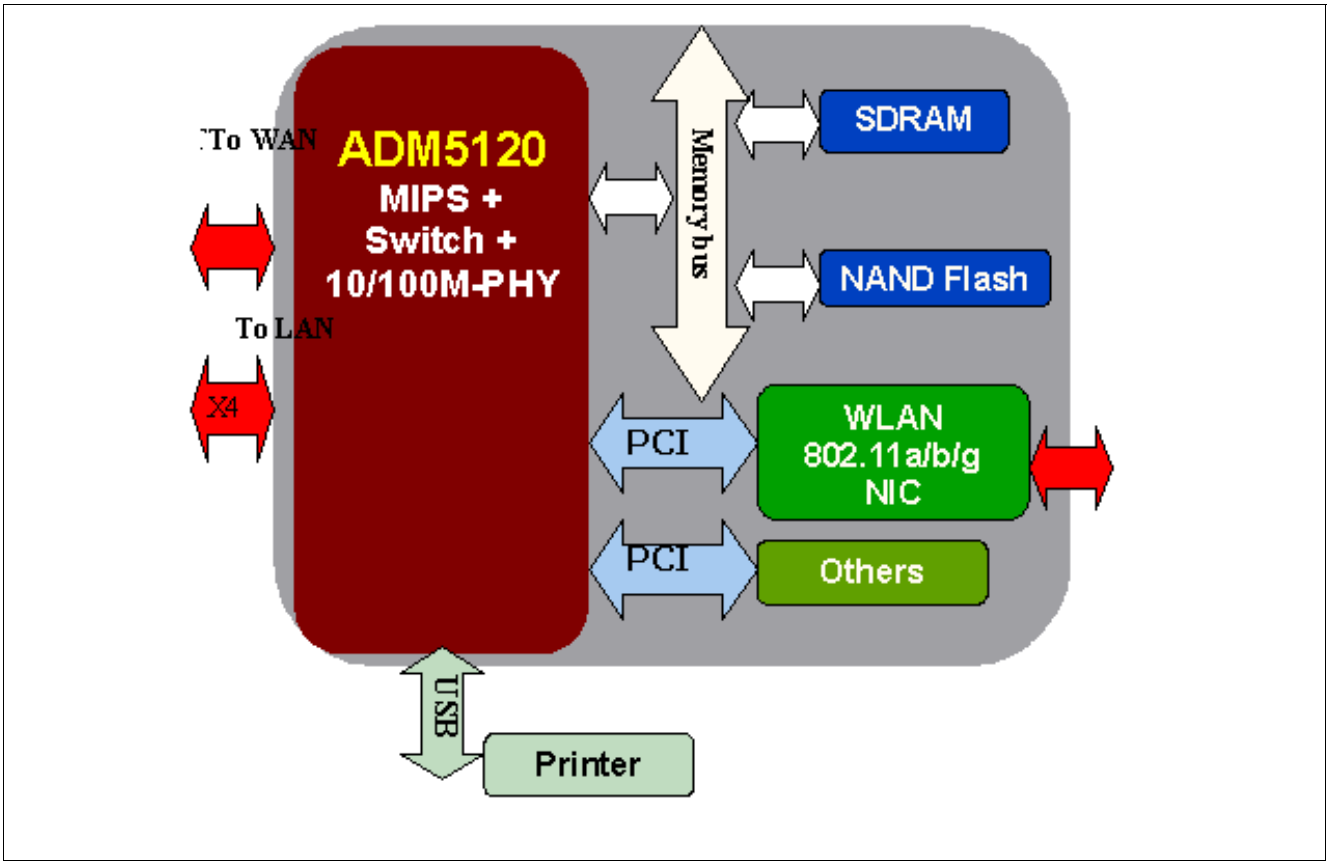


Figure 1 ADM5120 Application

1.3 Conventions

The convention descriptions are described below:

1.3.1 Data Lengths

qword = 64 bits
dword = 32 bits
word = 16 bits
byte = 8 bits
nibble = 4 bits

2 Interface Description

2.1 Pin Description by Function

ADM5120 pins are categorized into one of the following groups:

- **Section 2.1.4, Network Media Connection**
- **Section 2.1.5, Clock for Network**
- **Section 2.1.6, LED**
- **Section 2.1.7, GMII/MII Management**
- **Section 2.1.8, Memory Bus**
- **Section 2.1.9, SDRAM Control Signals**
- **Section 2.1.10, UART**
- **Section 2.1.11 JTAG**
- **Section 2.1.12, General Purpose I/O (GPIO)**
- **Section 2.1.13, PCI**
- **Section 2.1.14, USB**
- **Section 2.1.15, NAND Flash**
- **Section 2.1.16, External CS/INT/Wait**

2.1.1 Section

- **Section 2.1.17, Power and Ground**
- **Section 2.1.18, Regulator Interface**
- **Section 2.1.19, Miscellaneous**

Note: All default settings are 0.

2.1.2 Pin Diagram for P-BGA-324 Package

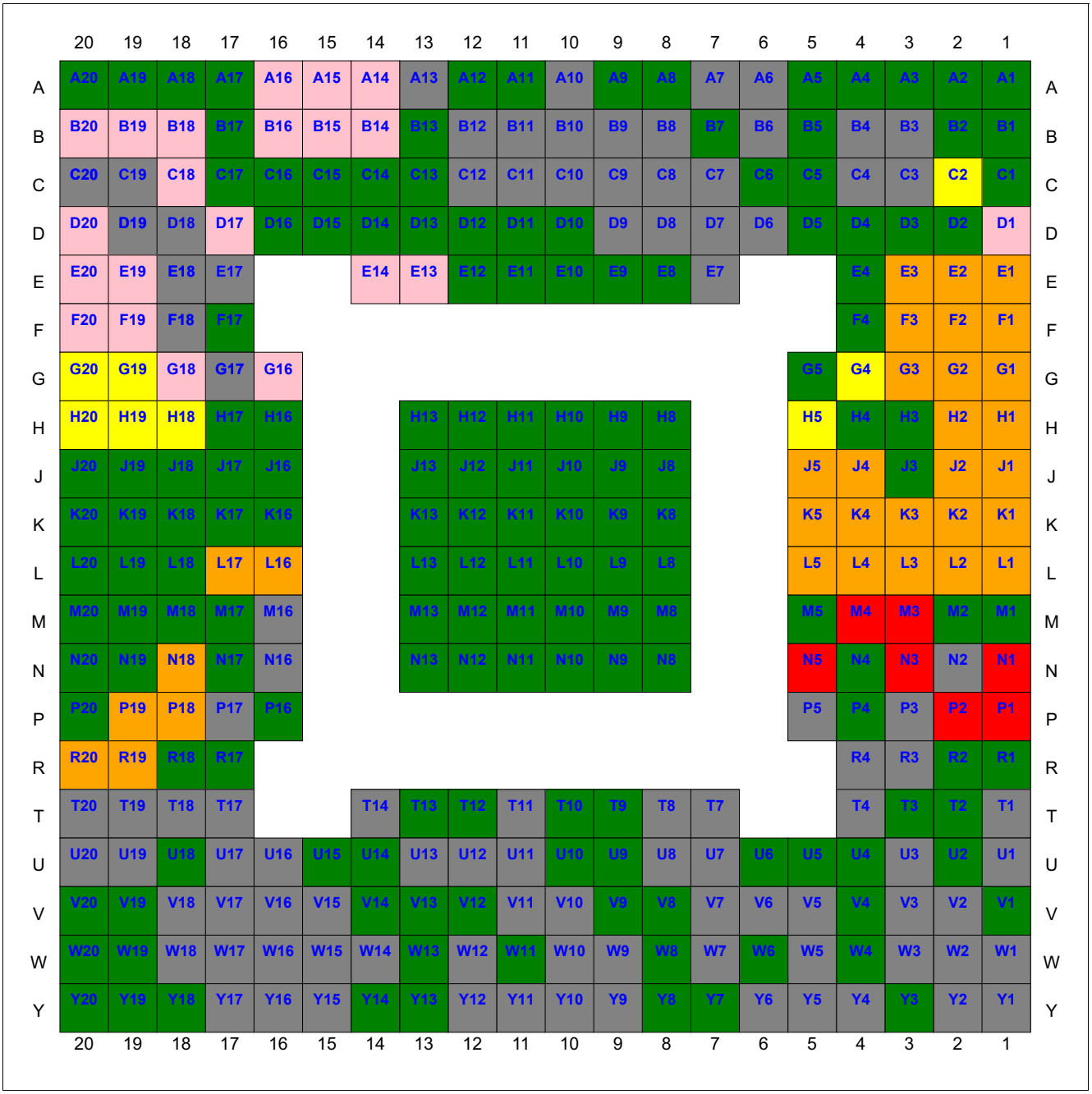


Figure 2 Pin Diagram BGA-324

2.1.3 Abbreviations

Table 1 Abbreviations for Pin Type

Abbreviations	Description
I	Standard input-only pin. Digital levels.
O	Output. Digital levels.
I/O	I/O is a bidirectional input/output signal.
AI	Input. Analog levels.
AO	Output. Analog levels.
AI/O	Input or Output. Analog levels.
PWR	Power
GND	Ground
MCL	Must be connected to Low (JEDEC Standard)
MCH	Must be connected to High (JEDEC Standard)
NU	Not Usable (JEDEC Standard)
NC	Not Connected (JEDEC Standard)

Table 2 Abbreviations for Buffer Type

Abbreviations	Description
Z	High impedance
PU1	Pull up, 10 kΩ
PD1	Pull down, 10 kΩ
PD2	Pull down, 20 kΩ
TS	Tristate capability: The corresponding pin has 3 operational states: Low, high and high-impedance.
OD	Open Drain. The corresponding pin has 2 operational states, active low and tristate, and allows multiple devices to share as a wire-OR. An external pull-up is required to sustain the inactive state until another agent drives it, and must be provided by the central resource.
OC	Open Collector
PP	Push-Pull. The corresponding pin has 2 operational states: Active-low and active-high (identical to output with no type attribute).
OD/PP	Open-Drain or Push-Pull. The corresponding pin can be configured either as an output with the OD attribute or as an output with the PP attribute.
ST	Schmitt-Trigger characteristics
TTL	TTL characteristics
A	Analog Differential pair or Analog PAD

2.1.4 Network Media Connection

Table 3 Network Media Connection

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
B10	RXP4	AI	A	**Receive Pair** Differential data is received on these pins.
B8	RXP3			
A7	RXP2			
B4	RXP1			
E7	RXP0			
A10	RXN4			
C8	RXN3			
A6	RXN2			
B3	RXN1			
D6	RXN0			
B9	TXP4	AO	A	**Transmit Pair** Differential data is transmitted on these pins.
D9	TXP3			
B6	TXP2			
D8	TXP1			
C3	TXP0			
C10	TXN4			
C9	TXN3			
C7	TXN2			
D7	TXN1			
C4	TXN0			

2.1.5 Clock for Network

Table 4 Clock for Network

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
H5	XOI	O	A	**Crystal Clock Output** 25 MHz crystal output
G4	XI	I	A	**External Clock Input** 25 MHz crystal input
C2	RTX	I	A	**Reference Voltage**

2.1.6 LED

Table 5 LED

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
F20	LED4_2	O	PD	**LED 4** LED4_2 state, default = 1010, duplex/colLED4_1 state, default = 0101, speedLED4_0 state, default = 1001, link/activity
G18	LED4_1			
F19	LED4_0			
G16	LED3_2	O	PD	**LED 3** LED3_2 state, default = 1010, duplex/colLED3_1 state, default = 0101, speedLED3_0 state, default = 1001, link/activity
E20	LED3_1			
D20	LED3_0			
E19	LED2_2	O	PD	**LED 2** LED2_2 state, default = 1010, duplex/colLED2_1 state, default = 0101, speedLED2_0 state, default = 1001, link/activity
E14	LED2_1			
A16	LED2_0			
B16	LED1_2	O	PD	**LED 1** LED1_2 state, default = 1010, duplex/colLED1_1 state, default = 0101, speedLED1_0 state, default = 1001, link/activity
A15	LED1_1			
B15	LED1_0			
A14	LED0_2	O	PD	**LED 0** LED0_2 state, default = 1010, duplex/colLED0_1 state, default = 0101, speedLED0_0 state, default = 1001, link/activity
E13	LED0_1			
B14	LED0_0			

*Note: Registers, not hardware pins, control the LED display. There are 3 LEDs per port, and they can be programmed to any state, the programming information can be found in **Table 6** below.*

Table 6 LED Program

Function	State
GPIO_in (or GPIO_disable)	0000
GPIO_output_flash	0001
GPIO_output_1	0010
GPIO_output_0	0011
Link (steady)	0100
Speed (steady)	0101
Duplex (steady)	0110
Activity (flash)	0111
Collision (flash)	1000
Link+activity	1001
Duplex+collision	1010
10 M_link+activity	1011
100 M_link+activity	1100
Reserved	1101

Table 6 LED Program (cont'd)

Function	State
Reserved	1110
Reserved	1111

2.1.7 GMII/MII Management

Table 7 GMII/MII Management

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
L3	MDC	O	PP	**Clock Input MDIO** Runs at a 1 MHz frequency clock for MII port auto-negotiation result monitoring.
F3	G_TXD_7	O		**Transmit Data** All internal pull down. 1. The force speed, duplex & flow control can be set by switch control register (B+14) 2. The reverse MII can only be set by switch control register (B+30)
E1	G_TXD_6			
G3	G_TXD_5			
F2	G_TXD_4			
F1	G_TXD_3			
G2	G_TXD_2			
J5	G_TXD_1			
G1	G_TXD_0			
H2	G_TXE	O		**Transmit Enable** Internal pull down.
H1	G_TXC	O		**Transmit Clock** Internal pull down.
K4	G_RXDV	I	TTL/PU	**Receive Data Valid** Internal pull down.
L2	G_RXD_7	I	TTL/PU	**Receive Data** Internal pull down.
L1	G_RXD_6			
K1	G_RXD_5			
K2	G_RXD_4			
L5	G_RXD_3			
K3	G_RXD_2			
K5	G_RXD_1			
J2	G_RXD_0			
J1	G_RXC	I	TTL/PD	**Receive Clock** Internal pull down.
J4	TXC	I	TTL/PD	**Transmit Clock** Internal pull down.
E3	GCRS	I	TTL/PD	**Carrier Sense** Internal pull down.

Table 7 GMII/MII Management (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
E2	G_COL	I	TTL/PD	**Collision** Internal pull down.
L4	MDIO	BI	PD	**Internal Pull Down** Bi-directional serial pin used to write and read from the registers of the device.

2.1.8 Memory Bus

Table 8 Memory Bus

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
P5	DATA_31	BI	PD	**Data Bus 31-0** Internal pull down. Data bus for SDRAM, flash memory, and external device.
U1	DATA_30			
W1	DATA_29			
V2	DATA_28			
V3	DATA_27			
Y2	DATA_26			
W3	DATA_25			
V5	DATA_24			
T4	DATA_23			
U3	DATA_22			
W2	DATA_21			
Y1	DATA_20			
T1	DATA_19			
R4	DATA_18			
R3	DATA_17			
P3	DATA_16			
W18	DATA_15			
T17	DATA_14			
V17	DATA_13			
U16	DATA_12			
V15	DATA_11			
Y16	DATA_10			
W15	DATA_9			
T14	DATA_8			
W12	DATA_7			
W14	DATA_6			
Y15	DATA_5			
U13	DATA_4			
W16	DATA_3			
Y17	DATA_2			
V16	DATA_1			
W17	DATA_0			

Table 8 Memory Bus (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
P17	ADDR_19	O	PD	**Address Bus 19** Address bus for SDRAM, flash memory, and external device. Internal pull down. Pull down = Little Endian. (default)
U20	ADDR_18			**Address Bus 18-17** Internal pull down. Can be pulled up and down as following: 00_B , boot in 8-bit mode (Flash memory) (default) 01_B , boot in 16-bit mode 10_B , boot in 32-bit mode 11_B , Reserved
U19	ADDR_17			
T18	ADDR_16			**Address Bus 16-14** Test mode purpose. Normal mode = 000(Default)
N16	ADDR_15			
U17	ADDR_14			
V18	ADDR_13			**Address Bus 13** Default value: 0 0_B , PHY separate power on disable 1_B , PHY separate power on enable
U7	ADDR_12			**Address Bus 12** 0_B , BGA package(Default) 1_B , 208 PQFP package
V7	ADDR_11			**Address Bus 11-5**
W9	ADDR_10			
Y6	ADDR_9			
W7	ADDR_8			
Y9	ADDR_7			
V10	ADDR_6			
W10	ADDR_5			
V11	ADDR_4			**Address Bus 4-3** Can be pulled up and down as following: PLL frequency setting. 00_B , 175 MHz (Default) 01_B , 200 MHz $1X_B$, Reserved
Y12	ADDR_3			
Y11	ADDR_2			**Address Bus 2** 0_B , Enable AutoMDIX 1_B , Disable AutoMDIX (Default)
U11	ADDR_1			**Address Bus 1** Default value: 0_B 0_B , NAND boot disable 1_B , NAND boot enable
Y10	ADDR_0			**Address Bus 0** Default value: 0_B 0_B , Normal operation 1_B , Simulation mode

2.1.9 SDRAM Control Signals

Table 9 SDRAM Control Signals

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
Y5	CLK_OUT	O	TS	**Clock Out** SDRAM clock, the frequency is set by ADDR_4:3 *Note: 1=pull up, 0=pull down* 00_D , 87.5 MHz (Default) 01_D , 100 MHz
M16	F_OE_N	O	PP	**Output Enable for External Memory** Output enable for external memory banks, active low.
T19	WE_N	O	PP	**Write Enable for External Memory** Write Enable for external memory banks and SDRAM.
N2	F_CS1_N	O	PP	**Chip Select for External Memory** Chip select for external memory, like flash, bank1, active low.
T20	F_CS0_N	O	PP	**Chip Select for External Memory** Chip select for external memory, like flash, bank0, active low.
T8	RAS_N	O	PP	**Raw Address Strobe** Raw address strobe, active low.
U8	SD_RAM_CS0_N	O	PP	**SDRAM Chip Select 0** SDRAM chip select 0.
V6	CAS_N	O	PP	**Column Address Strobe** Column address strobe, active low.
W5	SD_RAM_CS1_N	O	PP	**SDRAM Chip Select 1** SDRAM chip select 1.
T11	DQM_3	O	PD	**Data Mask Output to SDRAM**
U12	DQM_2			
T7	DQM_1			
Y4	DQM_0			

2.1.10 UART

Table 10 UART

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
R19	UDCD	I	PD	**Data Carrier Detect** UART0 Data carrier detect (modem status input), active low.
P18	UDSR			**Data Set Ready** UART0 Data set ready (modem status input), active low.
L17	UCTS			**Clear to Send** UART0 clear to send (modem status input), active low.
R20	UDI0			**Receive Serial Data Input** UART0 receive serial data input, Internal pull down.
N18	UDO0	O		**Transmit Serial Data Output** UART0 transmit serial data output.
P19	UDI1	I		**Receive Serial Data Input** UART1 receive serial data input, Internal pull down.
L16	UDO1	O		**Transmit Serial Data Output** UART1 transmit serial data output.

2.1.11 JTAG

Table 11 JTAG

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
H18	TCK	I	PD	**Test Clock** JTAG test clock, Internal pull down.
G19	TMS			**Test Mode Select** JTAG test mode select, Internal pull down.
G20	TDO	O		**Test Data Out** JTAG test data out.
H19	TDI	I		**Test Data In** JTAG test data in, Internal pull down.
H20	TRST_N	I	TTL	**Asynchronous Reset** JTAG asynchronous reset (active low).

2.1.12 General Purpose I/O (GPIO)

Table 12 General Purpose I/O (GPIO)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
E17	GPIO_7	BI	PD	**BI General Purpose I/O Pin** GPIO_0 is internal pull up. GPIO_2:1 are internal pull down.Note: In the BGA version GPIO5 can be programmed to SDRAM memory address A20 for 2 M x 16-bit Flash thus supporting large Flash memory.
D18	GPIO_6			
C19	GPIO_5			
C20	GPIO_4			
E18	GPIO_3			
G17	GPIO_2			
D19	GPIO_1			
F18	GPIO_0			
	LEDN_2			**General Purpose I/O Pin 8:3** General purpose I/O pin GPIO_8:3 are internal pull down. *Note: Refer to LED section for Ball Numbers.*
	LEDN_1			
	LEDN_0			

2.1.13 PCI

Table 13 PCI

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
W4	PCI_AD_31	BI	TS	**Address and Data Bus** Address and data bus are multiplexed on the same PCI pins. A bus transaction consists of an address phase followed by one or more data phase. The PCI host bridge supports both read and write bursts.
Y7	PCI_AD_30			
V8	PCI_AD_29			
T9	PCI_AD_28			
W8	PCI_AD_27			
Y8	PCI_AD_26			
V9	PCI_AD_25			
U9	PCI_AD_24			
Y13	PCI_AD_23			
T12	PCI_AD_22			
W13	PCI_AD_21			
Y14	PCI_AD_20			
V13	PCI_AD_19			
T13	PCI_AD_18			
U15	PCI_AD_17			
Y18	PCI_AD_16			
Y19	PCI_AD_15			
Y20	PCI_AD_14			
W19	PCI_AD_13			
V20	PCI_AD_12			
U18	PCI_AD_11			
R18	PCI_AD_10			
V19	PCI_AD_9			
N17	PCI_AD_8			
P20	PCI_AD_7			
N19	PCI_AD_6			
N20	PCI_AD_5			
J16	PCI_AD_4			
M18	PCI_AD_3			
M19	PCI_AD_2			
M20	PCI_AD_1			
L18	PCI_AD_0			
U6	PCI_CBE_3	BI	TS	**Bus Command and Byte Enable**
V4	PCI_CBE_2			
U5	PCI_CBE_1			
U4	PCI_CBE_0			

Table 13 PCI (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
M1	PCI_DEVSEL	BI	TS	**Device Select** When actively indicates the driving device has decoded its address as the target of the current access.
M5	PCI_FRAME	BI	TS	**Cycle Frame** PCI_FRAME is asserted to indicate a bus transaction is beginning and data transfers continue
J19	PCI_GNT_2	O	TS	**Grant** It indicates to the master that access to the PCI bus has been granted.
J20	PCI_GNT_1			
K20	PCI_GNT_0			
M2	PCI_IRDY	BI	TS	**Initiator Ready**
T2	PCI_PAR	BI	TS	**Parity**
T3	PCI_PERR	BI	TS	**Parity Error**
K19	PCI_REQ_2	I	TTL	**PCI Bus Request**
K18	PCI_REQ_1			
J17	PCI_REQ_0			
R2	PCI_SER	BI	TS	**System Error**
U2	PCI_STOP	BI	TS	**Stop Indicates** Stop indicates the current target is requesting the host to stop the current transaction due to unusual condition
R1	PCI_TRDY	BI	TS	**Target Ready**
L20	PCI_INTA_2	I	TTL	**PCI Interrupt Input**
L19	PCI_INTA_1			
K16	PCI_INTA_0			
H16	PCI_RESET	O	TS	**PCI Bus Reset**
J18	PCI_CLK33	I	TTL	**PCI Bus Clock Input**
C17	CLKO33M	0	PP	**31.25 MHz Clock Output** This is 31.25 MHz clock output.

2.1.14 USB

Table 14 USB

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
C12	DMNS1	BI	A	**Data- of USB Port1** Differential data bus conforming to the USB 1.1.
B12	DPLS1			**Data+ of USB Port1** Differential data bus conforming to the USB 1.1.
C11	DMNS0			**Data- of USB Port0** Differential data bus conforming to the USB 1.1.
B11	DPLS0			**Data+ of USB Port0** Differential data bus conforming to the USB 1.1.
A13	CLK48M	I	TTL	**USB Clock Input**

2.1.15 NAND Flash

Table 15 NAND Flash

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
N3	NAND_OE_N	O	PP	**Read Enable**
N5	NAND_WE_N			**Write Enable**
M3	CLE			**Command Latch Enable**
N1	ALE			**Address Latch Enable**
M4	WP			**Write Protect**
P1	RDY	I	TTL/PU	**Ready/Busy Input**
P2	SP	O	PP	**Spare Enable**

2.1.16 External CS/INT/Wait

Table 16 External CS/INT/Wait

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
F18	$\overline{\text{WAIT}}$	I	TTL	**$\overline{\text{WAIT}}$** $\overline{\text{WAIT}}$ is available in switch control register **Chapter 6**, bit **GPIO_conf0** and **GPIO_conf2**. When CSX active and SMC programmable wait_state time-out, then check the $\overline{\text{WAIT}}$ if high, then complete the accessif low, then wait until $\overline{\text{WAIT}}$ go high.
G17	INTX0	I		**External Interrupt Input 0** External interrupt input, active high, available if en_csx_intx enable in the switch control register GPIO_config2 (B+BC), bit[4].
C20	$\overline{\text{INTX1}}$	I		**External Interrupt Input 1** Internal pull down, external interrupt input 1, active high, available if en_csx_intx_1 enable in the switch control register GPIO_config2 (B+BC), bit[5].
D19	$\overline{\text{CSX0}}$	O	TS	**External Chip Select 0** External chip select, active low, available if en_csx_intx enable in the switch control register **Chapter 6** bit **GPIO_conf0** and **GPIO_conf2**.
E18	$\overline{\text{CSX\_1}}$	O	TS	**External Chip Select 1** External chip select 1, active low, available if en_csx_intx_1 enable in the switch control register GPIO_config2 (B+BC), bit[5].

2.1.17 Power and Ground

Table 17 Power and Ground

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
F17, D13, D12, V14, V12, U10, T10, H3, H4, K17, P16	VDD		A	**Positive Power for Digital Core, 1.8 V**
J3, D14, D15, H17, W6, W11, U14, R17, M17, N4, P4	DVDD		A	**Positive Power for I/O, 3.3 V**
D4, A5, B5, A8, A9	VDDTS2		A	**Positive Power for Analog Circuitry, 1.8 V**
C5, B7, D11	VCCAD		A	**Positive Power for Analog Circuitry, 3.3 V**

Table 17 Power and Ground (cont'd)

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
H11, H12, A11, A20, A19, A18, A17, B17, D16, C16, C15, C14, C13, H13, N12, N11, L10, M10, N10, V1, Y3, N8, N9, M8, K8, H9, J9, H10, J10, K10, K9, B2, A1, A2, A3, A4, C6, D5, E8, H8, J8, L8, M9, M11, M12, W20, N13, M13, L9, L11, L12, K13, J11, J12, J13, K11, K12, L13	VSS		A	**GND for Digital Circuit**
E12, E11, D10, E10, E9	VSSA		A	**GND for Analog Circuitry**
D2	VCCPLL		A	**Power for Phase Lock Loop, 1.8 V**
E4	VCCRG		A	**Power for Regulator, 3.3 V**
D3	VCCBIAS		A	**Power for BIAS, 3.3 V**
C1, B1	GNDRG		A	**Ground**
A12	AV33		A	**Power for USB PHY, 3.3 V**
B13	AG33		A	**Power for USB GND**

2.1.18 Regulator Interface

Table 18 Regulator Interface

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
G5	VREF	AI	A	**Reference Voltage Input** This pin is used as the reference voltage for regulator and generate the 1.8 V output from the 3.3 V power source.
F4	CONTROL	A0	A	**FET Control Output**

2.1.19 Miscellaneous

Table 19 Miscellaneous

Pin or Ball No.	Name	Pin Type	Buffer Type	Function
C18	TEST	I	TTL/PD	**Test Pin** This pin is for test purpose and should be connected to GND for normal operation.
B18	RESET_N	I	TTL	**System Reset** Active low will initial the chip to default state.
D17	CLKO25M	O	PP	**25 MHz clock output**
D1, B19, B20	NC			**Not Connected**

3 System

The **Figure 3** show the blocks of ADM5120. The blocks are broken down into:

- MIPS 4Kc
- PCI bridge
- Multi port memory controller
- USB 1.1 host controller
- UART controller
- Ether switch
  - SW2CPU DMA: this block handle the packets TX/RX from/to CPU
  - Embedded data buffer
  - Embedded link table/MC table/addr. table
  - 802.3MAC and DMA
- 10/100 DSP PHY

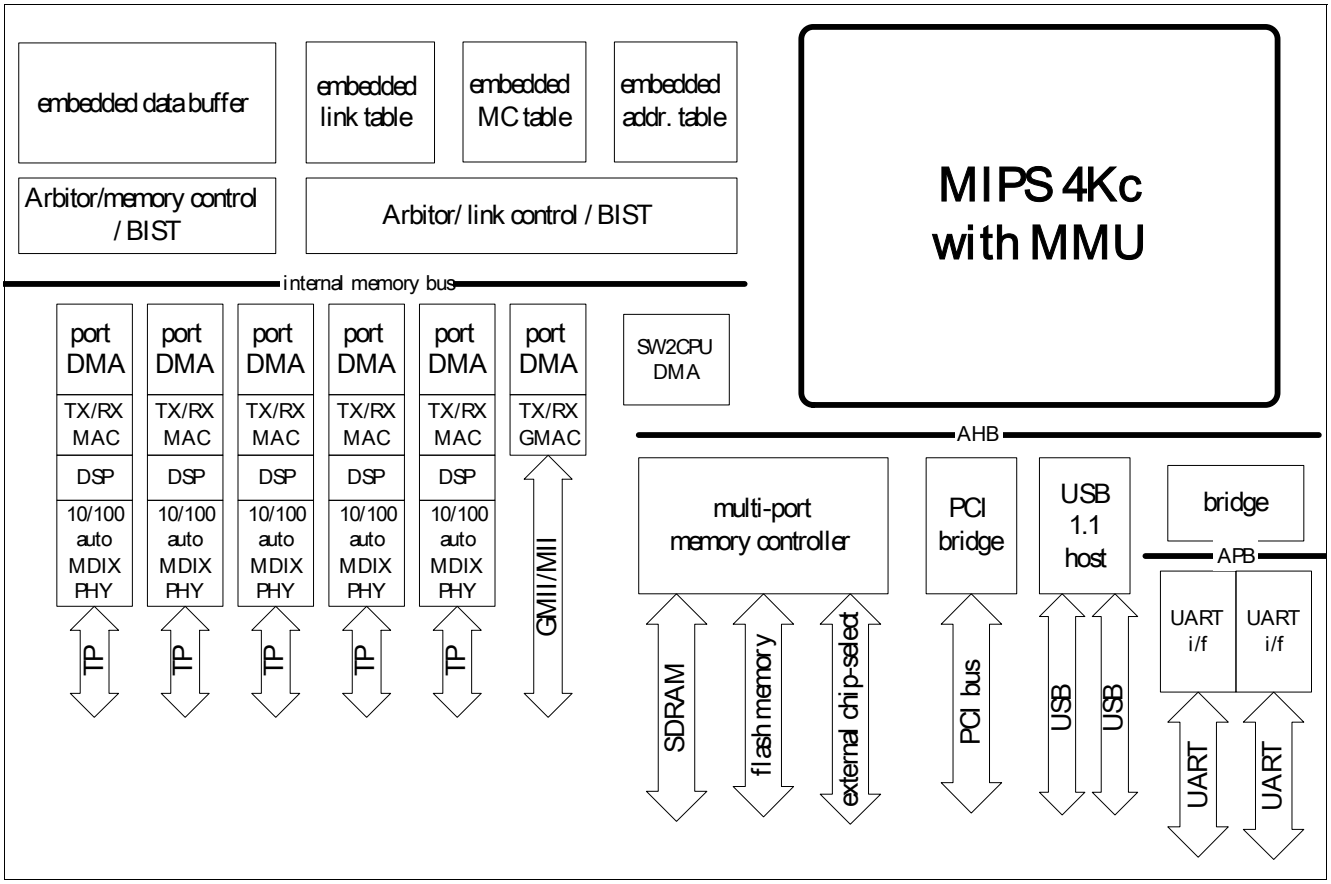


Figure 3 ADM5120 Block Diagram

3.1 System Memory Map

Figure 4 shows the system memory allocation. The following lists a detailed reference for each allocation:

- Boot address is located in SRAM_0, for detailed information refer to MPMC Chapter
  - The address is fixed at $1FC0\text{-}0000_H$ and the maximum size is 4 Mbyte. The data-width is pin setting and register readable. The size is programmable.
- SRAM_1, the bank1 of NOR or NAND flash, refer to MPMC Chapter
  - The address is fixed at $1000\text{-}0000_H$
  - The maximun size is 8 Mbyte. The size is programmable.
  - If in the NAND type mode, no SRAM_0 device. The first 2K bytes of boot code will be loaded into the embedded SRAM from NAND flash in the initial time.

- Ext_IO_0 and ext_IO_1 are also the generic SRAM space, refer to MPMC Chapter
- For SRAM_0, SRAM_1, ext_IO_0 and ext_IO_1 the address and data width relation are
  - Byte access, address = [20:0], max size = 2 Mbytes
  - 16-bit access, address = [21:1] (shift A0 out), max size = 4 Mbytes
  - 32-bit access, address = [22:2] (shift A0, A1 out), max size = 8 Mbytes
  - Use DQM to select the bytes
- SDRAM_0 and SDRAM_1 are generic SDRAM space, the detail information refer to MPMC Chapter
- MPMC is the Multi Port Memory controller which includes Static and Dynamic memory controller. The detail information refer MPMC.
- USB 1.1 host controller, refer the USB Chapter
- Switch part is Ether Switch which support 6 ports switch and one CPU DMA port. The detail information refer the Switch
- There are two serial ports UART_0 and UART_1, refer the UART.
- For PCI application, ADM5120 can be used as PCI host. The PCI_COnfiguration_Addr and PCI_Configuration_Data are used to access the PCI configuration space and configure the PCI device. While PCI_IO and PCI_Memory are used to access PCI device for IO and Memory space. For more detail information refer **Chapter 4**
  - $1140\text{-}0000_H$ to $114F\text{-}FFFF_H$ is PCI memory access
  - $1150\text{-}0000_H$ to $115F\text{-}FFEF_H$ is PCI I/O space
  - $115F\text{-}FFF0_H$ is PCI configuration address port (Dword Access only)
  - $115F\text{-}FFF8_H$ is PCI configuration data port(Dword Access only)
- INTC is interrupt controller.

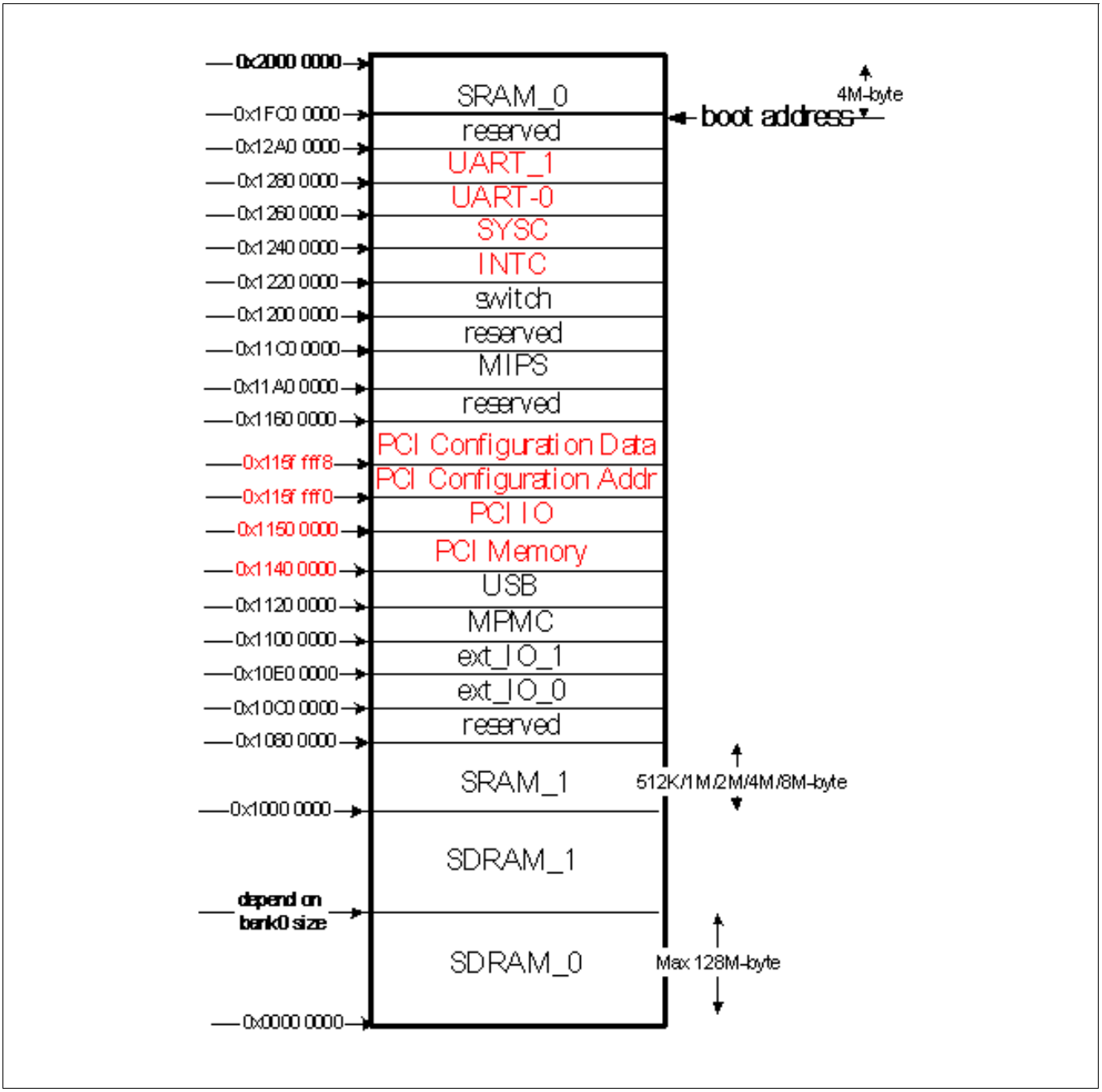


Figure 4 System Memory Map

3.1.1 Memory Mapping Notes

The following describes the memory mapping in detail:

Design Notes

- PCI device
  - PCI_Device_ID : 5120_H
  - PCI_Vendor_ID : 1317_H
  - PCI_Class_Code : 060000_H
  - PCI_Revision_ID : 00_H
  - PCI_Subsystem_ID : 0000_H
  - PCI_Subsystem_Vendor_ID : 0000_H
  - AHB to PCI Bridge IDSEL is PCI_AD[11] external PCI device's IDSEL can use PCI_AD[12] or higher bits.

3.2 System and Interrupt Registers

The following chapter describes the system and interrupt Registers.

3.2.1 Interrupt Control Register Map

Interrupt register description:

Table 20 Address Space

Module	Base Address	End Address	Note
Interrupt Control	1220 0000$_H$	1220 0024$_H$	

Table 21 Registers Overview from Interrupt Registers

Register Short Name	Register Long Name	Offset Address
IRQS	Interrupt Request Status	00$_H$
IRQRS	Interrupt Request Raw Status	04$_H$
IRQE	Interrupt Request Enable	08$_H$
IRQEC	Interrupt Request Enable Clear	0C$_H$
Res_0	Reserved 0	10$_H$
INT_M	Interrupt Mode	14$_H$
FIQ_S	Fast Interrupt Request Status	18$_H$
IRQ_TS	Interrupt Request Test Source	1C$_H$
IRQSS	Interrupt Request Source Sel	20$_H$
INT_L	Interrupt Level	24$_H$

3.2.2 Register Access Types

Table 22 Register Access Types

Mode	Symbol	Description HW	Description SW
read/write	rw	Register is used as input for the HW	Register is read and writable by SW
read	r	Register is written by HW (register between input and output -> one cycle delay)	Value written by software is ignored by hardware; that is, software may write any value to this field without affecting hardware behavior (= Target for development.)
Read only	ro	Register is set by HW (register between input and output -> one cycle delay)	SW can only read this register
Read virtual	rv	Physically, there is no new register, the input of the signal is connected directly to the address multiplexer.	SW can only read this register
Latch high, self clearing	lhsc	Latch high signal at high level, clear on read	SW can read the register
Latch low, self clearing	llsc	Latch high signal at low-level, clear on read	SW can read the register
Latch high, mask clearing	lhmk	Latch high signal at high level, register cleared with written mask	SW can read the register, with write mask the register can be cleared (1 clears)

Table 22 Register Access Types (cont'd)

Mode	Symbol	Description HW	Description SW
Latch low, mask clearing	llmk	Latch high signal at low-level, register cleared on read	SW can read the register, with write mask the register can be cleared (1 clears)
Interrupt high, self clearing	ihsc	Differentiate the input signal (low->high) register cleared on read	SW can read the register
Interrupt low, self clearing	ilsc	Differentiate the input signal (high->low) register cleared on read	SW can read the register
Interrupt high, mask clearing	ihmk	Differentiate the input signal (high->low) register cleared with written mask	SW can read the register, with write mask the register can be cleared
Interrupt low, mask clearing	ilmk	Differentiate the input signal (low->high) register cleared with written mask	SW can read the register, with write mask the register can be cleared
Interrupt enable register	ien	Enables the interrupt source for interrupt generation	SW can read and write this register
latch_on_reset	lor	rw register, value is latched after first clock cycle after reset	Register is read and writable by SW
Read/write self clearing	rwsc	Register is used as input for the hw, the register will be cleared due to a HW mechanism.	Writing to the register generates a strobe signal for the HW (1 pdi clock cycle) Register is read and writable by SW.

3.2.3 Registers Description

The interrupt controller support level sensitive. The external input level can be programmed as active high or low.

Interrupt Request Status

IRQS	Offset	Reset Value
Interrupt Request Status	**00_H**	**0_H**

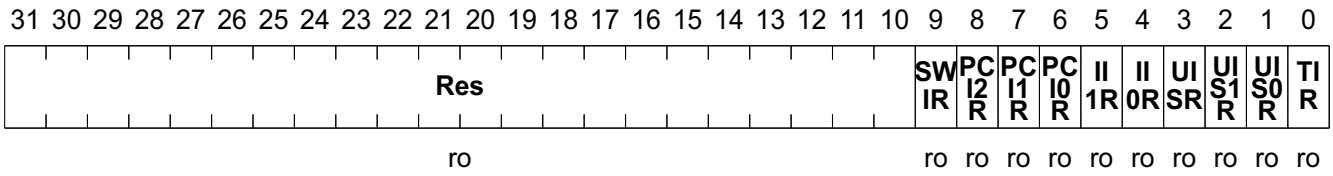
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res																						SWI	PCI2	PCI1	PCI0	II 1	II 0	UIS	UIS1	UIS0	TI
rw																						ro	ro	ro	ro	ro	ro	ro	ro	ro	ro

Field	Bits	Type	Description
Res	31:10		**Reserved**
SWI	9	ro	**Switch Interrupt Status After Mask** Switch interrupt, refer to switch registers offset 0xB0 and 0xB4 for more detail.
PCI2	8	ro	**PCI INT2 Status After Mask**
PCI1	7	ro	**PCI INT1 Status After Mask**
PCI0	6	ro	**PCI INT0 Status After Mask**
II1	5	ro	**Internal Interrupt 1 Status After Mask** Internal interrupt 1, refer to switch registers offset 0xBC bit[5] for more detail. pin GPIO 4 is the source.

Field	Bits	Type	Description
II0	4	ro	**Internal Interrupt 0 Status After Mask** Internal interrupt 0, refer to switch registers offset 0xBC bit[5] for more detail. pin GPIO 2 is the source.
UIS	3	ro	**USB Interrupt Source Status After Mask**
UIS1	2	ro	**UART1 Interrupt Source Status After Mask**
UIS0	1	ro	**UART0 Interrupt Source Status After Mask**
TI	0	ro	**Timer Interrupt Status After Mask** Timer interrupt, refer to switch registers offset 0xF0 and 0xF4 for more detail.

Interrupt Request Raw Status

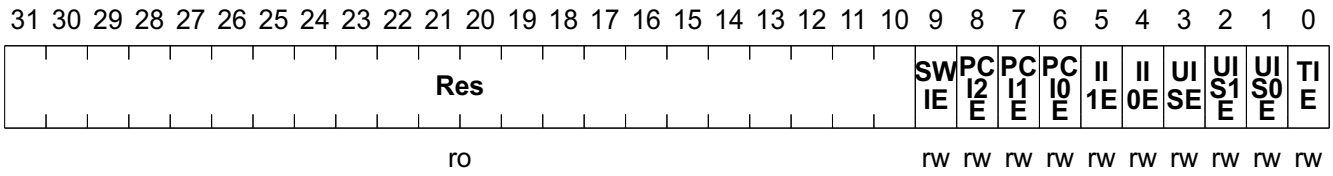
IRQRS	Offset	Reset Value
Interrupt Request Raw Status	04_H	0_H



Field	Bits	Type	Description
Res	31:10	ro	**Reserved**
SWIR	9	ro	**Switch Interrupt Raw Status** The Switch interrupt status before masking. refer to switch registers offset 0xB0 and 0xB4 for more detail.
PCI2R	8	ro	**PCI INT2 Raw Status**
PCI1R	7	ro	**PCI INT1 Raw Status**
PCI0R	6	ro	**PCI INT0 Raw Status**
II1R	5	ro	**Internal Interrupt 1 Raw Status** Internal interrupt 1, refer to switch registers offset 0xBC bit[5] for more detail. pin GPIO 4 is the source.
II0R	4	ro	**Internal Interrupt 0 Raw Status** Internal interrupt 0, refer to switch registers offset 0xBC bit[5] for more detail. pin GPIO 2 is the source.
UISR	3	ro	**USB Interrupt Raw Status**
UIS1R	2	ro	**UART1 Interrupt Raw staus**
UIS0R	1	ro	**UART0 Interrupt Raw staus**
TIR	0	ro	**Timer Interrupt Raw Status** Timer interrupt, refer to switch registers offset 0xF0 and 0xF4 for more detail.

Interrupt Request Enable

IRQE	Offset	Reset Value
Interrupt Request Enable	08_H	0_H

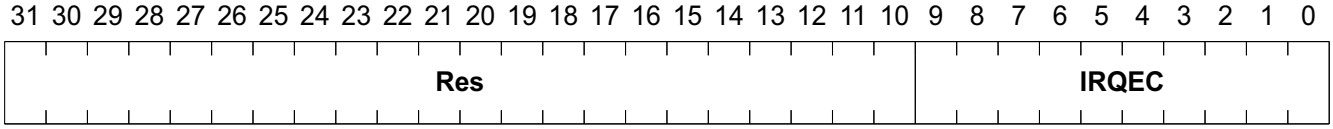


Field	Bits	Type	Description
Res	31:10	ro	**Reserved** Not Applicable.
SWIE	9	rw	**Switch Interrupt Enable** The enable register is used to maks the switch interrupt source. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS
PCI2E	8	rw	**PCI INT2 Enable** The enable register is used to maks the PCI interrupt 2 source. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS
PCI1E	7	rw	**PCI INT1 Enable** The enable register is used to maks the PCI interrupt 1source. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS
PCI0E	6	rw	**PCI INT0 Enable** The enable register is used to maks the PCI interrupt 0 source. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS
II1E	5	rw	**Internal Interrupt 1 Enable** The enable register is used to maks the GPIO 4 interrupt source. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS
II0E	4	rw	**Internal Interrupt 0 Enable** The enable register is used to maks the GPIO 2 interrupt source. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS
UISE	3	rw	**USB Interrupt Enable** The enable register is used to maks the USB interrupt source. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS
UIS1E	2	rw	**UART1 Interrupt Enable** The enable register is used to maks the UART 1 interrupt source. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS

Field	Bits	Type	Description
UIS0E	1	rw	**UART0 Interrupt Enable** The enable register is used to maks the UART 0 interrupt source. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS
TIE	0	rw	**Timer Interrupt Enable** The enable register is used to maks the Timer interrupt source. refer to B+F0 and B+F4. 0_B , No effect 1_B , Enable the interrupt and allow the interrupt request to MIPS

Interrupt Request Enable Clear

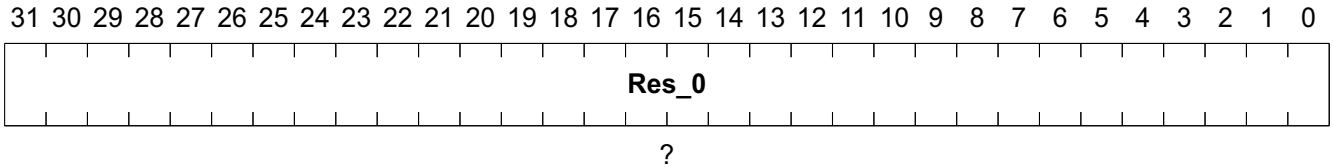
IRQEC	Offset	Reset Value
Interrupt Request Enable Clear	**$0C_H$**	**0_H**



Field	Bits	Type	Description
Res	31:10		**Reserved** Not Applicable.
IRQEC	9:0	rw	**IRQ Enable Clear 9:0** The clear bits of the IRQ_enable. 0_B , No effect 1_B , Clear the corresponding bit of IRQ_enable

Reserved 0

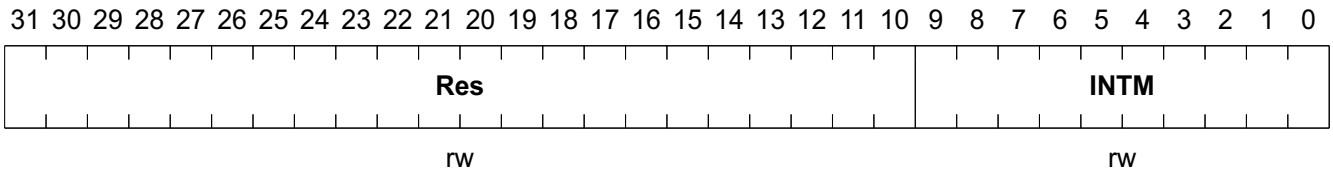
Res_0	Offset	Reset Value
Reserved 0	**10_H**	**0_H**



Field	Bits	Type	Description
Res_0	31:0		**Reserved** Not Applicable.

