LAN9250

10/100 Industrial Ethernet Controller & PHY

Highlights

- 16-bit 10/100 industrial Ethernet controller & PHY
- Interfaces to most 8/16-bit embedded controllers and 32-bit embedded controllers with an 8/16-bit bus
- Integrated Ethernet PHY with HP Auto-MDIX
- Integrated Ethernet MAC
- Compliant with Energy Efficient Ethernet 802.3az
- Wake on LAN (WoL) support
- Integrated IEEE 1588v2 hardware time stamp unit
- Cable diagnostic support
- 1.8V to 3.3V variable voltage I/O
- Integrated 1.2V regulator for single 3.3V operation
- Low pin count and small body size package

Target Applications

- Cable, satellite, and IP set-top boxes
- Digital televisions & video recorders
- VoIP/Video phone systems
- Home gateways
- Test/Measurement equipment
- Industrial automation systems

Key Benefits

- Single-chip Ethernet controller
  - Fully compliant with IEEE 802.3/802.3u standards
  - Integrated Ethernet MAC and PHY
  - 10BASE-T and 100BASE-TX support
  - 100BASE-FX support for external fiber transceiver
  - Automatic polarity detection and correction (HP Auto-MDIX)
  - Full- and Half-duplex support
  - Full-duplex flow control
  - Backpressure for half-duplex flow control
  - Preamble generation and removal
  - Automatic 32-bit CRC generation and checking
  - Automatic payload padding and pad removal
  - Loop-back modes
- Eliminates dropped packets
  - Internal buffer memory can store over 200 packets
  - Automatic PAUSE and back-pressure flow control
- Flexible address filtering modes
  - One 48-bit perfect address
  - 64 hash-filtered multicast addresses
  - Pass all multicast
  - Promiscuous mode
  - Inverse filtering
  - Pass all incoming with status report
  - Disable reception of broadcast packets
- 8/16-Bit Host Bus Interface
  - Indexed register or multiplexed bus
  - 16Kbyte FIFO with flexible TX/RX allocation
  - SPI / Quad SPI support
- IEEE 1588v2 hardware time stamp unit
  - Global 64-bit tunable clock
  - Ordinary clock: master / slave, one-step / two-step, end-to-end / peer-to-peer delay
  - Fully programmable timestamp on TX or RX, timestamp on GPIO
  - 64-bit timer comparator event generation (GPIO or IRQ)
- Comprehensive power management features
  - 3 power-down levels
  - Wake on link status change (energy detect)
  - Magic packet wakeup, Wake on LAN (WoL), wake on broadcast, wake on perfect DA
  - Wakeup indicator event signal
  - Link status change
- Power and I/O
  - Integrated power-on reset circuit
  - Latch-up performance exceeds 150mA per EIA/JESD78, Class II
  - JEDEC Class 3A ESD performance
  - Single 3.3V power supply (integrated 1.2V regulator)
- Additional Features
  - Multifunction GPIOs
  - General purpose timer
  - Optional EEPROM interface
  - Ability to use low cost 25MHz crystal for reduced BOM
- Packaging
  - Pb-free RoHS compliant 64-pin QFN or 64-pin TQFP-EP
- Available in commercial, industrial, and extended industrial* temp. ranges

*Extended temp. (105ºC) is supported only in the 64-QFN with an external voltage regulator (internal regulator must be disabled) and 2.5V (typ) Ethernet magnetics.

TO OUR VALUED CUSTOMERS

It is our intention to provide our valued customers with the best documentation possible to ensure successful use of your Microchip products. To this end, we will continue to improve our publications to better suit your needs. Our publications will be refined and enhanced as new volumes and updates are introduced.

If you have any questions or comments regarding this publication, please contact the Marketing Communications Department via E-mail at **docerrors@microchip.com**. We welcome your feedback.

Most Current Documentation

To obtain the most up-to-date version of this documentation, please register at our Worldwide Web site at:

http://www.microchip.com

You can determine the version of a data sheet by examining its literature number found on the bottom outside corner of any page. The last character of the literature number is the version number, (e.g., DS30000000A is version A of document DS30000000).

Errata

An errata sheet, describing minor operational differences from the data sheet and recommended workarounds, may exist for current devices. As device/documentation issues become known to us, we will publish an errata sheet. The errata will specify the revision of silicon and revision of document to which it applies.

To determine if an errata sheet exists for a particular device, please check with one of the following:

- Microchip's Worldwide Web site; http://www.microchip.com
- Your local Microchip sales office (see last page)

When contacting a sales office, please specify which device, revision of silicon and data sheet (include -literature number) you are using.

Customer Notification System

Register on our web site at **www.microchip.com** to receive the most current information on all of our products.

1.0 PREFACE

1.1 General Terms

TABLE 1-1: GENERAL TERMS

Term	Description
10BASE-T	10 Mbps Ethernet, IEEE 802.3 compliant
100BASE-TX	100 Mbps Fast Ethernet, IEEE802.3u compliant
ADC	Analog-to-Digital Converter
ALR	Address Logic Resolution
AN	Auto-Negotiation
BLW	Baseline Wander
BM	Buffer Manager - Part of the switch fabric
BPDU	Bridge Protocol Data Unit - Messages which carry the Spanning Tree Protocol information
Byte	8 bits
CSMA/CD	Carrier Sense Multiple Access/Collision Detect
CSR	Control and Status Registers
CTR	Counter
DA	Destination Address
DWORD	32 bits
EPC	EEPROM Controller
FCS	Frame Check Sequence - The extra checksum characters added to the end of an Ethernet frame, used for error detection and correction.
FIFO	First In First Out buffer
FSM	Finite State Machine
GPIO	General Purpose I/O
Host	External system (Includes processor, application software, etc.)
IGMP	Internet Group Management Protocol
Inbound	Refers to data input to the device from the host
Level-Triggered Sticky Bit	This type of status bit is set whenever the condition that it represents is asserted. The bit remains set until the condition is no longer true and the status bit is cleared by writing a zero.
lsb	Least Significant Bit
LSB	Least Significant Byte
LVDS	Low Voltage Differential Signaling
MDI	Medium Dependent Interface
MDIX	Media Independent Interface with Crossover
MII	Media Independent Interface
MIIM	Media Independent Interface Management
MIL	MAC Interface Layer
MLD	Multicast Listening Discovery
MLT-3	Multi-Level Transmission Encoding (3-Levels). A tri-level encoding method where a change in the logic level represents a code bit “1” and the logic output remaining at the same level represents a code bit “0”.
msb	Most Significant Bit
MSB	Most Significant Byte

TABLE 1-1: GENERAL TERMS (CONTINUED)

Term	Description
NRZI	Non Return to Zero Inverted. This encoding method inverts the signal for a “1” and leaves the signal unchanged for a “0”
N/A	Not Applicable
NC	No Connect
OUI	Organizationally Unique Identifier
Outbound	Refers to data output from the device to the host
PISO	Parallel In Serial Out
PLL	Phase Locked Loop
PTP	Precision Time Protocol
RESERVED	Refers to a reserved bit field or address. Unless otherwise noted, reserved bits must always be zero for write operations. Unless otherwise noted, values are not guaranteed when reading reserved bits. Unless otherwise noted, do not read or write to reserved addresses.
RTC	Real-Time Clock
SA	Source Address
SFD	Start of Frame Delimiter - The 8-bit value indicating the end of the preamble of an Ethernet frame.
SIPO	Serial In Parallel Out
SMI	Serial Management Interface
SQE	Signal Quality Error (also known as “heartbeat”)
SSD	Start of Stream Delimiter
UDP	User Datagram Protocol - A connectionless protocol run on top of IP networks
UUID	Universally Unique IDentifier
WORD	16 bits

1.2 Buffer Types

TABLE 1-2: BUFFER TYPES

Buffer Type	Description
IS	Schmitt-triggered input
VIS	Variable voltage Schmitt-triggered input
VO8	Variable voltage output with 8 mA sink and 8 mA source
VOD8	Variable voltage open-drain output with 8 mA sink
VO12	Variable voltage output with 12 mA sink and 12 mA source
VOD12	Variable voltage open-drain output with 12 mA sink
VOS12	Variable voltage open-source output with 12 mA source
VO16	Variable voltage output with 16 mA sink and 16 mA source
PU	50 µA (typical) internal pull-up. Unless otherwise noted in the pin description, internal pull-ups are always enabled. Internal pull-up resistors prevent unconnected inputs from floating. Do not rely on internal resistors to drive signals external to the device. When connected to a load that must be pulled high, an external resistor must be added.
PD	50 µA (typical) internal pull-down. Unless otherwise noted in the pin description, internal pull-downs are always enabled. Internal pull-down resistors prevent unconnected inputs from floating. Do not rely on internal resistors to drive signals external to the device. When connected to a load that must be pulled low, an external resistor must be added.
AI	Analog input
AIO	Analog bidirectional
ICLK	Crystal oscillator input pin
OCLK	Crystal oscillator output pin
ILVPECL	Low voltage PECL input pin
OLVPECL	Low voltage PECL output pin
P	Power pin

1.3 Register Nomenclature

TABLE 1-3: REGISTER NOMENCLATURE

Register Bit Type Notation	Register Bit Description
R	**Read:** A register or bit with this attribute can be read.
W	**Read:** A register or bit with this attribute can be written.
RO	**Read only:** Read only. Writes have no effect.
WO	**Write only:** If a register or bit is write-only, reads will return unspecified data.
WC	**Write One to Clear:** Writing a one clears the value. Writing a zero has no effect
WAC	**Write Anything to Clear:** Writing anything clears the value.
RC	**Read to Clear:** Contents is cleared after the read. Writes have no effect.
LL	**Latch Low:** Clear on read of register.
LH	**Latch High:** Clear on read of register.
SC	**Self-Clearing:** Contents are self-cleared after the being set. Writes of zero have no effect. Contents can be read.
SS	**Self-Setting:** Contents are self-setting after being cleared. Writes of one have no effect. Contents can be read.
RO/LH	**Read Only, Latch High:** Bits with this attribute will stay high until the bit is read. After it is read, the bit will either remain high if the high condition remains, or will go low if the high condition has been removed. If the bit has not been read, the bit will remain high regardless of a change to the high condition. This mode is used in some Ethernet PHY registers.
NASR	**Not Affected by Software Reset.** The state of NASR bits do not change on assertion of a software reset.
RESERVED	**Reserved Field:** Reserved fields must be written with zeros to ensure future compatibility. The value of reserved bits is not guaranteed on a read.

2.0 GENERAL DESCRIPTION

The LAN9250 is a full-featured, single-chip 10/100 Ethernet controller designed for embedded applications where performance, flexibility, ease of integration and system cost control are required. The LAN9250 has been specifically designed to provide high performance and throughput for 16-bit applications. The LAN9250 complies with the IEEE 802.3 (full/half-duplex 10BASE-T and 100BASE-TX) Ethernet protocol, IEEE 802.3az Energy Efficient Ethernet (EEE) (100Mbps only), and the IEEE 1588v2 precision time protocol. 100BASE-FX is supported via an external fiber transceiver.

The LAN9250 includes an integrated Ethernet MAC and PHY with a high-performance SRAM-like slave interface. The integrated checksum offload engines enable the automatic generation of the 16-bit checksum for received and transmitted Ethernet frames, offloading the task from the CPU. The LAN9250 also includes large transmit and receive data FIFOs to accommodate high latency applications. In addition, the LAN9250 memory buffer architecture allows highly efficient use of memory resources by optimizing packet granularity.

The LAN9250 also supports features which reduce or eliminate packet loss. The internal 16-KByte SRAM can hold over 200 received packets. If the receive FIFO gets too full, the LAN9250 can automatically generate flow control packets to the remote node, or assert back-pressure on the remote node by generating network collisions.

Two user selectable host bus interface options are available:

- ***Indexed register access***
  This implementation provides three index/data register banks, each with independent Byte/WORD to DWORD conversion. Internal registers are accessed by first writing one of the three index registers, followed by reading or writing the corresponding data register. Three index/data register banks support up to 3 independent driver threads without access conflicts. Each thread can write its assigned index register without the issue of another thread overwriting it. Two 16-bit cycles or four 8-bit cycles are required within the same 32-bit index/data register - however, these access can be interleaved. Direct (non-indexed) read and write accesses are supported to the packet data FIFOs. The direct FIFO access provides independent Byte/WORD to DWORD conversion, supporting interleaved accesses with the index/data registers. Direct FIFO access also supports burst reading of the data FIFO.
- ***Multiplexed address/data bus***
  This implementation provides a multiplexed address and data bus with both single phase and dual phase address support. The address is loaded with an address strobe followed by data access using a read or write strobe. Two back to back 16-bit data cycles or 4 back to back 8-bit data cycles are required within the same 32-bit DWORD. These accesses must be sequential without any interleaved accesses to other registers. Burst read and write accesses are supported to the packet data and status FIFOs by performing one address cycle followed by multiple read or write data cycles.

The HBI supports 8/16-bit operation with big, little, and mixed endian operations. Four separate FIFO mechanisms (TX/RX Data FIFO's, TX/RX Status FIFO's) interface the HBI to the Host MAC and facilitate the transferring of packet data and status information between the host CPU and the device. A configurable host interrupt pin allows the device to inform the host CPU of any internal interrupts.

An SPI / Quad SPI slave controller provides a low pin count synchronous slave interface that facilitates communication between the device and a host system. The SPI / Quad SPI slave allows access to the System CSRs, internal FIFOs and memories. It supports single and multiple register read and write commands with incrementing, decrementing and static addressing. Single, Dual and Quad bit lanes are supported with a clock rate of up to 80 MHz.

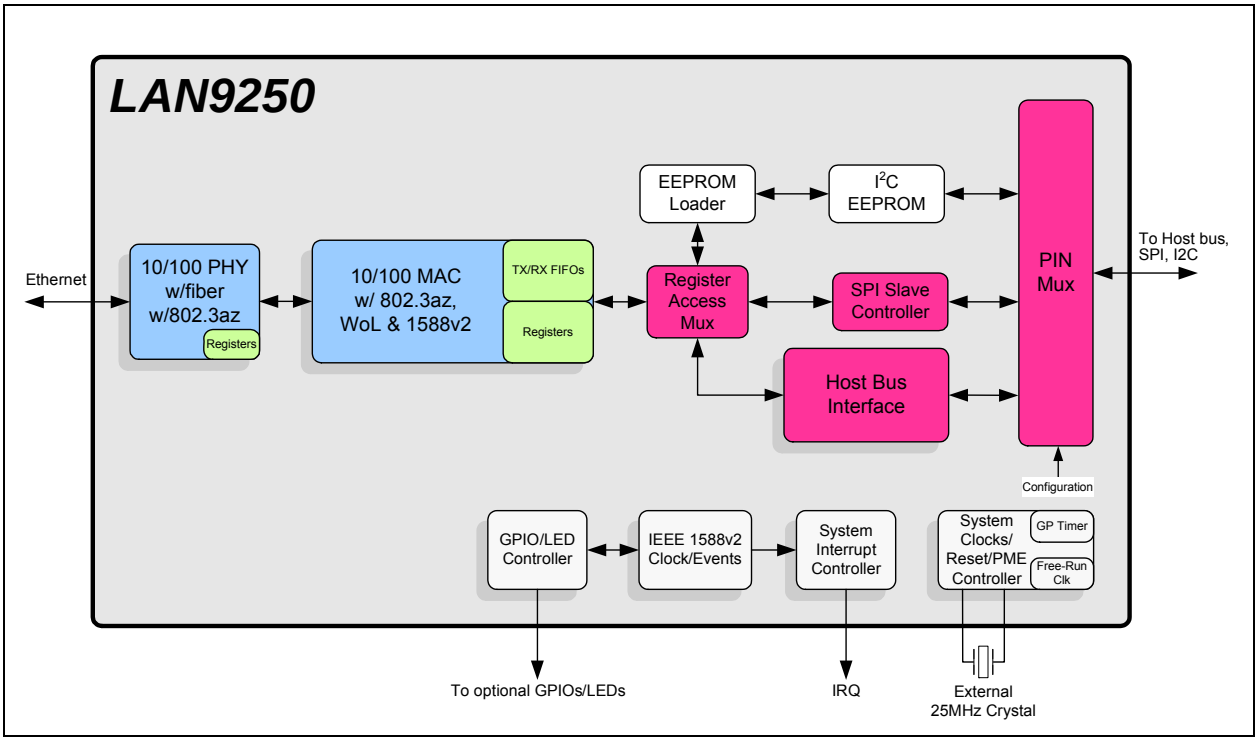
The LAN9250 contains an I^2C master EEPROM controller for connection to an optional EEPROM. This allows for the storage and retrieval of static data. The internal EEPROM Loader can be optionally configured to automatically load stored configuration settings from the EEPROM into the device at reset.

The LAN9250 supports numerous power management and wakeup features. The LAN9250 can be placed in a reduced power mode and can be programmed to issue an external wake signal (PME) via several methods, including "Magic Packet", "Wake on LAN", wake on broadcast, wake on perfect DA, and "Link Status Change". This signal is ideal for triggering system power-up using remote Ethernet wakeup events. The device can be removed from the low power state via a host processor command or one of the wake events.

The LAN9250 can be configured to operate via a single 3.3V supply utilizing an integrated 3.3V to 1.2V linear regulator. The linear regulator may be optionally disabled, allowing usage of a high efficiency external regulator for lower system power dissipation.

The LAN9250 is available in commercial, industrial, and extended industrial temperature ranges. Figure 2-1 provides an internal block diagram of the LAN9250.

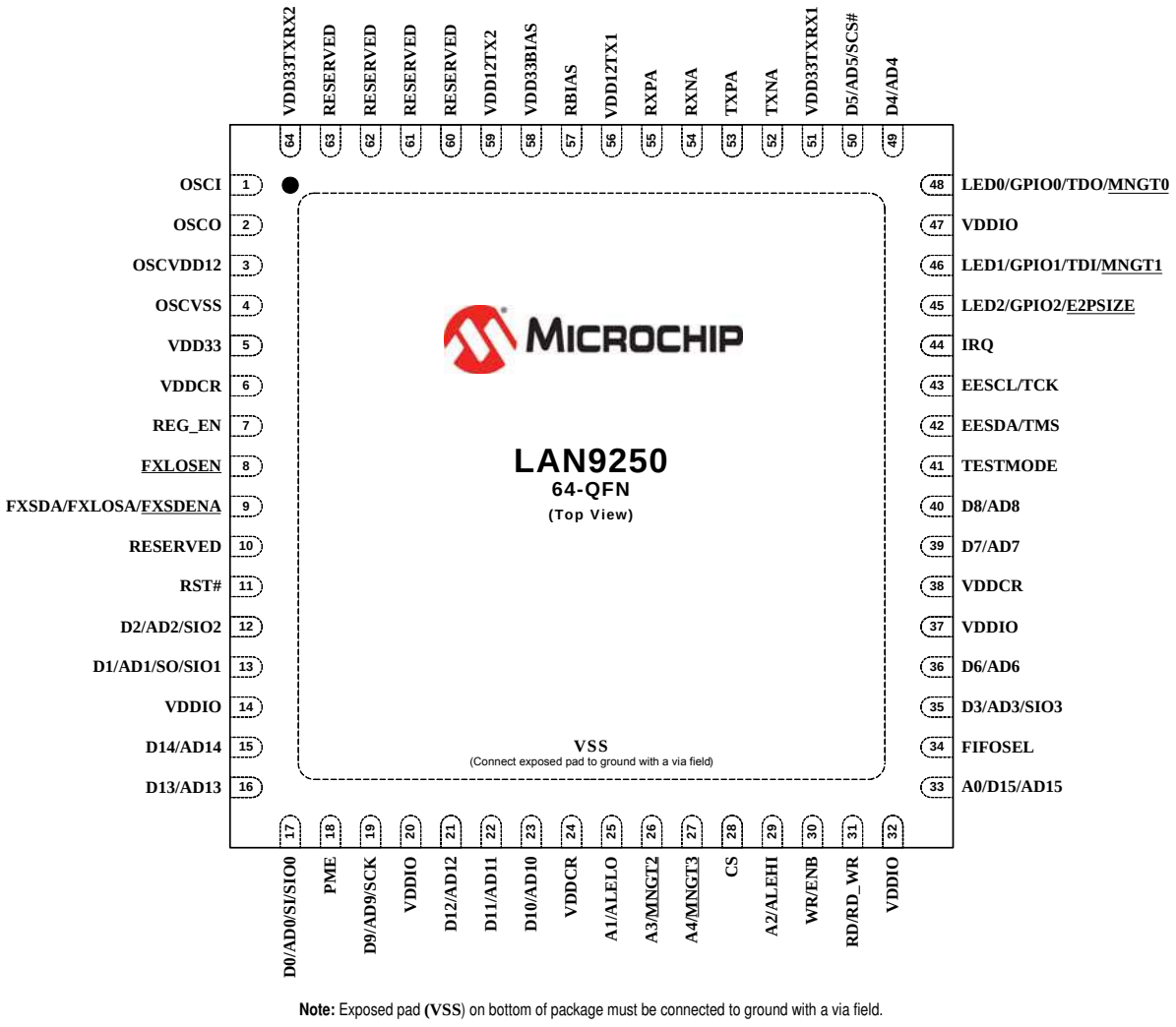
FIGURE 2-1: INTERNAL BLOCK DIAGRAM



3.0 PIN DESCRIPTIONS AND CONFIGURATION

3.1 64-QFN Pin Assignments

FIGURE 3-1: 64-QFN PIN ASSIGNMENTS (TOP VIEW)



Note: Exposed pad **(VSS)** on bottom of package must be connected to ground with a via field.

Note: When a "#" is used at the end of the signal name, it indicates that the signal is active low. For example, **RST#** indicates that the reset signal is active low.

The buffer type for each signal is indicated in the "Buffer Type" column of the pin description tables in Section 3.3, "Pin Descriptions". A description of the buffer types is provided in Section 1.2, "Buffer Types".

Table 3-1 details the 64-QFN package pin assignments in table format. As shown, select pin functions may change based on the device's mode of operation. For modes where a specific pin has no function, the table cell will be marked with "-".

TABLE 3-1: 64-QFN PACKAGE PIN ASSIGNMENTS

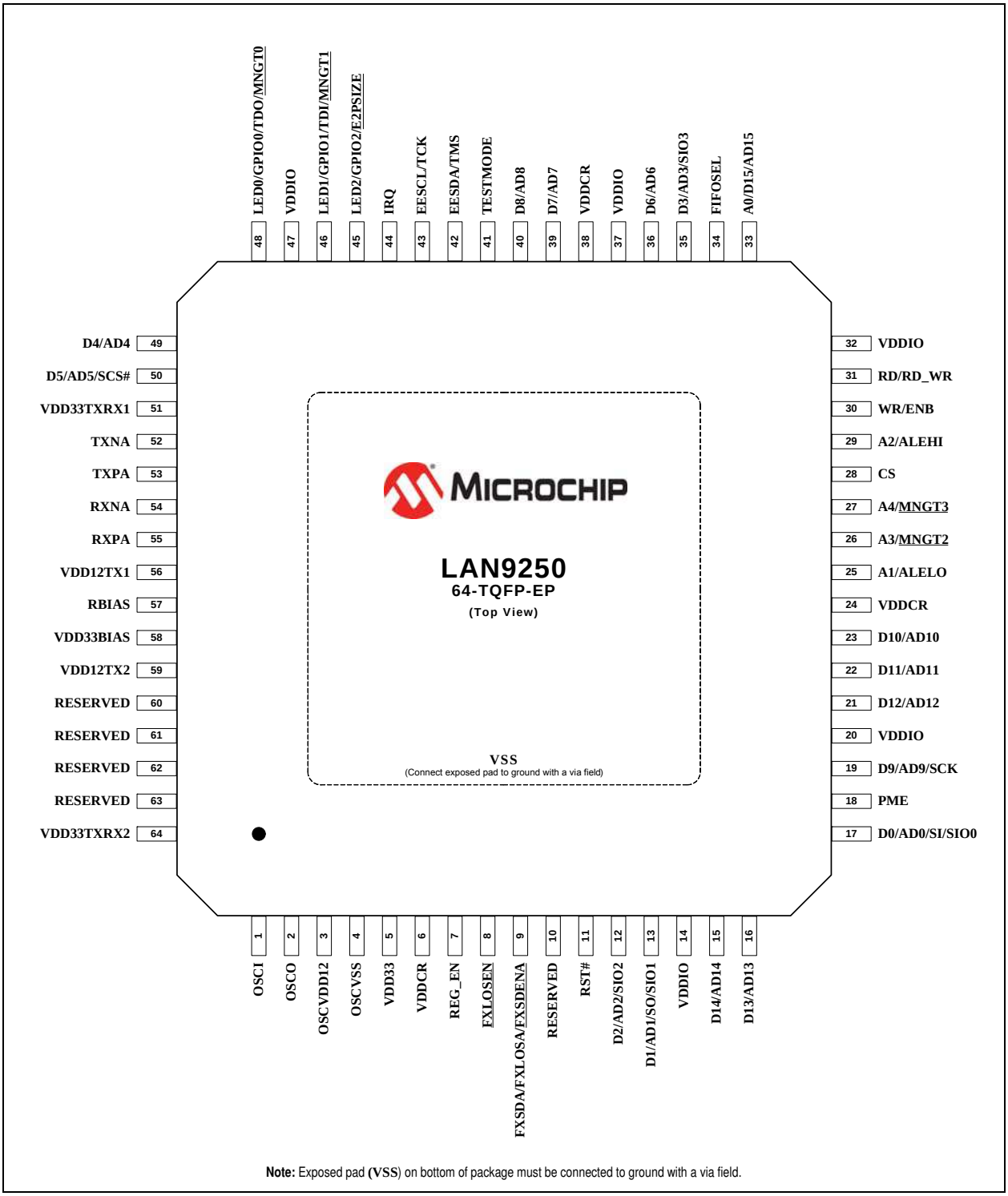
Pin Number	HBI Indexed Mode Pin Name	HBI Multiplexed Mode Pin Name	SPI Mode Pin Name
1	OSCI		
2	OSCO		
3	OSCVDD12		
4	OSCVSS		
5	VDD33		
6	VDDCR		
7	REG_EN		
8	FXLOSEN		
9	FXSDA/FXLOSA/FXSDENA		
10	RESERVED		
11	RST#		
12	D2	AD2	SIO2
13	D1	AD1	SO/SIO1
14	VDDIO		
15	D14	AD14	-
16	D13	AD13	-
17	D0	AD0	SI/SIO0
18	PME		
19	D9	AD9	SCK
20	VDDIO		
21	D12	AD12	-
22	D11	AD11	-
23	D10	AD10	-
24	VDDCR		
25	A1	ALELO	-
26	A3	MNGT2	-
27	A4	MNGT3	-
28	CS		-
29	A2	ALEHI	-
30	WR/ENB		-
31	RD/RD_WR		-
32	VDDIO		

TABLE 3-1: 64-QFN PACKAGE PIN ASSIGNMENTS (CONTINUED)

Pin Number	HBI Indexed Mode Pin Name	HBI Multiplexed Mode Pin Name	SPI Mode Pin Name
33	A0/D15	AD15	-
34	FIFOSEL	-	
35	D3	AD3	SIO3
36	D6	AD6	-
37	VDDIO		
38	VDDCR		
39	D7	AD7	-
40	D8	AD8	-
41	TESTMODE		
42	EESDA/TMS		
43	EESCL/TCK		
44	IRQ		
45	LED2/GPIO2/E2PSIZE		
46	LED1/GPIO1/TDI/MNGT1		
47	VDDIO		
48	LED0/GPIO0/TDO/MNGT0		
49	D4	AD4	-
50	D5	AD5	SCS#
51	VDD33TXRX1		
52	TXNA		
53	TXPA		
54	RXNA		
55	RXPA		
56	VDD12TX1		
57	RBIAS		
58	VDD33BIAS		
59	VDD12TX2		
60	RESERVED		
61	RESERVED		
62	RESERVED		
63	RESERVED		
64	VDD33TXRX2		
Exposed Pad	VSS		

3.2 64-TQFP-EP Pin Assignments

FIGURE 3-2: 64-TQFP-EP PIN ASSIGNMENTS (TOP VIEW)



Note: Exposed pad (**VSS**) on bottom of package must be connected to ground with a via field.

Note: When a "#" is used at the end of the signal name, it indicates that the signal is active low. For example, **RST#** indicates that the reset signal is active low.

The buffer type for each signal is indicated in the "Buffer Type" column of the pin description tables in Section 3.3, "Pin Descriptions". A description of the buffer types is provided in Section 1.2, "Buffer Types".

Table 3-2 details the 64-TQFP-EP package pin assignments in table format. As shown, select pin functions may change based on the device's mode of operation. For modes where a specific pin has no function, the table cell will be marked with "-".

TABLE 3-2: 64-TQFP-EP PACKAGE PIN ASSIGNMENTS

Pin Number	HBI Indexed Mode Pin Name	HBI Multiplexed Mode Pin Name	SPI Mode Pin Name
1	OSCI		
2	OSCO		
3	OSCVDD12		
4	OSCVSS		
5	VDD33		
6	VDDCR		
7	REG_EN		
8	FXLOSEN		
9	FXSDA/FXLOSA/FXSDENA		
10	RESERVED		
11	RST#		
12	D2	AD2	SIO2
13	D1	AD1	SO/SIO1
14	VDDIO		
15	D14	AD14	-
16	D13	AD13	-
17	D0	AD0	SI/SIO0
18	PME		
19	D9	AD9	SCK
20	VDDIO		
21	D12	AD12	-
22	D11	AD11	-
23	D10	AD10	-
24	VDDCR		
25	A1	ALELO	-
26	A3	MNGT2	-
27	A4	MNGT3	-
28	CS		-

TABLE 3-2: 64-TQFP-EP PACKAGE PIN ASSIGNMENTS (CONTINUED)

Pin Number	HBI Indexed Mode Pin Name	HBI Multiplexed Mode Pin Name	SPI Mode Pin Name
29	A2	ALEHI	-
30	WR/ENB		-
31	RD/RD_WR		-
32	VDDIO		
33	A0/D15	AD15	-
34	FIFOSEL	-	
35	D3	AD3	SIO3
36	D6	AD6	-
37	VDDIO		
38	VDDCR		
39	D7	AD7	-
40	D8	AD8	-
41	TESTMODE		
42	EESDA/TMS		
43	EESCL/TCK		
44	IRQ		
45	LED2/GPIO2/E2PSIZE		
46	LED1/GPIO1/TDI/MNGT1		
47	VDDIO		
48	LED0/GPIO0/TDO/MNGT0		
49	D4	AD4	-
50	D5	AD5	SCS#
51	VDD33TXRX1		
52	TXNA		
53	TXPA		
54	RXNA		
55	RXPA		
56	VDD12TX1		
57	RBIAS		
58	VDD33BIAS		
59	VDD12TX2		
60	RESERVED		
61	RESERVED		
62	RESERVED		
63	RESERVED		

TABLE 3-2: 64-TQFP-EP PACKAGE PIN ASSIGNMENTS (CONTINUED)

Pin Number	HBI Indexed Mode Pin Name	HBI Multiplexed Mode Pin Name	SPI Mode Pin Name
64	VDD33TXRX2		
Exposed Pad	VSS		

3.3 Pin Descriptions

This section contains descriptions of the various LAN9250 pins. The pin descriptions have been broken into functional groups as follows:

- LAN Pin Descriptions
- Host Bus Pin Descriptions
- SPI/SQI Pin Descriptions
- EEPROM Pin Descriptions
- GPIO, LED & Configuration Strap Pin Descriptions
- Miscellaneous Pin Descriptions
- JTAG Pin Descriptions
- Core and I/O Power Pin Descriptions

TABLE 3-3: LAN PIN DESCRIPTIONS

Num Pins	Name	Symbol	Buffer Type	Description
1	TP TX/RX Positive Channel 1	TXPA	AIO	Twisted Pair Transmit/Receive Positive Channel 1. See Note 1
	FX TX Positive		OLVPECL	Fiber Transmit Positive.
1	TP TX/RX Negative Channel 1	TXNA	AIO	Twisted Pair Transmit/Receive Negative Channel 1. See Note 1.
	FX TX Negative		OLVPECL	Fiber Transmit Negative.
1	TP TX/RX Positive Channel 2	RXPA	AIO	Twisted Pair Transmit/Receive Positive Channel 2. See Note 1.
	FX RX Positive		AI	Fiber Receive Positive.
1	TP TX/RX Negative Channel 2	RXNA	AIO	Twisted Pair Transmit/Receive Negative Channel 2. See Note 1.
	FX RX Negative		AI	Fiber Receive Negative.

TABLE 3-3: LAN PIN DESCRIPTIONS (CONTINUED)

Num Pins	Name	Symbol	Buffer Type	Description
1	FX Signal Detect (SD)	**FXSDA**	ILVPECL	Fiber Signal Detect. When FX-LOS mode is not selected, this pin functions as the Signal Detect input from the external transceiver. A level above 2 V (typ.) indicates valid signal. When FX-LOS mode is selected, the input buffer is disabled.
	FX Loss Of Signal (LOS)	**FXLOSA**	IS (PU)	Fiber Loss of Signal. When FX-LOS mode is selected (via fx_los_strap_1), this pin functions as the Loss of Signal input from the external transceiver. A high indicates LOS while a low indicates valid signal. When FX-LOS mode is not selected, the input buffer and pull-up are disabled.
	FX-SD Enable Strap	**FXSDENA**	AI	FX-SD Enable. When FX-LOS mode is not selected, this strap input selects between FX-SD and copper twisted pair mode. A level above 1 V (typ.) selects FX-SD. When FX-LOS mode is selected, the input buffer is disabled. See Note 2.
1	Bias Reference	**RBIAS**	AI	Used for internal bias circuits. Connect to an external 12.1 kΩ, 1% resistor to ground. Refer to the device reference schematic for connection information. **Note:** The nominal voltage is 1.2 V and the resistor will dissipate approximately 1 mW of power.
1	FX-LOS Enable Strap	**FXLOSEN**	AI	FX-LOS Enable. This strap input selects between FX-LOS and FX-SD / copper twisted pair mode. A level below 1 V (typ.) selects FX-SD / copper twisted pair for the port, further determined by FXSDENA. A level of 1.5 V (typ.) or above selects FX-LOS for the port. See Note 2.
1	+3.3 V Analog Power Supply	**VDD33TXRX1**	P	See Note 3.
1	+3.3 V Analog Power Supply	**VDD33TXRX2**	P	See Note 3.
1	+3.3 V Master Bias Power Supply	**VDD33BIAS**	P	See Note 3.

TABLE 3-3: LAN PIN DESCRIPTIONS (CONTINUED)

Num Pins	Name	Symbol	Buffer Type	Description
1	Transmitter +1.2 V Power Supply	**VDD12TX1**	P	This pin is supplied from either an external 1.2 V supply or from the device's internal regulator via the PCB. This pin must be tied to the **VDD12TX2** pin for proper operation. See Note 3.
1	Transmitter +1.2 V Power Supply	**VDD12TX2**	P	This pin is supplied from either an external 1.2 V supply or from the device's internal regulator via the PCB. This pin must be tied to the **VDD12TX1** pin for proper operation. See Note 3.

Note 1: In copper mode, either channel 1 or 2 may function as the transmit pair while the other channel functions as the receive pair. The pin name symbols for the twisted pair pins apply to a normal connection. If HP Auto-MDIX is enabled and a reverse connection is detected or manually selected, the RX and TX pins will be swapped internally.

Note 2: Configuration strap pins are identified by an underlined symbol name. Configuration strap values are latched on power-on reset or **RST#** de-assertion. Refer to Section 7.0, "Configuration Straps," on page 54 for more information.

Note 3: Refer to Section 4.0, "Power Connections," on page 26, the device reference schematics, and the device LANCheck schematic checklist for additional connection information.

TABLE 3-4: HOST BUS PIN DESCRIPTIONS

Num Pins	Name	Symbol	Buffer Type	Description
1	Read	**RD**	VIS	This pin is the host bus read strobe. Normally active low, the polarity can be changed via the HBI_rd_rdwr_polarity_strap.
	Read or Write	**RD_WR**	VIS	This pin is the host bus direction control. Used in conjunction with the ENB pin, it indicates a read or write operation. The normal polarity is read when 1, write when 0 (R/nW) but can be changed via the HBI_rd_rdwr_polarity_strap.
1	Write	**WR**	VIS	This pin is the host bus write strobe. Normally active low, the polarity can be changed via the HBI_wr_en_polarity_strap.
	Enable	**ENB**	VIS	This pin is the host bus data enable strobe. Used in conjunction with the RD_WR pin it indicates the data phase of the operation. Normally active low, the polarity can be changed via the HBI_wr_en_polarity_strap.

TABLE 3-4: HOST BUS PIN DESCRIPTIONS (CONTINUED)

Num Pins	Name	Symbol	Buffer Type	Description
1	Chip Select	**CS**	VIS	This pin is the host bus chip select and indicates that the device is selected for the current transfer. Normally active low, the polarity can be changed via the HBI_cs_polarity_strap.
1	FIFO Select	**FIFOSEL**	VIS	This input directly selects the Host MAC TX and RX Data FIFOs for non-multiplexed address mode.
5	Address	**A[4:0]**	VIS	These pins provide the address for non-multiplexed address mode. In 16-bit data mode, bit 0 is not used.
16	Data	**D[15:0]**	VIS/VO8	These pins are the host bus data bus for non-multiplexed address mode. In 8-bit data mode, bits 15-8 are not used and their input and output drivers are disabled.
	Address & Data	**AD[15:0]**	VIS/VO8	These pins are the host bus address / data bus for multiplexed address mode. Bits 15-8 provide the upper byte of address for single phase multiplexed address mode. Bits 7-0 provide the lower byte of address for single phase multiplexed address mode and both bytes of address for dual phase multiplexed address mode. In 8-bit data dual phase multiplexed address mode, bits 15-8 are not used and their input and output drivers are disabled.
1	Address Latch Enable High	**ALEHI**	VIS	This pin indicates the address phase for multiplexed address modes. It is used to load the higher address byte in dual phase multiplexed address mode. Normally active low (address saved on rising edge), the polarity can be changed via the HBI_ale_polarity_strap.
1	Address Latch Enable Low	**ALELO**	VIS	This pin indicates the address phase for multiplexed address modes. It is used to load both address bytes in single phase multiplexed address mode and the lower address byte in dual phase multiplexed address mode. Normally active low (address saved on rising edge), the polarity can be changed via the HBI_ale_polarity_strap.

TABLE 3-5: SPI/SQI PIN DESCRIPTIONS

Num Pins	Name	Symbol	Buffer Type	Description
1	SPI/SQI Slave Chip Select	**SCS#**	VIS (PU)	This pin is the SPI/SQI slave chip select input. When low, the SPI/SQI slave is selected for SPI/SQI transfers. When high, the SPI/SQI serial data output(s) is(are) 3-stated.
1	SPI/SQI Slave Serial Clock	**SCK**	VIS (PU)	This pin is the SPI/SQI slave serial clock input.
4	SPI/SQI Slave Serial Data Input/Output	**SIO[3:0]**	VIS/VO8 (PU)	These pins are the SPI/SQI slave data input and output for multiple bit I/O.
	SPI Slave Serial Data Input	**SI**	VIS (PU)	This pin is the SPI slave serial data input. **SI** is shared with the **SIO0** pin.
	SPI Slave Serial Data Output	**SO**	VO8 (PU) Note 4	This pin is the SPI slave serial data output. **SO** is shared with the **SIO1** pin.

Note 4: Although this pin is an output for SPI instructions, it includes a pull-up since it is also SIO bit 1.

TABLE 3-6: EEPROM PIN DESCRIPTIONS

Num Pins	Name	Symbol	Buffer Type	Description
1	EEPROM I^2C Serial Data Input/Output	**EESDA**	VIS/VOD8	When the device is accessing an external EEPROM this pin is the I^2C serial data input/open-drain output. **Note:** This pin must be pulled-up by an external resistor at all times.
1	EEPROM I^2C Serial Clock	**EESCL**	VIS/VOD8	When the device is accessing an external EEPROM this pin is the I^2C clock input/open-drain output. **Note:** This pin must be pulled-up by an external resistor at all times.

TABLE 3-7: GPIO, LED & CONFIGURATION STRAP PIN DESCRIPTIONS

Num Pins	Name	Symbol	Buffer Type	Description
1	LED 2	**LED2**	VO12/ VOD12/ VOS12	This pin is configured to operate as an LED when the LED 2 Enable bit of the LED Configuration Register (LED_CFG) is set. The buffer type depends on the setting of the LED Function 2-0 (LED_FUN[2:0]) field in the LED Configuration Register (LED_CFG) and is configured to be either a push-pull or open-drain/open-source output. When selected as an open-drain/open-source output, the polarity of this pin depends upon the **E2PSIZE** strap value sampled at reset. **Note:** Refer to Section 16.3, "LED Operation," on page 385 to additional information.
	General Purpose I/O 2	**GPIO2**	VIS/VO12/ VOD12 (PU)	This pin is configured to operate as a GPIO when the LED 2 Enable bit of the LED Configuration Register (LED_CFG) is clear. The pin is fully programmable as either a push-pull output, an open-drain output or a Schmitt-triggered input by writing the General Purpose I/O Configuration Register (GPIO_CFG) and the General Purpose I/O Data & Direction Register (GPIO_DATA_DIR).
	EEPROM Size Configuration Strap	**E2PSIZE**	VIS (PU)	This strap configures the value of the EEPROM size hard-strap. See Note 5. A low selects 1K bits (128 x 8) through 16K bits (2K x 8). A high selects 32K bits (4K x 8) through 512K bits (64K x 8).
1	LED 1	**LED1**	VO12/ VOD12/ VOS12	This pin is configured to operate as an LED when the LED 1 Enable bit of the LED Configuration Register (LED_CFG) is set. The buffer type depends on the setting of the LED Function 2-0 (LED_FUN[2:0]) field in the LED Configuration Register (LED_CFG) and is configured to be either a push-pull or open-drain/open-source output. When selected as an open-drain/open-source output, the polarity of this pin depends upon the **MNGT1** strap value sampled at reset. **Note:** Refer to Section 16.3, "LED Operation," on page 385 to additional information.
	General Purpose I/O 1	**GPIO1**	VIS/VO12/ VOD12 (PU)	This pin is configured to operate as a GPIO when the LED 1 Enable bit of the LED Configuration Register (LED_CFG) is clear. The pin is fully programmable as either a push-pull output, an open-drain output or a Schmitt-triggered input by writing the General Purpose I/O Configuration Register (GPIO_CFG) and the General Purpose I/O Data & Direction Register (GPIO_DATA_DIR).
	Host Interface Configuration Strap 1	**MNGT1**	VIS (PU)	This strap, along with **MNGT0**, **MNGT2**, and **MNGT3** configures the host interface mode. See Note 5. See Table 7-3, "HBI Strap Mapping," on page 61 for the host interface strap settings.

TABLE 3-7: GPIO, LED & CONFIGURATION STRAP PIN DESCRIPTIONS (CONTINUED)

Num Pins	Name	Symbol	Buffer Type	Description
1	LED 0	**LED0**	VO12/ VOD12/ VOS12	This pin is configured to operate as an LED when the LED 0 Enable bit of the LED Configuration Register (LED_CFG) is set. The buffer type depends on the setting of the LED Function 2-0 (LED_FUN[2:0]) field in the LED Configuration Register (LED_CFG) and is configured to be either a push-pull or open-drain/open-source output. When selected as an open-drain/open-source output, the polarity of this pin depends upon the **MNGT0** strap value sampled at reset. **Note:** Refer to Section 16.3, "LED Operation," on page 385 to additional information.
	General Purpose I/O 0	**GPIO0**	VIS/VO12/ VOD12 (PU)	This pin is configured to operate as a GPIO when the LED 0 Enable bit of the LED Configuration Register (LED_CFG) is clear. The pin is fully programmable as either a push-pull output, an open-drain output or a Schmitt-triggered input by writing the General Purpose I/O Configuration Register (GPIO_CFG) and the General Purpose I/O Data & Direction Register (GPIO_DATA_DIR).
	Host Interface Configuration Strap 0	**MNGT0**	VIS (PU)	This strap, along with **MNGT1**, **MNGT2**, and **MNGT3** configures the host mode. See Note 5. See Table 7-3, “HBI Strap Mapping,” on page 61 for the host interface strap settings.
1	Host Interface Configuration Strap 3	**MNGT3**	VIS (PU)	This strap, along with **MNGT0**, **MNGT1**, and **MNGT2** configures the host mode. See Note 5. See Table 7-3, “HBI Strap Mapping,” on page 61 for the host interface strap settings.
1	Host Interface Configuration Strap 2	**MNGT2**	VIS (PU)	This strap, along with **MNGT0**, **MNGT1**, and **MNGT3** configures the host mode. See Note 5. See Table 7-3, “HBI Strap Mapping,” on page 61 for the host interface strap settings.

Note 5: Configuration strap pins are identified by an underlined symbol name. Configuration strap values are latched on power-on reset or **RST#** de-assertion. Refer to Section 7.0, "Configuration Straps," on page 54 for more information.

TABLE 3-8: MISCELLANEOUS PIN DESCRIPTIONS

Num Pins	Name	Symbol	Buffer Type	Description
1	Power Management Event Output	**PME**	VO8/VOD8	When programmed accordingly this signal is asserted upon detection of a wakeup event. The polarity and buffer type of this signal is programmable via the PME Enable (PME_EN) bit of the Power Management Control Register (PMT_CTRL). Refer to Section 6.0, "Clocks, Resets, and Power Management," on page 37 for additional information on the power management features.
1	Interrupt Output	**IRQ**	VO8/VOD8	Interrupt request output. The polarity, source and buffer type of this signal is programmable via the Interrupt Configuration Register (IRQ_CFG). For more information, refer to Section 8.0, "System Interrupts," on page 62.
1	System Reset Input	**RST#**	VIS (PU)	As an input, this active low signal allows external hardware to reset the device. The device also contains an internal power-on reset circuit. Thus this signal may be left unconnected if an external hardware reset is not needed. When used this signal must adhere to the reset timing requirements as detailed in the Section 19.0, "Operational Characteristics," on page 399.
1	Regulator Enable	**REG_EN**	AI	When tied to 3.3 V, the internal 1.2 V regulators are enabled.
1	Test Mode	**TESTMODE**	VIS (PD)	This pin must be tied to **VSS** for proper operation.
1	Crystal Input	**OSCI**	ICLK	External 25 MHz crystal input. This signal can also be driven by a single-ended clock oscillator. When this method is used, **OSCO** should be left unconnected.
1	Crystal Output	**OSCO**	OCLK	External 25 MHz crystal output.
1	Crystal +1.2 V Power Supply	**OSCVDD12**	P	Supplied by the on-chip regulator unless configured for regulator off mode via **REG_EN**.
1	Crystal Ground	**OSCVSS**	P	Crystal ground.
5	Reserved	**RESERVED**	-	This pin is reserved and must be left unconnected for proper operation.

TABLE 3-9: JTAG PIN DESCRIPTIONS

Num Pins	Name	Symbol	Buffer Type	Description
1	JTAG Test Mux Select	**TMS**	VIS	JTAG test mode select
1	JTAG Test Clock	**TCK**	VIS	JTAG test clock
1	JTAG Test Data Input	**TDI**	VIS	JTAG data input
1	JTAG Test Data Output	**TDO**	VO12	JTAG data output

TABLE 3-10: CORE AND I/O POWER PIN DESCRIPTIONS

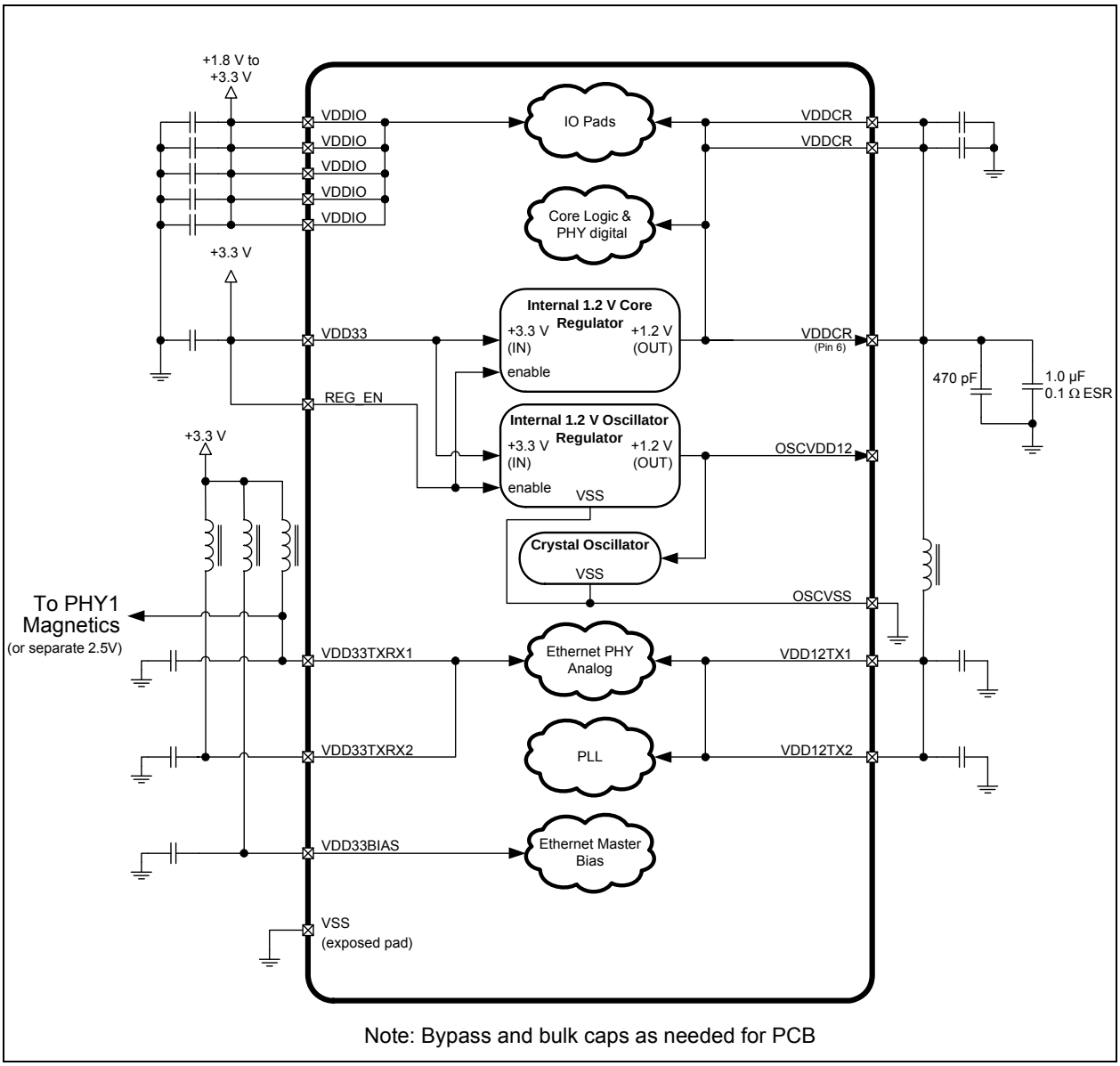
Num Pins	Name	Symbol	Buffer Type	Description
1	Regulator +3.3 V Power Supply	**VDD33**	P	+3.3 V power supply for internal regulators. See Note 6. **Note:** +3.3 V must be supplied to this pin even if the internal regulators are disabled.
5	+1.8 V to +3.3 V Variable I/O Power	**VDDIO**	P	+1.8 V to +3.3 V variable I/O power. See Note 6.
3	+1.2 V Digital Core Power Supply	**VDDCR**	P	Supplied by the on-chip regulator unless configured for regulator off mode via REG_EN. 1 µF and 470 pF decoupling capacitors in parallel to ground should be used on pin 6. See Note 6.
1 pad	Ground	**VSS**	P	Common ground. This exposed pad must be connected to the ground plane with a via array.

Note 6: Refer to Section 4.0, "Power Connections," on page 26, the device reference schematic, and the device LANCheck schematic checklist for additional connection information.

4.0 POWER CONNECTIONS

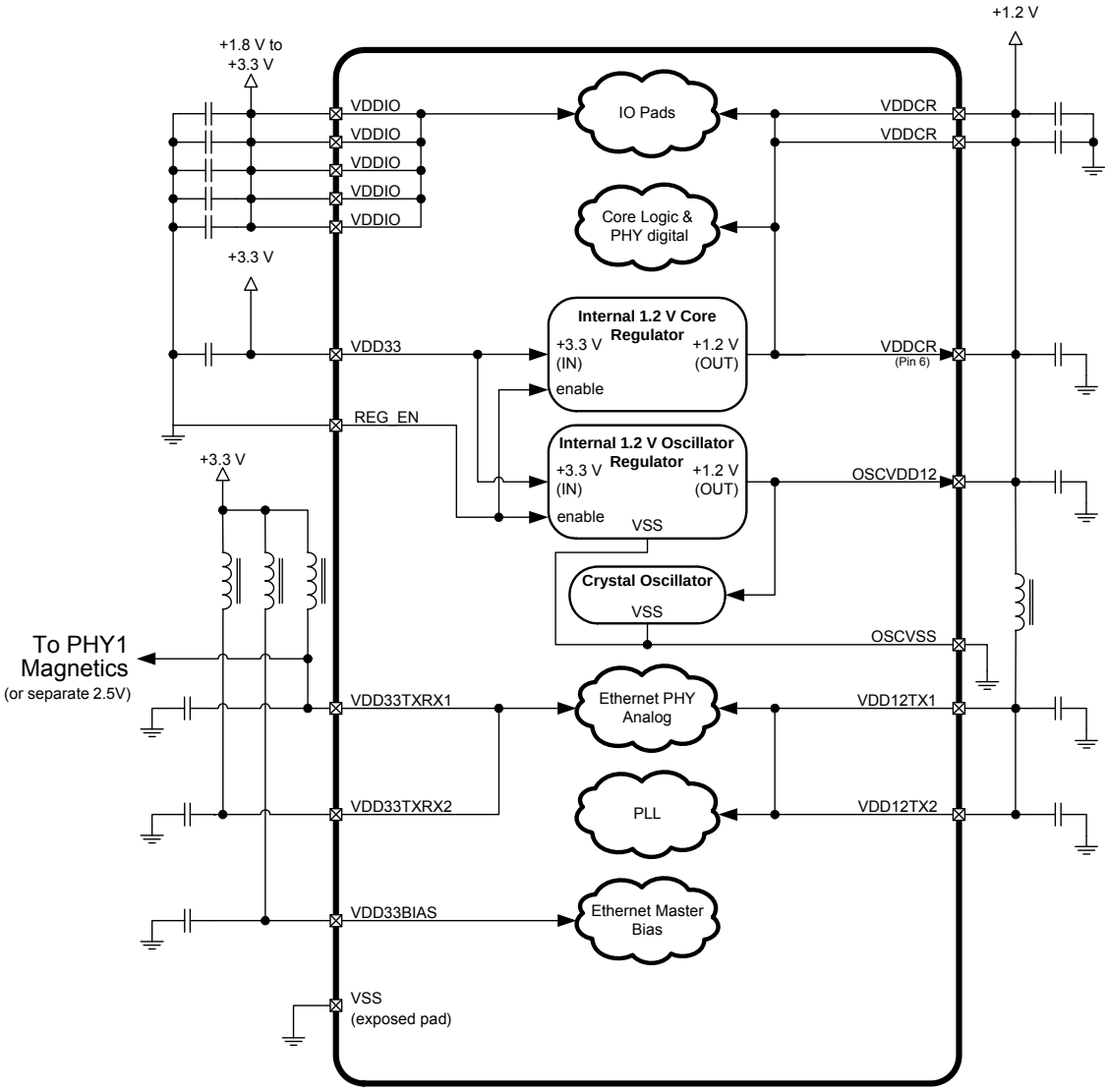
Figure 4-1 and Figure 4-2 illustrate the device power connections for regulator enabled and disabled cases, respectively. Refer to the device reference schematic and the device LANCheck schematic checklist for additional information. Section 4.1 provides additional information on the devices internal voltage regulators.

FIGURE 4-1: POWER CONNECTIONS - REGULATORS ENABLED



Note: Bypass and bulk caps as needed for PCB

FIGURE 4-2: POWER CONNECTIONS - REGULATORS DISABLED



Note: Bypass and bulk caps as needed for PCB

4.1 Internal Voltage Regulators

The device contains two internal 1.2 V regulators:

- 1.2 V Core Regulator
- 1.2 V Crystal Oscillator Regulator

4.1.1 1.2 V CORE REGULATOR

The core regulator supplies 1.2 V volts to the main core digital logic, the I/O pads, and the PHY's digital logic and can be used to supply the 1.2 V power to the PHY analog sections (via an external connection).

When the **REG_EN** input pin is connected to 3.3 V, the core regulator is enabled and receives 3.3 V on the **VDD33** pin. A 1.0 uF 0.1 Ω ESR capacitor must be connected to the **VDDCR** pin associated with the regulator.

When the **REG_EN** input pin is connected to **VSS**, the core regulator is disabled. However, 3.3 V must still be supplied to the **VDD33** pin. The 1.2 V core voltage must then be externally input into the **VDDCR** pins.

4.1.2 1.2 V CRYSTAL OSCILLATOR REGULATOR

The crystal oscillator regulator supplies 1.2 V volts to the crystal oscillator. When the **REG_EN** input pin is connected to 3.3 V, the crystal oscillator regulator is enabled and receives 3.3 V on the **VDD33** pin. An external capacitor is not required.

When the **REG_EN** input pin is connected to **VSS**, the crystal oscillator regulator is disabled. However, 3.3 V must still be supplied to the **VDD33** pin. The 1.2 V crystal oscillator voltage must then be externally input into the **OSCVDD12** pin.

5.0 REGISTER MAP

This chapter details the device register map and summarizes the various directly addressable System Control and Status Registers (CSRs). Detailed descriptions of the System CSRs are provided in the chapters corresponding to their function. Additional indirectly addressable registers are available in the various sub-blocks of the device. These registers are also detailed in their corresponding chapters.

Directly Addressable Registers

- Section 11.10.1, "TX/RX FIFOs," on page 157
- Section 5.1, "System Control and Status Registers," on page 31

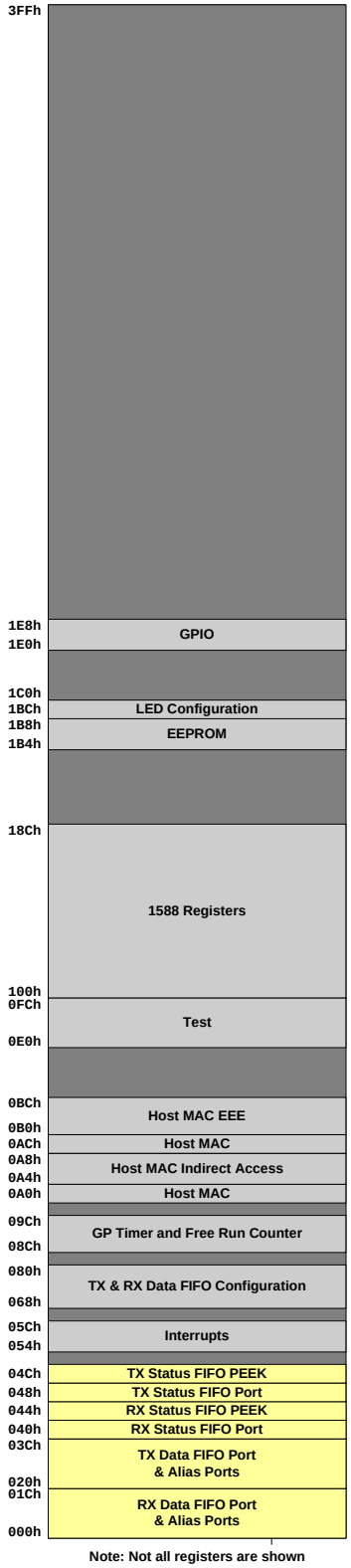
Indirectly Addressable Registers

- Section 11.15, "Host MAC Control and Status Registers," on page 192
- Section 12.2.18, "PHY Registers," on page 230

Figure 5-1 contains an overall base register memory map of the device. This memory map is not drawn to scale, and should be used for general reference only. Table 5-1 provides a summary of all directly addressable CSRs and their corresponding addresses.

Note:	Register bit type definitions are provided in Section 1.3, "Register Nomenclature," on page 7. Not all device registers are memory mapped or directly addressable. For details on the accessibility of the various device registers, refer the register sub-sections listed above.

FIGURE 5-1: REGISTER ADDRESS MAP



Note: Not all registers are shown

5.1 System Control and Status Registers

The System CSRs are directly addressable memory mapped registers with a base address offset range of 050h to 1F8h. These registers are addressable by the Host via the Host Bus Interface (HBI) or SPI/SQI. For more information on the various device modes and their corresponding address configurations, see Section 2.0, "General Description," on page 8.

Table 5-1 lists the System CSRs and their corresponding addresses in order. All system CSRs are reset to their default value on the assertion of a chip-level reset.

The System CSRs can be divided into the following sub-categories. Each of these sub-categories is located in the corresponding chapter and contains the System CSR descriptions of the associated registers. The register descriptions are categorized as follows:

- Section 6.2.3, "Reset Registers," on page 42
- Section 6.3.5, "Power Management Registers," on page 49
- Section 8.3, "Interrupt Registers," on page 65
- Section 11.14, "Host MAC & FIFO Interface Registers," on page 176
- Section 16.4, "GPIO/LED Registers," on page 386
- Section 13.5, "I2C Master EEPROM Controller Registers," on page 294
- Section 14.8, "1588 Registers," on page 316
- Section 17.1, "Miscellaneous System Configuration & Status Registers," on page 392

Note: Unlisted registers are reserved for future use.

TABLE 5-1: SYSTEM CONTROL AND STATUS REGISTERS

Address	Register Name (Symbol)
000h - 04Ch	TX/RX FIFOs
050h	Chip ID and Revision (ID_REV)
054h	Interrupt Configuration Register (IRQ_CFG)
058h	Interrupt Status Register (INT_STS)
05Ch	Interrupt Enable Register (INT_EN)
064h	Byte Order Test Register (BYTE_TEST)
068h	FIFO Level Interrupt Register (FIFO_INT)
06Ch	Receive Configuration Register (RX_CFG)
070h	Transmit Configuration Register (TX_CFG)
074h	Hardware Configuration Register (HW_CFG)
078h	Receive Datapath Control Register (RX_DP_CTRL)
07Ch	RX FIFO Information Register (RX_FIFO_INF)
080h	TX FIFO Information Register (TX_FIFO_INF)
084h	Power Management Control Register (PMT_CTRL)
08Ch	General Purpose Timer Configuration Register (GPT_CFG)
090h	General Purpose Timer Count Register (GPT_CNT)
09Ch	Free Running 25MHz Counter Register (FREE_RUN)
0A0h	Host MAC RX Dropped Frames Counter Register (RX_DROP)
0A4h	Host MAC CSR Interface Command Register (MAC_CSR_CMD)
0A8h	Host MAC CSR Interface Data Register (MAC_CSR_DATA)
0ACh	Host MAC Automatic Flow Control Configuration Register (AFC_CFG)
0B0h	Host MAC RX LPI Transitions Register (HMAC_RX_LPI_TRANSITION)
0B4h	Host MAC RX LPI Time Register (HMAC_RX_LPI_TIME)
0B8h	Host MAC TX LPI Transitions Register (HMAC_TX_LPI_TRANSITION)
0BCh	Host MAC TX LPI Time Register (HMAC_TX_LPI_TIME)

TABLE 5-1: SYSTEM CONTROL AND STATUS REGISTERS (CONTINUED)

Address	Register Name (Symbol)
1588 Registers	
100h	1588 Command and Control Register (1588_CMD_CTL)
104h	1588 General Configuration Register (1588_GENERAL_CONFIG)
108h	1588 Interrupt Status Register (1588_INT_STS)
10Ch	1588 Interrupt Enable Register (1588_INT_EN)
110h	1588 Clock Seconds Register (1588_CLOCK_SEC)
114h	1588 Clock NanoSeconds Register (1588_CLOCK_NS)
118h	1588 Clock Sub-NanoSeconds Register (1588_CLOCK_SUBNS)
11Ch	1588 Clock Rate Adjustment Register (1588_CLOCK_RATE_ADJ)
120h	1588 Clock Temporary Rate Adjustment Register (1588_CLOCK_TEMP_RATE_ADJ)
124h	1588 Clock Temporary Rate Duration Register (1588_CLOCK_TEMP_RATE_DURATION)
128h	1588 Clock Step Adjustment Register (1588_CLOCK_STEP_ADJ)
12Ch	1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) x=A
130h	1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) x=A
134h	1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RELOAD_SEC_x) x=A
138h	1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TARGET_RELOAD_NS_x) x=A
13Ch	1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) x=B
140h	1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) x=B
144h	1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RELOAD_SEC_x) x=B
148h	1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TARGET_RELOAD_NS_x) x=B
14Ch	1588 User MAC Address High-WORD Register (1588_USER_MAC_HI)
150h	1588 User MAC Address Low-DWORD Register (1588_USER_MAC_LO)
154h	1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL)
158h	1588 Port Latency Register (1588_LATENCY)
158h	1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG)
158h	1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG)
15Ch	1588 Port Asymmetry and Peer Delay Register (1588_ASYM_PEERDLY)
15Ch	1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG)
15Ch	1588 Port TX Timestamp Configuration Register (1588_TX_TIMESTAMP_CONFIG)
15Ch	1588 GPIO Capture Configuration Register (1588_GPIO_CAP_CONFIG)
160h	1588 Port Capture Information Register (1588_CAP_INFO)
160h	1588 Port RX Timestamp Insertion Configuration Register (1588_RX_TS_INSERT_CONFIG)
164h	1588 Port TX Modification Register (1588_TX_MOD)
168h	1588 Port RX Filter Configuration Register (1588_RX_FILTER_CONFIG)
168h	1588 Port TX Modification Register 2 (1588_TX_MOD2)
16Ch	1588 Port RX Ingress Time Seconds Register (1588_RX_INGRESS_SEC)
16Ch	1588 Port TX Egress Time Seconds Register (1588_TX_EGRESS_SEC)
16Ch	1588 GPIO x Rising Edge Clock Seconds Capture Register (1588_GPIO_RE_CLOCK_SEC_CAP_x)
170h	1588 Port RX Ingress Time NanoSeconds Register (1588_RX_INGRESS_NS)
170h	1588 Port TX Egress Time NanoSeconds Register (1588_TX_EGRESS_NS)

TABLE 5-1: SYSTEM CONTROL AND STATUS REGISTERS (CONTINUED)

Address	Register Name (Symbol)
170h	1588 GPIO x Rising Edge Clock NanoSeconds Capture Register (1588_GPIO_RE_CLOCK_NS_-CAP_x)
174h	1588 Port RX Message Header Register (1588_RX_MSG_HEADER)
174h	1588 Port TX Message Header Register (1588_TX_MSG_HEADER)
178h	1588 Port RX Pdelay_Req Ingress Time Seconds Register (1588_RX_PDREQ_SEC)
178h	1588 Port TX Delay_Req Egress Time Seconds Register (1588_TX_DREQ_SEC)
178h	1588 GPIO x Falling Edge Clock Seconds Capture Register (1588_GPIO_FE_CLOCK_SEC_CAP_x)
17Ch	1588 Port RX Pdelay_Req Ingress Time NanoSeconds Register (1588_RX_PDREQ_NS)
17Ch	1588 Port TX Delay_Req Egress Time NanoSeconds Register (1588_TX_DREQ_NS)
17Ch	1588 GPIO x Falling Edge Clock NanoSeconds Capture Register (1588_GPIO_FE_CLOCK_NS_-CAP_x)
180h	1588 Port RX Pdelay_Req Ingress Correction Field High Register (1588_RX_PDREQ_CF_HI)
180h	1588 TX One-Step Sync Upper Seconds Register (1588_TX_ONE_STEP_SYNC_SEC)
184h	1588 Port RX Pdelay_Req Ingress Correction Field Low Register (1588_RX_PDREQ_CF_LOW)
188h	1588 Port RX Checksum Dropped Count Register (1588_RX_CHKSUM_DROPPED_CNT)
18Ch	1588 Port RX Filtered Count Register (1588_RX_FILTERED_CNT)
EEPROM/LED Registers	
1B4h	EEPROM Command Register (E2P_CMD)
1B8h	EEPROM Data Register (E2P_DATA)
1BCh	LED Configuration Register (LED_CFG)
GPIO Registers	
1E0h	General Purpose I/O Configuration Register (GPIO_CFG)
1E4h	General Purpose I/O Data & Direction Register (GPIO_DATA_DIR)
1E8h	General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN)
Reset Register	
1F8h	Reset Control Register (RESET_CTL)

5.2 Special Restrictions on Back-to-Back Cycles

5.2.1 BACK-TO-BACK WRITE-READ CYCLES

It is important to note that there are specific restrictions on the timing of back-to-back host write-read operations. These restrictions concern reading registers after any write cycle that may affect the register. In all cases there is a delay between writing to a register and the new value becoming available to be read. In other cases, there is a delay between writing to a register and the subsequent side effect on other registers.

In order to prevent the host from reading stale data after a write operation, minimum wait periods have been established. These periods are specified in Table 5-2. The host processor is required to wait the specified period of time after writing to the indicated register before reading the resource specified in the table. Note that the required wait period is dependent upon the register being read after the write.

Performing “dummy” reads of the Byte Order Test Register (BYTE_TEST) register is a convenient way to guarantee that the minimum write-to-read timing restriction is met. Table 5-2 shows the number of dummy reads that are required before reading the register indicated. The number of BYTE_TEST reads in this table is based on the minimum cycle timing of 45ns. For microprocessors with slower busses the number of reads may be reduced as long as the total time is equal to, or greater than the time specified in the table. Note that dummy reads of the BYTE_TEST register are not required as long as the minimum time period is met.

Note that depending on the host interface mode in use, the basic host interface cycle may naturally provide sufficient time between writes and read. It is required of the system design and register access mechanisms to ensure the proper timing. For example, a write and read to the same register may occur faster than a write and read to different registers.

For 8 and 16-bit write cycles, the wait time for the back-to-back write-read operation applies only to the writing of the last BYTE or WORD of the register, which completes a single DWORD transfer.

For Indexed Address mode HBI operation, the wait time for the back-to-back write-read operation applies only to access to the internal registers and FIFOs. It does not apply to the Host Bus Interface Index Registers or the Host Bus Interface Configuration Register.

TABLE 5-2: READ AFTER WRITE TIMING RULES

After Writing...	wait for this many nanoseconds...	or Perform this many Reads of BYTE_TEST… (assuming T_{cyc} of 45ns)	before reading...
any register	45	1	the same register or any other register affected by the write
Host MAC TX Data FIFO	135	3	TX FIFO Information Register (TX_FIFO_INF)
Interrupt Configuration Register (IRQ_CFG)	60	2	Interrupt Configuration Register (IRQ_CFG)
Interrupt Enable Register (INT_EN)	90	2	Interrupt Configuration Register (IRQ_CFG)
	60	2	Interrupt Status Register (INT_STS)
Interrupt Status Register (INT_STS)	180	4	Interrupt Configuration Register (IRQ_CFG)
	170	4	Interrupt Status Register (INT_STS)
FIFO Level Interrupt Register (FIFO_INT)	90	2	Interrupt Configuration Register (IRQ_CFG)
	80	2	Interrupt Status Register (INT_STS)

TABLE 5-2: READ AFTER WRITE TIMING RULES (CONTINUED)

After Writing...	wait for this many nanoseconds...	or Perform this many Reads of BYTE_TEST… (assuming T_{cyc} of 45ns)	before reading...
Receive Configuration Register (RX_CFG)	80	2	Interrupt Configuration Register (IRQ_CFG)
	60	2	Interrupt Status Register (INT_STS)
	50	2	Interrupt Configuration Register (IRQ_CFG)
Power Management Control Register (PMT_CTRL)	165	4	Power Management Control Register (PMT_CTRL)
	170	4	Interrupt Configuration Register (IRQ_CFG)
	160	4	Interrupt Status Register (INT_STS)
General Purpose Timer Configuration Register (GPT_CFG)	55	2	General Purpose Timer Configuration Register (GPT_CFG)
	170	4	General Purpose Timer Count Register (GPT_CNT)
1588 Command and Control Register (1588_CMD_CTL)	70	2	Interrupt Configuration Register (IRQ_CFG)
	50	2	Interrupt Status Register (INT_STS)
	50	2	1588 Interrupt Status Register (1588_INT_STS)
1588 Interrupt Status Register (1588_INT_STS)	60	2	Interrupt Configuration Register (IRQ_CFG)
1588 Interrupt Enable Register (1588_INT_EN)	60	2	Interrupt Configuration Register (IRQ_CFG)
General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN)	60	2	Interrupt Configuration Register (IRQ_CFG)

5.2.2 BACK-TO-BACK READ CYCLES

There are also restrictions on specific back-to-back host read operations. These restrictions concern reading specific registers after reading a resource that has side effects. In many cases there is a delay between reading the device, and the subsequent indication of the expected change in the control and status register values.

In order to prevent the host from reading stale data on back-to-back reads, minimum wait periods have been established. These periods are specified in Table 5-3. The host processor is required to wait the specified period of time between read operations of specific combinations of resources. The wait period is dependent upon the combination of registers being read.

Performing “dummy” reads of the Byte Order Test Register (BYTE_TEST) register is a convenient way to guarantee that the minimum wait time restriction is met. Table 5-3 below also shows the number of dummy reads that are required for back-to-back read operations. The number of BYTE_TEST reads in this table is based on the minimum timing for T_{cyc}

(45ns). For microprocessors with slower busses the number of reads may be reduced as long as the total time is equal to, or greater than the time specified in the table. Dummy reads of the BYTE_TEST register are not required as long as the minimum time period is met.

Note that depending on the host interface mode in use, the basic host interface cycle may naturally provide sufficient time between reads. It is required of the system design and register access mechanisms to ensure the proper timing. For example, multiple reads to the same register may occur faster than reads to different registers.

For 8 and 16-bit read cycles, the wait time for the back-to-back read operation is required only after the reading of the last BYTE or WORD of the register, which completes a single DWORD transfer. There is no wait requirement between the BYTE or WORD accesses within the DWORD transfer.

TABLE 5-3: READ AFTER READ TIMING RULES

After reading...	wait for this many nanoseconds...	or Perform this many Reads of BYTE_TEST… (assuming T_{cyc} of 45ns)	before reading...
Host MAC RX Data FIFO	135	3	RX FIFO Information Register (RX_FIFO_INF)
Host MAC RX Status FIFO	135	3	RX FIFO Information Register (RX_FIFO_INF)
Host MAC TX Status FIFO	135	3	TX FIFO Information Register (TX_FIFO_INF)
Host MAC RX Dropped Frames Counter Register (RX_DROP)	180	4	Host MAC RX Dropped Frames Counter Register (RX_DROP)

6.0 CLOCKS, RESETS, AND POWER MANAGEMENT

6.1 Clocks

The device provides generation of all system clocks as required by the various sub-modules of the device. The clocking sub-system is comprised of the following:

- Crystal Oscillator
- PHY PLL

6.1.1 CRYSTAL OSCILLATOR

The device requires a fixed-frequency 25 MHz clock source for use by the internal clock oscillator and PLL. This is typically provided by attaching a 25 MHz crystal to the **OSCI** and **OSCO** pins as specified in Section 19.7, "Clock Circuit," on page 413. Optionally, this clock can be provided by driving the **OSCI** input pin with a single-ended 25 MHz clock source. If a single-ended source is selected, the clock input must run continuously for normal device operation. Power savings modes allow for the oscillator or external clock input to be halted.

The crystal oscillator can be disabled as describe in Section 6.3.4, "Chip Level Power Management," on page 47.

For system level verification, the crystal oscillator output can be enabled onto the IRQ pin. See Section 8.2.9, "Clock Output Test Mode," on page 65.

Power for the crystal oscillator is provided by a dedicated regulator or separate input pin. See Section 4.1.2, "1.2 V Crystal Oscillator Regulator," on page 28.

Note:	Crystal specifications are provided in Table 19-13, “Crystal Specifications,” on page 413.

6.1.2 PHY PLL

The PHY module receives the 25 MHz reference clock and, in addition to its internal clock usage, outputs a main system clock that is used to derive device sub-system clocks.

The PHY PLL can be disabled as describe in Section 6.3.4, "Chip Level Power Management," on page 47. The PHY PLL will be disabled only when requested *and* if the PHY port is in a power down mode.

Power for PHY PLL is provided by an external input pin, usually sourced by the device’s 1.2V core regulator. See Section 4.0, "Power Connections," on page 26.

6.2 Resets

The device provides multiple hardware and software reset sources, which allow varying levels of the device to be reset. All resets can be categorized into three reset types as described in the following sections:

- Chip-Level Resets
  - Power-On Reset (POR)
  - RST# Pin Reset
- Multi-Module Resets
  - DIGITAL RESET (DIGITAL_RST)
- Single-Module Resets
  - PHY Reset
  - Host MAC Sub-System Reset
  - 1588 Reset

The device supports the use of configuration straps to allow automatic custom configurations of various device parameters. These configuration strap values are set upon de-assertion of all chip-level resets and can be used to easily set the default parameters of the chip at power-on or pin (RST#) reset. Refer to Section 6.3, "Power Management," on page 44 for detailed information on the usage of these straps.

Table 6-1 summarizes the effect of the various reset sources on the device. Refer to the following sections for detailed information on each of these reset types.

TABLE 6-1: RESET SOURCES AND AFFECTED DEVICE FUNCTIONALITY

Module/ Functionality	POR	RST# Pin	Digital Reset
25 MHz Oscillator	(1)		
Voltage Regulators	(2)		
Host MAC sub-system	X	X	X
PHY	X	X	
PHY Common	(3)		
Voltage Supervision	(3)		
PLL	(3)		
1588 Clock / Event Gen.	X	X	X
1588 Timestamp Unit 0	X	X	X
1588 Timestamp Unit 1	X	X	X
1588 Timestamp Unit 2	X	X	X
SPI/SQI Slave	X	X	X
Host Bus Interface	X	X	X
Power Management	X	X	X
Device EEPROM Loader	X	X	X
I2C Master	X	X	X
GPIO/LED Controller	X	X	X
General Purpose Timer	X	X	X
Free Running Counter	X	X	X
System CSR	X	X	X
Config. Straps Latched	YES	YES	NO(4)
EEPROM Loader Run	YES	YES	YES
Reload Host MAC Addr.	(5)	(5)	(5)
Tristate Output Pins(6)	YES	YES	

Note
1: POR is performed by the XTAL voltage regulator, not at the system level
2: POR is performed internal to the voltage regulators
3: POR is performed internal to the PHY
4: Strap inputs are not re-latched, however Soft-straps are returned to their previously latched pin defaults before they are potentially updated by the EEPROM values.
5: Part of EEPROM loading
6: Only those output pins that are used for straps

TABLE 6-1: RESET SOURCES AND AFFECTED DEVICE FUNCTIONALITY (CONTINUED)

Module/ Functionality	POR	RST# Pin	Digital Reset
RST# Pin Driven Low			

Note 1: POR is performed by the XTAL voltage regulator, not at the system level
2: POR is performed internal to the voltage regulators
3: POR is performed internal to the PHY
4: Strap inputs are not re-latched, however Soft-straps are returned to their previously latched pin defaults before they are potentially updated by the EEPROM values.
5: Part of EEPROM loading
6: Only those output pins that are used for straps

6.2.1 CHIP-LEVEL RESETS

A chip-level reset event activates all internal resets, effectively resetting the entire device. A chip-level reset is initiated by assertion of any of the following input events:

- Power-On Reset (POR)
- RST# Pin Reset

Chip-level reset/configuration completion can be determined by first polling the Byte Order Test Register (BYTE_TEST). The returned data will be invalid until the Host interface resets are complete. Once the returned data is the correct byte ordering value, the Host interface resets have completed.

The completion of the entire chip-level reset must be determined by polling the READY bit of the Hardware Configuration Register (HW_CFG) or Power Management Control Register (PMT_CTRL) until it is set. When set, the READY bit indicates that the reset has completed and the device is ready to be accessed.

With the exception of the Hardware Configuration Register (HW_CFG), Power Management Control Register (PMT_CTRL), Byte Order Test Register (BYTE_TEST), and Reset Control Register (RESET_CTL), read access to any internal resources should not be done by S/W while the READY bit is cleared. Writes to any address are invalid until the READY bit is set.

A chip-level reset involves tuning of the variable output level pads, latching of configuration straps and generation of the master reset.

CONFIGURATION STRAPS LATCHING

During POR or **RST#** pin reset, the latches for the straps are open. Following the release of POR or **RST#** pin reset, the latches for the straps are closed.

VARIABLE LEVEL I/O PAD TUNING

Following the release of the POR or **RST#** pin resets, a 1 uS pulse (active low), is sent into the VO tuning circuit. 2 uS later, the output pins are enabled. The 2 uS delay allows time for the variable output level pins to tune before enabling the outputs and also provides input hold time for strap pins that are shared with output pins.

MASTER RESET AND CLOCK GENERATION RESET

Following the enabling of the output pins, the reset is synchronized to the main system clock to become the master reset. Master reset is used to generate the local resets and to reset the clocks generation.

6.2.1.1 Power-On Reset (POR)

A power-on reset occurs whenever power is initially applied to the device or if the power is removed and reapplied to the device. This event resets all circuitry within the device. Configuration straps are latched and EEPROM loading is performed as a result of this reset. The POR is used to trigger the tuning of the Variable Level I/O Pads as well as a chip-level reset.

Following valid voltage levels, a POR reset typically takes approximately 21 ms, plus any additional time (91 us per byte) for data loaded from the EEPROM. A full 64KB EEPROM load would complete in approximately 6 seconds.

6.2.1.2 RST# Pin Reset

Driving the RST# input pin low initiates a chip-level reset. This event resets all circuitry within the device. Use of this reset input is optional, but when used, it must be driven for the period of time specified in Section 19.6.3, "Reset and Configuration Strap Timing," on page 410. Configuration straps are latched, and EEPROM loading is performed as a result of this reset.

A RST# pin reset typically takes approximately 760 µs plus any additional time (91 us per byte) for data loaded from the EEPROM. A full 64KB EEPROM load would complete in approximately 6 seconds.

Note:	The RST# pin is pulled-high internally. If unused, this signal can be left unconnected. Do not rely on internal pull-up resistors to drive signals external to the device.

Please refer to Table 3-8, “Miscellaneous Pin Descriptions,” on page 24 for a description of the RST# pin.

6.2.2 BLOCK-LEVEL RESETS

The block level resets contain an assortment of reset register bit inputs and generate resets for the various blocks. Block level resets can affect one or multiple modules.

6.2.2.1 Multi-Module Resets

Multi-module resets activate multiple internal resets, but do not reset the entire chip. Configuration straps are *not* latched upon multi-module resets. A multi-module reset is initiated by assertion of the following:

- DIGITAL RESET (DIGITAL_RST)

Multi-module reset/configuration completion can be determined by first polling the Byte Order Test Register (BYTE_TEST). The returned data will be invalid until the Host interface resets are complete. Once the returned data is the correct byte ordering value, the Host interface resets have completed.

The completion of the entire chip-level reset must be determined by polling the READY bit of the Hardware Configuration Register (HW_CFG) or Power Management Control Register (PMT_CTRL) until it is set. When set, the READY bit indicates that the reset has completed and the device is ready to be accessed.

With the exception of the Hardware Configuration Register (HW_CFG), Power Management Control Register (PMT_CTRL), Byte Order Test Register (BYTE_TEST), and Reset Control Register (RESET_CTL), read access to any internal resources should not be done by S/W while the READY bit is cleared. Writes to any address are invalid until the READY bit is set.

Note:	The digital reset does not reset register bits designated as NASR.

DIGITAL RESET (DIGITAL_RST)

A digital reset is performed by setting the DIGITAL_RST bit of the Reset Control Register (RESET_CTL). A digital reset will reset all device sub-modules except the Ethernet PHY. EEPROM loading is performed following this reset. Configuration straps are *not* latched as a result of a digital reset. However, soft straps are first returned to their previously latched pin values and register bits that default to strap values are reloaded.

A digital reset typically takes approximately 760 µs plus any additional time (91 uS per byte) for data loaded from the EEPROM. A full 64KB EEPROM load would complete in approximately 6 seconds.

6.2.2.2 Single-Module Resets

A single-module reset will reset only the specified module. Single-module resets do *not* latch the configuration straps or initiate the EEPROM Loader. A single-module reset is initiated by assertion of the following:

- PHY Reset
- Host MAC Sub-System Reset
- 1588 Reset

PHY Reset

A PHY reset is performed by setting the PHY_RST bit of the Reset Control Register (RESET_CTL) or the Soft Reset bit in the PHY Basic Control Register (PHY_BASIC_CONTROL). Upon completion of the PHY reset, the PHY_RST and Soft Reset bits are automatically cleared. No other modules of the device are affected by this reset.

PHY reset completion can be determined by polling the PHY_RST bit in the Reset Control Register (RESET_CTL) or the Soft Reset bit in the PHY Basic Control Register (PHY_BASIC_CONTROL) until it clears. Under normal conditions, the PHY_RST and Soft Reset bit will clear approximately 102 uS after the PHY reset occurrence.

Note: When using the Soft Reset bit to reset the PHY, register bits designated as NASR are not reset.