Interrupt Mode

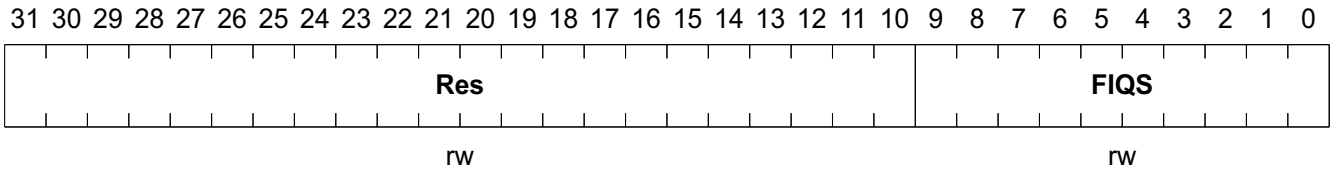
INT_M	Offset	Reset Value
Interrupt Mode	14_H	0_H



Field	Bits	Type	Description
Res	31:10		**Reserved** Not Applicable.
INTM	9:0	rw	**INT Mode 9:0** The interrupt type of the interrupt sources. 0_B , The corresponding Interrupt port generate the IRQ to MIPS 1_B , The corresponding Interrupt port generate the FIQ to MIPS

Fast Interrupt Request Status

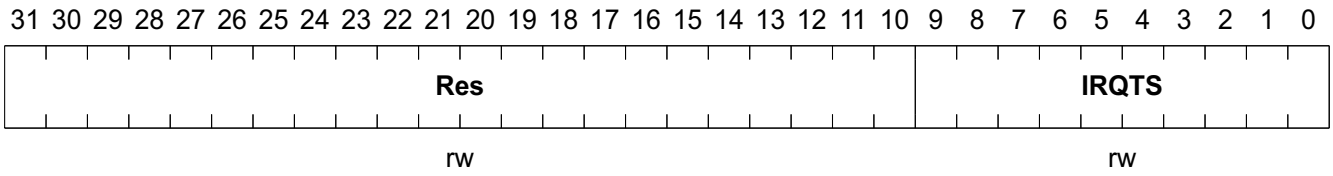
FIQ_S	Offset	Reset Value
Fast Interrupt Request Status	18_H	0_H



Field	Bits	Type	Description
Res	31:10		**Reserved** Not Applicable.
FIQS	9:0	rw	**FIQ Status 9:0** The status of the fast interrupt sources after masking. 1_B , The corresponding IRQ is active, and generate the interrupt to MIPS

Interrupt Request Test Source

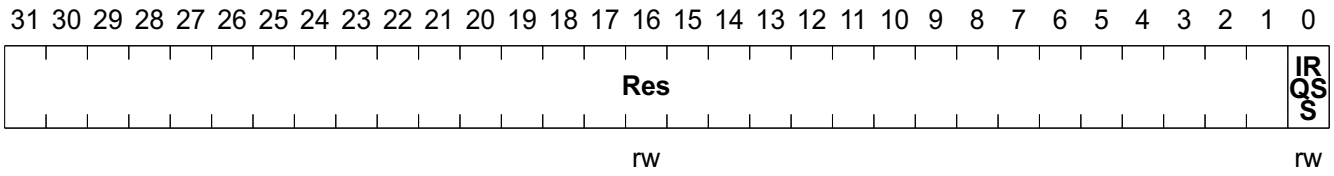
IRQ_TS	Offset	Reset Value
Interrupt Request Test Source	$1C_H$	0_H



Field	Bits	Type	Description
Res	31:10		**Reserved** Not Applicable.
IRQTS	9:0	rw	**IRQ Test Source 9:0** The test data for the IRQ_raw_status.

Interrupt Request Source Sel

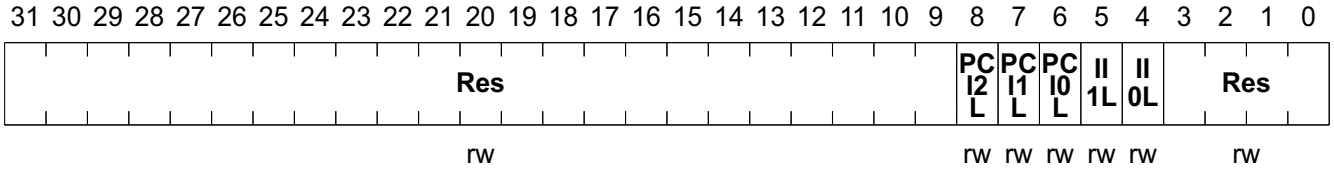
IRQSS	Offset	Reset Value
Interrupt Request Source Sel	20_H	0_H



Field	Bits	Type	Description
Res	31:1		**Reserved** Not Applicable.
IRQSS	0	rw	**IRQ Source Selection** 1_B , Load the IRQ_test_source into IRQ_raw_status

Interrupt Level

INT_L	Offset	Reset Value
Interrupt Level	24_H	$E4_H$



Field	Bits	Type	Description
Res	31:5		**Reserved** Not Applicable.
II1L	4	rw	**Internal Interrupt 1 Level** Level specify for GPIO 4 interrupt source. 0_B , Active high (default) 1_B , Active low
II0L	3	rw	**Internal Interrupt 0 Level** Level specify for GPIO 2 interrupt source. 0_B , Active high (default) 1_B , Active low
Res	2:0		**Reserved** Not Applicable.

4 Main Processor

The CPU description covers:

1. Feature list (**Chapter 4.1**)
2. Functional description (**Chapter 4.2**)

4.1 4Kc CPU Core Features

The 4Kc CPU supports:

- 32-bit Data and Address Paths
- MIPS32™ Compatible Instruction Set
  - All MIPS II™ Instructions
  - Multiply-Add and Multiply-Subtract Instructions (MADD, MADDU, MSUB, MSUBU)
  - Targeted Multiply Instruction (MUL)
  - Zero and One Detect Instructions (CLZ, CLO)
  - Wait Instruction (WAIT)
  - Conditional Move Instructions (MOVZ, MOVN)
  - Prefetch Instructions (PREF)
- MIPS16e Application Specific Extension
  - 16 bit encoding of 32 bit instructions to improve code density
  - Special PC relative instructions for efficient loading of addresses and constants
  - Data type conversion instructions (ZEB, SEB, ZEH, SEH)
  - Compact Jumps (JRC, JALRC)
  - Stack frame set-up and tear down “macro” instructions (SAVE and RESTORE)
- Instruction and Data Cache
  - 8 KByte Instruction Cache Size in a 4-Way set associative organization
  - 4 KByte Data Cache Size in a 2-Way set associative organization
  - Loads that miss in the cache are blocked only until critical word is available
  - Supports Write-back with write-allocation and Write-through with or without write- allocation
  - 16-byte cache line size, word sectored
  - Virtually indexed, physically tagged
  - Support for cache line locking
  - Non-blocking prefetches
- MIPS32™ privileged resource architecture
  - Count/Compare registers for real-time timer interrupts
  - Instruction and Data watch registers for software breakpoints
  - Separate interrupt exception vector
- Memory Management Unit
  - 16 dual-entry MIPS32 style JTLB with variable page sizes
  - 4-entry instruction micro TLB
  - 4-entry data micro TLB
- Core Bus Interface Unit (Core BIU)
  - All I/Os fully registered
  - Separate, unidirectional 32-bit address and data buses
  - Two 16 Byte collapsing write buffers
- Multiply Divide Unit
  - Maximum issue rate of one 32 ⌋ 16 multiply per clock
  - Maximum issue rate of one 32 ⌋ 32 multiply every other clock
  - Early-in divide control. Minumum 11, maximum 34 clock cycles
- Power Control
  - No minimum clock frequency

  - Power Down mode (triggered by WAIT instruction)
  - Support for software controlled clock divider
- EJTAG Debug Support
  - CPU control with start, stop and single-step feature
  - Software Breakpoints via SDBBP instruction
  - Hardware Breakpoints on virtual addresses
  - Test Access Port (TAP) facilitates high speed download of application
  - Optional EJTAG Trace hardware to enable real-time tracing of executed code

4.2 Functional Description

The block diagram of the main processor subsystem is given in **Figure 5**.

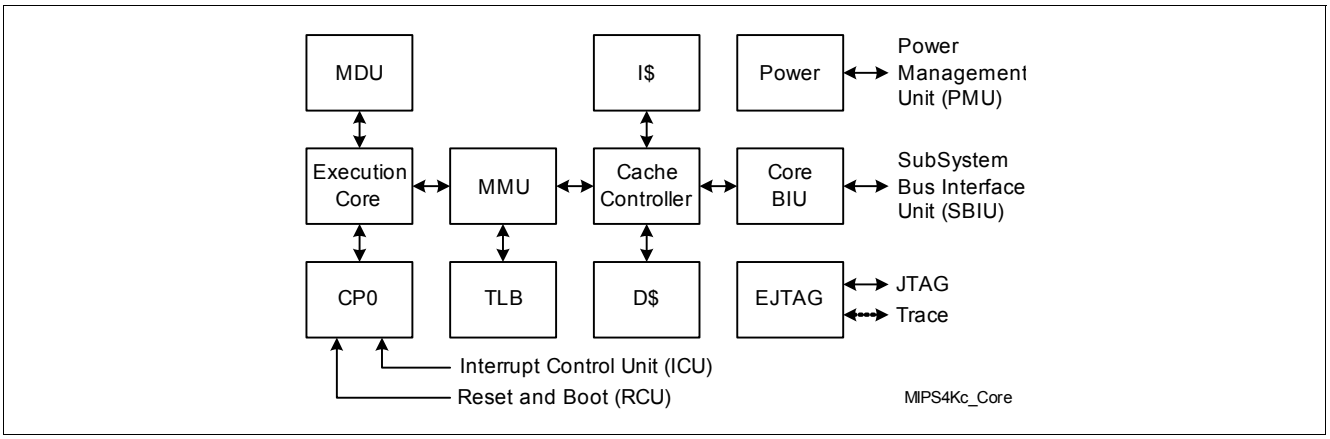


Figure 5 Main Processor Subsystem

The main processor subsystem consists of the below major parts:

- The MMU enabled MIPS 4KC core and associated cache system
- The bus wrapper block translates the MIPS 4Kc EC bus to the system bus

4.2.1 Endianness Mode

The main processor sub-system is capable of running at both big endian and little endian mode. By default, it is set to little endian mode. It can be switched to big endian mode by pin strapping the signal : ADDR[19](Ball P17).

Table 23 Endian Setting

Signal ADDR[19]	Mode
1	Big Endian
0	Little Endian (default)

4.2.2 Coprocessor CP0

In the MIPS architecture, System Control Coprocessor (CP0) is responsible for the virtual-to-physical address translation, cache protocols, exception control system, processor's diagnostics capability, the operating mode selection (Kernel, Supervisor, User, and Debug) and enabling/disabling of interrupts. Information such as CPU status, performance and configuration (including caches) are available by accessing the CP0 registers.

4.2.3 Execution Unit

The 4Kc™ core execution unit implements a load/store architecture with single-cycle ALU operations (logical, shift, add, subtract). The 4Kc™ core contains thirty-two 32-bit general-purpose registers used for scalar integer

operations and address calculation. The register file consists of two read ports and one write port and is fully bypassed to minimize operation latency in the pipeline.

The execution unit includes:

- 32-bit adder used for calculating arithmetic results and the data addresses
- Address Unit for calculating the next instruction address
- Logic for branch determination and branch target address calculation
- Load aligner
- Bypass multiplexers used to avoid stalls when executing instructions streams where data producing instructions are followed closely by consumers of their results
- Zero/One detect unit for implementing the Count Leading Zero/One (CLZ, CLO) instructions
- Logic unit for performing bitwise logical operations
- Shifter & Store Aligner

4.2.4 Multiply Divide Unit

The multiply divide unit (MDU) performs multiply and divide operations. The pipeline MDU supports execution of a 16:16 or a 32:16 on every clock cycle. 32:32 multiply operations can be issued every other clock cycle. Divide operations are implemented with a 1-bit per clock iterative algorithm and requires 35 clock cycles in worst case to complete. Early-in, the algorithm detects sign extensions of the dividend, reducing the amount of iterations. An attempt to issue subsequent MDU instruction while a divide is still active, causes a pipeline stall until the divide operation is completed.

The MUL instruction specifies that the lower 32 bits of the multiply result is placed in the register file instead of the HI/LO register pair. By avoiding the explicit move from LO (MFLO) instruction, required when using the LO register, and by supporting multiple destination registers, the throughput of multiply intensive operations is increased.

The multiply add (MADD, MADDU) and multiply subtract (MSUB, MSUBU) are used to perform the multiply add and multiply subtract operations, which are commonly used in in Digital Signal Processing (DSP) algorithms.

4.2.5 Memory Management Unit

The Memory Management Unit (MMU) of the MIPS 4Kc™ CPU is implemented as a Translation Lookaside Buffer (TLB). The MMU translates any virtual address to a physical address as well as providing memory protection mechanisms.

The MIPS 4Kc™ CPU supports three operating modes: the User Mode, the Kernel Mode and the Debug Mode. User Mode is often used for application programs. Kernel Mode is typically used for operating system tasks. Debug Mode is used for software debugging purposes.

The address translation performed by the MMU depends on the mode in which the processor is operating. The 4 Gbyte virtual memory space addressed by a 32-bit virtual address is segmented differently depending on the mode of operation (user, kernel, debug). Each of the segments are either mapped or unmapped. Mapped segments use the TLB to translate from virtual to physical addresses, while the unmapped segments have a fixed simple translation.

4.2.6 Cache System

The CPU Core incorporates on-chip instruction and data caches that can each be accessed in a single processor cycle.

- 8 Kbyte of instruction and 8 KByte data cache
- Instruction organized as 2-way set associative organization
- Data cache organized as 2-way set associative organization
- Each cache has its own 32-bit data path both caches can be accessed in the same pipeline clock cycle
- Single processor cycle access
- 16 byte cache line size, word sectored
- Cache locking on a per line basis CACHE instruction to manipulate the Data and Tag arrays of the instruction as well as data cache, including cache line locking

- Virtually indexed and physically tagged virtual to physical address translation occurs in parallel with the cache access
- Hit under refill
- Support of streaming instruction or data is forwarded during cache refill
- Non blocking prefetches
- PREF instructions supported by the data cache controller, which ar used to increase the performance by suggesting the processor
- Non blocking loads
- Supports Write-back with write-allocation and Write-through with or without write- allocation

4.2.7 EJTAG Debug Unit

All cores provide basic EJTAG support with debug mode, run control, single step and software breakpoint instruction (SDBBP) as part of the core. These features allow for the basic software debug of user and kernel code.

Additional EJTAG features include:

- Hardware breakpoints: The hardware instruction breakpoints can be configured to generate a debug exception when an instruction is executed anywhere in the virtual address space. Bit mask and Address Space Identifier (ASID) values may apply in the address compare. These breakpoints are not limited to code in RAM like the software instruction breakpoint (SDBBP). The data breakpoints can be configured to generate a debug exception on a data transaction. The data transaction may be qualified with both virtual address, data value, size and load/store transaction type. Bit mask and ASID values may apply in the address compare, and byte mask may apply in the value compare.
- A TAP, enabling communication between an EJTAG probe and the CPU through a dedicated port. This provides the possibility for debugging without debug code in the application, and for download of application code to the system.
- An optional block is EJTAG Trace which enables real-time tracing capability. The trace information can be stored to (either an on-chip trace memory, or to) an off-chip trace probe. The trace of program flow is highly flexible and can include instruction program counter as well as data addresses and data values. The trace features provides a powerful software debugging mechanism.

5 MultiPort Memory Controller (MPMC)

The MultiPort Memory Controller (MPMC) description covers:

- Feature list (**Chapter 5.1**)
- Functional description (**Chapter 5.2**)
- External Interface; described in the dedicated chapter of the different interfaces
- Registers (**Chapter 5.3**)

5.1 Feature List

The MPMC offers the following features:

- Dynamic memory interface support including SDRAM, JEDEC low-power SDRAM
- Asynchronous static memory device support including SRAM, ROM and NOR Flash with or without asynchronous page mode
- Read and write buffers to reduce latency and to improve performance
- 8-bit, 16-bit and 32-bit wide static memory support
- Static memory features include:
  - Programmable wait states
  - Output enable, and write enable delays
  - Extended wait
  - Bus turnaround delay
  - Asynchronous page mode read
- Controller supports 2K, 4K and 8K row address synchronous memory parts. That is typical 512M, 256M, 128M and 16MB parts with 8, 16 or 32DQ bits per device.

5.2 Functional Description

The following describes the MPMC's functions

5.2.1 Static Memory Controller

Static memory descriptions:

5.2.1.1 Extended Wait Transfers

The static memory controller supports extremely long transfer times. In normal use the memory transfers are timed using the **MPMC Static Wait Rd 0** and **MPMC Static Wait Wr 0** registers. These registers enable transfers with up to 32 wait states. However, if an extremely slow static memory device has to be accessed you can enable the Extended Wait(EW) bit in register **MPMC Static Config 0**. When this bit is enabled the **MPMC Static Extended Wait** register is used to time both the read and write transfers. This register enables transfers to have up to 16368 wait states.

5.2.1.2 Wait State Generation

Each bank of the MPMC must be configured for external transfer wait states in read and write accesses. This is achieved by programming the appropriate fields of the bank control registers:

- **MPMC Static Config 0**
- **MPMC Static Wait Wen 0**
- **MPMC Static Wait Oen 0**
- **MPMC Static Wait Rd 0**
- **MPMC Static Wait Wr 0**
- **MPMC Static Wait Page 0**

- **MPMC Static Wait Turn 0**
- **MPMC Static Extended Wait**

The number of cycles in which an AMBA transfer completes is controlled by two additional factors:

- Access width
- External memory width

5.2.1.3 Static Memory Read Control

The static memory read controls are described in the following:

- The delay between the assertion of the chip select and the output enable is programmable from 0 to 15 cycles using the WAITOEN bits of the **MPMC Static Wait Oen 0** registers. The output enable is always deasserted at the same time as the chip select, at the end of the transfer. The **Figure 6** shows that the WAIOEN is set to 2 in the **MPMC Static Wait Oen 0** registers.
- The read access time is determined by the number of wait state programmed for the WAITRD field of the **MPMC Static Wait Rd 0** register. The WAITTURN field in the **MPMC Static Wait Turn 0** register determines the minimum number of bus turnaround wait states added between external read and write transfers. The **Figure 7** shows that the WAITRD is set to 2 in the **MPMC Static Wait Rd 0** register.
- The MPMC supports asynchronous page mode read up to four memory transfers by updating address bits A[1] and A[0]. This feature increase the bandwidth by using a reduced access time for the read accesses that are in page mode. The first read access takes **MPMC Static Wait Rd 0** and WAITRD cycles. Subsequent read access that are in page mode take **MPMC Static Wait Page 0** and WAITPAGE cycles. The chip select and output enable lines are held during the burst, and only the lower two address bits change between sunsequent accesses. At the end of the burst the chip select and output enable lines are deasserted together. The **Figure 8** shows the page mode read with 2 WAITRD cycles and 1 WAITPAGE cycle.

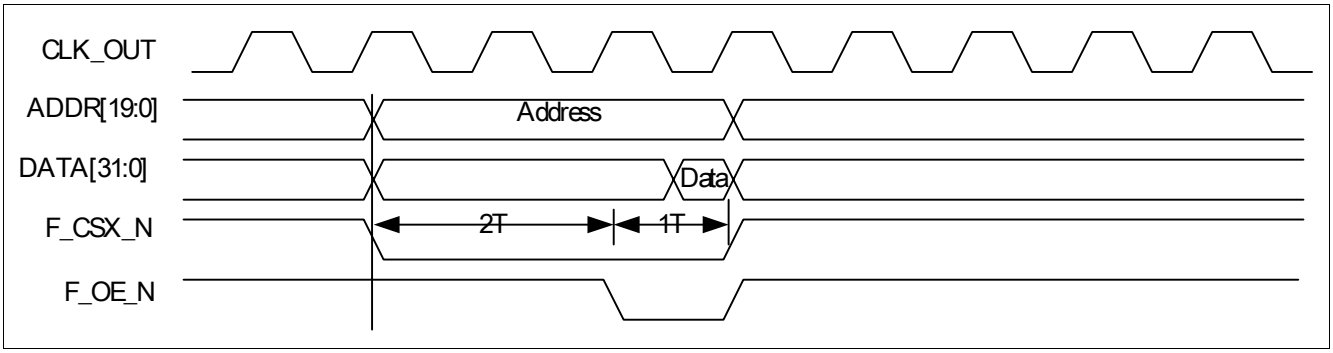


Figure 6 Read with Two Clock Delay for Read Enable

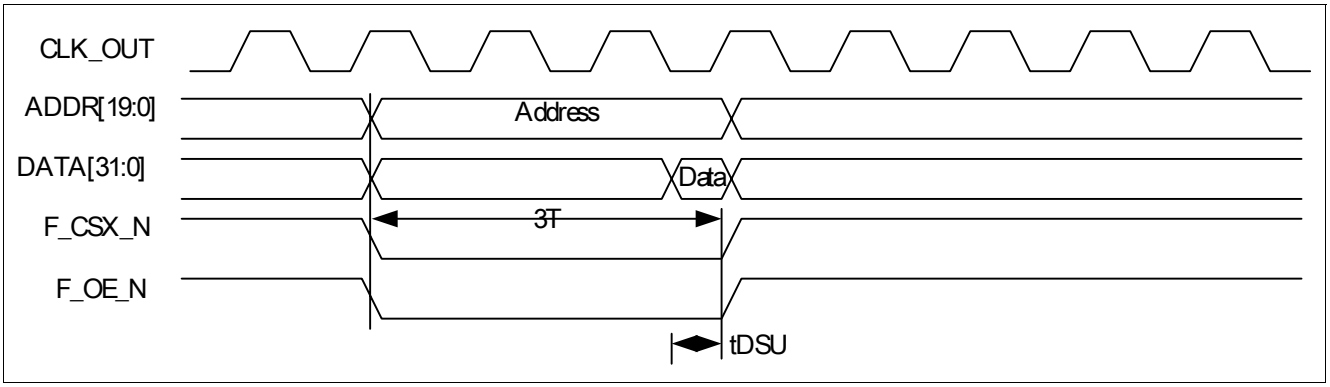


Figure 7 Read with Two Wait State

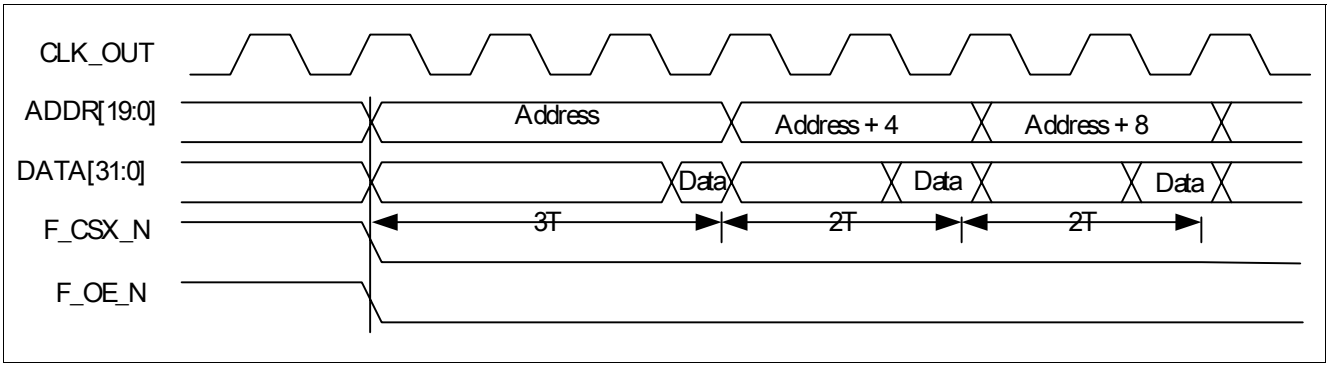


Figure 8 Asynchronous Page Mode Read with 2 Wait State and 1 Sequential Wait State

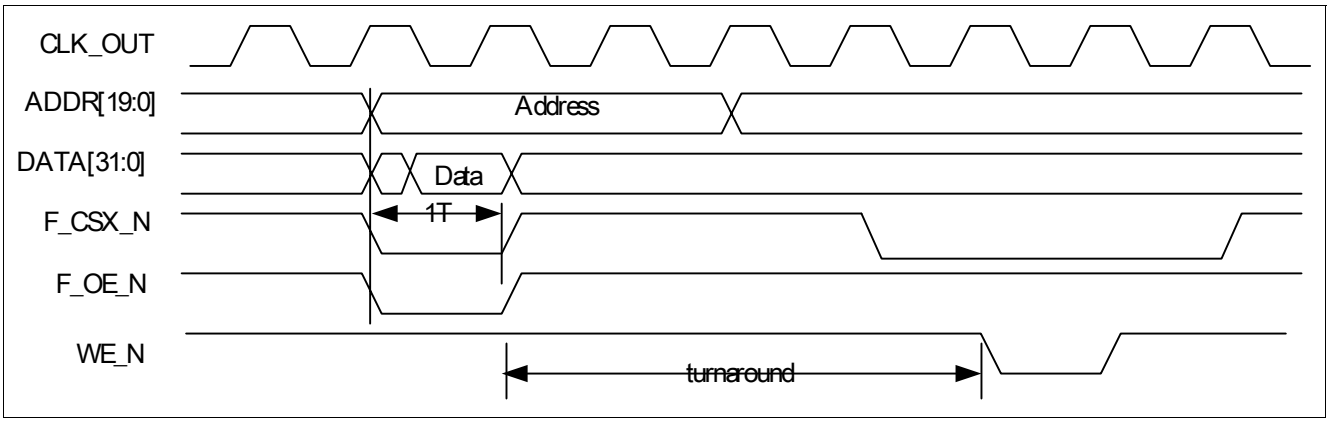


Figure 9 Bus Turnaround

5.2.1.4 Static Memory Write Control

Write timing is described in the following:

- The delay between the assertion of the chip select and the write enable is programmable from 0 to 15 cycles using the WAITWEN bits of the **MPMC Static Wait Wen 0** registers. The write enable is asserted on the rising edge of MPMCCLK after the assertion of the chip select for zero wait state. The write enable is always deasserted a cycle before the chip select, at the end of the transfer.
- The write access time is determined by the number of wait states programmed for the WAITWR field of the **MPMC Static Wait Wr 0** register. The WAITTURN field in the **MPMC Static Wait Turn 0** register determines the number of bus turnaround wait states added between external read and write transfers.

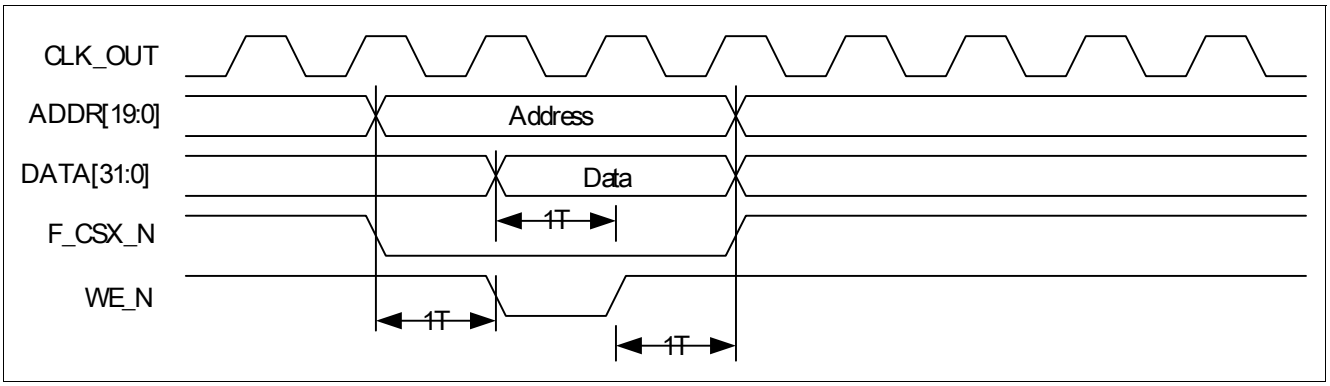


Figure 10 Write with Zero Wait State

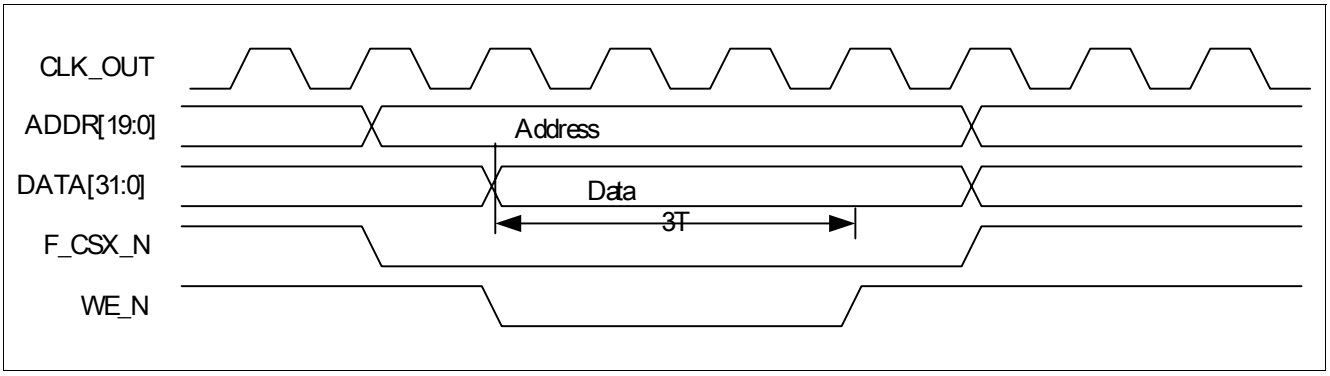


Figure 11 Write with Two Wait State

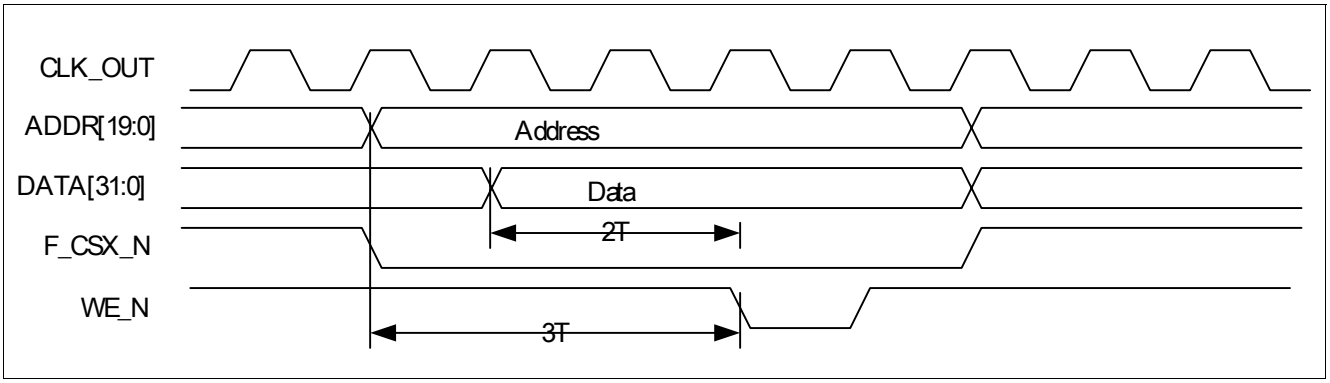


Figure 12 Write with Two Clock Delay for Wrtie Enable

5.2.2 Dynamic Memory Controller

The following describes the Dynamic Controller.

5.2.2.1 Dynamic Memory Controller Command Descriptions

The dynamic memory controller block in MPMC supports the SDRAM memory commands shown in list below:

- ACT: Opens an SDRAM row
- REF: CAS before RAS style refresh
- SREF: self-refresh
- PRE: Pre-charge, close a bank
- RD: Read from an open row, row left open
- WR: Write to an open row, row left open
- RDA: Read followed by pre-charge
- WRA: Write followed by pre-charge

The command listed above are generated automatically.

The commands in the list below are generated under software control by programming the SDRAM initialization and deep sleep mode fields of the **MPMC Dynamic Control** register.

- MRS: Mode register set, programs SDRAM mode register
- NOP: No operation, used during the SDRAM initialization sequence
- PALL: Pre-charge all, used during the SDRAM initialization sequence
- DSM: Deep sleep mode, for low-power SDRAM

5.2.2.2 Generic SDRAM Initialization Example

On power-on reset, software must initialize the MPMC and each of the dynamic memories connected to the controller. Check the dynamic memory data sheet for the start up procedure. A generic example initialization sequence is shown below:

- Wait 100 ms after the power is applied and the clocks have stabilized.
- Set the SDRAM Initialization (I) value to NOP in the **MPMC Dynamic Control** register. This automatically issues a NOP command to the SDRAM memories.
- Wait 200 ms.
- Set the SDRAM Initialization (I) value to PALL ine the **MPMC Dynamic Control** register. This automatically issues a pre-charge all instruction (PRE-ALL) to the SDRAM memories. This pre-charge all banks and places the device into the all banks idle state.
- Perform a number of refresh cycles, by writing 1 into the refresh register, **MPMC Dynamic Refresh**. This provides a memory refresh every 16 AHB clock cycles.
- Wait until eight SDRAM refresh cycles have occured (128 AHB clock cycles)
- Program the operational value into the refresh register, **MPMC Dynamic Refresh**
- Program the operational value into the latency register, **MPMC Dynamic RAS**
- Program the operational values into the configuration register, **MPMC Dynamic Config 0**. The buffers nust be disabled during initialization.
- Set the SDRAM initialization value (I) to MODE in the **MPMC Dynamic Control** register.
- Program the SDRAM memories mode register. The mode register enables the following parameters to be programmed.
  - Burst length
  - Burst type
  - CAS latency
  - Operating mode
  - Write burst mode
- Set the SDRAM initialization value (I) to NORMAL in the **MPMC Dynamic Control** register.
- Enable the buffers in the **MPMC Dynamic Config 0** configuration register. The SDRAM is now ready for normal operation.

5.3 MPMC Registers Description

The below describes the MPMC registers in great details.

Note: For ALL Reserved Registers please note: Read is defined and all must be written as zeros.

5.3.1 MPMC Registers

Table 24 Registers Address Space

Module	Base Address	End Address	Note
MPMC	1100 0000$_H$	1100 0278$_H$	

Table 25 Registers Overview

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_C	MPMC Control	000$_H$	**55**
MPMC_S	MPMC Status	004$_H$	**55**
MPMC_Conf	MPMC Configuration	008$_H$	**57**
MPMC_DC	MPMC Dynamic Control	020$_H$	**58**
MPMC_DR	MPMC Dynamic Refresh	024$_H$	**59**
MPMC_DRP	MPMC Dynamic RP	030$_H$	**60**
MPMC_DRAS	MPMC Dynamic RAS	034$_H$	**60**
MPMC_DSREX	MPMC Dynamic SREX	038$_H$	**60**
MPMC_DAPR	MPMC Dynamic APR	03C$_H$	**62**
MPMC_DDAL	MPMC Dynamic DAL	040$_H$	**62**
MPMC_DWR	MPMC Dynamic WR	044$_H$	**62**
MPMC_DRC	MPMC Dynamic RC	048$_H$	**64**
MPMC_DRFC	MPMC Dynamic RFC	04C$_H$	**64**
MPMC_DXSR	MPMC Dynamic XSR	050$_H$	**64**
MPMC_DRRD	MPMC Dynamic RRD	054$_H$	**66**
MPMC_DMRD	MPMC Dynamic MRD	058$_H$	**66**
MPMC_SEW	MPMC Static Extended Wait	080$_H$	**66**
MPMC_DC0	MPMC Dynamic Config 0	100$_H$	**68**
MPMC_DRC0	MPMC Dynamic Ras Cas 0	104$_H$	**72**
MPMC_DC1	MPMC Dynamic Config 1	120$_H$	**69**
MPMC_DRC1	MPMC Dynamic Ras Cas 1	124$_H$	**72**
MPMC_DC2	MPMC Dynamic Config 2	140$_H$	**69**
MPMC_DRC2	MPMC Dynamic Ras Cas 2	144$_H$	**72**
MPMC_DC3	MPMC Dynamic Config 3	160$_H$	**69**
MPMC_DRC3	MPMC Dynamic Ras Cas 3	164$_H$	**72**
MPMC_SC0	MPMC Static Config 0	200$_H$	**73**
MPMC_SWW0	MPMC Static Wait Wen 0	204$_H$	**75**
MPMC_SWO0	MPMC Static Wait Oen 0	208$_H$	**76**
MPMC_SWR0	MPMC Static Wait Rd 0	20C$_H$	**77**
MPMC_SWP0	MPMC Static Wait Page 0	210$_H$	**78**
MPMC_SWWR0	MPMC Static Wait Wr 0	214$_H$	**79**
MPMC_SWT0	MPMC Static Wait Turn 0	218$_H$	**80**
MPMC_SC1	MPMC Static Config 1	220$_H$	**74**
MPMC_SWW1	MPMC Static Wait Wen 1	224$_H$	**75**

Table 25 Registers Overview (cont'd)

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_SWO1	MPMC Static Wait Oen 1	228_H	76
MPMC_SWR1	MPMC Static Wait Rd 1	$22C_H$	77
MPMC_SWP1	MPMC Static Wait Page 1	230_H	78
MPMC_SWWR1	MPMC Static Wait Wr 1	234_H	79
MPMC_SWT1	MPMC Static Wait Turn 1	238_H	80
MPMC_SC2	MPMC Static Config 2	240_H	74
MPMC_SWW2	MPMC Static Wait Wen 2	244_H	75
MPMC_SWO2	MPMC Static Wait Oen 2	248_H	76
MPMC_SWR2	MPMC Static Wait Rd 2	$24C_H$	77
MPMC_SWP2	MPMC Static Wait Page 2	250_H	78
MPMC_SWWR2	MPMC Static Wait Wr 2	254_H	79
MPMC_SWT2	MPMC Static Wait Turn 2	258_H	80
MPMC_SC3	MPMC Static Config 3	260_H	74
MPMC_SWW3	MPMC Static Wait Wen 3	264_H	75
MPMC_SWO3	MPMC Static Wait Oen 3	268_H	76
MPMC_SWR3	MPMC Static Wait Rd 3	$26C_H$	77
MPMC_SWP3	MPMC Static Wait Page 3	270_H	78
MPMC_SWWR3	MPMC Static Wait Wr 3	274_H	79
MPMC_SWT3	MPMC Static Wait Turn 3	278_H	80

The register is addressed wordwise.

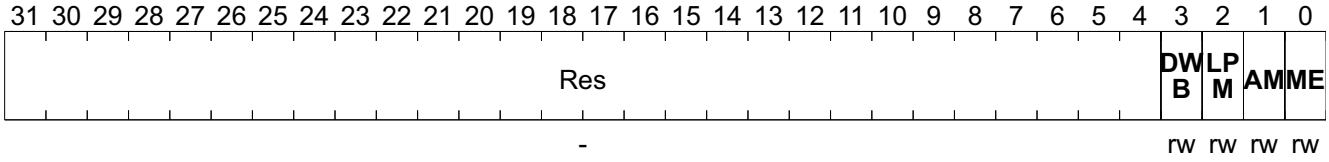
Table 26 Registers Access Types

Mode	Symbol	Description Hardware (HW)	Description Software (SW)
Basic Access Types			
read/write	rw	Register is used as input for the HW	Register is read and writable by SW
read/write virtual	rwv	Physically, there is no new register in the generated register file. The real readable and writable register resides in the attached hardware.	Register is read and writable by SW (same as rw type register)
read	r	Register is written by HW (register between input and output -> one cycle delay)	Value written by SW is ignored by HW; that is, SW may write any value to this field without affecting HW behavior
read only	ro	Same as r type register	Same as r type register
read virtual	rv	Physically, there is no new register in the generated register file. The real readable register resides in the attached hardware.	Value written by SW is ignored by HW; that is, SW may write any value to this field without affecting HW behavior (same as r type register)
write	w	Register is written by software and affects hardware behavior with every write by software.	Register is writable by SW. When read, the register does not return the value that has been written previously, but some constant value instead.
write virtual	wv	Physically, there is no new register in the generated register file. The real writable register resides in the attached hardware.	Register is writable by SW (same as w type register)
read/write hardware affected	rwh	Register can be modified by hardware and software at the same time. A priority scheme decides, how the value changes with simultaneous writes by hardware and software.	Register can be modified by HW and SW, but the priority SW versus HW has to be specified. SW can read the register.

5.3.1.1 Registers Description

MPMC Control

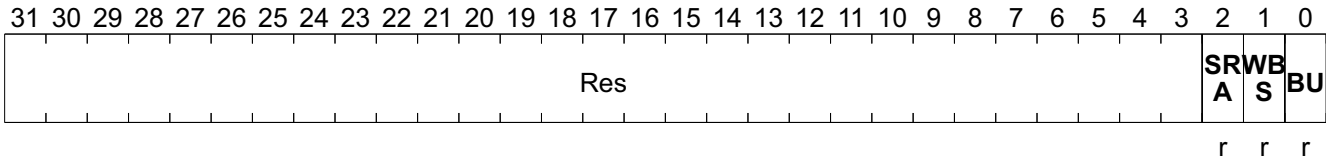
MPMC_C	Offset	Reset Value
MPMC Control	000_H	1_H



Field	Bits	Type	Description
Res	31:4	-	**Not applicable**
DWB	3	rw	**Drain Write Buffers** 0_B , Buffers operate normally (reset value on nPOR, and HRESETn) 1_B , Drain write buffers.
LPM	2		**Low-Power Mode** Indicate normal, or low-power mode. Entering low-power mode reduces memory controller power consumption. Dynamic memory is refreshed as necessary. The memory controller returns to normal function mode by clearing the low-power mode bit (L), or by system, or power-on reset. 0_B , Normal mode (reset value on nPOR, and HRESETn) 1_B , Low-power mode.
AM	1		**Address Mirror** Indicates normal or reset memory map. Static memory chip select 1 is mirrorred onto chip select 0 and chip select 4 (reset value on nPOR). On power-on reset, chip select 1 is mirrorred to both chip select 0 and chip select 1 and chip 4 memory areas. Clearing the M bit enables chip select 0 and chip select 4 memory to be accessed. 0_B , Normal memory map 1_B , Reset meory map.
ME	0		**MPMC Enable** Indicates if the PrimeCell MPMC is enabled or disabled. Disabling the PrimeCell MPMC reduces power consumption. When the memory controller is disabled the memory is not refreshed. The memory controller is enabled by setting the enable bit, or by system, or power-on reset. 0_B , Disabled 1_B , Enabled (reset value on nPOR, and HRESETn).

MPMC Status

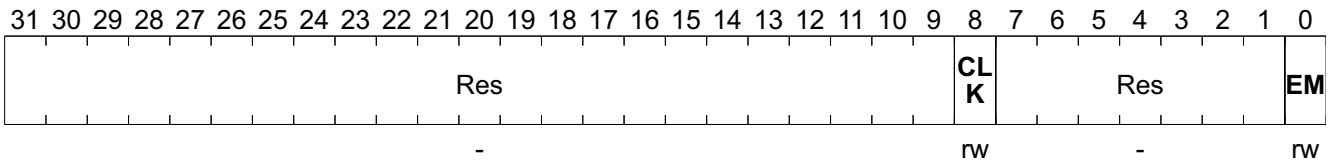
MPMC_S	Offset	Reset Value
MPMC Status	004_H	0_H



Field	Bits	Type	Description
Res	31:3		**Reserved** Not applicable
SRA	2	r	**Self-Refresh Acknowledge** This read only bit indicates the operating mode of the MPMC. 0_B , Normal mode (reset value on nPOR) 1_B , Self-refresh acknowledge.
WBS	1		**Write Buffer Status** This read only bit enables the PrimeCell MPMC to enter low-power mode or disabled mode cleanly. 0_B , Write buffers empty (reset value on nPOR) 1_B , Write buffers contain data.
BU	0		**Busy** This read-only bit is used to ensure that the memory controller enters the low-power or disabled mode cleanly. 0_B , MPMC is idle (reset value on nPOR, and HRESETn) 1_B , MPMC is busy performing memory transactions.

MPMC Configuration

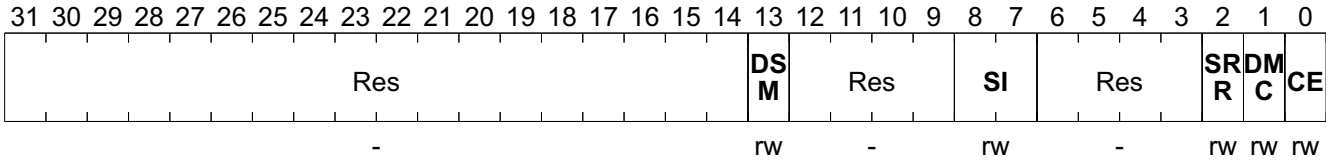
MPMC_Conf	Offset	Reset Value
MPMC Configuration	008_H	0_H



Field	Bits	Type	Description
Res	31:9	-	**Reserved** Read undefined, must be written as zeros.
CLK	8	rw	**Clock Ratio** HCLK: MPMCCLKOUT3:0 ratio. 0_B , 1:1 (reset value on nPOR) 1_B , 1:2.
Res	7:1	-	**Reserved** Read undefined, must be written as zeros.
EM	0	rw	**Endian Mode** The value of the endian bit on power-on-reset (nPOR) is determined by the MPMCBIGENDIAN signal. This value can be overridden by software. This field is unaffected by the AHB reset (HRESETn). 0_B , Little-endian mode 1_B , Big-endian mode.

MPMC Dynamic Control

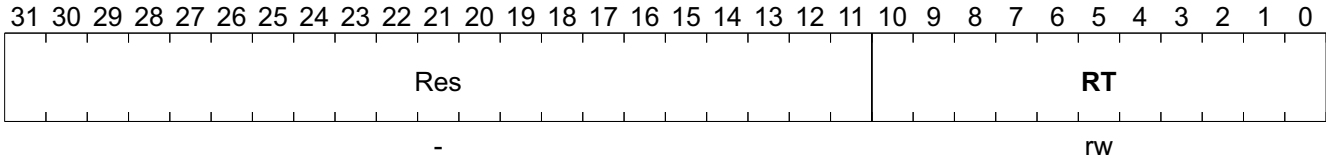
MPMC_DC	Offset	Reset Value
MPMC Dynamic Control	020_H	2_H



Field	Bits	Type	Description
Res	31:14	-	**Reserved** Read undefined. Must be written as zeros.
DSM	13	rw	**Low-Power SDRAM Deep Sleep Mode** 0_B , Normal operation (reset value on nPOR) 1_B , Enter deep power down mode.
Res	12:9	-	**Reserved** Read undefined. Must be written as zeros.
SI	8:7	rw	**SDRAM Initialization** 00_B , Issue SDRAM NORMAL operation command (reset value on nPOR) 01_B , Issue SDRAM MODE command 10_B , Issue SDRAM PALL (precharge all) command 11_B , Issue SDRAM NOP (no operation) command.
Res	6:3	-	**Reserved** Read undefined. Must be written as zeros.
SRR	2	rw	**Self-Refresh Request (SR)** By writing 1 to this bit self-refresh can be entered under software control. Writing 0 to this bit returns the MPMC to normal mode. The self-refresh acknowledge bit in the MPMCStatus register must be polled to discover the current operating mode of the MPMC. 0_B , Normal mode (reset value on nPOR) 1_B , Enter self-refresh mode.
DMC	1	rw	**Dynamic Memory Clock Control** When clock control is LOW the output clock MPMCCLKOUT is stopped when there are no SDRAM transactions. The clock is also stopped during self-refresh mode. 0_B , MPMCCLKOUT stops when all SDRAMs are idle and during self-refrsh mode 1_B , MPMCCLKOUT runs continuously (reset value on nPOR).
CE	0	rw	**Dynamic Memory Clock Enable** 0_B , Clock enable of devices are deasserted to save power (reset value on nPOR) 1_B , All clock enables are driven HIGH continuously.

MPMC Dynamic Refresh

MPMC_DR	Offset	Reset Value
MPMC Dynamic Refresh	024_H	0_H

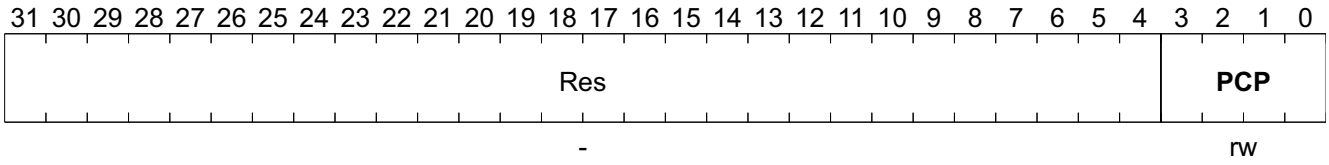


Field	Bits	Type	Description
Res	31:11	-	**Reserved** Read undefined. Must be written as zeros.
RT	10:0	rw	**Refresh Timer** 0_D , Refresh disabled (reset value on nPOR) 1_D , 16 HCLK ticks between SDRAM refresh cycles $\cdots_D$, n_D , n x16 HCLK ticks between SDRAM refresh cycles.

MPMC Dynamic RP

Note: The delay is in MPMCCLK cycles.

MPMC_DRP	Offset	Reset Value
MPMC Dynamic RP	030_H	F_H

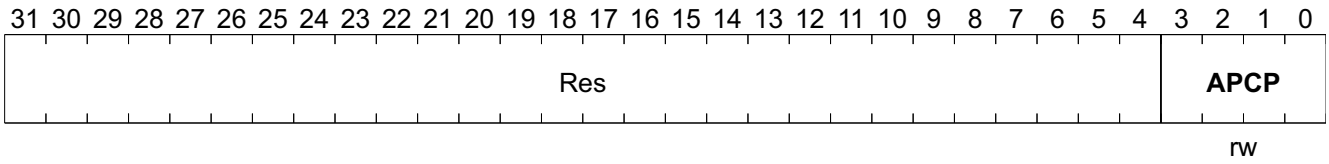


Field	Bits	Type	Description
Res	31:4	-	**Reserved** Read undefined. Must be written as zeros.
PCP	3:0	rw	**Precharge Command Period** 0_H , 1 clock cycle …$_H$, F_H , 16 clock cycles (reset value on nPOR)

MPMC Dynamic RAS

Note: The delay is in MPMCCLK cycles.

MPMC_DRAS	Offset	Reset Value
MPMC Dynamic RAS	034_H	F_H

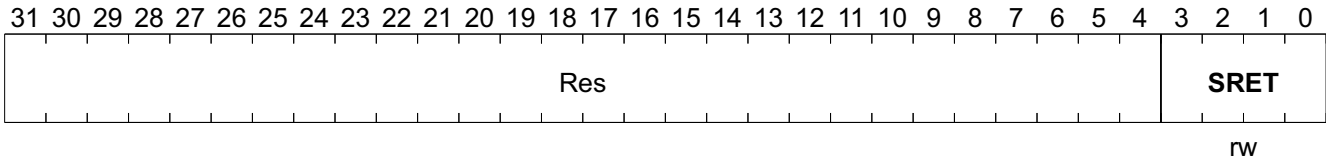


Field	Bits	Type	Description
Res	31:4		**Reserved** Read undefined. Must be written as zeros.
APCP	3:0	rw	**Active to Precharge Command Period** 0_H , 1 clock cycle …$_H$, F_H , 16 clock cycles (reset value on nPOR)

MPMC Dynamic SREX

Note: The delay is in MPMCCLK cycles.

MPMC_DSREX	Offset	Reset Value
MPMC Dynamic SREX	038_H	F_H

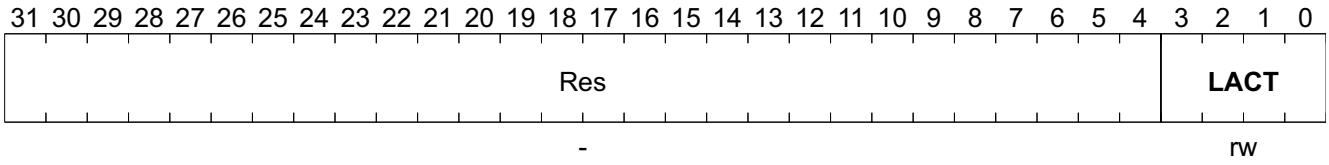


Field	Bits	Type	Description
Res	31:4		**Reserved** Read undefined. Must be written as zeros.
SRET	3:0	rw	**Self-Refresh Exit Time** 0_H , 1 clock cycle ...H , F_H , 16 clock cycles (reset value on nPOR)

MPMC Dynamic APR

Note: The delay is in MPMCCLK cycles.