In addition to the methods above, the PHY is automatically reset after returning from a PHY power-down mode. This reset differs in that the PHY power-down mode reset does not reload or reset any of the PHY registers. Refer to Section 12.2.10, "PHY Power-Down Modes," on page 222 for additional information.

Refer to Section 12.2.12, "Resets," on page 224 for additional information on PHY resets.

Host MAC Sub-System Reset

A Host MAC sub-system reset is performed by setting the HMAC_RST bit in the Reset Control Register (RESET_CTL). In addition, the MAC address of the Host MAC is reloaded from the EEPROM, using the device EEPROM loader.

This will reset the Host MAC and FIFOs, including:

- MAC
- Address Filtering
- Wake-On-LAN
- RX Checksum Offload
- TX Checksum Offload
- Energy Efficient Ethernet Control and Counters
- FIFOs
- Flow Control Logic

The following registers and register fields will be reset:

- All registers described in Section 11.14, "Host MAC & FIFO Interface Registers," on page 176

  Note: The HBI register locks associated with these register are also reset.

- All registers described in Section 11.15, "Host MAC Control and Status Registers," on page 192
  - The EEPROM Controller Busy (EPC_BUSY) and Configuration Loaded (CFG_LOADED) bits in the EEPROM Command Register (E2P_CMD) (set and cleared respectively)

  Note: The bits in the Interrupt Status Register (INT_STS) listed in Section 8.2.4, "Host MAC Interrupts" are not reset.

  Note: Host MAC and FIFO related bits in the Hardware Configuration Register (HW_CFG) are not reset.

Host MAC reset completion can be determined by polling the HMAC_RST bit in the Reset Control Register (RESET_CTL) until it clears.

1588 Reset

A reset of all 1588 related logic, including the clock/event generation and 1588 TSUs, is performed by setting the 1588 Reset (1588_RESET) bit in the 1588 Command and Control Register (1588_CMD_CTL).

The registers described in Section 14.0, "IEEE 1588," on page 298 are reset.

No other modules of the device are affected by this reset.

1588 reset completion can be determined by polling the 1588 Reset (1588_RESET) bit in the 1588 Command and Control Register (1588_CMD_CTL) until it clears.

6.2.3 RESET REGISTERS

6.2.3.1 Reset Control Register (RESET_CTL)

Offset: 1F8h Size: 32 bits

This register contains software controlled resets.

Note: This register can be read while the device is in the reset or not ready / power savings states without leaving the host interface in an intermediate state. If the host interface is in a reset state, returned data may be invalid.

It is not necessary to read all four bytes of this register. DWORD access rules do not apply to this register.

Bits	Description	Type	Default
31:7	**RESERVED**	RO	-
6	**RESERVED**	RO	-
5	**Host MAC Reset (HMAC_RST)** Setting this bit resets the Host MAC sub-system. When the Host MAC sub-system is released from reset, this bit is automatically cleared. All writes to this bit are ignored while this bit is set. **Note:** This bit is not accessible via the EEPROM Loader's register initialization function (Section 13.4.5).	R/W SC	0b
4	**RESERVED**	RO	-
3	**RESERVED**	RO	-
2	**RESERVED**	RO	-

Bits	Description	Type	Default
1	**PHY Reset (PHY_RST)** Setting this bit resets the PHY. The internal logic automatically holds the PHY reset for a minimum of 102uS. When the PHY is released from reset, this bit is automatically cleared. All writes to this bit are ignored while this bit is set. **Note:** This bit is not accessible via the EEPROM Loader's register initialization function (Section 13.4.5).	R/W SC	0b
0	**Digital Reset (DIGITAL_RST)** Setting this bit resets the complete chip except the PLL and PHY. All system CSRs are reset except for any NASR type bits. Any in progress EEPROM commands (including RELOAD) are terminated. The EEPROM Loader will automatically reload the configuration following this reset, but will not reset the PHY. If desired, the above PHY reset can be issued once the device is configured. When the chip is released from reset, this bit is automatically cleared. All writes to this bit are ignored while this bit is set. **Note:** This bit is not accessible via the EEPROM Loader's register initialization function (Section 13.4.5).	R/W SC	0b

6.3 Power Management

The device supports several block and chip level power management features as well as wake-up event detection and notification.

6.3.1 WAKE-UP EVENT DETECTION

6.3.1.1 Host MAC Wake on LAN (WoL)

The Host MAC provides the following Wake-on-LAN detection modes:

Perfect DA (Destination Address): This mode, enabled by the Perfect DA Wakeup Enable (PFDA_EN) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR), will trigger a WoL event when an incoming frame has a destination address field that exactly matches the MAC address programmed into the MAC. The Perfect DA Frame Received (PFDA_FR) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) will be set when PFDA_EN is set, and a Perfect DA event occurs.

Broadcast: This mode, enabled by the Broadcast Wakeup Enable (BCST_EN) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR), will trigger a WoL event when an incoming frame is a broadcast frame. The Broadcast Frame Received (BCAST_FR) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) will be set when BCST_EN is set, and a Broadcast event occurs.

Remote Wake-up Frame: This mode, enabled by the Wake-Up Frame Enable (WUEN) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR), will trigger a WoL event when an incoming frame is accepted based on the Wake-Up Filter registers in the Host MAC. The Remote Wake-Up Frame Received (WUFR) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) will be set when WUEN is set, and a Remove Wake-up Frame event occurs.

Magic Packet: This mode, enabled by the Magic Packet Enable (MPEN) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR), will trigger a WoL event when an incoming frame is accepted and is a Magic Packet. The Magic Packet Received (MPR) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) will be set when MPEN is set, and a Magic Packet event occurs.

If any of the PFDA_FR, BCAST_FR, WUFR or MPR bits are set, the Wake On Status (WOL_STS) bit of the Power Management Control Register (PMT_CTRL) will be set. The Wake-On-Enable (WOL_EN) enables this bit as a PME event.

In addition, when the Power Management Wakeup (PM_WAKE) bit in the Power Management Control Register (PMT_CTRL) is set, these events can wake up the chip.

Refer to Section 11.6.1, "Perfect DA Detection," on page 144, Section 11.6.2, "Broadcast Detection," on page 145, Section 11.6.3, "Wake-up Frame Detection," on page 145 and Section 11.6.4, "Magic Packet Detection," on page 150 for additional details on these features.

6.3.1.2 PHY Energy Detect

Energy Detect Power Down mode reduces PHY power consumption. In energy-detect power-down mode, the PHY will resume from power-down when energy is seen on the cable (typically from link pulses) and set the ENERGYON interrupt bit in the PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE).

Refer to Section 12.2.10.2, "Energy Detect Power-Down," on page 223 for details on the operation and configuration of the PHY energy-detect power-down mode.

Note:	If a carrier is present when Energy Detect Power Down is enabled, then detection will occur immediately.

If enabled, via the PHY Interrupt Mask Register (PHY_INTERRUPT_MASK), the PHY will generate an interrupt. This interrupt is reflected in the Interrupt Status Register (INT_STS), bit 26 (PHY_INT). The INT_STS register bit will trigger the IRQ interrupt output pin if enabled, as described in Section 8.2.2, "Ethernet PHY Interrupts," on page 63.

The energy-detect PHY interrupt will also set the **Energy-Detect Status (ED_STS)** bit of the Power Management Control Register (PMT_CTRL). The **Energy-Detect Enable (ED_EN)** bit will enable the status bit as a PME event.

Note:	Any PHY interrupt will set the above status bits. The Host should only enable the appropriate PHY interrupt source in the PHY Interrupt Mask Register (PHY_INTERRUPT_MASK).

Note:

6.3.2 WAKE-UP (PME) NOTIFICATION

A simplified diagram of the logic that controls the **PME** output pin and PME interrupt can be seen in Figure 6-1.

The PME module handles the latching of the Host MAC Wake On Status (WOL_STS) bit and the PHY **Energy-Detect Status (ED_STS)** bit in the Power Management Control Register (PMT_CTRL).

This module also masks the status bits with the corresponding enable bits (Wake-On-Enable (WOL_EN) and **Energy-Detect Enable (ED_EN)**) and combines the results together to generate the Power Management Interrupt Event (PME_INT) status bit in the Interrupt Status Register (INT_STS). The PME_INT status bit is then masked with the Power Management Event Interrupt Enable (PME_INT_EN) bit and combined with the other interrupt sources to drive the **IRQ** output pin.

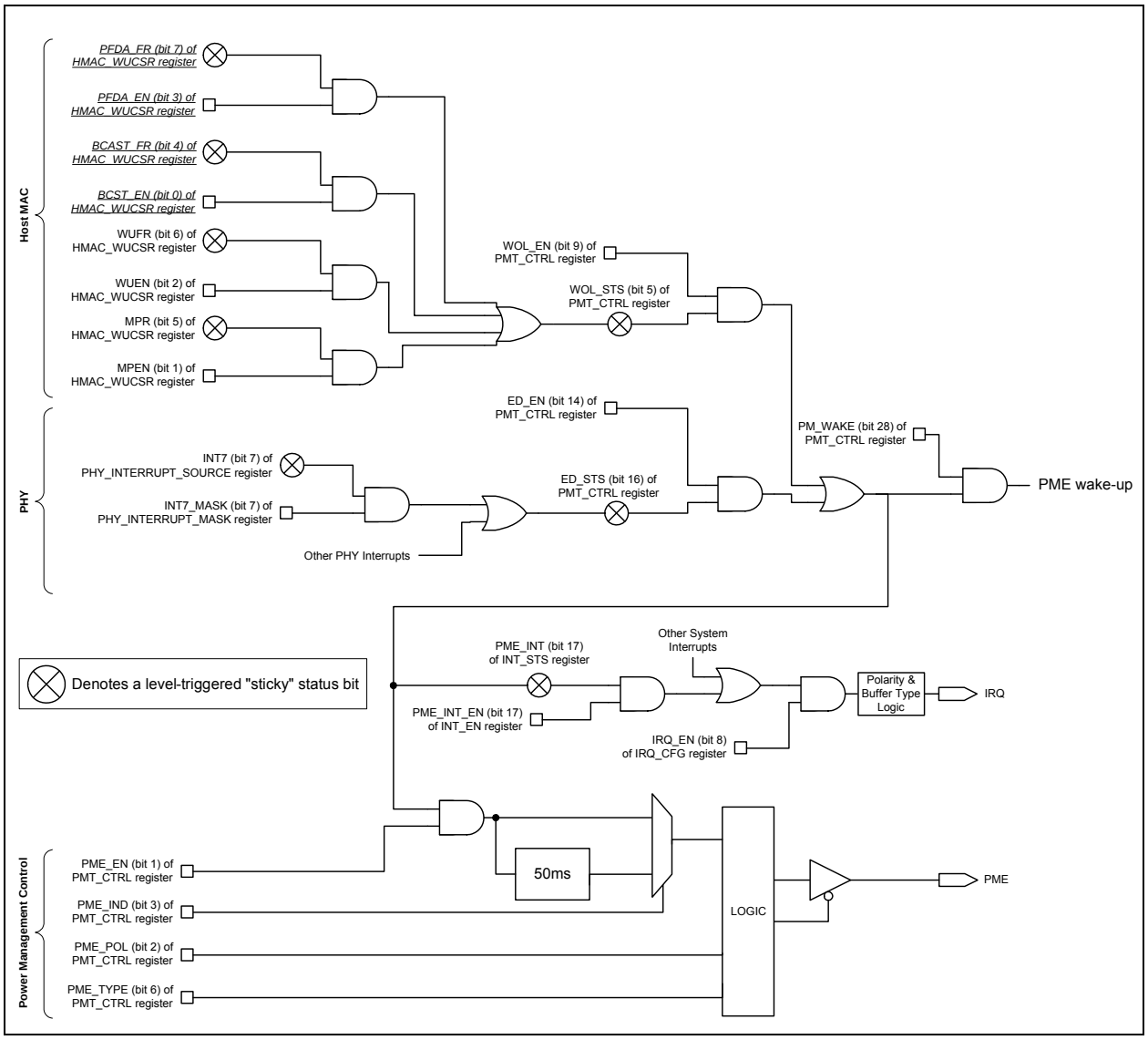
Note:	The PME interrupt status bit (PME_INT) in the INT_STS register is set regardless of the setting of PME_INT_EN.

In addition to generating interrupt events, the PME event can also drive the **PME** output pin to indicate wake-up events exclusively. The PME event is enabled with the PME Enable (PME_EN) in the Power Management Control Register (PMT_CTRL), The **PME** output pin characteristics can be configured via the PME Buffer Type (PME_TYPE), PME Indication (PME_IND) and PME Polarity (PME_POL) bits of the Power Management Control Register (PMT_CTRL). These bits allow the **PME** output pin to be open-drain, active high push-pull or active-low push-pull and configure the output to be continuous, or pulse for 50 ms.

In system configurations where the **PME** output pin is shared among multiple devices (wired ORed), the WOL_STS and ED_STS bits within the PMT_CTRL register can be read to determine which device is driving the PME signal.

When the PM_WAKE bit of the Power Management Control Register (PMT_CTRL) is set, the PME event will automatically wake up the system in certain chip level power modes, as described in Section 6.3.4.2, "Exiting Low Power Modes," on page 48. This is done independent from the values of the PME_EN, PME_POL, PME_IND and PME_TYPE register bits.

FIGURE 6-1: PME PIN AND PME INTERRUPT SIGNAL GENERATION



6.3.3 BLOCK LEVEL POWER MANAGEMENT

The device supports software controlled clock disabling of various modules in order to reduce power consumption.

Note:	Disabling individual blocks does not automatically reset the block, it only places it into a static non-operational state in order to reduce the power consumption of the device. If a block reset is not performed before re-enabling the block, then care must be taken to ensure that the block is in a state where it can be disabled and then re-enabled.

6.3.3.1 Disabling The Host MAC

The entire Host MAC may be disabled by setting the HMAC_DIS bit in the Power Management Control Register (PMT_CTRL). As a safety precaution, in order for this bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.

The Host MAC WoL detection can be left functioning by using the HMAC_SYS_ONLY_DIS bit instead, which keeps the RX and TX clocks enabled. As a safety precaution, in order for this bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.

6.3.3.2 Disabling The 1588 Unit

The entire 1588 Unit, including the CSRs, may be disabled by setting the 1588_DIS bit in the Power Management Control Register (PMT_CTRL). As a safety precaution, in order for this bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.

The Timestamp Unit, including CSRs, may be disabled by setting the appropriate 1588_TSU_DIS bit in the Power Management Control Register (PMT_CTRL). As a safety precaution, in order for a bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.

6.3.3.3 PHY Power Down

The PHY may be placed into power-down as described in Section 12.2.10, "PHY Power-Down Modes," on page 222.

6.3.3.4 LED Pins Power Down

All LED outputs may be disabled by setting the LED_DIS bit in the Power Management Control Register (PMT_CTRL).

Open-drain / open-source LEDs are un-driven. Push-pull LEDs are still driven but are set to their inactive state.

APPLICATION NOTE: Individual LEDs can be disabled by setting them open-drain GPIO outputs with a data value of 1.

6.3.4 CHIP LEVEL POWER MANAGEMENT

The device supports power-down modes to allow applications to minimize power consumption.

Power is reduced by disabling the clocks as outlined in Table 6-2, "Power Management States". All configuration data is saved when in any power state. Register contents are not affected unless specifically indicated in the register description.

There is one normal operating power state, D0, and three power saving states: D1, D2 and D3. Although appropriate for various wake-up detection functions, the power states do not directly enable and are not enforced by these functions.

D0: Normal Mode - This is the normal mode of operation of this device. In this mode, all functionality is available. This mode is entered automatically on any chip-level reset (POR, **RST#** pin reset).

D1: System Clocks Disabled, XTAL, PLL and network clocks enabled - In this low power mode, all clocks derived from the PLL clock are disabled. The network clocks remain enabled if supplied by the PHY. The crystal oscillator and the PLL remain enabled. Exit from this mode may be done manually or automatically.

This mode is useful for MAC WoL mode, where the PHY is enabled and the MAC is configured for WoL detection.

This mode could be used for PHY General Power Down mode and PHY Energy Detect Power Down mode.

D2: System Clocks Disabled, PLL disable requested, XTAL enabled - In this low power mode, all clocks derived from the PLL clock are disabled. The PLL is allowed to be disabled (and will disable if the PHY is in either Energy Detect or General Power Down). The network clocks remain enabled if supplied by the PHY. The crystal oscillator remains enabled. Exit from this mode may be done manually or automatically.

This mode is useful for PHY Energy Detect Power Down mode. This mode could be used for PHY General Power Down mode.

D3: System Clocks Disabled, PLL disabled, XTAL disabled - In this low power mode, all clocks derived from the PLL clock are disabled. The PLL will be disabled. The crystal oscillator is disabled. Exit from this mode may be only be done manually.

This mode is useful for PHY General Power Down mode.

The Host must place the PHY into General Power Down mode by setting the Power Down (PHY_PWR_DWN) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) before setting this power state.

TABLE 6-2: POWER MANAGEMENT STATES

Clock Source	D0	D1	D2	D3
25 MHz Crystal Oscillator	ON	ON	ON	OFF
PLL	ON	ON	OFF(2)	OFF
system clocks (100 MHz, 50 MHz, 25 MHz and others)	ON	OFF	OFF	OFF
network clocks	available(1)	available(1)	available(1)	OFF(3)

Note 1: If supplied by the PHY
2: PLL is requested to be turned off and will disable if the PHY is in either Energy Detect or General Power Down
3: PHY clocks are off

6.3.4.1 Entering Low Power Modes

To enter any of the low power modes (D1 - D3) from normal mode (D0), follow these steps:

1. Write the PM_MODE and PM_WAKE fields in the Power Management Control Register (PMT_CTRL) to their desired values
2. Set the wake-up detection desired per Section 6.3.1, "Wake-Up Event Detection".
3. Set the appropriate wake-up notification per Section 6.3.2, "Wake-Up (PME) Notification".
4. Ensure that the device is in a state where it can safely be placed into a low power mode (all packets transmitted, receivers disabled, packets processed / flushed, etc.)
5. Set the PM_SLEEP_EN bit in the Power Management Control Register (PMT_CTRL).

Note: The PM_MODE field cannot be changed at the same time as the PM_SLEEP_EN bit is set and the PM_SLEEP_EN bit cannot be set at the same time that the PM_MODE field is changed.

Note: The EEPROM Loader Register Data burst sequence (Section 13.4.5) can be used to achieve an initial power down state without the need of software by:

- First setting the PHY into General Purpose Power Down by setting the PHY_PWR_DWN bit in PHY_BASIC_CONTROL indirectly via the HMAC_MII_DATA / HMAC_MII_ACC via the MAC_CSR_CMD / MAC_CSR_DATA registers.
- Setting the PM_MODE and PM_SLEEP_EN bits in the Power Management Control Register (PMT_CTRL).

Upon entering any low power mode, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) and the Power Management Control Register (PMT_CTRL) is forced low.

Note: Upon entry into any of the power saving states the host interfaces are not functional.

6.3.4.2 Exiting Low Power Modes

Exiting from a low power mode can be done manually or automatically.

An automatic wake-up will occur based on the events described in Section 6.3.2, "Wake-Up (PME) Notification". Automatic wake-up is enabled with the Power Management Wakeup (PM_WAKE) bit in the Power Management Control Register (PMT_CTRL).

A manual wake-up is initiated by the host when:

- an HBI write (**CS** and **WR** or **CS**, **RD_WR** and **ENB**) is performed to the device. Although all writes are ignored until the device has been woken and a read performed, the host should direct the write to the Byte Order Test Register (BYTE_TEST). Writes to any other addresses should not be attempted until the device is awake.
- an SPI/SQI cycle (**SCS#** low and **SCK** high) is performed to the device. Although all reads and writes are ignored until the device has been woken, the host should direct the use a read of the Byte Order Test Register (BYTE_TEST) to wake the device. Reads and writes to any other addresses should not be attempted until the device is awake.

To determine when the host interface is functional, the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) or the Power Management Control Register (PMT_CTRL) can be polled to determine when the device is fully awake.

For both automatic and manual wake-up, the Device Ready (READY) bit will go high once the device is returned to power savings state D0 and the PLL has re-stabilized. The PM_MODE and PM_SLEEP_EN fields in the Power Management Control Register (PMT_CTRL) will also clear at this point.

Under normal conditions, the device will wake-up within 2 ms.

6.3.5 POWER MANAGEMENT REGISTERS

6.3.5.1 Power Management Control Register (PMT_CTRL)

Offset: 084h Size: 32 bits

This read-write register controls the power management features and the PME pin of the device. The ready state of the device be determined via the Device Ready (READY) bit of this register.

Note:	This register can be read while the device is in the reset or not ready / power savings states without leaving the host interface in an intermediate state. If the host interface is in a reset state, returned data may be invalid. It is not necessary to read all four bytes of this register. DWORD access rules do not apply to this register.

Bits	Description	Type	Default
31:29	**Power Management Mode (PM_MODE)** This register field determines the chip level power management mode that will be entered when the Power Management Sleep Enable (PM_SLEEP_EN) bit is set. **000: D0** **001: D1** **010: D2** **011: D3** **100: Reserved** **101: Reserved** **110: Reserved** **111: Reserved** Writes to this field are ignored if Power Management Sleep Enable (PM_SLEEP_EN) is also being written with a 1. This field is cleared when the device wakes up.	R/W/SC	000b

Bits	Description	Type	Default
28	**Power Management Sleep Enable (PM_SLEEP_EN)** Setting this bit enters the chip level power management mode specified with the Power Management Mode (PM_MODE) field. **0: Device is not in a low power sleep state** **1: Device is in a low power sleep state** This bit can not be written at the same time as the PM_MODE register field. The PM_MODE field must be set, and then this bit must be set for proper device operation. Writes to this bit with a value of 1 are ignored if Power Management Mode (PM_MODE) is being written with a new value. **Note:** Although not prevented by H/W, this bit should not be written with a value of 1 while Power Management Mode (PM_MODE) has a value of “D0”. This field is cleared when the device wakes up.	R/W/SC	0b
27	**Power Management Wakeup (PM_WAKE)** When set, this bit enables automatic wake-up based on PME events. **0: Manual Wakeup only** **1: Auto Wakeup enabled**	R/W	0b
26	**LED Disable (LED_DIS)** This bit disables LED outputs. Open-drain / open-source LEDs are un-driven. Push-pull LEDs are still driven but are set to their inactive state. **0: LEDs are enabled** **1: LEDs are disabled**	R/W	0b
25	**1588 Clock Disable (1588_DIS)** This bit disables the clocks for the entire 1588 Unit. **0: Clocks are enabled** **1: Clocks are disabled** In order for this bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.	R/W	0b
24:23	**RESERVED**	RO	-
22	**1588 Timestamp Unit Clock Disable (1588_TSU_DIS)** This bit disables the clocks for the 1588 timestamp unit. **0: Clocks are enabled** **1: Clocks are disabled** In order for this bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.	R/W	0b
21	**RESERVED**	RO	-
20	**RESERVED**	RO	-

Bits	Description	Type	Default
19	**Host MAC Clock Disable (HMAC_DIS)** This bit disables the 25 and 100 MHz, RX and TX clocks to the MAC. **0: Clocks are enabled** **1: Clocks are disabled** In order for this bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.	R/W	0b
18	**Host MAC System Clock Only Disable (HMAC_SYS_ONLY_DIS)** This bit disables the 25 and 100 MHz clocks to the MAC but leaves the RX and TX clocks active. **0: Clocks are enabled** **1: Clocks are disabled** In order for this bit to be set, it must be written as a 1 two consecutive times. A write of a 0 will reset the count.	R/W	0b
17	**RESERVED**	RO	-
16	**Energy-Detect Status (ED_STS)** This bit indicates an energy detect event occurred on the PHY. In order to clear this bit, it is required that the event in the PHY be cleared as well. The event sources are described in Section 6.3, "Power Management," on page 44.	R/WC	0b
15	**RESERVED**	RO	-
14	**Energy-Detect Enable (ED_EN)** When set, the PME_INT bit in the Interrupt Status Register (INT_STS) will be asserted upon an energy-detect event from the port. When set, the **PME** output pin (if enabled via the PME_EN bit) will also be asserted in accordance with the PME_IND bit upon an energy-detect event from the port.	R/W	0b
13:10	**RESERVED**	RO	-
9	**Wake-On-Enable (WOL_EN)** When set, the PME_INT bit in the Interrupt Status Register (INT_STS) will be asserted upon a Host MAC WOL event. When set, the **PME** output pin (if enabled via the PME_EN bit) will also be asserted in accordance with the PME_IND bit upon a Host MAC WOL event.	R/W	0b
8:7	**RESERVED**	RO	-
6	**PME Buffer Type (PME_TYPE)** When this bit is cleared, the **PME** output pin functions as an open-drain buffer for use in a wired-or configuration. When set, the **PME** output pin is a push-pull driver. When the **PME** output pin is configured as an open-drain output, the PME_POL field of this register is ignored and the output is always active low. **0: PME pin open-drain output** **1: PME pin push-pull driver**	R/W NASR	0b

Bits	Description	Type	Default
5	**Wake On Status (WOL_STS)** This bit indicates that a Wake-Up, Magic Packet, Perfect DA, or Broadcast frame was detected by the Host MAC. In order to clear this bit, it is required that the event in the Host MAC be cleared as well. The event sources are described in Section 6.3, "Power Management," on page 44.	R/WC	0b
4	**RESERVED**	RO	-
3	**PME Indication (PME_IND)** The PME signal can be configured as a pulsed output or a static signal, which is asserted upon detection of a wake-up event. When set, the PME signal will pulse active for 50mS upon detection of a wake-up event. When cleared, the PME signal is driven continuously upon detection of a wake-up event. **0: PME driven continuously on detection of event** **1: PME 50mS pulse on detection of event** The PME signal can be deactivated by clearing the above status bit(s) or by clearing the appropriate enable(s).	R/W	0b
2	**PME Polarity (PME_POL)** This bit controls the polarity of the PME signal. When set, the PME output is an active high signal. When cleared, it is active low. **Note:** When PME is configured as an open-drain output, this field is ignored and the output is always active low. **0: PME active low** **1: PME active high**	R/W NASR	0b
1	**PME Enable (PME_EN)** When set, this bit enables the external PME signal pin. When cleared, the external PME signal is disabled. **Note:** This bit does not affect the PME_INT interrupt bit of the Interrupt Status Register (INT_STS). **0: PME pin disabled** **1: PME pin enabled**	R/W	0b
0	**Device Ready (READY)** When set, this bit indicates that the device is ready to be accessed. Upon power-up, RST# reset, return from power savings states, Host MAC module level reset or digital reset, the host processor may interrogate this field as an indication that the device has stabilized and is fully active. This rising edge of this bit will assert the Device Ready (READY) bit in INT_STS and can cause an interrupt if enabled. **Note:** With the exception of the HW_CFG, PMT_CTRL, BYTE_TEST, and RESET_CTL registers, read access to any internal resources is forbidden while the READY bit is cleared. Writes to any address are invalid until this bit is set. **Note:** This bit is identical to bit 27 of the Hardware Configuration Register (HW_CFG).	RO	0b

6.4 Device Ready Operation

The device supports a Ready status register bit that indicates to the Host software when the device is fully ready for operation. This bit may be read via the Power Management Control Register (PMT_CTRL) or the Hardware Configuration Register (HW_CFG).

Following power-up reset, **RST#** reset, or digital reset (see Section 6.2, "Resets"), the Device Ready (READY) bit indicates that the device has read, and is configured from, the contents of the EEPROM.

An EEPROM RELOAD command, via the EEPROM Command Register (E2P_CMD), will restart the EEPROM Loader, temporarily causing the Device Ready (READY) to be low.

A Host MAC reset, via the Reset Control Register (RESET_CTL), will utilize the EEPROM Loader, temporarily causing the Device Ready (READY) to be low.

Entry into any power savings state (see Section 6.3.4, "Chip Level Power Management") other than D0 will cause Device Ready (READY) to be low. Upon wake-up, the Device Ready (READY) bit will go high once the device is returned to power savings state D0 and the PLL has re-stabilized.

7.0 CONFIGURATION STRAPS

Configuration straps allow various features of the device to be automatically configured to user defined values. Configuration straps can be organized into two main categories: Hard-Straps and Soft-Straps. Both hard-straps and soft-straps are latched upon Power-On Reset (POR), or pin reset (**RST#**). The primary difference between these strap types is that soft-strap default values can be overridden by the EEPROM Loader, while hard-straps cannot.

Configuration straps which have a corresponding external pin include internal resistors in order to prevent the signal from floating when unconnected. If a particular configuration strap is connected to a load, an external pull-up or pull-down resistor should be used to augment the internal resistor to ensure that it reaches the required voltage level prior to latching. The internal resistor can also be overridden by the addition of an external resistor.

Note:	The system designer must guarantee that configuration strap pins meet the timing requirements specified in Section 19.6.3, "Reset and Configuration Strap Timing". If configuration strap pins are not at the correct voltage level prior to being latched, the device may capture incorrect strap values.

7.1 Soft-Straps

Soft-strap values are latched on the release of POR or **RST#** and are overridden by values from the EEPROM Loader (when an EEPROM is present). These straps are used as direct configuration values or as defaults for CPU registers. Some, but not all, soft-straps have an associated pin. Those that do not have an associated pin, have a tie off default value. All soft-strap values can be overridden by the EEPROM Loader. Refer to Section 13.4, "EEPROM Loader," on page 290 for information on the operation of the EEPROM Loader and the loading of strap values. Table 13-4, “EEPROM Configuration Bits,” on page 292 defines the soft-strap EEPROM bit map.

Straps which have an associated pin are also fully defined in Section 3.0, "Pin Descriptions and Configuration," on page 10.

Table 7-1 provides a list of all soft-straps and their associated pin or default value.

Note:	The use of the term “configures” in the “Description” section of Table 7-1 indicates the register bit is loaded with the strap value, while the term “Affects” means the value of the register bit is determined by the strap value and some other condition(s).

Upon setting the Digital Reset (DIGITAL_RST) bit in the Reset Control Register (RESET_CTL) or upon issuing a RELOAD command via the EEPROM Command Register (E2P_CMD), these straps return to their original latched (non-overridden) values if an EEPROM is no longer attached or has been erased. The associated pins are not re-sampled (i.e. the value latched on the pin during the last POR or **RST#** will be used, not the value on the pin during the digital reset or RELOAD command issuance). If it is desired to re-latch the current configuration strap pin values, a POR or **RST#** must be issued.

TABLE 7-1: SOFT-STRAP CONFIGURATION STRAP DEFINITIONS

Strap Name	Description	Pin / Default Value
LED_en_strap[2:0]	**LED Enable Straps:** Configures the default value for the LED Enable 2-0 (LED_EN[2:0]) bits of the LED Configuration Register (LED_CFG).	111b
LED_fun_strap[2:0]	**LED Function Straps:** Configures the default value for the LED Function 2-0 (LED_FUN[2:0]) bits of the LED Configuration Register (LED_CFG).	000b

TABLE 7-1: SOFT-STRAP CONFIGURATION STRAP DEFINITIONS (CONTINUED)

Strap Name	Description	Pin / Default Value
HBI_ale_qualification_strap	**HBI ALE Qualification Strap:** Configures the HBI interface to qualify the ALEHI and ALELO signals with the CS signal. **0 = address input is latched with ALEHI and ALELO** **1 = address input is latched with ALEHI and ALELO only when CS is active**	1b
HBI_rw_mode_strap	**HBI Read / Write Mode Strap:** Configures the HBI interface for separate read & write signals or direction and enable signals. **0 = read & write** **1 = direction & enable**	0b
HBI_cs_polarity_strap	**HBI Chip Select Polarity Strap:** Configures the polarity of the HBI interface chip select signal. **0 = active low** **1 = active high**	0b
HBI_rd_rdwr_polarity_strap	**HBI Read, Read / Write Polarity Strap:** Configures the polarity of the HBI interface read signal. **0 = active low read** **1 = active high read** Configures the polarity of the HBI interface read / write signal. **0 = read when 1, write when 0 (R/nW)** **1 = write when 1, read when 0 (W/nR)**	0b
HBI_wr_en_polarity_strap	**HBI Write, Enable Polarity Strap:** Configures the polarity of the HBI interface write signal. **0 = active low write** **1 = active high write** Configures the polarity of the HBI interface enable signal. **0 = active low enable** **1 = active high enable**	0b
HBI_ale_polarity_strap	**HBI ALE Polarity Strap:** Configures the polarity of the HBI interface **ALEHI** and **ALELO** signals. **0 = active low strobe (address saved on rising edge)** **1 = active high strobe (address saved on falling edge)**	1b
1588_enable_strap	**1588 Enable Strap:** Configures the default value of the 1588 Enable (1588_ENABLE) bit in the 1588 Command and Control Register (1588_CMD_CTL).	0b

TABLE 7-1: SOFT-STRAP CONFIGURATION STRAP DEFINITIONS (CONTINUED)

Strap Name	Description	Pin / Default Value
auto_mdix_strap_1	**PHY Auto-MDIX Enable Strap:** Configures the default value of the AMDIX_EN Strap State bit of the Hardware Configuration Register (HW_CFG). This strap is also used in conjunction with manual_mdix_strap_1 to configure PHY Auto-MDIX functionality when the Auto-MDIX Control (AMDIXCTRL) bit in the PHY Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND) indicates the strap settings should be used for auto-MDIX configuration. **Note:** This has no effect when the PHY is in 100BASE-FX mode. Refer to the respective register definition sections for additional information.	1b
manual_mdix_strap_1	**PHY Manual MDIX Strap:** Configures MDI(0) or MDIX(1) for the PHY when the auto_mdix_strap_1 is low and the Auto-MDIX Control (AMDIXCTRL) bit in the PHY Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND) indicates the strap settings are to be used for auto-MDIX configuration. **Note:** This has no effect when the PHY is in 100BASE-FX mode. Refer to the respective register definition sections for additional information.	0b
autoneg_strap_1	**PHY Auto Negotiation Enable Strap:** Configures the default value of the Auto-Negotiation Enable (PHY_AN) enable bit in the PHY Basic Control Register (PHY_BASIC_CONTROL). This strap also affects the default value of the following register bits: • Speed Select LSB (PHY_SPEED_SEL_LSB) and Duplex Mode (PHY_DUPLEX) bits of the PHY Basic Control Register (PHY_BASIC_CONTROL) • 10BASE-T Full Duplex and 10BASE-T Half Duplex bits of the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV) • PHY Mode (MODE[2:0]) bits of the PHY Special Modes Register (PHY_SPECIAL_MODES) **Note:** This has no effect when the PHY is in 100BASE-FX mode. Refer to the respective register definition sections for additional information.	1b

TABLE 7-1: SOFT-STRAP CONFIGURATION STRAP DEFINITIONS (CONTINUED)

Strap Name	Description	Pin / Default Value
speed_strap_1	**PHY Speed Select Strap:** This strap affects the default value of the following register bits: • Speed Select LSB (PHY_SPEED_SEL_LSB) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) • 10BASE-T Full Duplex and 10BASE-T Half Duplex bits of the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV) • PHY Mode (MODE[2:0]) bits of the PHY Special Modes Register (PHY_SPECIAL_MODES) **Note:** This has no effect when the PHY is in 100BASE-FX mode. Refer to the respective register definition sections for additional information.	1b
duplex_strap_1	**PHY Duplex Select Strap:** This strap affects the default value of the following register bits: • Duplex Mode (PHY_DUPLEX) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) • 10BASE-T Full Duplex bit of the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV) • PHY Mode (MODE[2:0]) bits of the PHY Special Modes Register (PHY_SPECIAL_MODES) Refer to the respective register definition sections for additional information.	1b
FD_FC_strap_1	**PHY Full-Duplex Flow Control Enable Strap:** This strap affects the default value of the following register bits: • Asymmetric Pause bit of the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV) **Note:** This has no effect when the PHY is in 100BASE-FX mode. Refer to the respective register definition sections for additional information.	1b
manual_FC_strap_1	**PHY Manual Flow Control Enable Strap:** This strap affects the default value of the following register bits: • Asymmetric Pause and Symmetric Pause bit of the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV) **Note:** This has no effect when the PHY is in 100BASE-FX mode. Refer to the respective register definition sections for additional information.	0b

TABLE 7-1: SOFT-STRAP CONFIGURATION STRAP DEFINITIONS (CONTINUED)

Strap Name	Description	Pin / Default Value
EEE_enable_strap_1	**Host MAC Energy Efficient Ethernet Enable Strap:** Configures the default value of the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Section 11.15.1, "Host MAC Control Register (HMAC_CR)," on page 193. Refer to the respective register definition sections for additional information.	1b
EEE_enable_strap_1 (cont.)	**PHY Energy Efficient Ethernet Enable Strap:** This strap affects the default value of the following register bits: • PHY Energy Efficient Ethernet Enable (PHYEEEEN) bit of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG) • 100BASE-TX EEE bit of the PHY EEE Capability Register (PHY_EEE_CAP) • 100BASE-TX EEE bit of the PHY EEE Advertisement Register (PHY_EEE_ADV) **Note:** This has no effect when the PHY is in 100BASE-FX mode. Refer to the respective register definition sections for additional information.	1b

7.2 Hard-Straps

Hard-straps are latched upon Power-On Reset (POR) or pin reset (**RST#**) only. Unlike soft-straps, hard-straps always have an associated pin and cannot be overridden by the EEPROM Loader. These straps are used as either direct configuration values or as register defaults. Table 7-2 provides a list of all hard-straps and their associated pin. These straps, along with their pin assignments are also fully defined in Section 3.0, "Pin Descriptions and Configuration," on page 10.

TABLE 7-2: HARD-STRAP CONFIGURATION STRAP DEFINITIONS

Strap Name	Description	Pins
eeprom_size_strap	**EEPROM Size Strap:** Configures the EEPROM size range. A low selects 1K bits (128 x 8) through 16K bits (2K x 8). A high selects 32K bits (4K x 8) through 512K bits (64K x 8).	**E2PSIZE**

TABLE 7-2: HARD-STRAP CONFIGURATION STRAP DEFINITIONS (CONTINUED)

<table>
<tr><th>Strap Name</th><th>Description</th><th>Pins</th></tr>
<tr><td>host_intf_mode_strap</td><td>Host Interface Mode Strap: Configures the host management mode.

0 = SPI Mode
1 = HBI Mode

The operating mode results from the following mapping:

<table><tr><th>MNGT1 : MNGT0</th><th>mngt_mode_strap</th></tr><tr><td>00</td><td>0 (SPI)</td></tr><tr><td>01, 10 or 11</td><td>1 (HBI)</td></tr></table>
See Table 7-3 for the combined host interface strapping.

Note: Refer to Section 2.0, "General Description," on page 8 for additional information on the various modes of the device.</td><td>MNGT1 :
MNGT0</td></tr>
<tr><td>HBI_addr_mode_strap</td><td>HBI Address Mode Strap: Configures the HBI interface for multiplexed or non-multiplexed indexed addressing modes.

0 = multiplexed
1 = non-multiplexed indexed

See Table 7-3 for the combined host interface strapping.

Note: Refer to Section 2.0, "General Description," on page 8 for additional information on the various modes of the device.</td><td>MNGT1</td></tr>
<tr><td>HBI_addr_phase_strap</td><td>HBI Address Phase Strap: Configures the number of address cycles for the HBI interface when in multiplexed address mode.

0 = single phase multiplexed
1 = dual phase multiplexed

See Table 7-3 for the combined host interface strapping.

Note: Refer to Section 2.0, "General Description," on page 8 for additional information on the various modes of the device.</td><td>MNGT3</td></tr>
</table>

TABLE 7-2: HARD-STRAP CONFIGURATION STRAP DEFINITIONS (CONTINUED)

<table>
<tr><th>Strap Name</th><th>Description</th><th>Pins</th></tr>
<tr><td>HBI_data_mode_strap[1:0]</td><td>HBI Data Mode Straps: Configures the data width of the HBI interface.

00 = 8-bit
01 = 16-bit
1X = RESERVED

The data mode results from the following mapping:

<table>
<tr><th><u>MNGT2</u> : <u>MNGT1</u> : <u>MNGT0</u></th><th>HBI_data_mode_strap</th></tr>
<tr><td>X00 (SPI)</td><td>reserved</td></tr>
<tr><td>001 (HBI multiplexed)</td><td>00 (8-bit)</td></tr>
<tr><td>101 (HBI multiplexed)</td><td>01 (16-bit)</td></tr>
<tr><td>X10 (HBI non-multi-plexed indexed)</td><td>00 (8-bit)</td></tr>
<tr><td>X11 (HBI non-multi-plexed indexed)</td><td>01 (16-bit)</td></tr>
</table>

Note: Effectively, the data mode is determined by <u>MNGT2</u> in multiplexed mode and by <u>MNGT0</u> in non-multiplexed indexed mode.

See Table 7-3 for the combined host interface strapping.

Note: Refer to Section 2.0, "General Description," on page 8 for additional information on the various modes of the device.</td><td><u>MNGT2</u> :
<u>MNGT1</u> :
<u>MNGT0</u></td></tr>
<tr><td>fx_mode_strap_1</td><td>PHY FX Mode Strap: Selects FX mode for the PHY.

This strap is set high when <u>FXLOSEN</u> is above 1 V (typ.) or <u>FXSDENA</u> is above 1 V (typ.).</td><td><u>FXLOSEN</u> :
<u>FXSDENA</u></td></tr>
<tr><td>fx_los_strap_1</td><td>PHY FX-LOS Select Strap: Selects Loss of Signal mode for the PHY.

This strap is set high when <u>FXLOSEN</u> is above 1 V (typ.).</td><td><u>FXLOSEN</u></td></tr>
</table>

Note 1: The combined host interface strap chart is as follows:

TABLE 7-3: HBI STRAP MAPPING

MNGT1	MNGT0	MNGT3	MNGT2	Host Mode
0	0	X	X	SPI
0	1	0	0	HBI Multiplexed 1 Phase 8-bit
0	1	0	1	HBI Multiplexed 1 Phase 16-bit
0	1	1	0	HBI Multiplexed 2 Phase 8-bit
0	1	1	1	HBI Multiplexed 2 Phase 16-bit
1	0	X	X	HBI Indexed 8-bit
1	1	X	X	HBI Indexed 16-bit

8.0 SYSTEM INTERRUPTS

8.1 Functional Overview

This chapter describes the system interrupt structure of the device. The device provides a multi-tier programmable interrupt structure which is controlled by the System Interrupt Controller. The programmable system interrupts are generated internally by the various device sub-modules and can be configured to generate a single external host interrupt via the **IRQ** interrupt output pin. The programmable nature of the host interrupt provides the user with the ability to optimize performance dependent upon the application requirements. The **IRQ** interrupt buffer type, polarity and de-assertion interval are modifiable. The **IRQ** interrupt can be configured as an open-drain output to facilitate the sharing of interrupts with other devices. All internal interrupts are maskable and capable of triggering the **IRQ** interrupt.

8.2 Interrupt Sources

The device is capable of generating the following interrupt types:

- 1588 Interrupts
- Ethernet PHY Interrupts
- GPIO Interrupts
- Host MAC Interrupts (FIFOs)
- Power Management Interrupts
- General Purpose Timer Interrupt (GPT)
- Software Interrupt (General Purpose)
- Device Ready Interrupt
- Clock Output Test Mode

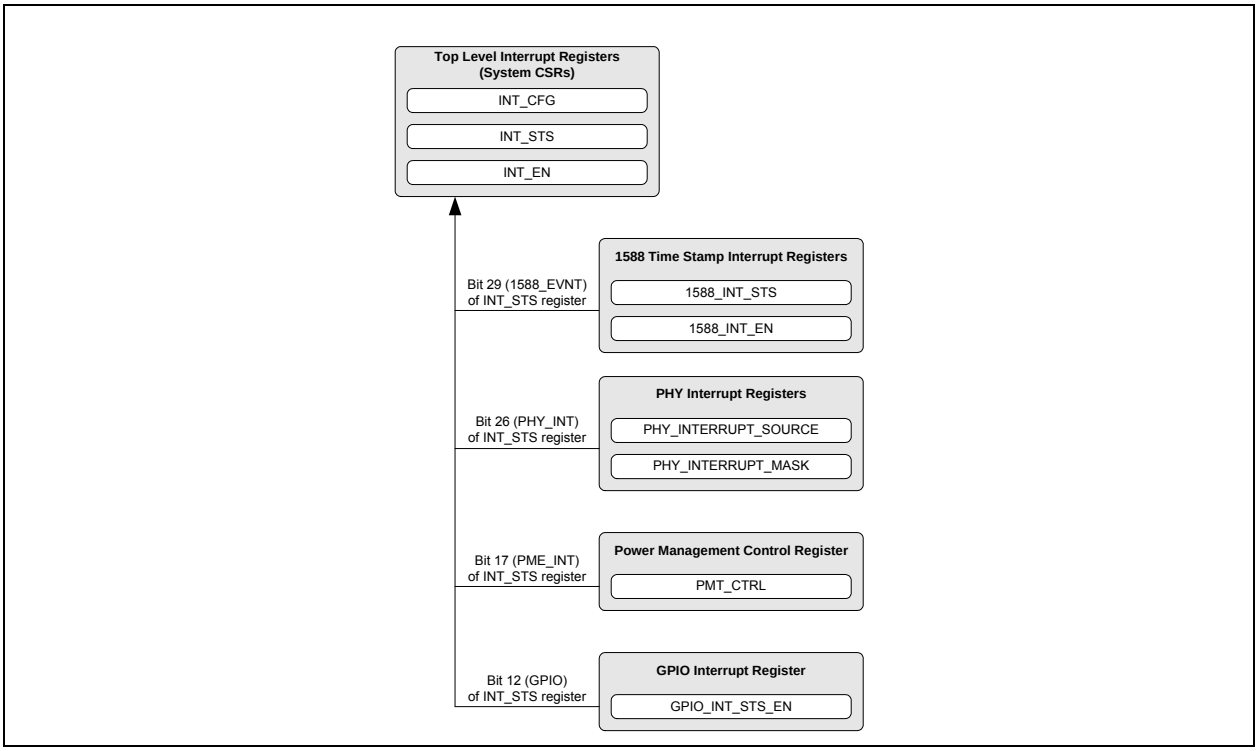
All interrupts are accessed and configured via registers arranged into a multi-tier, branch-like structure, as shown in Figure 8-1. At the top level of the device interrupt structure are the Interrupt Status Register (INT_STS), Interrupt Enable Register (INT_EN) and Interrupt Configuration Register (IRQ_CFG).

The Interrupt Status Register (INT_STS) and Interrupt Enable Register (INT_EN) aggregate and enable/disable all interrupts from the various device sub-modules, combining them together to create the **IRQ** interrupt. These registers provide direct interrupt access/configuration to the Host MAC, General Purpose Timer, software and device ready interrupts. These interrupts can be monitored, enabled/disabled and cleared, directly within these two registers. In addition, event indications are provided for the 1588, Power Management, GPIO and Ethernet PHY interrupts. These interrupts differ in that the interrupt sources are generated and cleared in other sub-block registers. The INT_STS register does not provide details on what specific event within the sub-module caused the interrupt and requires the software to poll an additional sub-module interrupt register (as shown in Figure 8-1) to determine the exact interrupt source and clear it. For interrupts which involve multiple registers, only after the interrupt has been serviced and cleared at its source will it be cleared in the INT_STS register.

The Interrupt Configuration Register (IRQ_CFG) is responsible for enabling/disabling the **IRQ** interrupt output pin as well as configuring its properties. The IRQ_CFG register allows the modification of the **IRQ** pin buffer type, polarity and de-assertion interval. The de-assertion timer guarantees a minimum interrupt de-assertion period for the **IRQ** output and is programmable via the Interrupt De-assertion Interval (INT_DEAS) field of the Interrupt Configuration Register (IRQ_CFG). A setting of all zeros disables the de-assertion timer. The de-assertion interval starts when the **IRQ** pin de-asserts, regardless of the reason.

Note:	The de-assertion timer does not apply to the PME interrupt. The PME interrupt is ORed into the IRQ logic following the deassertion timer gating. Assertion of the PME interrupt does not affect the de-assertion timer.

FIGURE 8-1: FUNCTIONAL INTERRUPT HIERARCHY



The following sections detail each category of interrupts and their related registers. Refer to the corresponding function's chapter for bit-level definitions of all interrupt registers.

8.2.1 1588 INTERRUPTS

Multiple 1588 Time Stamp interrupt sources are provided by the device. The top-level 1588 Interrupt Event (1588_EVNT) bit of the Interrupt Status Register (INT_STS) provides indication that a 1588 interrupt event occurred in the 1588 Interrupt Status Register (1588_INT_STS).

The 1588 Interrupt Enable Register (1588_INT_EN) provides enabling/disabling of all 1588 interrupt conditions. The 1588 Interrupt Status Register (1588_INT_STS) provides the status of all 1588 interrupts. These include TX/RX 1588 clock capture indication, 1588 clock capture for GPIO events, as well as 1588 timer interrupt indication.

In order for a 1588 interrupt event to trigger the external IRQ interrupt pin, the desired 1588 interrupt event must be enabled in the 1588 Interrupt Enable Register (1588_INT_EN), bit 29 (1588_EVNT_EN) of the Interrupt Enable Register (INT_EN) must be set and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the of the Interrupt Configuration Register (IRQ_CFG).

For additional details on the 1588 Time Stamp interrupts, refer to Section 14.0, "IEEE 1588," on page 298.

8.2.2 ETHERNET PHY INTERRUPTS

The top-level PHY Interrupt Event (PHY_INT) bit of the Interrupt Status Register (INT_STS) provides indication that a PHY interrupt event occurred in the PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE).

PHY interrupts are enabled/disabled via their respective PHY Interrupt Mask Register (PHY_INTERRUPT_MASK). The source of a PHY interrupt can be determined and cleared via the PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE). Unique interrupts are generated based on the following events:

- ENERGYON Activated
- Auto-Negotiation Complete
- Remote Fault Detected
- Link Down (Link Status Negated)

- Link Up (Link Status Asserted)
- Auto-Negotiation LP Acknowledge
- Parallel Detection Fault
- Auto-Negotiation Page Received

In order for an interrupt event to trigger the external **IRQ** interrupt pin, the desired PHY interrupt event must be enabled in the corresponding PHY Interrupt Mask Register (PHY_INTERRUPT_MASK), the PHY Interrupt Event Enable (PHY_INT_EN) bit of the Interrupt Enable Register (INT_EN) must be set and the IRQ output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

For additional details on the Ethernet PHY interrupts, refer to Section 12.2.9, "PHY Interrupts," on page 220.

8.2.3 GPIO INTERRUPTS

Each GPIO of the device is provided with its own interrupt. The top-level GPIO Interrupt Event (GPIO) bit of the Interrupt Status Register (INT_STS) provides indication that a GPIO interrupt event occurred in the General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN). The General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN) provides enabling/disabling and status of each GPIO interrupt.

In order for a GPIO interrupt event to trigger the external **IRQ** interrupt pin, the desired GPIO interrupt must be enabled in the General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN), the GPIO Interrupt Event Enable (GPIO_EN) bit of the Interrupt Enable Register (INT_EN) must be set and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

For additional details on the GPIO interrupts, refer to Section 16.2.1, "GPIO Interrupts," on page 384.

8.2.4 HOST MAC INTERRUPTS

The top-level Interrupt Status Register (INT_STS) and Interrupt Enable Register (INT_EN) provide the status and enabling/disabling of multiple Host MAC related interrupts. All Host MAC interrupts are monitored and configured directly within these two registers. The following Host MAC related interrupt events are supported:

- TX Stopped
- RX Stopped
- RX Dropped Frame Counter Halfway
- TX IOC
- RX DMA
- TX Status FIFO Overflow
- Receive Watchdog Time-Out
- Receiver Error
- Transmitter Error
- TX Data FIFO Overrun
- TX Data FIFO Available
- TX Status FIFO Full
- TX Status FIFO Level
- RX Dropped Frame
- RX Status FIFO Full
- RX Status FIFO Level

In order for a Host MAC interrupt event to trigger the external **IRQ** interrupt pin, the desired Host MAC interrupt event must be enabled in the Interrupt Enable Register (INT_EN) and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

Refer to the Interrupt Status Register (INT_STS) on page 69 and Section 11.0, "Host MAC," on page 139 for additional information on bit definitions and Host MAC operation.

8.2.5 POWER MANAGEMENT INTERRUPTS

Multiple Power Management Event interrupt sources are provided by the device. The top-level Power Management Interrupt Event (PME_INT) bit of the Interrupt Status Register (INT_STS) provides indication that a Power Management interrupt event occurred in the Power Management Control Register (PMT_CTRL).

The Power Management Control Register (PMT_CTRL) provides enabling/disabling and status of all Power Management conditions. These include energy-detect on the PHY and Wake-On-LAN (Perfect DA, Broadcast, Wake-up frame or Magic Packet) detection by the Host MAC.

In order for a Power Management interrupt event to trigger the external **IRQ** interrupt pin, the desired Power Management interrupt event must be enabled in the Power Management Control Register (PMT_CTRL), the Power Management Event Interrupt Enable (PME_INT_EN) bit of the Interrupt Enable Register (INT_EN) must be set and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit 8 of the Interrupt Configuration Register (IRQ_CFG).

The power management interrupts are only a portion of the power management features of the device. For additional details on power management, refer to Section 6.3, "Power Management," on page 44.

8.2.6 GENERAL PURPOSE TIMER INTERRUPT

A GP Timer (GPT_INT) interrupt is provided in the top-level Interrupt Status Register (INT_STS) and Interrupt Enable Register (INT_EN). This interrupt is issued when the General Purpose Timer Count Register (GPT_CNT) wraps past zero to FFFFh and is cleared when the GP Timer (GPT_INT) bit of the Interrupt Status Register (INT_STS) is written with 1.

In order for a General Purpose Timer interrupt event to trigger the external **IRQ** interrupt pin, the GPT must be enabled via the General Purpose Timer Enable (TIMER_EN) bit in the General Purpose Timer Configuration Register (GPT_CFG), the GP Timer Interrupt Enable (GPT_INT_EN) bit of the Interrupt Enable Register (INT_EN) must be set and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

For additional details on the General Purpose Timer, refer to Section 15.1, "General Purpose Timer," on page 380.

8.2.7 SOFTWARE INTERRUPT

A general purpose software interrupt is provided in the top level Interrupt Status Register (INT_STS) and Interrupt Enable Register (INT_EN). The Software Interrupt (SW_INT) bit of the Interrupt Status Register (INT_STS) is generated when the Software Interrupt Enable (SW_INT_EN) bit of the Interrupt Enable Register (INT_EN) changes from cleared to set (i.e. on the rising edge of the enable). This interrupt provides an easy way for software to generate an interrupt and is designed for general software usage.

In order for a Software interrupt event to trigger the external **IRQ** interrupt pin, the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

8.2.8 DEVICE READY INTERRUPT

A device ready interrupt is provided in the top-level Interrupt Status Register (INT_STS) and Interrupt Enable Register (INT_EN). The Device Ready (READY) bit of the Interrupt Status Register (INT_STS) indicates that the device is ready to be accessed after a power-up or reset condition. Writing a 1 to this bit in the Interrupt Status Register (INT_STS) will clear it.

In order for a device ready interrupt event to trigger the external **IRQ** interrupt pin, the Device Ready Enable (READY_EN) bit of the Interrupt Enable Register (INT_EN) must be set and the **IRQ** output must be enabled via the IRQ Enable (IRQ_EN) bit of the Interrupt Configuration Register (IRQ_CFG).

8.2.9 CLOCK OUTPUT TEST MODE

In order to facilitate system level debug, the crystal clock can be enabled onto the **IRQ** pin by setting the IRQ Clock Select (IRQ_CLK_SELECT) bit of the Interrupt Configuration Register (IRQ_CFG).

The **IRQ** pin should be set to a push-pull driver by using the IRQ Buffer Type (IRQ_TYPE) bit for the best result.

8.3 Interrupt Registers

This section details the directly addressable interrupt related System CSRs. These registers control, configure and monitor the **IRQ** interrupt output pin and the various device interrupt sources. For an overview of the entire directly addressable register map, refer to Section 5.0, "Register Map," on page 29.

TABLE 8-1: INTERRUPT REGISTERS

ADDRESS	REGISTER NAME (SYMBOL)
054h	Interrupt Configuration Register (IRQ_CFG)

TABLE 8-1: INTERRUPT REGISTERS (CONTINUED)

ADDRESS	REGISTER NAME (SYMBOL)
058h	Interrupt Status Register (INT_STS)
05Ch	Interrupt Enable Register (INT_EN)

8.3.1 INTERRUPT CONFIGURATION REGISTER (IRQ_CFG)

Offset: 054h Size: 32 bits

This read/write register configures and indicates the state of the IRQ signal.

Bits	Description	Type	Default
31:24	**Interrupt De-assertion Interval (INT_DEAS)** This field determines the Interrupt Request De-assertion Interval in multiples of 10 microseconds. Setting this field to zero causes the device to disable the INT_DEAS Interval, reset the interval counter and issue any pending interrupts. If a new, non-zero value is written to this field, any subsequent interrupts will obey the new setting. This field does not apply to the PME_INT interrupt.	R/W	00h
23:15	**RESERVED**	RO	-
14	**Interrupt De-assertion Interval Clear (INT_DEAS_CLR)** Writing a 1 to this register clears the de-assertion counter in the Interrupt Controller, thus causing a new de-assertion interval to begin (regardless of whether or not the Interrupt Controller is currently in an active de-assertion interval). **0: Normal operation** **1: Clear de-assertion counter**	R/W SC	0h
13	**Interrupt De-assertion Status (INT_DEAS_STS)** When set, this bit indicates that the interrupt controller is currently in a de-assertion interval and potential interrupts will not be sent to the IRQ pin. When this bit is clear, the interrupt controller is not currently in a de-assertion interval and interrupts will be sent to the IRQ pin. **0: Interrupt controller not in de-assertion interval** **1: Interrupt controller in de-assertion interval**	RO	0b
12	**Master Interrupt (IRQ_INT)** This read-only bit indicates the state of the internal IRQ line, regardless of the setting of the IRQ_EN bit, or the state of the interrupt de-assertion function. When this bit is set, one of the enabled interrupts is currently active. **0: No enabled interrupts active** **1: One or more enabled interrupts active**	RO	0b
11:9	**RESERVED**	RO	-
8	**IRQ Enable (IRQ_EN)** This bit controls the final interrupt output to the IRQ pin. When clear, the IRQ output is disabled and permanently de-asserted. This bit has no effect on any internal interrupt status bits. **0: Disable output on IRQ pin** **1: Enable output on IRQ pin**	R/W	0b
7:5	**RESERVED**	RO	-

Bits	Description	Type	Default
4	**IRQ Polarity (IRQ_POL)** When cleared, this bit enables the IRQ line to function as an active low output. When set, the IRQ output is active high. When the IRQ is configured as an open-drain output (via the IRQ_TYPE bit), this bit is ignored and the interrupt is always active low. **0: IRQ active low output** **1: IRQ active high output**	R/W NASR Note 1	0b
3:2	**RESERVED**	RO	-
1	**IRQ Clock Select (IRQ_CLK_SELECT)** When this bit is set, the crystal clock may be output on the IRQ pin. This is intended to be used for system debug purposes in order to observe the clock and not for any functional purpose. **Note:** When using this bit, the IRQ pin should be set to a push-pull driver.	R/W	0b
0	**IRQ Buffer Type (IRQ_TYPE)** When this bit is cleared, the IRQ pin functions as an open-drain output for use in a wired-or interrupt configuration. When set, the IRQ is a push-pull driver. **Note:** When configured as an open-drain output, the IRQ_POL bit is ignored and the interrupt output is always active low. **0: IRQ pin open-drain output** **1: IRQ pin push-pull driver**	R/W NASR Note 1	0b

Note 1: Register bits designated as NASR are not reset when the DIGITAL_RST bit in the Reset Control Register (RESET_CTL) is set.

8.3.2 INTERRUPT STATUS REGISTER (INT_STS)

Offset: 058h Size: 32 bits

This register contains the current status of the generated interrupts. A value of 1 indicates the corresponding interrupt conditions have been met, while a value of 0 indicates the interrupt conditions have not been met. The bits of this register reflect the status of the interrupt source regardless of whether the source has been enabled as an interrupt in the Interrupt Enable Register (INT_EN). Where indicated as R/WC, writing a 1 to the corresponding bits acknowledges and clears the interrupt.

Bits	Description	Type	Default
31	**Software Interrupt (SW_INT)** This interrupt is generated when the Software Interrupt Enable (SW_INT_EN) bit of the Interrupt Enable Register (INT_EN) is set high. Writing a one clears this interrupt.	R/WC	0b
30	**Device Ready (READY)** This interrupt indicates that the device is ready to be accessed after a power-up or reset condition.	R/WC	0b
29	**1588 Interrupt Event (1588_EVNT)** This bit indicates an interrupt event from the IEEE 1588 module. This bit should be used in conjunction with the 1588 Interrupt Status Register (1588_INT_STS) to determine the source of the interrupt event within the 1588 module.	RO	0b
28	**RESERVED**	RO	-
27	**RESERVED**	RO	-
26	**PHY Interrupt Event (PHY_INT)** This bit indicates an interrupt event from PHY. The source of the interrupt can be determined by polling the PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE).	RO	0b
25	**TX Stopped (TXSTOP_INT)** This interrupt is issued when the Stop Transmitter (STOP_TX) bit in Transmit Configuration Register (TX_CFG) is set and the Host MAC transmitter is halted.	R/WC	0b
24	**RX Stopped (RXSTOP_INT)** This interrupt is issued when the Receiver Enable (RXEN) bit in Host MAC Control Register (HMAC_CR) is cleared and the Host MAC receiver is halted.	R/WC	0b
23	**RX Dropped Frame Counter Halfway (RXDFH_INT)** This interrupt is issued when the Host MAC RX Dropped Frames Counter Register (RX_DROP) counts past its halfway point (7FFFFFFFh to 80000000h).	R/WC	0b
22	**RESERVED**	RO	-
21	**TX IOC Interrupt (TX_IOC)** This interrupt is generated when a buffer with the IOC flag set has been fully loaded into the TX Data FIFO.	R/WC	0b

Bits	Description	Type	Default
20	**RX DMA Interrupt (RXD_INT)** This interrupt is issued when the amount of data programmed in the RX DMA Count (RX_DMA_CNT) field of the Receive Configuration Register (RX_CFG) has been transferred out of the RX Data FIFO.	R/WC	0b
19	**GP Timer (GPT_INT)** This interrupt is issued when the General Purpose Timer Count Register (GPT_CNT) wraps past zero to FFFFh.	R/WC	0b
18	**RESERVED**	RO	-
17	**Power Management Interrupt Event (PME_INT)** This interrupt is issued when a Power Management Event is detected as configured in the Power Management Control Register (PMT_CTRL). This interrupt functions independent of the PME signal and will still function if the PME signal is disabled. Writing a '1' clears this bit regardless of the state of the PME hardware signal. In order to clear this bit, all unmasked bits in the Power Management Control Register (PMT_CTRL) must first be cleared. **Note:** The Interrupt De-assertion interval does not apply to the PME interrupt.	R/WC	0b
16	**TX Status FIFO Overflow (TXSO)** This interrupt is generated when the TX Status FIFO overflows.	R/WC	0b
15	**Receive Watchdog Time-out (RWT)** This interrupt is generated when a frame greater than or equal to 2048 bytes has been received by the Host MAC. Frames greater than or equal to 2049 bytes are truncated to 2048 bytes.	R/WC	0b
14	**Receiver Error (RXE)** Indicates that the Host MAC receiver has encountered an error. Please refer to Section 11.12.5, "Receiver Errors," on page 174 for a description of the conditions that will cause an RXE.	R/WC	0b
13	**Transmitter Error (TXE)** When generated, indicates that the Host MAC transmitter has encountered an error. Please refer to Section 11.11.7, "Transmitter Errors," on page 170 for a description of the conditions that will cause a TXE.	R/WC	0b
12	**GPIO Interrupt Event (GPIO)** This bit indicates an interrupt event from the General Purpose I/O. The source of the interrupt can be determined by polling the General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN)	RO	0b
11	**RESERVED**	RO	-
10	**TX Data FIFO Overrun Interrupt (TDFO)** This interrupt is generated when the TX Data FIFO is full and another write is attempted.	R/WC	0b
9	**TX Data FIFO Available Interrupt (TDFA)** This interrupt is generated when the TX Data FIFO available space is greater than the programmed level in the TX Data Available Level field of the FIFO Level Interrupt Register (FIFO_INT).	R/WC	0b
8	**TX Status FIFO Full Interrupt (TSFF)** This interrupt is generated when the TX Status FIFO is full.	R/WC	0b

Bits	Description	Type	Default
7	**TX Status FIFO Level Interrupt (TSFL)** This interrupt is generated when the TX Status FIFO reaches the programmed level in the TX Status Level field of the FIFO Level Interrupt Register (FIFO_INT).	R/WC	0b
6	**RX Dropped Frame Interrupt (RXDF_INT)** This interrupt is issued whenever a receive frame is dropped by the Host MAC.	R/WC	0b
5	**RESERVED**	RO	-
4	**RX Status FIFO Full Interrupt (RSFF)** This interrupt is generated when the RX Status FIFO is full and another status write is attempted by the device.	R/WC	0b
3	**RX Status FIFO Level Interrupt (RSFL)** This interrupt is generated when the RX Status FIFO reaches the programmed level in the RX Status Level field of the FIFO Level Interrupt Register (FIFO_INT).	R/WC	0b
2:1	**RESERVED**	RO	-
0	**RESERVED**	RO	-

8.3.3 INTERRUPT ENABLE REGISTER (INT_EN)

Offset: 05Ch Size: 32 bits

This register contains the interrupt enables for the **IRQ** output pin. Writing 1 to any of the bits enables the corresponding interrupt as a source for **IRQ**. Bits in the Interrupt Status Register (INT_STS) register will still reflect the status of the interrupt source regardless of whether the source is enabled as an interrupt in this register (with the exception of Software Interrupt Enable (SW_INT_EN). For descriptions of each interrupt, refer to the Interrupt Status Register (INT_STS) bits, which mimic the layout of this register.

Bits	Description	Type	Default
31	**Software Interrupt Enable (SW_INT_EN)**	R/W	0b
30	**Device Ready Enable (READY_EN)**	R/W	0b
29	**1588 Interrupt Event Enable (1588_EVNT_EN)**	R/W	0b
28:27	**RESERVED**	RO	-
26	**PHY Interrupt Event Enable (PHY_INT_EN)**	R/W	0b
25	**TX Stopped Interrupt Enable (TXSTOP_INT_EN)**	R/W	0b
24	**RX Stopped Interrupt Enable (RXSTOP_INT_EN)**	R/W	0b
23	**RX Dropped Frame Counter Halfway Interrupt Enable (RXDFH_INT_EN)**	R/W	0b
22	**RESERVED**	RO	-
21	**TX IOC Interrupt Enable (TIOC_INT_EN)**	R/W	0b
20	**RX DMA Interrupt Enable (RXD_INT_EN)**	R/W	0b
19	**GP Timer Interrupt Enable (GPT_INT_EN)**	R/W	0b
18	**RESERVED**	RO	-
17	**Power Management Event Interrupt Enable (PME_INT_EN)**	R/W	0b
16	**TX Status FIFO Overflow Interrupt Enable (TXSO_EN)**	R/W	0b
15	**Receive Watchdog Time-out Interrupt Enable (RWT_INT_EN)**	R/W	0b
14	**Receiver Error Interrupt Enable (RXE_INT_EN)**	R/W	0b
13	**Transmitter Error Interrupt Enable (TXE_INT_EN)**	R/W	0b
12	**GPIO Interrupt Event Enable (GPIO_EN)**	R/W	0b
11	**RESERVED**	RO	-
10	**TX Data FIFO Overrun Interrupt Enable (TDFO_EN)**	R/W	0b
9	**TX Data FIFO Available Interrupt Enable (TDFA_EN)**	R/W	0b
8	**TX Status FIFO Full Interrupt Enable (TSFF_EN)**	R/W	0b
7	**TX Status FIFO Level Interrupt Enable (TSFL_EN)**	R/W	0b
6	**RX Dropped Frame Interrupt Enable (RXDF_INT_EN)**	R/W	0b

Bits	Description	Type	Default
5	**RESERVED**	RO	-
4	**RX Status FIFO Full Interrupt Enable (RSFF_EN)**	R/W	0b
3	**RX Status FIFO Level Interrupt Enable (RSFL_EN)**	R/W	0b
2:1	**RESERVED**	RO	-
0	**RESERVED**	RO	-

9.0 HOST BUS INTERFACE

9.1 Functional Overview

The Host Bus Interface (HBI) module provides a high-speed asynchronous slave interface that facilitates communication between the device and a host system. The HBI allows access to the System CSRs and internal FIFOs and memories and handles byte swapping based on the endianness select.

The following is an overview of the functions provided by the HBI:

- **Address bus input:** Two addressing modes are supported. These are a multiplexed address / data bus and a demultiplexed address bus with address index register accesses. The mode selection is done through a configuration input.
- **Selectable data bus width:** The host data bus width is selectable. 16 and 8-bit data modes are supported. This selection is done through a configuration input. The HBI performs BYTE and WORD to DWORD assembly on write data and keeps track of the BYTE / WORD count for reads. Individual BYTE access in 16-bit mode is not supported.
- **Selectable read / write control modes:** Two control modes are available. Separate read and write pins or an enable and direction pin. The mode selection is done through a configuration input.
- **Selectable control line polarity:** The polarity of the chip select, read / write and address latch signals is selectable through configuration inputs.
- **Dynamic Endianness control:** The HBI supports the selection of big and little endian host byte ordering based on the endianness signal. This highly flexible interface provides mixed endian access for registers and memory. Depending on the addressing mode of the device, this signal is either configuration register controlled or as part of the strobed address input.
- **Direct FIFO access:** A FIFO direct select signal directs all host write operations to the TX Data FIFO and all host read operations from the RX Data FIFO . Depending on the addressing mode of the device, this signal is either directly provided by the host or is strobed as part of the address input.

  When used with the Indexed Addressing mode, burst read access is supported by toggling the lower address bits.

9.2 Read / Write Control Signals

The device supports two distinct read / write signal methods:

- read (**RD**) and write (**WR**) strobes are input on separate pins.
- read and write signals are decoded from an enable input (**ENB**) and a direction input (**RD_WR**).

9.3 Control Line Polarity

The device supports polarity control on the following:

- chip select input (**CS**)
- read strobe (**RD**) / direction input (**RD_WR**)
- write strobe (**WR**) / enable input (**ENB**)
- address latch control (**ALELO** and **ALEHI**)

9.4 Multiplexed Address / Data Mode

In Multiplexed Address / Data mode, the address, FIFO Direct Select and endianness select inputs are shared with the data bus. Two methods are supported, a single phase address, utilizing up to 16 address / data pins and a dual phase address, utilizing only the lower 8 data bits.

9.4.1 ADDRESS LATCH CYCLES

9.4.1.1 Single Phase Address Latching

In Single Phase mode, all address bits, the FIFO Direct Select signal and the endianness select are strobed into the device using the trailing edge of the **ALELO** signal. The address latch is implemented on all 16 address / data pins. In 8-bit data mode, where pins **AD[15:8]** are used exclusively for addressing, it is not necessary to drive these upper address lines with a valid address continually through read and write operations. However, this operation, referred to as Partial Address Multiplexing, is acceptable since the device will never drive these pins.

Qualification of the **ALELO** signal with the **CS** signal is selectable. When qualification is enabled, **CS** must be active during **ALELO** in order to strobe the address inputs. When qualification is not enabled, **CS** is a don't care during the address phase.

The address is retained for all future read and write operations. It is retained until either a reset event occurs or a new address is loaded. This allows multiple read and write requests to take place to the same address, without requiring multiple address latching operations.

9.4.1.2 Dual Phase Address Latching

In Dual Phase mode, the lower 8 address bits are strobed into the device using the inactive going edge of the **ALELO** signal and the remaining upper address bits, the FIFO Direct Select signals and the endianness select are strobed into the device using the trailing edge of the **ALEHI** signal. The strobes can be in either order. In 8-bit data mode, pins **AD[15:8]** are not used. In 16-bit data mode, pins **D[15:8]** are used only for data.

Qualification of the **ALELO** and **ALEHI** signals with the CS signal is selectable. When qualification is enabled, **CS** must be active during **ALELO** and **ALEHI** in order to strobe the address inputs. When qualification is not enabled, **CS** is a don't care during the address phase.

The address is retained for all future read and write operations. It is retained until either a reset event occurs or a new address is loaded. This allows multiple read and write requests to take place to the same address, without requiring multiple address latching operations.

9.4.1.3 Address Bit to Address / Data Pin Mapping

In 8-bit data mode, address bit 0 is multiplexed onto pin **AD[0]**, address bit 1 onto pin **AD[1]**, etc. The highest address bit is bit 9 and is multiplexed onto pin **AD[9]** (single phase) or **AD[1]** (dual phase). The address latched into the device is considered a BYTE address and covers 1K bytes (0 to 3FFh).

In 16-bit data mode, address bit 1 is multiplexed onto pin **AD[0]**, address bit 2 onto pin **AD[1]**, etc. The highest address bit is bit 9 and is multiplexed onto pin **AD[8]** (single phase) or **AD[0]** (dual phase). The address latched into the device is considered a WORD address and covers 512 words (0 to 1FFh).

When the address is sent to the rest of the device, it is converted to a BYTE address.

9.4.1.4 Endianness Select to Address / Data Pin Mapping

The endianness select is included into the multiplexed address to allow the host system to dynamically select the endianness based on the memory address used. This allows for mixed endian access for registers and memory.

The endianness selection is multiplexed to the data pin one bit above the last address bit.

9.4.1.5 FIFO Direct Select to Address / Data Pin Mapping

The FIFO Direct Select signal is included into the multiplexed address to allow the host system to address the TX and RX Data FIFOs as if they were a large flat address space.

The FIFO Direct Select signal is multiplexed to the data pin two bits above the last address bit.

9.4.2 DATA CYCLES

The host data bus can be 16 or 8-bits wide while all internal registers are 32 bits wide. The Host Bus Interface performs the conversion from WORDs or BYTEs to DWORD, while in 8 or 16-bit data mode. Two or four contiguous accesses within the same DWORD are required in order to perform a write or read.

9.4.2.1 Write Cycles

A write cycle occurs when **CS** and **WR** are active (or when **ENB** is active with **RD_WR** indicating write). The host address and endianness were already captured during the address latch cycle.

On the trailing edge of the write cycle (either **WR** or **CS** or **ENB** going inactive), the host data is captured into registers in the HBI. Depending on the bus width, either a WORD or a BYTE is captured. For 8 or 16-bit data modes, this functions as the DWORD assembly with the affected WORD or BYTE determined by the lower address inputs. BYTE swapping is also done at this point based on the endianness.

WRITES FOLLOWING INITIALIZATION

Following device initialization, writes from the Host Bus are ignored until after a read cycle is performed.

WRITES DURING AND FOLLOWING POWER MANAGEMENT

During and following any power management mode other than D0, writes from the Host Bus are ignored until after a read cycle is performed.

8 AND 16-BIT ACCESS

While in 8 or 16-bit data mode, the host is required to perform two or four, 16 or 8-bit writes to complete a single DWORD transfer. No ordering requirements exist. The host can access either the low or high WORD or BYTE first, as long as the other write(s) is(are) performed to the remaining WORD or BYTEs.

Note:	Writing the same WORD or BYTEs in the same DWORD assemble cycle may cause undefined or undesirable operation. The HBI hardware does not protect against this operation.

A write BYTE / WORD counter keeps track of the number of writes. At the trailing edge of the write cycle, the counter is incremented. Once all writes occur, a 32-bit write is performed to the internal register.

The write BYTE / WORD counter is reset if the power management mode is set to anything other than D0.

9.4.2.2 Read Cycles

A read cycle occurs when **CS** and **RD** are active (or when **ENB** is active with **RD_WR** indicating read). The host address and endianness were already captured during the address latch cycle.

At the beginning of the read cycle, the appropriate register is selected and its data is driven onto the data pins. Depending on the bus width, either a WORD or a BYTE is read. For 8 or 16-bit data modes, the returned BYTE or WORD is determined by the endianness and the lower address inputs.

POLLING FOR INITIALIZATION COMPLETE

Before device initialization, the HBI will not return valid data. To determine when the HBI is functional, the Byte Order Test Register (BYTE_TEST) should be polled. Each poll should consist of an address latch cycle(s) and a data cycle. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) can be polled to determine when the device is fully configured.

READS DURING AND FOLLOWING POWER MANAGEMENT

During any power management mode other than D0, reads from the Host Bus are ignored. If the power management mode changes back to D0 during an active read cycle, the tail end of the read cycle is ignored. Internal registers are not affected and the state of the HBI does not change.

8 AND 16-BIT ACCESS

For certain register accesses, the host is required to perform two or four consecutive 16 or 8-bit reads to complete a single DWORD transfer. No ordering requirements exist. The host can access either the low or high WORD or BYTE first, as long as the other read(s) is(are) performed from the remaining WORD or BYTEs.

Note:	Reading the same WORD or BYTEs from the same DWORD may cause undefined or undesirable operation. The HBI hardware does not protect against this operation. The HBI simply counts that four BYTEs have been read.

A read BYTE / WORD counter keeps track of the number of reads. This counter is separate from the write counter above. At the trailing edge of the read cycle, the counter is incremented. On the last read for the DWORD, an internal read is performed to update any Change on Read CSRs.

The read BYTE / WORD counter is reset if the power management mode is set to anything other than D0.

SPECIAL CSR HANDLING

Live Bits

Any register bit that is updated by a H/W event is held at the beginning of the read cycle to prevent it from changing during the read cycle.

Multiple BYTE / WORD Live Registers in 16 or 8-Bit Modes

Some registers have “live” fields or related fields that span across multiple BYTEs or WORDs. For 16 and 8-bit data reads, it is possible for the value of these fields to change between host read cycles. In order to prevent reading intermediate values, these registers are locked when the first byte or word is read and unlocked when the last byte or word is read.

The registers are unlocked if the power management mode is set to anything other than D0.

Change on Read Registers and FIFOs

FIFOs or “Change on Read” registers, are updated at the end of the read cycle.

For 16 and 8-bit modes, only one internal read cycle is indicated and occurs for the last byte or word.

Change on Read Live Register Bits

As described above, registers with live bits are held starting at the beginning of the read cycle and those that have multiple bits that span across BYTES or WORDS are also locked for 16 and 8-bit accesses. Although a H/W event that occurs during the hold or lock time would still update the live bit(s), the live bit(s) will be affected (cleared, etc.) at the end of the read cycle and the H/W event would be lost.

In order to prevent this, the individual CSRs defer the H/W event update until after the read or multiple reads.

Register Polling During Reset Or Initialization

Some registers support polling during reset or device initialization to determine when the device is accessible. For these registers, only one read may be performed without the need to read the other WORD or BYTEs. The same BYTE or WORD of the register may be re-read repeatedly.

A register that is 16 or 8-bit readable or readable during reset or device initialization, is noted in its register description.

9.4.2.3 Host Endianness

The device supports big and little endian host byte ordering based upon the endianness select that is latched during the address latch cycle. When the endianness select is low, host access is little endian and when high, host access is big endian. In a typical application the endianness select is connected to a high-order address line, making endian selection address-based. This highly flexible interface provides mixed endian access for registers and memory for both PIO and host DMA access.

All internal busses are 32-bit with little endian byte ordering. Logic within the Host Bus Interface re-orders bytes based on the appropriate endianness bit, and the state of the least significant address bits.

Data path operations for the supported endian configurations and data bus sizes are illustrated in FIGURE 9-1: Little Endian Ordering on page 78 and FIGURE 9-2: Big Endian Ordering on page 79.

FIGURE 9-1: LITTLE ENDIAN ORDERING

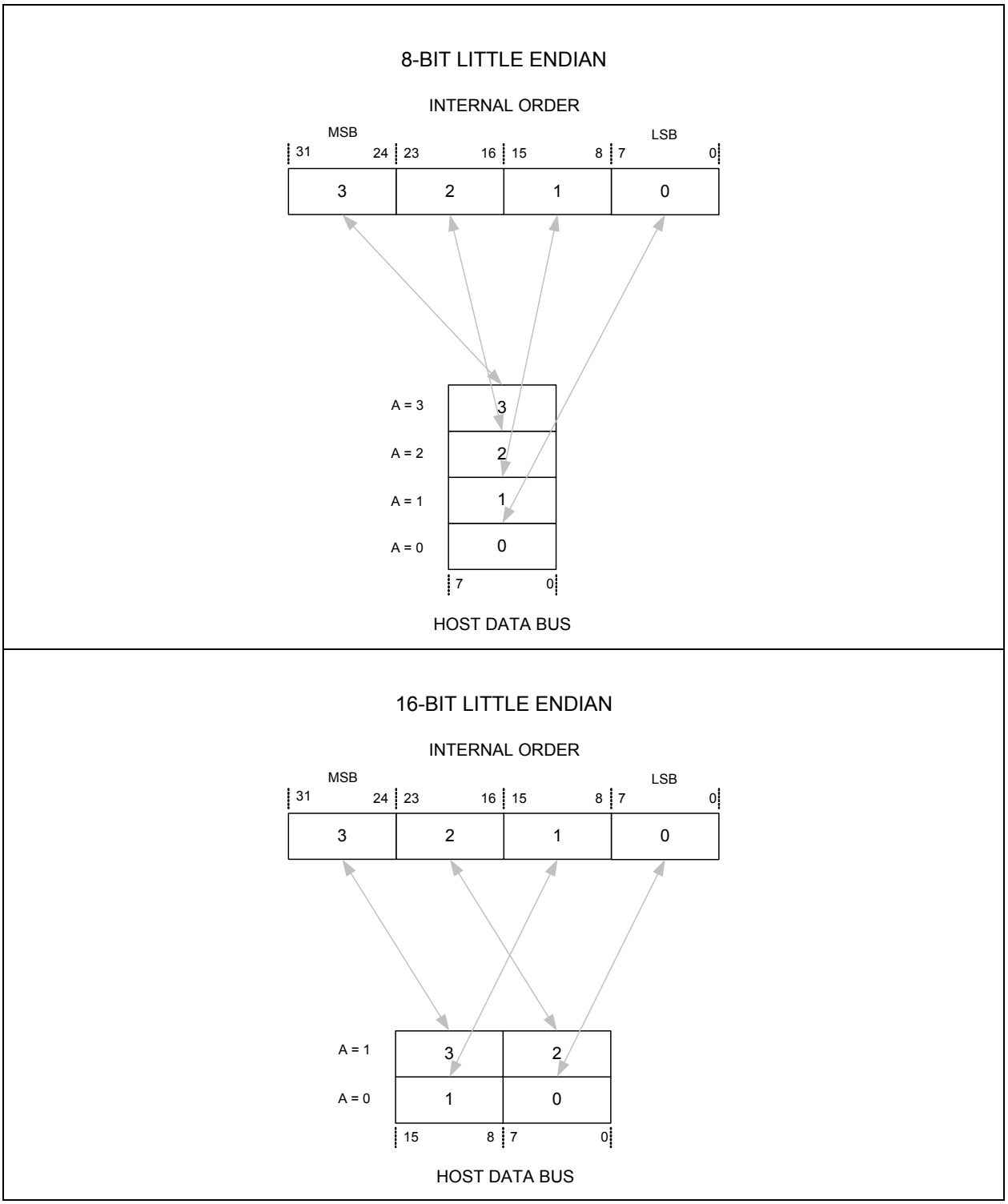
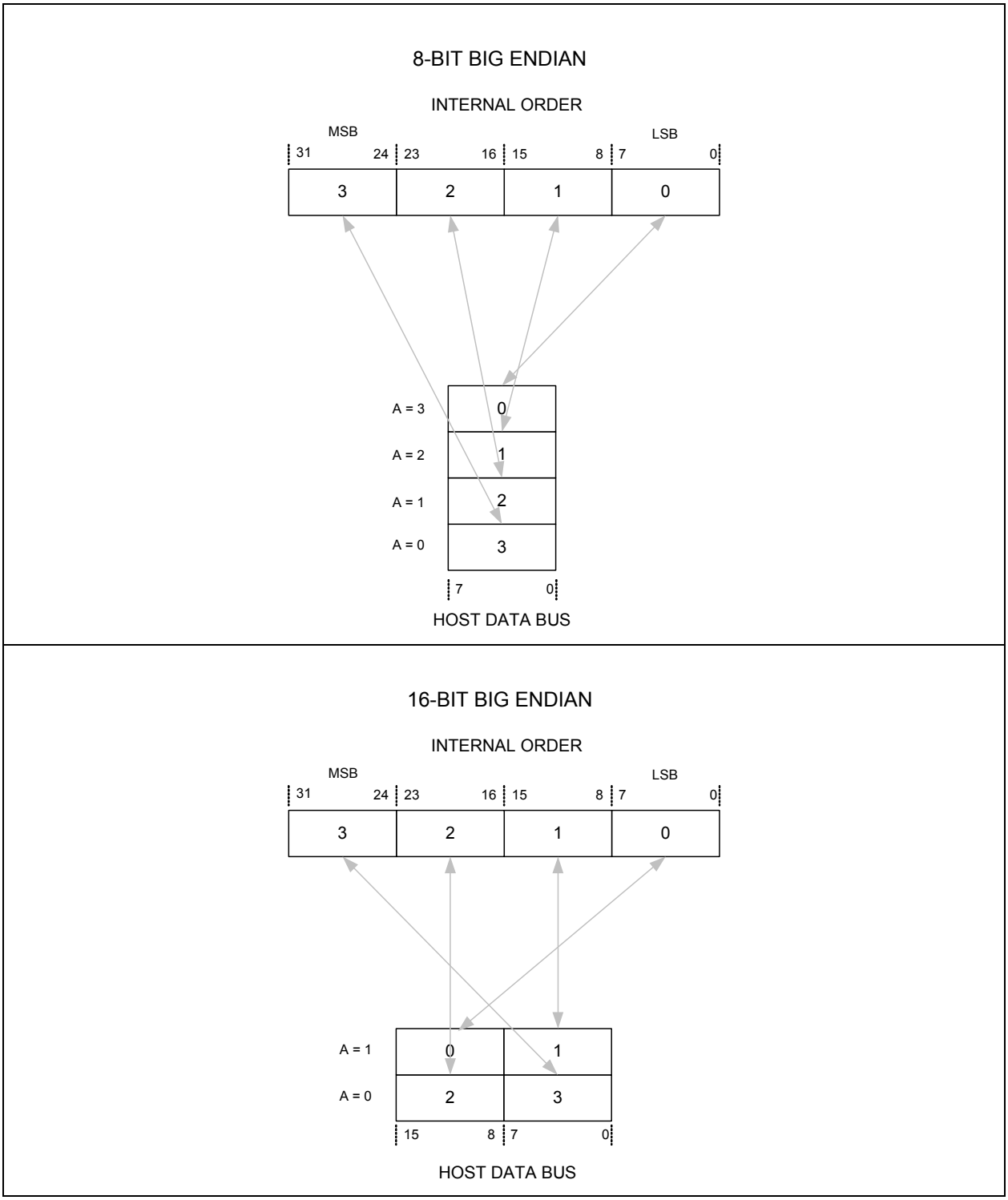


FIGURE 9-2: BIG ENDIAN ORDERING



9.4.3 TX AND RX FIFO ACCESS

9.4.3.1 TX and RX Status FIFO Peek Address Access

Normal read access to the TX or RX Status FIFO causes the FIFO to advance to its next entry.