MPMC_DAPR	Offset	Reset Value
MPMC Dynamic APR	$03C_H$	F_H

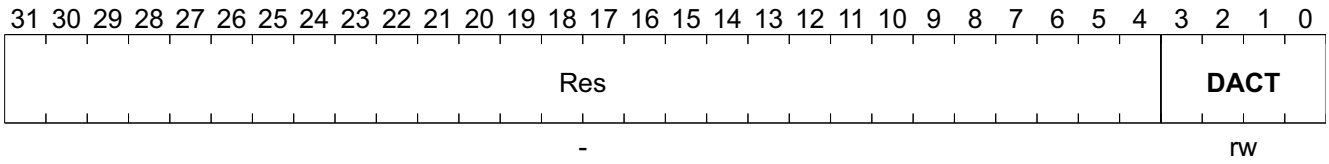


Field	Bits	Type	Description
Res	31:4	-	**Reserved** Read undefined. Must be written as zeros.
LACT	3:0	rw	**Last-Data-Out to Active Command Time** 0_H , 1 clock cycle ...H , F_H , 16 clock cycles (reset value on nPOR)

MPMC Dynamic DAL

Note: The delay is in MPMCCLK cycles.

MPMC_DDAL	Offset	Reset Value
MPMC Dynamic DAL	040_H	F_H

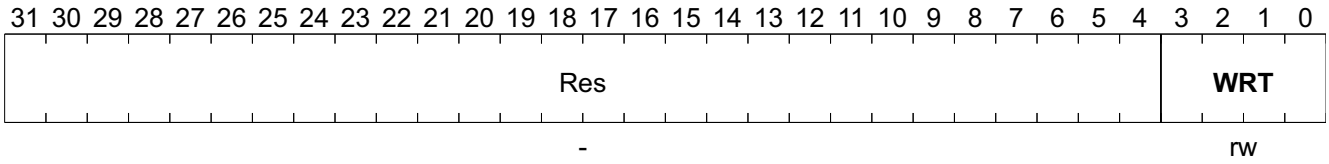


Field	Bits	Type	Description
Res	31:4	-	**Reserved** Read undefined. Must be written as zeros.
DACT	3:0	rw	**Data-In to Active Command Time** 0_H , 1 clock cycle ...H , F_H , 16 clock cycles (reset value on nPOR)

MPMC Dynamic WR

Note: The delay is in MPMCCLK cycles.

MPMC_DWR	Offset	Reset Value
MPMC Dynamic WR	044_H	F_H

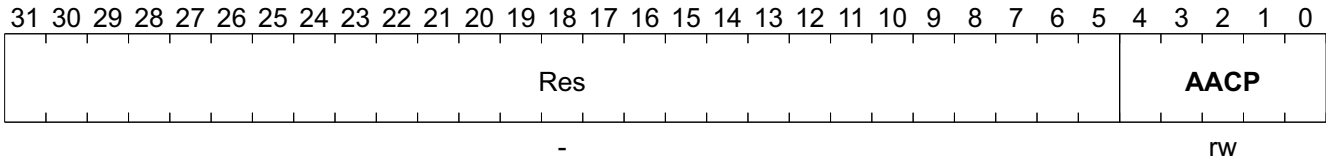


Field	Bits	Type	Description
Res	31:4	-	**Reserved**
WRT	3:0	rw	**Write Recovery Time** 0_H , 1 clock cycle ...H , F_H , 16 clock cycles (reset value on nPOR)

MPMC Dynamic RC

Note: The delay is in MPMCCLK cycles.

MPMC_DRC	Offset	Reset Value
MPMC Dynamic RC	048_H	$1F_H$

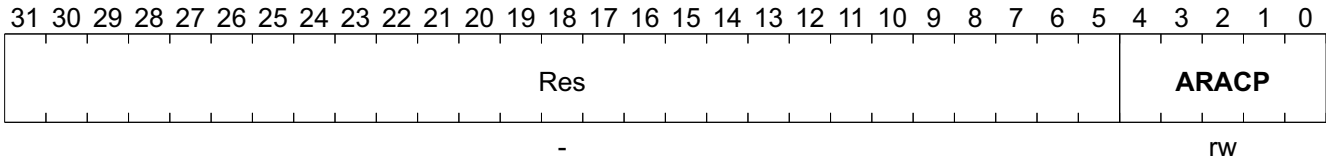


Field	Bits	Type	Description
Res	31:5	-	**Reserved**
AACP	4:0	rw	**Active to Active Command Period** 0_H , 1 clock cycle ...H , $1F_H$, 32 clock cycles (reset value on nPOR)

MPMC Dynamic RFC

Note: The delay is in MPMCCLK cycles.

MPMC_DRFC	Offset	Reset Value
MPMC Dynamic RFC	$04C_H$	$1F_H$

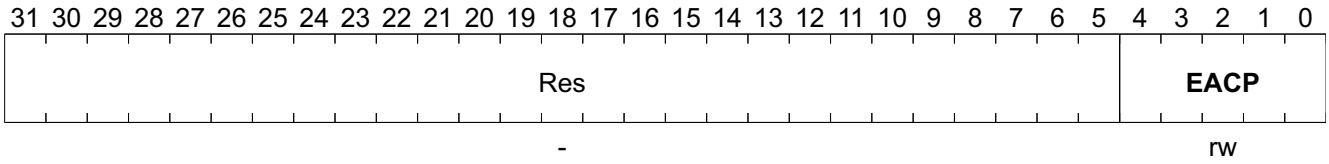


Field	Bits	Type	Description
Res	31:5	-	**Reserved**
ARACP	4:0	rw	**Auto Refresh Period and Auto Refresh to Active Command Period** 0_H , 1 clock cycle ...H , $1F_H$, 32 clock cycles (reset value on nPOR)

MPMC Dynamic XSR

Note: The delay is in MPMCCLK cycles.

MPMC_DXSR	Offset	Reset Value
MPMC Dynamic XSR	050_H	$1F_H$

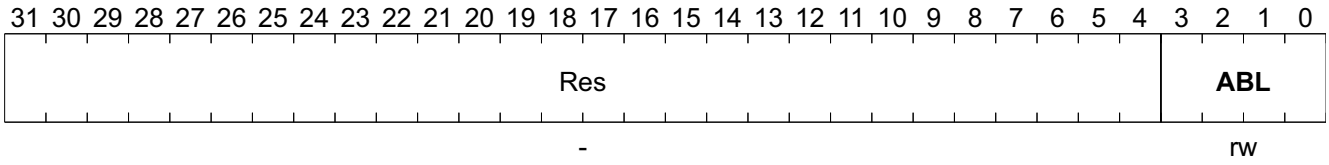


Field	Bits	Type	Description
Res	31:5	-	**Reserved**
EACP	4:0	rw	**Exit Self-Refresh to Active Command Period** 0_H , 1 clock cycle ...H , $1F_H$, 32 clock cycles (reset value on nPOR)

MPMC Dynamic RRD

Note: The delay is in MPMCCLK cycles.

MPMC_DRRD	Offset	Reset Value
MPMC Dynamic RRD	054_H	F_H

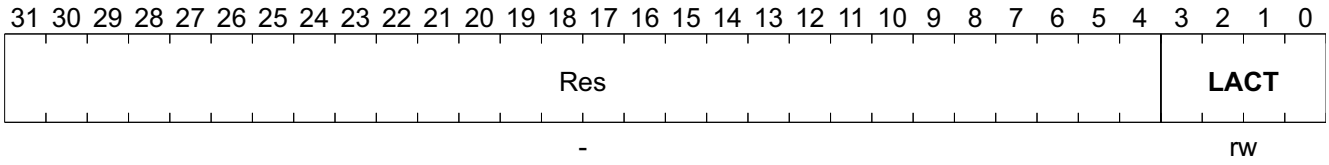


Field	Bits	Type	Description
Res	31:4	-	**Reserved**
ABL	3:0	rw	**Active Bank A to Active Bank B Latency** 0_H , 1 clock cycle ...H , F_H , 16 clock cycles (reset value on nPOR)

MPMC Dynamic MRD

Note: The delay is in MPMCCLK cycles.

MPMC_DMRD	Offset	Reset Value
MPMC Dynamic MRD	058_H	F_H

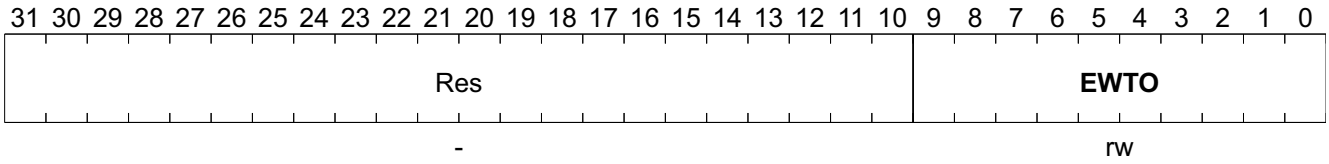


Field	Bits	Type	Description
Res	31:4	-	**Reserved**
LACT	3:0	rw	**Load Mode Register to Active Command Time** 0_H , 1 clock cycle ...H , F_H , 16 clock cycles (reset value on nPOR)

MPMC Static Extended Wait

Note: The delay is in HCLK cycles.

MPMC_SEW	Offset	Reset Value
MPMC Static Extended Wait	080_H	0_H

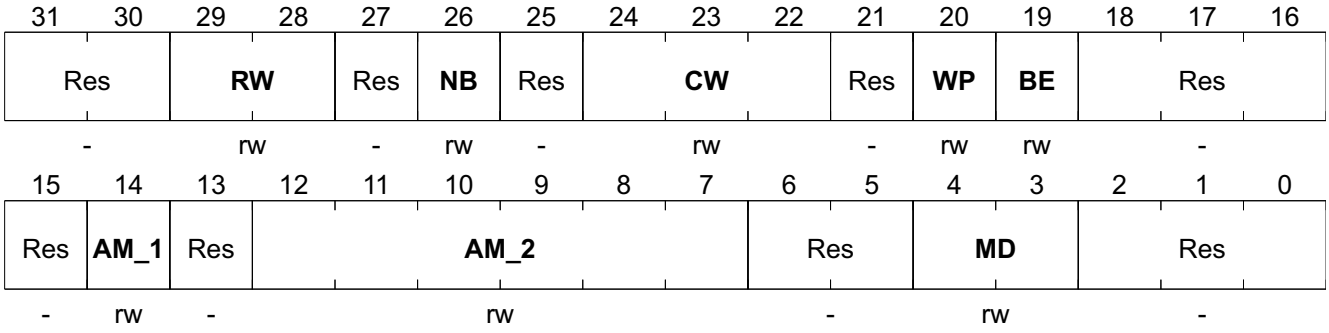


Field	Bits	Type	Description
Res	31:10	-	**Reserved**
EWTO	9:0	rw	**External Wait Time Out** 0_D , 16 clock cycle (reset value on nPOR) $\cdots_D$, n_D , (n+1) x16 clock cycles *Note: n = 0 to $3FF_H$*

MPMC Dynamic Config 0

Other Reserved Registers have the same structure and characteristics as **MPMC_DC0**; the names and offset addresses are listed in **Table 27**.

MPMC_DC0	Offset	Reset Value
MPMC Dynamic Config 0	100_H	0_H



Field	Bits	Type	Description
Res	31:30	-	**Reserved**
RW	29:28	rw	**Row Width** 00_B , 11-bit (reset value on nPOR) 01_B , 12-bit 10_B , 13-bit 11_B , Reserved
Res	27	-	**Reserved**
NB	26	rw	**Number of Banks** 0_B , Two banks (reset value on nPOR) 1_B , Four banks
Res	25	-	**Reserved**
CW	24:22	rw	**Column Width** 000_B , 6-bit (reset value on nPOR) 001_B , 7-bit 010_B , 8-bit 011_B , 9-bit 100_B , 10-bit 101_B , 11-bit 110_B , Reserved 111_B , Reserved
Res	21	-	**Reserved**
WP	20	rw	**Write Protect** 0_B , Writes not protected (reset value on nPOR) 1_B , Write protected.

Field	Bits	Type	Description
BE	19	rw	**Buffer Enable** 0_B , Buffer disabled for accesses to this chip select (reset value on nPOR) 1_B , Buffer enabled for accesses to this chip select.
Res	18:15	-	**Reserved**
AM_1	14	rw	**Address Mapping** See **Figure 13**, **Figure 14** and **Figure 15**. 0_B , Reset value on nPOR
Res	13	-	**Reserved**
AM_2	12:7	rw	**Address Mapping** See **Figure 13**, **Figure 14** and **Figure 15**. 00000000_B, Reset value on nPOR
Res	6:5	-	**Reserved**
MD	4:3	rw	**Memory Device** 00_B , SDRAM(reset value on nPOR) 01_B , Low-power SDRAM 10_B , Reserved 11_B , Reserved
Res	2:0	-	**Reserved**

Similar Registers

Table 27 MPMC_DCx Registers

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_DC1	MPMC Dynamic Config 1	120_H	
MPMC_DC2	MPMC Dynamic Config 2	140_H	
MPMC_DC3	MPMC Dynamic Config 3	160_H	

Address mapping

[14]	[12]	[11:9]	[8:7]	Description
16-bit external bus High performance address mapping (Row, Bank, Column)				
0	0	000	00	16Mb (2Mx8), 2 banks, row length = 11, column length = 9
0	0	000	01	16Mb (1Mx16), 2 banks, row length = 11, column length = 8
0	0	001	00	64Mb (8Mx8), 4 banks, row length = 12, column length = 9
0	0	001	01	64Mb (4Mx16), 4 banks, row length = 12, column length = 8
0	0	010	00	128Mb (16Mx8), 4 banks, row length = 12, column length = 10
0	0	010	01	128Mb (8Mx16), 4 banks, row length = 12, column length = 9
0	0	011	00	256Mb (32Mx8), 4 banks, row length = 13, column length = 10
0	0	011	01	256Mb (16Mx16), 4 banks, row length = 13, column length = 9
0	0	100	00	512Mb (64Mx8), 4 banks, row length = 13, column length = 11
0	0	100	01	512Mb (32Mx16), 4 banks, row length = 13, column length = 10
16-bit external bus Low-power SDRAM address mapping (Bank, Row, Column)				
0	1	000	00	16Mb (2Mx8), 2 banks, row length = 11, column length = 9
0	1	000	01	16Mb (1Mx16), 2 banks, row length = 11, column length = 8
0	1	001	00	64Mb (8Mx8), 4 banks, row length = 12, column length = 9
0	1	001	01	64Mb (4Mx16), 4 banks, row length = 12, column length = 8
0	1	010	00	128Mb (16Mx8), 4 banks, row length = 12, column length = 10
0	1	010	01	128Mb (8Mx16), 4 banks, row length = 12, column length = 9
0	1	011	00	256Mb (32Mx8), 4 banks, row length = 13, column length = 10
0	1	011	01	256Mb (16Mx16), 4 banks, row length = 13, column length = 9
0	1	100	00	512Mb (64Mx8), 4 banks, row length = 13, column length = 11
0	1	100	01	512Mb (32Mx16), 4 banks, row length = 13, column length = 10

Figure 13 Address Map, Part 1

1	0	001	00	64Mb (8Mx8), 4 banks, row length = 12, column length = 9
1	0	001	01	64Mb (4Mx16), 4 banks, row length = 12, column length = 8
1	0	001	10	64Mb (2Mx32), 4 banks, row length = 11, column length = 8
1	0	010	00	128Mb (16Mx8), 4 banks, row length = 12, column length = 10
1	0	010	01	128Mb (8Mx16), 4 banks, row length = 12, column length = 9
1	0	010	10	128Mb (4Mx32), 4 banks, row length = 12, column length = 8
1	0	011	00	256Mb (32Mx8), 4 banks, row length = 13, column length = 10
1	0	011	01	256Mb (16Mx16), 4 banks, row length = 13, column length = 9
1	0	011	10	256Mb (8Mx32), 4 banks, row length = 13, column length = 8
1	0	100	00	512Mb (64Mx8), 4 banks, row length = 13, column length = 11
1	0	100	01	512Mb (32Mx16), 4 banks, row length = 13, column length = 10
32-bit external bus Low-power SDRAM address mapping (Bank, Row, Column)				
1	1	000	00	16Mb (2Mx8), 2 banks, row length = 11, column length = 9
1	1	000	01	16Mb (1Mx16), 2 banks, row length = 11, column length = 8
1	1	001	00	64Mb (8Mx8), 4 banks, row length = 12, column length = 9
1	1	001	01	64Mb (4Mx16), 4 banks, row length = 12, column length = 8
1	1	001	10	64Mb (2Mx32), 4 banks, row length = 11, column length = 8
1	1	010	00	128Mb (16Mx8), 4 banks, row length = 12, column length = 10
1	1	010	01	128Mb (8Mx16), 4 banks, row length = 12, column length = 9
1	1	010	10	128Mb (4Mx32), 4 banks, row length = 12, column length = 8
1	1	011	00	256Mb (32Mx8), 4 banks, row length = 13, column length = 10
1	1	011	01	256Mb (16Mx16), 4 banks, row length = 13, column length = 9
1	1	011	10	256Mb (8Mx32), 4 banks, row length = 13, column length = 8
1	1	100	00	512Mb (64Mx8), 4 banks, row length = 13, column length = 11
1	1	100	01	512Mb (32Mx16), 4 banks, row length = 13, column length = 10

Figure 14 Address Map, Part 2

32-bit external bus High performance address mapping (Row, Bank, Column)				
1	0	000	00	16Mb (2Mx8), 2 banks, row length = 11, column length = 9
1	0	000	01	16Mb (1Mx16), 2 banks, row length = 11, column length = 8

Figure 15 Address Map, Part 3

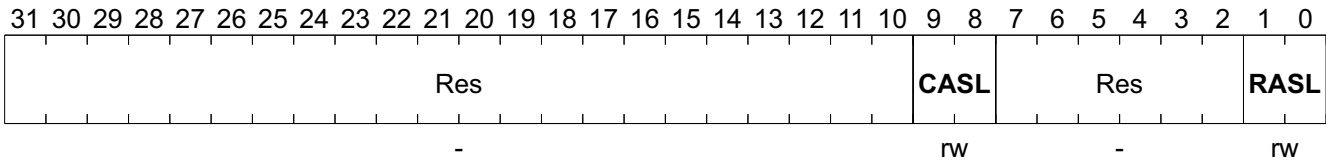
MPMC Dynamic Ras Cas 0

Other Reserved Registers have the same structure and characteristics as **MPMC_DRC0**; the names and offset addresses are listed in **Table 28**.

Notes

1. *The RAS to CAS latency (RAS) and CAS latency (CAS) are both defined in MPMCCLK cycles.*

MPMC_DRC0	Offset	Reset Value
MPMC Dynamic Ras Cas 0	104_H	303_H



Field	Bits	Type	Description
Res	31:10	-	**Reserved**
CASL	9:8	rw	**CAS Latency** 00_B , Reserved 01_B , One clock cycle(a) 10_B , Two clock cycle 11_B , Three clock cycle(reset value on nPOR).
Res	7:2	-	**Reserved**
RASL	1:0	rw	**RAS Latency** Active to read or write delay 00_B , Reserved 01_B , One clock cycle(a) 10_B , Two clock cycles 11_B , Three clock cycles (reset value on nPOR).

Similar Registers

Table 28 MPMC_DRCx Registers

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_DRC1	MPMC Dynamic Ras Cas 1	124_H	
MPMC_DRC2	MPMC Dynamic Ras Cas 2	144_H	
MPMC_DRC3	MPMC Dynamic Ras Cas 3	164_H	

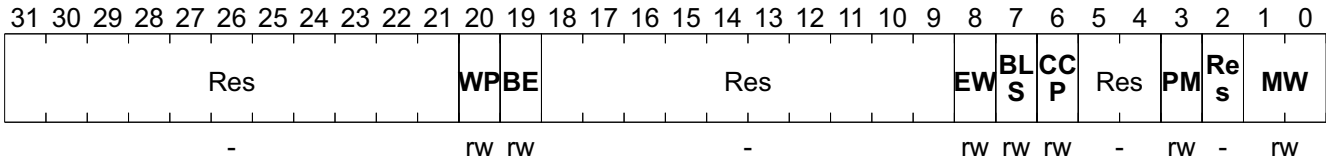
MPMC Static Config 0

Other Reserved Registers have the same structure and characteristics as **MPMC_SC0**; the names and offset addresses are listed in **Table 29**.

Notes

1. *Offset = 200_H, 220_H is for F_CS1_N and F_CS0_N respectively.*
2. *Synchronous burst mode memory devices are not supported.*

MPMC_SC0	Offset	Reset Value
MPMC Static Config 0	200_H	0_H



Field	Bits	Type	Description
Res	31:21	-	**Reserved**
WP	20	rw	**Write Protect** 0_B , Writes not protected (reset value on nPOR) 1_B , Write protected
BE	19		**Buffer Enable** 0_B , Write buffer disabled (reset value on nPOR) 1_B , Write buffer enabled
Res	18:9	-	**Reserved**
EW	8	rw	**Extended Wait** 0_B , Extended wait disabled (reset value on nPOR) 1_B , Extended wait enabled
BLS	7		**Byte Lane State** 0_B , For reads all the bits in nMPMCBLSOUT[3:0] are HIGH(reset value on nPOR).For writes the respective active bits innMPMCBLSOUT[3:0] are LOW. 1_B , For reads the respective active bits in nMPMCBLSOUT[3:0] are LOW. For writes the respectiveactive bits in nMPMCBLSOUT[3:0] are LOW.
CCP	6	rw	**Chip Celect Polarity** The value of the chip select polarity on power-on-reset(nPOR) is determined by the relevant MPMCSxPOL signal. This value can be overridden by software. This field isUnaffected by AHB reset (HRESETn). 0_B , Active LOW chip select 1_B , Active HIGH chip select
Res	5:4	-	**Reserved**
PM	3	rw	**Page Mode** 0_B , Disabled (reset value on nPOR) 1_B , Async page mode four enabled.

Field	Bits	Type	Description
Res	2	-	**Reserved**
MW	1:0	rw	**Memory Width** The value of the F_CS0_N controlled memory width field is determined by reset latched value of A[18:17] 00_B , 8 bit 01_B , 16 bit 10_B , 32 bit 11_B , Reserved.

Similar Registers

Table 29 MPMC_SCx Registers

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_SC1	MPMC Static Config 1	220_H	
MPMC_SC2	MPMC Static Config 2	240_H	
MPMC_SC3	MPMC Static Config 3	260_H	

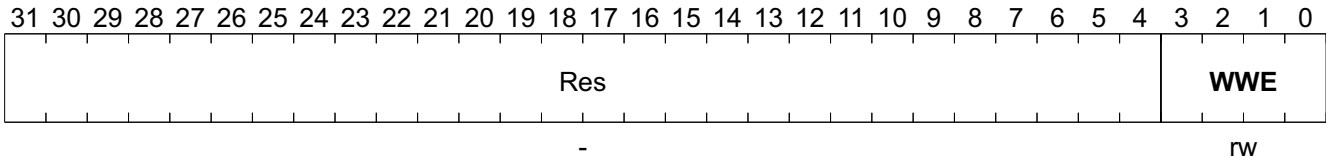
MPMC Static Wait Wen 0

Other Reserved Registers have the same structure and characteristics as **MPMC_SWW0**; the names and offset addresses are listed in **Table 30**.

Notes

1. *Offset = 204_H, 224_H is for F_CS1_N and F_CS0_N respectively.*
2. *The delay is (WAITWEN+1) x tHCLK.*

MPMC_SWW0	Offset	Reset Value
MPMC Static Wait Wen 0	204_H	0_H



Field	Bits	Type	Description
Res	31:4	-	**Reserved**
WWE	3:0	rw	**Wait Write Enable** Delay from chip select assertion to write enable. 0000_B, One HCLK cycle delay between assertion of chip select and write enable (reset value on nPOR) 0001 to 1111_B, =(n+1) HCLK cycle delay.

Similar Registers

Table 30 MPMC_SWWx Registers

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_SWW1	MPMC Static Wait Wen 1	224_H	
MPMC_SWW2	MPMC Static Wait Wen 2	244_H	
MPMC_SWW3	MPMC Static Wait Wen 3	264_H	

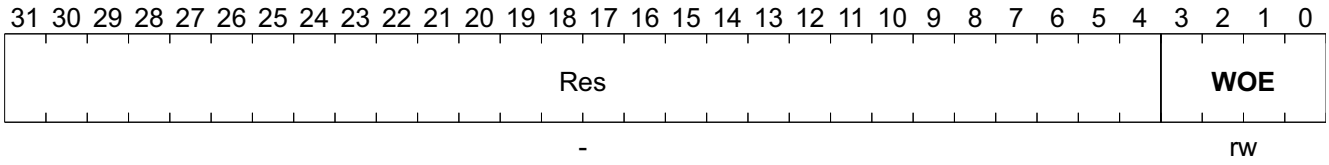
MPMC Static Wait Oen 0

Other Reserved Registers have the same structure and characteristics as **MPMC_SWO0**; the names and offset addresses are listed in **Table 31**.

Notes

1. *Offset = 208_H, 228_H is for F_CS1_N and F_CS0_N respectively.*
2. *The delay is WAITOEN x tHCLK.*

MPMC_SWO0	Offset	Reset Value
MPMC Static Wait Oen 0	**208_H**	**0_H**



Field	Bits	Type	Description
Res	31:4	-	**Reserved**
WOE	3:0	rw	**Wait Output Enable** Delay from chip select assertion to output enable. 0000_B, No delay (reset value on nPOR) 0001 to 1111_B, n cycle delay

Similar Registers

Table 31 MPMC_SWOx Registers

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_SWO1	MPMC Static Wait Oen 1	228_H	
MPMC_SWO2	MPMC Static Wait Oen 2	248_H	
MPMC_SWO3	MPMC Static Wait Oen 3	268_H	

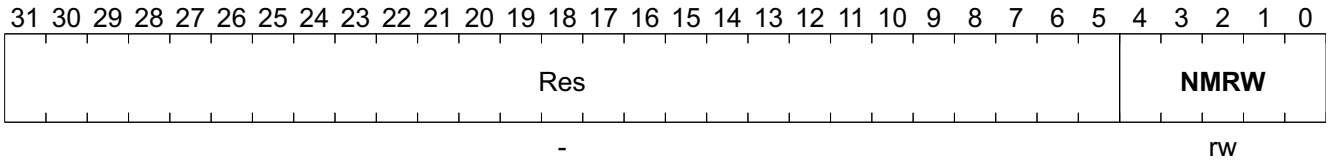
MPMC Static Wait Rd 0

Other Reserved Registers have the same structure and characteristics as **MPMC_SWR0**; the names and offset addresses are listed in **Table 32**.

Notes

1. *Offset = $20C_H$, $22C_H$ is for F_CS1_N and F_CS0_N respectively.*
2. *For non-sequential reads, the wait state time is (WAITRD+1) x tHCLK.*

MPMC_SWR0	Offset	Reset Value
MPMC Static Wait Rd 0	**$20C_H$**	**$1F_H$**



Field	Bits	Type	Description
Res	31:5	-	**Reserved**
NMRW	4:0	rw	**Nonpage Mode Read Wait** Nonpage mode read wait states or asynchronous page mode read first access wait state. **Nonpage Mode** 00000 to 11110_B, (n+1) HCLK cycles for read accesses 11111_B, 32 HCLK cycles for read accesses(reset value on nPOR). **Asynchronous Page Mode Read, First Read Only** 00000 to 11110_B, (n+1) HCLK cycles for burst read accesses 11111_B, 32 HCLK cycles for page read accesses (reset value on nPOR)

Similar Registers

Table 32 MPMC_SWRx Registers

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_SWR1	MPMC Static Wait Rd 1	$22C_H$	
MPMC_SWR2	MPMC Static Wait Rd 2	$24C_H$	
MPMC_SWR3	MPMC Static Wait Rd 3	$26C_H$	

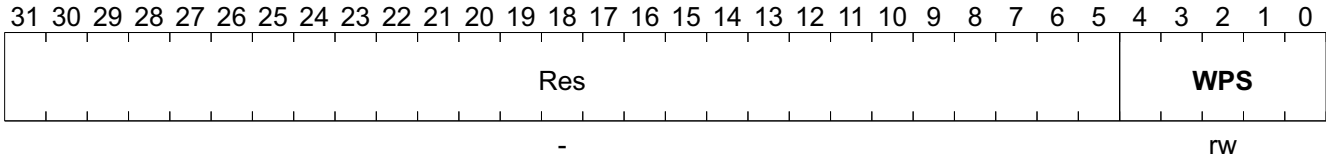
MPMC Static Wait Page 0

Other Reserved Registers have the same structure and characteristics as **MPMC_SWP0**; the names and offset addresses are listed in **Table 33**.

Notes

1. *Offset = 210_H, 230_H is for F_CS1_N and F_CS0_N respectively.*
2. *For asynchronous page mode read for sequential reade,the wait state time for page mode accesses after the first read is (WAITPAGE+1) x tHCLK.*

MPMC_SWP0	Offset	Reset Value
MPMC Static Wait Page 0	**210_H**	**$1F_H$**



Field	Bits	Type	Description
Res	31:5	-	**Reserved**
WPS	4:0	rw	**Asynchronous Page Mode Read After the First Access Wait States** Number of wait states for asynchronous page mode read accesses after the first read. 00000 to 11110_B, (n+1) HCLK cycle read access time 11111_B, 32 HCLK cycle read access time (reset value on nPOR).

Similar Registers

Table 33 MPMC_SWPx Registers

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_SWP1	MPMC Static Wait Page 1	230_H	
MPMC_SWP2	MPMC Static Wait Page 2	250_H	
MPMC_SWP3	MPMC Static Wait Page 3	270_H	

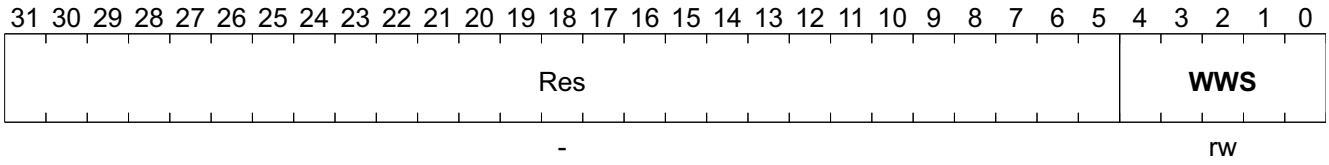
MPMC Static Wait Wr 0

Other Reserved Registers have the same structure and characteristics as **MPMC_SWWR0**; the names and offset addresses are listed in **Table 34**.

Notes

1. *Offset = 214_H, 234_H is for F_CS1_N and F_CS0_N respectively.*
2. *The wait state time for write accesses after the first read is WAITWR x tHCLK.*

MPMC_SWWR0	Offset	Reset Value
MPMC Static Wait Wr 0	**214_H**	**$1F_H$**



Field	Bits	Type	Description
Res	31:5	-	**Reserved**
WWS	4:0	rw	**Write Wait States** SRAM wait state time for write accesses after the first read. 00000 to 11110_B, (n+2) HCLK cycle write access time 11111_B, 33 HCLK cycle write access time (reset value on nPOR).

Similar Registers

Table 34 MPMC_SWWRx Registers

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_SWWR1	MPMC Static Wait Wr 1	234_H	
MPMC_SWWR2	MPMC Static Wait Wr 2	254_H	
MPMC_SWWR3	MPMC Static Wait Wr 3	274_H	

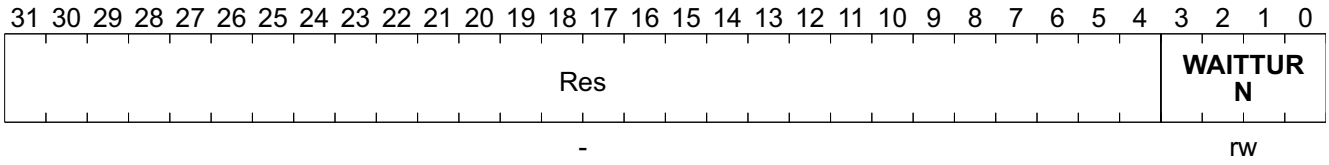
MPMC Static Wait Turn 0

Other Reserved Registers have the same structure and characteristics as **MPMC_SWT0**; the names and offset addresses are listed in **Table 35**.

Notes

1. *Offset = 218_H, 238_H is for F_CS1_N and F_CS0_N respectively.*
2. *Bus turnaround time is (WAITTURN+1) x tHCLK.*

MPMC_SWT0	Offset	Reset Value
MPMC Static Wait Turn 0	**218_H**	**F_H**



Field	Bits	Type	Description
Res	31:4	-	**Reserved**
WAITTURN	3:0	rw	**Bus Turnaround Cycles** 00000 to 1110_B, (n+1) HCLK turnaround cycles 1111_B, 16 HCLK turnaround cycles (reset value on nPOR).

Similar Registers

Table 35 MPMC_SWTx Registers

Register Short Name	Register Long Name	Offset Address	Page Number
MPMC_SWT1	MPMC Static Wait Turn 1	238_H	
MPMC_SWT2	MPMC Static Wait Turn 2	258_H	
MPMC_SWT3	MPMC Static Wait Turn 3	278_H	

6 Ethernet Switch controller

The following chapter describes the Ethernet Switch controller functions of the ADM5120.

6.1 Switch Engine

The switch engine description:

6.1.1 Hashing Function

ADM5120 provides an embedded 1K MAC address look-up table to implement address recognition. The entries of the hashing table are calculated by direct mapping or an XOR function to produce a 10-bit hashing address entry.

6.1.2 Learning Process

The address learning process is composed of the source address (SA) of packets and the hashing function. ADM5120 will compare the SA of each incoming packet:

- If the source address of an incoming packet is the same as the source MAC address table, then the aging status and port number will be updated.
- If the source address is different from the source MAC address table (mean address collision), then no learning process will occur.

Exceptional cases of address learning:

- The packets have errors
- The port learning has been disabled
- Address collision
- The source address is multicast
- The packets are from the CPU

6.1.3 Routing

When a packet comes from port A, ADM5120 will compare its destination MAC address with the MAC address in the MAC address lookup table. If the address is the same and port is port A this means that the packet is a local packet, it is thus discarded. If the address is the same but port numbers are different, the packet is a unicast packet, and will be forwarded to the assigned port. If the incoming packet is a broadcasted one, a multicast one, or an unknown one (i.e. the destination address cannot be found in the MAC address lookup table), then the routing scheme will broadcast it to all ports.

If the MAC address is a VLAN address, then the packet will be routed to the CPU port. The VLAN address is programmed by the CPU, but not from address learning.

6.1.4 Forwarding

ADM5120 provides a store-and-forward method as a forwarding scheme. Each outgoing packet, including “to-CPU” packets, will be stored,first in the buffer, and then directly sent to the assigned port or CPU via the DMA. However, only the good and non-local packets will be sent.

6.1.5 Buffer Management

The buffer memory is embedded in ADM5120 for the switch operations, which are designed based on output queuing and dynamic shared memory management architecture. It will assign buffer resources based on the traffic status. In addition, this efficiency method can avoid the problem of Head-on-Line (HOL) blocking and cause better transmitting performance.

6.1.6 Flow Control

The on/off status for flow control depends on the global empty buffer count and per-port waiting-transmit count. Based on this intelligent scheme, if the packet transmits to a destination port that is full, then the flow control is turned on. In this situation, the full condition is released, including packets transmitted out or disabled. The flow control is then turned off.

ADM5120 does not allow flow control to the CPU, ie. it never sends the flow control packets to the CPU port, so the firmware needs to monitor the buffer status to prevent packet loss.

6.1.7 Full Duplex

In full duplex flow control, ADM5120 follows IEEE 802.3x standards. If a PAUSE frame is received from a certain port, it will stop the port transmission of packets until the timer times out or another PAUSE frame with zero time is received. If the buffer is full and is in full-duplex mode, ADM5120 will send a PAUSE frame with the maximum value, to defer the receiving packet. When enough buffer space is released, the PAUSE frame with zero delay is sent.

- Pause Frame must meet all of the following specs:
  - Right DA: DA=0180c2000001 or unicast MAC address belongs to the CPU
  - Right type field = 8808
  - Right op-code = 0001
  - Right CRC

6.1.8 Half Duplex

In half-duplex operation, ADM5120 supports a back pressure feature. If free blocks in the buffer memory are below the threshold, a jam packet (jam mode) is sent to the connected segment, regardless of routing decisions.

6.1.9 Packet Priority and Class of Service (CoS)

ADM5120 can set the packets as high priority via registers as follows:

- Port based priority, refer to register **Priority Control** bits [5:0]
- VLAN tag, refer to registers **VLAN Priority**
- TCP/IP TOS/DS, refer to registers **TOS Enable**, **TOS Map 0** and **TOS Map 1**
- Customer defined type, refer to registers **Custom Priority 1** and **Custom Priority 2**

The priority setting by port means that all the packets received by the port will be priority frames. ADM5120 can also judge the priority of frames by checking the specific bits of VLAN tag or TCP/IP TOS/DS in the frame or the customer defined type.

It will determine the packet priority. First it will check if the packet type meets VLAN or TCP/IP. Then, it will check whether the value of the VLAN tag or the TCP/IP TOS/DS field meets the registers setting. Depending on these two conditions, the scheme of weighted round robin can determine the high and low priority of frames, and thus set the transmitting order.

ADM5120 provides a function to improve the delay-time sensitive traffic in the flow-control condition. When the port receives a priority frame, the back pressure & 802.3x flow control can be turned off until no priority frame occurs within 1 or 2 seconds, then turned back on again. So it can prevent the jitter caused by the flow control and give a better timing-variation result for the priority traffic. This is a register programmable function.

All the packets from the CPU port will be treated as high priority for the switch ports and the best effort result for the CPU traffic will be provided.

6.1.10 VLAN

ADM5120 supports a seven port-grouping VLAN. Each of thr VLAN will be treated as isolated ports.For the VLAN grouping setting, refer to the registers **VLAN Group I** and **VLAN Group II**.

ADM5120 provides the VLAN MAC address function, if the packet is assigned with the VLAN address as its destination MAC address, then this packet will be forwarded to the CPU via DMA.

For example, port0 is the WAN port. The others are the LAN ports, then the WAN ports are set as VLAN1 and the others set as VLAN2. The different MAC addresses for the VLAN1/2 are programmed into the address table. Then if the LAN ports receive packets with VLAN2 addresses, the packets will be forward to the CPU via DMA. After processing the packets (like NAT), the CPU can forward the packets to VLAN1.

6.1.11 Address Table Access

ADM5120 provides the access for the embedded MAC address.

- Read - refer to the registers **Search CMD**, **Address ST0** and **Address St1**
  Issue the search-start command. ADM5120 will automatically search the embedded address table and report the valid one only. If at the end of a table, it will also report the status.
- Write - refer to registers **MAC Write Address 0** and **MAC Write Address 1**
  Fill in the write address and other information, like port number (or VLAN number), age-time (or static), then issue the write command and wait for the write done bit.

6.1.12 Address Security

The ADM5120 supports the source MAC address security function, register **Port Conf1**(B+2C) bits [31:26]. It can check all-incoming packets in the enable ports – find if the source MAC exists in the MAC address table or not, if not, discard the packets and report the status, register **Port St**(B+18) bits [5:0].

6.1.13 Bandwidth Control Function

ADM5120 can provide the RX/TX separated bandwidth control (or traffic shaping) function, which can be programmed to 64 kbit / 128 kbit / 256 kbit / 512 kbit / 1 Mbit / 4 Mbit / 10 Mbit. Refer the registers **Bandwidth Control 0** and **Bandwidth Control 1**.

In a fixed period, ADM5120 will count the per port RX and TX byte number, and compare with the bandwidth control threshold. If it is over this threshold, ADM5120 will turn on the proprietary scheme to control the RX/TX behavior.

6.1.14 MII/GMII Port

The GMII/MII port can be programmed for the following: AN monitor on/off, force speed/duplex/flow-control, which can be set by Switch Control Register **Port Conf2**(B+30).

The GMII/MII direction is also programmable for the following: connect to PHY or MAC, which can be set by Switch Control Register **Port Conf2**(B+30).

The default GMII/MII mode is normal mode that ‘connect to PHY’. When it is configured to Reverse MII mode. The ADM5120 will output TXC, RXC, CRS and COL. The signals direction will change, and suggest the connection as below:

Table 36 Connection between ADM5120 and MAC Controller

5120 Signal	MAC Signal
RXC	RXC
TXE	RXDV
TXD	RXD
RXDV	TXEN
RXD	TXD
COL	COL
CRS	CRS

6.2 DMA Function Description

The DMA function provides the packets transmit and receive to/from CPU. There are two piority queues in each path -- transmit and receive. The start address is defined by the base address registers. You can refer to registers in **Send High Base Address**, **Send Low Base Address**, **Receive High Base Address** and **Receive Low Base Address** for details. Every queue is a ring architecture. See the tables for details.

When the packet is put on the data buffer and send descriptor is prepared, software can trigger the DMA to move the data to the internal buffer by setting the **Send Trigger** register.

6.2.1 Send Descriptors Content

If CPU sends the packet to switch, either LAN or WAN, then the ‘send descriptors’ are used as follows:

Bit	Bit[31]	Bit[28]	Bit[24:0]	Remark
Type		Control		Controlled by CPU except Own-bit
Function	Own bit	Ring end flag	buffer1_address[24:0]	

Bit	Bit[31]	Bit[24:0]	Remark
Type	Control		Controlled by CPU
Function	buffer2_enable	buffer2_address[24:0]	

Bit		Bit[10:0]	Remark
Type		Control	Controlled by CPU
Function		buffer1_length[10:0]	

Bit	Bit[31]	Bit[26:16]	Bit[13:8]	Bit[5:0]	Remark
Type	Control				Controlled by CPU
Function	append_chksum	pkt-length[10:0]	force desti-port[5:0]	To_VLAN[5:0]	

6.2.1.1 Control

- Own Bit:
  - If 0, the descriptor belongs to CPU. After the data is put in the buffer and control bits are set, this bit will change to 1 to indicate SW can process this packet.
  - If 1, the descriptor is for SW, and after the data is taken away, then it is loaded to SW data buffer, the bit will be then set to 0.
- Ring end:
  - If 1, the descriptor is the last one, the next descriptor needs to return to the base address.
- Buffer information:
  - Each descriptor can support two buffers.
  - The buffer address can be any byte alignment.
  - Buffer1 has length information, if packet size is larger than buffer1 size, then get the rest of the data from buffer2.
  - Buffer address must be valid when a descriptor belongs to a switch, the switch engine will not check the address status.
  - Buffer2 has an enable bit to control whether the address is valid or not.
  - If the buffer2 is disabled and buffer1 is not long enough, then the remaining data will not be padded 0 to make the packet meet 64-bytes standard.
- Append_chksum: need to append the IP (0800_H) and PPPoE (8864_H) packets IP-checksum by hardware
  - The packet checksum field must be pre-filled with all 0’s
- Packet length: the packet length in bytes, excluding CRC if CRC is not padded. (See the register, CPUp_conf)
  - Auto-padding: the engine can automatically pad the 0 into the packet which data size is less than 60 B (or 64). The setting example: buffer 1 size=14, buffer 2 disable, the pkt length=60 (or 64 without CRC padding).
- Force desti-port[5:0]: the packet needs force forwarding to designated ports, and it is the highest priority of routing. If forced, then ignore the routing and To_VLAN flag.

- To_VLAN[5:0]: the bit-map, the packet forwards to the designated VLAN group. Use this flag to control the packets to LAN, WAN, or HPNA ports.

6.2.2 Receive Descriptors Content

If switch sends the packet to CPU, either LAN or WAN, then the 'receive descriptors' are used as follows:

Bit	Bit[31]	Bit[28]	Bit[24:0]	Remark
Type		Control		Controlled by CPU except Own-bit
Function	Own bit	Ring end flag	buffer1_address[24:0]	

Bit	Bit[31]	Bit[24:0]	Remark
Type	Control		Controlled by CPU
Function	buffer2_enable	buffer2_address[24:0]	

Bit		Bit[10:0]	Remark
Type		Control	Controlled by CPU
Function		buffer1_length[10:0]	

Bit	[26:16]	[14:12]		[5:4]	[3]	[2]	[1:0]	Remark
Type	Packet status							Updated by switch
Function	pkt-length[10:0]	Source port number		00: UC01: MC10: BC	IP checksum fail	VLAN tag	00: 0800_H01: 8864_H11, 10: reserved	

6.2.2.1 Control

- Own bit:
  - If 0, the descriptor belongs to CPU.
  - If 1, the descriptor is released to WAN MAC or LAN SW, which means it can store the incoming packet based on the buffer address. If this is done, change the bit to 0.
- Buffer information:
  - Each descriptor can support two buffers.
  - The buffer address can be any byte alignment.
  - Buffer1 has length information, if packet size is over the buffer1 size, then put the rest of the data into buffer2.
  - Buffer1 address must be valid when descriptor belongs to switch.
  - Buffer2 has a enable bit to control whether the address is valid or not.
  - The buffer2 size must be larger than the remaining data.
  - If buffer2 is disabled and buffer1 has not enough space, then the remaining data will be dropped, and no status reported.

6.2.2.2 Status

- Packet length: the packet length in bytes including 4-byte CRC
- Source port: the source port of packet
- DA status:
  - 00: UC, the packet is the forwarded UC packet
  - 01: MC, the packet has 1 in the LSB of first byte of DA
  - 11: BC, the packet has DA=FFFFFFFFFFFF
  - IP checksum fail: if 1 = the IP checksum result is error

Note: Only checked if type = 0800_H (IP) or 8864_H (PPPoE)

- The VLAN tagged frame status (type = 8100_H)
- Packet type:
  - 00: type = 0800_H, IP
  - 01: type = 8864_H, PPPoE
  - 10,11 reserved

6.3 Switch Control Register Map

Table 37 Address Space

Module	Base Address	End Address	Note
Switch Control	$1200\ 0000_H$	$1200\ 0110_H$	Xxxxx

Table 38 Registers Overview from Switch Control Register

Register Short Name	Register Long Name	Offset Address
Code	Code	00_H
Sft_Res	Software Reset	04_H
Boot_D	Boot Done	08_H
SW_Res	Software Reset	$0C_H$
port0_LED	Port 0 LED	100_H
port1_LED	Port 1 LED	104_H
port2_LED	Port 2 LED	108_H
port3_LED	Port 3 LED	$10C_H$
Gl_St	Global St	10_H
port4_LED	Port 4 LED	110_H
PHY_St	PHY St	14_H
Port_St	Port St	18_H
Mem_Cont	Memory Control	$1C_H$
SW_conf	SW Conf	20_H
CPUp_conf	CPUp Conf	24_H
Port_conf0	Port Conf0	28_H
Port_conf1	Port Conf1	$2C_H$
Port_conf2	Port Conf2	30_H
Res_1	Reserved 1	34_H
Res_2	Reserved 2	38_H

Table 38 Registers Overview from Switch Control Register (cont'd)

Register Short Name	Register Long Name	Offset Address
Res_3	Reserved 3	$3C_H$
VLAN_GI	VLAN Group I	40_H
VLAN_GII	VLAN Group II	44_H
Send_trig	Send Trigger	48_H
Srch_cmd	Search CMD	$4C_H$
ADDR_st0	Address ST0	50_H
ADDR_st1	Address St1	54_H
MAC_wt0	MAC Write Address 0	58_H
MAC_wt1	MAC Write Address 1	$5C_H$
BW_cntl0	Bandwidth Control 0	60_H
BW_cntl1	Bandwidth Control 1	64_H
PHY_cntl0	PHY Control 0	68_H
PHY_cntl1	PHY Control 1	$6C_H$
FC_th	Switch Control Threshold	70_H
adj_port_th	Adj Port Threshold	74_H
Port_th	Port Threshold	78_H
PHY_cntl2	PHY Control 2	$7C_H$
PHY_cntl3	PHY Control 3	80_H
Pri_cntl	Priority Control	84_H
VLAN_pri	VLAN Priority	88_H
TOS_en	TOS Enable	$8C_H$
TOS_map0	TOS Map 0	90_H
TOS_map1	TOS Map 1	94_H
Custom_pri1	Custom Priority 1	98_H
Custom_pri2	Custom Priority 2	$9C_H$
PHY_cntl4	PHY Control 4	$A0_H$
Empty_cnt	Empty Control	$A4_H$
Port_cnt_sel	Port Control Select	$A8_H$
Port_cnt	Port Controller	AC_H
Int_st	Int St	$B0_H$
Int_mask	Interrupt Mask	$B4_H$
GPIO_conf0	GPIO Conf 0	$B8_H$
GPIO_conf2	GPIO Conf 2	BC_H
Wdog_0	Watchdog 0	$C0_H$
Wdog_1	Watchdog 1	$C4_H$
Swap_in	Swap In	$C8_H$
Swap_out	Swap Out	CC_H
send_Hbaddr	Send High Base Address	$D0_H$
send_Lbaddr	Send Low Base Address	$D4_H$
rec_Hbaddr	Receive High Base Address	$D8_H$
rec_Lbaddr	Receive Low Base Address	DC_H

Table 38 Registers Overview from Switch Control Register (cont'd)

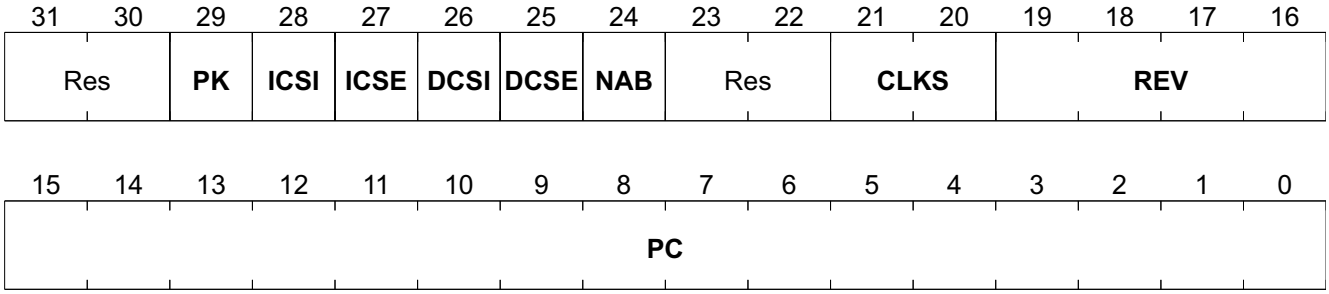
Register Short Name	Register Long Name	Offset Address
send_Hwaddr	Send High Working Address	$E0_H$
send_Lwaddr	Send Low Working Address	$E4_H$
rec_Hwaddr	Receive High Working Address	$E8_H$
rec_Lwaddr	Receive Low Working Address	EC_H
Timer_int	Timer Interrupt	$F0_H$
Timer	Timer	$F4_H$
Res_4	Reserved 4	$F8_H$
Res_5	Reserved 5	FC_H

Note: Although some registers may be marked with a certain type, it may be possible that some bits in this register are different. This is explained in the register description.