For access to the TX and RX Status FIFO Peek addresses, the FIFO does not advance to its next entry.

9.4.3.2 FIFO Direct Select Access

A FIFO Direct Select signal is provided allows the host system to address the TX and RX Data FIFOs as if they were a large flat address space. When the FIFO Direct Select signal, which was latched during the address latch cycle, is active all host write operations are to the TX Data FIFO and all host read operations are from the RX Data FIFO. Only the lower latched address signals are decoded in order to select the proper BYTE or WORD. All other address inputs are ignored in this mode. All other operations are the same (DWORD assembly, FIFO popping, etc.).

The endianness of FIFO Direct Select accesses is determined by the endianness select that was latched during the address latch cycle.

Burst access when reading the RX Data FIFO is not supported. However, since the FIFO Direct Select signal is retained until either a reset event occurs or a new address is loaded, multiple read or write requests can occur without requiring multiple address latching operations.

9.4.4 MULTIPLEXED ADDRESSING MODE FUNCTIONAL TIMING DIAGRAMS

The following timing diagrams illustrate example multiplexed addressing mode read and write cycles for various address/data configurations and bus sizes. These diagrams do not cover every supported host bus permutation, but are selected to detail the main configuration differences (bus size, dual/single phase address latching) within the multiplexed addressing mode of operation.

The following should be noted for the timing diagrams in this section:

- The diagrams in this section depict active-high **ALEHI/ALELO**, **CS**, **RD**, and **WR** signals. The polarities of these signals are selectable via the HBI_ale_polarity_strap, HBI_cs_polarity_strap, HBI_rd_rdwr_polarity_strap, and HBI_wr_en_polarity_strap, respectively. Refer to Section 9.3, "Control Line Polarity," on page 74 for additional details.
- The diagrams in this section depict little endian byte ordering. However, dynamic big and little endianess are supported via the endianess signal. Endianess changes only the order of the bytes involved, and not the overall timing requirements. Refer to Section 9.4.1.4, "Endianness Select to Address / Data Pin Mapping," on page 75 for additional information.
- The diagrams in Section 9.4.4.1, "Dual Phase Address Latching" and Section 9.4.4.2, "Single Phase Address Latching" utilize **RD** and **WR** signals. Alternative **RD_WR** and **ENB** signaling is also supported, as shown in Section 9.4.4.3, "RD_WR / ENB Control Mode Examples". The HBI read/write mode is selectable via the HBI_rw_-mode_strap. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI_rd_rdwr_polarity_strap, and HBI_wr_en_polarity_strap.
- Qualification of the **ALELO** and/or **ALEHI** with the **CS** signal is selectable via the HBI_ale_qualification_strap. Refer to Section 9.4.1.1, "Single Phase Address Latching," on page 74 and Section 9.4.1.2, "Dual Phase Address Latching," on page 75 for additional information.
- In dual phase address latching mode, the **ALEHI** and **ALELO** cycles can be in any order. Either or both **ALELO** and **ALEHI** cycles maybe skipped and the device retains the last latched address.
- In single phase address latching mode, the **ALELO** cycle maybe skipped and the device retains the last latched address.

Note:	In 8 and 16-bit modes, the **ALELO** cycle is normally not skipped since sequential BYTEs or WORDs are accessed in order to satisfy a complete DWORD cycle. However, there are registers for which a single BYTE or WORD access is allowed, in which case multiple accesses to these registers may be performed without the need to re-latch the repeated address.

- For 16 and 8-bit modes, consecutive address cycles must be within the same DWORD until the DWORD is completely accessed (with the register exceptions noted above). Although BYTEs and WORDs can be accessed in any order, the diagrams in this section depict accessing the lower address BYTE or WORD first.

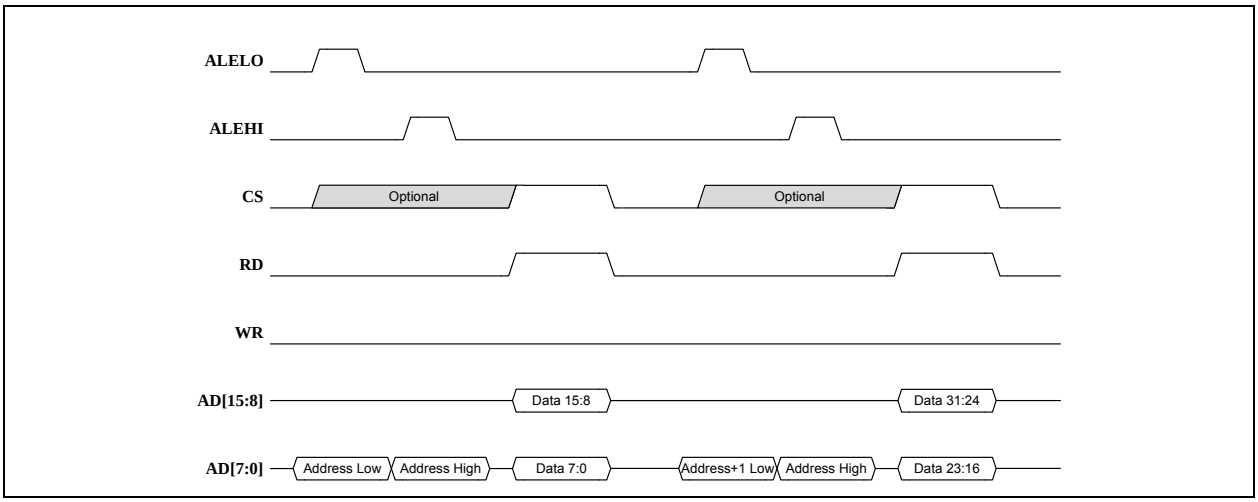
9.4.4.1 Dual Phase Address Latching

The figures in this section detail read and write operations in multiplexed addressing mode with dual phase address latching for 16 and 8-bit modes.

16-BIT READ

The address is latched sequentially from **AD[7:0]**. **AD[15:8]** is not used or driven for the address phase. A read on **AD[15:0]** follows. The cycle is repeated for the other 16-bits of the DWORD.

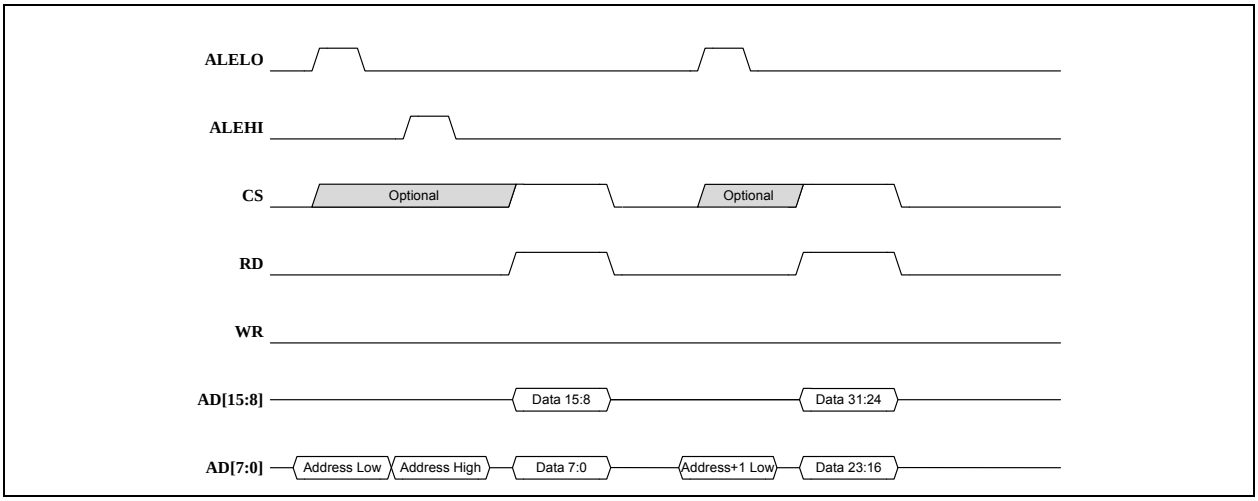
FIGURE 9-3: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT READ



16-BIT READ WITH SUPPRESSED ALEHI

The address is latched sequentially from **AD[7:0]**. **AD[15:8]** is not used or driven for the address phase. A read on **AD[15:0]** follows. The lower address is then updated to access the opposite WORD.

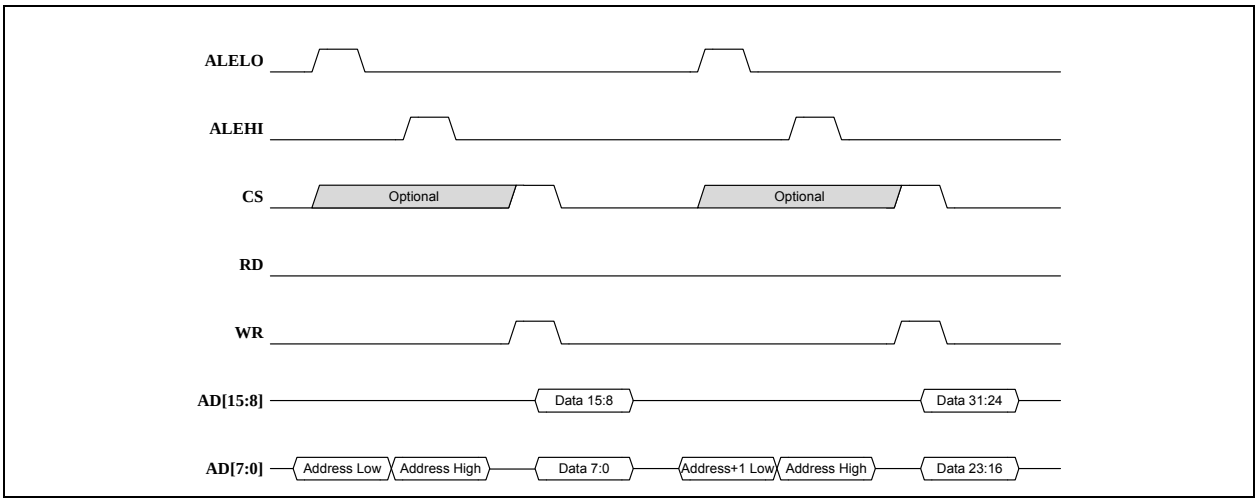
FIGURE 9-4: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT READ WITHOUT ALEHI



16-BIT WRITE

The address is latched sequentially from **AD[7:0]**. **AD[15:8]** is not used or driven for the address phase. A write on **AD[15:0]** follows. The cycle is repeated for the other 16-bits of the DWORD.

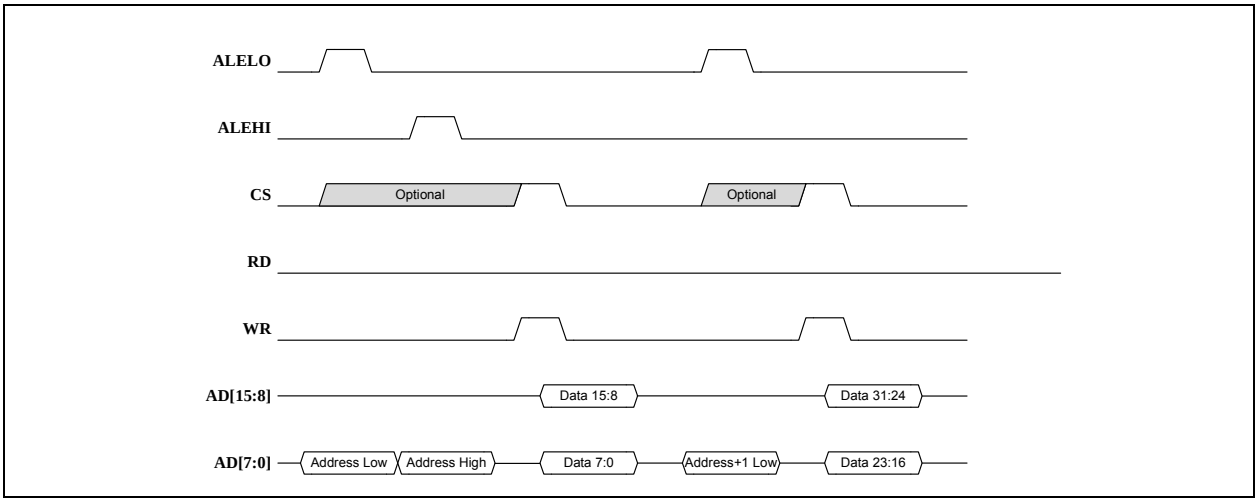
FIGURE 9-5: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT WRITE



16-BIT WRITE WITH SUPPRESSED ALEHI

The address is latched sequentially from **AD[7:0]**. **AD[15:8]** is not used or driven for the address phase. A write on **AD[15:0]** follows. The lower address is then updated to access the opposite WORD.

FIGURE 9-6: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT WRITE WITHOUT ALEHI

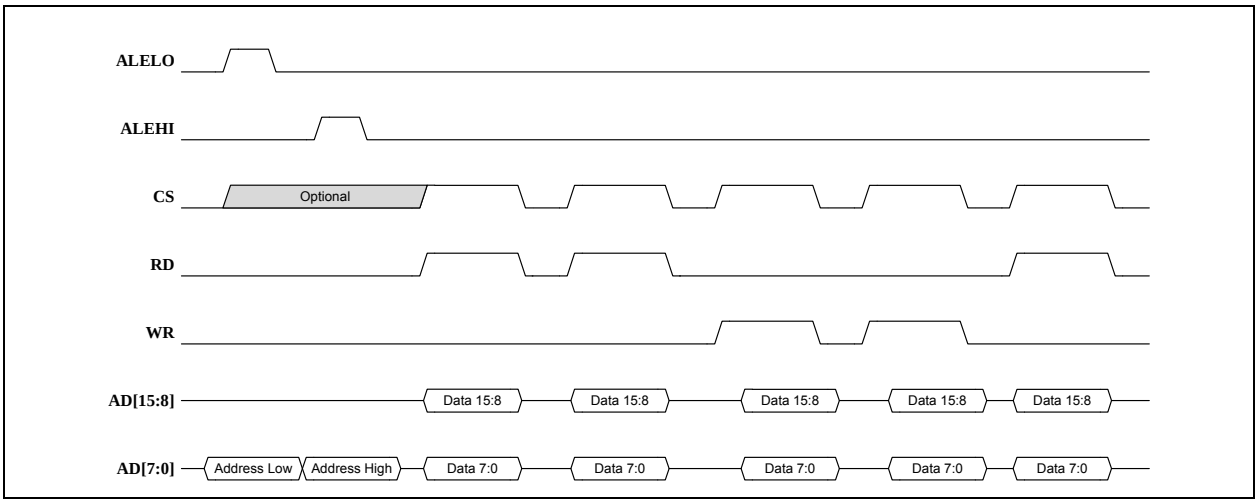


16-BIT READS AND WRITES TO CONSTANT ADDRESS

The address is latched sequentially from **AD[7:0]**. **AD[15:8]** is not used or driven for the address phase. A mix of reads and writes on **AD[15:0]** follows.

Note:	Generally, two 16-bit reads to opposite WORDs of the same DWORD are required, with at least the lower address changing using **ALELO**. 16-bit reads and writes to the same WORD is a special case.

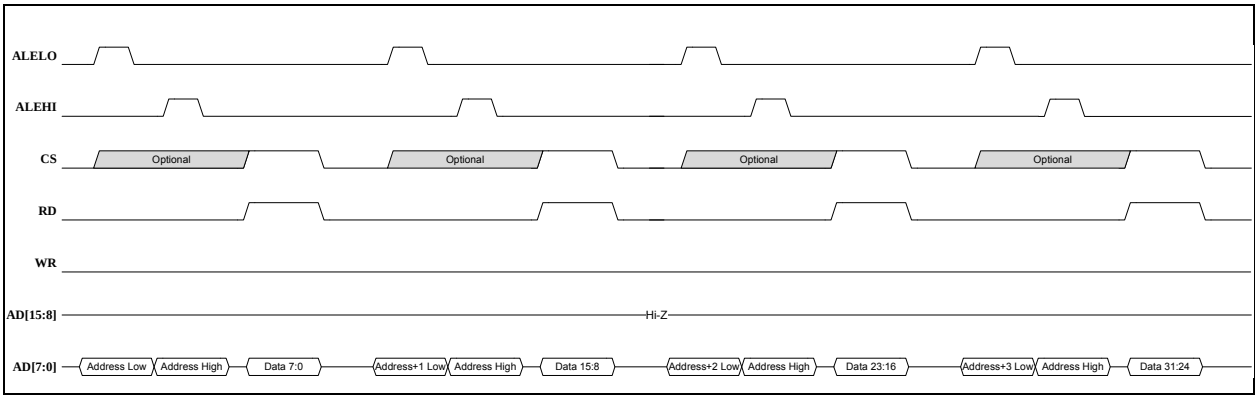
FIGURE 9-7: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 16-BIT READS AND WRITES CONSTANT ADDRESS



8-BIT READ

The address is latched sequentially from **AD[7:0]**. A read on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven. The cycle is repeated for the other BYTEs of the DWORD.

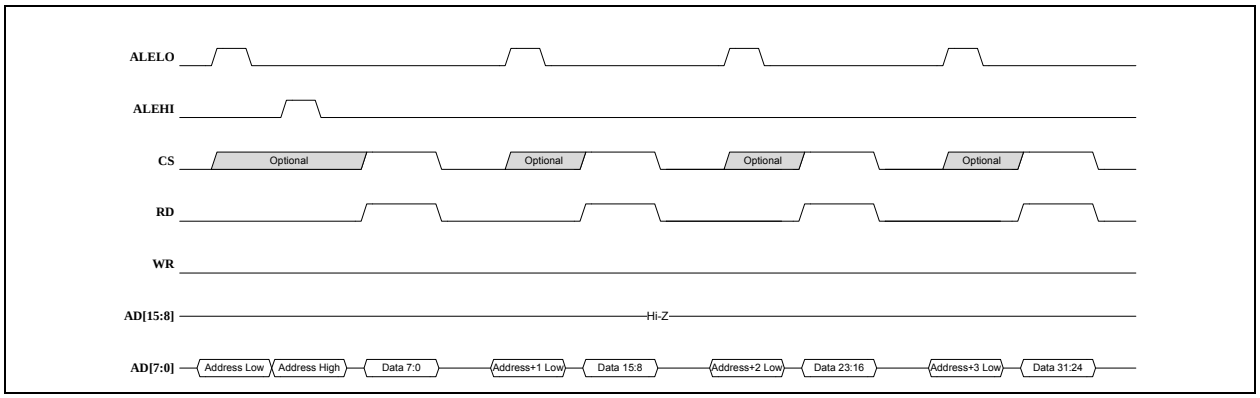
FIGURE 9-8: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT READS



8-BIT READ WITH SUPPRESSED ALEHI

The address is latched sequentially from **AD[7:0]**. A read on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven. The lower address is then updated to access the other BYTEs.

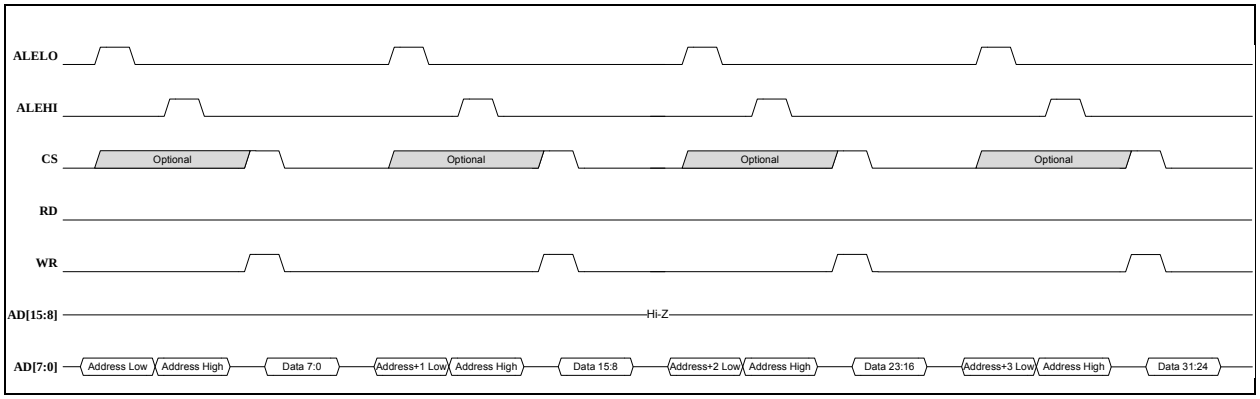
FIGURE 9-9: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT READS WITHOUT ALEHI



8-BIT WRITE

The address is latched sequentially from **AD[7:0]**. A write on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven. The cycle is repeated for the other BYTEs of the DWORD.

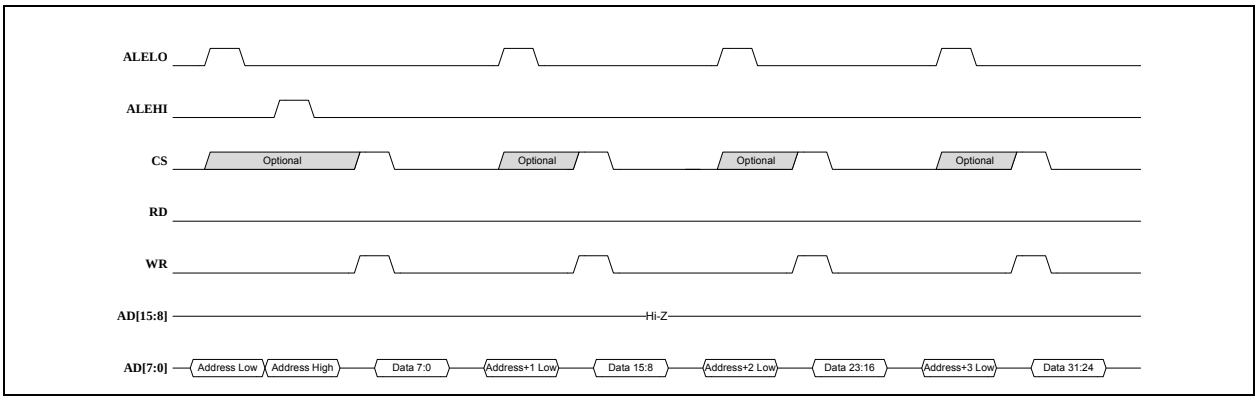
FIGURE 9-10: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT WRITE



8-BIT WRITE WITH SUPPRESSED ALEHI

The address is latched sequentially from **AD[7:0]**. A write on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven. The lower address is then updated to access the other BYTEs.

FIGURE 9-11: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT WRITE WITHOUT ALEHI

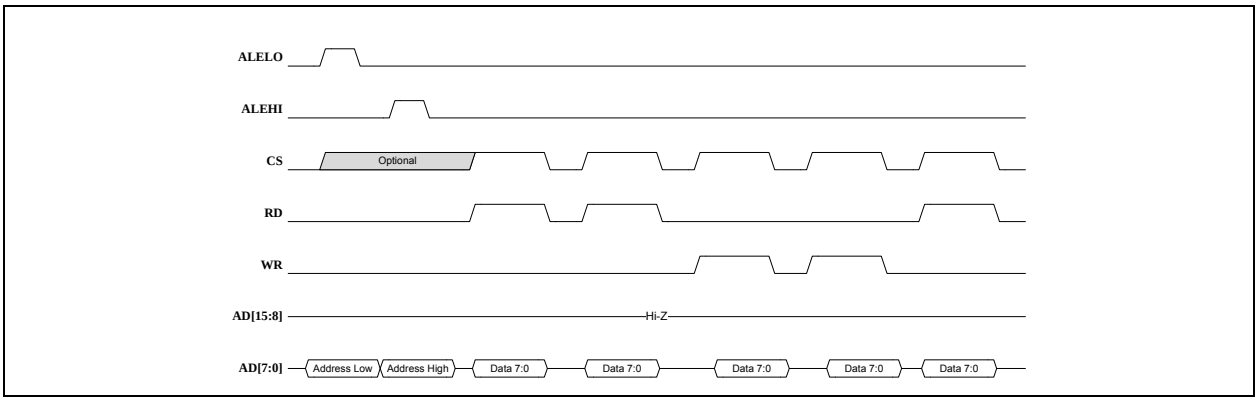


8-BIT READS AND WRITES TO CONSTANT ADDRESS

The address is latched sequentially from **AD[7:0]**. A mix of reads and writes on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven.

Note:	Generally, four 8-bit reads to opposite BYTEs of the same DWORD are required, with at least the lower address changing using **ALELO**. 8-bit reads and writes to the same BYTE is a special case.

FIGURE 9-12: MULTIPLEXED ADDRESSING WITH DUAL PHASE LATCHING - 8-BIT READS AND WRITES CONSTANT ADDRESS



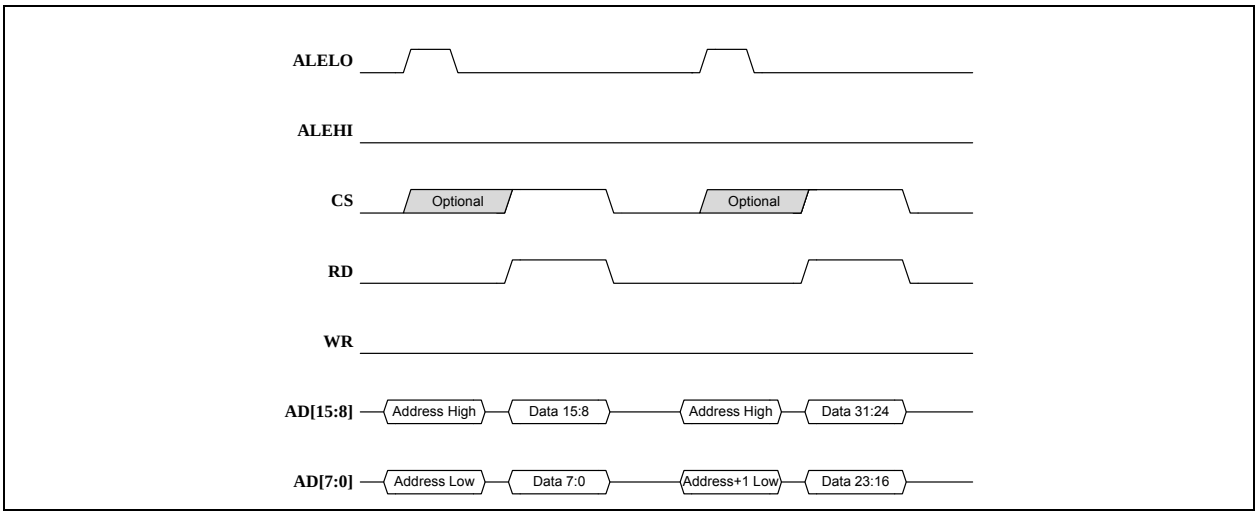
9.4.4.2 Single Phase Address Latching

The figures in this section detail multiplexed addressing mode with single phase addressing for 16 and 8-bit modes of operation.

16-BIT READ

The address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A read on **AD[15:0]** follows. The cycle is repeated for the other 16-bits of the DWORD.

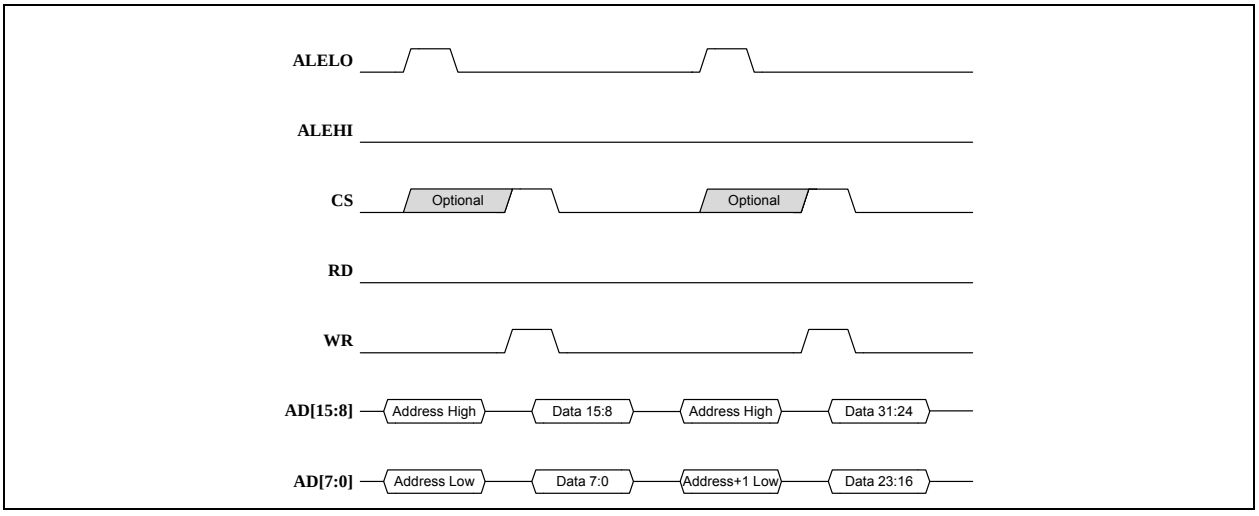
FIGURE 9-13: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 16-BIT READ



16-BIT WRITE

The address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A write on **AD[15:0]** follows. The cycle is repeated for the other 16-bits of the DWORD.

FIGURE 9-14: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 16-BIT WRITE



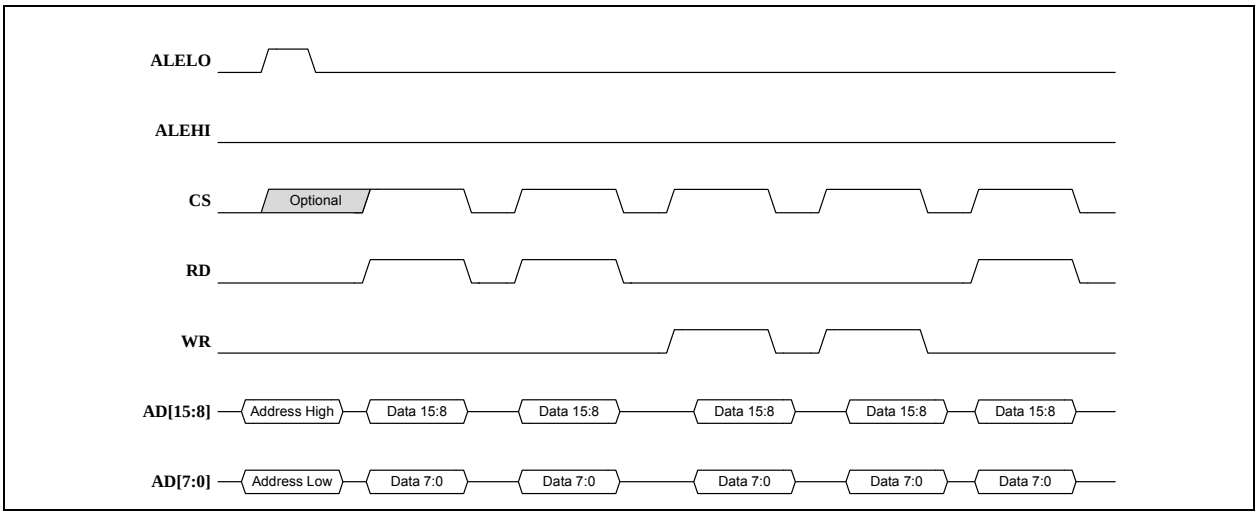
16-BIT READS AND WRITES TO CONSTANT ADDRESS

The address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A mix of reads and writes on **AD[15:0]** follows.

Note:	Generally, two 16-bit reads to opposite WORDs of the same DWORD are required. 16-bit reads and writes to the same WORD is a special case.

.

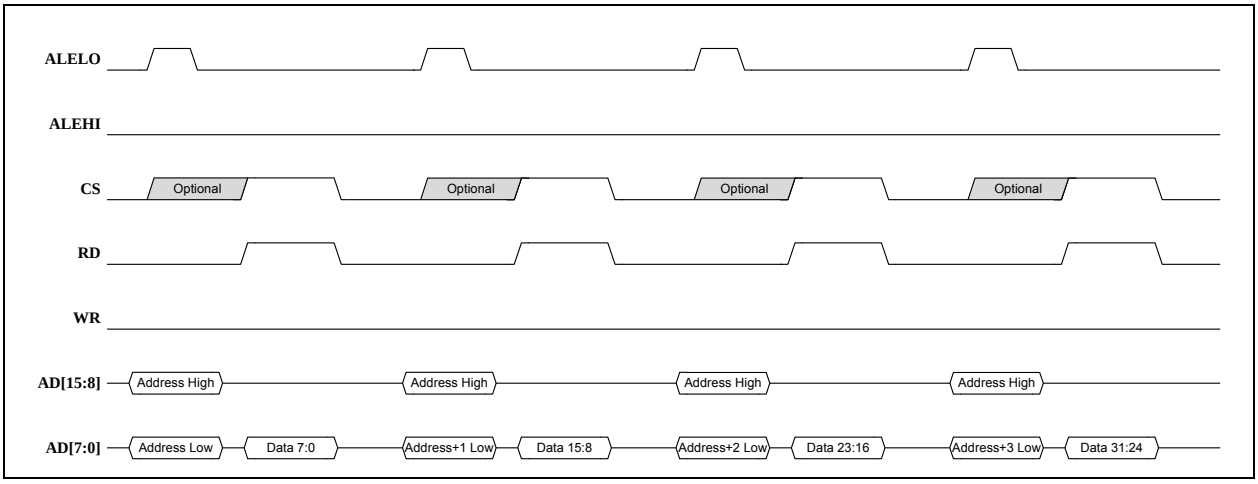
FIGURE 9-15: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 16-BIT READS AND WRITES CONSTANT ADDRESS



8-BIT READ

The address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A read on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven for the data phase as the host could potentially continue to drive the upper address on these signals. The cycle is repeated for the other BYTEs of the DWORD.

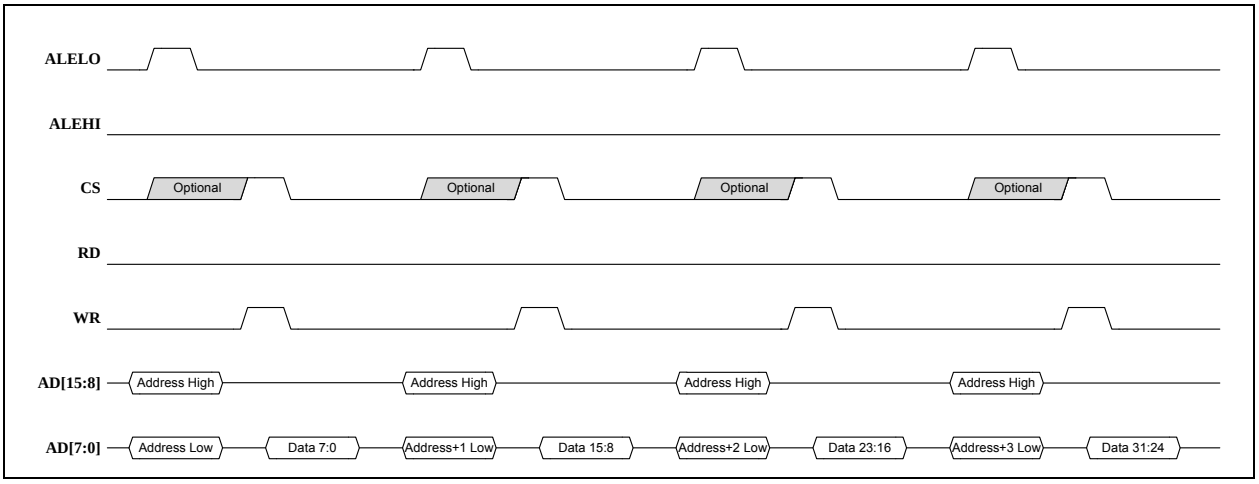
FIGURE 9-16: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 8-BIT READ



8-BIT WRITE

The address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A write on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven for the data phase as the host could potentially continue to drive the upper address on these signals. The cycle is repeated for the other BYTEs of the DWORD.

FIGURE 9-17: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 8-BIT WRITE

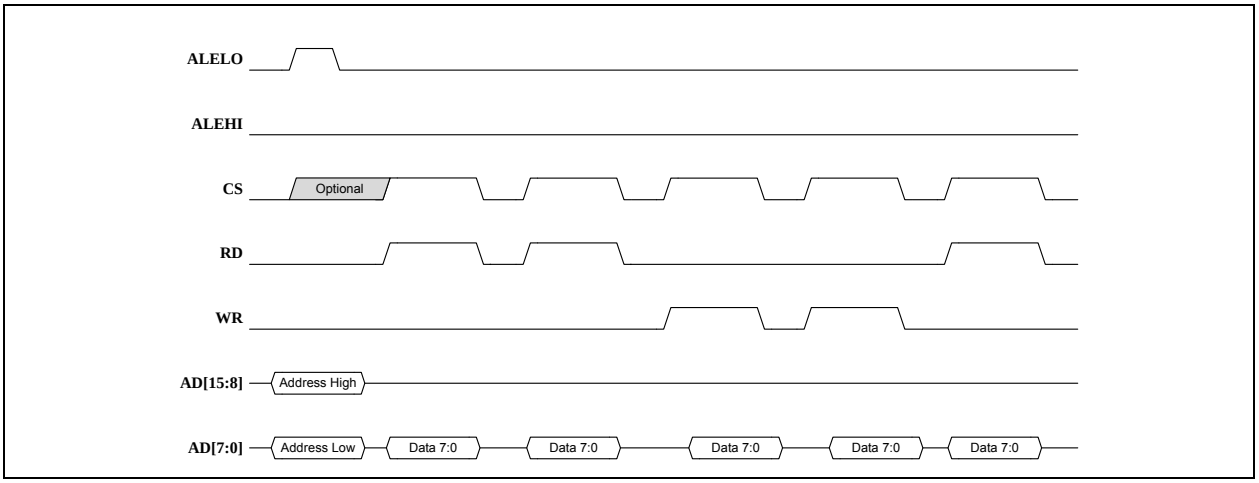


8-BIT READS AND WRITES TO CONSTANT ADDRESS

The address is latched simultaneously from **AD[7:0]** and **AD[15:8]**. A mix of reads and writes on **AD[7:0]** follows. **AD[15:8]** pins are not used or driven for the data phase as the host could potentially continue to drive the upper address on these signals.

Note:	Generally, four 8-bit reads to opposite BYTEs of the same DWORD are required. 8-bit reads and writes to the same BYTE is a special case.

FIGURE 9-18: MULTIPLEXED ADDRESSING WITH SINGLE PHASE LATCHING - 8-BIT READS AND WRITES CONSTANT ADDRESS



9.4.4.3 RD_WR / ENB Control Mode Examples

The figures in this section detail read and write operations utilizing the alternative **RD_WR** and **ENB** signaling. The HBI read/write mode is selectable via the HBI_rw_mode_strap.

Note: The examples in this section detail 16-bit mode with dual phase latching. However, the **RD_WR** and **ENB** signaling can be used identically in all other multiplexed addressing modes of operation.

The examples in this section show the **ENB** signal active-high and the **RD_WR** signal low for read and high for write. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI_rd_rdwr_polarity_strap, and HBI_wr_en_polarity_strap.

16-BIT

FIGURE 9-19: MULTIPLEXED ADDRESSING RD_WR / ENB CONTROL MODE EXAMPLE - 16-BIT READ

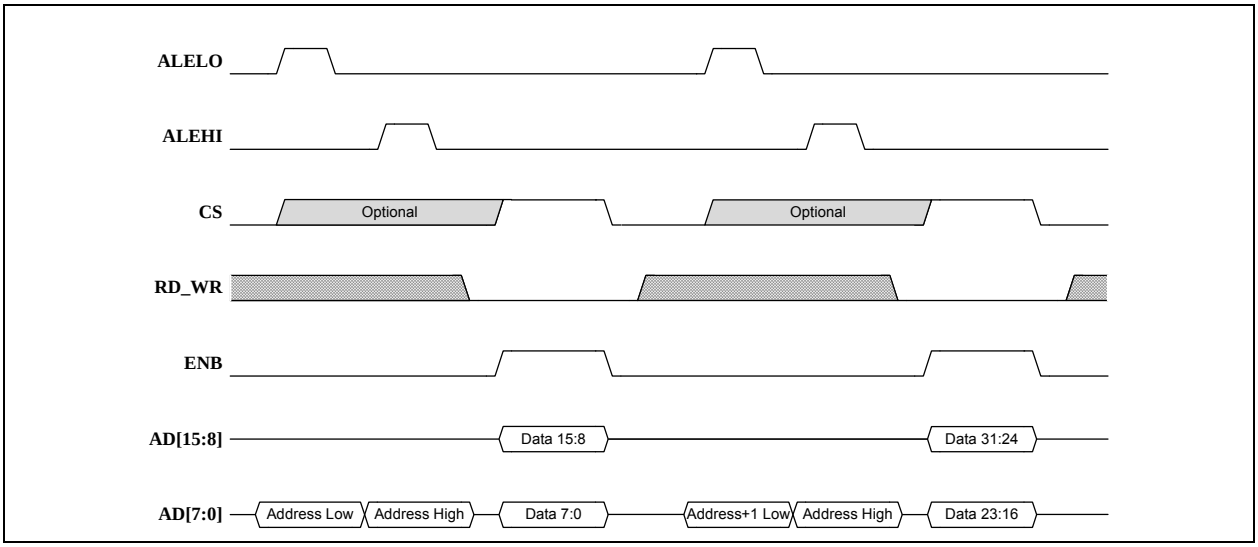
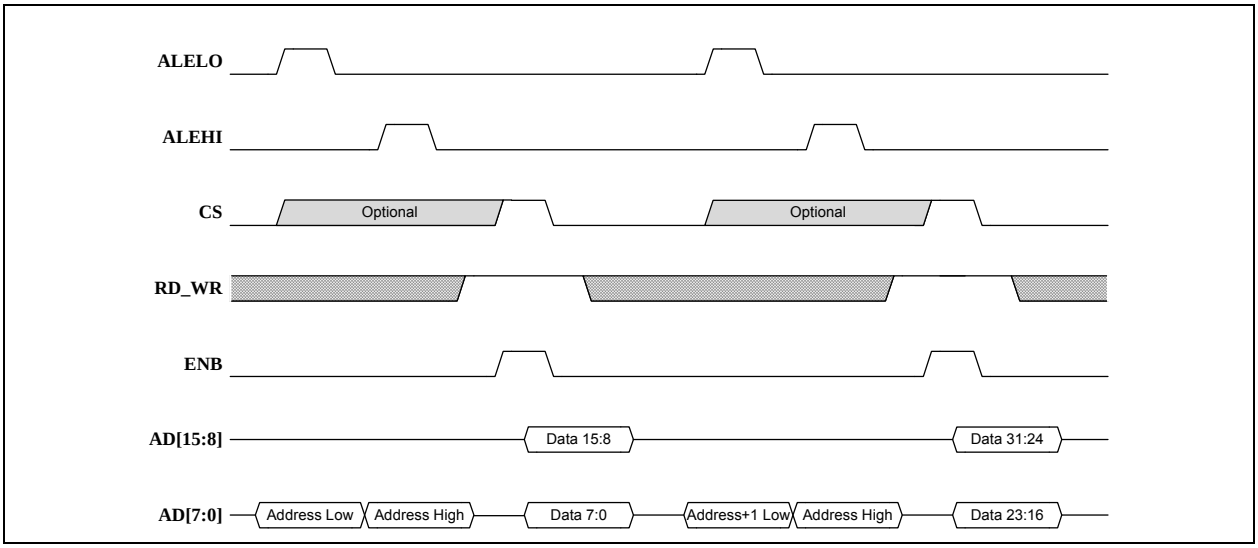


FIGURE 9-20: MULTIPLEXED ADDRESSING RD_WR / ENB CONTROL MODE EXAMPLE - 16-BIT WRITE



9.4.5 MULTIPLEXED ADDRESSING MODE TIMING REQUIREMENTS

The following figures and tables specify the timing requirements during Multiplexed Address / Data mode. Since timing requirements are similar across the multitude of operations (e.g. dual vs. single phase, 8 vs. 16-bit), many timing requirements are illustrated onto the same figures and do not necessarily represent any particular functional operation.

The following should be noted for the timing specifications in this section:

- The diagrams in this section depict active-high **ALEHI/ALELO**, **CS**, **RD**, **WR**, **RD_WR** and **ENB** signals. The polarities of these signals are selectable via the HBI_ale_polarity_strap, HBI_cs_polarity_strap, HBI_rd_rdwr_polarity_strap, and HBI_wr_en_polarity_strap, respectively. Refer to Section 9.3, "Control Line Polarity," on page 74 for additional details.
- Qualification of the **ALELO** and/or **ALEHI** with the **CS** signal is selectable via the HBI_ale_qualification_strap. This is shown as a dashed line. Timing requirements between **ALELO** / **ALEHI** and **CS** only apply when this mode is active.
- In dual phase address latching mode, the **ALEHI** and **ALELO** cycles can be in any order. **ALEHI** first is depicted in solid line. **ALELO** first is depicted in dashed line.
- A read cycle maybe followed by followed by an address cycle, a write cycle or another read cycle. A write cycle maybe followed by followed by a read cycle or another write cycle. These are shown in dashed line.

9.4.5.1 Read Timing Requirements

If **RD** and **WR** signaling is used, a host read cycle begins when **RD** is asserted with **CS** active. The cycle ends when **RD** is de-asserted. **CS** maybe asserted and de-asserted along with **RD** but not during **RD** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host read cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a read. The cycle ends when **ENB** is de-asserted. **CS** maybe asserted and de-asserted along with **ENB** but not during **ENB** active.

Please refer to Section 9.4.4, "Multiplexed Addressing Mode Functional Timing Diagrams," on page 80 for functional descriptions.

FIGURE 9-21: MULTIPLEXED ADDRESSING READ CYCLE TIMING

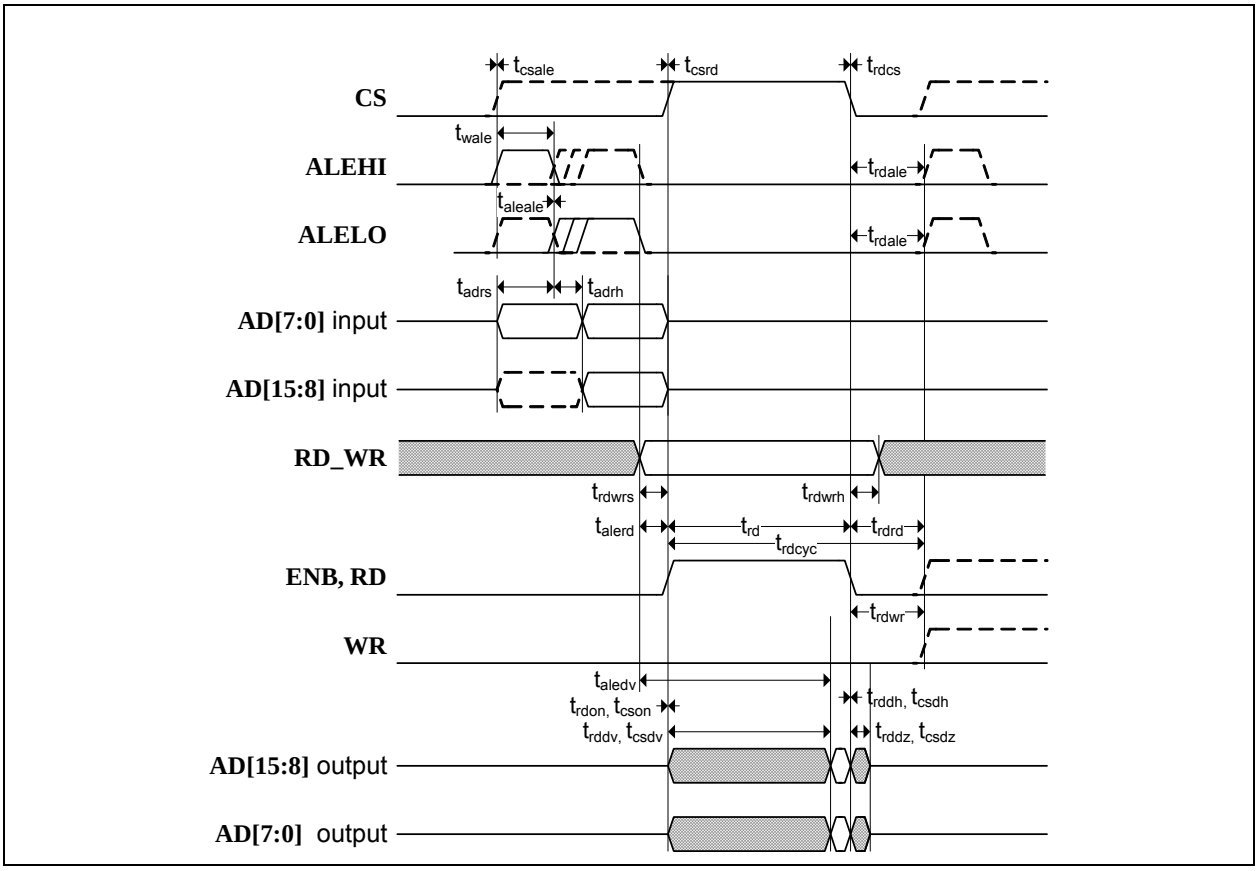


TABLE 9-1: MULTIPLEXED ADDRESSING READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	max	units
t_{csale}	**CS** Setup to **ALELO**, **ALEHI** Active Note 3, Note 2	0			ns
t_{csrd}	**CS** Setup to **RD** or **ENB** Active	0			ns
t_{rdcs}	**CS** Hold from **RD** or **ENB** Inactive	0			ns
t_{wale}	**ALELO**, **ALEHI** Pulse Width	10			ns
t_{adrs}	Address Setup to **ALELO**, **ALEHI** Inactive	10			ns
t_{adrh}	Address Hold from **ALELO**, **ALEHI** Inactive	5			ns
t_{aleale}	**ALELO** Inactive to **ALEHI** Active **ALEHI** Inactive to **ALELO** Active Note 1, Note 2	0			ns
t_{alerd}	**ALELO**, **ALEHI** Inactive to **RD** or **ENB** Active Note 2	5			ns
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 4	5			ns
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 4	5			ns
t_{rdon}	**RD** or **ENB** to Data Buffer Turn On	0			ns
t_{rddv}	**RD** or **ENB** Active to Data Valid			30	ns
t_{rddh}	Data Output Hold Time from **RD** or **ENB** Inactive	0			ns
t_{rddz}	Data Buffer Turn Off Time from **RD** or **ENB** Inactive			9	ns
t_{cson}	**CS** to Data Buffer Turn On	0			ns
t_{csdv}	**CS** Active to Data Valid			30	ns
t_{csdh}	Data Output Hold Time from **CS** Inactive	0			ns
t_{csdz}	Data Buffer Turn Off Time from **CS** Inactive			9	ns
t_{aledv}	**ALELO**, **ALEHI** Inactive to Data Valid Note 2			35	ns
t_{rd}	**RD** or **ENB** Active Time	32			ns
t_{rdcyc}	**RD** or **ENB** Cycle Time	45			ns
t_{rdale}	**RD** or **ENB** De-assertion Time before Address Phase	13			ns
t_{rdrd}	**RD** or **ENB** De-assertion Time before Next **RD** or **ENB** Note 5	13			ns
t_{rdwr}	**RD** De-assertion Time before Next **WR** Note 5, Note 6	13			ns

Note 1: Dual Phase Addressing

Note 2: Depends on **ALEHI** / **ALELO** order.

Note 3: **ALELO** and/or **ALEHI** qualified with the **CS**.

Note 4: **RD_WR** and **ENB** signaling.

Note 5: No interposed address phase.

Note 6: **RD** and **WR** signaling.

Note: Timing values are with respect to an equivalent test load of 25 pF.

9.4.5.2 Write Timing Requirements

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is de-asserted. **CS** maybe asserted and de-asserted along with **WR** but not during **WR** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is de-asserted. **CS** maybe asserted and de-asserted along with **ENB** but not during **ENB** active.

Please refer to Section 9.4.4, "Multiplexed Addressing Mode Functional Timing Diagrams," on page 80 for functional descriptions.

FIGURE 9-22: MULTIPLEXED ADDRESSING WRITE CYCLE TIMING

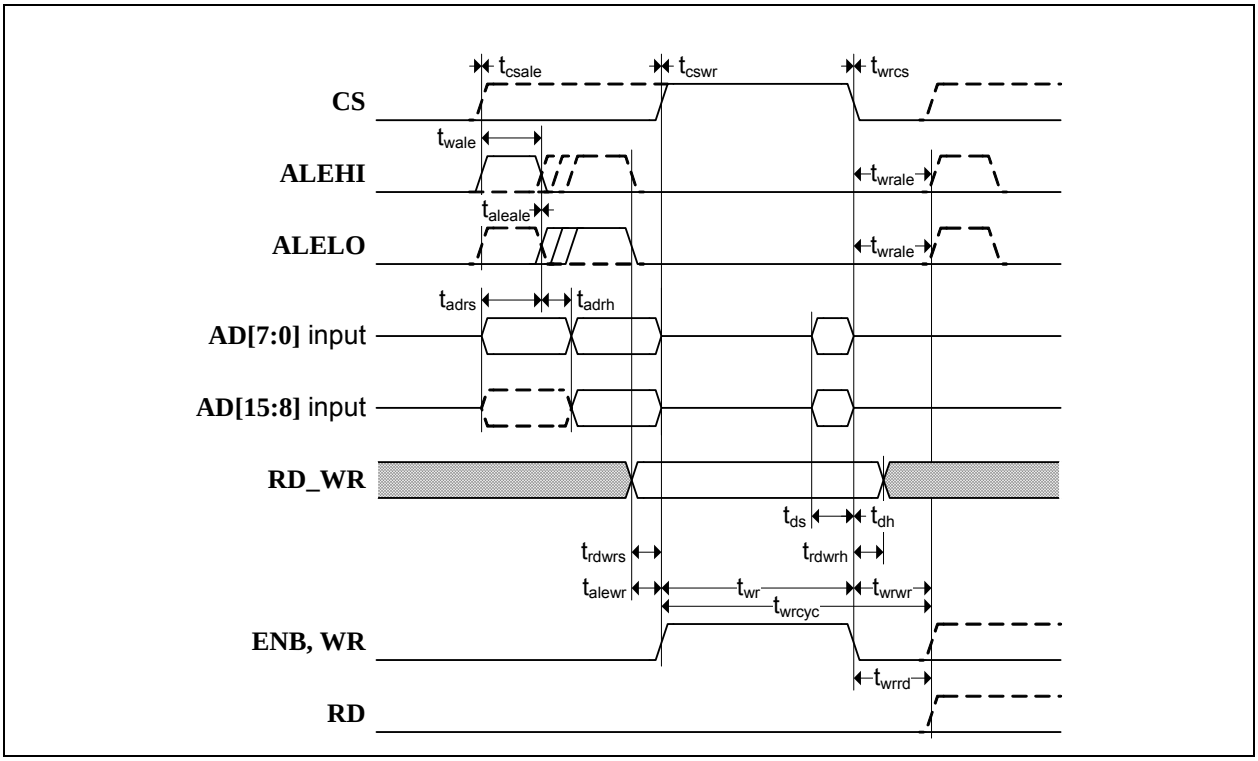


TABLE 9-2: MULTIPLEXED ADDRESSING WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csale}	**CS** Setup to **ALELO**, **ALEHI** Active Note 9, Note 8	0			ns
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			ns
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			ns
t_{wale}	**ALELO**, **ALEHI** Pulse Width	10			ns
t_{adrs}	Address Setup to **ALELO**, **ALEHI** Inactive	10			ns
t_{adrh}	Address Hold from **ALELO**, **ALEHI** Inactive	5			ns
t_{aleale}	**ALELO** Inactive to **ALEHI** Active **ALEHI** Inactive to **ALELO** Active Note 7, Note 8	0			ns
t_{alewr}	**ALELO**, **ALEHI** Inactive to **WR** or **ENB** Active Note 8	5			ns
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 10	5			ns
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 10	5			ns
t_{ds}	Data Setup to **WR** or **ENB** Inactive	7			ns
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			ns
t_{wr}	**WR** or **ENB** Active Time	32			ns
t_{wrcyc}	**WR** or **ENB** Cycle Time	45			ns
t_{wrale}	**WR** or **ENB** De-assertion Time before Address Phase	13			ns
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB** Note 11	13			ns
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 11, Note 12	13			ns

Note 7: Dual Phase Addressing

Note 8: Depends on **ALEHI** / **ALELO** order.

Note 9: **ALELO** and/or **ALEHI** qualified with the **CS**.

Note 10: **RD_WR** and **ENB** signaling.

Note 11: No interposed address phase.

Note 12: **RD** and **WR** signaling.

9.5 Indexed Address Mode

In Indexed Address mode, access to the internal registers and memory of the device are indirectly mapped using Index and Data registers. The desired internal address is written into the device at a particular offset. The value written is then used as the internal address when the associate Data register address is accessed. Three Index / Data register sets are provided allowing for multi-threaded operation without the concern of one thread corrupting the Index set by another thread. Endianness can be configured per Index / Data pair. Another Data register is provided for access to the FIFOs.

The host address register map is given below. In 8-bit data mode, the host address input (ADDR[4:0]) is a BYTE address. In 16-bit data mode, ADDR0 is not provided and the host address input (ADDR[4:1]) is a WORD address.

As discussed below in Section 9.5.5.2, "Index Register Bypass FIFO Access", the TX and RX Data FIFOs are accessed when reading or writing at address 18h-1Bh.

As discussed below in Section 9.5.5.3, "FIFO Direct Select Access", when the FIFOSEL input is active, all access is to or from the TX and RX Data FIFOs.

TABLE 9-3: HOST BUS INTERFACE INDEXED ADDRESS MODE REGISTER MAP

FIFOSEL	BYTE ADDRESS	SYMBOL	REGISTER NAME
0	00h-03h	HBI_IDX_0	Host Bus Interface Index Register 0
0	04h-07h	HBI_DATA_0	Host Bus Interface Data Register 0
0	08h-0Bh	HBI_IDX_1	Host Bus Interface Index Register 1
0	0Ch-0Fh	HBI_DATA_1	Host Bus Interface Data Register 1
0	10h-13h	HBI_IDX_2	Host Bus Interface Index Register 2
0	14h-17h	HBI_DATA_2	Host Bus Interface Data Register 2
0	18h-1Bh	TX_DATA_FIFO RX_DATA_FIFO	TX Data FIFO RX Data FIFO
0	1Ch-1Fh	HBI_CFG	Host Bus Interface Configuration Register
1	na	TX_DATA_FIFO RX_DATA_FIFO	TX Data FIFO RX Data FIFO

9.5.1 HOST BUS INTERFACE INDEX REGISTER

The Index registers are writable as WORDs or as BYTEs, depending upon the data mode. There is no concern about DWORD assembly rules when writing these registers. The Index registers are formatted as follows:

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:0	**Internal Address** The address used when the corresponding Data register is accessed. **Note:** The internal address provided by each Index register is always considered to be a BYTE address.	R/W	1234h Note 13

Note 13: The default may be used to help determine the endianness of the register.

9.5.2 HOST BUS INTERFACE CONFIGURATION REGISTER

The HBI Configuration register is used to specify the endianness of the interface. Endianess for each Index / Data pair and for FIFO accesses can be individually specified.

The endianness of this register is irrelevant since each byte is shadowed into 4 positions.

The HBI Configuration register is writable as WORDs or as BYTEs, depending upon the data mode. There is no concern about DWORD assembly rules when writing this register. The Configuration register is formatted as follows:

Bits	Description	Type	Default
31:28	**RESERVED**	RO	-
27	**FIFO Endianness Shadow 3** This bit is a shadow of bit 3.	R/W	0b
26	**Host Bus Interface Index / Data Register 2 Endianness Shadow 3** This bit is a shadow of bit 2.	R/W	0b
25	**Host Bus Interface Index / Data Register 1 Endianness Shadow 3** This bit is a shadow of bit 1.	R/W	0b
24	**Host Bus Interface Index / Data Register 0 Endianness Shadow 3** This bit is a shadow of bit 0.	R/W	0b
23:20	**RESERVED**	RO	-
19	**FIFO Endianness Shadow 2** This bit is a shadow of bit 3.	R/W	0b
18	**Host Bus Interface Index / Data Register 2 Endianness Shadow 2** This bit is a shadow of bit 2.	R/W	0b
17	**Host Bus Interface Index / Data Register 1 Endianness Shadow 2** This bit is a shadow of bit 1.	R/W	0b
16	**Host Bus Interface Index / Data Register 0 Endianness Shadow 2** This bit is a shadow of bit 0.	R/W	0b
15:12	**RESERVED**	RO	-
11	**FIFO Endianness Shadow 1** This bit is a shadow of bit 3.	R/W	0b
10	**Host Bus Interface Index / Data Register 2 Endianness Shadow 1** This bit is a shadow of bit 2.	R/W	0b

Bits	Description	Type	Default
9	**Host Bus Interface Index / Data Register 1 Endianness Shadow 1** This bit is a shadow of bit 1.	R/W	0b
8	**Host Bus Interface Index / Data Register 0 Endianness Shadow 1** This bit is a shadow of bit 0.	R/W	0b
7:4	**RESERVED**	RO	-
3	**FIFO Endianness** This bit specifies the endianness of FIFO accesses when they are accessed by means other than the Index / Data Register method. **0 = Little Endian** **1 = Big Endian** **Note:** In order to avoid any ambiguity with the endianness of this register, bits 3, 11, 19 and 27 are shadowed. If any of these bits are set during a write, all of the bits will be set.	R/W	0b
2	**Host Bus Interface Index / Data Register 2 Endianness** This bit specifies the endianness of the Index and Data register set 2. **0 = Little Endian** **1 = Big Endian** **Note:** In order to avoid any ambiguity with the endianness of this register, bits 2, 10, 18 and 26 are shadowed. If any of these bits are set during a write, all of the bits will be set.	R/W	0b
1	**Host Bus Interface Index / Data Register 1 Endianness** This bit specifies the endianness of the Index and Data register set 1. **0 = Little Endian** **1 = Big Endian** **Note:** In order to avoid any ambiguity with the endianness of this register, bits 1, 9, 17 and 25 are shadowed. If any of these bits are set during a write, all of the bits will be set.	R/W	0b
0	**Host Bus Interface Index / Data Register 0 Endianness** This bit specifies the endianness of the Index and Data register set 0. **0 = Little Endian** **1 = Big Endian** **Note:** In order to avoid any ambiguity with the endianness of this register, bits 0, 8, 16 and 24 are shadowed. If any of these bits are set during a write, all of the bits will be set.	R/W	0b

9.5.3 INDEX AND CONFIGURATION REGISTER DATA ACCESS

The host data bus can be 16 or 8-bits wide. The HBI Index registers and the HBI Configuration register are 32-bits wide and are writable as WORDs or as BYTEs, depending upon the data mode. They do not have nor do they require WORDs or BYTEs to DWORD conversion.

9.5.3.1 Write Cycles

A write cycle occurs when CS and WR are active (or when ENB is active with RD_WR indicating write).

On the trailing edge of the write cycle (either WR or CS or ENB going inactive), the host data is captured into the Configuration register or one for the Index registers.

Depending on the bus width, either a WORD or a BYTE is written. The affected WORD or BYTE is determined by the endianness of the register (specified in the Host Bus Interface Configuration Register) and the lower address inputs. Individual BYTE (in 16-bit data mode) access is not supported.

WRITES FOLLOWING INITIALIZATION

Following device initialization, writes from the Host Bus are ignored until after a read cycle is performed.

WRITES DURING AND FOLLOWING POWER MANAGEMENT

During and following any power management mode other than D0, writes from the Host Bus are ignored until after a read cycle is performed.

9.5.3.2 Read Cycles

A read cycle occurs when CS and RD are active (or when ENB is active with RD_WR indicating read). The host address is used directly from the Host Bus.

At the beginning of the read cycle, the appropriate register is selected and its data is driven onto the data pins. Depending on the bus width, either a WORD or a BYTE is read. For 8 or 16-bit data modes, the returned BYTE or WORD is determined by the endianness of the register (specified in the Host Bus Interface Configuration Register) and the lower host address inputs.

9.5.4 INTERNAL REGISTER DATA ACCESS

The host data bus can be 16 or 8-bits wide while all internal registers are 32 bits wide. The Host Bus Interface performs the conversion from WORDs or BYTEs to DWORD, while in 8 or 16-bit data mode. Two or four accesses within the same DWORD are required in order to perform a write or read.

Each Data register, along with the FIFO direct address access, has a separate WORD or BYTE to DWORD conversion. Accesses may be mixed among these (and the HBI Index and Configuration registers) without concern of data corruption.

9.5.4.1 Write Cycles

A write cycle occurs when CS and WR are active (or when ENB is active with RD_WR indicating write). The host address from the Host Bus selects the contents of one of the Index registers. The result of this operation is captured on the leading edge of the write cycle.

The host address inputs and the FIFO Direct Select signal from the Host Bus are also captured on the leading edge of the write cycle. These are used to increment the appropriate write BYTE / WORD counter (for 8 or 16-bit data mode described below) as well as to select the correct DWORD assembly register.

On the trailing edge of the write cycle (either WR or CS or ENB going inactive), the host data is captured into one of the Data registers. Depending on the bus width, either a WORD or a BYTE is captured. For 8 or 16-bit data modes, this functions as the DWORD assembly with the affected WORD or BYTE determined by the lower host address inputs. BYTE swapping is also done at this point based on the endianness of the register (specified in the Host Bus Interface Configuration Register).

Note:	There are separate write BYTE / WORD counters and DWORD assembly registers for each of the three Data Registers as well as for FIFO access.

WRITES FOLLOWING INITIALIZATION

Following device initialization, writes from the Host Bus are ignored until after a read cycle is performed.

WRITES DURING AND FOLLOWING POWER MANAGEMENT

During and following any power management mode other than D0, writes from the Host Bus are ignored until after a read cycle is performed.

8 AND 16-BIT ACCESS

While in 8 or 16-bit data mode, the host is required to perform two or four, 16 or 8-bit writes to complete a single DWORD transfer. No ordering requirements exist. The host can access either the low or high WORD or BYTE first, as long as the other write(s) is(are) performed to the remaining WORD or BYTEs.

Note:	Writing the same WORD or BYTEs into the same DWORD may cause undefined or undesirable operation. The HBI hardware does not protect against this operation. Accessing the same internal register using two Index / Data register pairs may cause undefined or undesirable operation. The HBI hardware does not protect against this operation. Mixing reads and writes into the same Data register may cause undefined or undesirable operation. The HBI hardware does not protect against this operation.

A write BYTE / WORD counter keeps track of the number of writes. Each Data Register has its own BYTE / WORD counter. At the trailing edge of the write cycle, the appropriate counter (based on the captured host address from above) is incremented. Once all writes occur, a 32-bit write is performed to the internal register selected by the captured address from above. The data that is written is selected from one of the three DWORD assembly registers based on the captured host address from above.

All of the write BYTE / WORD counters are reset if the power management mode is set to anything other than D0.

9.5.4.2 Read Cycles

A read cycle occurs when CS and RD are active (or when ENB is active with RD_WR indicating read). The host address from the Host Bus selects the contents of one of the Index registers. The result of this operation is used to select the internal register to be read and also is captured on the leading edge of the read cycle.

The host address inputs and the FIFO Direct Select signal from the Host Bus are also captured on the leading edge of the read cycle. These are used to increment the appropriate read BYTE / WORD counter (for 8 or 16-bit data mode described below).

At the beginning of the read cycle, the appropriate register is selected and its data is driven onto the data pins. Depending on the bus width, either a WORD or a BYTE is read. For 8 or 16-bit data modes, the returned BYTE or WORD is determined by the endianness of the Data register (specified in the Host Bus Interface Configuration Register) and the lower host address inputs.

Note:	There are separate read BYTE / WORD counters for each of the three Data Registers as well as for FIFO access.

POLLING FOR INITIALIZATION COMPLETE

Before device initialization, the HBI will not return valid data. To determine when the HBI is functional, first the Host Bus Interface Index Register 0 should be polled, then the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) can be polled to determine when the device is fully configured.

READS DURING AND FOLLOWING POWER MANAGEMENT

During any power management mode other than D0, reads from the Host Bus are ignored. If the power management mode changes back to D0 during an active read cycle, the tail end of the read cycle is ignored. Internal registers are not affected and the state of the HBI does not change.

8 AND 16-BIT ACCESS

For certain register accesses, the host is required to perform two or four consecutive 16 or 8-bit reads to complete a single DWORD transfer. No ordering requirements exist. The host can access either the low or high WORD or BYTE first, as long as the other read(s) is(are) performed from the remaining WORD or BYTEs.

Note:	Reading the same WORD or BYTEs from the same DWORD may cause undefined or undesirable operation. The HBI hardware does not protect against this operation. The HBI simply counts that four BYTEs have been read. Accessing the same internal register using two Index / Data register pairs may cause undefined or undesirable operation. The HBI hardware does not protect against this operation. Mixing reads and writes into the same Data register may cause undefined or undesirable operation. The HBI hardware does not protect against this operation.

A read BYTE / WORD counter keeps track of the number of reads. Each Data Register has its own BYTE / WORD counter. These counters are separate from the write counters above. At the trailing edge of the read cycle, the appropriate counter (based on the captured host address from above) is incremented. On the last read for the DWORD, an internal read is performed to update any Change on Read CSRs.

All of the read BYTE / WORD counters are reset if the power management mode is set to anything other than D0.

SPECIAL CSR HANDLING

Live Bits

Any register bit that is updated by a H/W event is held at the beginning of the read cycle to prevent it from changing during the read cycle.

Multiple BYTE / WORD Live Registers in 16 or 8-Bit Modes

Some internal registers have fields or related fields that span across multiple BYTEs or WORDs. For 16 and 8-bit data reads, it is possible that the value of these fields change between host read cycles. In order to prevent reading intermediate values, these registers are locked when the first byte or word is read and unlocked when the last byte or word is read.

The registers are unlocked if the power management mode is set to anything other than D0.

Change on Read Registers and FIFOs

FIFOs or “Change on Read” registers, are updated at the end of the read cycle.

For 16 and 8-bit modes, only one internal read cycle is indicated and occurs for the last byte or word.

Change on Read Live Register Bits

As described above, registers with live bits are held starting at the beginning of the read cycle and those that have multiple bits that span across BYTES or WORDS are also locked for 16 and 8-bit accesses. Although a H/W event that occurs during the hold or lock time would still update the live bit(s), the live bit(s) will be affected (cleared, etc.) at the end of the read cycle and the H/W event would be lost.

In order to prevent this, the individual CSRs defer the H/W event update until after the read or multiple reads.

Registers Polling During Reset or Initialization

Some registers support polling during reset or device initialization to determine when the device is accessible. For these registers, only one read may be performed without the need to read the other WORD or BYTEs. The same BYTE or WORD of the register may be re-read repeatedly.

A register that is 16 or 8-bit readable or readable during reset or device initialization, is noted in its register description.

9.5.4.3 Host Endianness

The device supports big and little endian host byte ordering based upon the endianness bits in the Host Bus Interface Configuration Register. When the appropriate endianness bit is low, host access is little endian and when high, host access is big endian. Endianness is specified for each Index / Data pair and for FIFO Direct Select accesses.

All internal busses are 32-bit with little endian byte ordering. Logic within the Host Bus Interface re-orders bytes based on the appropriate endianness bit, and the state of the least significant address lines (ADDR[1:0]).

Data path operations for the supported endian configurations and data bus sizes are illustrated in FIGURE 9-23: Little Endian Ordering on page 100 and FIGURE 9-24: Big Endian Ordering on page 101.

FIGURE 9-23: LITTLE ENDIAN ORDERING

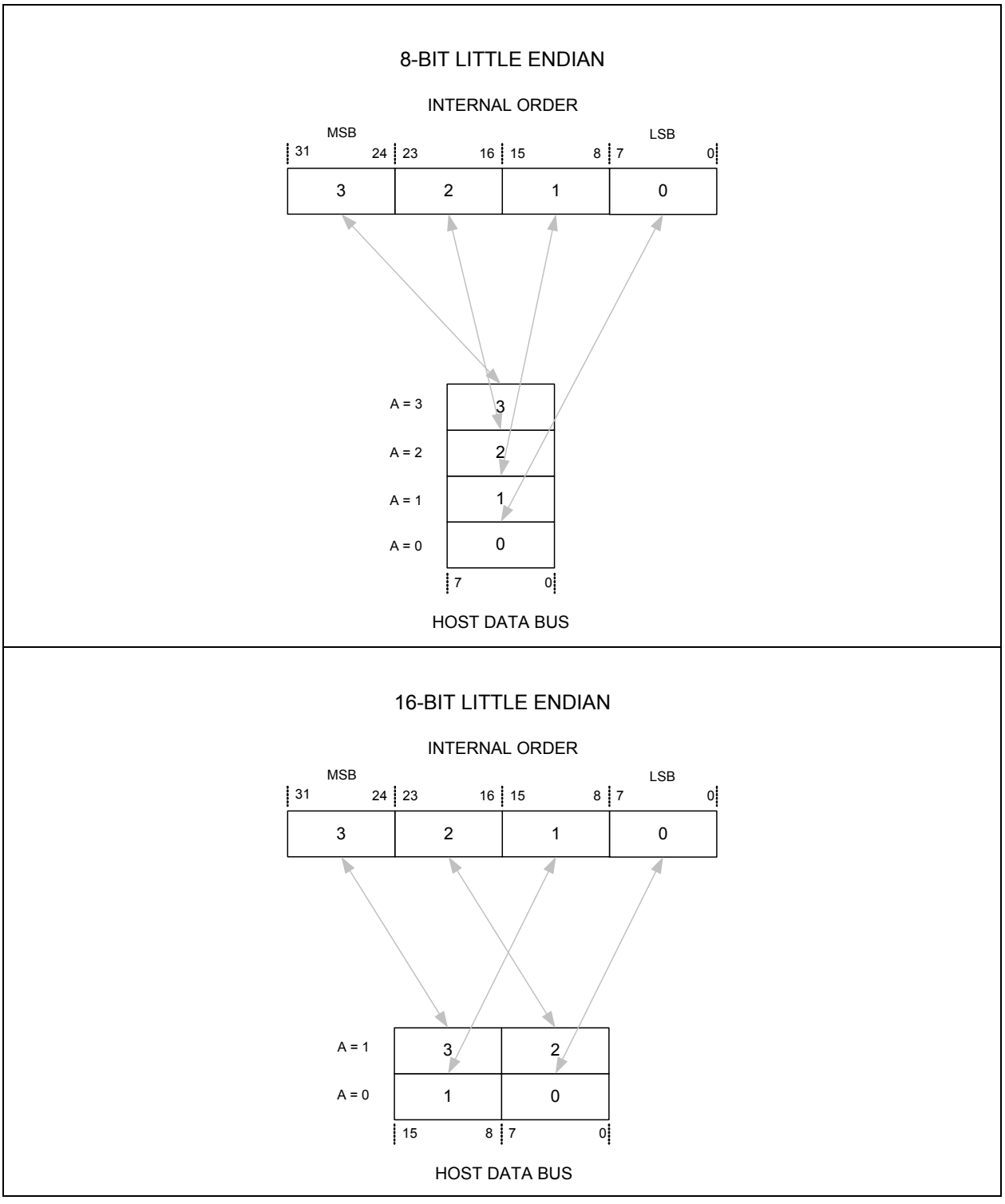
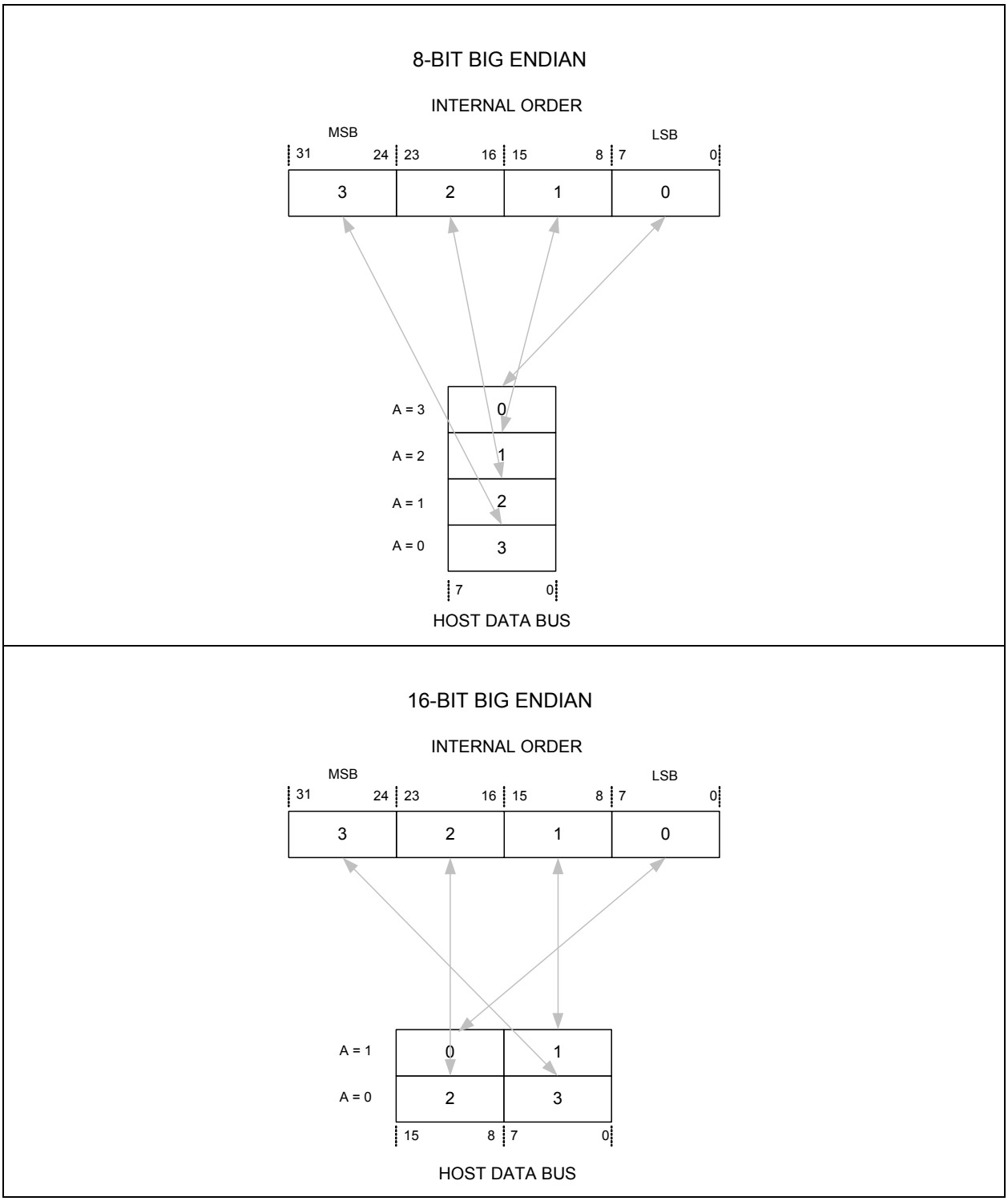


FIGURE 9-24: BIG ENDIAN ORDERING



9.5.5 TX AND RX FIFO ACCESS

9.5.5.1 TX and RX Status FIFO Peek Address Access

Normal read access to the TX or RX Status FIFO causes the FIFO to advance to its next entry.

For access to the TX and RX Status FIFO Peek addresses, FIFO does not advance to its next entry.

9.5.5.2 Index Register Bypass FIFO Access

In addition to the indexed access, the Index Registers can be bypassed and the FIFOs accessed at address 18h-1Bh. At this address, host write operations are to the TX Data FIFO and host read operations are from the RX Data FIFO . There is no associated Index Register.

Index Register Bypass and FIFO Direct Select accesses share the same read and write BYTE / WORD counters and the same write DWORD assembly registers.

The endianness of FIFO accesses using this method is specified by the FIFO Endianness bit in the Host Bus Interface Configuration Register.

9.5.5.3 FIFO Direct Select Access

In addition to the indexed access, a FIFO Direct Select signal is provided. This allows the host system to access the TX and RX Data FIFOs as if they were a large flat address space. When the **FIFOSEL** input is active, all host write operations are to the TX Data FIFO and all host read operations are from the RX Data FIFO. The lower host address signals are decoded in order to select the proper BYTE or WORD and to delimit DWORDs during a burst access.

Burst access is supported when reading the RX Data FIFO. With the **FIFOSEL** input active, **CS** and **RD** (or **ENB** with **RD_WR** indicating read) may be left active while the lower address lines toggle. Each change of the lower address bits provides the next WORD or BYTE of data. The HBI performs an internal FIFO pop (read cycle) when it detects that a DWORD boundary has been crossed (**A[2]** has toggled).

The endianness of FIFO Direct Select accesses is specified by the FIFO Endianness bit in the Host Bus Interface Configuration Register.

9.5.6 INDEXED ADDRESS MODE FUNCTIONAL TIMING DIAGRAMS

The following timing diagrams illustrate example indexed (non-multiplexed) addressing mode read and write cycles for various configurations and bus sizes. These diagrams do not cover every supported host bus permutation, but are selected to detail the main configuration differences (bus size, Configuration/Index/Data/FIFO-Direct cycles) within the indexed addressing mode of operation.

The following should be noted for the timing diagrams in this section:

- The diagrams in this section depict active-high **CS**, **RD**, and **WR** signals. The polarities of these signals are selectable via the HBI_cs_polarity_strap, HBI_rd_rdwr_polarity_strap, and HBI_wr_en_polarity_strap, respectively. Refer to Section 9.3, "Control Line Polarity," on page 74 for additional details.
- The diagrams in this section depict little endian byte ordering. However, configurable big and little endianess are supported via the endianness bits in the Host Bus Interface Configuration Register. Endianess changes only the order of the bytes involved, and not the overall timing requirements. Refer to Section 9.5.4.3, "Host Endianness," on page 99 for additional information.
- The diagrams in this section utilize **RD** and **WR** signals. Alternative **RD_WR** and **ENB** signaling is also supported, similar to the multiplexed example in Section 9.4.4.3, "RD_WR / ENB Control Mode Examples". The HBI read/write mode is selectable via the HBI_rw_mode_strap. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI_rd_rdwr_polarity_strap, and HBI_wr_en_polarity_strap.
- When accessing internal registers or FIFOs in 16 and 8-bit modes, consecutive address cycles must be within the same DWORD until the DWORD is completely accessed (some internal registers are excluded from this requirement). Although BYTEs and WORDs can be accessed in any order, the diagrams in this section depict accessing the lower address BYTE or WORD first.