6.4 Registers Description

Code

Code	Offset	Reset Value
Code	**00_H**	**12085120_H**



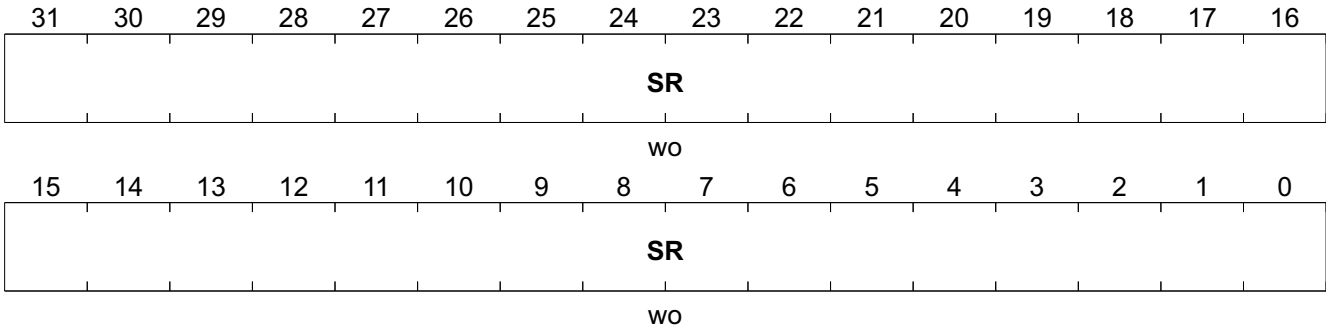
ro

Field	Bits	Type	Description
Res	31:30		**Reserved** Not Applicable.
PK	29	ro	**Package type** 0_B , BGA 1_B , 208 PQFP/disable GMII
ICSI	28	ro	**Icache Size** 0_B , 1 Way 1_B , 2 Ways
ICSE	27	ro	**Icache Set** 0_B , 4K per way 1_B , 2K per way

Field	Bits	Type	Description
DCSI	26	ro	**Dcache Size** 0_B , 1 Way 1_B , 2 Ways
DCSE	25	ro	**Dcache Set** 0_B , 4K per way 1_B , 2K per way
NAB	24	ro	**NAND Boot** Configured in the NAND flash boot.
Res	23:22		**Reserved**
CLKS	21:20	ro	**Clock SPD** The PLL setting. 00_B , 175 MHz (Default) 01_B , 200 MHz $1x_B$, Reserved
REV	19:16	ro	**Revision** Revision code = 1000
PC	15:0	ro	**Product Code** Product code = 5120_H

Software Reset

Sft_Res	Offset	Reset Value
Software Reset	**04_H**	**1_H**

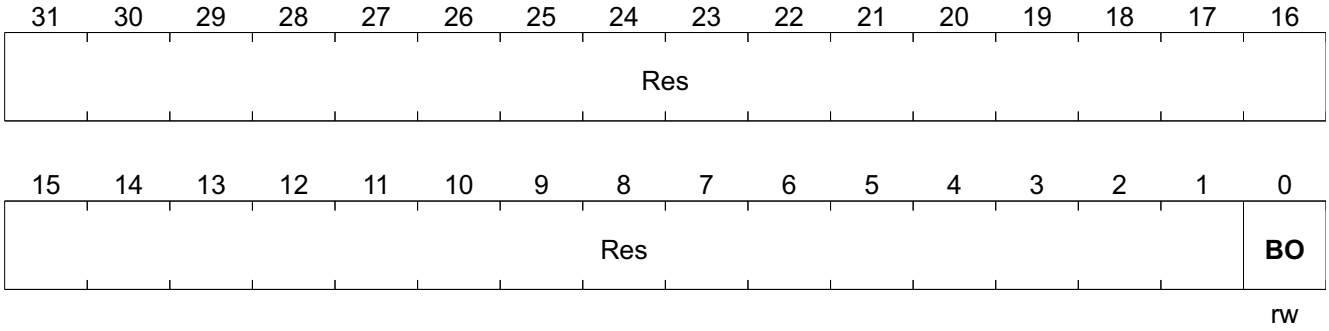


Field	Bits	Type	Description
SR	31:0	wo	**Software Reset** Do Software reset when write, reset all logic, PHY and memory, and down load the NAND flash content again. Same as hardware reset.

Note: Whenever you write the register offset 04_H, the SftReset will be active.

Boot Done

Boot_D	Offset	Reset Value
Boot Done	08_H	0_H

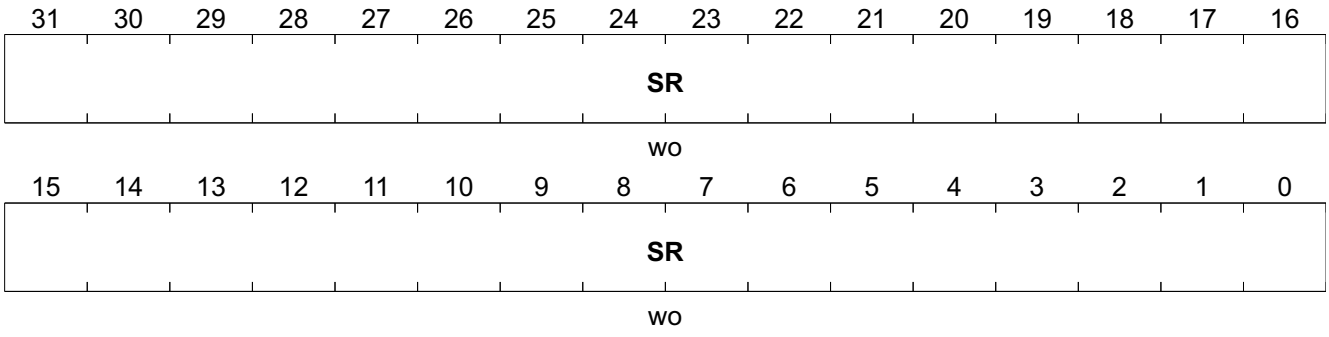


Field	Bits	Type	Description
Res	31:1		**Reserved**
BO	0	rw	**Boot** 1_B , The software boot process is done and the address table can return to switch controller.

Switch Reset

Note: Whenever you write the register offset $0C_H$ the SWReset will be active.

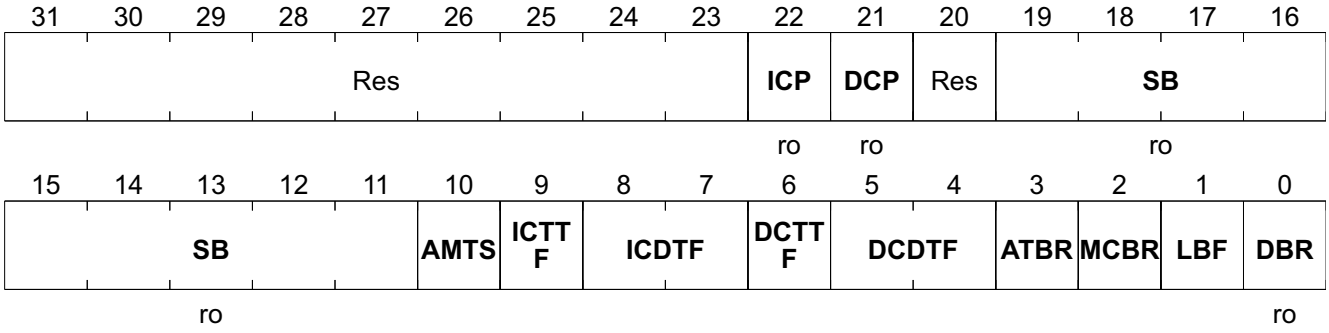
SW_Res	Offset	Reset Value
Software Reset	$0C_H$	1_H



Field	Bits	Type	Description
SWR	31:0	wo	**Switch Reset** Do Switch reset when write, including Switch engine, data-buffer, link table, and PHY excluding address table. (Recommend stop PHY before reset switch).

Global St

GI_St	Offset	Reset Value
Global St	**10_H**	**400_H**

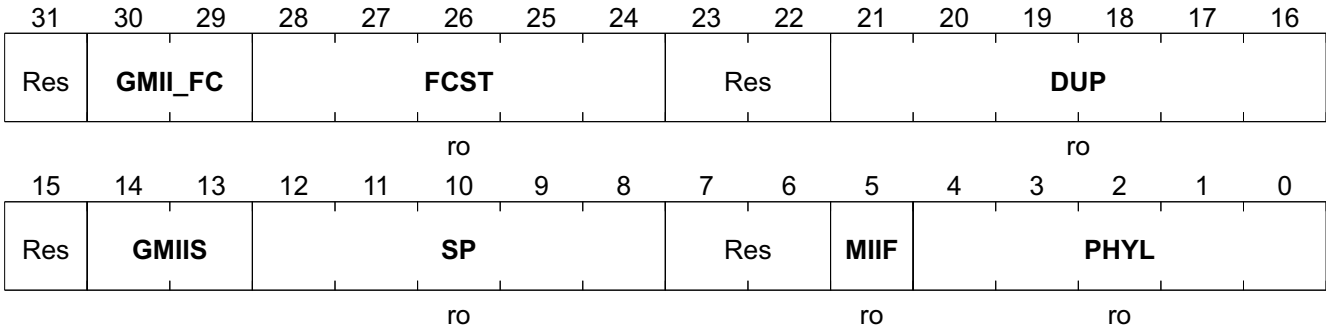


Field	Bits	Type	Description
Res	31:23		**Reserved** Not Applicable.
ICP	22	ro	**Icache Portion** For debugging purpose of embedded SRAM.
DCP	21		**Dcache Portion** For debugging purpose of embedded SRAM.
Res	20		**Reserved**
SB	19:11	ro	**Skip Blocks** The number of block is skipped up to 64 blocks.
AMTS	10	ro	**All Embedded Memory Test Completed** 1_B , Complete
ICTTF	9	ro	**Icache Tag Test Fail** The memory of I-cache tag. 0_B , Pass
ICDTF	8:7	ro	**Icache Data Test Fail** Bit 8 and Bit 7 are respectively for the upper and lower 32-bit of I-cache memory for data. 00_B , Pass
DCTTF	6	ro	**Dcache Tag Test Fail** The memory of D-cache tag. 0_B , Pass
DCDTF	5:4	ro	**Dcache Data Test Fail** Bit 5 and Bit 4 are respectively for the upper and lower 32-bit of D-cache memory for data. 00_B , Pass
ATBR	3	ro	**Address Table BIST Result** 0_B , Pass
MCBR	2	ro	**MC Table BIST Result** 0_B , Pass

Field	Bits	Type	Description
LBF	1	ro	**Link Table BIST Result** 0_B , Pass
DBR	0	ro	**Data Buffer BIST Result** 0_B , Pass

PHY St

PHY_St	Offset	Reset Value
PHY St	14_H	0_H

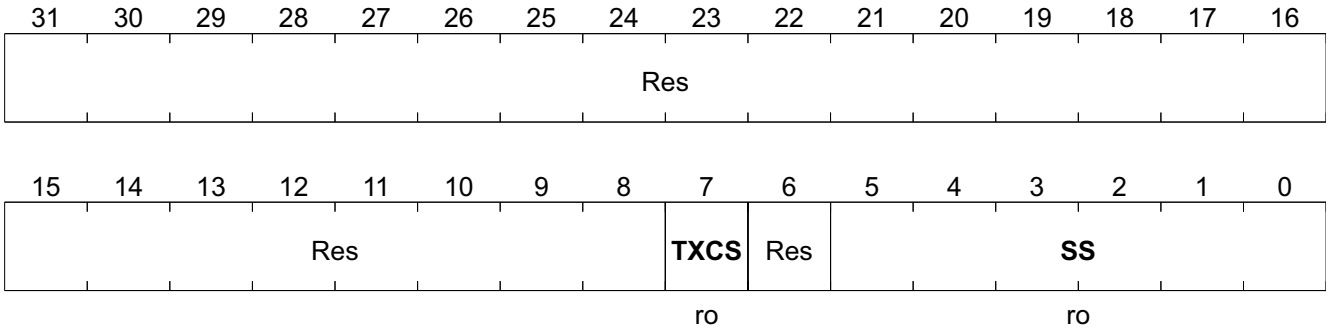


Field	Bits	Type	Description
Res	31		**Reserved** Not Applicable.
GMII_FC	30:29	ro	**FC Status of GMII Port** Bit 29 represents the RX path Flow Control status. bit 30 represnts the TX path Flow Control status if in Giga mode. If 10/100 mode, only bit 29 is used for flow control status. 0_B , FC off 1_B , FC on
FCST	28:24	ro	**FC** Bit [28:24] represent PHY port [4:0] flow control status respectively. 1_B , Full duplex and 802.3x flow control ON (after AN or forced).
Res	23:22		**Reserved** Not Applicable.
DUP	21:16	ro	**Duplex** 0_B , Half duplex 1_B , Full duplex
Res	15		**Reserved** Not Applicable.
GMIIS	14:13	ro	**GMII Port Speed** 00_B , 10M 01_B , 100M 10_B , Giga (disable on PQFP)
SP	12:8	ro	**Speed** 0_B , 10M 1_B , 100M

Field	Bits	Type	Description
Res	7:6		**Reserved** Not Applicable.
MIIF	5	ro	**MII Fail to MII** 0_B , Port fail 1_B , Link ok
PHYL	4:0	ro	**PHY Link** Bit[4:0] represent PHY port[4:0] link status respectively 0_B , Down 1_B , Up

Port St

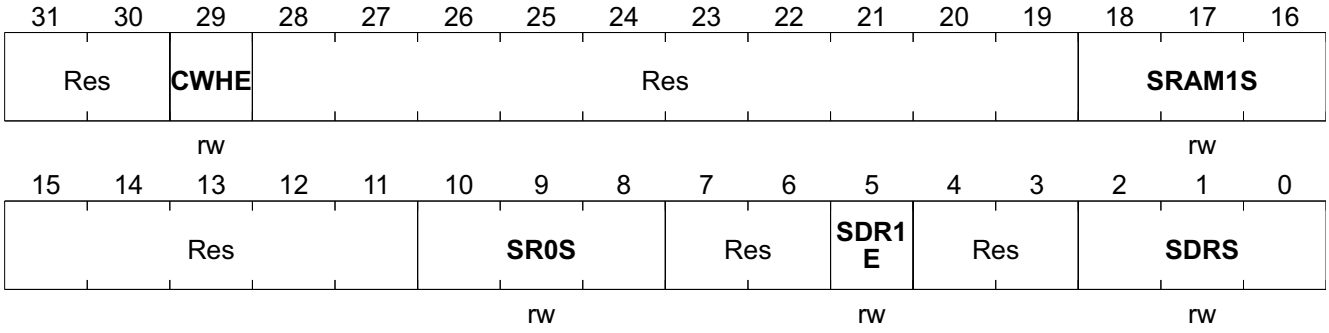
Port_St	Offset	Reset Value
Port St	18_H	0_H



Field	Bits	Type	Description
Res	31:8		**Reserved** Not Applicable.
TXCS	7	ro	**TXC St, MII Port TXC Status** 1_B , Error, no TXC or too long period
Res	6		**Reserved** Not Applicable.
SS	5:0	ro	**Security Status** 1_B , Has intruder coming if turn on the SA_securd mode

Memory Control

Mem_Cont	Offset	Reset Value
Memory Control	$1C_H$	204_H

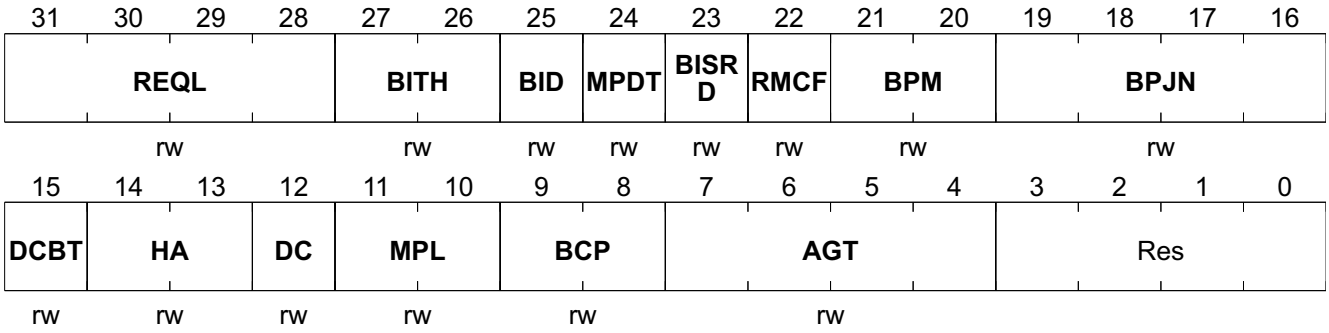


Field	Bits	Type	Description
Res	31:30		**Reserved** Not Applicable.
CWHE	29	rw	**CSX0 Write Hold Extend** Extend the write data hold time from one clock to 3 clock (clock period = CLK_OUT). 0_B , Normal, 1 clock, default 1_B , Extend to 3 clocks
Res	28:19		**Reserved** Not Applicable.
SRAM1S	18:16	rw	**SRAM1 Size** 000_B , Disable (default) 001_B , 512 Kbyte 010_B , 1 Mbyte 011_B , 2 Mbyte 100_B , 4 Mbyte 101_B , 8 Mbyte 110_B , Reserved 111_B , Reserved
Res	15:11		**Reserved** Not Applicable.
SR0S	10:8	rw	**SRAM0 Size** 000_B , Disable if in the NAND mode 001_B , 512 Kbyte 010_B , 1 Mbyte (default 011_B , 2 Mbyte 100_B , 4 Mbyte 110_B , Reserved 111_B , Reserved
Res	7:6		**Reserved** Not Applicable.

Field	Bits	Type	Description
SDR1E	5	rw	**SDRAM1 Enable** The bank1 of SDRAM enable. 0_B , Disable (default) 1_B , Enable (must in the single write)
Res	4:3		**Reserved** Not Applicable.
SDRS	2:0	rw	**SDRAM Size** One bank information, the 2 nd bank (SDRAM_CS1) is the same. 000_B , Reserved 001_B , 1M x 32 (4 Mbyte) 010_B , 2M x 32 (8 Mbyte) (suggested setting) 011_B , 4M x 32 (16 Mbyte) 100_B , 16M x 32 (64 Mbyte) 101_B , 32M x 32 (128 Mbyte) 110_B , Reserved 111_B , Reserved

SW_conf

SW_conf	Offset	Reset Value
SW Conf	20_H	$402A1010_H$

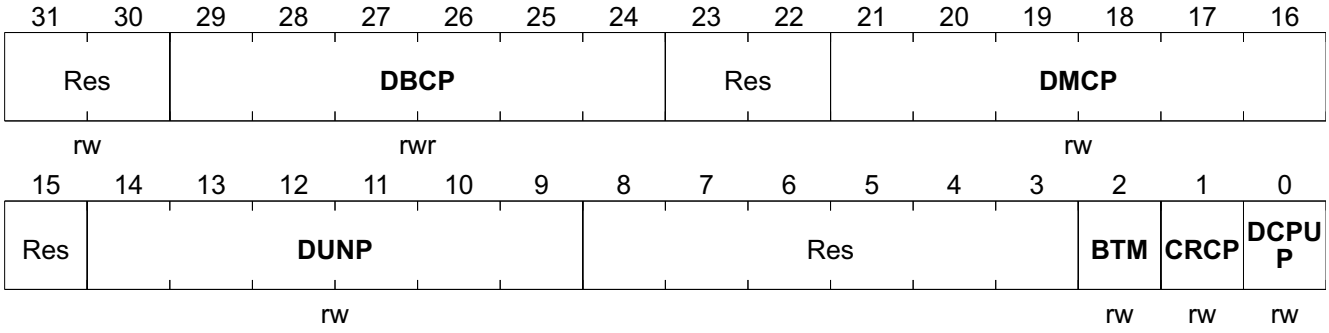


Field	Bits	Type	Description
REQL	31:28	rw	**AHB Request Latency** The minimun number of cycles between the AHB bus request. 4_B , 4 clocks latency between requests
BITH	27:26	rw	**The Threshold of BISS** 00_B , Skip if fail 16 (default from pins) 01_B , Skip if fail 48 10_B , Skip if fail 64 11_B , Skip if fail 8 blocks
BID	25	rw	**Build-in Self Skip Disable** 0_B , Enable skip function (default, from pin A[6])
MPDT	24	rw	**MII0 Port Disable was Transmit** 0_B , Enable 1_B , Disable was_transmit (good for late CRS PHY, like HPNA2.0 or power-LAN)

Field	Bits	Type	Description
BISRD	23	rw	**Build-in Self Repair Disable** 0_B , Enable skip function (default, from pin A[5])
RMCF	22	rw	**Reserved MC Filtering** *Note: Reserved MC = 01-80-c2-00-00-00 (BPDU) and 01-80-c2-00-00-02 to 01-80-c2-00-00-0f* 0_B , Forwarded 1_B , Filtered
BPM	21:20	rw	**Back Pressure Mode** 00_B , Disable 01_B , BP jam, the jam number is set by BP_num 10_B , BP jamALL, jam packet until the BP condition is released (default), 11_B , BP carrier, use carrier insertion to do back pressure
BPJN	19:16	rw	**Back Pressure Jam Number** The consecutive jam time when back pressure. 1010_B, Default value, 10 packet jam then one no-jam
DCBT	15	rw	**Disable the Collision Back off Timer** 0_B , Follow standard 1_B , Re-transmit immediately after collision
HA	14:13	rw	**MAC Address Hashing Algorithm** 00_B , Direct mode, using last 10-bit as hashing address 01_B , XOR48 mode 10_B , XOR32 mode 11_B , Reserved
DC	12	rw	**Disable Collision** 1_B , Disable collision 16 packet abort
MPL	11:10	rw	**Maximum Packet Length** 00_B , 1536 01_B , 1518 10_B , 1522 11_B , Reserved
BCP	9:8	rw	**Broadcast Prevention** 00_B , Disable, BC will be blocked 01_B , 64 blocks 10_B , 48 blocks 11_B , 32 blocks
AGT	7:4	rw	**Aging Timer** 0000_B , Disable age 0001_B , 300 s 0010_B , 600 …… 0111_B , 38400 s $1xxx_B$, Fast age
Res	3:0		**Reserved** Not Applicable.

CPUp_conf

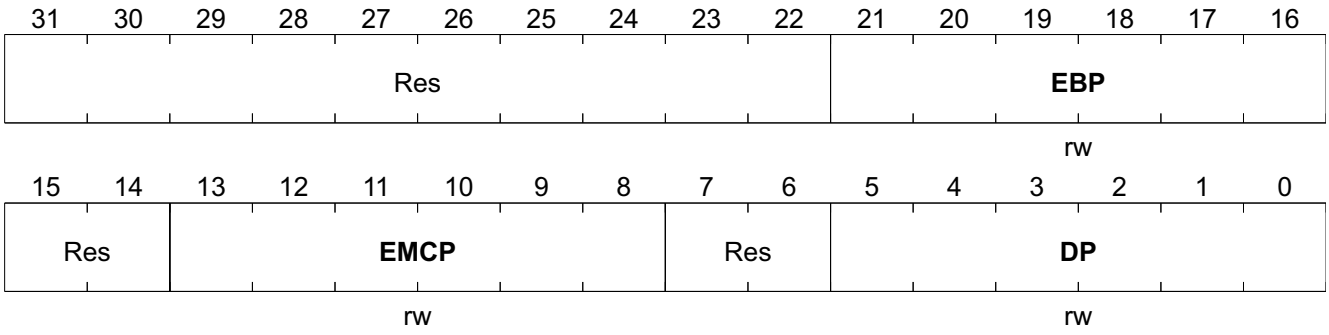
CPUp_conf	Offset	Reset Value
CPUp Conf	24_H	$3F3F7E01_H$



Field	Bits	Type	Description
Res	31:30	rw	**Reserved** Not Applicable
DBCP	29:24	rwr	**Disable Broadcast Packets, from port(s), Forward to CPU** The bit [29:24] control the MII port and PHY port [4:0] respectively. 1_B , Disable sending BC from the MAC port to CPU
Res	23:22		**Reserved** Not Applicable
DMCP	21:16	rw	**Disable Multicast Packets, from Port(s), Forward to CPU** The bit [21:16] control the MII port and PHY port [4:0] respectively. 1_B , No send MC from the MAC port to CPU
Res	15		**Reserved** Not Applicable
DUNP	14:9	rw	**Disable Unknown Packets, from Port(s), Forward to CPU** The bit [14:9] control the MII port and PHY port [4:0] respectively. 1_B , No send unknown packet from the MAC port to CPU
Res	8:3		**Reserved** Not Applicable
BTM	2	rw	**Bridge Testing Mode** 0_B , Default 1_B , Forward to CPU if the DA is the port, belonged to the other VLAN (for the bridge mode testing).
CRCP	1	rw	**CRC Padding from CPU** 1_B , The packet from CPU with CRC
DCPUP	0	rw	**Disable CPU Port** 1_B , Disable the switch CPU port, and so send packets to CPU and clear all the packets in the switch buffer.

Port_conf0

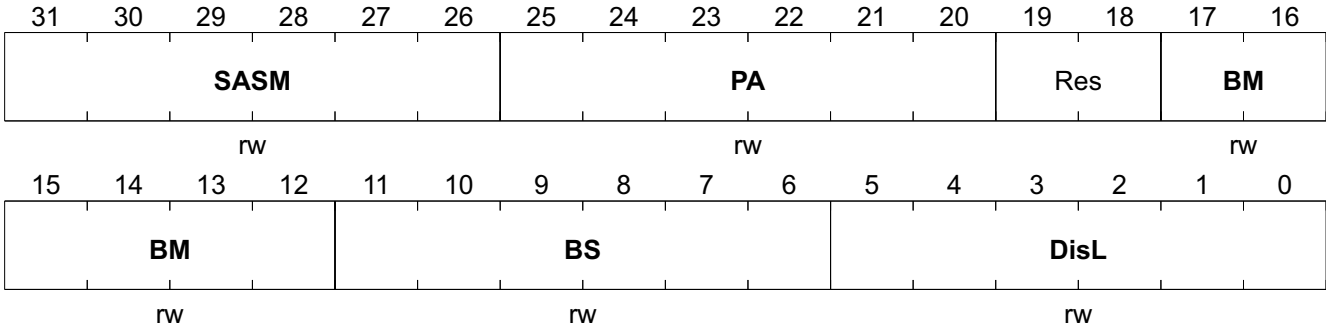
Port_conf0	Offset	Reset Value
Port Conf0	28_H	$3F3F3F_H$



Field	Bits	Type	Description
Res	31:22		**Reserved** Not Applicable.
EBP	21:16	rw	**Enable Back Pressure** 1_B , Enable back pressure (but need qualify BP_mode)
Res	15:14		**Reserved** Not Applicable.
EMCP	13:8	rw	**Enable All MC Packet** Enable all MC packet broadcast to port in the same VLAN (not including CPU). 1_B , Enable Layer2 MC broadcast to ports, 0: do not broadcast MC
Res	7:6		**Reserved** Not Applicable
DP	5:0	rw	**Disable Port** 1_B , Port disable (if dumb mode, default = 0)

Port_conf1

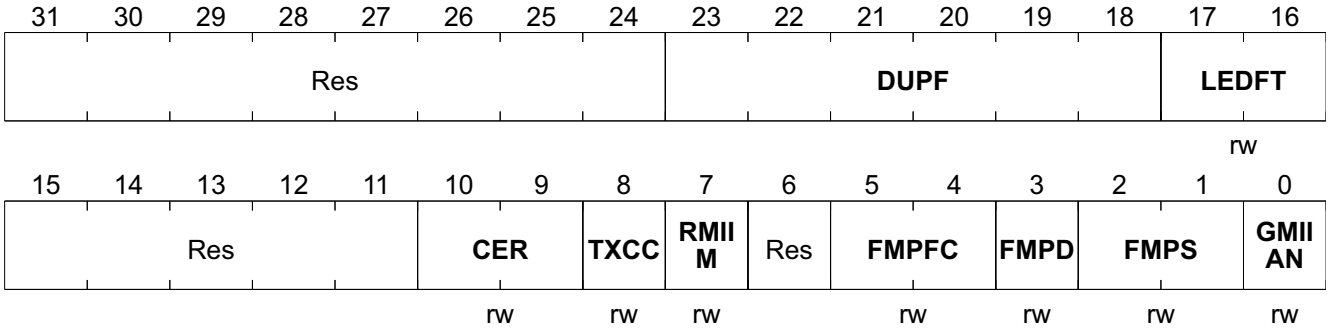
Port_conf1	Offset	Reset Value
Port Conf1	$2C_H$	$3F00000_H$



Field	Bits	Type	Description
SASM	31:26	rw	**SA Secured Mode** **Notes:** *1. Set Dis_Learn and SA_secured at the same time, then only forward the packets with the SA and port number matched.* *2. 2. set SA_secured only, then no any secured function* 0_B , Don't care SA match 1_B , The packets' SA need match, otherwise discard the packets
PA	25:20	rw	**Port Aging** 1_B , Enable aging 0_B , Disable aging that the MAC address is belong to Programmed port(s)
Res	19:18		**Reserved** Not Applicable.
BM	17:12	rw	**Blocking Mode** If in blocking state. 0_B , Forward all packets to CPU 1_B , Only forward control packets to CPU
BS	11:6	rw	**Blocking State** Only do the forwarding to CPU, and no learning, and no transmitting (except the packet from CPU). 0_B , Normal state 1_B , Block state
DisL	5:0	rw	**Dis Learn** Stop SA learning. 0_B , Enable SA learn

Port_conf2

Port_conf2	Offset	Reset Value
Port Conf2	30_H	$10C_H$

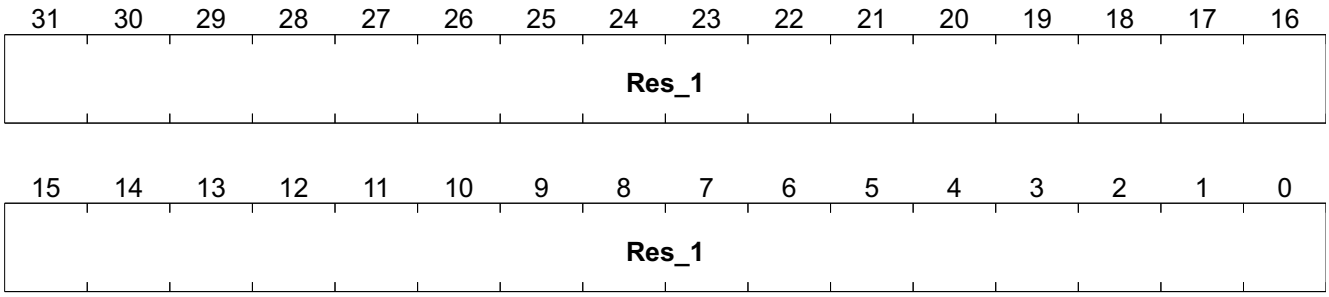


Field	Bits	Type	Description
Res	31:24		**Reserved** Not Applicable.
DUPF	23:18		**Disable Unicast Pause Frame 5:0** 00000_B , Default value
LEDFT	17:16	rw	**The Frequency of LED Flash** 00_B , 30 ms 01_B , 60 ms 10_B , 240 ms 11_B , 480 ms
Res	15:11		**Reserved** Not Applicable.
CER	10:9	rw	**Configuration Early Ready** Giga Port early preamble configuration. 00_B , Default value
TXCC	8	rw	**TXC Check** Check the MII port TXC period, if more than 400 µs, then disable MII port 0_B , Enable check 1_B , Disable check TXC (only for 10/100M) (default)
RMIIM	7	rw	**Reversed MII Mode** 0_B , Normal MII mode (default) 1_B , Enable (if in the dumb mode set to 1)
Res	6		**Reserved** Not Applicable.
FMPFC	5:4	rw	**Force MIIport FC** Force MII port 802.3x flow control ON if AN monitor disable. 00_B , No forced 01_B , Forced the FC of 10M/100M, forced the RX FC of giga 10_B , Forced the TX FC of giga

Field	Bits	Type	Description
FMPD	3	rw	**Force MII Port Duplex if AN Monitor Disable** *Note: Duplex is forced in full, if speed is forced in the giga* 0_B , Forced in half duplex 1_B , Forced in full duplex
FMPS	2:1	rw	**Force MII Port Speed if AN Monitor Disable** 00_B , Force d in 10M 01_B , Forced in 100M 10_B , Forced in giga 11_B , Reserved
GMIIAN	0	rw	**MII Port AN Monitor Enable** 0_B , Disable 1_B , Enable MII AN monitor via MDIO (if in the dumb mode, set to 1)

Reserved 1

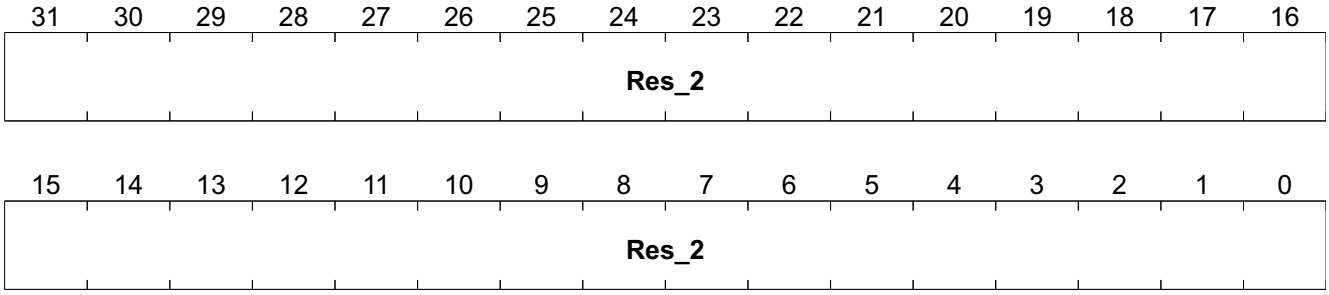
Res_1	Offset	Reset Value
Reserved 1	**34_H**	**0_H**



Field	Bits	Type	Description
Res_1	31:0		**Reserved** Not Applicable.

Reserved 2

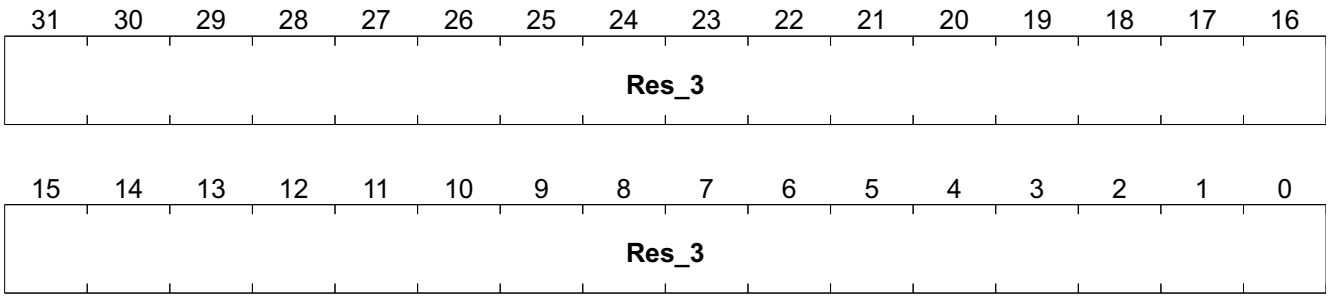
Res_2	Offset	Reset Value
Reserved 2	38_H	0_H



Field	Bits	Type	Description
Res_2	31:0		**Reserved** Not Applicable.

Reserved 3

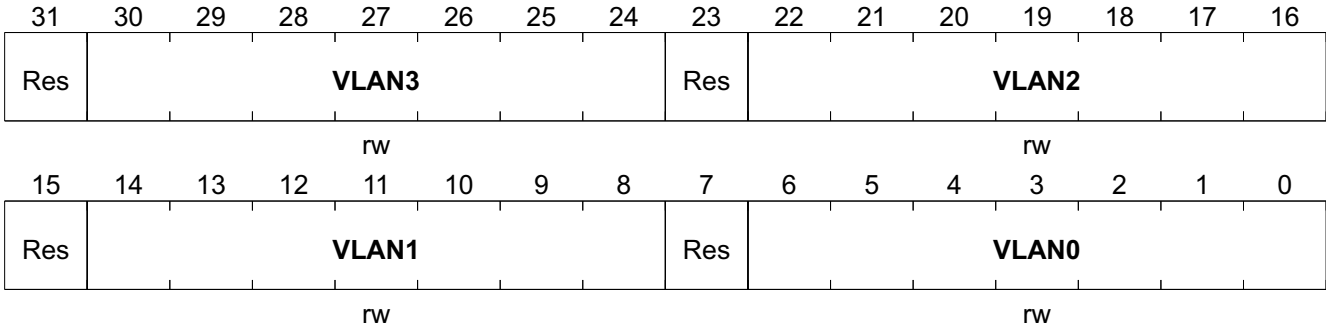
Res_3	Offset	Reset Value
Reserved 3	$3C_H$	0_H



Field	Bits	Type	Description
Res_3	31:0		**Reserved** Not Applicable.

VLAN Group I

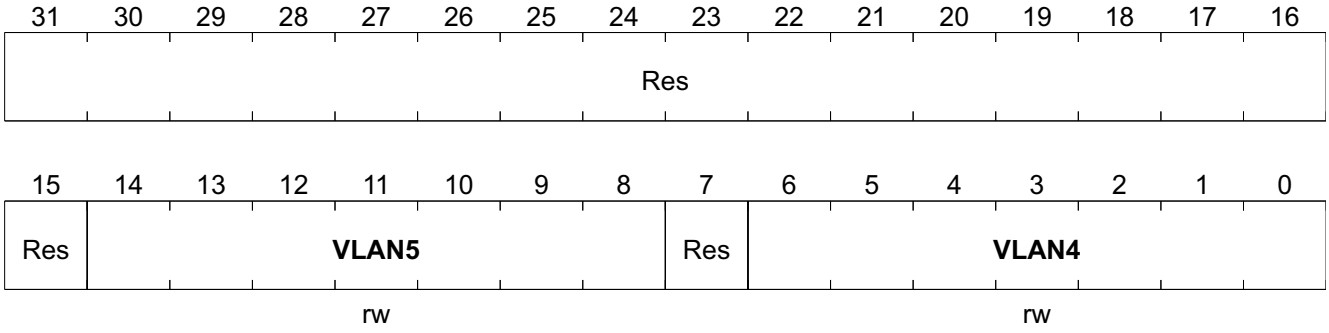
VLAN_GI	Offset	Reset Value
VLAN Group I	40_H	$7F_H$



Field	Bits	Type	Description
Res	31		**Reserved** Not Applicable.
VLAN3	30:24	rw	**VLAN Group 3** The ports in VLAN group 3, 0000000
Res	23		**Reserved** Not Applicable.
VLAN2	22:16	rw	**VLAN Group 2** The ports in VLAN group 2, 0000000
Res	15		**Reserved** Not Applicable.
VLAN1	14:8	rw	**VLAN Group 1** The ports in VLAN group 1, 0000000
Res	7		**Reserved** Not Applicable.
VLAN0	6:0	rw	**VLAN Group 0** The ports in VLAN group 0, the 6 th bit is CPU port. 1111111_B , All ports in the VLAN group 0

VLAN Group II

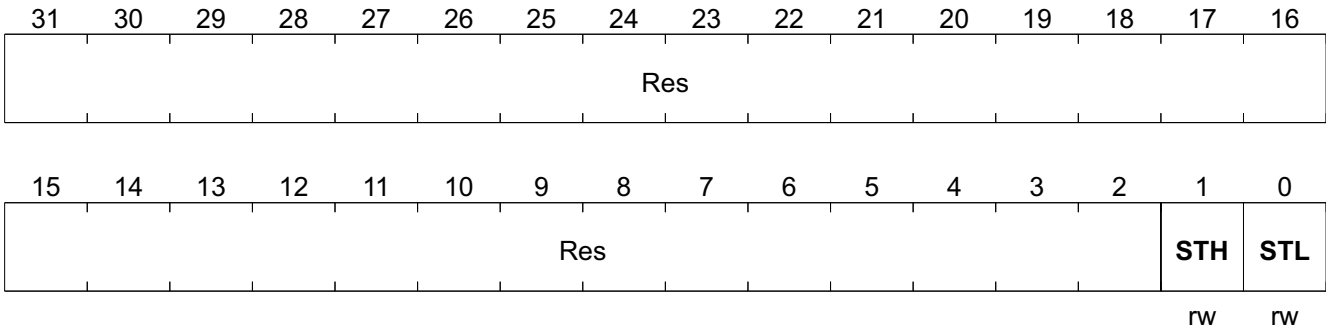
VLAN_GII	Offset	Reset Value
VLAN Group II	44_H	0_H



Field	Bits	Type	Description
Res	31:15		**Reserved** Not Applicable.
VLAN5	14:8	rw	**VLAN Group 5** The ports in VLAN group 5, 0000000
Res	7		**Reserved** Not Applicable.
VLAN4	6:0	rw	**VLAN Group 4** The ports in VLAN group 4, 0000000

Send Trigger

Send_trig	Offset	Reset Value
Send Trigger	48_H	0_H

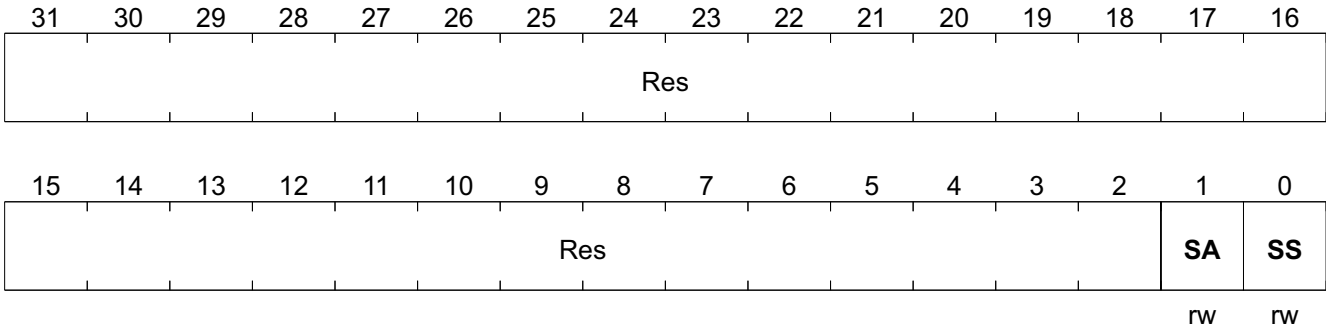


Field	Bits	Type	Description
Res	31:2		**Reserved** Not Applicable.

Field	Bits	Type	Description
STH	1	rw	**Send Trigger High** CPU Send packet to LAN/WAN DMA trigger (high priority), self_clear
STL	0	rw	**Send Trigger Low** CPU Send packet to LAN/WAN DMA trigger (normal priority), self_clear

Search CMD

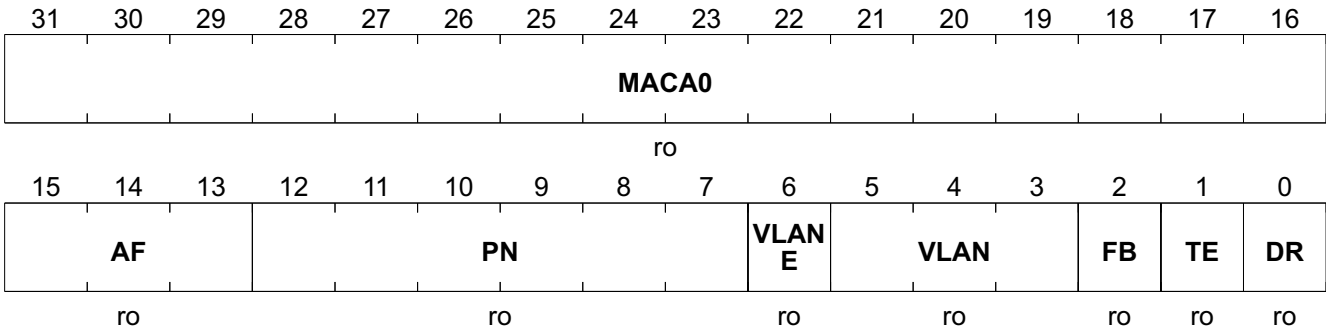
Srch_cmd	Offset	Reset Value
Search CMD	$4C_H$	0_H



Field	Bits	Type	Description
Res	31:2		**Reserved** Not Applicable.
SA	1	rw	**Search Again** Search for the next available address, self_clear (program again after data_rdy).
SS	0	rw	**Search Start** Searching from the start of address table, self_clear.

Address ST0

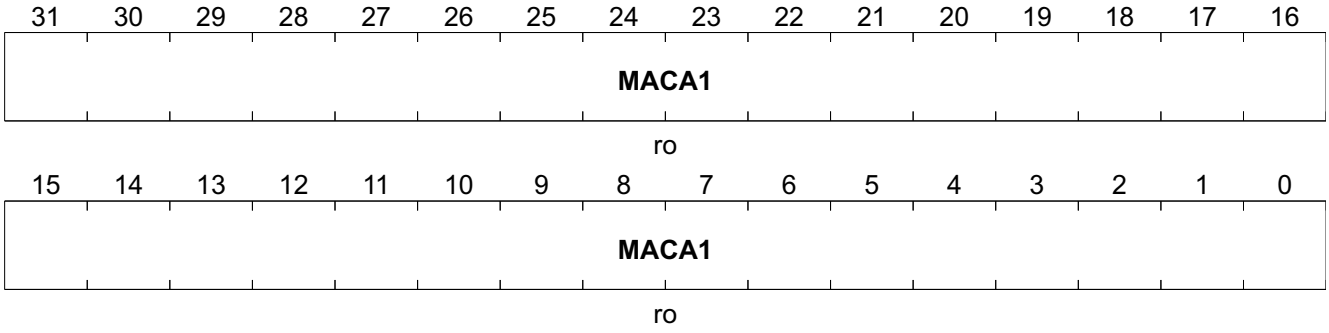
ADDR_st0	Offset	Reset Value
Address ST0	50_H	0_H



Field	Bits	Type	Description
MACA0	31:16	ro	**MAC Address 15:0**
AF	15:13		**Age Field** 000_B , Empty 001 to 110_B, Existed MAC 111_B , Static address
PN	12:7		**Port Map Number** Port number (refer to B+58, bit7-12)
VLANE	6		**VLAN Field is Enable**
VLAN	5:3		**VLAN Number**
FB	2		**Filter Bit**
TE	1		**Search to Table End** 1_B , Hit the end of MAC address table
DR	0		**Data is Ready** ADDR_st, ADDR_srch0 and ADDR_srch1 are ready, read_clear

ADDR St1

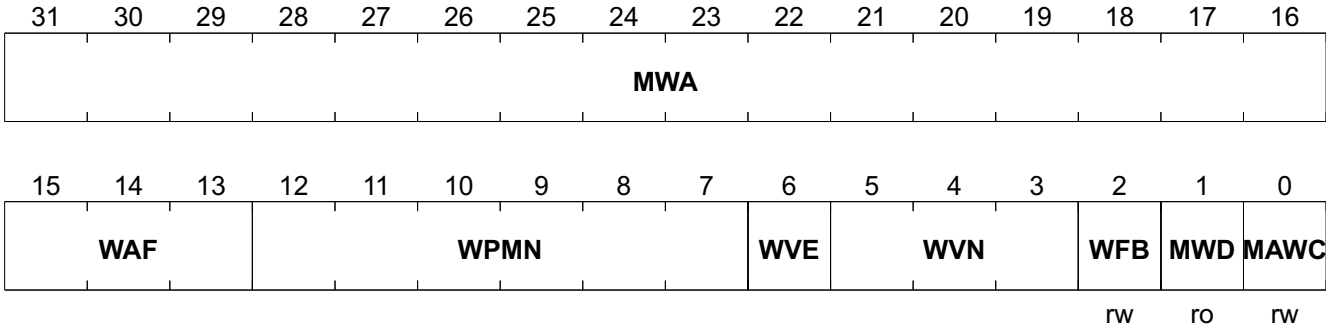
ADDR_st1 **Offset** **Reset Value**
Address St1 **54_H** **0_H**



Field	Bits	Type	Description
MACA1	31:0	ro	**MAC Address 47:16**

MAC Write Address 0

MAC_wt0 **Offset** **Reset Value**
MAC Write Address 0 **58_H** **0_H**



Field	Bits	Type	Description
MWA	31:16	rw	**MAC Write Address 15:0**
WAF	15:13	rw	**Write Age Field** 000_B , Empty 001 to 110_B, Existed MAC 111_B , Static address
WPMN	12:7	rw	**Write Port Map Number**
WVE	6	rw	**Write VLAN Enable**
WVN	5:3	rw	**Write VLAN Number**
WFB	2	rw	**Write Filter Bit**

Field	Bits	Type	Description
MWD	1	ro	**MAC Write Done** 1_B , MAC address write complete, read_clear
MAWC	0	rw	**MAC Address Write Command** 1_B , The MAC write data is ready and write toMAC table, self_clear

MAC Write Address 1

MAC_wt1 **Offset** **Reset Value**

MAC Write Address 1 **$5C_H$** **0_H**

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

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

Field	Bits	Type	Description
MWA	31:0	rw	**MAC Write Address 47:16**

Bandwidth Control 0

BW_cntl0 **Offset** **Reset Value**

Bandwidth Control 0 **60_H** **0_H**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	P3RBC			Res	P3TBC			Res	P2RBC			Res	P2TBC		
	rw				rw				rw				rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	P1RBC			Res	P1TBC			Res	P0RBC			Res	P0TBC		
	rw				rw				rw				rw		

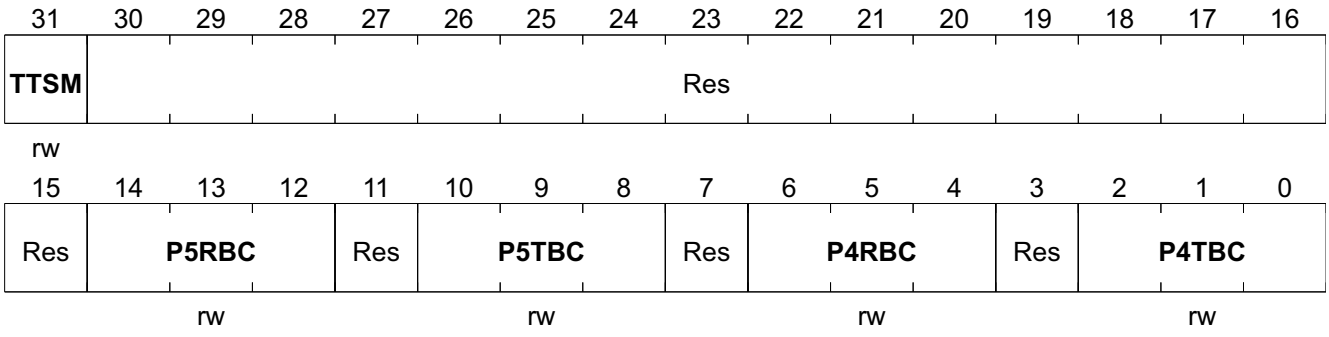
Field	Bits	Type	Description
Res	31		**Reserved** Not Applicable.