9.5.6.1 Configuration Register Data Access

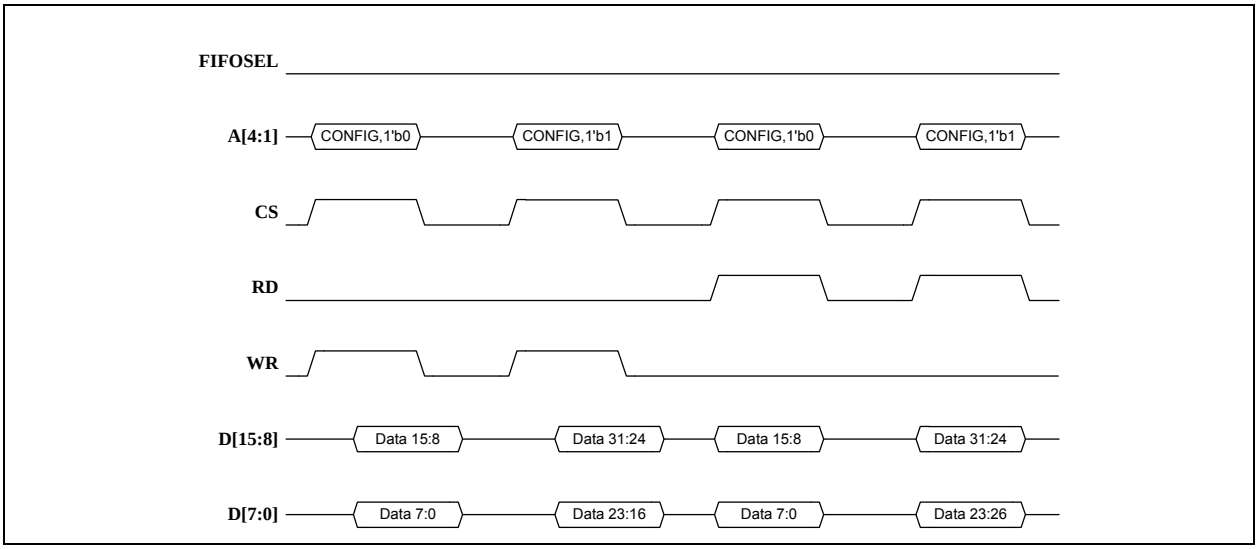
The figures in this section detail configuration register read and write operations in indexed address mode for 16 and 8-bit modes.

16-BIT READ AND WRITE

For writes, the address is set to access the lower WORD of the Configuration Register and **FIFOSEL** is held low. Data on **D[15:0]** is written on the trailing edge of **WR**. The cycle repeats for the upper WORD of the Configuration Register, if desired by the host.

For reads, the address is set to access the lower WORD of the Configuration Register and **FIFOSEL** is held low. Read data is driven on **D[15:0]** during **RD** active. The cycle repeats for the upper WORD of the Configuration Register, if desired by the host.

FIGURE 9-25: INDEXED ADDRESSING CONFIGURATION REGISTER ACCESS - 16-BIT WRITE/ READ

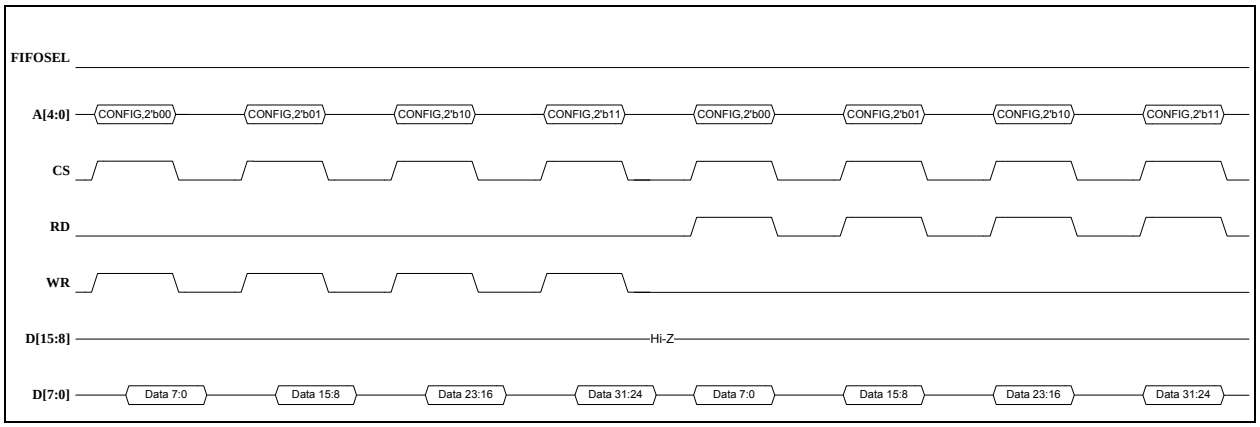


8-BIT READ AND WRITE

For writes, the address is set to access the lower BYTE of the Configuration Register and **FIFOSEL** is held low. Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Configuration Register, if desired by the host.

For reads, the address is set to access the lower BYTE of the Configuration Register and **FIFOSEL** is held low. Read data is driven on **D[7:0]** during **RD** active. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Configuration Register, if desired by the host.

FIGURE 9-26: INDEXED ADDRESSING CONFIGURATION REGISTER ACCESS - 8-BIT WRITE/ READ



9.5.6.2 Index Register Data Access

The figures in this section detail index register read and write operations in indexed address mode for 16 and 8-bit modes.

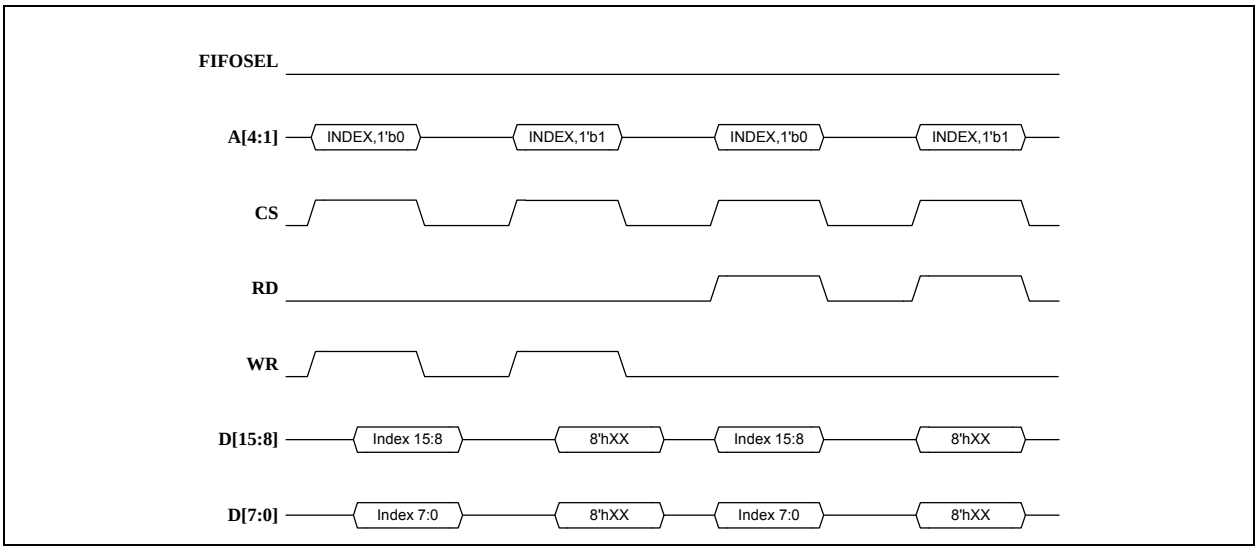
16-BIT READ AND WRITE

For writes, the address is set to access the lower WORD of one of the Index Registers and **FIFOSEL** is held low. Data on **D[15:0]** is written on the trailing edge of **WR**. The cycle repeats for the upper WORD of the Index Register, if desired by the host.

For reads, the address is set to access the lower WORD of one of the Index Registers and **FIFOSEL** is held low. Read data is driven on **D[15:0]** during **RD** active. The cycle repeats for the upper WORD of the Index Register, if desired by the host.

Note:	The upper WORD of Index Registers is reserved and don't care. Therefore reads and writes to that WORD are not useful.

FIGURE 9-27: INDEXED ADDRESSING INDEX REGISTER ACCESS - 16-BIT WRITE/READ



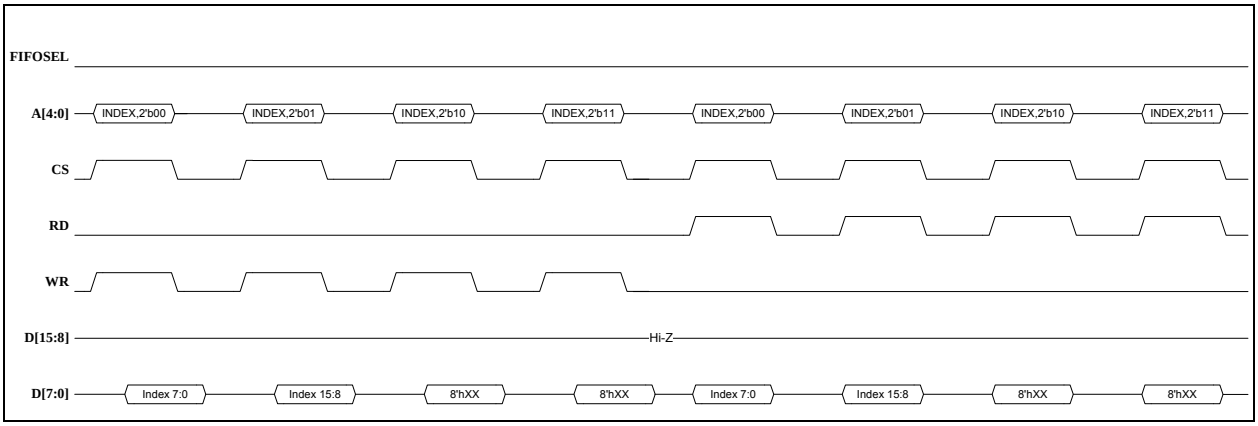
8-BIT READ AND WRITE

For writes, the address is set to access the lower BYTE of one of the Index Registers and **FIFOSEL** is held low. Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Index Register, if desired by the host.

For reads, the address is set to access the lower BYTE of one of the Index Registers and **FIFOSEL** is held low. Read data is driven on **D[7:0]** during **RD** active. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Index Register, if desired by the host.

Note:	The upper WORD of Index Registers is reserved and don't care. Therefore reads and writes to those BYTEs are not useful.

FIGURE 9-28: INDEXED ADDRESSING INDEX REGISTER ACCESS - 8-BIT WRITE/READ



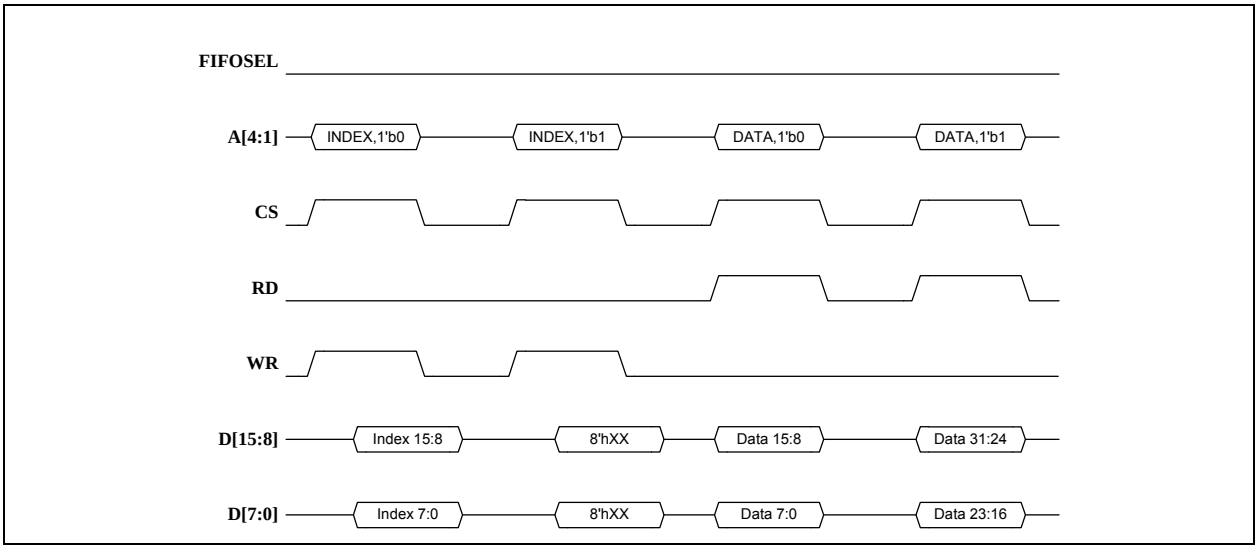
9.5.6.3 Internal Register Data Access

The figures in this section detail typical internal register data read and write cycles in indexed address mode for 16 and 8-bit modes. This includes an index register write followed by either a data read or write.

16-BIT READ

One of the Index Registers is set as described above. The address is then set to access the lower WORD of the corresponding Data Register and **FIFOSEL** is held low. Read data is driven on **D[15:0]** during **RD** active. The cycle repeats for the upper WORD of the Data Register.

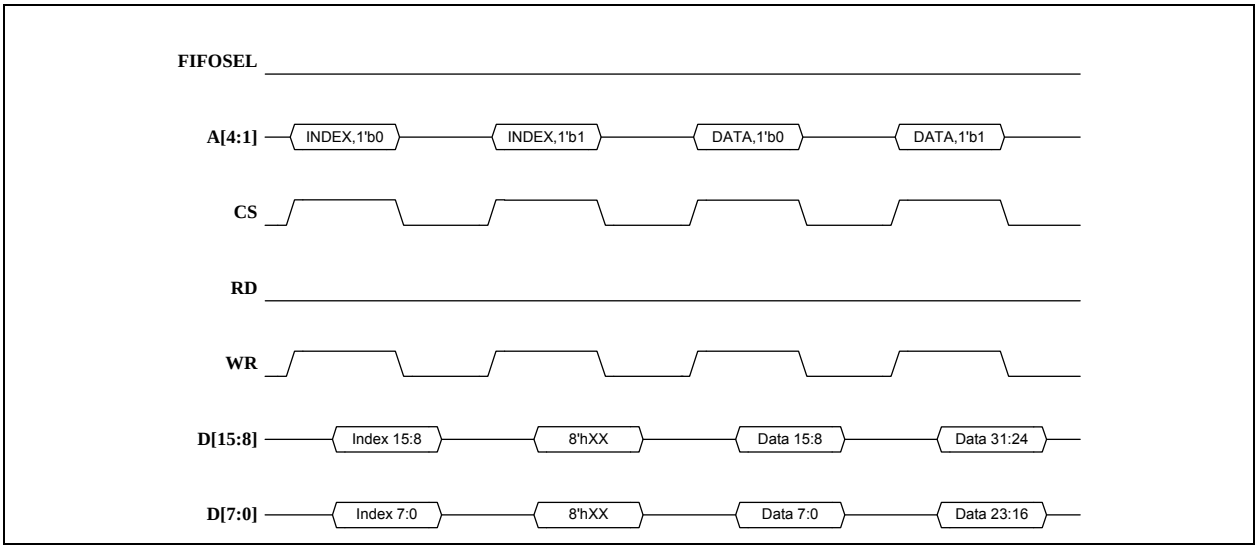
FIGURE 9-29: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 16-BIT READ



16-BIT WRITE

One of the Index Registers is set as described above. The address is then set to access the corresponding Data Register and **FIFOSEL** is held low. Data on **D[15:0]** is written on the trailing edge of **WR**. The cycle repeats for the upper WORD of the Data Register.

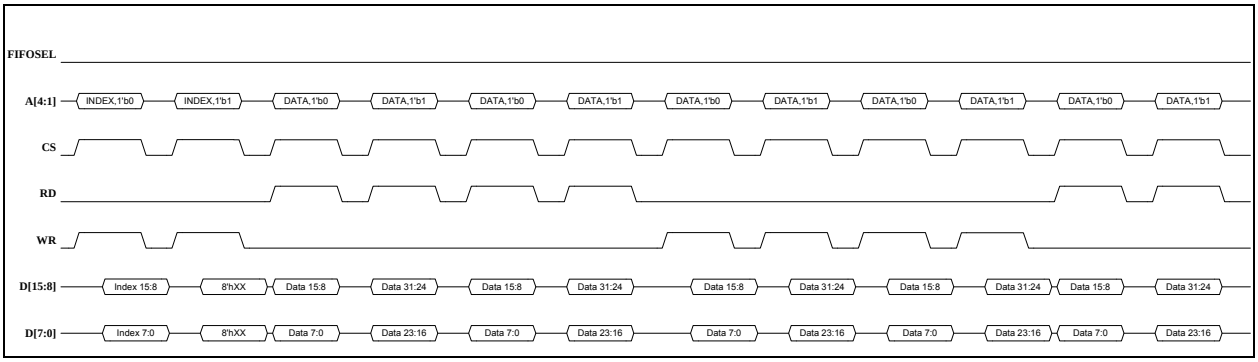
FIGURE 9-30: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 16-BIT WRITE



16-BIT READS AND WRITES TO CONSTANT INTERNAL ADDRESS

One of the Index Registers is set as described above. A mix of reads and writes on **D[15:0]** follows, with each read or write consisting of an access to both the lower and upper WORDs of the corresponding Data Register.

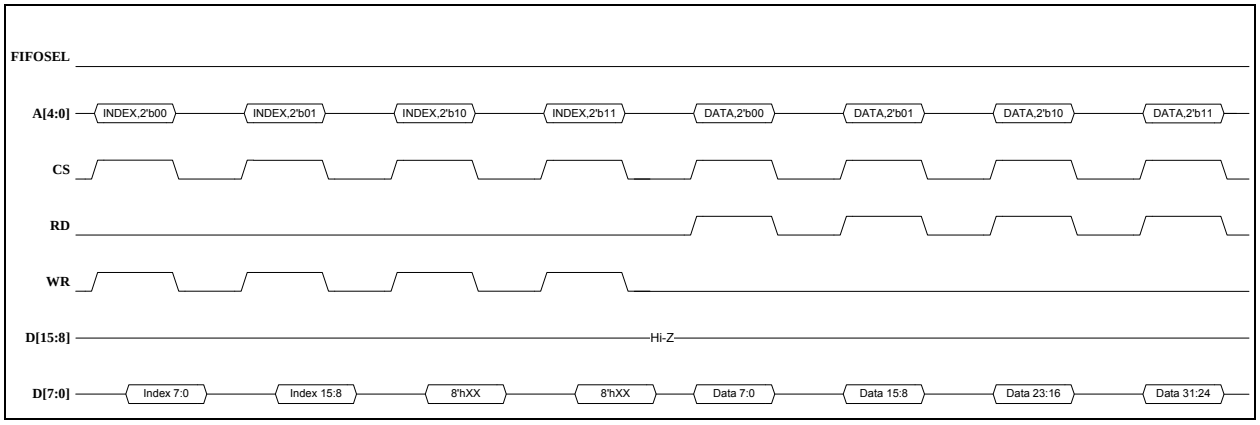
FIGURE 9-31: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 16-BIT READS/ WRITES CONSTANT ADDRESS



8-BIT READ

One of the Index Registers is set as described above. The address is then set to access the lower BYTE of the corresponding Data Register and **FIFOSEL** is held low. Read data is driven on **D[7:0]** during **RD** active. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Data Register.

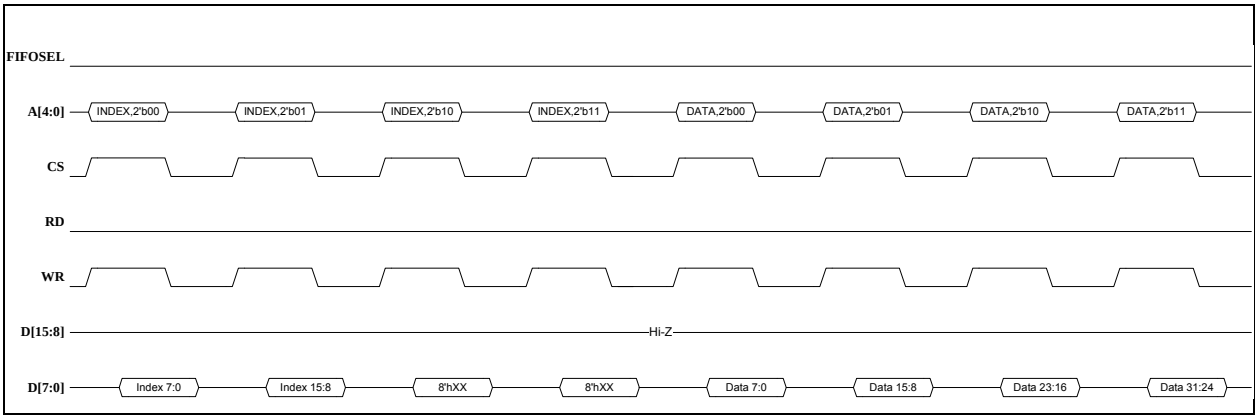
FIGURE 9-32: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 8-BIT READ



8-BIT WRITE

One of the Index Registers is set as described above. The address is then set to access the corresponding Data Register and **FIFOSEL** is held low. Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining BYTEs of the Data Register.

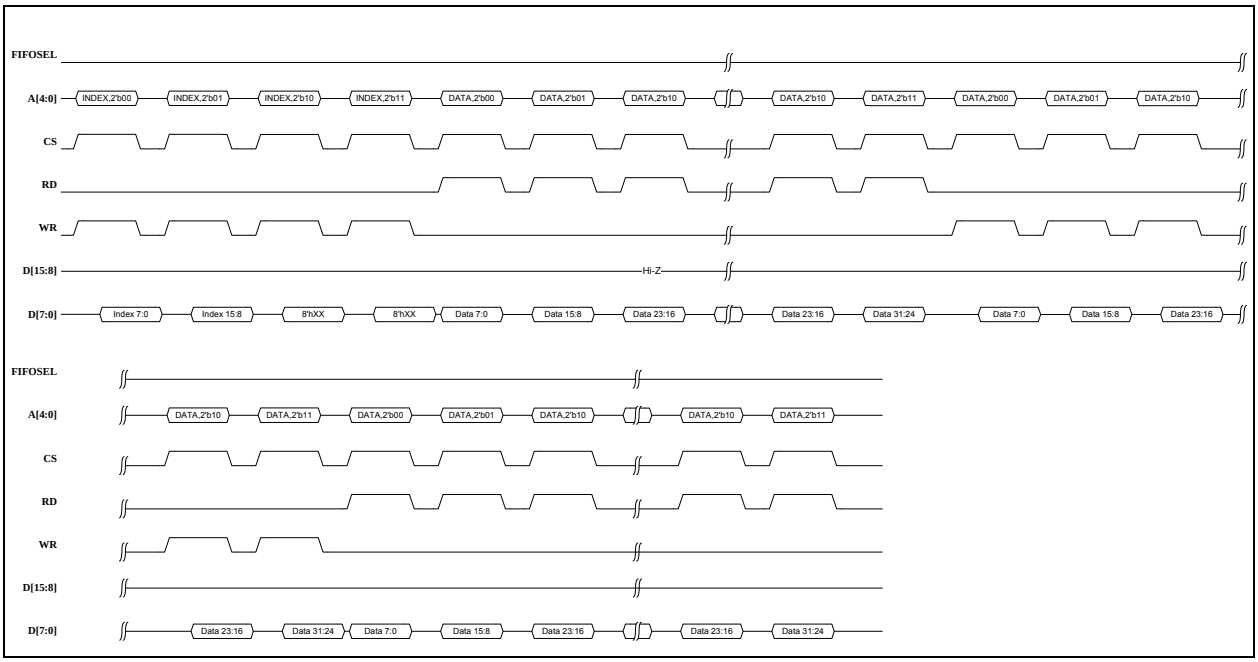
FIGURE 9-33: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 8-BIT WRITE



8-BIT READS AND WRITES TO CONSTANT INTERNAL ADDRESS

One of the Index Registers is set as described above. A mix of reads and writes on **D[7:0]** follows, with each read or write consisting of an access to all four BYTES of the corresponding Data Register.

FIGURE 9-34: INDEXED ADDRESSING INTERNAL REGISTER DATA ACCESS - 8-BIT READS/ WRITES CONSTANT ADDRESS



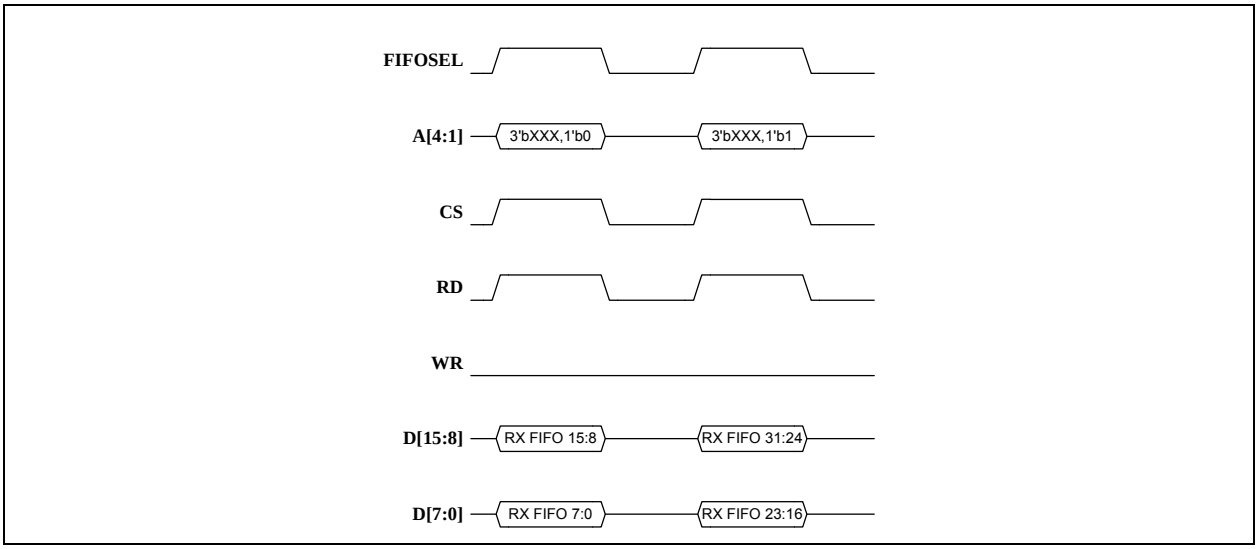
9.5.6.4 FIFO Direct Select Access

The figures in this section detail FIFO Direct Select (**FIFOSEL**) read and write cycles in indexed address mode for 16 and 8-bit modes. Additionally, FIFO Direct Select Burst reads are shown for 16, and 8-bit modes. FIFO Direct Select Burst mode supports up to 8 DWORDs of consecutive reads. FIFO Direct Select (**FIFOSEL**) read and write cycles do not required an index register write.

16-BIT READ

FIFOSEL is set high and address bit 1 is set to access the lower WORD of the FIFO, while address bits 4:2 are don't care. Read data is driven on **D[15:0]** during **RD** active. The cycle repeats for the upper WORD of the FIFO.

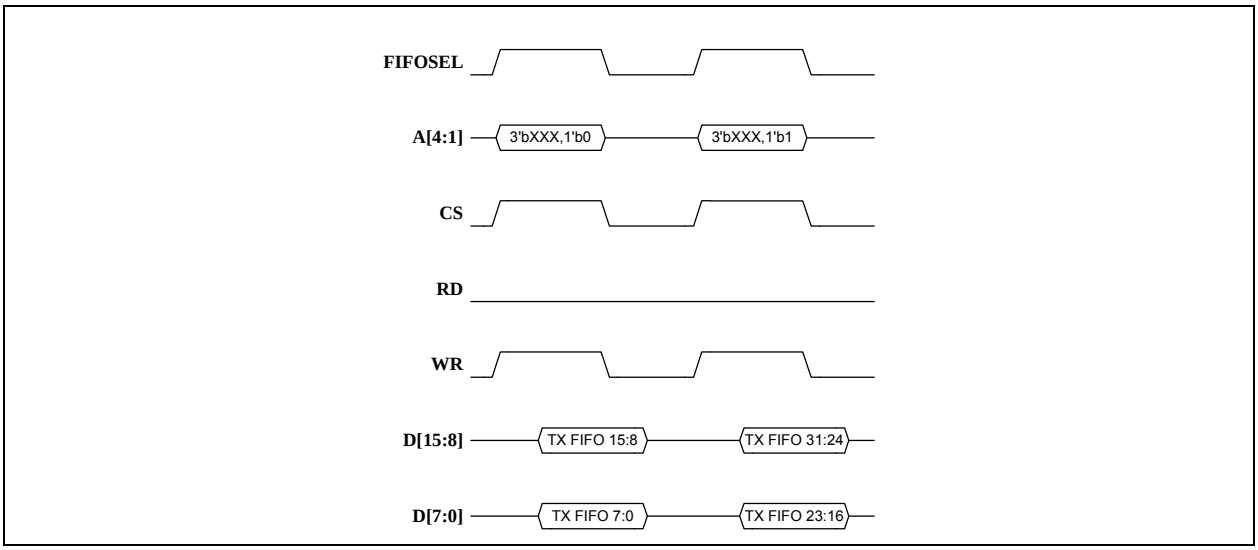
FIGURE 9-35: INDEXED ADDRESSING FIFO DIRECT SELECT ACCESS - 16-BIT READ



16-BIT WRITE

FIFOSEL is set high and address bit 1 is set to access the lower WORD of the FIFO, while address bits 4:2 are don't care. Data on **D[15:0]** is written on the trailing edge of **WR**. The cycle repeats for the upper WORD of the FIFO.

FIGURE 9-36: INDEXED ADDRESSING FIFO DIRECT SELECT ACCESS - 16-BIT WRITE

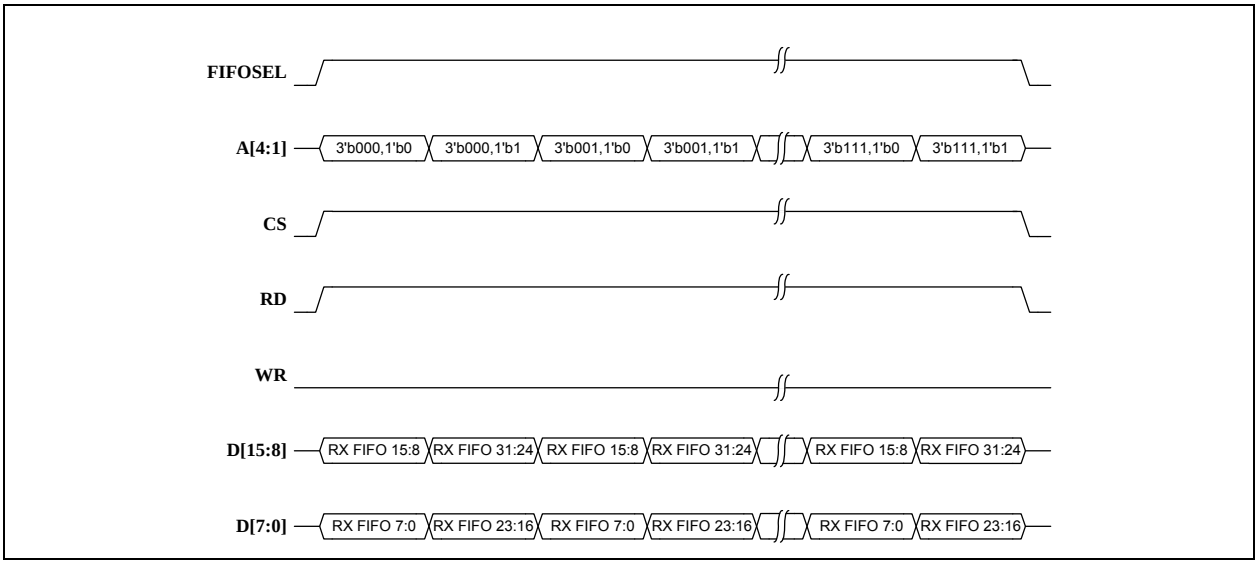


16-BIT BURST READ

FIFOSEL is set high and address bit 1 is set to access the lower WORD of the FIFO, while address bits 4:2 start at 0. Read data is driven on **D[15:0]** during **RD** active. Address bit 1 is then set to access the WORD of the FIFO as **RD** is held active.

While **RD** is held active, address bits 4:2 are cycled from 0 through 7 to access the 8 DWORDs as address bit 1 is toggled to access each WORD. Fresh data is supplied each time **A[1]** toggles. The FIFO is popped when **A[2]** toggles.

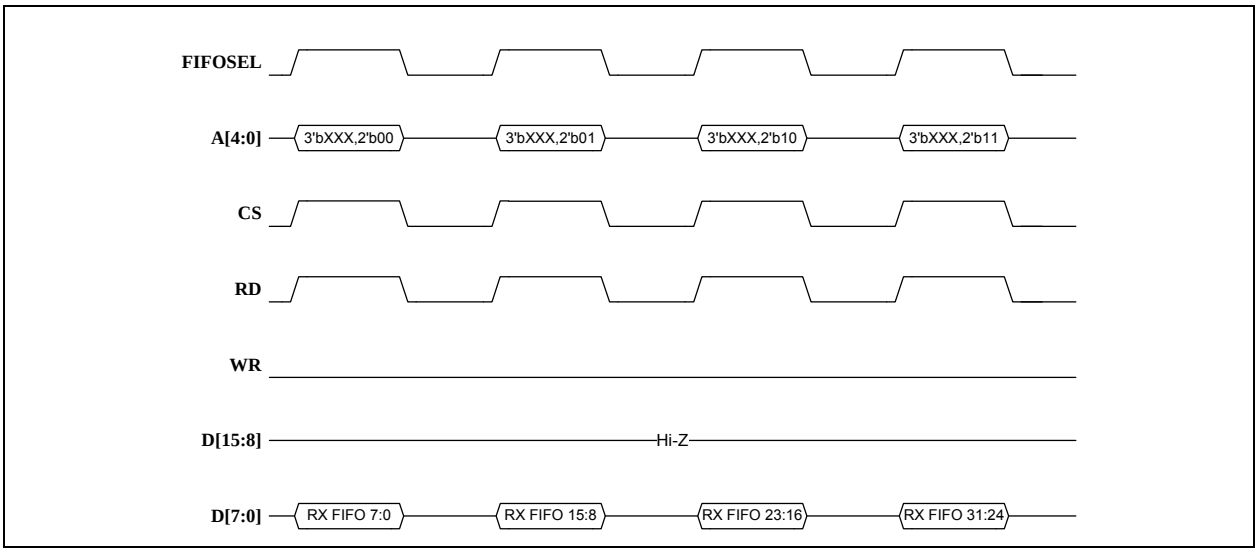
FIGURE 9-37: INDEXED ADDRESSING FIFO DIRECT SELECT ACCESS - 16-BIT BURST READ



8-BIT READ

FIFOSEL is set high and address bits 1 and 0 are set to access the lower BYTE of the FIFO, while address bits 4:2 are don't care. Read data is driven on **D[7:0]** during **RD** active. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining 3 BYTEs of the FIFO's DWORD.

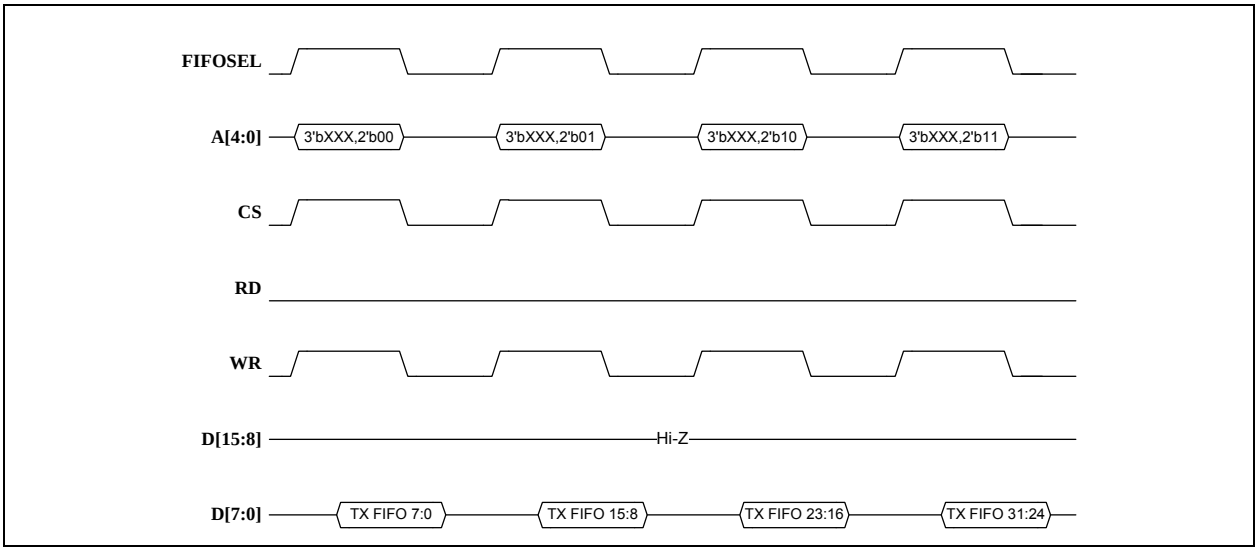
FIGURE 9-38: INDEXED ADDRESSING FIFO DIRECT SELECT ACCESS - 8-BIT READ



8-BIT WRITE

FIFOSEL is set high and address bits 1 and 0 are set to access the lower BYTE of the FIFO, while address bits 4:2 are don't care. Data on **D[7:0]** is written on the trailing edge of **WR**. **D[15:8]** pins are not used or driven. The cycle repeats for the remaining 3 BYTEs of the FIFO's DWORD.

FIGURE 9-39: INDEXED ADDRESSING FIFO DIRECT SELECT ACCESS - 8-BIT WRITE

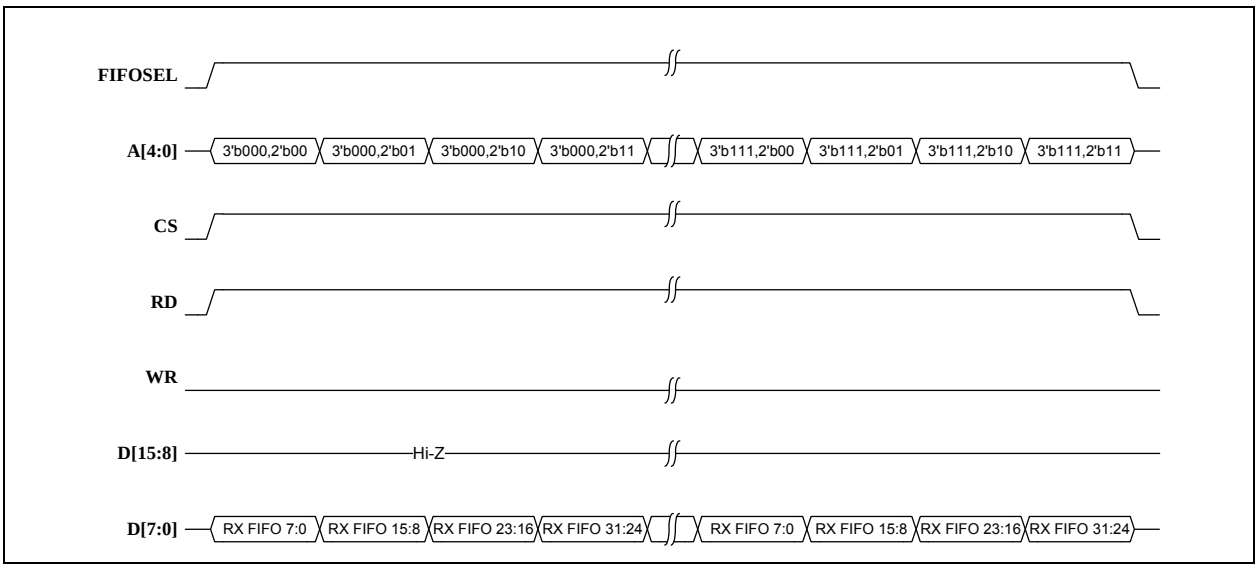


8-BIT BURST READ

FIFOSEL is set high and address bits 1 and 0 are set to access the lower BYTE of the FIFO, while address bits 4:2 start at 0. Read data is driven on **D[7:0]** during **RD** active. **D[15:8]** pins are not used or driven. Address bits 1 & 0 are then cycled from 0 through 3 to access the next 3 BYTEs of the FIFO as **RD** is held active.

While **RD** is held active, address bits 4:2 are cycled from 0 through 7 to access the 8 DWORDs as address bits 1 & 0 are cycled from 0 through 3 to access each BYTE. Fresh data is supplied each time **A[0]** toggles. The FIFO is popped when **A[2]** toggles.

FIGURE 9-40: INDEXED ADDRESSING FIFO DIRECT SELECT ACCESS - 8-BIT BURST READ



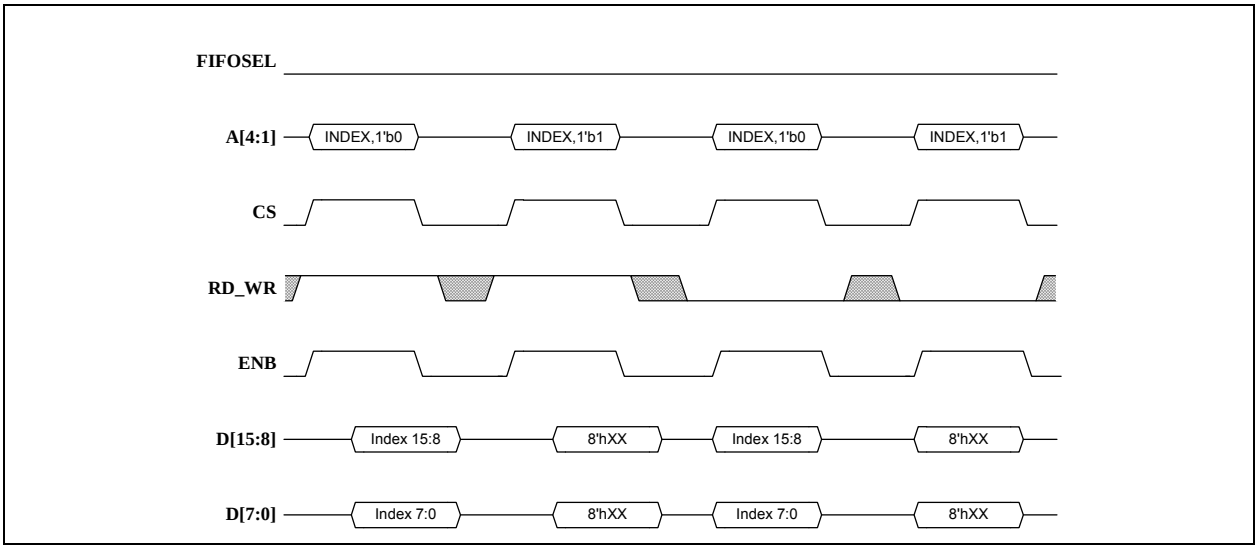
9.5.6.5 RD_WR / ENB Control Mode Examples

The figures in this section detail read and write operations utilizing the alternative **RD_WR** and **ENB** signaling. The HBI read/write mode is selectable via the HBI_rw_mode_strap.