Field	Bits	Type	Description
P3RBC	30:28	rw	**Port 3 Receive Bandwidth Control** 000_B , Disable 001_B , 64K bit per second 010_B , 128K bit per second 011_B , 256K bit per second 100_B , 512K bit per second 101_B , 1M bit per second 110_B , 4M bit per second 111_B , 10M bit per second
Res	27		**Reserved** Not Applicable.
P3TBC	26:24	rw	**Port 3 Transmit Bandwidth Control** 000_B , Disable 001_B , 64K bit per second 010_B , 128K bit per second 011_B , 256K bit per second 100_B , 512K bit per second 101_B , 1M bit per second 110_B , 4M bit per second 111_B , 10M bit per second
Res	23		**Reserved** Not Applicable.
P2RBC	22:20	rw	**Port 2 Receive Bandwidth Control** Please refer **P3RBC** for bandwidth define.
Res	19		**Reserved** Not Applicable.
P2TBC	18:16	rw	**Port 2 Transmit Bandwidth Control** Please refer **P3RBC** for bandwidth define.
Res	15		**Reserved** Not Applicable.
P1RBC	14:12	rw	**Port 1 Receive Bandwidth Control** Please refer **P3RBC** for bandwidth define.
Res	11		**Reserved** Not Applicable.
P1TBC	10:8	rw	**Port 1 Transmit Bandwidth Control** Please refer **P3RBC** for bandwidth define.
Res	7		**Reserved** Not Applicable.
P0RBC	6:4	rw	**Port 0 Receive Bandwidth Control** Please refer **P3RBC** for bandwidth define.
Res	3		**Reserved** Not Applicable.

Field	Bits	Type	Description
P0TBC	2:0	rw	**Port 0 Transmit Bandwidth Control** 000_B , Disable 001_B , 64K bit per second 010_B , 128K bit per second 011_B , 256K bit per second 100_B , 512K bit per second 101_B , 1M bit per second 110_B , 4M bit per second 111_B , 10M bit per second

Bandwidth Control 1

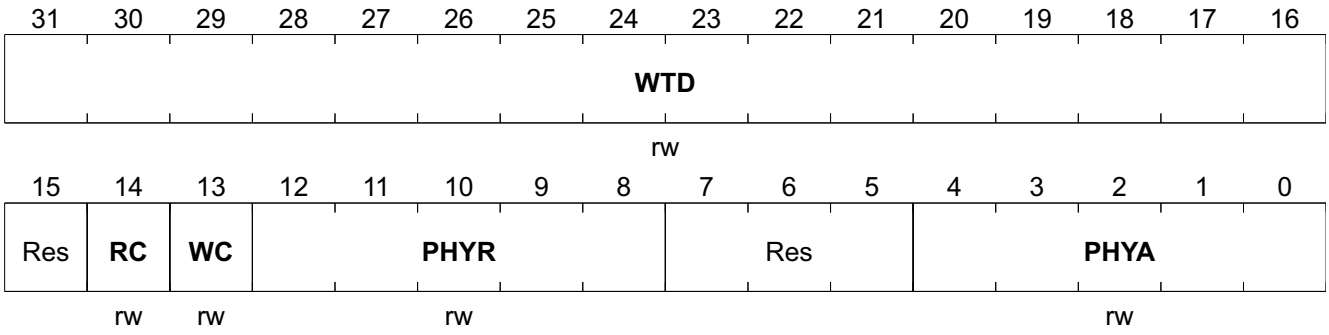
BW_cntl1	Offset	Reset Value
Bandwidth Control 1	**64_H**	**0_H**



Field	Bits	Type	Description
TTSM	31	rw	**The Transmit Traffic Shaper Mode** 0_B , Best effort mode (default) 1_B , Average Inter Packet Gap (IPG) in the 1 second period
Res	30:15		**Reserved** Not Applicable.
P5RBC	14:12	rw	**Port 5 Receive Bandwidth Control** Please refer **P3RBC** for bandwidth define.
Res	11		**Reserved** Not Applicable.
P5TBC	10:8	rw	**Port 5 Transmit Bandwidth Control** Please refer **P3RBC** for bandwidth define.
Res	7		**Reserved** Not Applicable.
P4RBC	6:4	rw	**Port 4 Receive Bandwidth Control** Please refer **P3RBC** for bandwidth define.
Res	3		**Reserved** Not Applicable.
P4TBC	2:0	rw	**Port 4 Transmit Bandwidth Control** Please refer **P3RBC** for bandwidth define.

PHY Control 0

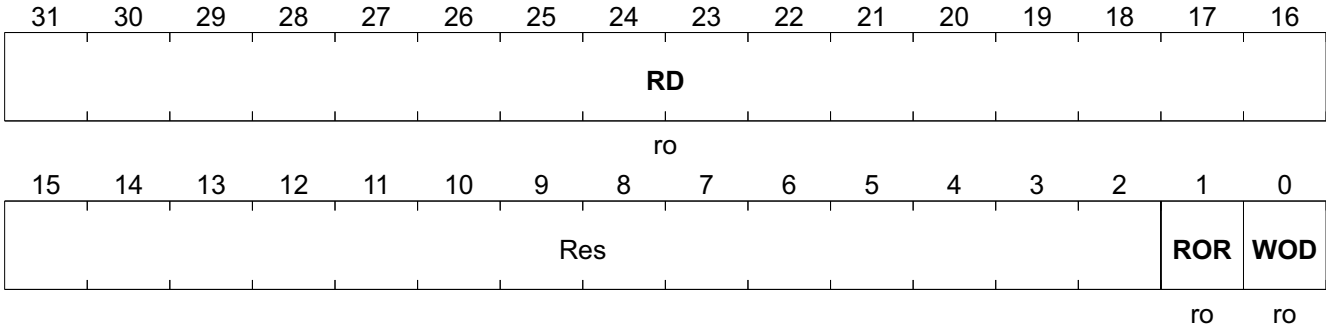
PHY_cntl0	Offset	Reset Value
PHY Control 0	68_H	0_H



Field	Bits	Type	Description
WTD	31:16	rw	**The Data be Written into the PHY**
Res	15		**Reserved** Not Applicable.
RC	14	rw	**Read Command, self_clear**
WC	13		**Write Command, self_clear**
PHYR	12:8		**PHY Register Address**
Res	7:5		**Reserved** Not Applicable.
PHYA	4:0	rw	**PHY Address**

PHY Control 1

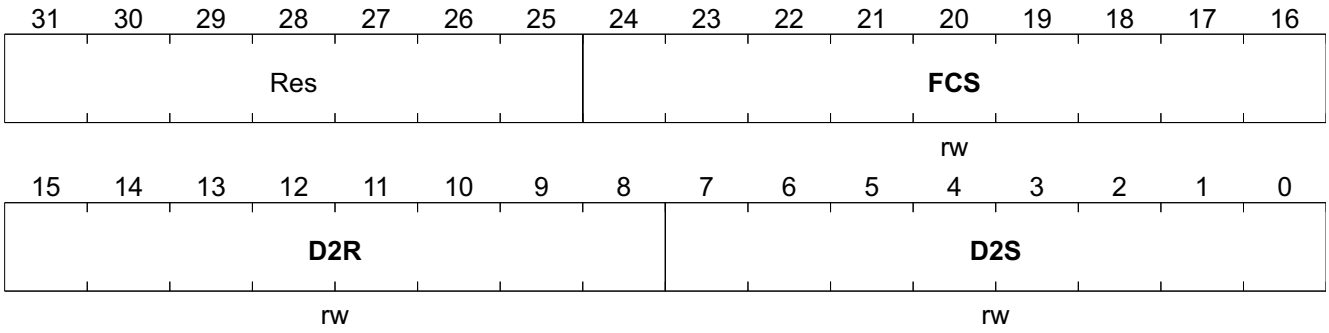
PHY_cntl1	Offset	Reset Value
PHY Control 1	$6C_H$	0_H



Field	Bits	Type	Description
RD	31:16	ro	**The Read Data**
Res	15:2		**Reserved** Not Applicable.
ROR	1	ro	**Read Operation is Complete and Data is Ready, read_clear**
WOD	0		**Write Operation is Done, read_clear**

Switch Control Threshold

FC_th	Offset	Reset Value
Switch Control Threshold	70_H	$DC866A_H$



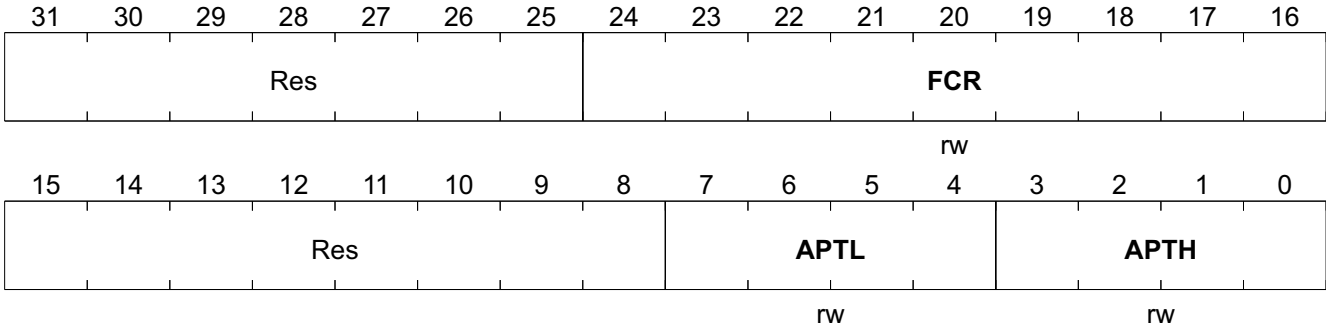
Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.

Field	Bits	Type	Description
FCS	24:16	rw	**Switch Flow Control Set Threshold, 220 Free Blocks**
D2R	15:8		**Switch Drop 2 Release Threshold, 134 Free Blocks**
D2S	7:0		**Switch Drop 2 Set Threshold, 106 Free Blocks**

*Note: The working global thresholds = (register value) * 2, The Drop1_set[7:0] threshold default value = 137 blocks, The default working Drop1_set threshold = 137 x 2 =274.*

Adj Port Threshold

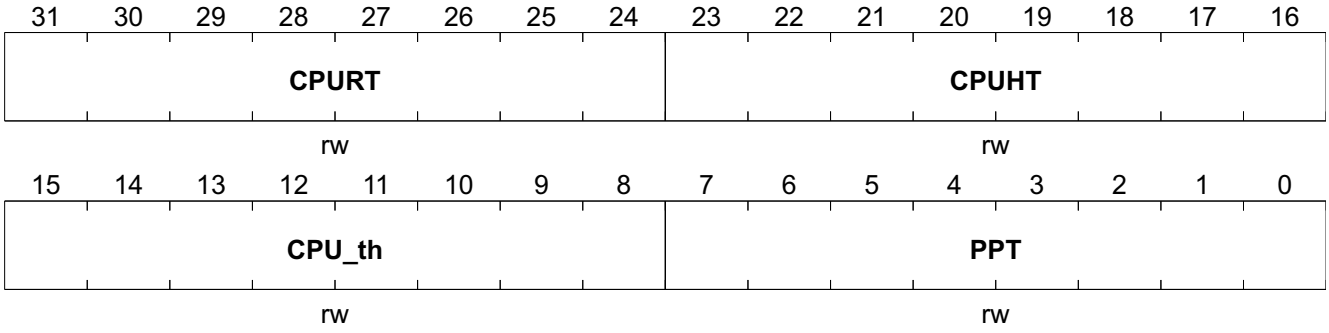
adj_port_th	Offset	Reset Value
Adj Port Threshold	**74_H**	**$10C2033_H$**



Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
FCR	24:16	rw	**Switch Flow Control Release Threshold, 268 Free Blocks**
GPT	15:8	rw	**Giga Port Buffer Threshold, 32 Occupied Blocks**
APTL	7:4	rw	**Per Port Guaranteed High Priority pkt, 3 Blocks**
APTH	3:0		**Per Port Guaranteed Normal Priority pkt, 3 Blocks**

Port Threshold

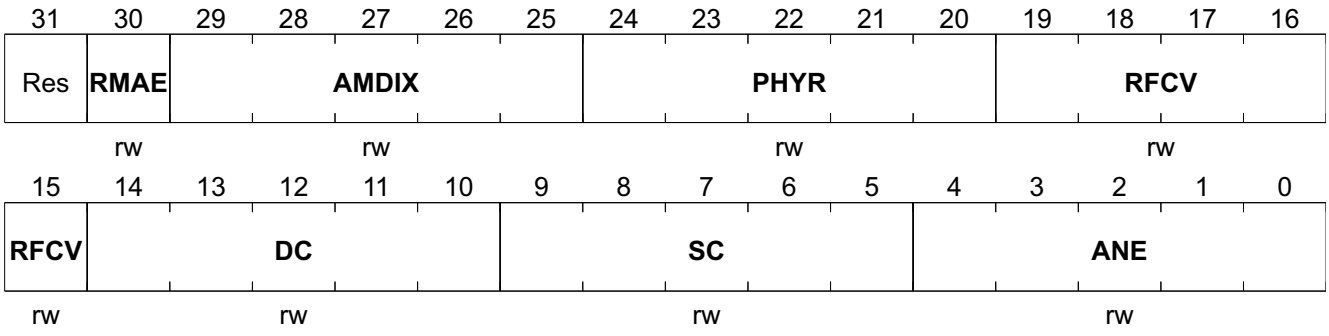
Port_th	Offset	Reset Value
Port Threshold	78_H	$8478300D_H$



Field	Bits	Type	Description
CPURT	31:24	rw	**The CPU Hold Release Threshold, Default 132 Free Block**
CPUHT	23:16		**The CPU Hold Threshold for All Ports, Default 120 Free Block**
CPU_th	15:8		**CPU Port Buffer Threshold, 48 Occupied Blocks**
PPT	7:0		**Per Port Buffer Threshold, 13 Occupied Blocks**

PHY Control 2

PHY_cntl2	Offset	Reset Value
PHY Control 2	$7C_H$	$BE0FFFFF_H$

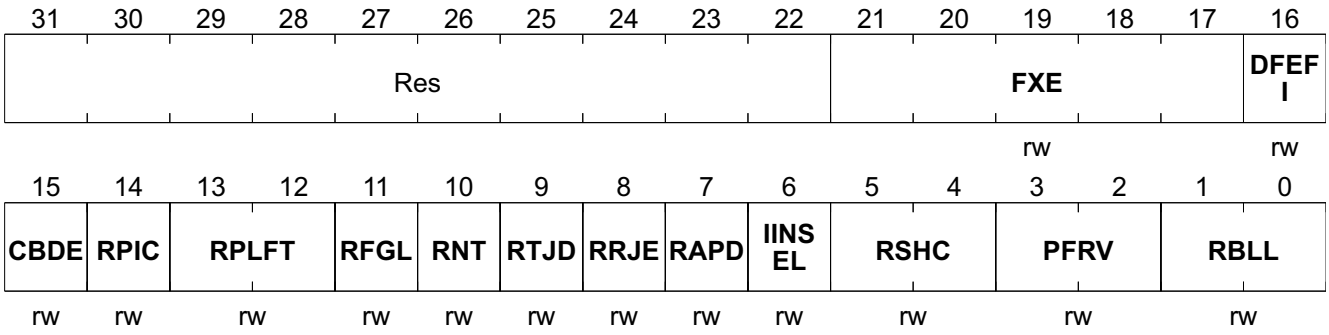


Field	Bits	Type	Description
Res	31		**Reserved** Not Applicable.

Field	Bits	Type	Description
RMAE	30	rw	**Recommend MCC Average Enable** Per port PHY auto MDIX enable. 0_B , Default value
AMDIX	29:25		**Auto MDIX enable** *Note: [25] = port0, [26] = port 1 etc…* 0_B , disable auto MDIX. 1_B , enable auto MDIX. (default)
PHYR	24:20		**PHY Reset** *Note: [20] = port0, [21] = port 1 etc…* 0_B , Reset(default) 1_B , Normal
RFCV	19:15		**Recommended FC Value (reg4, bit10)** *Note: [15] = port0, [16] = port 1 etc…* 0_B , No forced 1_B , FC_rec ON
DC	14:10		**Duplex Control** *Note: [10] = port0, [11] = port 1 etc…* 0_B , Half 1_B , Full
SC	9:5		**Speed Control** *Note: [5] = port0, [6] = port 1 etc…* 0_B , 10M 1_B , 100M
ANE	4:0		**Auto Negotiation Enable** *Note: [0] = port0, [1] = port 1 etc…* 1_B , Enable

PHY Control 3

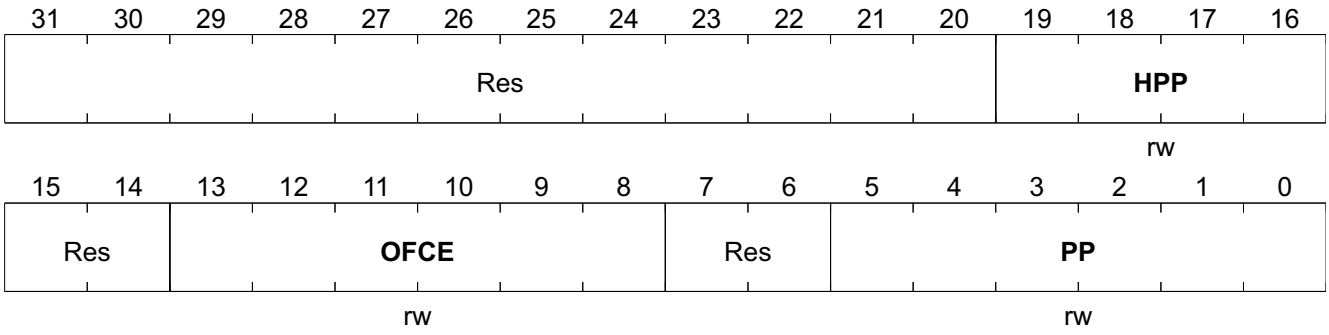
PHY_cntl3	Offset	Reset Value
PHY Control 3	80_H	$1420B_H$



Field	Bits	Type	Description
Res	31:22		**Reserved** Not Applicable.
FXE	21:17	rw	**Per Port FX Enable** *Note: [17] = port0, [18] = port 1 etc…* 0_B , Default value, disable 1_B , Enable FX
DFEFI	16		**Disable far_end Fault Indication**
CBDE	15		**Recommend Cable Broken Detect Enable**
RPIC	14		**Recommend Polarity LIU/LID Interval Check**
RPLFT	13:12		**Recommend Polarity Link Fail Timer Select** 00_B , Default value (2 sec) 01_B , 3 sec 10_B , 4 sec 11_B , 8 sec
RFGL	11		**Recommend Force Good Link**
RNT	10		**Recommend Normal Threshold**
RTJD	9		**Recommend Tansmit Jabber Disable**
RRJE	8		**Recommend Receive Jabber Enable**
RAPD	7		**Recommend Auto Polarity Disable**
IINSEL	6		**IINSEL** 0_B , Default value
RSHC	5:4		**Recommend Sample-Hold Current**
PFRV	3:2		**Pre-filter Recommend Value**
RBLL	1:0		**Recommend Base-line Limit**

Priority Control

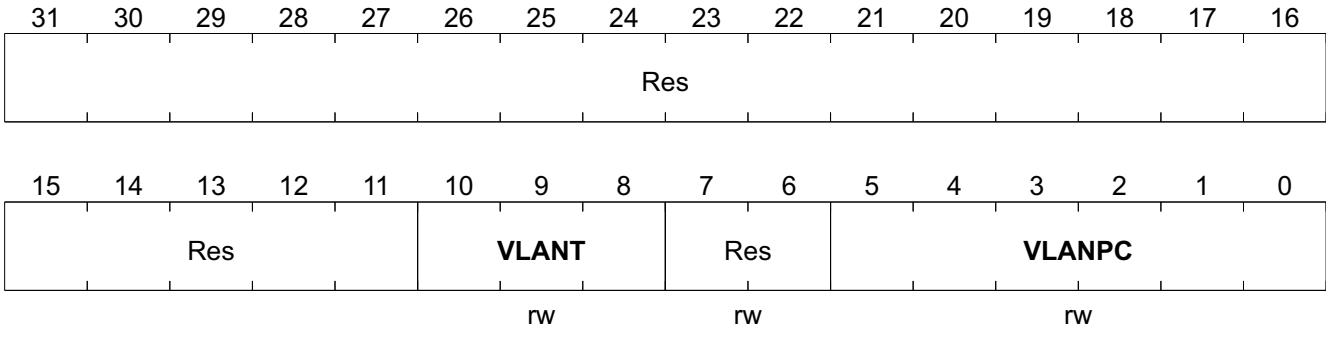
Pri_cntl	Offset	Reset Value
Priority Control	**84_H**	**0_H**



Field	Bits	Type	Description
Res	31:20		**Reserved** Not Applicable.
HPP	19:16	rw	**The Proportion of Normal and High Priority Packet** The transmit ratio between high/low queue will be 8xN:1, N is the value. 0000_B, Unlimited 0001_B, 8:1 0010_B, 16:1 0011_B, 24:1,....etc
Res	15:14		**Reserved** Not Applicable.
OFCE	13:8	rw	**Auto-turn-off FC** Auto-turn-off FC when the programmed ports receive priority packet. *Note: [8] = port0, [9] = port 1 etc…* 0_B , Disable
Res	7:6		**Reserved** Not Applicable.
PP	5:0	rw	**Port Priority** Force all the packet from the programmed port(s) are priority. *Note: [0] = port0, [1] = port 1 etc…* 0_B , Normal 1_B , high priority

VLAN Priority

VLAN_pri	Offset	Reset Value
VLAN Priority	88_H	400_H

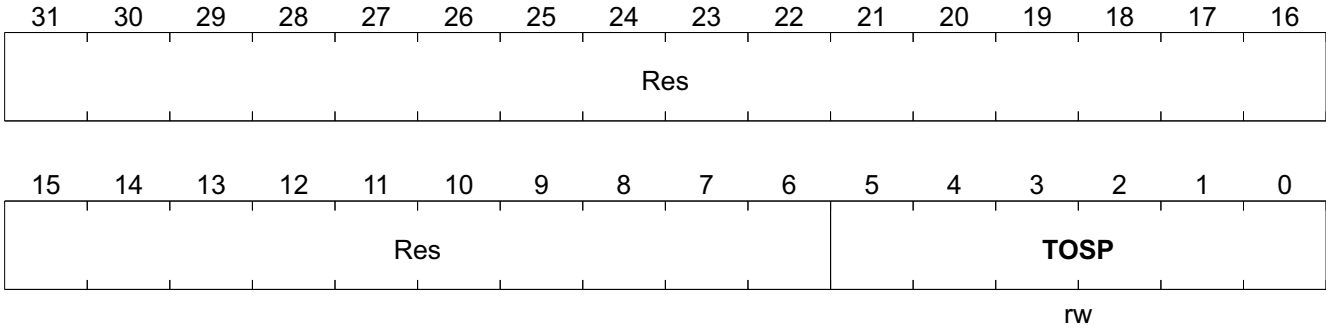


Field	Bits	Type	Description
Res	31:11		**Reserved** Not Applicable.

Field	Bits	Type	Description
VLANT	10:8	rw	**VLAN High Priority Threshold** If the priority field in VLAN tag is not less than this threshold, then the packet is high priority. else it is low priority. 100_B , Default value
Res	7:6		**Reserved** Not Applicable.
VLANPC	5:0		**Enable VLAN Priority Check** *Note: [0] = port0, [1] = port 1 etc…* 0_B , Disable

TOS Enable

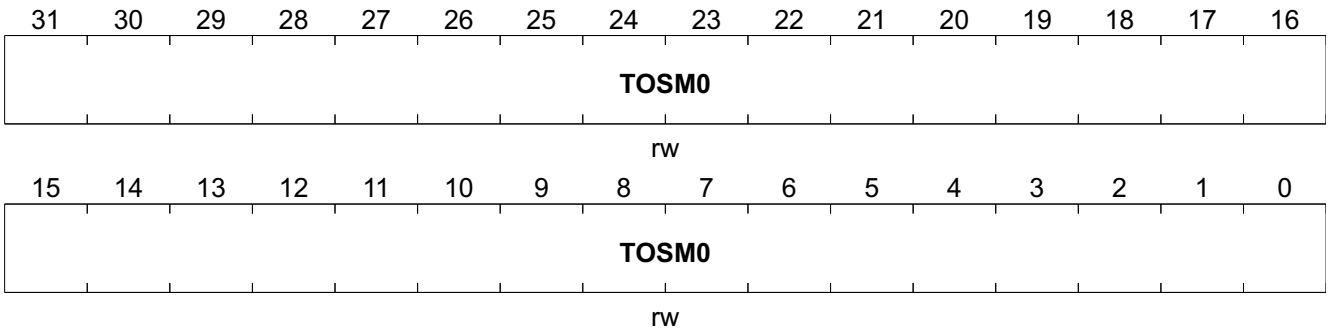
TOS_en **Offset** **Reset Value**
TOS Enable **$8C_H$** **0_H**



Field	Bits	Type	Description
Res	31:6		**Reserved** Not Applicable.
TOSP	5:0	rw	**Enable TCP/IP TOS Priority Check** *Note: [0] = port0, [1] = port 1 etc…* 0_B , Disable 1_B , Enable

TOS Map 0

TOS_map0 **Offset** **Reset Value**
TOS Map 0 **90_H** **0_H**

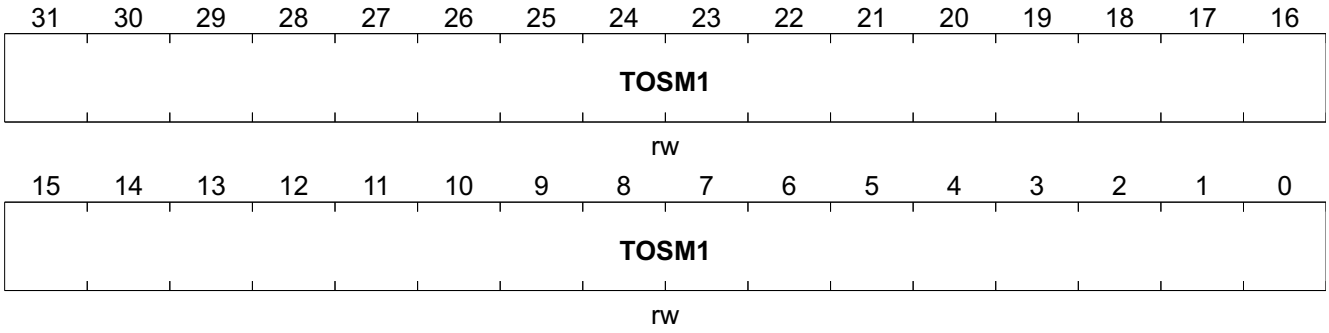


Field	Bits	Type	Description
TOSM0	31:0	rw	**TOS Bit Map 31:0** The TOS bit map totally has 64 bits. The bit define the incoming packet will be high or normal priority. The TOS value(0~63) of IP packets is used to select the relative bit map. 0_B , Normal priority 1_B , High priority

TOS Map 1

TOS_map1 **Offset** **Reset Value**

TOS Map 1 **94_H** **0_H**

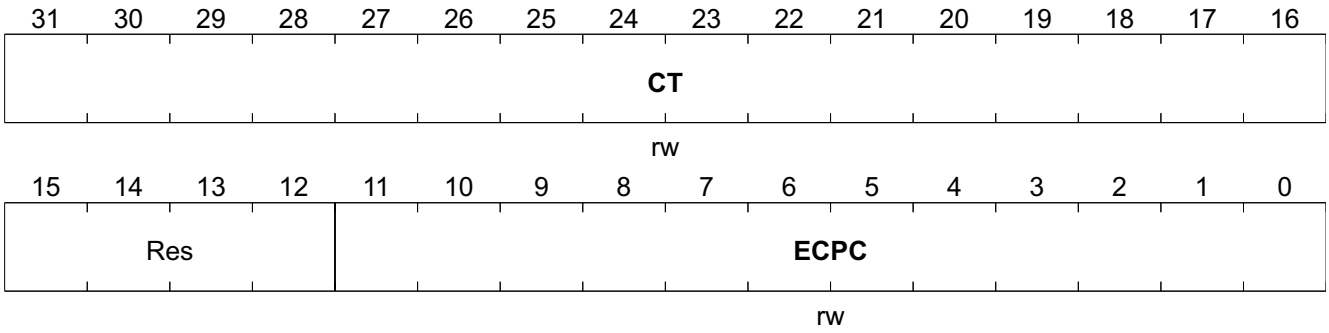


Field	Bits	Type	Description
TOSM1	31:0	rw	**TOS Bit Map 64:32**

Custom Priority 1

Custom_pri1 **Offset** **Reset Value**

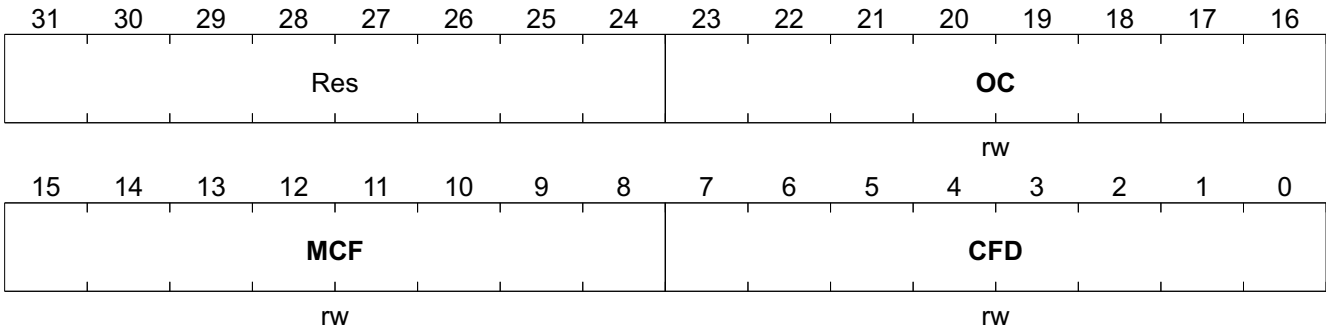
Custom Priority 1 **98_H** **0_H**



Field	Bits	Type	Description
CT	31:16	rw	**Custom Type** This field defines the value of Customer Type that will be matched at the Type field of ether packets.
Res	15:12		**Reserved** Not Applicable.
ECPC	11:0	rw	**Enable Custom Packet Check** *Note: Enable custom packet check :bit [1:0] for port0, [3:2] for port1, [5:4] for port2, ...* 00_B , Disable, default 01_B , Treat as high priority 10_B , Filtered 11_B , Reserved

Custom Priority 2

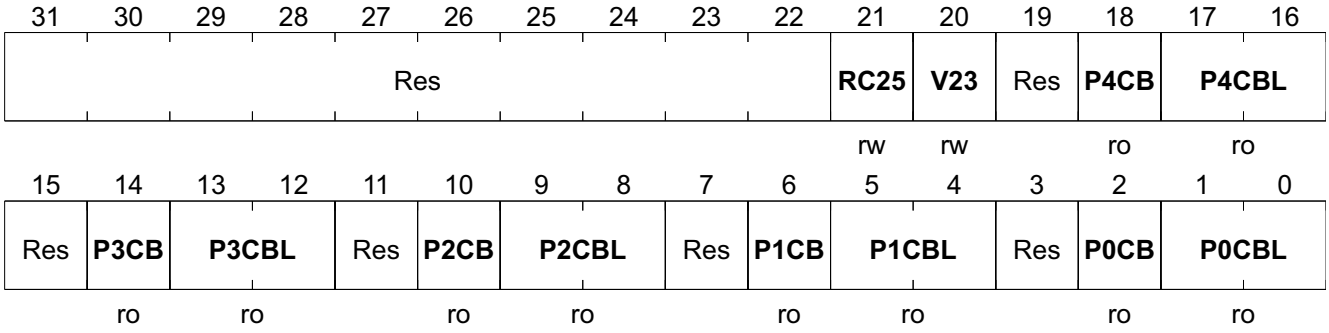
Custom_pri2	Offset	Reset Value
Custom Priority 2	$9C_H$	0_H



Field	Bits	Type	Description
Res	31:24		**Reserved** Not Applicable.
OC	23:16	rw	**Offset Count from SA 7:0** This offset define the data will be extracted from the packets. The data will be compared with the Custome Field and Mask. The offset is counted from SA0 field of packet. If VLAN type found, it will add 4-byte automatically.
MCF	15:8		**Mask of Custom Field** The mask data for the Custom Field.
CFD	7:0		**Custom Field Define** This data defines the Custom Field that will be treat as higher priority or filtered.

PHY Control 4

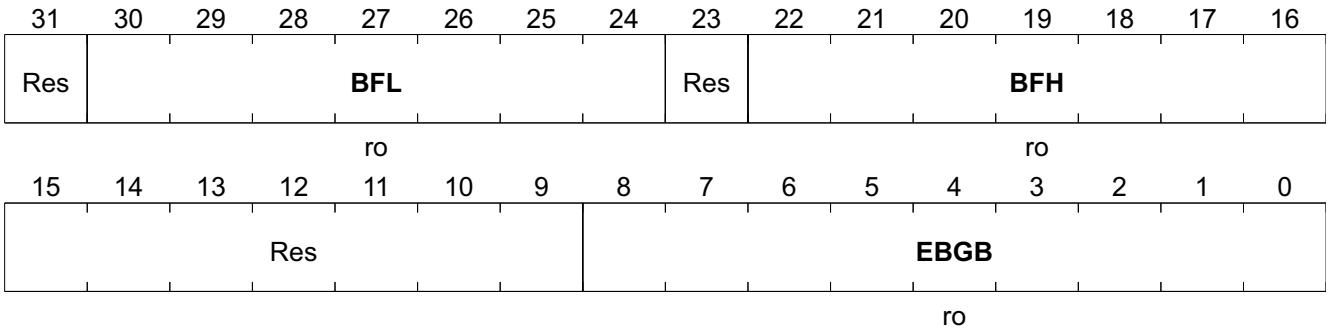
PHY_cntl4	Offset	Reset Value
PHY Control 4	$A0_H$	0_H



Field	Bits	Type	Description
Res	31:22		**Reserved** Not Applicable.
RC25	21	rw	**Rom Code 25** 0_B , 2.2 V (default) 1_B , Fix the ROM code to 2.5 V
V23	20	rw	**Volt 23** 0_B , 2.2 V (default) 1_B , 10BaseT voltage 2.3 V
Res	19		**Reserved** Not Applicable.
P4CB	18	ro	**Port 4 Cable Broken**
P4CBL	17:16	ro	**Port 4 Cable Broken Length**
Res	15		**Reserved** Not Applicable.
P3CB	14	ro	**Port 3 Cable Broken**
P3CBL	13:12	ro	**Port 3 Cable Broken Length**
Res	11		**Reserved** Not Applicable.
P2CB	10	ro	**Port 2 Cable Broken**
P2CBL	9:8	ro	**Port 2 Cable Broken Length**
Res	7		**Reserved** Not Applicable.
P1CB	6	ro	**Port 1 Cable Broken**
P1CBL	5:4	ro	**Port 1 Cable Broken Length**
Res	3		**Reserved** Not Applicable.
P0CB	2	ro	**Port 0 Cable Broken**
P0CBL	1:0	ro	**Port 0 Cable Broken Length**

Empty Control

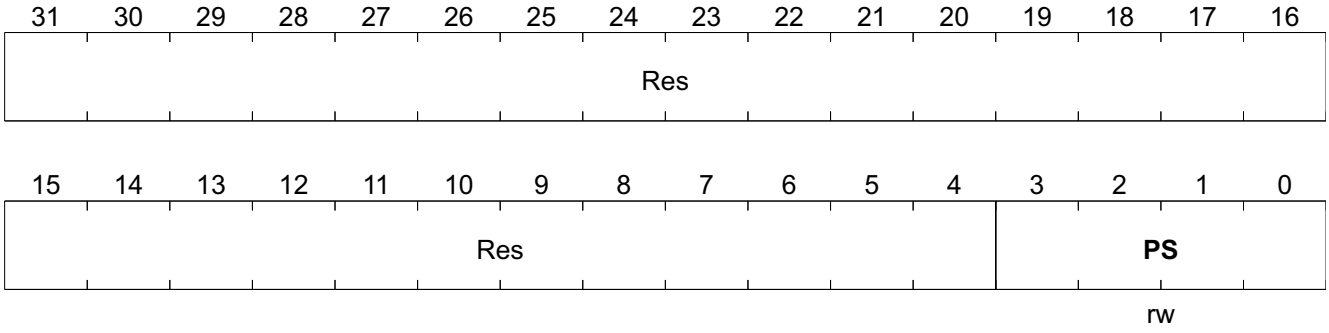
Empty_cnt	Offset	Reset Value
Empty Control	$A4_H$	0_H



Field	Bits	Type	Description
Res	31		**Reserved** Not Applicable.
BFL	30:24	ro	**Low-pri Out-queue Full status for CPU and Port[5:0]**
Res	23		**Reserved** Not Applicable.
BFH	22:16	ro	**The High-pri Out-queue Full status for CPU and Port[5:0]**
Res	15:9		**Reserved** Not Applicable.
EBGB	8:0	ro	**Empty Block in the Global Buffer**

Port Control Select

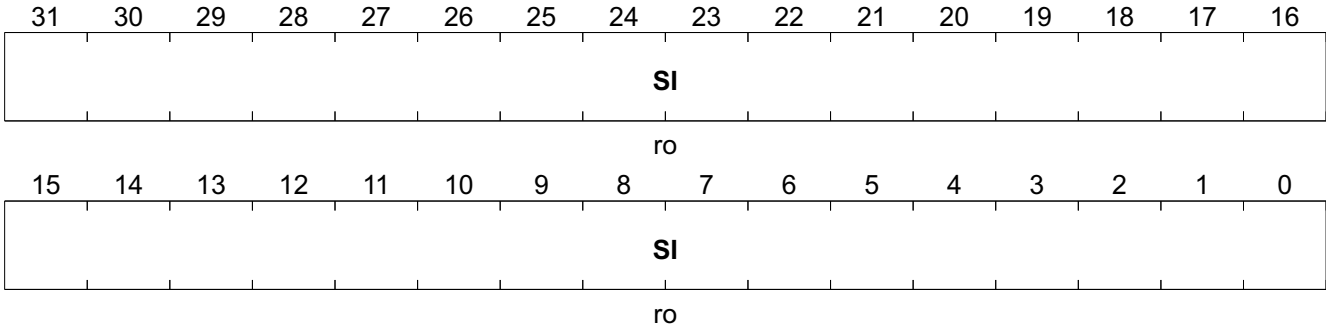
Port_cnt_sel	Offset	Reset Value
Port Control Select	$A8_H$	0_H



Field	Bits	Type	Description
Res	31:4		**Reserved** Not Applicable.
PS	3:0	rw	**Port Selected for the port_cnt**

Port Controller

Port_cnt	Offset	Reset Value
Port Controller	AC_H	0_H



Field	Bits	Type	Description
SI	31:0	ro	**Sel Info** If port_sel=0, [24:16] port0 high packet count [8:0] port0 low packet count If port_sel=1, [24:16] port1 high packet count [8:0] port1 low packet count If port_sel=2, [24:16] port2 high packet count [8:0] port2 low packet count If port_sel=3, [24:16] port3 high packet count [8:0] port3 low packet count If port_sel=4, [24:16] port4 high packet count [8:0] port4 low packet count If port_sel=5, [24:16] port5 high packet count [8:0] port5 low packet count If port_sel=6, [24:16] port6 high packet count [8:0] port6 low packet count If port_sel=7, [24:16] port7 high packet count [8:0] port7 low packet count If port_sel=9, [8:0] flow control status If port_sel=10, [24:16] testing [8:0] ever flow control port If port_sel=11, [24:16] no_pkt status [7:0] no_pkt_status If port_sel=12, [24:16] receive bandwidth control port, [8:0] transmit bandwidth control port

Int St

Note: All bits are “write 1 clear”

Int_st	Offset	Reset Value
Int St	$B0_H$	0_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res							CPUH	SDE	RDE	W1TE	W0TE	IP	PSC	Res	BCSS
							rw	rw	rw	rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MD	GET	CPUP	Res	P5QF	P4QF	P3QF	P2QF	P1QF	P0QF	LDF	HDF	RXLD	RXHD	SLD	SHD
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
CPUH	24	rw	**CPU Send Packets are Hold in Descriptor** Write one clear the status.
SDE	23		**Send Descriptor Error** Write one clear the status.
RDE	22		**Receive Descriptor Error** Write one clear the status.
W1TE	21		**Watchdog1 Timer Expired** Write one clear the status.
W0TE	20		**Watchdog0 Timer Expired** Write one clear the status.
IP	19		**Any Secured Port has Intruder Packets** Write one clear the status.
PSC	18		**Port Status Change** Any port change the link status (link-up to/from link-down). Write one clear the status.
Res	17		**Reserved** Not Applicable.
BCSS	16	rw	**Accumulated BC Count over BC_storm Setting** Write one clear the status.
MD	15		**Must Drop** All the buffer almost full. Write one clear the status.
GET	14		**Global empty-th** Write one clear the status.
CPUP	13		**CPU Port** Meet the CPU port pre-th & global empty-th. Write one clear the status.

Field	Bits	Type	Description
Res	12		**Reserved** Not Applicable.
P5QF	11	rw	**Meet the Port5 port-th & global empty-th** Write one clear the status.
P4QF	10		**Meet the Port4 port-th & global empty-th** Write one clear the status.
P3QF	9		**Meet the Port3 port-th & global empty-th** Write one clear the status.
P2QF	8		**Meet the Port2 port-th & global empty-th** Write one clear the status.
P1QF	7		**Meet the Port1 port-th & global empty-th** Write one clear the status.
P0QF	6		**Meet the Port0 port-th & global empty-th** Write one clear the status.
LDF	5		**Descriptor, “normal priority receive”, are Full** Write one clear the status.
HDF	4		**Descriptor, “high priority receive”, are Full** Write one clear the status.
RXLD	3		**DMA Receive one Normal Priority Packet to CPU** Write one clear the status.
RXHD	2		**DMA Receive one High Priority Packet to CPU** Write one clear the status.
SLD	1		**DMA Send one Normal Priority Packet to Switch** Write one clear the status.
SHD	0		**DMA Send one High Priority Packet to Switch** Write one clear the status.

Interrupt Mask

Note: 1: Mask the interrupt

Int_mask	**Offset**	**Reset Value**
Interrupt Mask	**$B4_H$**	**$1FDEFFF_H$**

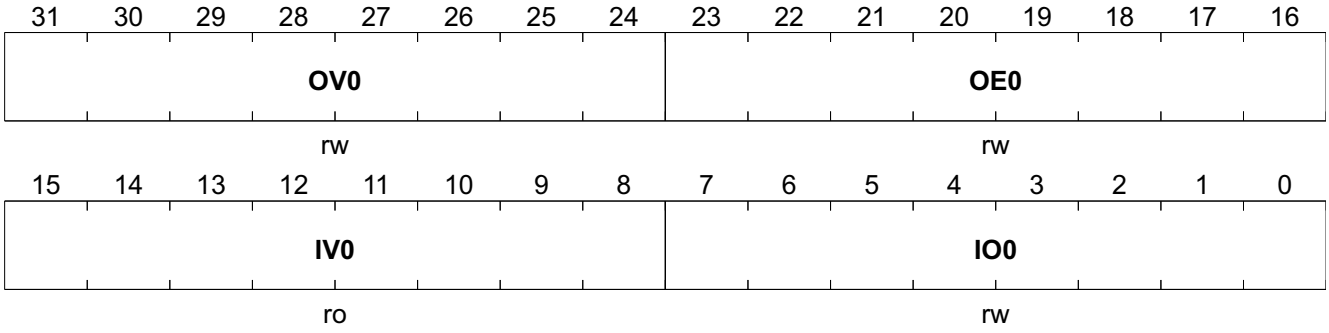
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res							**CPUH**	**SDE**	**RDE**	**W1TE**	**W0TE**	**MI**	**PSC**	Res	**BCS**
							rw	rw	rw	rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MD	**GQF**	**CPUQ**	Res	**P5QF**	**P4QF**	**P3QF**	**P2QF**	**P1QF**	**P0QF**	**LDF**	**HDF**	**RLD**	**RHD**	**SLD**	**SHD**
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
CPUH	24	rw	**Mask CPU Hold**
SDE	23		**Mask Send Description Error**
RDE	22		**Mask Receive Description Error**
W1TE	21		**Mask Wachdog1 Timer Expired**
W0TE	20		**Mask Wachdog0 Timer Expired**
MI	19		**Mask Intruder**
PSC	18		**Mask Port Status Change**
Res	17		**Reserved** Not Applicable.
BCS	16	rw	**Mask BC Storm**
MD	15		**Mast Drop**
GQF	14		**Mask Global Queue Full**
CPUQ	13		**Mask CPU Queue Full**
Res	12		**Reserved** Not Applicable.
P5QF	11	rw	**Mask Port5 Queue Full**
P4QF	10		**Mask Port4 Queue Full**
P3QF	9		**Mask Port3 Queue Full**
P2QF	8		**Mask Port2 Queue Full**
P1QF	7		**Mask Port1 Queue Full**
P0QF	6		**Mask Port0 Queue Full**
LDF	5		**Mask Low Descriptor Full**
HDF	4		**Mask High Descriptor Full**
RLD	3		**Mask Receive Low Done**
RHD	2		**Mask Receive High Done**
SLD	1		**Mask Send Low Done**
SHD	0		**Mask Send High Done**

GPIO Conf 0

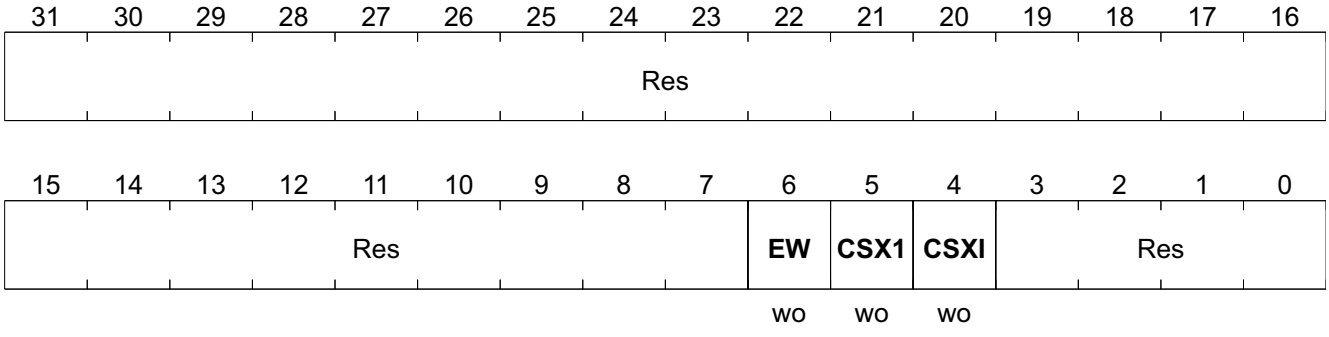
GPIO_conf0	Offset	Reset Value
GPIO Conf 0	$B8_H$	FF_H



Field	Bits	Type	Description
OV0	31:24	rw	**GPIO 7:0, Output Value if the Output Enable is one**
OE0	23:16	rw	**GPIO 7:0, Output Enable control** 0_B , Input (default) 1_B , Output Enable
IV0	15:8	ro	**GPIO 7:0, Input Value if in the Input Mode**
IO0	7:0	rw	**Reserved**

GPIO Conf 2

GPIO_conf2	Offset	Reset Value
GPIO Conf 2	BC_H	0_H

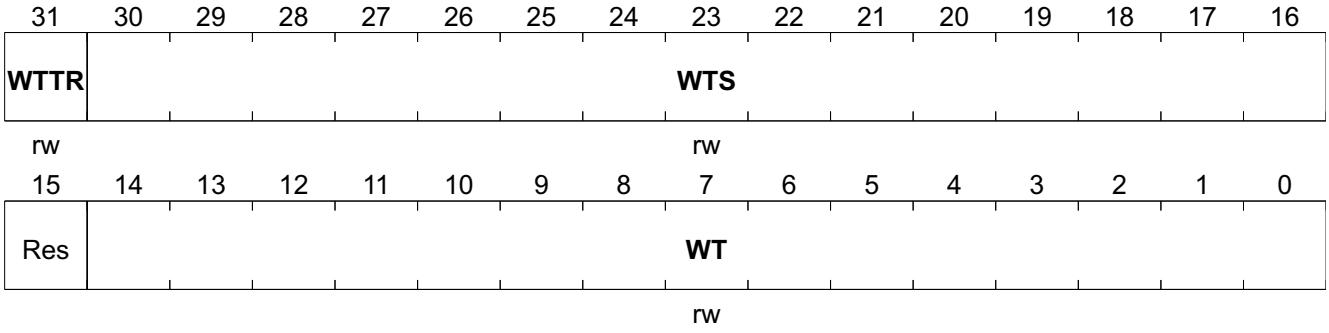


Field	Bits	Type	Description
Res	31:7		**Reserved** Not Applicable.

Field	Bits	Type	Description
EW	6	wo	**Enable Wait State** 0_B , Disable wait state for external IO control CSX0 and CSX1(default) 1_B , Enable the wait state pin GPIO[0] for external IO control CSX0 and CSX1
CSX1	5	wo	**Enable CSX1, INTX1 in GPIO 3:4** 0_B , Disable the CSX1 and INTX1 function(default) 1_B , Enable the CSX0 and INTX0 function
CSX0	4	wo	**Enable CSX0, INTX0 Interface in GPIO 1:2** 0_B , Disable the CSX0/INTX0 function.(default) 1_B , Enable the CSX0 and INTX0 function
Res	3:0		**Reserved** Not Applicable.

Watchdog 0

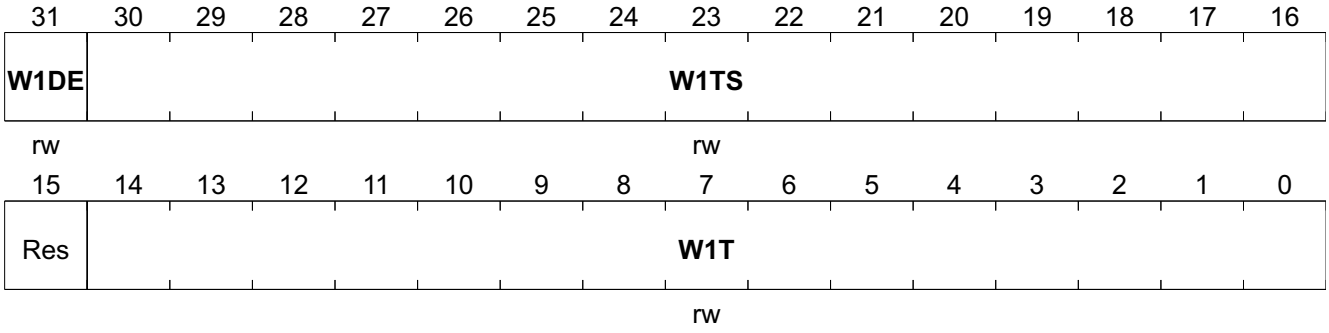
Wdog_0	Offset	Reset Value
Watchdog 0	$C0_H$	$7FFF0000_H$



Field	Bits	Type	Description
WTTR	31	rw	**Watchdog Timer Trigger Reset** 0_B , Disable(default) 1_B , Reset the whole chip if watchdog timer expired
WTS	30:16	rw	**Watchdog Timer Set** The time out setting of timer, if timer set is equal to timer, then it mean timer is expired. Maximum 32767.
Res	15		**Reserved** Not Applicable.
WT	14:0	rw	**Watchdog Timer** Count up timer, mask-able, write clear, unit 10 ms. If reach timer set mean time up and keep the counter until write-clear by software, maximum 327 s.

Watchdog 1

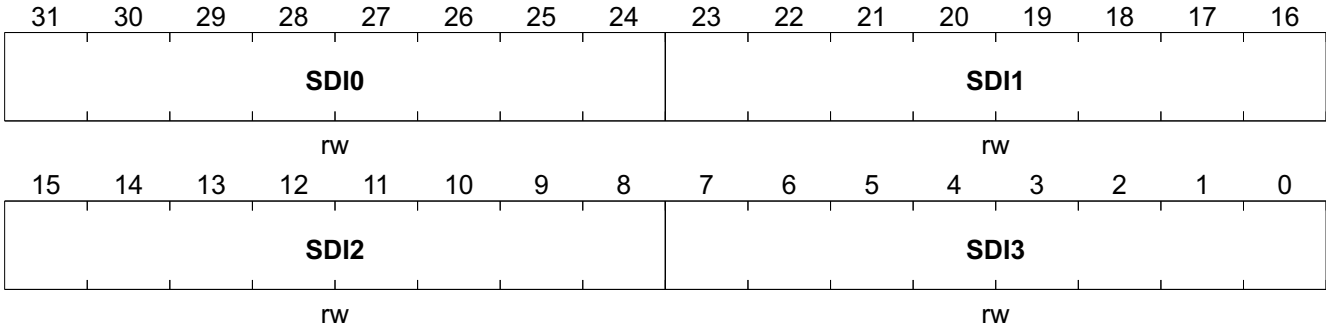
Wdog_1	Offset	Reset Value
Watchdog 1	$C4_H$	$7FFF0000_H$



Field	Bits	Type	Description
W1DE	31	rw	**Watchdog Timer Stop CPU Port Receiving** No issue flow control in ports. Auto-recover when the timer is cleaned. 0_B , Disable(default) 1_B , Force all to CPU packets drop if timer is expired, and no issue flow control in ports. Auto-recover when the timer is cleaned
W1TS	30:16	rw	**Watchdog 1 Timer Set** The time out setting of timer, if timer set is equal to timer, then it mean timer is expired. Maximum 32767.
Res	15		**Reserved** Not Applicable.
W1T	14:0	rw	**Watchdog 1 Timer** Count up timer, mask-able, write clear, unit 10 ms. If reach timer set mean time up and keep the counter until write-clear by software, maximum 327 s.

Swap In

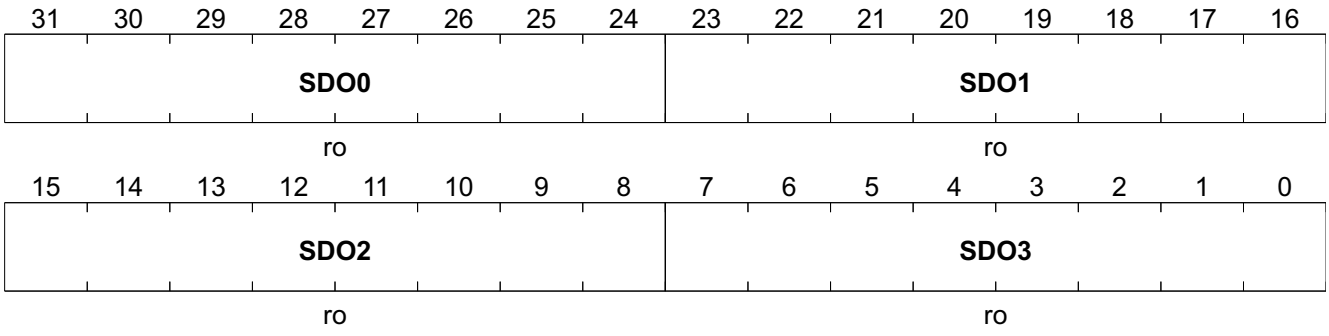
Swap_in	Offset	Reset Value
Swap In	$C8_H$	0_H



Field	Bits	Type	Description
SDI0	31:24	rw	**Swap_in 31:24 = Swap_out 7:0**
SDI1	23:16		**Swap_in 23:16 = Swap_out 15:8**
SDI2	15:8		**Swap_in 15:8 = Swap_out 15:8**
SDI3	7:0		**Swap_in 7:0 = Swap_out 31:24**

Swap Out

Swap_out	Offset	Reset Value
Swap Out	CC_H	0_H

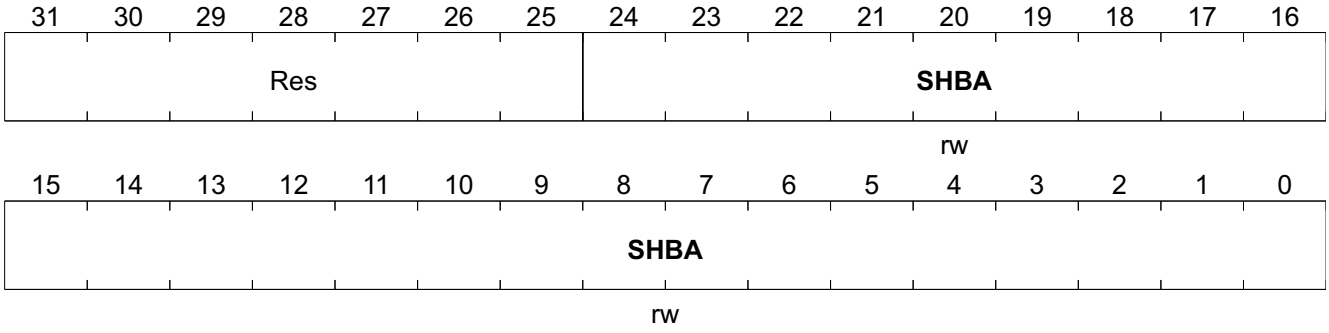


Field	Bits	Type	Description
SDO0	31:24	ro	**Swap_out 31:24 = Swap_in 7:0**
SDO1	23:16		**Swap_out 23:16 = Swap_in 15:8**
SDO2	15:8		**Swap_out 15:8 = Swap_in 23:16**
SDO3	7:0		**Swap_out 7:0 = Swap_in 31:24**

Send High Base Address

send_Hbaddr — **Offset** — **Reset Value**

Send High Base Address — **$D0_H$** — **0_H**

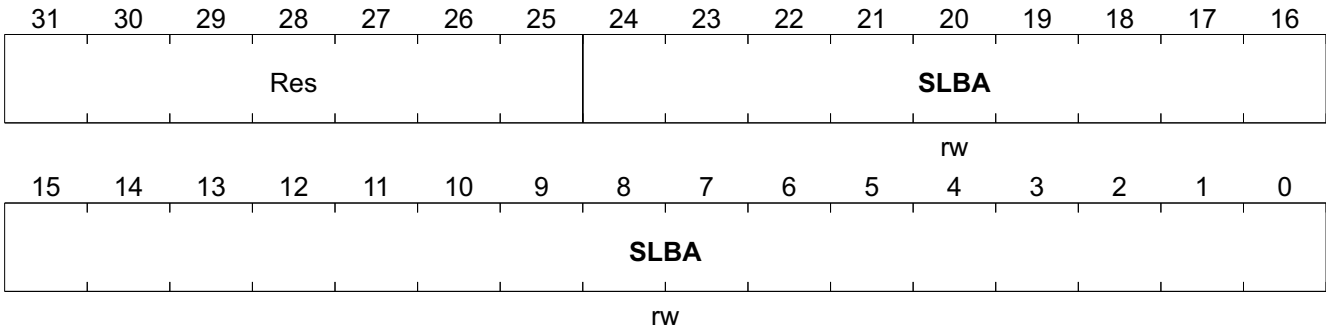


Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
SHBA	24:0	rw	**The Descriptor Base Address of CPU_to_SW (high priority)**

Send Low Base Address

send_Lbaddr — **Offset** — **Reset Value**

Send Low Base Address — **$D4_H$** — **0_H**

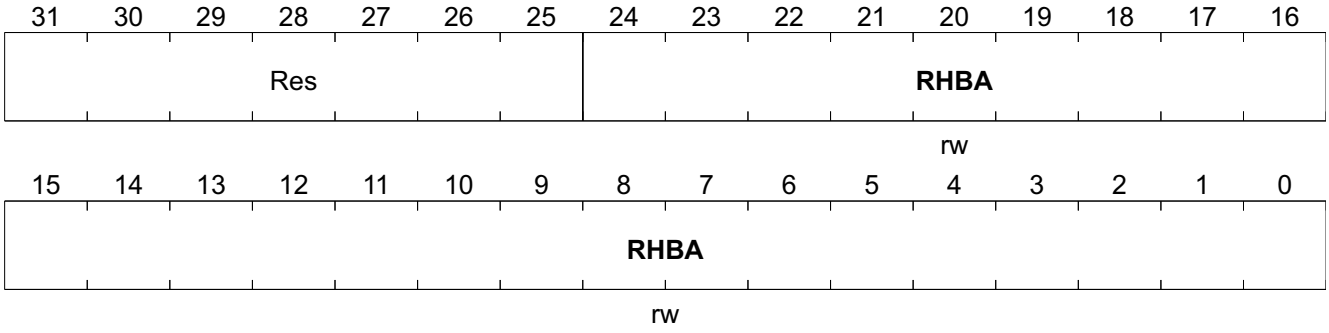


Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
SLBA	24:0	rw	**The Descriptor Base Address of CPU_to_SW (normal priority)**

Receive High Base Address

rec_Hbaddr — **Offset** $D8_H$ — **Reset Value** 0_H

Receive High Base Address

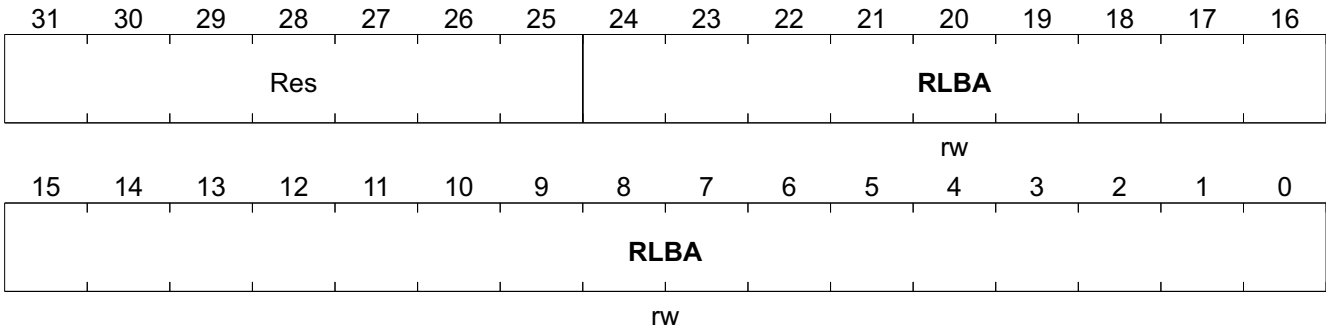


Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
RHBA	24:0	rw	**The Descriptor Base Address of SW_to_CPU (high priority)**

Receive Low Base Address

rec_Lbaddr — **Offset** DC_H — **Reset Value** 0_H

Receive Low Base Address

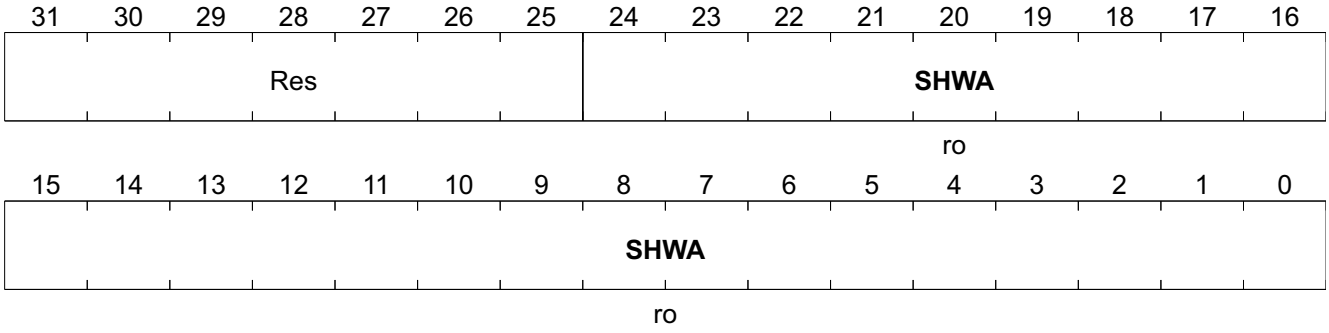


Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
RLBA	24:0	rw	**The Descriptor Base Address of SW_to_CPU (normal priority)**

Send High Working Address

send_Hwaddr — **Offset** — **Reset Value**

Send High Working Address — **$E0_H$** — **0_H**

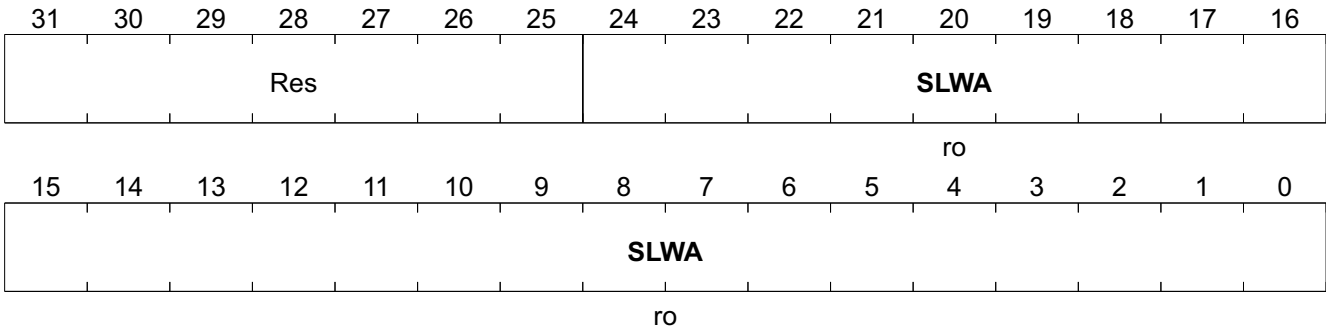


Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
SHWA	24:0	ro	**The Descriptor WORKING Address of CPU_to_SW (high priority)**

Send Low Working Address

send_Lwaddr — **Offset** — **Reset Value**

Send Low Working Address — **$E4_H$** — **0_H**

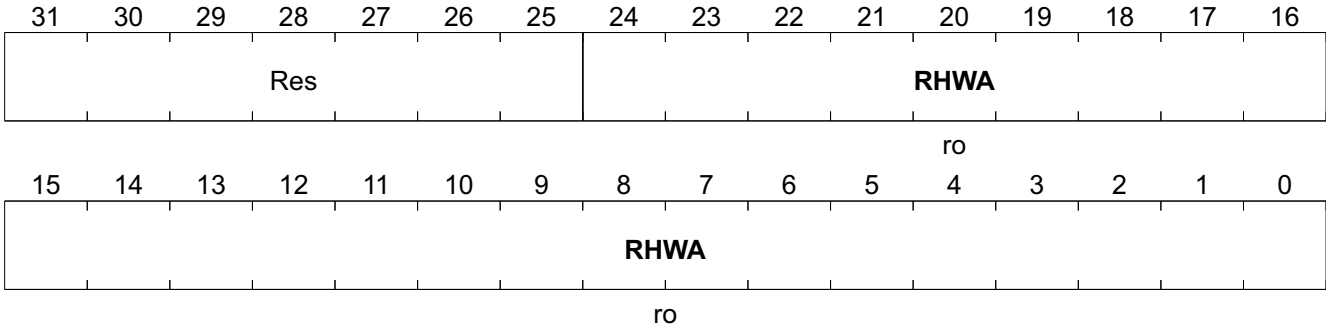


Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
SLWA	24:0	ro	**The Descriptor WORKING Address of CPU_to_SW (normal priority)**

Receive High Working Address

rec_Hwaddr — **Offset** — **Reset Value**

Receive High Working Address — **$E8_H$** — **0_H**

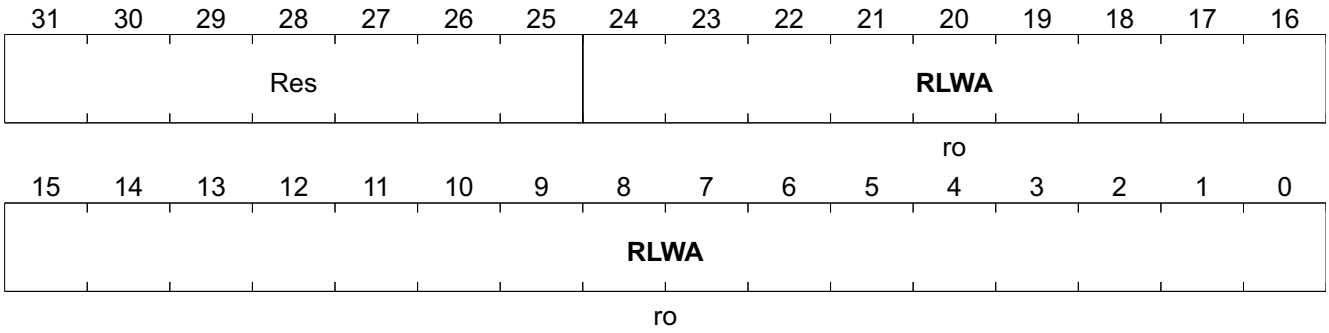


Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
RHWA	24:0	ro	**The Descriptor WORKING Address of SW_to_CPU (high priority)**

Receive Low Working Address

rec_Lwaddr — **Offset** — **Reset Value**

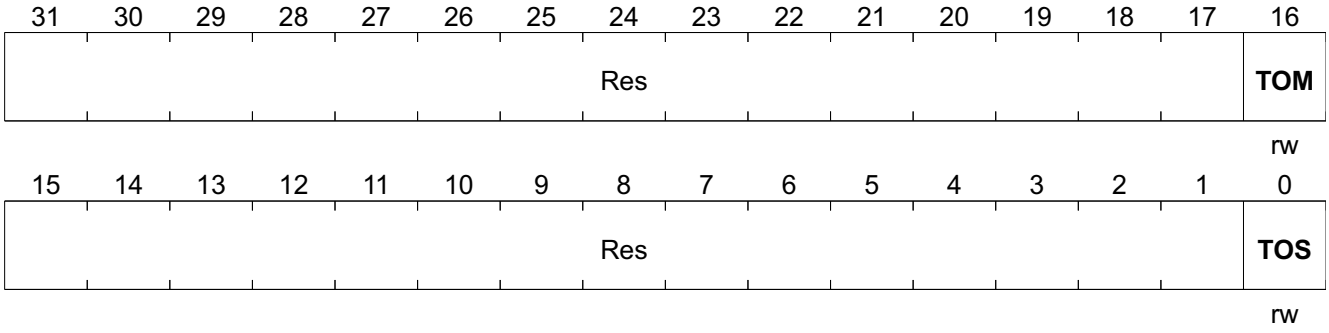
Receive Low Working Address — **EC_H** — **0_H**



Field	Bits	Type	Description
Res	31:25		**Reserved** Not Applicable.
RLWA	24:0	ro	**The Descriptor WORKING Address of SW_to_CPU (normal priority)**

Timer Interrupt

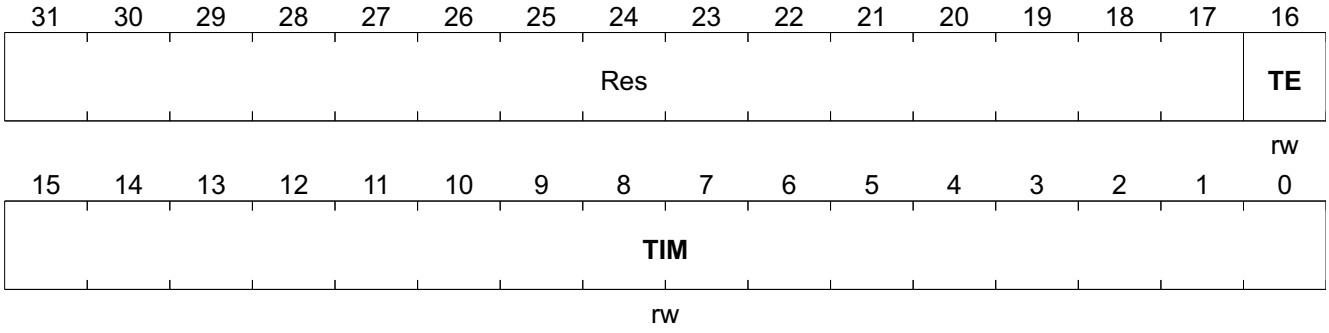
Timer_int	Offset	Reset Value
Timer Interrupt	$F0_H$	10000_H



Field	Bits	Type	Description
Res	31:17		**Reserved** Not Applicable.
TOM	16	rw	**Time Out Mask** 1_D , Mask the time out interrupt(default) 0_D , enable the time out interrupt
Res	15:1		**Reserved** Not Applicable.
TOS	0	rw	**Time Out Status** Write 1 clear the time out status. Show the time out status on read. The status is down count to zero on **Timer** register.

Timer

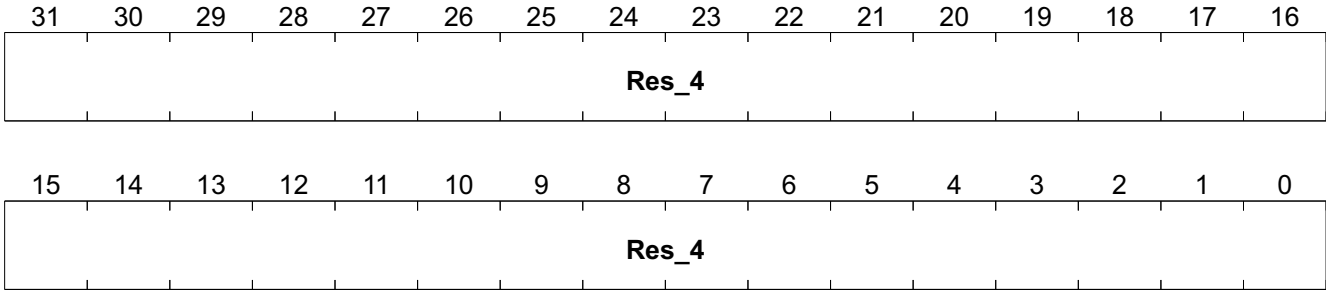
Timer	Offset	Reset Value
Timer	$F4_H$	$FFFF_H$



Field	Bits	Type	Description
Res	31:17		**Reserved** Not Applicable
TE	16	rw	**Timer Enable** 0_B , Disable the timer count down(default) 1_B , Enable the timer count down
TIM	15:0		**Timer** Unit 640 ns, auto-reload. Set the time out value on write. Read will return the current timer value. $FFFF_H$, Default value

Reserved 4

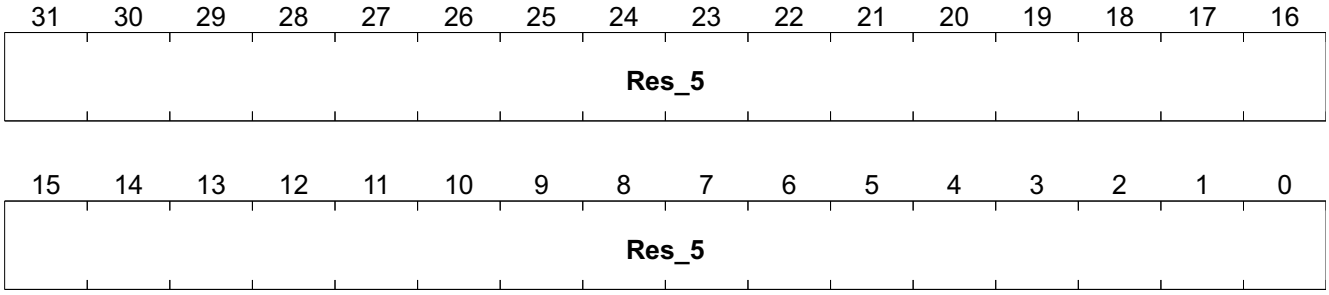
Res_4	Offset	Reset Value
Reserved 4	$F8_H$	0_H



Field	Bits	Type	Description
Res_4	31:0		**Reserved** Not Applicable.

Reserved 5

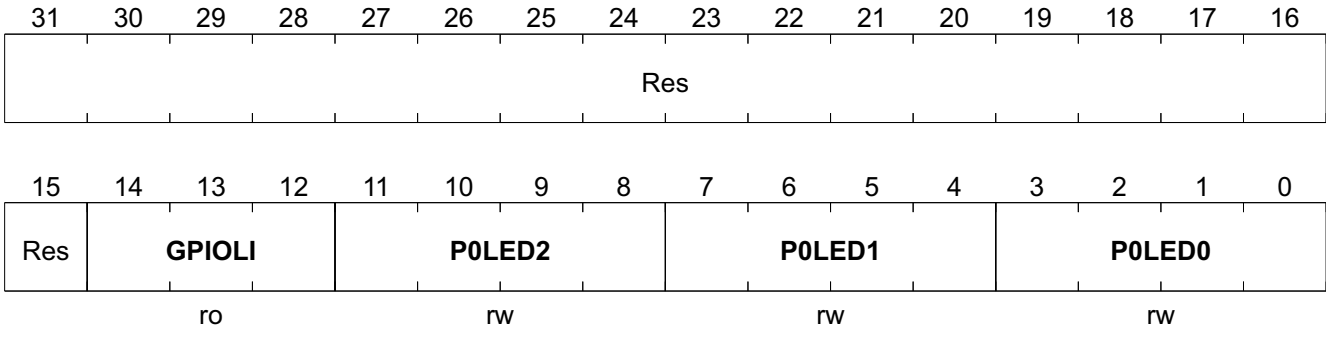
Res_5	Offset	Reset Value
Reserved 5	FC_H	0_H



Field	Bits	Type	Description
Res_5	31:0		**Reserved** Not Applicable.

Port 0 LED

port0_LED	Offset	Reset Value
Port 0 LED	100_H	$A59_H$



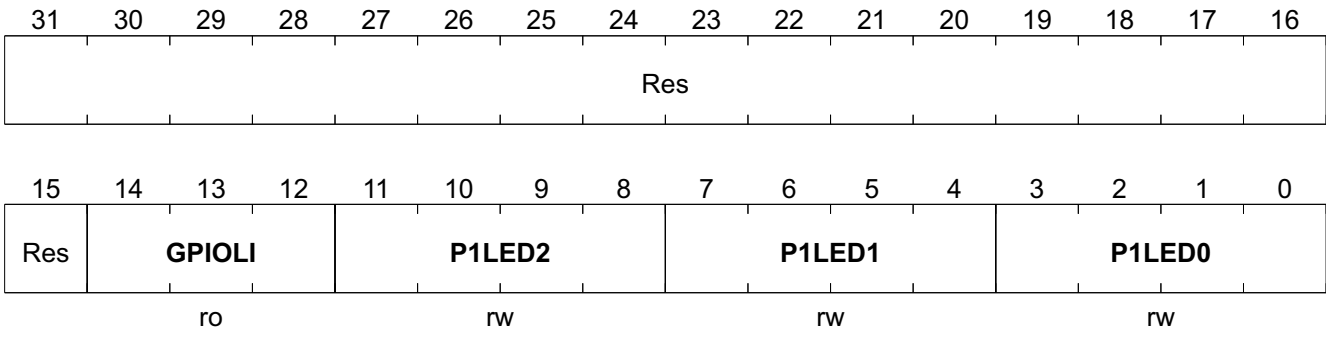
Field	Bits	Type	Description
Res	31:15		**Reserved** Not Applicable.
IN_P0LED2	14	ro	**Input in Port 0 LED2** Input value at pin Port 0 LED2 when it is configured to GPIO_in mode
IN_P0LED1	13	ro	**Input in Port 0 LED1** Input value at pin Port 0 LED1 when it is configured to GPIO_in mode
NI_P0LED0	12	ro	**Input in Port 0 LED0** Input value at pin Port 0 LED0 when it is configured to GPIO_in mode

Field	Bits	Type	Description
P0LED2	11:8	rw	**Port 0 LED2 State** 0000_B, GPIO_in. 0001_B, GPIO_output_flash. 0010_B, GPIO_output_1 0011_B, GPIO_output_0. 0100_B, Link (steady). 0101_B, Speed (steady) 0110_B, Duplex (steady). 0111_B, Activity (flash). 1000_B, Collision (flash) 1001_B, Link + activity. 1010_B, Default value, duplex/col 1011_B, 10M_link + activity 1100_B, 100M_link + activity. 1101_B, Reserved. 1110_B, Reserved. 1111_B, Reserved.
P0LED1	7:4		**Port 0 LED1 State** Refer the definition in **P0LED2** except the default value. 0101_B, Default value, speed
P0LED0	3:0		**Port 0 LED0 State** Refer the definition in **P0LED2** except the default value. 1001_B, Default value, Link/activity

Note: Port0 LED[2:0] pin (164,165,166) configuration register.

Port 1 LED

port1_LED	Offset	Reset Value
Port 1 LED	104_H	$A59_H$



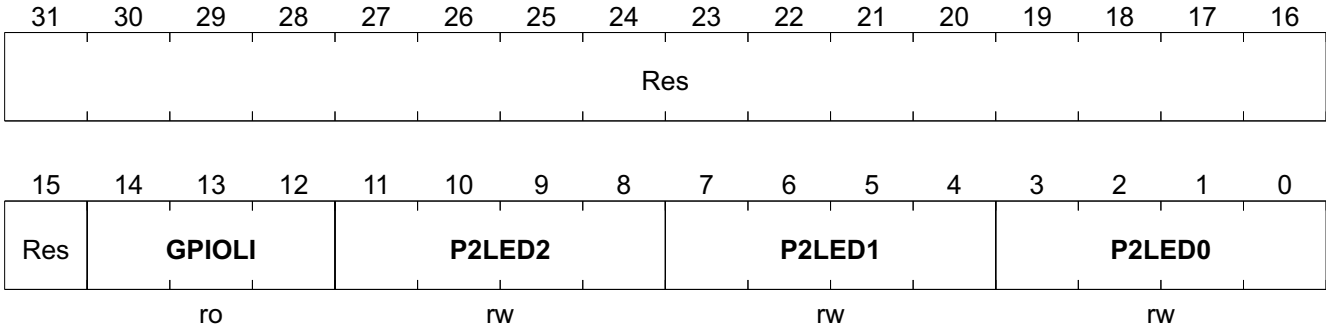
Field	Bits	Type	Description
Res	31:15		**Reserved** Not Applicable.
IN_P1LED2	14	ro	**Input in Port 1 LED2** Input value at pin Port 1 LED2 when it is configured to GPIO_in mode

Field	Bits	Type	Description
IN_P1LED1	13	ro	**Input in Port 1 LED1** Input value at pin Port 1 LED1 when it is configured to GPIO_in mode
IN_P1LED0	12	ro	**Input in Port 1 LED0** Input value at pin Port 1 LED0 when it is configured to GPIO_in mode
P1LED2	11:8	rw	**Port 1 LED2 State** Refer the definition in **P0LED2** except the default value. 1010_B, Default value, duplex/col
P1LED1	7:4		**Port 1 LED1 State** Refer the definition in **P0LED2** except the default value. 0101_B, Default value, speed
P1LED0	3:0		**Port 1 LED0 State** Refer the definition in **P0LED2** except the default value. 1001_B, Default value, link/activity

Note: Port1 LED[2:0] pin (161,162,163) configuration register.

Port 2 LED

port2_LED	Offset	Reset Value
Port 2 LED	108_H	$A59_H$

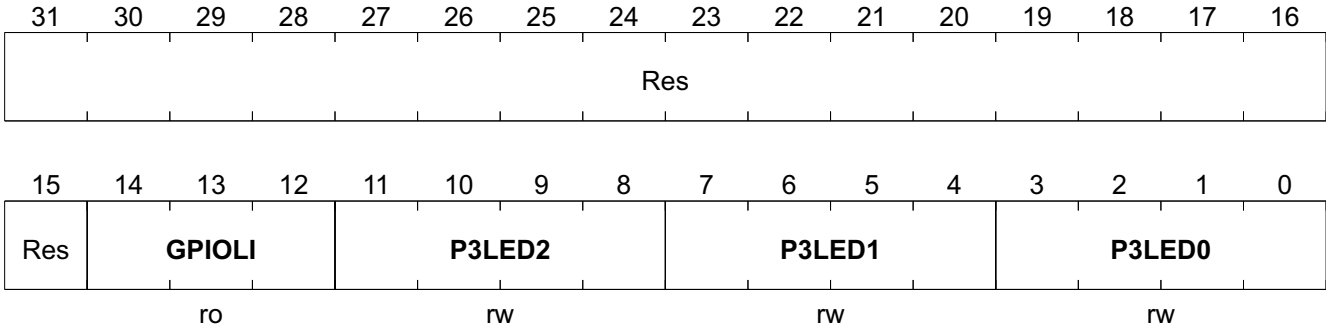


Field	Bits	Type	Description
Res	31:15		**Reserved** Not Applicable.
IN_P2LED2	14	ro	**Input in Port 2 LED2** Input value at pin Port 2 LED2 when it is configured to GPIO_in mode
IN_P2LED1	13	ro	**Input in Port 2 LED1** Input value at pin Port 2 LED1 when it is configured to GPIO_in mode
IN_P2LED0	12	ro	**Input in Port 2 LED0** Input value at pin Port 2 LED0 when it is configured to GPIO_in mode
P2LED2	11:8	rw	**Port 2 LED2 State** Refer the definition in **P0LED2** except the default value. 1010_B, Default value, duplex/col
P2LED1	7:4		**Port 2 LED1 State** Refer the definition in **P0LED2** except the default value. 0101_B, Default value, speed
P2LED0	3:0		**Port 2 LED0 State** Refer the definition in **P0LED2** except the default value. 1001_B, Default value, link/activity

Note: Port2 LED[2:0] pin (147,158,160) configuration register.

port3_LED

port3_LED	Offset	Reset Value
Port 3 LED	$10C_H$	$A59_H$

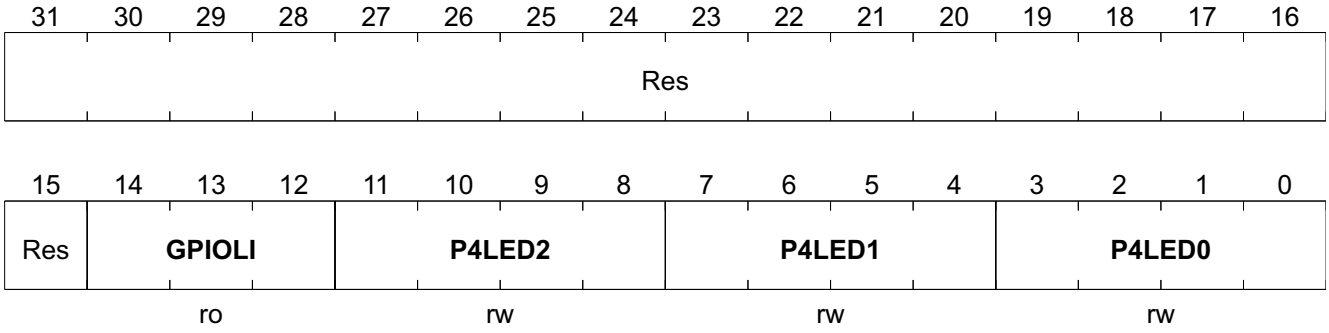


Field	Bits	Type	Description
Res	31:15		**Reserved** Not Applicable.
IN_P3LED2	14	ro	**Input in Port 3 LED2** Input value at pin Port 3 LED2 when it is configured to GPIO_in mode
IN_P3LED1	13	ro	**Input in Port 3 LED1** Input value at pin Port 3 LED1 when it is configured to GPIO_in mode
IN_P3LED0	12	ro	**Input in Port 3 LED0** Input value at pin Port 3 LED0 when it is configured to GPIO_in mode
P3LED2	11:8	rw	**Port 3 LED2 State** Refer the definition in **P0LED2** except the default value. 1010_B, Default value, duplex/col
P3LED1	7:4		**Port 3 LED1 State** Refer the definition in **P0LED2** except the default value. 0101_B, Default value, speed
P3LED0	3:0		**Port 3 LED0 State** Refer the definition in **P0LED2** except the default value. 1001_B, Default value, link/activity

Note: Port3 LED[2:0] pin (144,145,146) configuration register.

port4_LED

port4_LED	Offset	Reset Value
Port 4 LED	110_H	$A59_H$



Field	Bits	Type	Description
Res	31:15		**Reserved** Not Applicable.
IN_P4LED2	14	ro	**Input in Port 4 LED2** Input value at pin Port 4 LED2 when it is configured to GPIO_in mode
IN_P4LED1	13	ro	**Input in Port 4 LED1** Input value at pin Port 4 LED1 when it is configured to GPIO_in mode
IN_P4LED0	12	ro	**Input in Port 4 LED0** Input value at pin Port 4 LED0 when it is configured to GPIO_in mode
P4LED2	11:8	rw	**Port 4 LED2 State** Refer the definition in **P0LED2** except the default value. 1010_B, Default value, duplex/col
P4LED1	7:4		**Port 4 LED1 State** Refer the definition in **P0LED2** except the default value. 0101_B, Default value, speed
P4LED0	3:0		**Port 4 LED0 State** Refer the definition in **P0LED2** except the default value. 1001_B, Default value, link/activity

Note: Port4 LED[2:0] pin (141,142,143) configuration register.

7 UART

The UART description covers:

- Feature list (**Chapter 7.1**)
- Functional description (**Chapter 7.2**)
- External Interface; described in the dedicated chapter of the different interfaces
- Registers (**Chapter 7.3**)

7.1 Feature List

The UART offers the following features:

- Seperate 16 x 8 transmit and 16 x 12 receive FIFO to reduce CPU interrupts
- Programmable baud rate generator
- Standard asynchronous communication bits(start, stop and parity). These are added prior to transmission and removed on reception.
- Fully-programmable serial interface charateristic:
  - Data can be 5, 6, 7 or 8 bits
  - Even, odd, stick or no-parity bit generation and detection
  - 1 or 2 stop bit generation
  - Baud rate generation
- Programmable hardware flow control

7.2 Functional Description

The UART performs:

- Serial-to-parallel conversion on data received from a peripheral device
- Parallel-to-serial conversion on data transmitted to the peripheral device.

The CPU reads and writes data and control/status information through the AMBA APB interface. The transmit and receive paths are buffered with internal FIFO memories enabling up to 16-bytes to be stored independently in both transmit and receive modes.

The UART:

- Includes a programmable baud rate generator that generates a common transmit and receive internal clock from the UART internal reference clock input, UARTCLK = 62.5 MHz
- Offers similar functionality to industry-standard 16C550 UART device
- Supports baud rates up to 460.8 Kbits/s, subject to UARTCLK reference clock frequency.
- UART operation are controlled by the line control register(**UARTLCR_H**)
- The baud rate values are controlled by **UARTLCR_M** and **UARTLCR_L** registers
- Can generate individually maskable interrupts from the receive, transmit, modem status and error conditions
- Support moden status input signals CTS, DCD, DSR and RI
- Support modem output control lines RTS and DTR
- Uses the nUARTCTS input and nUARTRTS output to automatically control the serail data flow.

Figure 16 shows a block diagram of UART.

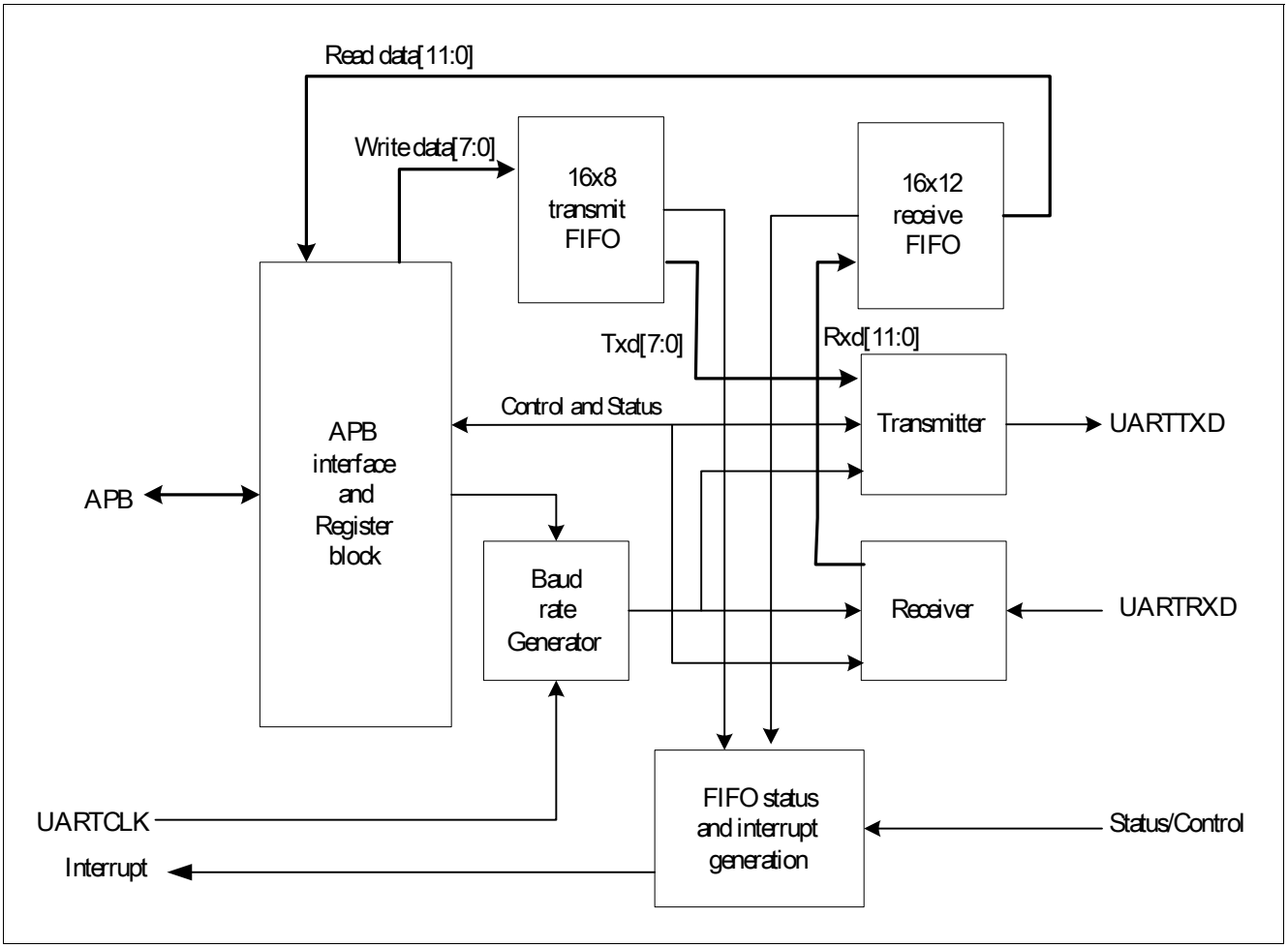


Figure 16 UART block diagram

7.2.1 AMBA APB Interface

The AMBA APB interface generates read and write decodes for accesses to status/control registers and transmit/receive FIFO memories.

7.2.2 Register Block

The register block stores data written, or to be read across the AMBA APB interface.

7.2.3 Baud Rate Generator

The baud rate generator contains free-running counters that generate the internal x16 clocks, Baud16. Baud16 provides timing information for UART transmit and receive control. Baud16 is a stream of pulses with a width of one UARTCLK clock period and a frequency of 16 times the baud rate.

7.2.4 Transmit FIFO

The transmit FIFO is an 8-bit wide, 16 location deep, FIFO memory buffer. CPU data written across the APB interface is stored in the FIFO until read out by the transmit logic. You can disable the transmit FIFO to act like a one-bye holding register.

7.2.5 Receive FIFO

The receive FIFO is a 12-bit wide, 16 location deep, FIFO memory buffer. Received data and corresponding error bits, are stored in the receive FIFO by the receive logic until read out by the CPU across the APB interface. The receive FIFO can be disabled to act like a one-byte holding register.

7.2.6 Transmit Logic

The transmit logic performs parallel-to-serial conversion on the data read from the transmit FIFO. COntrol logic outputs the serial bit stream beginning with a start bit, data bits with the LSB first, fellowed by the parity bit, and then the stop bits according to the programmed configuration in control registers.

7.2.7 Receive Logic

The receive logic performs serial-to-parallel conversion on the received bit stream after a valid start pulse has been detected. Overrun, parity, frame error checking, and line break detection are also performed, and their status accompanies the dtaa that is written to the receive FIFO.

7.3 Register Description

7.3.1 UART Registers Overview

There are two UART port, one base address is 0x1260 0000 and the other is 0x1280 0000.

Table 39 Address Space

Module	Base Address	End Address	Note
UART	1260 0000_H	$1F_H$	

Table 40 Registers Overview from UART Registers

Register Short Name	Register Long Name	Offset Address
UART_D	UART Data	00_H
UARTRRS_ECR	UART Receive Status Register/Error Clear	04_H
UARTLCR_H	UART Line Control Register, High Byte	08_H
UARTLCR_M	UART Line Control Register, Middle Byte	$0C_H$
UARTLCR_L	UART Line control Register, Low Byte	10_H
UARTCR	UART Control	14_H
UARTFR	UART Flag	18_H
UARTIIR_UARTICR	UART Interrupt Identification/Clear	$1C_H$

7.3.2 UART Registers Description

UART Data

UART_D	Offset	Reset Value
UART Data	00_H	0_H

7	6	5	4	3	2	1	0
UDR							
rw							

Field	Bits	Type	Description
UDR	7:0	rw	**Data Register** Receive (read) data character Transmit (write) data character

UART Receive Status Register/Error Clear

UARTRRS_ECR	Offset	Reset Value
UART Receive Status Register/Error Clear	04_H	0_H

7	6	5	4	3	2	1	0
RSR	Res			OE	BE	PE	FE
w	r			r	r	r	r

Field	Bits	Type	Description
RSR	7	w	**RSR** A write to this register clears the framing, parity, break and overrun errors. The data value is not important.
Res	6:4	r	**Reserved** Not applicable.
OE	3		**Overrun Error** This bit is set to 1 if data is received and the FIFO is already full.
BE	2		**Break Error** This bit is set to 1 if a break condition was detected, indicating that the received data input was held LOW for longer than a full-word transmission time.
PE	1		**Parity Error** When this bit is set to 1, it indicates that the parity of received data character does pot match the parity selected in UARTLCR_H (bit 2)
FE	0		**Framing Error** When this bit is set to 1, it indicates that the received character did not have a valid stop bit.

UART Line Control Register, High Byte

UARTLCR_H	Offset	Reset Value
UART Line Control Register, High Byte	08_H	0_H

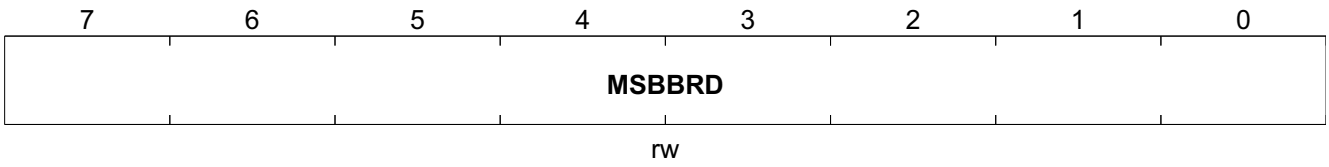
7	6	5	4	3	2	1	0
Res	WLEN		FEN	STP2	EPS	PEN	BRK
	rw		rw	rw	rw	rw	rw

Field	Bits	Type	Description
Res	7		**Reserved** Not applicable

Field	Bits	Type	Description
WLEN	6:5	rw	**Word Length [1:0]** The select bits indicate the number of data bits transmitted or received in a frame as follows:11 = 8 bits10 = 7 bits01 = 6 bits00 = 5 bits
FEN	4		**Enable FIFOs** If this bit is set to 1,transmit and receive FIFO buffers are enabled (FIFO mode). When cleared to 0 the FIFOs are disable (character mode) that is, the FIFOs become 1 byte-deep holding register.
STP2	3		**Two Stop Bits Select** If this bit is set to 1, two stop bits are transmitted at the end of the frame. The receive logic does not check for two stop bits being received.
EPS	2		**Even Parity Select** If this bit is set to 1, even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. When cleared to 0 then odd parity is performed which checks for an odd number of 1s.this bit has no effect when parity is disabled by parity enable being cleared to 0.
PEN	1		**Parity Enable** If this bit is set to 1, parity checking and generation is enabled, else parity is disabled and no parity bit assed to the data frame.
BRK	0		**Send Break** If this bit is set to 1, a low-level is continually output on the UARTTXD output, after completing transmission of the current character. This bit must be asserted for at least one complete frame transmission time in order to generate a break condition. The transmit FIFO contents remain unaffected during a break condition.For normal use, this bit must be cleared to 0.

UART Line Control Register, Middle Byte

UARTLCR_M	Offset	Reset Value
UART Line Control Register, Middle Byte	$0C_H$	0_H



Field	Bits	Type	Description
MSBBRD	7:0	rw	**Most Significant Byte of Baud Rate Divisor** These bits are cleared to 0 on reset.

UART Line Control Register, Low Byte

Note: The baud rate divisor is calculated as follow:
*Baud rat divisor BAUDDIV = (FUARTCLK/(16*Baud rate))-1*
Where FUARTCLK is the UART reference clock frequency
The below table show some typical bit rates and their corresponding divisor, given a UART clock frequency of 62.5 MHz. A divisor value of zero is illegal, and so no transmission or reception will occur.

UARTLCR_L	Offset	Reset Value
UART Line control Register, Low Byte	10_H	0_H

7	6	5	4	3	2	1	0
LSBBRD							
rw							

Field	Bits	Type	Description
LSBBRD	7:0	rw	**Least Significant Byte of Baud Rate Divisor** These bits are cleared to 0 on reset.

UART Control

UARTCR	Offset	Reset Value
UART Control	14_H	0_H

7	6	5	4	3	2	1	0
Res	RTIE	TIE	RIE	MSIE	Res		UARTEN
	rw	rw	rw	rw			rw

Field	Bits	Type	Description
Res	7		**Reserved** Not applicable.
RTIE	6	rw	**Receive Timeout Interrupt Enable** If this bit is set to 1, the receive timeout interrupt is enabled. The receive timeout interrupt is asserted when the receive FIFO is not empty and no further data is received over a 32-bit period. The receive timeout interrupt is cleared when the FIFO becomes empty through reading al the data.

Field	Bits	Type	Description
TIE	5	rw	**Transmit Interrupt Enable** If this bit is set to 1, the transmit interrupt is enabled. The transmit interrupt changes state when one of the following events occurs: 1.if the FIFOs are enabled and the transmit FIFO is at least half empty, then the transmit interrupt is asserted HIGH. It is cleared by filling the transmit FIFO to more than half full. 2.if the FIFOs are disabled and there is no data preset in the transmitters single location, the transmit FIFO is asserted HIGH. It is cleared by performing a single write to the transmitter FIFO.
RIE	4	rw	**Receive Interrupt Enable** If this bit is set to 1, the receive interrupt is enabled. Thereceive interrupt changes state when one of the following events occurs: 1.if the FIFOs are enabled and the receive FIFO is half or more full, then the receive interrupt is asserted HIGH. It is cleared by reading data from the receive FIFO until it becomes less than half full. 2.if the FIFOs are disabled and data is received thereby filling the location, the receive interrupt is asserted HIGH. The receive interrupt is cleared by performing a single read to the receive FIFO.
MSIE	3	rw	**Modem Status Interrupt Enable** If this bit is set to 1, the modem interrupt is enabled. The modem status interrupt is asserted if any of the modem status lines (CTS,DCD,DSR) change. Modem status change when one of the following events occurs: (1) 0 → 1 (2) 1 → 0
Res	2:1		**Reserved** Not applicable
UARTEN	0	rw	**UART Enable** If this bit set to 1, the UART is enabled.

UART Flag

UARTFR	Offset	Reset Value
UART Flag	18_H	0_H

7	6	5	4	3	2	1	0
TXFE	RXFF	TXFF	RXFE	BUSY	DCD	DSR	CTS
ro	ro	ro	ro	ro	ro	ro	ro

Field	Bits	Type	Description
TXFE	7	ro	**Transmit FIFO Empty** The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H registerIf the FIFO is disabled, this bit is set when the transmit holding register is empty. If the FIFO is enabled, the TXFE bit is set when the transmit FIFO is empty.
RXFF	6		**Receive FIFO Full** The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the receive holding register is full. If the FIFO is enabled, the RXFF bit is set when receive FIFO is full.
TXFF	5		**Transmit FIFO Full** The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when transmit holding register is full. If the FIFO is enabled, the RXFE bit is set when the transmit FIFO is full.
RXFE	4		**Receive FIFO Empty** The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bits is set when receive holding register is empty. If the FIFO is enabled, the RXFE bit is set when the receive FIFO is empty.
BUSY	3		**UART Busy** If this bit is set to 1, the UART is busy transmitting data. This bit remains set until the complete byte, including all the stop bits, has been sent from he shift register. This bit is set as soon as the transmit FIFO becomes non-empty.
DCD	2		**Data Carrier Detect** This bit is the complement of the UART data carrier detect (nUARTDCD) modem status input. That is , the bit is 1 when the modem status input is 0.
DSR	1		**Data Set Ready** This bit is the complement of the UART data set ready (nUARTDSR) modem status input. That is , the bit is 1 when the modem status input is 0.
CTS	0		**Clear to Send** This bit is the complement of the UART clear to send (nUARTCTS) modem status input. That is , the bit is 1 when the modem status input is 0.

UART Interrupt Identification/Clear

UARTIIR_UARTICR	Offset	Reset Value
UART Interrupt Identification/Clear	$1C_H$	0_H

7	6	5	4	3	2	1	0
UICR	Res			RTIS	TIS	RIS	MIS
w				r	r	r	r

Field	Bits	Type	Description
UICR	7	w	**Interrupt Clear** A write to this register clears the modem status interrupt, regardless of the value written.
Res	6:4		**Reserved** Not applicable.
RTIS	3	r	**Receive Timout Interrupt** This bit is set to 1 if the UARTRTINTR receive timeout interrupt is asserted.
TIS	2		**Transmit Interrupt** This bit is set to 1 if the UARTRTINTR transmit interrupt is asserted.
RIS	1		**Receive Interrupt** This bit is set to 1 if the UARTRTINTR receive interrupt is asserted.
MIS	0		**Modem Interrupt Status** This bit is set to 1 if the UARTRTINTR modem status interrupt is asserted.

8 USB 1.1 Host Controller

The USB 1.1 host controller provides a USB host solution that can communicate with full speed and low speed devices.

The USB 1.1 host controller covers:

- Feature list (**Chapter 8.1**)
- Functional description (**Chapter 8.2**)
- DMA operation **Chapter 8.3**
- Registers (**Chapter 8.5**)

8.1 Feature List

- 32 bit high performance AMBA AHB bus interface
- Little/Big endian byte ordering
- 32-bit Tx/Rx buffer management architecture
- Supports full speed (12Mbps) and low speed (1.5Mbps)
- Supports embedded DPLL to operate from 48 MHz crystal or oscillator
- Supports automatic generation of SOF and CRC5/16
- Supports DMA mode for USB Control, Interrupt, Bulk, and Isochronous packets
- Supports descriptor chain architecture for effective packet scheduling
- Support two device ports

8.2 USB 1.1 Function Description

8.2.1 Block Diagram

The following block diagram describes the functional blocks of the Infineon USB 1.1 Host controller.

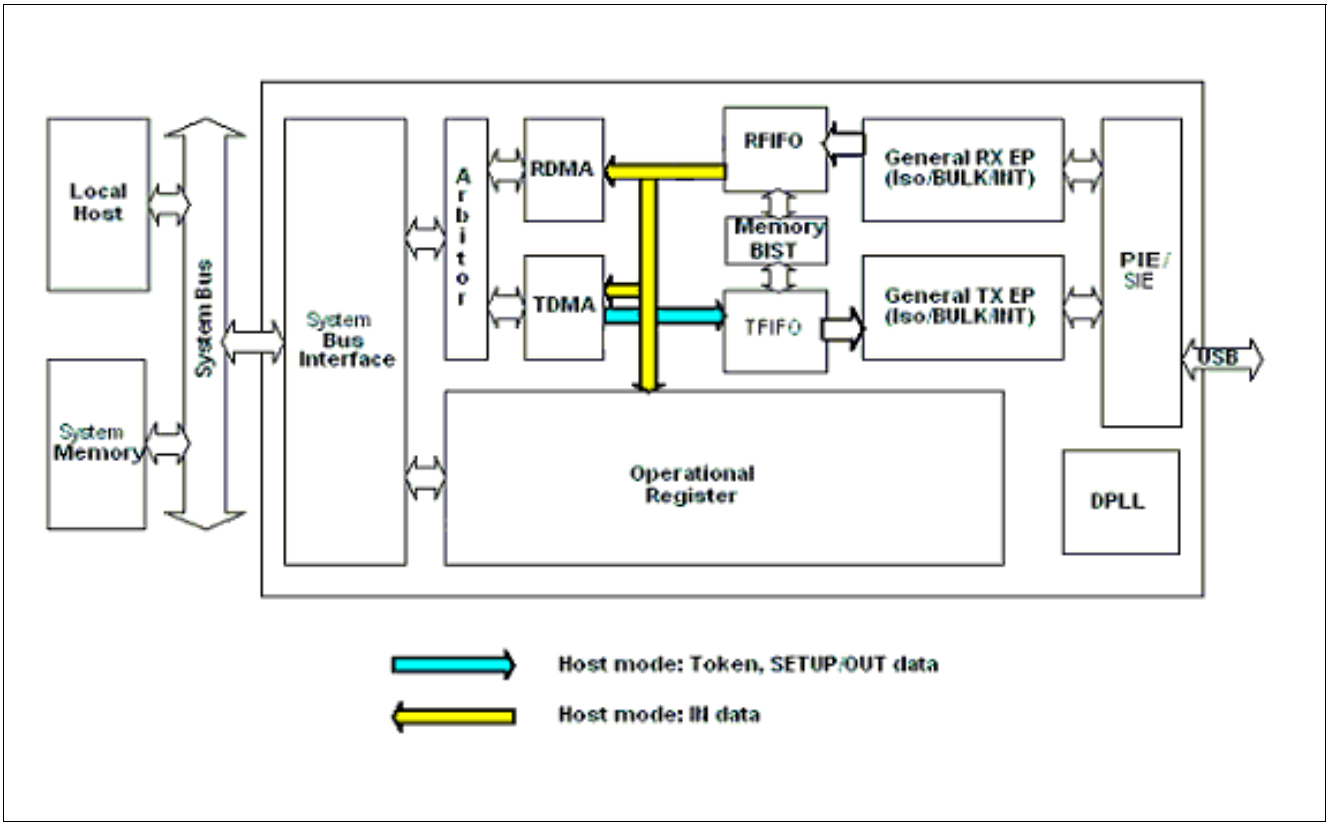


Figure 17 Block Diagram of Infineon USB 1.1 Host Controller

8.2.2 System Bus Interface

This block provides the USB Host controller with the connection to the AHB bus interface. The AHB bus is a 32-bit wide data bus, high-performance pipeline architecture. This block contains the AHB master interface and slave interface. The Host can program the USB Host controller operational register via the AHB slave interface. The DMA units within the USB Host controller will act as bus masters and access the system memory through the AHB master interface.

8.2.3 Operational Register

This block is the CSR (configure and status register) of USB 1.1 Host controller. The local host configures USB 1.1 Host controller via these registers. It includes DMA, endpoint, enable/disable, and interrupt control. The local host gets the status of the USB 1.1 Host controller by reading the registers. It includes the DMA, interrupt and USB bus status. The operational register also provides the interface for the local host to transfer the data for control and interrupt endpoint.

8.2.4 SIE

The SIE handles the link layer protocol of USB. It includes the following items

- Identify the USB SYNC field
- Identify the USB address, endpoint field
- Decode/encode the NRZI
- Generate/check the Bit Stuffing and the CRC
- Convert the USB incoming serial data to 8-bit parallel data
- Convert 8-bit parallel data to USB serial data
- Detect/report/generate USB bus events such as Reset, Suspend and Resume

8.2.5 DPLL

The DPLL block is a digital phase lock loop for extracting clock and data from the USB bus.

8.2.6 Memory BIST

The Memory BIST block is used for testing TFIFO and RFIFO. In this memory BIST, the MARCH C- test algorithm is adopted and the test data bus is 32 bits in width for each FIFO block. During the test period, all FIFO blocks are tested concurrently and the test procedure will be aborted if any fault is detected.

8.3 DMA Operation

4 kinds of Endpoints data transfer (Control, Interrupt, Bulk and Isochronous) are supported in host mode.

8.3.1 Endpoint Descriptor Format

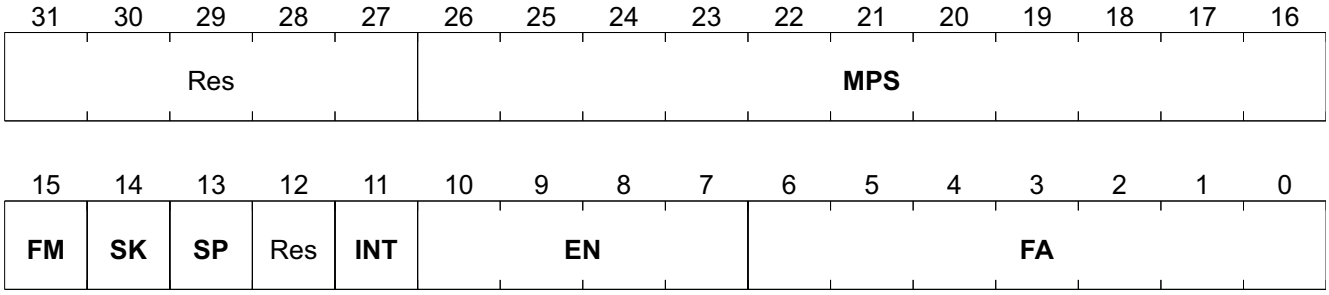
	31	0
DWORD 0	Control	
DWORD 1	Tail Transfer Descriptor	
DWORD 2	Head Transfer Descriptor	
DWORD 3	Next Endpoint Transfer Descriptor	

Table 41 Registers Overview

Register Short Name	Register Long Name	Offset Address
Contr_Buf_L	Controller/Buffer Length	H
Control_Reg	Control Register	H
Data_Buf_P	Data Buffer Pointer	H
Head_Trans_Des	Head Transfer Descriptor	H
Next_Endp_Trans_Des	Next Endpoint Transfer Descriptor	H
Next_Tra_Des_P	Next Transfer Descriptor Pointer	H
Status_Reg	Status Register	H
Tail_Trans_Des	Tail Transfer Descriptor	H

Control Register

Control_Reg	Offset	Reset Value
Control Register	H	0_H

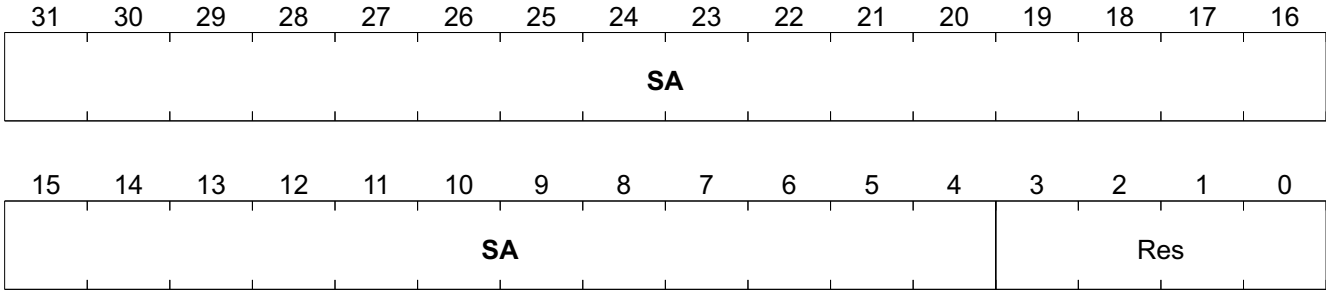


Field	Bits	Type	Description
Res	31:27		**Reserved**
MPS	26:16		**Maximum Packet Size** The maximum data that can transmit/receive in one USB transaction.
FM	15		**Format** This bit indicates that this packet is for isochronous. 0_B , The data in this descriptor is for general data transfer 1_B , The data in this descriptor is for isochronous transfer
SK	14		**Skip** When this bit is set, DMA will continue on the next descriptor in the link list, this is used for isochronous and periodic data transmission/reception.
SP	13		**Speed** This bit indicates that the speed of the data transfer.
Res	12		**Reserved**
INT	11		**Interrupt** This bit indicate this ED is an interrupt endpoint.
EN	10:7		**Endpoint Number** Endpoint number of the current USB function.
FA	6:0		**Function Address** Function address of the current USB device.

Tail Transfer Descriptor

Tail_Trans_Des **Offset** **Reset Value**

Tail Transfer Descriptor **H** **0_H**

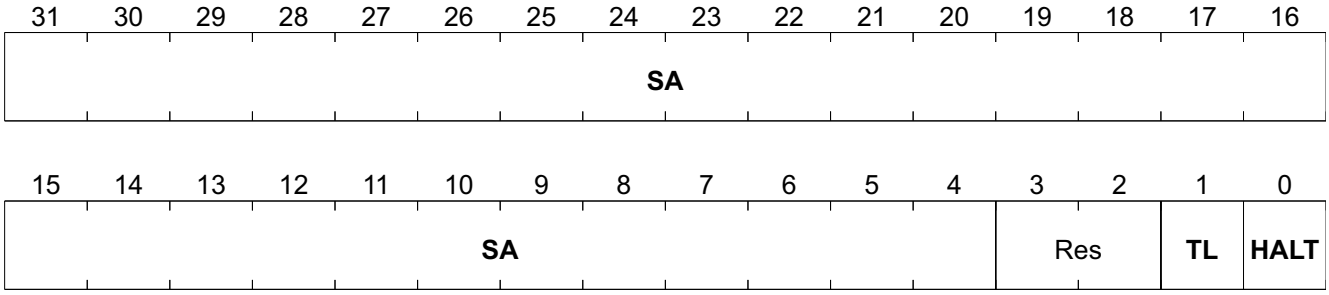


Field	Bits	Type	Description
SA	31:4		**Starting Address** Starting address of the tail transfer descriptor in host memory space.
Res	3:0		**Reserved**

Head Transfer Descriptor

Head_Trans_Des **Offset** **Reset Value**

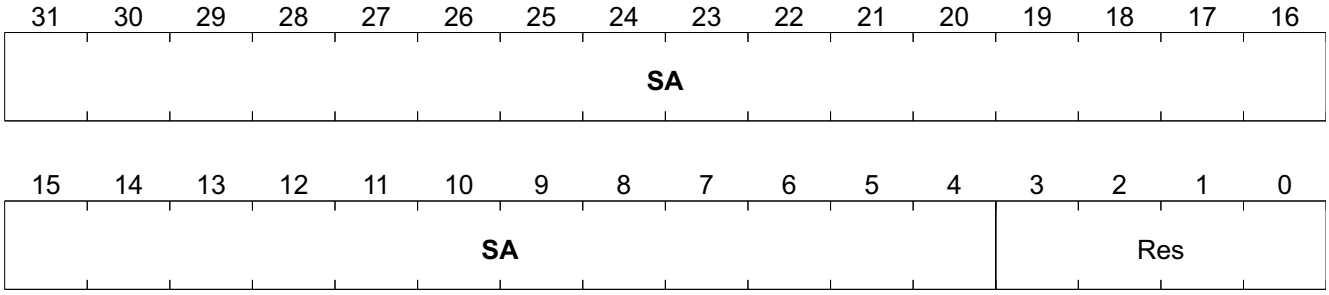
Head Transfer Descriptor **H** **0_H**



Field	Bits	Type	Description
SA	31:4		**Starting Address** Starting address of the head transfer descriptor in host memory space.
Res	3:2		**Reserved**
TL	1		**Togglecarry** This bit indicate the current toggle value in this transaction.
HALT	0		**Halt** This bit indicates that this endpoint is halted due to error.

Next Endpoint Transfer Descriptor

Next_Endp_Trans_Des	Offset	Reset Value
Next Endpoint Transfer Descriptor	$_H$	0_H



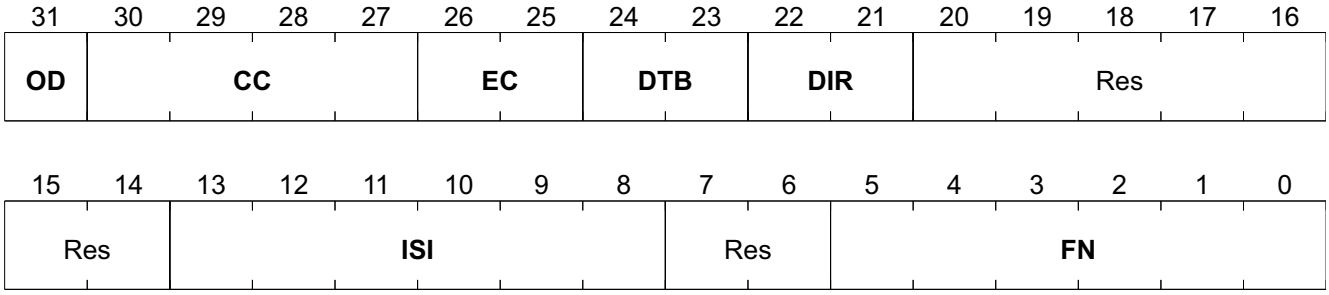
Field	Bits	Type	Description
SA	31:4		**Starting Address** Starting address of the next endpoint transfer descriptor in host memory space.
Res	3:0		**Reserved**

8.3.2 Transfer Descriptor Format

	31	0
DWORD 0	Status	
DWORD 1	Data Buffer Pointer	
DWORD 2	Controller/Buffer Length	
DWORD 3	Next Transfer Descriptor Pointer	

Status Register

Status_Reg	Offset	Reset Value
Status Register	$_H$	0_H



Field	Bits	Type	Description
OD	31		**Owner Descriptor** Descriptor ownership bit – set to 0 when the host owns the descriptor, set to 1 by host to tell the USB Host controller owns the descriptor, the controller will clear this bit when reception has done.
CC	30:27		**Complete Code** The transfer status of each USB transfer. 0000_B , No Error 0001_B , CRC Check Error 0010_B , Bit-Stuffing Error 0011_B , Data Toggle Error 0100_B , STALL 0101_B , Device No Response (Timeout) 0110_B , PID Error (Invalid PID) 0111_B , Unexpected PID 1000_B , Data Overrun (Packet Overrun) 1001_B , Data Underrun (Packet Underrrun) 1100_B , Buffer Overrun 1101_B , Buffer Underrrun
EC	26:25		**Error Count** Error count thet error that happens at each of USB transfer.
DTB	24:23		**Data Toggle Bit** This field is used for data PID value. When 1, use but 23 as the toggle bit. 24_B , When 0, use togglecarry bit in ED as the PID 23_B , Toggle value
DIR	22:21		**Direction** These bits indicate this packet's direction. 00_B , Setup packet 01_B , Out packet 10_B , IN packet 11_B **Res**, Reserved
Res	20:14		**Reserved**

Field	Bits	Type	Description
ISI	13:8		**Interrrupt Service Interval** This field indicate the frame interval that the interrupt transaction occurs. The frame interval = bit [13:8] + 1
Res	7:6		**Reserved**
FN	5:0		**Frame Number** This field indicate the frame number that receive/transmit this data, this field is only valid when configurated in Isochronous and interrupt transaction. For Isochronous transaction, it indicates the frame number in which the isochronous transaction should occur. For interrupt transaction, software use this field to indicate to hardware for the “starting frame number” of the interrupt transaction, hardware will update this field to the “next frame number” after the current transaction is done.

Data Buffer Pointer

Data_Buf_P **Offset** **Reset Value**

Data Buffer Pointer $_H$ **0_H**

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

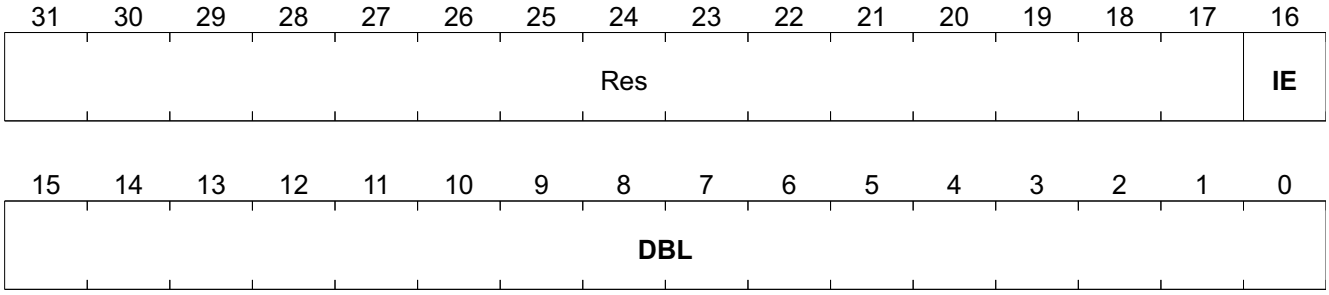
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADB															

Field	Bits	Type	Description
SADB	31:0		**Starting Address of the Data Buffer** This field indicates the starting address of the data buffer. Data buffer may be aligned on any byte. When an OUT or SETUP packet has been transmitted, this field will be updated as the next start address of the data buffer.

Controller/Buffer Length

Contr_Buf_L — **Offset** — **Reset Value**

Controller/Buffer Length — $_H$ — 0_H

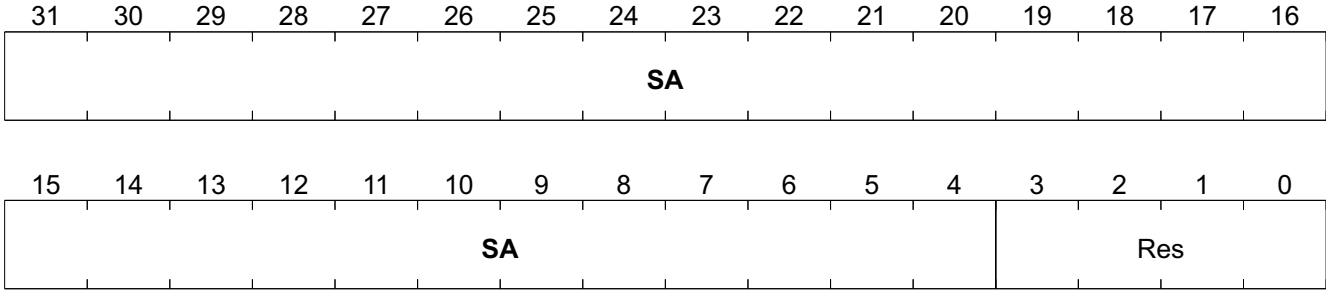


Field	Bits	Type	Description
Res	31:17		**Reserved**
IE	16		**Interrupt Enable** This field indicates that whether the interrupt will be asserted when this descriptor is completed.
DBL	15:0		**Length of Data Buffer** This field indicates the length of the current data buffer.

Next Transfer Descriptor Pointer

Next_Tra_Des_P — **Offset** — **Reset Value**

Next Transfer Descriptor Pointer — $_H$ — 0_H



Field	Bits	Type	Description
SA	31:4		**Starting Address** Starting address of the next descriptor in host memory space.
Res	3:0		**Reserved**

8.3.3 DMA Operation

To provide a high-performance and effective way for software packet scheduling, the DMA is able to handle both transmit and receive packets in a single descriptor chain. In the endpoint and transfer descriptors, the software can specify the descriptor format (Isochronous or none-Isochronous), direction, speed, and data toggle bit.

If there is any isochronous or periodic data that needs to be transmitted/received, this descriptor needs to link at the beginning of the link list to guarantee the bus bandwidth. After these descriptors, the control or bulk descriptor is linked.

DMA starts to access the first descriptor in the link list and transmit/receive the data through the USB bus. Since there might be several USB packets segmented in one descriptor. After one USB packet is transmitted, DMA will update the **transmit status**,**data length**,**start address of the data buffer**, and **length of data buffer** for further access, and advance to the next descriptor.

When the DMA finishes it's transmit/receive of a descriptor, it depends on the setting of the **interrupt enable** to generate HC_INT to indicate this descriptor is ready for the software driver process.

If all the descriptors have been accessed once, and the frame is not yet over, then the DMA will try to access the general descriptor again in order to get a high performance.

If a USB zero length packet is received, then the received data length will be zero, and this buffer is retired due to short packet received.

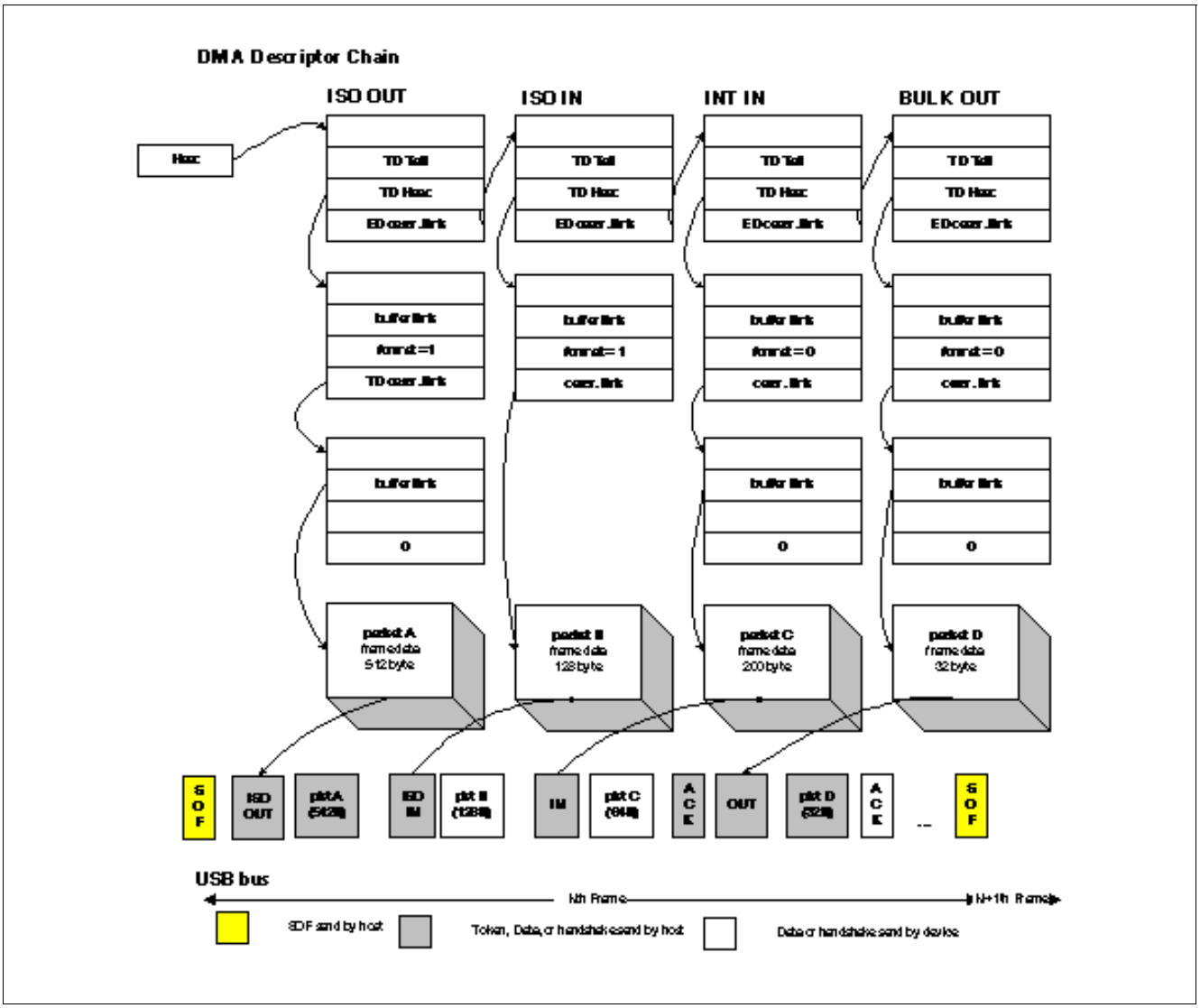


Figure 18 DMA Operation in Host Mode

For Interrupt IN/OUT transactions, each ED just contains one valid TD. Since the Interrupt transaction is periodic, two parameters are defined in TD by software to guide hardware for doing Interrupt transfer, they are Frame Number and Interrupt service period. The Frame Number is used by the software to indicate the frame number of which the first Interrupt transaction occurs, of this TD. The Interrupt service period indicates the frame interval between the current Interrupt transaction and the next one, Frame_Interval is calculated by the following formula:

Frame_Internal = (Interrupt service period + 1)

The transfer of interrupt transaction is activated just when the current frame number matches the Frame Number of the TD, after the current transaction is served, the next frame number will be updated to TD descriptor by hardware, then the hardware waits for the next matched frame number to serve the Interrupt transaction, and so on. The next Frame Number is calculated by (current_Frame_Number) + Frame_Interval. The following diagram describes how Frame_Number and Interrupt_service_period work.

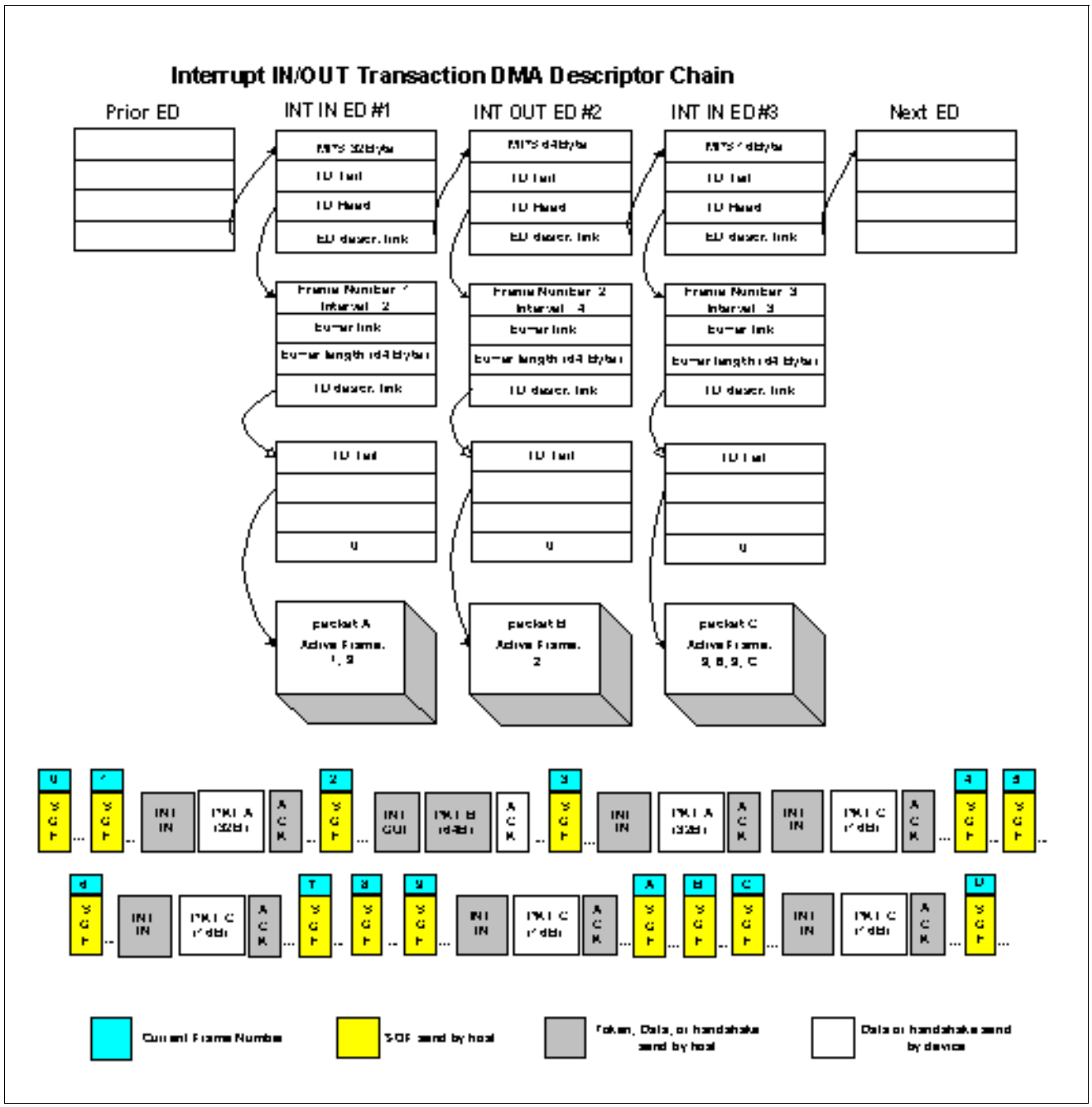


Figure 19 Interrupt IN/OUT Transactions

8.4 USB Control Status Register Map

Table 42 Address Space

Module	Base Address	End Address	Note
USB Control Status	1120 0000_H	1120 0080_H	Xxxxx

Table 43 Registers Overview from USB Control Status Register

Register Short Name	Register Long Name	Offset Address
...	...	...$_H$
GC	General Control	00_H
INT_S	Interrupt Status	04_H
INT_E	Interrupt Enable	08_H
Res_6	Reserved 6	$0C_H$
H_Gen_Cntl	Host General Control	10_H
Res_7	Reserved 7	14_H
SOF_FI	SOF Frame Interval	18_H
SOF_FN	SOF Frame Number	$1C_H$
RR0	Reserved 0	20_H
RR_1	Reserved Register 1	21_H
RR_2	Reserved Register 2	22_H
RR_76	Reserved Register 76	$6C_H$
Low_STh	Low Speed Threshold	70_H
RH_D	RH Descriptor	74_H
PX_St	Port X Status	78_H
HDHS_Ad	Host Descriptor Head Starting Address	80_H

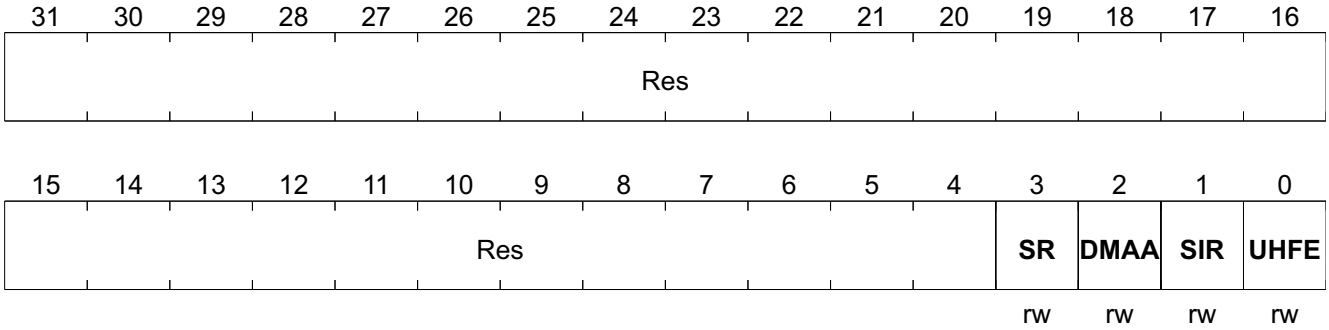
8.5 USB Control Status Registers Description

Host processors can only access USB 1.1 host/device controller registers with double word (32 bits) reads or writes on double word boundaries.

8.5.1 Registers Description

General Control

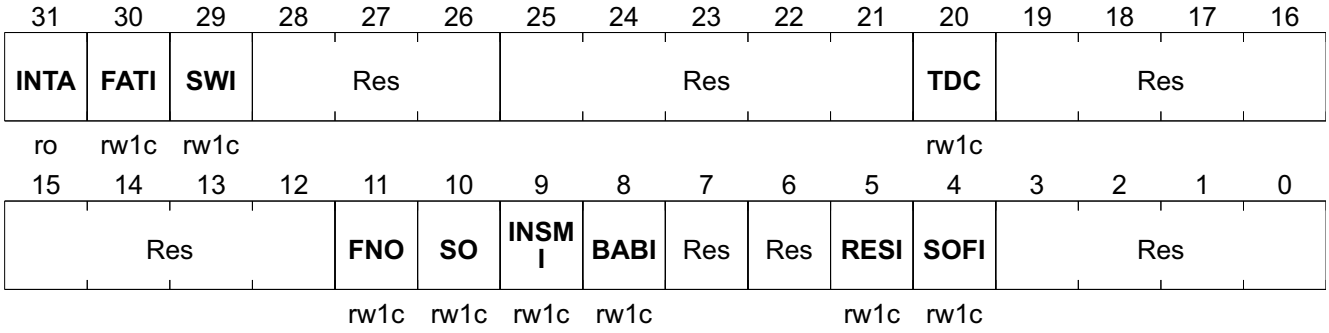
GC	Offset	Reset Value
General Control	**00_H**	**0_H**



Field	Bits	Type	Description
Res	31:4		**Reserved** Not Applicable.
SR	3	rw	**Software Reset, Both Modes** Setting this bit resets the device controller to its initial state. This bit is auto-cleared after reset. Writing a 0 to this bit takes no effect.
DMAA	2		**DMA Arbitration Control, Both Modes** 0_B , Receive = Transmit (1:1) 1_B , Receive > Transmit
SIR	1		**Software Interrupt Request, Both Modes** When this bit is set to 1, the controller's interrupt pin become active. Reading this bit always returns zero.When SW_INT in interrupt is clear, this bit is clear as well.
UHFE	0		**USB Host Function Enable, Both Modes** This bit enables the USB host functions, when 1'b1, the controller acts as USB host.

Interrupt Status

INT_S	Offset	Reset Value
Interrupt Status	04_H	0_H



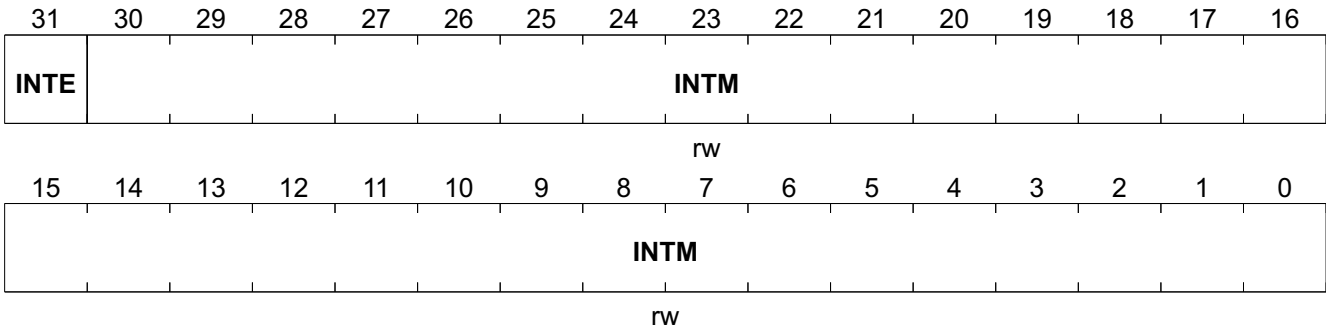
Field	Bits	Type	Description
INTA	31	ro	**Interrupt Active** When this bit is set, it indicates that at least one unmasked interrupt status is set.
FATI	30	rw1c	**Fatal Interrupt, Device Mode** Reserved. Host mode: 1_B , Fatal system bus error occurs
SWI	29		**Software Interrupt, Both Modes** 1_B , Software Interrupt. This bit is set when software set one to SW_INT_REQ 00_H, and is cleared after software writes one to this bit.
Res	28:26		**Reserved** Not Applicable
Res	25:21		**Reserved** Not Applicable
TDC	20	rw1c	**A TD is Completed**
Res	19:12		**Reserved** Not Applicable
FNO	11	rw1c	**Frame Number Overflow** This bit is set when the MSB of the frame number changes.
SO	10		**Scheduling Overrun** This bit is set when USB schedules for current frame overruns.
INSMI	9		**Root Hub Status Change** 1_B , Detected device insertion or remove. This bit will only be set for the device or hub, which is attached to host directly.
BABI	8		**Babble Detected, Host Mode** 1_B , Detected babble
Res	7		**Reserved** Not Applicable

Field	Bits	Type	Description
Res	6		**Reserved** Not Applicable
RESI	5	rw1c	**Resume Detected** 1_B , USB resume event is detected. Controller set this bit to one when resume signal is detected on USB bus.
SOFI	4		**SOF Transmitted/Received, Host Mode** 1_B , Issue a SOF token. The frame number value is stored in $1C_H$ Frame Number
Res	3:0		**Reserved** Not Applicable

Interrupt Enable

INT_E **Offset** **Reset Value**

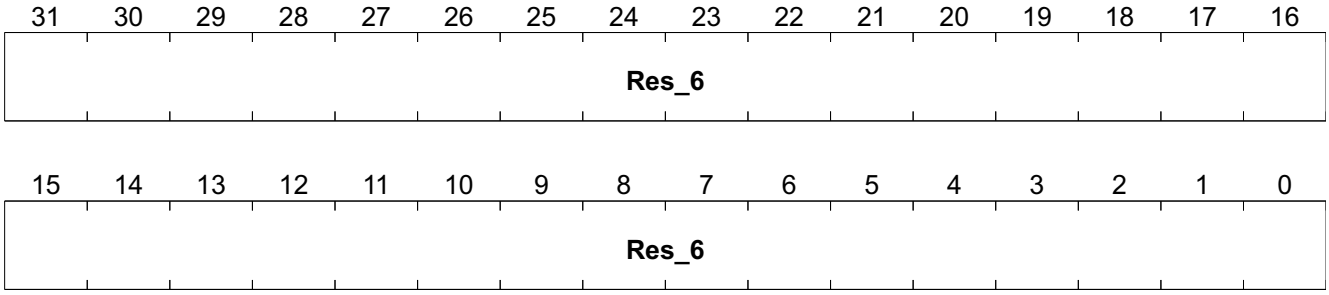
Interrupt Enable **08_H** **0_H**



Field	Bits	Type	Description
INTE	31		**Interrupt Enable** 0_B , Disable the controller to assert interrupt 1_B , Enable the controller to assert interrupt
INTM	30:0	rw	**Interrupt Mask** Bits are set to allow the corresponding interrupts (bit 21:0 in Interrupt Status register) to generate an interrupt request. And cleared to prevent the interrupt from happening.

Reserved 6

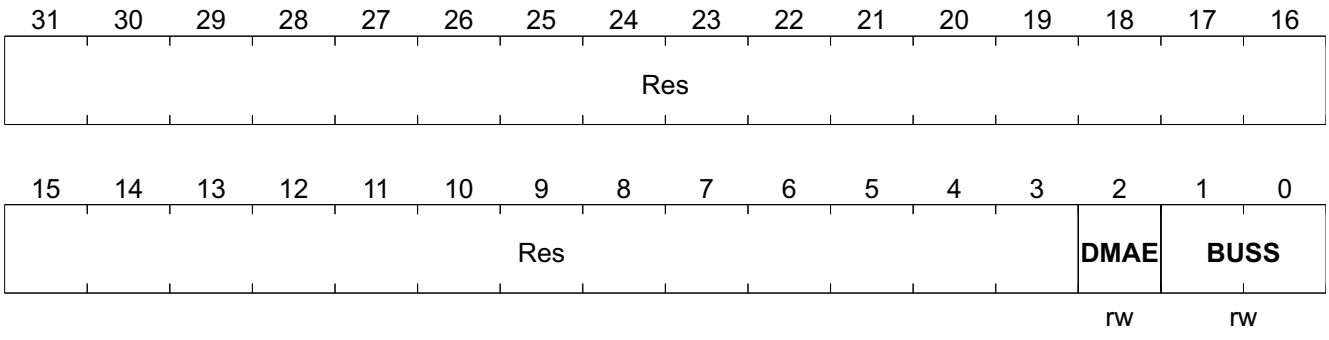
Res_6	Offset	Reset Value
Reserved 6	$0C_H$	0_H



Field	Bits	Type	Description
Res_6	31:0		**Reserved** Not Applicable.

Host General Control

H_Gen_Cntl	Offset	Reset Value
Host General Control	10_H	0_H

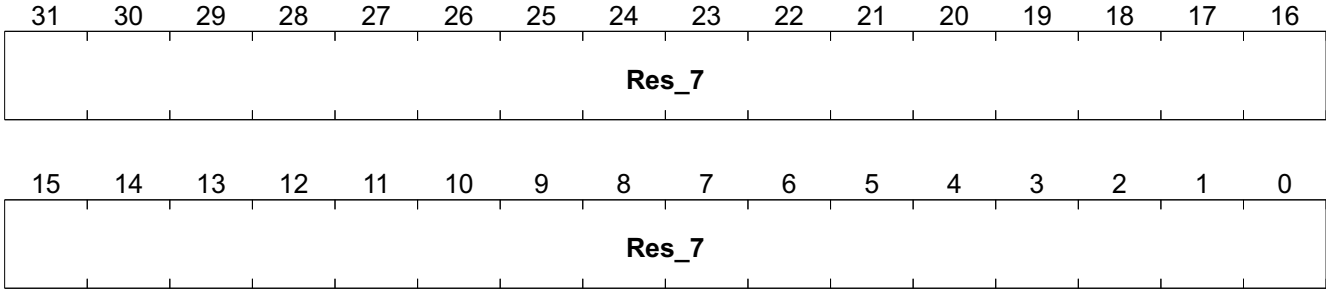


Field	Bits	Type	Description
Res	31:3		**Reserved** Not Applicable.

Field	Bits	Type	Description
DMAE	2	rw	**USB Host DMA Enable** This bit enables the host controller DMA functionality. When enable the DMA will start to fetch the descriptor for processing.
BUSS	1:0		**USB Bus State** A transition to USB operational state will cause the SOF generation to start 1 ms later. 00_B , USB reset state. 01_B , USB resume state 10_B , USB operational state 11_B , USB suspend state.

Reserved 7

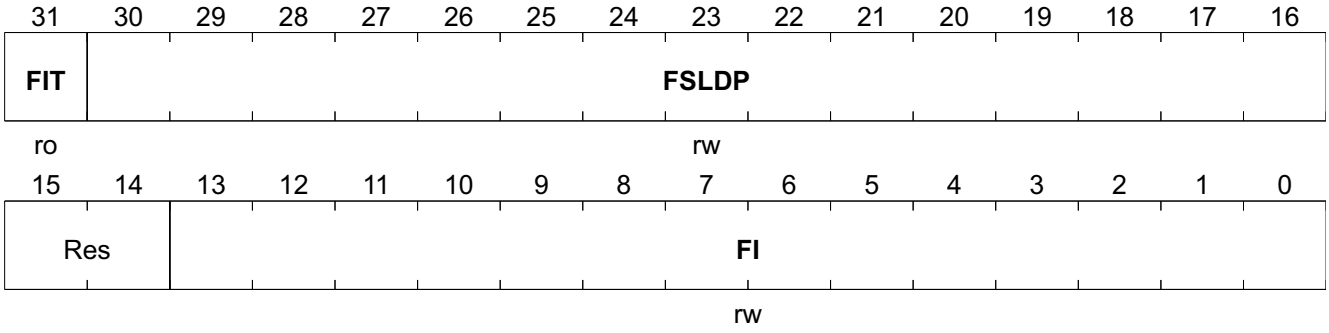
Res_7	Offset	Reset Value
Reserved 7	**14_H**	**0_H**



Field	Bits	Type	Description
Res_7	31:0		**Reserved** Not Applicable

SOF Frame Interval

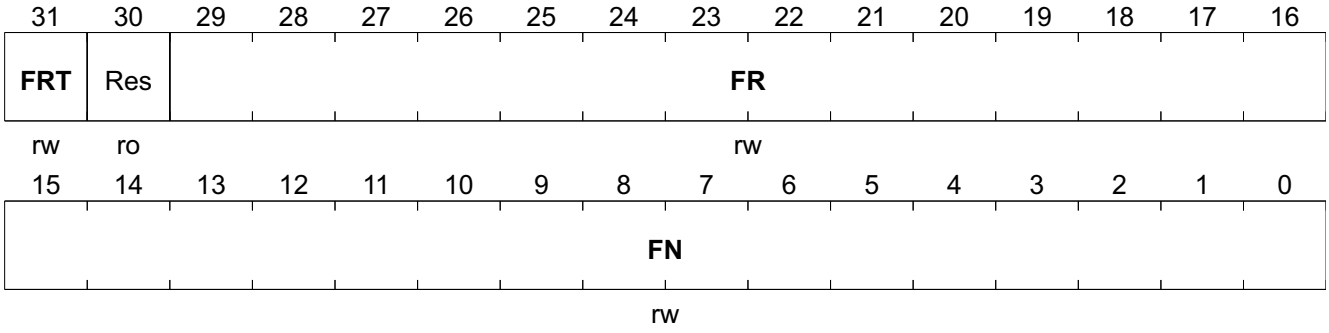
SOF_FI	Offset	Reset Value
SOF Frame Interval	18_H	$2EDF_H$



Field	Bits	Type	Description
FIT	31	ro	**Frame Interval Toggle** Software toggles this bit whenever it loads a new value to FM_INTERVAL.
FSLDP	30:16	rw	**FS Largest Data Packet** This field specifies a value which is loaded into the Largest Data Packet Counter at the beginning of each frame. The counter value represents the largest amount of data in bits which can be sent or received by the HC in a single transaction at any given time without causing scheduling overrun. The field value is calculated by the software.
Res	15:14		**Reserved** Not Applicable
FI	13:0	rw	**Frame Interval** This specifies the interval between two consecutive SOFs in bit times. The nominal value is set to be 11,999. Software should store the current value of this field before resetting HC.

SOF Frame Number

SOF_FN	Offset	Reset Value
SOF Frame Number	$1C_H$	0_H

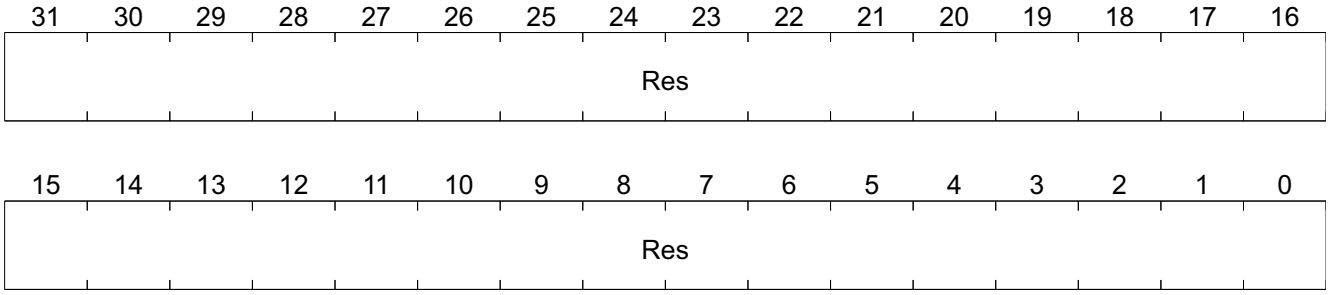


Field	Bits	Type	Description
FRT	31	rw	**Frame Remaining Toggle** This bit is loaded from the FM_INTERVAL_TOG field of FM_INTERVAL whenever FM_REMAIN remaining reaches 0. This bit is used by software for the synchronization between FM_INTERVALand FM_REMAIN.
Res	30	ro	**Reserved**
FR	29:16	rw	**Frame Remaining** This counter is decremented at each bit time. When it reaches zero, it is reset by loading the frame Interval value specified in FM_INTERVAL at the next bit time boundary. When entering theUSBOPERATIONAL state, HC re-loads the content with the FM_INTERVAL of and uses the updated value from the next SOF.
FN	15:0		**Frame Number** This field is a 16-bit counter. It provides a timing reference among events happening in the Host Controller and the Host Controller Driver. The Host Controller Driver may use the 16-bit value specified in this register and generate a 32-bit frame number without requiring frequent access to the register.

Reserved 0

RR0 **Offset** **Reset Value**

Reserved 0 **20_H** **0_H**



Field	Bits	Type	Description
Res	31:0		**Reserved** Not Applicable.

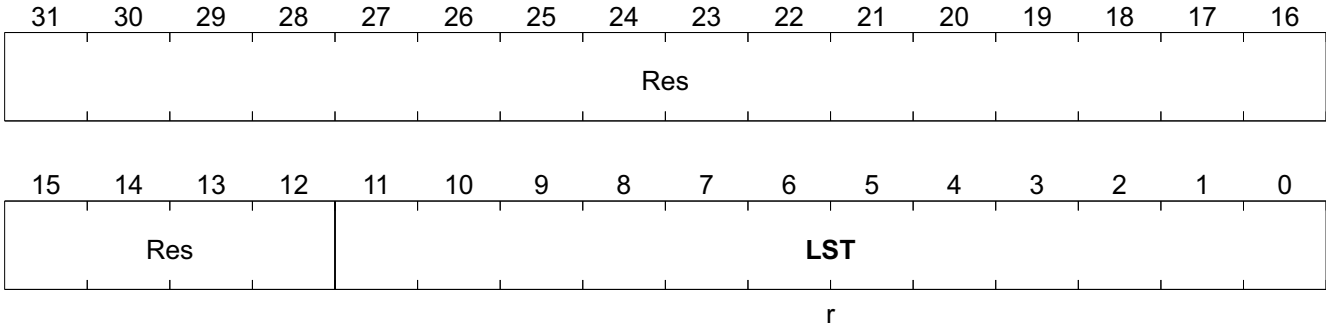
Other Reserved Registers have the same structure and characteristics as **Reserved 0**; the names and offset addresses are listed in **Table 44**.

Table 44 Reserved Registers

Register Short Name	Register Long Name	Offset Address
RR_1	Reserved Register 1	21_H
RR_2	Reserved Register 2	22_H
...	...	...$_H$
RR_76	Reserved Register 76	$6C_H$

Low Speed Threshold

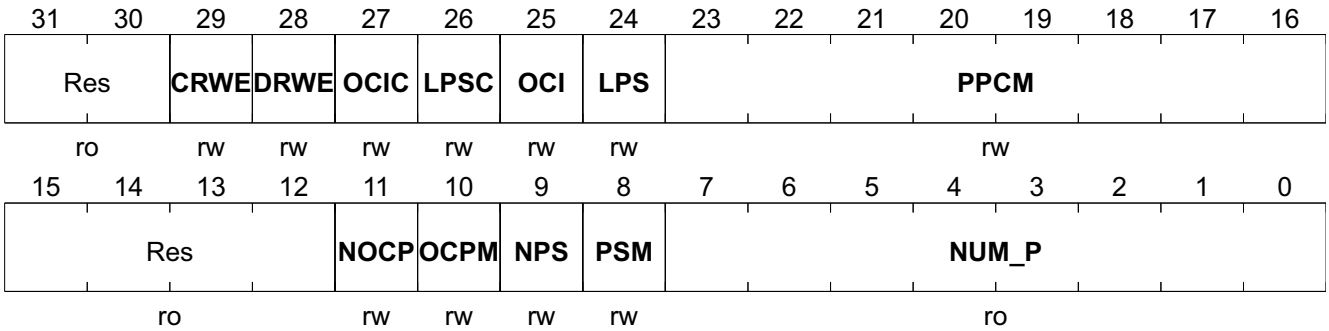
Low_STh	Offset	Reset Value
Low Speed Threshold	70_H	628_H



Field	Bits	Type	Description
LST	11:0	r	**Low Speed Threshold** This field contains a value which is compared to the FM_REMAIN field prior to initiating a Low Speed transaction. The transaction is started only if FM_REMAIN > = this field. The value is calculated by HCD with the consideration of transmission and setup overhead.
Res	31:12		**Reserved**

RH Descriptor

RH_D	Offset	Reset Value
RH Descriptor	74_H	1_H



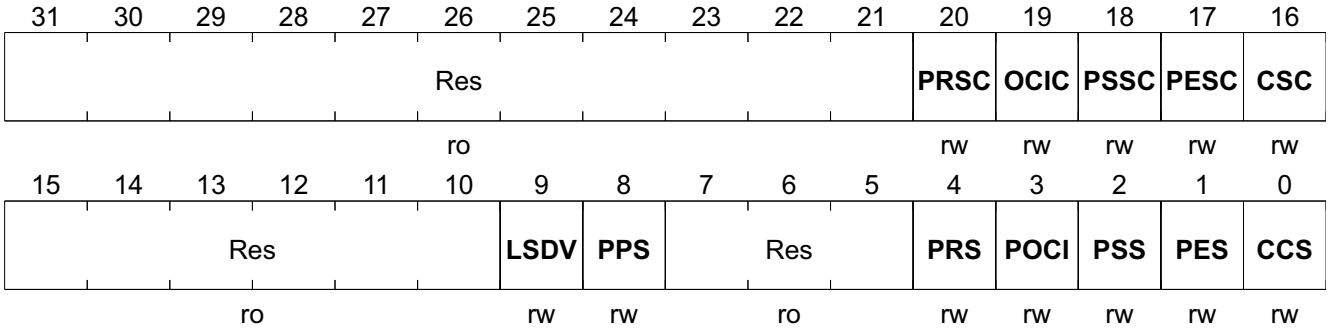
Field	Bits	Type	Description
Res	31:30	ro	**Reserved** Not Applicable.

Field	Bits	Type	Description
CRWE	29	rw	**Clear Remote Wakeup Enable** 0_B , Has no effect. 1_B , Clears Device Remove Wakeup Enable
DRWE	28		**Device Remote Wakeup Enable** This bit enables a Connect Status Change bit as a resume event, causing a USBSUSPEND to USBRESUME state transition and setting the Resume Detected interrupt. 0_B , Connect Status Change is not a remote wakeup event 1_B , Connect Status Change is a remote wakeup event
OCIC	27		**Over Current Indication Change** This bit is set by hardware when a change has occurred to the OCI field of this register. The HCD clears this bit by writing a 1. Writing a 0 has no effect.
LPSC	26		**Local Power Switch Change (read)** This bit is always read as 0. **Set Global Power (write)** In global power mode (PSM =0), This bit is written to 1 to turn on power to all ports (clearPPS). In per-port power mode, it sets PPS only on ports whose PPCM bit is not set. Writing a 0 has no effect.
OCI	25		**Over Current Indication** This bit reports overcurrent conditions when the global reporting is implemented. When set, an overcurrent condition exists. When cleared, all power operations are normal. If per-port overcurrent protection is implemented this bit is always 0
LPS	24		**Local Power Switch (read)** This bit is always read as 0. **Clear Global Power (write)** In global power mode (PSM =0), This bit is written to 1 to turn off power to all ports (clearPPS). In per-port power mode, it clears PPS only on ports whose PPCM bit is not set. Writing a 0 has no effect.
PPCM	23:16		**Port Power Control** Each bit indicates if a port is affected by a global power control command when PSM is set. When set, the port's power state is only affected by per-port power control (Set/ClearPortPower). When cleared, the port is controlled by the global power (switchSet/ClearGlobalPower). If the deviceis configured to global switching mode (PSM =0), this field is not valid. Bit 0: Reserved Bit 1: Ganged-power mask on Port #1 Bit 2: Ganged-power mask on Port #2 ... Bit7: Ganged-power mask on Port #7
Res	15:12	ro	**Reserved** Not Applicable

Field	Bits	Type	Description
NOCP	11	rw	**No Over Current Protect** This bit describes how the overcurrent status for the Root Hub ports are reported. When this bit is cleared, the OCPM field specifies global or per-port reporting. 0_B , Over-current status is reported collectively for all downstream ports 1_B , No overcurrent protection supported
OCPM	10		**Over Current Protect Mode** This bit describes how the overcurrent status for the Root Hub ports are reported. At reset, this fields should reflect the same mode as PSM. This field is valid only if the NOCP field is cleared. 0_B , Over-current status is reported collectively for all downstream ports 1_B , Over-current status is reported on a per-port basis
NPS	9		**No Power Switch** These bits are used to specify whether power switching is supported or port are always powered. It is implementation-specific. When this bit is cleared, the PSM specifies global or per-port switching. 0_B , Ports are power switched 1_B , Ports are always powered on when the HC is powered on
PSM	8		**Power Switch Mode** This bit is used to specify how the power switching of the Root Hub ports is controlled. It is implementation-specific. This field is only valid if the NPS field is cleared. 0_B , All ports are powered at the same time 1_B , Each port is powered individually. This mode allows portpower to be controlled by either the global switch or per-port switching. If the PPCM bit is set per-port switching. If the PPCM bit is set, the port responds only to port power commands (Set/ClearPortPower). If the port mask is cleared, then the port is controlled only by the global power switch (Set/ClearGlobalPower).
NUM_P	7:0	ro	**Number Port**

Port X Status

PX_St	Offset	Reset Value
Port X Status	78_H	0_H



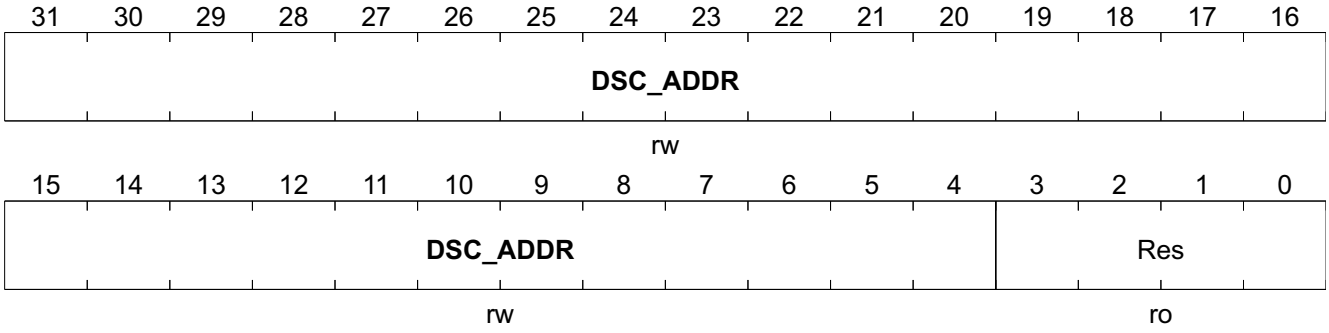
Field	Bits	Type	Description
Res	31:21	ro	**Reserved** Not Applicable.
PRSC	20	rw	**Port Reset Status Change** This bit is set at the end of the 10 ms port reset signal. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. 0_B , Port reset is not complete 1_B , Port reset is complete
OCIC	19		**Over Current Indicator Change** This bit is valid only if overcurrent conditions are reported on a per-port basis. This bit is set when Root Hub changes the POCI bit. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. 0_B , No change in POCI 1_B , OCI has changed
PSSC	18		**Port Suspend Status Change** This bit is set when the full resume sequence has been completed. This sequence includes the 20 s resume pulse, LS EOP, and 3 ms resychronization delay. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. This bit is also cleared when RSC is set. 0_B , Resume is not completed 1_B , Resume completed
PESC	17		**Port Enable Status Change** This bit is set when hardware events cause the PES bit to be cleared. Changes from HCD writes do not set this bit. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. 0_B , No change in PES 1_B , Change in PES

Field	Bits	Type	Description
CSC	16	rw	**Connect Status Change** This bit is set whenever a connect or disconnect event occurs. The HCD writes a 1 to clear this bit. Writing a 0 has no effect. If CCS is cleared when a SPR, SPE, or SPS write occurs, this bit is set to force the driver to re-evaluate the connection status since these writes should not occur if the port is disconnected. *Note: This bit is set only after a Root Hub reset to inform the system that the device is attached.* 0_B , No change in CCS 1_B , Change in CCS
Res	15:10	ro	**Reserved** Not Applicable.
LSDV	9	rw	**Low Speed Device Attached (read)** This bit indicates the speed of the device attached to this port. When set, a Low Speed device is attached to this port. When clear, a Full Speed device is attached to this port. This field is valid only when the CurrentConnectStatus is set. 0_B , Full speed device attached 1_B , Low speed device attached **Clear Port Power (write)** The HCD clears the PortPowerStatus bit by writing a 1 to this bit. Writing a 0 has no effect.
PPS	8		**Port Power Status (read)** This bit reflects the port's power status, regardless of the type of power switching implemented. This bit is cleared if an overcurrent condition is detected. HCD sets this bit by writing SPP or SGP. HCD clears this bit by writing CPP or CGP. Which power control switches are enabled is determined by PSM and PPCM[NDP] . In global switching mode (PSM =0), only Set/ClearGlobalPower controls this bit. In per-port power switching (PSM =1), if the PPCM[NDP] bit for the port is set, only Set/ClearPortPower commands are enabled. If the mask is not set, only Set/ClearGlobalPower commands are enabled. When port power is disabled, CCS,PES, PSS, and PRS should be reset. 0_B , Port power is off 1_B , Port power is on **Set Port Power (write)** The HCD writes a 1 to set the PPS bit. Writing a 0 has no effect. *Note: This bit is always reads 1 if power switching is not supported.*
Res	7:5	ro	**Reserved** Not Applicable.
PRS	4	rw	**Port Reset Status (read)** When this bit is set by a write to SPR, port reset signaling is asserted. When reset is completed, this bit is cleared when PRSC is set. This bit cannot be set if CCS is cleared. 0_B , Port reset signal is not active 1_B , Port reset signal is active **Set Port Reset (write)** The HCD sets the port reset signaling by writing a 1 to this bit Writing a 0 has no effect. If CCS is cleared, this write does not set PRS, but instead sets CSC. This informs the driver that it attempted to reset a disconnected port.

Field	Bits	Type	Description
POCI	3	rw	**Port Over Current Indicator (read)** This bit is only valid when the Root Hub is configured in such a way that overcurrent conditions are reported on a per-port basis. If per-port overcurrent reporting is not supported, this bit is set to 0. If cleared, all power operations are normal for this port. If set, an overcurrent condition exists on this port. This bit always reflects the overcurrent input signal. 0_B , No overcurrent condition 1_B , Overcurrent condition detected **Clear Suspend Status (write)** The HCD writes a 1 to initiate a resume. Writing a 0 has no effect. A resume is initiated only if PSS is set.
PSS	2		**Port Suspend Status (read)** This bit indicates the port is suspended or in the resume sequence. It is set by a SSS write and cleared when PSSC is set at the end of the resume interval. This bit cannot be set if CCS is cleared. This bit is also cleared when PRSC is set at the end of the port reset or when the HC is placed in the USBRESUME state. If an upstream resume is in progress, it should propagate to the HC. 0_B , Port is not suspended 1_B , Port is suspended **Set Port Suspend (write)** The HCD sets the PSS bit by writing a 1 to this bit. Writing a 0 has no effect. If CCS is cleared, this write does not set PSS; instead it sets CSC. This informs the driver that it attempted to suspend a disconnected port.
PES	1		**Port Enable Status (read)** This bit indicates whether the port is enabled or disabled. The Root Hub may clear this bit when an overcurrent condition, disconnect event, switched-off power, or operational bus error such as babble is detected. This change also causes PESC to be set. HCD sets this bit by writing SPE and clears itby writing CPE. This bit cannot be set when CCS is cleared. This bit is also set, if not already, at the completion of a port reset when RSC is set or port suspend when SSC is set. *Note: This informs the driver that it attempted to enable a disconnected port.* 0_B , Port is disabled 1_B , Port is enabled **Set Port Enable (write)** The HCD sets PortEnableStatus by writing a 1. Writing a 0 has no effect. If CCS is cleared, this write does not set PES, but instead sets CSC.This informs the driver that it attempted to enable a disconnected port.
CCS	0		**Current Connect Status (read)** This bit reflects the current state of the downstream port. 0_B , No device connected 1_B , Device connected **Clear Port Enable (write)** The HCD writes a 1 to this bit to clear the PES. Writing a 0 has no effect. The CCS is not affected by any write. *Note: This bit is always read 1 when the attached device is nonremovable*

Host Descriptor Head Starting Address

HDHS_Ad	Offset	Reset Value
Host Descriptor Head Starting Address	80_H	0_H



Field	Bits	Type	Description
DSC_ADDR	31:4	rw	**Descriptor Chain Address** This field indicates the starting address of the host mode descriptor chain. DMA read the descriptor from this location when it is first enabled.
Res	3:0	ro	**Reserved**

9 Electrical Characteristics

9.1 Absolute Maximum Ratings

Table 45 Absolute Maximum Ratings

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Supply Voltage	V_{CC}	-0.5	–	1.9	V	–
Input Voltage	V_I	-0.5	–	V_{CC} +0.3	V	–
Output Voltage	V_O	-0.5	–	V_{CC} +0.3	V	–
Storage Temperature	T_S	-65	–	150	°C	–
Ambient Temperature	T_A	-0	–	70	°C	–
ESD Protection	V_{ESD}	-0	–	2000	V	–

Attention: Stresses above the max. values listed here may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.

9.2 DC Characteristics

Table 46 DC Characteristics

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Supply Voltage	V_{CC}	1.7	1.8	1.9	V	–
I/O Supply Voltage	V_{IO_CC}	3.0	3.3	3.6	V	–
Power Supply Current	I_{CC}	–	–	–	mA	V_{CC} = 1.8 V
I/ O Power Supply Current	I_{IO_CC}	–	–	–	mA	V_{CC} = 3.3 V
Input Low Voltage	V_{IL}	-0.5	–	0.8	V	–
Input High Voltage	V_{IH}	2.0	–	3.8	V	–
Input Low Leakage Current	I_{ILL}	-10	–	10	µA	V_{IN} = 0.8 V
Input High Leakage Current	I_{IHL}	-10	–	10	µA	V_{IN} = 2.0 V
Output Low Voltage	V_{OL}	–	–	0.4	V	I_{OUT} = 2~8 mA
Output High Voltage	V_{OH}	2.4	–	–	V	I_{OUT} = 2~-8 mA
Input Pin Capacitance	C_{IP}	5	–	8	pF	–
Pin Inductance	L_{PI}	TBD	–	–	nH	–

9.3 AC Timing

9.3.1 SDRAM Interface

(Unit: ns, Min: best case, Max: worst case)

Table 47 SDRAM Interface Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Clock cycle time	t_{CK}	–	11.4	–	ns	–
Command/address setup delay time in precharge stage	t_{PS}	1.5	–	–	ns	–
Command/address hold delay time in precharge stage	t_{PH}	1	–	–	ns	–
Command/address setup delay time in active stage	t_{AS}	1.5	–	–	ns	–
Command/address hold delay time in active stage	t_{AH}	1	–	–	ns	–
Command/address setup delay time in write stage	t_{WS}	1.5	–	–	ns	–
Command/address hold delay time in write stage	t_{WH}	1	–	–	ns	–
Command/address setup delay time in read stage	t_{RS}	1.5	–	–	ns	–
Command/address hold delay time in read stage	t_{RH}	1	–	–	ns	–

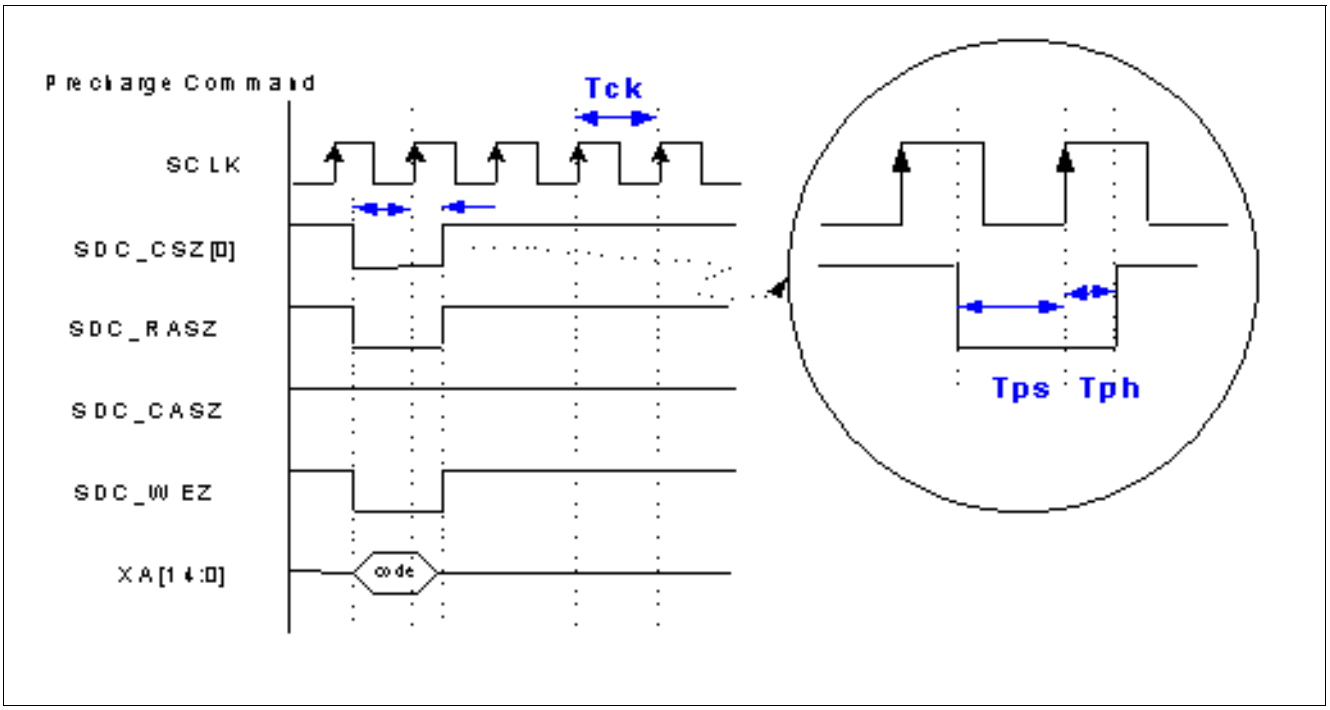


Figure 20 Precharge Command

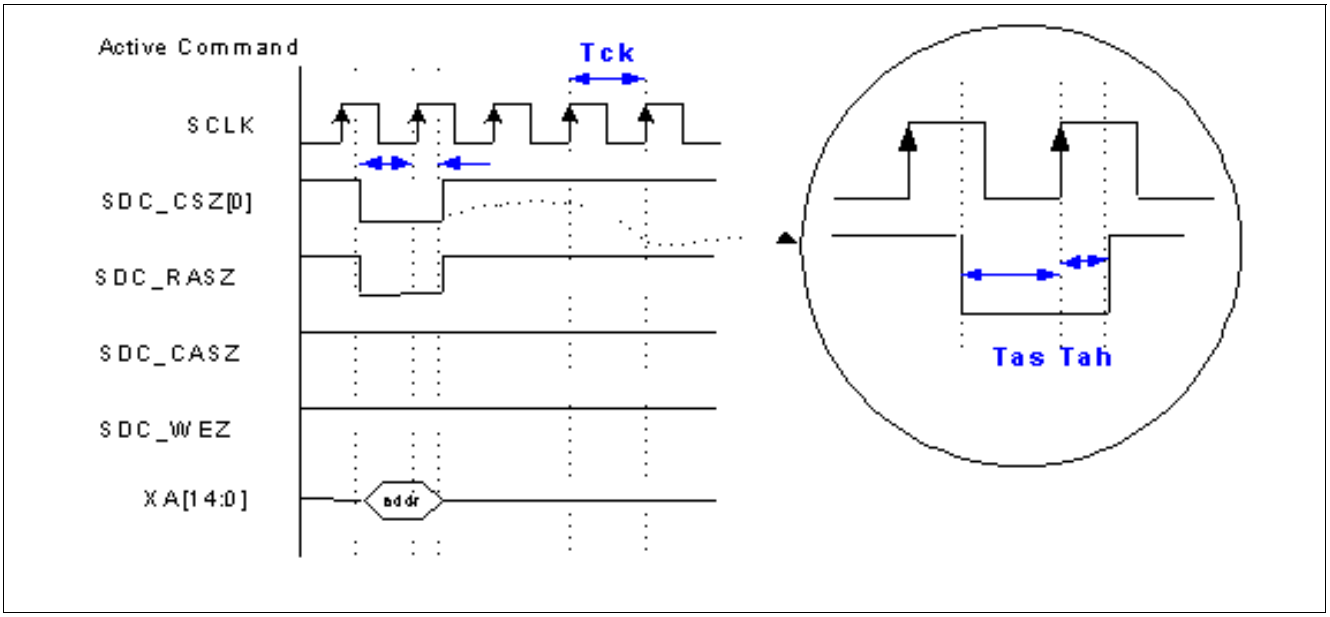


Figure 21 Active Command

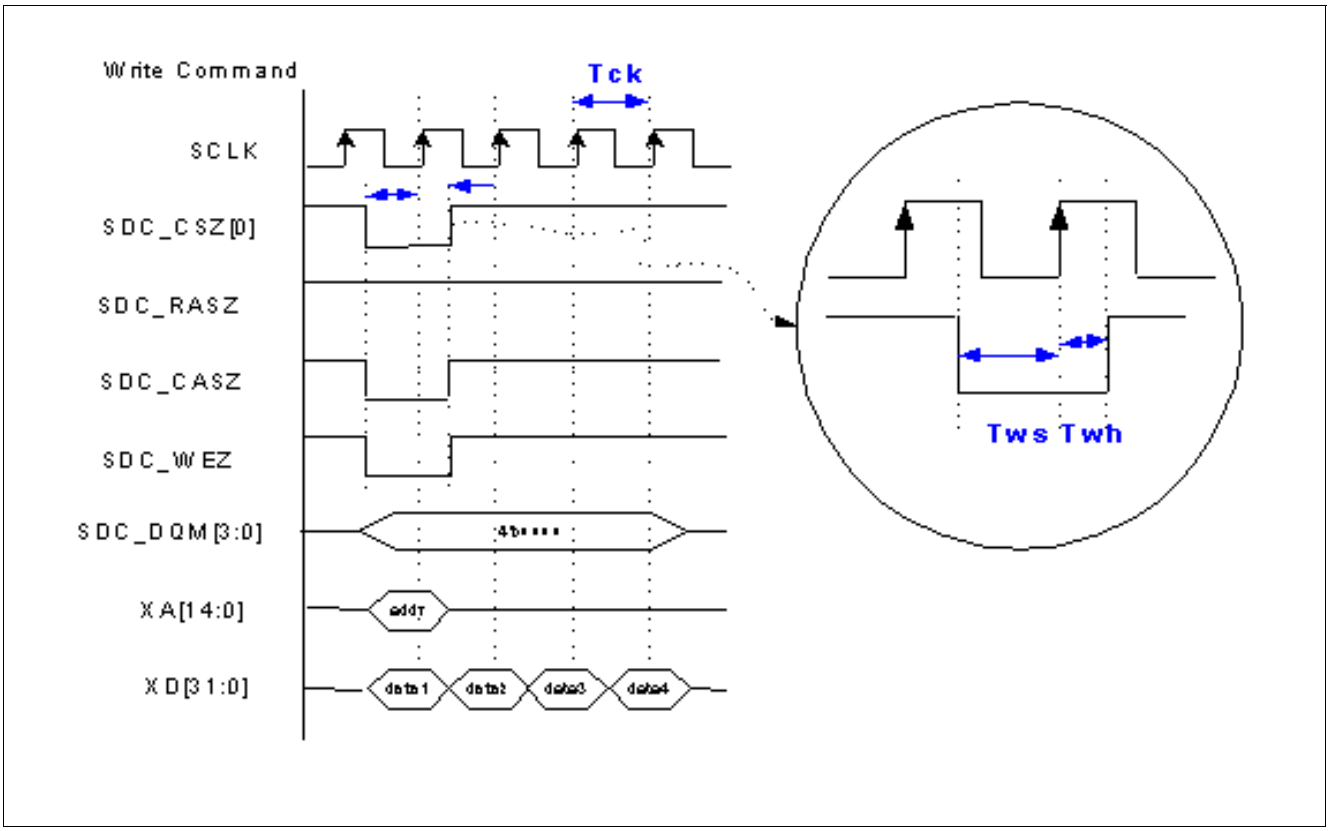


Figure 22 Write Command

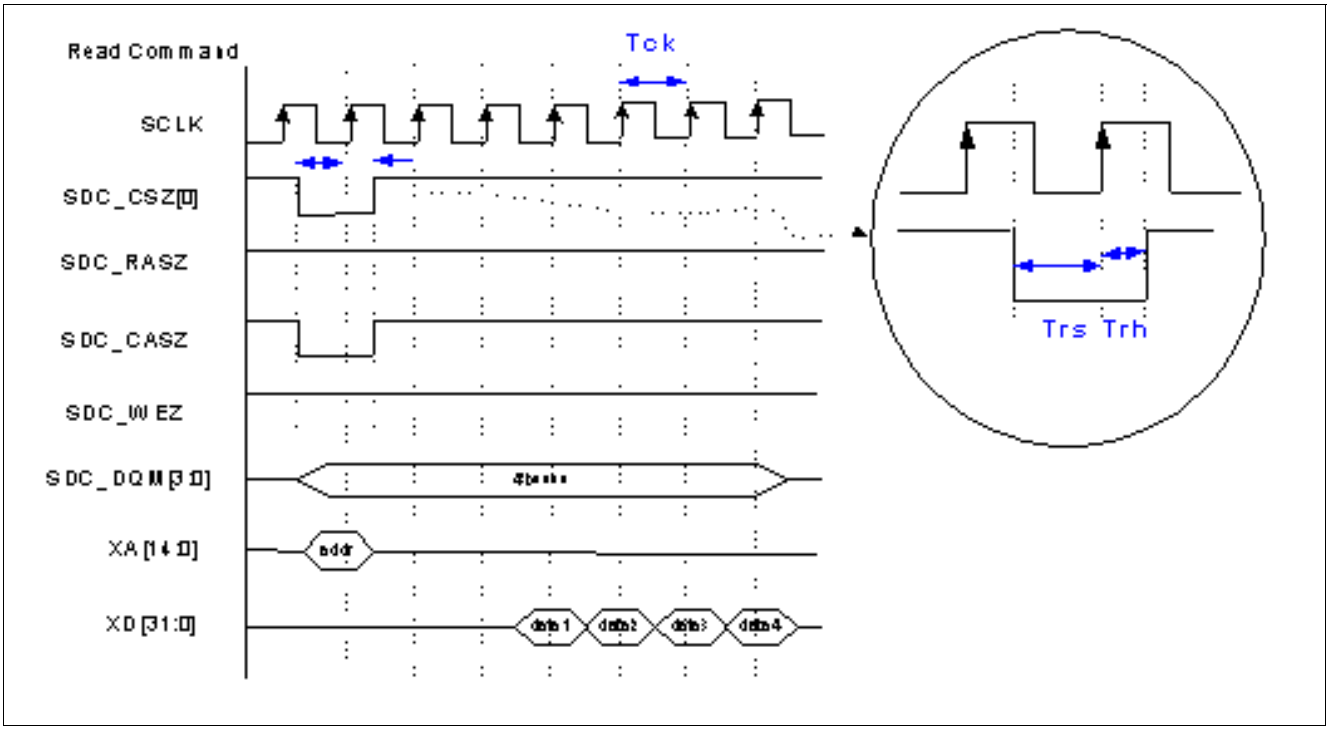


Figure 23 Read Command

9.3.2 Memory Bus Read Timing

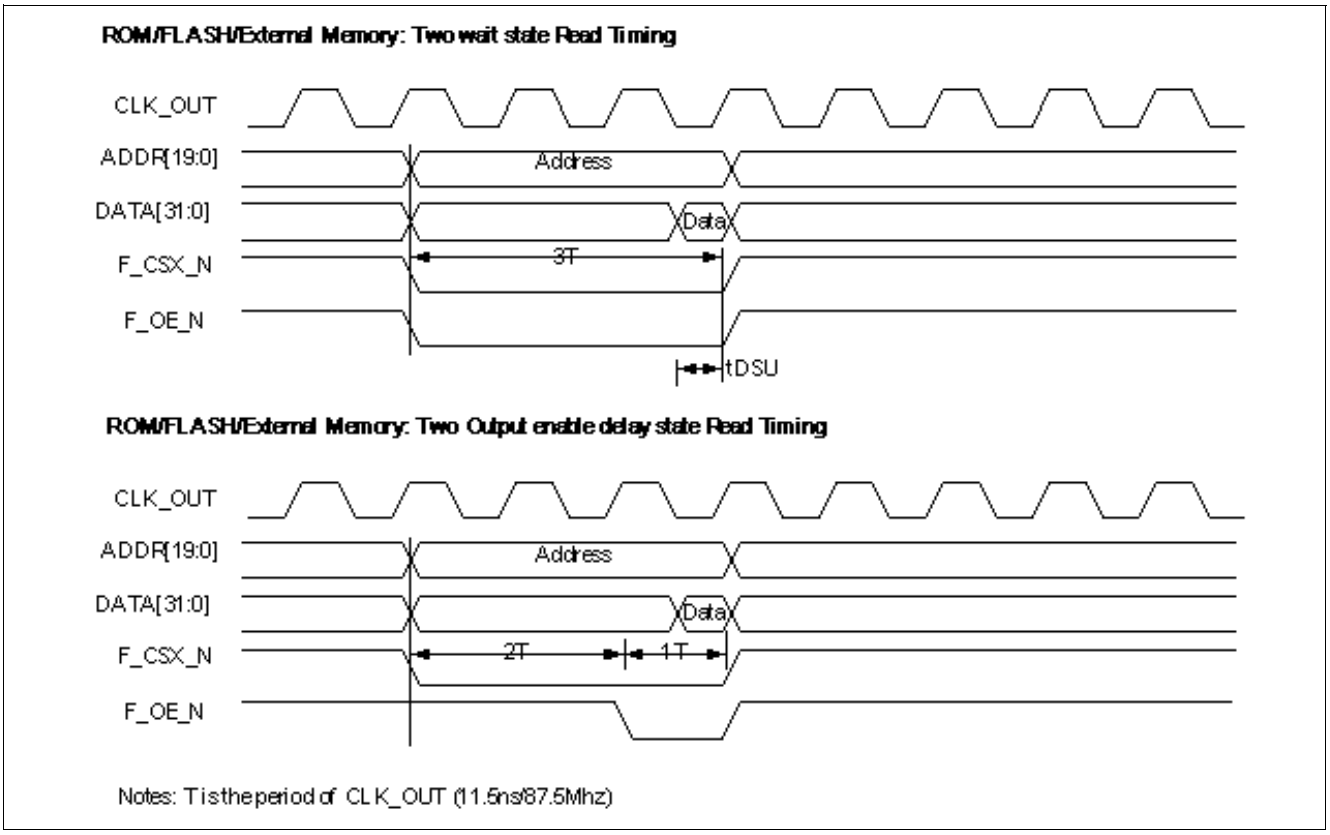


Figure 24 Memory Bus Read Timing

Note: T is the period of CLK_OUT (11.5 ns/87.5 MHz)

Table 48 Memory Bus Read Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Data to CLK_OUT rising setup time	t_{RDSU}	TBD	–	–		–
Data to CLK_OUT rising hold time	t_{RDH}	TBD	–	–		–
Address/F_CSX_N pulse width	t_{AC}	–	(n+1)T	–		–
Address/F_CSX_N to F_OE_N setup	t_{AOE}	–	nT	–		–

9.3.3 Memory Bus Write Timing

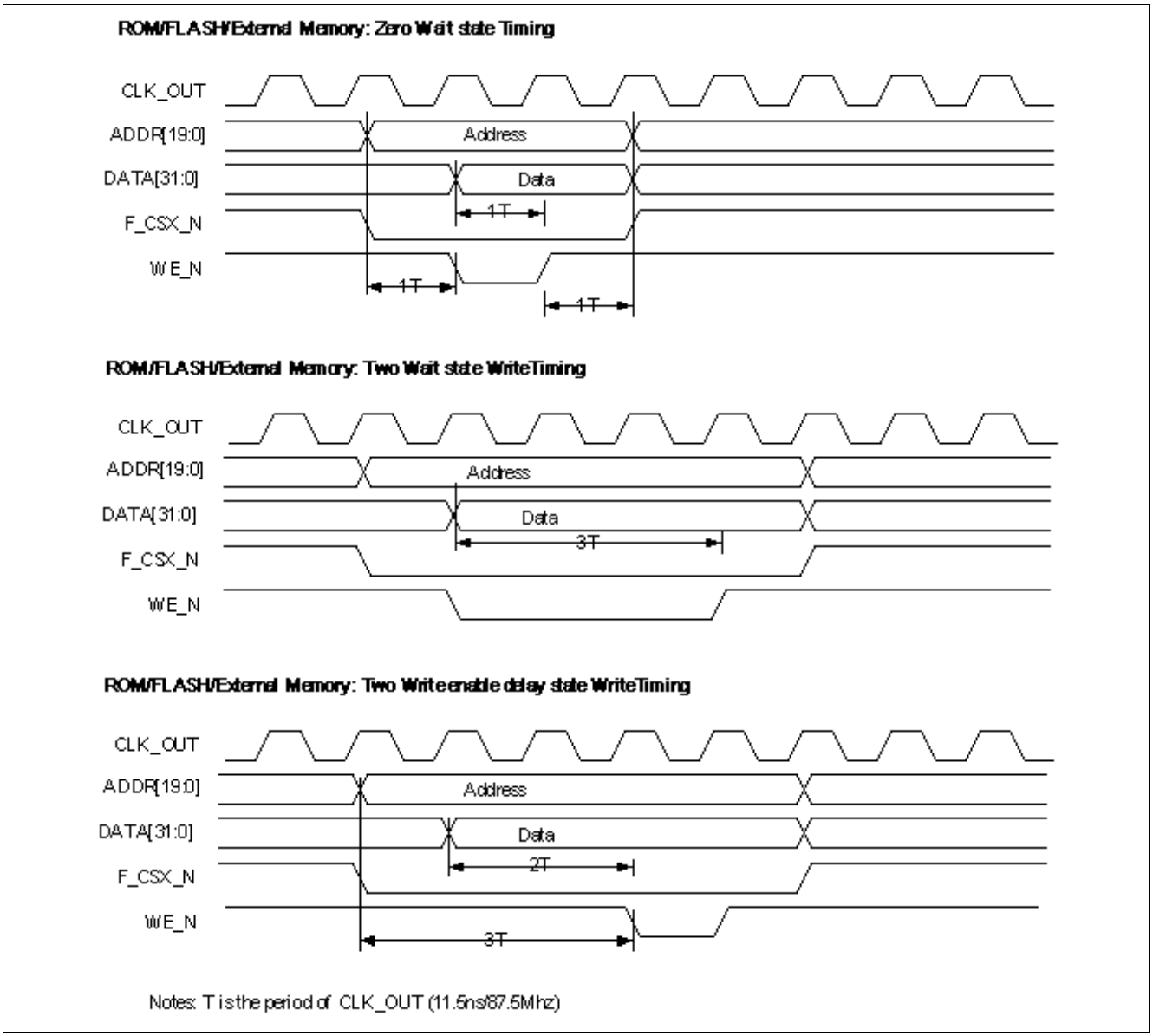


Figure 25 Memory Bus Write Timing

Note: T is the period of CLK_OUT (11.5 ns/87.5 MHz)

Table 49 Memory Bus Write Timing

Parameter	Symbol	Values			Unit	Note / Test Condition
		Min.	Typ.	Max.		
Address/CS to WE_N falling setup time	t_{ASU}	–	(n+1)T	–		–
Data to WE_N rising setup time	t_{WDSU}	–	(n+1)T	–		–
Data to WE_N rising hold time	t_{WDH}	–	1T	–		–
WE_N pulse width	t_{WEP}	–	(n+1)T	–		–

10 Package Outlines

10.1 Ball Grid Array (BGA) 324-pin

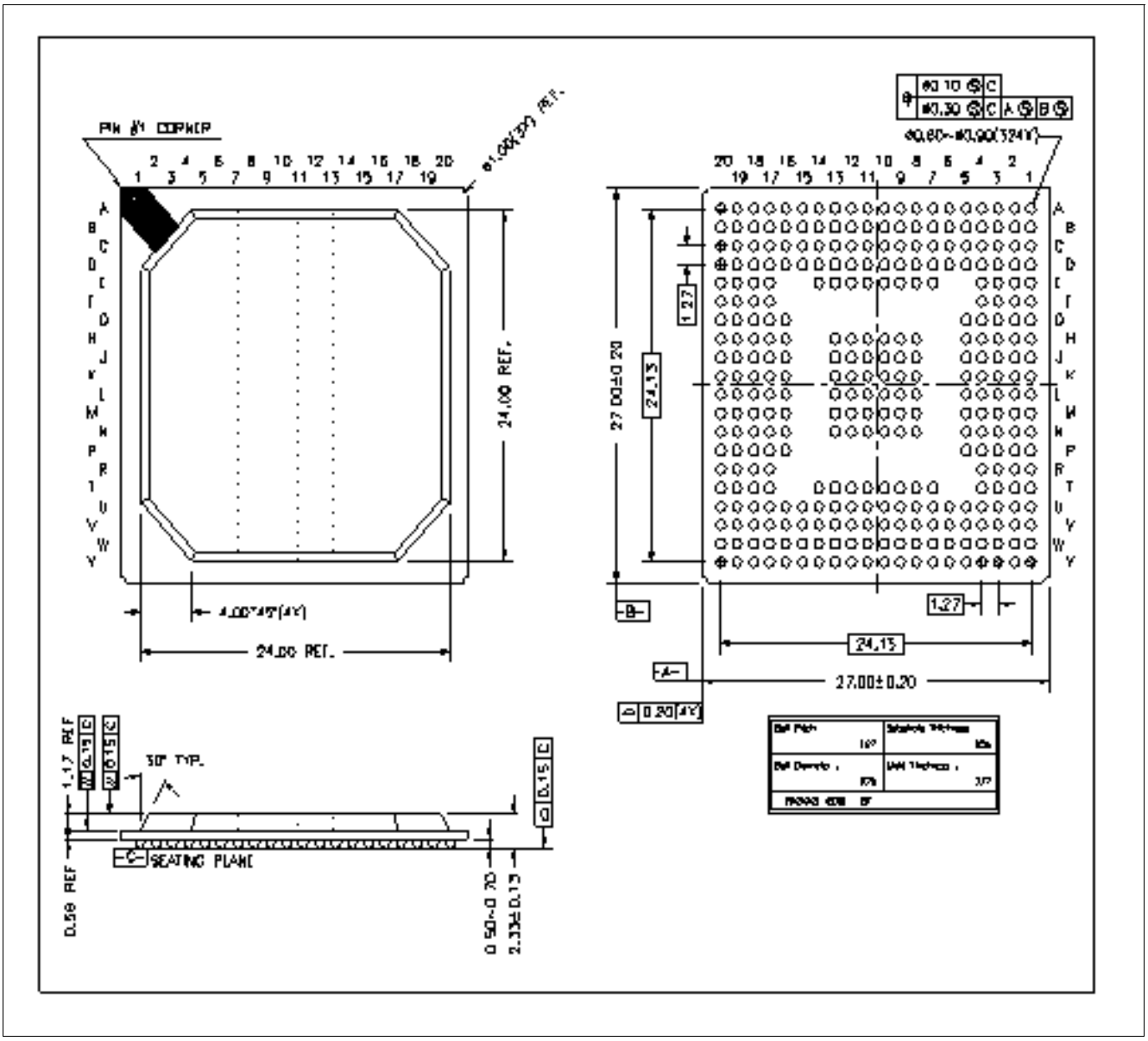


Figure 26 **P-BGA-324-1** (Plastic Ball Grid Array Package)

10.2 Plastic Quad Flat Pack (PQFP) 208-pin

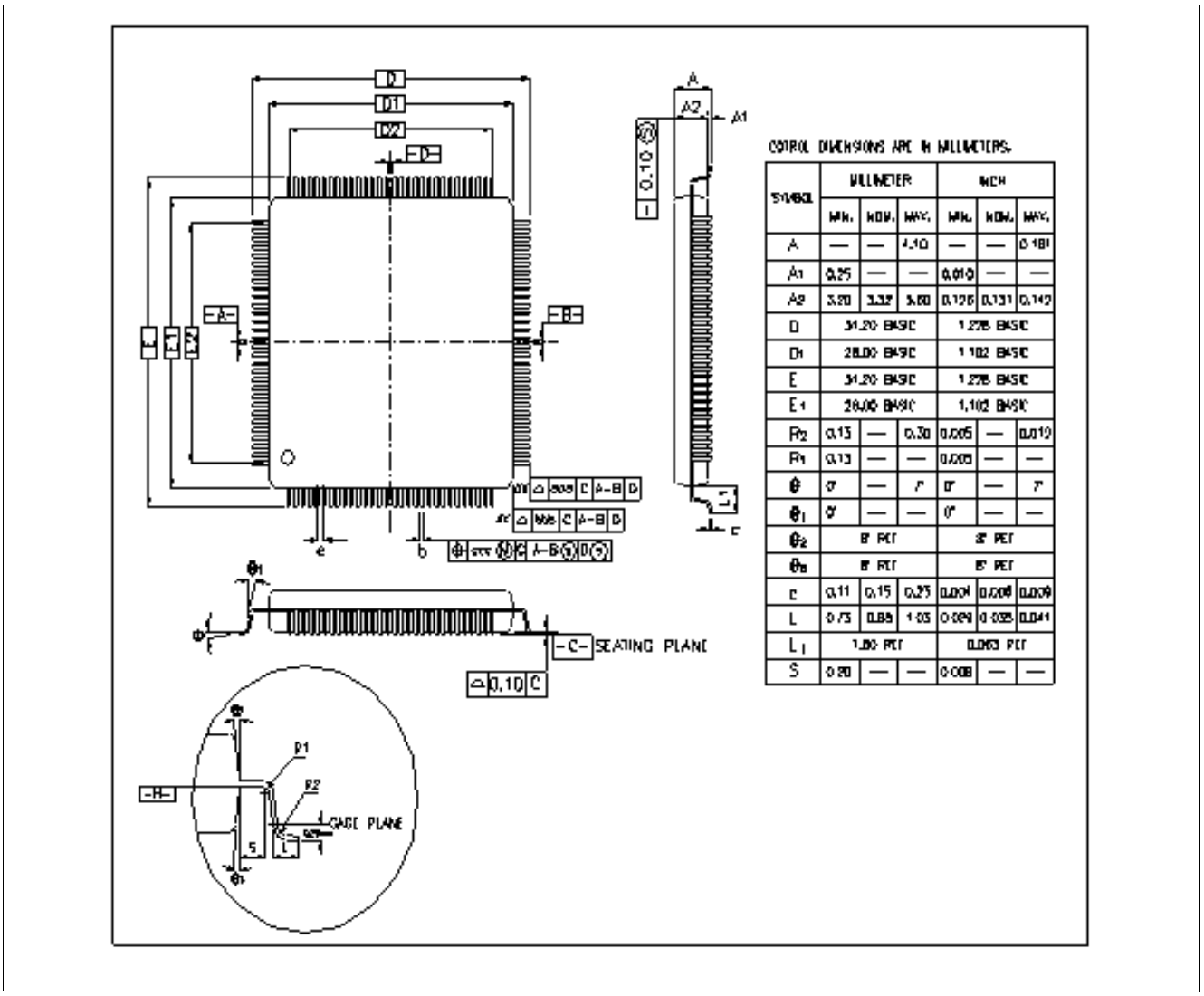


Figure 27 P-QFP-208-1 (Plastic Quad Flat Package)

Note: Dimensions in mm

Symbol	Millimeter			Inch		
	Min.	Typ.	Max.	Min.	Typ.	Max.
b	0.17	0.20	0.27	0.007	0.008	0.011
e	0.50 BSC.			0.020 BSC.		
D2	25.50			1.004		
E2	25.50			1.004		
aaa	0.25			0.010		
bbb	0.20			0.008		
ccc	0.08			0.003		

Terminology

A	
AHB	Advance High performance Bus
ALE	Address Latch Enable
AN	Auto-Negotiation
APB	Advanced Peripheral Bus
ASB	Advanced System Bus
ASIC	Application Specific Integrated Circuit
B	
BC	BroadCast
BP	Back Pressure
BPDU	Bridge Protocol Data Unit
BISS	Build In Self test error Skip
BIST	Build In Self Test
C	
CLK	Clock
COL	Collision
CoS	Class of Service
CRC	Cyclic Redundancy Check
CRS	Carrier Sense
CSX	Chip Select for external I/O bank0
D	
DFE	Decision Feedback Equalization
DMA	Direct Memory Access
F	
FC	Flow Control
FIFO	First-In-First-Out
G	
GND	Ground
GPIO	General Purpose I/O
GPIOL	GPIO of groupL
GPIOM	GPIO of groupM
GPSI	General Purpose Serial Interface
H	
HOL	Head-on-Line
I	
INTC	Interrupt Control Registers
INTX	Interrupt for external I/O bank0
IPG	Inter Packet Gap
IRQ	Interrupt ReQuest
J	
JTAG	Joint Test Action Group

L

LSb	Least Significant Bit
LSB	Least Significant Byte

M

MAC	Media Access Control
MC	Multicast
MDC	Management Data Clock
MDIO	Management Data I/O
MDI	Medium dependent interface
MDIX	MDI Crossover
MII	Media Independent Interface
MIPS	Million Instructions Per Second
MMU	Memory Management Unit

N

NAT	Network Address Translation
NRZI	Non Return Zero Invert
NRZ	Non Return Zero

P

PCS	Physical Coding Sublayer
PHY	PHYsical Layer
PLL	Phase Locked Loop
PMA	Physical Medium Attachment
PMD	Physical medium Dependent
PQFP	Plastic Quad Flat Package

R

RISC	Reduced Instruction Set Computer
RX	Receive
RXD	Receive Data
RXDV	Receive Data Valid

S

SA	Source Address
SMC	Flash Control Registers
SW	Switch
SYSC	System Control Registers

T

TOS	Type Of Service
TX	Transmit
TXC	Transmit Clock
TXE	Transmit Enable
TXD	Transmit Data

U

UART	Universal Asynchronous Receiver Transmitter

V

VLAN	Virtual LAN
W	
WAN	Wide Area Networks

www.infineon.com

Published by Infineon Technologies AG