Note:	The examples in this section detail 16-bit mode with access to an Index Register. However, the **RD_WR** and **ENB** signaling can be used identically for all other accesses including FIFO Direct Select Access. The examples in this section show the **ENB** signal active-high and the **RD_WR** signal low for read and high for write. The polarities of the **RD_WR** and **ENB** signals are selectable via the HBI_rd_rdwr_polarity_strap, and HBI_wr_en_polarity_strap.

16-BIT

FIGURE 9-41: INDEXED ADDRESSING RD_WR / ENB CONTROL MODE EXAMPLE - 16-BIT WRITE/READ



9.5.7 INDEXED ADDRESSING MODE TIMING REQUIREMENTS

The following figures and tables specify the timing requirements during Indexed Address mode. Since timing requirements are similar across the multitude of operations (e.g. 8 vs. 16-bit, Index vs. Configuration vs. Data registers, FIFO Direct Select), many timing requirements are illustrated in the same figures and do not necessarily represent any particular functional operation.

The following should be noted for the timing specifications in this section:

- The diagrams in this section depict active-high **CS**, **RD**, **WR**, **RD_WR** and **ENB** signals. The polarities of these signals are selectable via the HBI_cs_polarity_strap, HBI_rd_rdwr_polarity_strap, and HBI_wr_en_polarity_strap, respectively. Refer to Section 9.3, "Control Line Polarity," on page 74 for additional details.
- A read cycle maybe followed by followed by a write cycle or another read cycle. A write cycle maybe followed by followed by a read cycle or another write cycle. These are shown in dashed line.

9.5.7.1 Read Timing Requirements

If **RD** and **WR** signaling is used, a host read cycle begins when **RD** is asserted with **CS** active. The cycle ends when **RD** is de-asserted. **CS** maybe asserted and de-asserted along with **RD** but not during **RD** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host read cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a read. The cycle ends when **ENB** is de-asserted. **CS** maybe asserted and de-asserted along with **ENB** but not during **ENB** active.

Please refer to Section 9.5.6, "Indexed Address Mode Functional Timing Diagrams," on page 102 for functional descriptions.

FIGURE 9-42: INDEXED ADDRESSING READ CYCLE TIMING

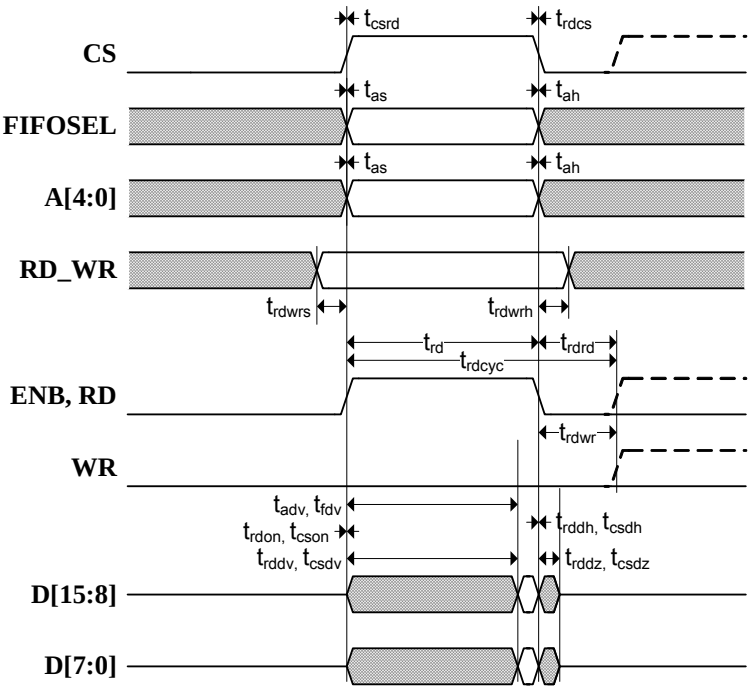


TABLE 9-4: INDEXED ADDRESSING READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csrd}	**CS** Setup to **RD** or **ENB** Active	0			ns
t_{rdcs}	**CS** Hold from **RD** or **ENB** Inactive	0			ns
t_{as}	Address, **FIFOSEL** Setup to **RD** or **ENB** Active	0			ns
t_{ah}	Address, **FIFOSEL** Hold from to **RD** or **ENB** Inactive	0			ns
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 14	5			ns
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 14	5			ns
t_{rdon}	**RD** or **ENB** to Data Buffer Turn On	0			ns
t_{rddv}	**RD** or **ENB** Active to Data Valid			30	ns
t_{rddh}	Data Output Hold Time from **RD** or **ENB** Inactive	0			ns
t_{rddz}	Data Buffer Turn Off Time from **RD** or **ENB** Inactive			9	ns
t_{cson}	**CS** to Data Buffer Turn On	0			ns
t_{csdv}	**CS** Active to Data Valid			30	ns
t_{csdh}	Data Output Hold Time from **CS** Inactive	0			ns
t_{csdz}	Data Buffer Turn Off Time from **CS** Inactive			9	ns
t_{fdv}	**FIFOSEL** to Data Valid			30	ns
t_{adv}	Address to Data Valid			30	ns
t_{rd}	**RD** or **ENB** Active Time	32			ns
t_{rdcyc}	**RD** or **ENB** Cycle Time	45			ns
t_{rdrd}	RD or **ENB** De-assertion Time before Next **RD** or **ENB**	13			ns
t_{rdwr}	**RD** De-assertion Time before Next **WR** Note 15	13			ns

Note 14: **RD_WR** and **ENB** signaling.

Note 15: **RD** and **WR** signaling.

Note: Timing values are with respect to an equivalent test load of 25 pF.

9.5.7.2 FIFO Direct Select Burst Timing Requirements

If **RD** and **WR** signaling is used, a host burst read from the FIFO begins when **RD** is asserted with **CS** active and **FIFOSEL** high. As the address changes, the next data is read. The cycle ends when **RD** is de-asserted. **CS** maybe asserted and de-asserted along with **RD** but not during **RD** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host burst read from the **FIFO** begins when **ENB** is asserted with **CS** active, **RD_WR** indicating a read and **FIFOSEL** high. As the address changes, the next data is read. The cycle ends when **ENB** is de-asserted. **CS** maybe asserted and de-asserted along with **ENB** but not during **ENB** active.

Please refer to Section 9.5.6, "Indexed Address Mode Functional Timing Diagrams," on page 102 for functional descriptions.

FIGURE 9-43: INDEXED ADDRESSING FIFO DIRECT SELECT BURST READ CYCLE TIMING

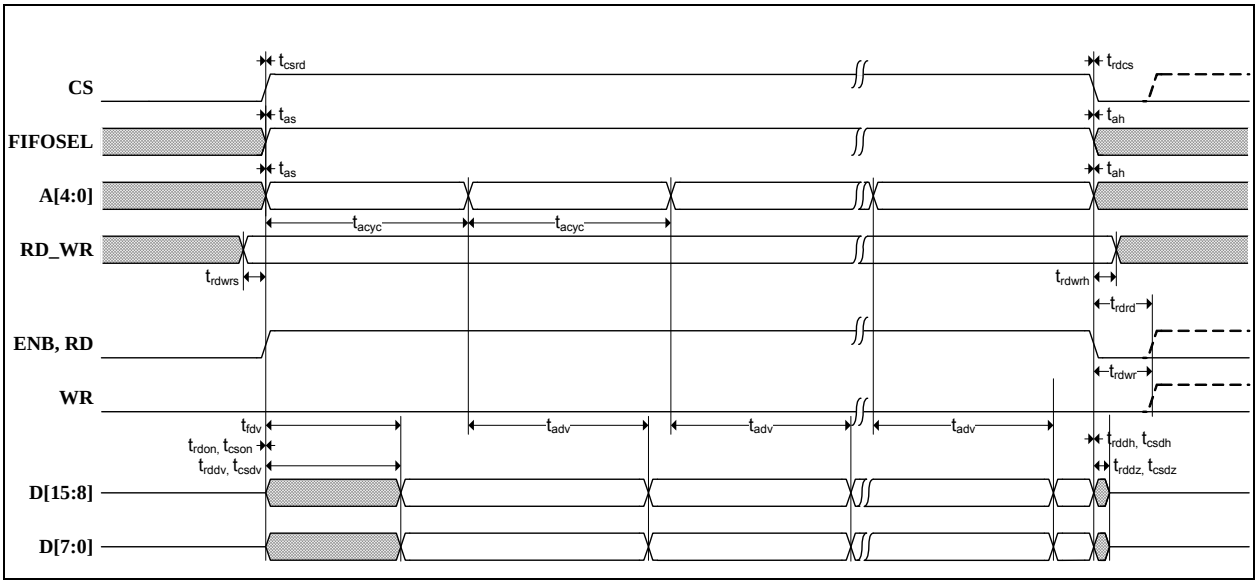


TABLE 9-5: INDEXED ADDRESSING FIFO DIRECT SELECT BURST READ CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{csrd}	**CS** Setup to **RD** or **ENB** Active	0			ns
t_{rdcs}	**CS** Hold from **RD** or **ENB** Inactive	0			ns
t_{as}	Address, **FIFOSEL** Setup to **RD** or **ENB** Active	0			ns
t_{ah}	Address, **FIFOSEL** Hold from to **RD** or **ENB** Inactive	0			ns
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 16	5			ns
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 16	5			ns
t_{rdon}	**RD** or **ENB** to Data Buffer Turn On	0			ns
t_{rddv}	**RD** or **ENB** Active to Data Valid			30	ns
t_{rddh}	Data Output Hold Time from **RD** or **ENB** Inactive	0			ns
t_{rddz}	Data Buffer Turn Off Time from **RD** or **ENB** Inactive			9	ns

TABLE 9-5: INDEXED ADDRESSING FIFO DIRECT SELECT BURST READ CYCLE TIMING VALUES (CONTINUED)

Symbol	Description	Min	Typ	Max	Units
t_{cson}	**CS** to Data Buffer Turn On	0			ns
t_{csdv}	**CS** Active to Data Valid			30	ns
t_{csdh}	Data Output Hold Time from **CS** Inactive	0			ns
t_{csdz}	Data Buffer Turn Off Time from **CS** Inactive			9	ns
t_{fdv}	**FIFOSEL** to Data Valid			30	ns
t_{adv}	Address Change to Next Data Valid			40	ns
t_{acyc}	Address Cycle Time	45			ns
t_{rdrd}	**RD** or **ENB** De-assertion Time before Next **RD** or **ENB**	13			ns
t_{rdwr}	**RD** De-assertion Time before Next **WR** Note 17	13			ns

Note 16: **RD_WR** and **ENB** signaling.

Note 17: **RD** and **WR** signaling.

Note: Timing values are with respect to an equivalent test load of 25 pF.

9.5.7.3 Write Timing Requirements

If **RD** and **WR** signaling is used, a host write cycle begins when **WR** is asserted with **CS** active. The cycle ends when **WR** is de-asserted. **CS** maybe asserted and de-asserted along with **WR** but not during **WR** active.

Alternatively, if **RD_WR** and **ENB** signaling is used, a host write cycle begins when **ENB** is asserted with **CS** active and **RD_WR** indicating a write. The cycle ends when **ENB** is de-asserted. **CS** maybe asserted and de-asserted along with **ENB** but not during **ENB** active.

Please refer to Section 9.5.6, "Indexed Address Mode Functional Timing Diagrams," on page 102 for functional descriptions.

FIGURE 9-44: INDEXED ADDRESSING WRITE CYCLE TIMING

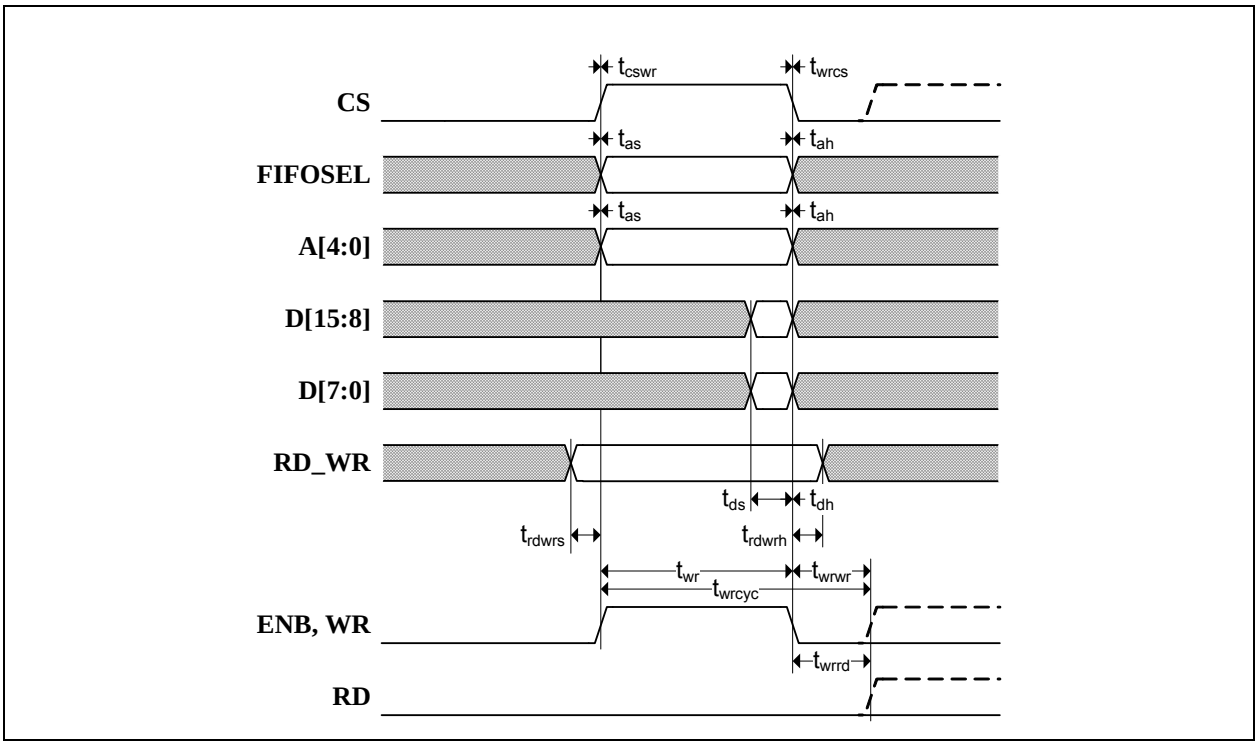


TABLE 9-6: INDEXED ADDRESSING WRITE CYCLE TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{cswr}	**CS** Setup to **WR** or **ENB** Active	0			ns
t_{wrcs}	**CS** Hold from **WR** or **ENB** Inactive	0			ns
t_{as}	Address, **FIFOSEL** Setup to **WR** or **ENB** Active	0			ns
t_{ah}	Address, **FIFOSEL** Hold from to **WR** or **ENB** Inactive	0			ns
t_{rdwrs}	**RD_WR** Setup to **ENB** Active Note 18	5			ns
t_{rdwrh}	**RD_WR** Hold from **ENB** Inactive Note 18	5			ns
t_{ds}	Data Setup to **WR** or **ENB** Inactive	7			ns
t_{dh}	Data Hold from **WR** or **ENB** Inactive	0			ns

TABLE 9-6: INDEXED ADDRESSING WRITE CYCLE TIMING VALUES (CONTINUED)

Symbol	Description	Min	Typ	Max	Units
t_{wr}	**WR** or **ENB** Active Time	32			ns
t_{wrcyc}	**WR** or **ENB** Cycle Time	45			ns
t_{wrwr}	**WR** or **ENB** De-assertion Time before Next **WR** or **ENB**	13			ns
t_{wrrd}	**WR** De-assertion Time before Next **RD** Note 19	13			ns

Note 18: **RD_WR** and **ENB** signaling.

Note 19: **RD** and **WR** signaling.

10.0 SPI/SQI SLAVE

10.1 Functional Overview

The SPI/SQI Slave module provides a low pin count synchronous slave interface that facilitates communication between the device and a host system. The SPI/SQI Slave allows access to the System CSRs and internal FIFOs and memories. It supports single and multiple register read and write commands with incrementing, decrementing and static addressing. Single, Dual and Quad bit lanes are supported in SPI mode with a clock rate of up to 80 MHz. SQI mode always uses four bit lanes and also operates at up to 80 MHz.

The following is an overview of the functions provided by the SPI/SQI Slave:

- **Serial Read:** 4-wire (clock, select, data in and data out) reads at up to 30 MHz. Serial command, address and data. Single and multiple register reads with incrementing, decrementing or static addressing.
- **Fast Read:** 4-wire (clock, select, data in and data out) reads at up to 80 MHz. Serial command, address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- **Dual / Quad Output Read:** 4 or 6-wire (clock, select, data in / out) reads at up to 80 MHz. Serial command and address, parallel data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- **Dual / Quad I/O Read:** 4 or 6-wire (clock, select, data in / out) reads at up to 80 MHz. Serial command, parallel address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- **SQI Read:** 6-wire (clock, select, data in / out) writes at up to 80 MHz. Parallel command, address and data. Dummy byte(s) for first access. Single and multiple register reads with incrementing, decrementing or static addressing.
- **Write:** 4-wire (clock, select, data in and data out) writes at up to 80 MHz. Serial command, address and data. Single and multiple register writes with incrementing, decrementing or static addressing.
- **Dual / Quad Data Write:** 4 or 6-wire (clock, select, data in / out) writes at up to 80 MHz. Serial command and address, parallel data. Single and multiple register writes with incrementing, decrementing or static addressing.
- **Dual / Quad Address / Data Write:** 4 or 6-wire (clock, select, data in / out) writes at up to 80 MHz. Serial command, parallel address and data. Single and multiple register writes with incrementing, decrementing or static addressing.
- **SQI Write:** 6-wire (clock, select, data in / out) writes at up to 80 MHz. Parallel command, address and data. Single and multiple register writes with incrementing, decrementing or static addressing.

10.2 SPI/SQI Slave Operation

Input data on the **SIO[3:0]** pins is sampled on the rising edge of the **SCK** input clock. Output data is sourced on the **SIO[3:0]** pins with the falling edge of the clock. The **SCK** input clock can be either an active high pulse or an active low pulse. When the **SCS#** chip select input is high, the **SIO[3:0]** inputs are ignored and the **SIO[3:0]** outputs are three-stated.

In SPI mode, the 8-bit instruction is started on the first rising edge of the input clock after **SCS#** goes active. The instruction is always input serially on **SI/SIO0**.

For read and write instructions, two address bytes follow the instruction byte. Depending on the instruction, the address bytes are input either serially, or 2 or 4 bits per clock. Although all registers are accessed as DWORDs, the address field is considered a byte address. Fourteen address bits specify the address. Bits 15 and 14 of the address field specifies that the address is auto-decremented (10b) or auto-incremented (01b) for continuous accesses.

For some read instructions, dummy byte cycles follow the address bytes. The device does not drive the outputs during the dummy byte cycles. The dummy byte(s) are input either serially, or 2 or 4 bits per clock.

For read and write instructions, one or more 32-bit data fields follow the dummy bytes (if present, else they follow the address bytes). The data is input either serially, or 2 or 4 bits per clock.

SQI mode is entered from SPI with the Enable Quad I/O (EQIO) instruction. Once in SQI mode, all further command, addresses, dummy bytes and data bytes are 4 bits per clock. SQI mode can be exited using the Reset Quad I/O (RSTQIO) instruction.

All instructions, addresses and data are transferred with the most-significant bit (msb) or di-bit (msd) or nibble (msn) first. Addresses are transferred with the most-significant byte (MSB) first. Data is transferred with the least-significant byte (LSB) first (little endian).

The SPI interface supports up to a 80 MHz input clock. Normal (non-high speed) reads instructions are limited to 30 MHz.

The SPI interface supports a minimum time of 50 ns between successive commands (a minimum SCS# inactive time of 50 ns).

The instructions supported in SPI mode are listed in Table 10-1. SQI instructions are listed in Table 10-2. Unsupported instructions are must not be used.

TABLE 10-1: SPI INSTRUCTIONS

Instruction	Description	Bit width Note 1	Inst. code	Addr. Bytes	Dummy Bytes	Data bytes	Max Freq.
Configuration							
EQIO	Enable SQI	1-0-0	38h	0	0	0	80 MHz
RSTQIO	Reset SQI	1-0-0	FFh	0	0	0	80 MHz
Read							
READ	Read	1-1-1	03h	2	0	4 to ∞	30 MHz
FASTREAD	Read at higher speed	1-1-1	0Bh	2	1	4 to ∞	80 MHz
SDOR	SPI Dual Output Read	1-1-2	3Bh	2	1	4 to ∞	80 MHz
SDIOR	SPI Dual I/O Read	1-2-2	BBh	2	2	4 to ∞	80 MHz
SQOR	SPI Quad Out-put Read	1-1-4	6Bh	2	1	4 to ∞	80 MHz
SQIOR	SPI Quad I/O Read	1-4-4	EBh	2	4	4 to ∞	80 MHz
Write							
WRITE	Write	1-1-1	02h	2	0	4 to ∞	80 MHz
SDDW	SPI Dual Data Write	1-1-2	32h	2	0	4 to ∞	80 MHz
SDADW	SPI Dual Address / Data Write	1-2-2	B2h	2	0	4 to ∞	80 MHz
SQDW	SPI Quad Data Write	1-1-4	62h	2	0	4 to ∞	80 MHz
SQADW	SPI Quad Address / Data Write	1-4-4	E2h	2	0	4 to ∞	80 MHz

Note 1: The bit width format is: command bit width, address / dummy bit width, data bit width.

TABLE 10-2: SQI INSTRUCTIONS

Instruction	Description	Bit width Note 2	Inst. code	Addr. Bytes	Dummy Bytes	Data bytes	Max Freq.
Configuration							
RSTQIO	Reset SQI	4-0-0	FFh	0	0	0	80 MHz
Read							
FASTREAD	Read at higher speed	4-4-4	0Bh	2	3	4 to ∞	80 MHz
Write							
WRITE	Write	4-4-4	02h	2	0	4 to ∞	80 MHz

Note 2: The bit width format is: command bit width, address / dummy bit width, data bit width.

10.2.1 DEVICE INITIALIZATION

Until the device has been initialized to the point where the various configuration inputs are valid, the SPI/SQI interface does not respond to and is not affected by any external pin activity.

Once device initialization completes, the SPI/SQI interface will ignore the pins until a rising edge of **SCS#** is detected.

10.2.1.1 SPI/SQI Slave Read Polling for Initialization Complete

Before device initialization, the SPI/SQI interface will not return valid data. To determine when the SPI/SQI interface is functional, the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) can be polled to determine when the device is fully configured.

Note:	The Host should only use single register reads (one data cycle per **SCS#** low) while polling the BYTE_TEST register.

10.2.2 ACCESS DURING AND FOLLOWING POWER MANAGEMENT

During any power management mode other than D0, reads and writes are ignored and the SPI/SQI interface does not respond to and is not affected by any external pin activity.

Once the power management mode changes back to D0, the SPI/SQI interface will ignore the pins until a rising edge of **SCS#** is detected.

To determine when the SPI/SQI interface is functional, the Byte Order Test Register (BYTE_TEST) should be polled. Once the correct pattern is read, the interface can be considered functional. At this point, the Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) can be polled to determine when the device is fully configured.

Note:	The Host should only use single register reads (one data cycle per **SCS#** low) while polling the BYTE_TEST register.

10.2.3 SPI CONFIGURATION COMMANDS

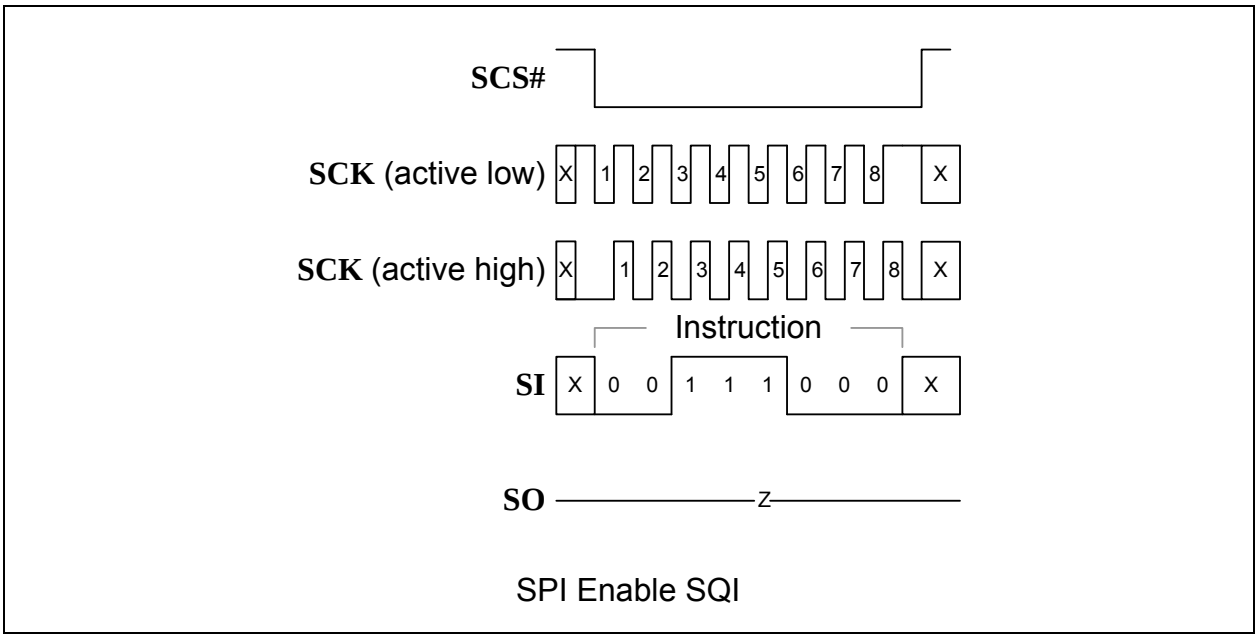
10.2.3.1 Enable SQI

The Enable SQI instruction changes the mode of operation to SQI. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit EQIO instruction, 38h, is input into the **SI/SIO[0]** pin one bit per clock. The **SCS#** input is brought inactive to conclude the cycle.

Figure 10-1 illustrates the Enable SQI instruction.

FIGURE 10-1: ENABLE SQI



10.2.3.2 Reset SQI

The Reset SQI instruction changes the mode of operation to SPI. This instruction is supported in SPI and SQI bus protocols with clock frequencies up to 80 MHz.

The SPI/SQI slave interface is selected by first bringing **SCS#** active. The 8-bit RSTQIO instruction, FFh, is input into the **SI/SIO[0]** pin, one bit per clock, in SPI mode and into the **SIO[3:0]** pins, four bits per clock, in SQI mode. The **SCS#** input is brought inactive to conclude the cycle.

Figure 10-2 illustrates the Reset SQI instruction for SPI mode. Figure 10-3 illustrates the Reset SQI instruction for SQI mode.

FIGURE 10-2: SPI MODE RESET SQI

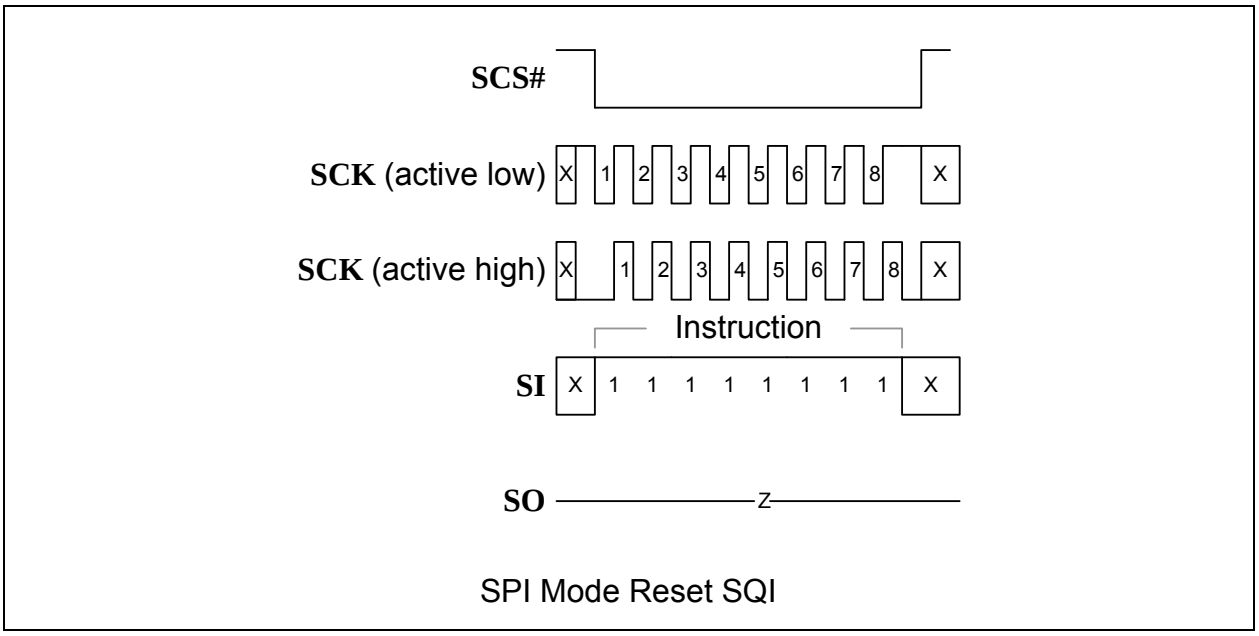
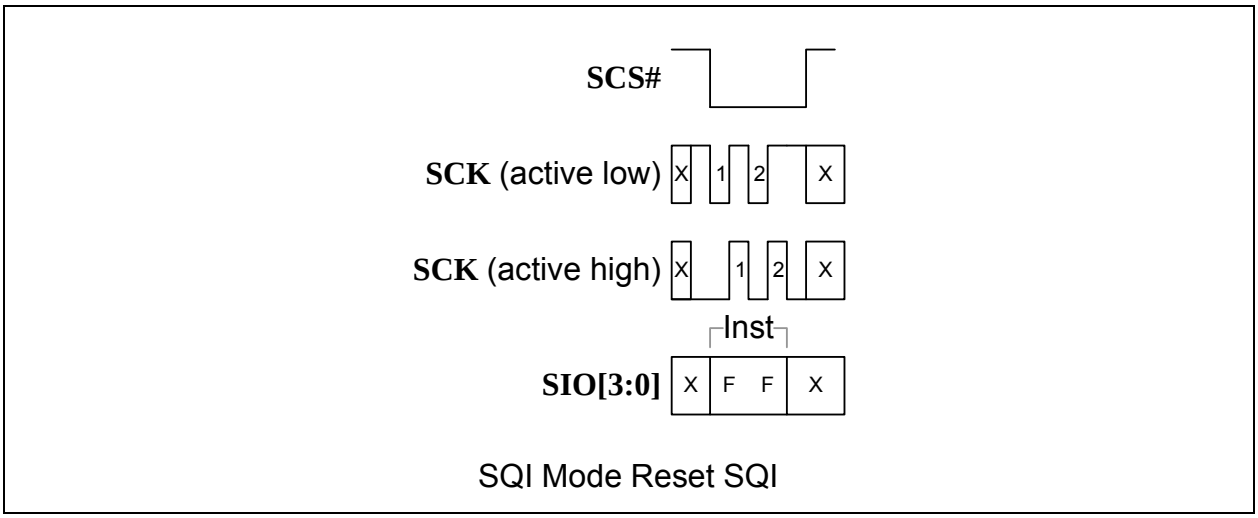


FIGURE 10-3: SQI MODE RESET SQI



10.2.4 SPI READ COMMANDS

Various read commands are support by the SPI/SQI slave. The following applies to all read commands.

MULTIPLE READS

Additional reads, beyond the first, are performed by continuing the clock pulses while **SCS#** is active. The upper two bits of the address specify auto-incrementing (address[15:14]=01b) or auto-decrementing (address[15:14]=10b). The internal DWORD address is incremented, decremented, or maintained based on these bits. Maintaining a fixed internal address is useful for register polling.

SPECIAL CSR HANDLING

Live Bits

Since data is read serially, the selected register's value is saved at the beginning of each 32-bit read to prevent the host from reading an intermediate value. The saving occurs multiple times in a multiple read sequence.

Change on Read Registers and FIFOs

Any register that is affected by a read operation (e.g. a clear on read bit or FIFO) is updated once the current data output shift has started. In the event that 32-bits are not read when the **SCS#** is returned high, the register is still affected and any prior data is lost.

Change on Read Live Register Bits

As described above, the current value from a register with live bits (as is the case of any register) is saved before the data is shifted out. Although a H/W event that occurs following the data capture would still update the live bit(s), the live bit(s) will be affected (cleared, etc.) once the output shift has started and the H/W event would be lost. In order to prevent this, the individual CSRs defer the H/W event update until after the read indication.

10.2.4.1 Read

The Read instruction inputs the instruction code and address bytes one bit per clock and outputs the data one bit per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 30 MHz. This instruction is not supported in SQI bus protocol.

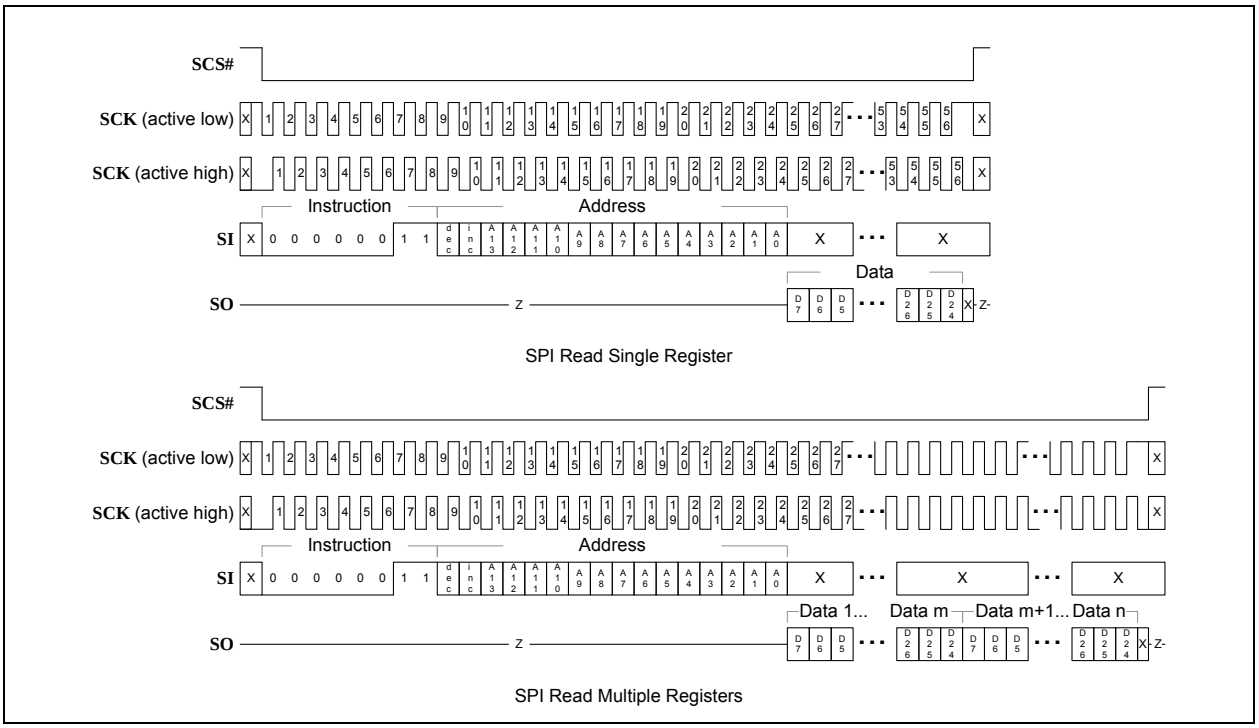
The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit READ instruction, 03h, is input into the **SI/SIO[0]** pin, followed by the two address bytes. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last address bit, the **SO/SIO[1]** pin is driven starting with the msb of the LSB of the selected register. The remaining register bits are shifted out on subsequent falling clock edges.

The **SCS#** input is brought inactive to conclude the cycle. The **SO/SIO[1]** pin is three-stated at this time.

Figure 10-4 illustrates a typical single and multiple register read.

FIGURE 10-4: SPI READ



10.2.4.2 Fast Read

The Read at higher speed instruction inputs the instruction code and the address and dummy bytes one bit per clock and outputs the data one bit per clock. In SQI mode, the instruction code and the address and dummy bytes are input four bits per clock and the data is output four bits per clock. This instruction is supported in SPI and SQI bus protocols with clock frequencies up to 80 MHz.

The SPI/SQI slave interface is selected by first bringing **SCS#** active. For SPI mode, the 8-bit FASTREAD instruction, 0Bh, is input into the **SI/SIO[0]** pin, followed by the two address bytes and 1 dummy byte. For SQI mode, the 8-bit FASTREAD instruction is input into the **SIO[3:0]** pins, followed by the two address bytes and 3 dummy bytes. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last dummy bit (or nibble), the **SO/SIO[1]** pin is driven starting with the msb of the LSB of the selected register. For SQI mode, **SIO[3:0]** are driven starting with the msn of the LSB of the selected register. The remaining register bits are shifted out on subsequent falling clock edges.

The **SCS#** input is brought inactive to conclude the cycle. The **SO/SIO[3:0]** pins are three-stated at this time.

Figure 10-5 illustrates a typical single and multiple register fast read for SPI mode. Figure 10-6 illustrates a typical single and multiple register fast read for SQI mode.

FIGURE 10-5: SPI FAST READ

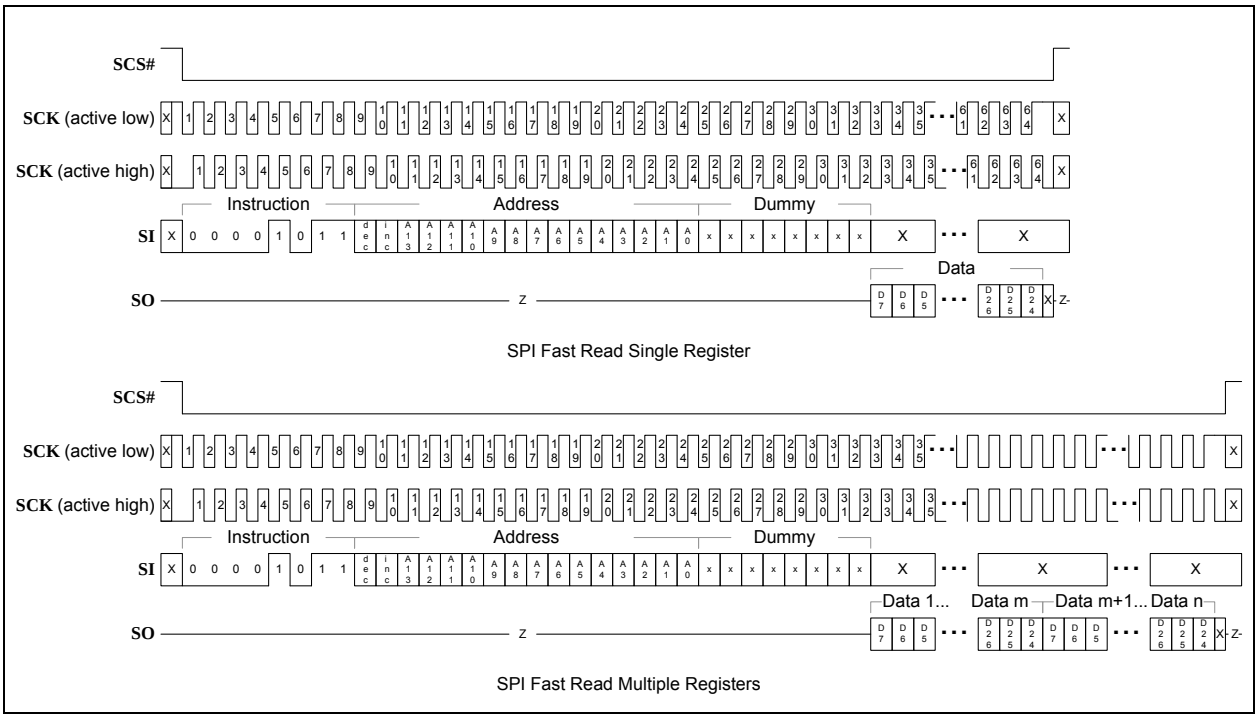
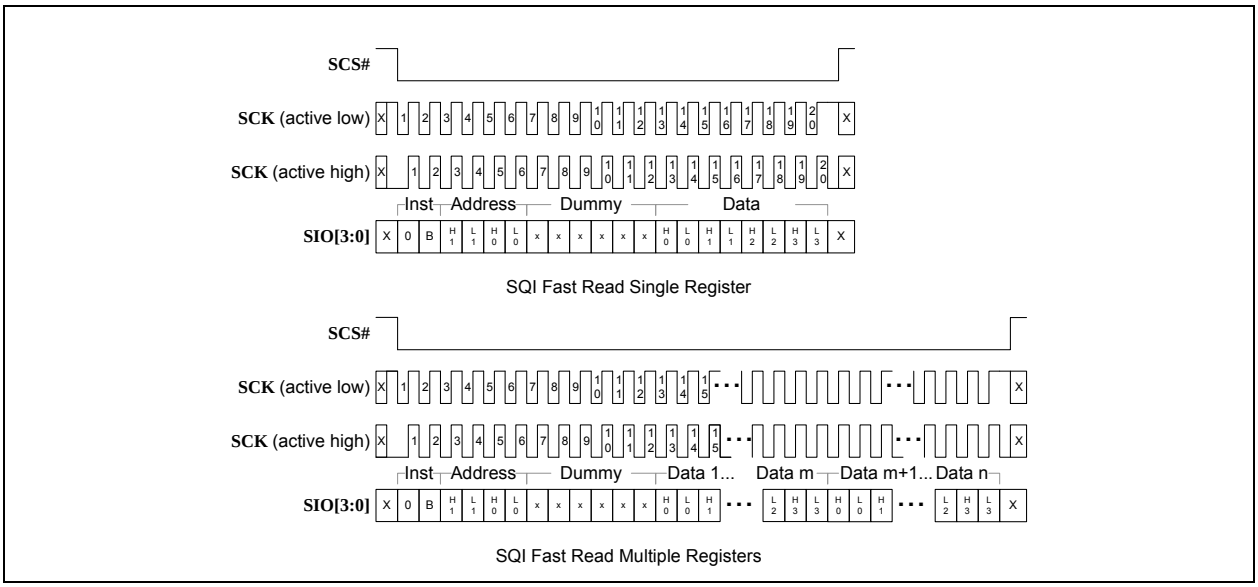


FIGURE 10-6: SQI FAST READ



10.2.4.3 Dual Output Read

The SPI Dual Output Read instruction inputs the instruction code and the address and dummy bytes one bit per clock and outputs the data two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

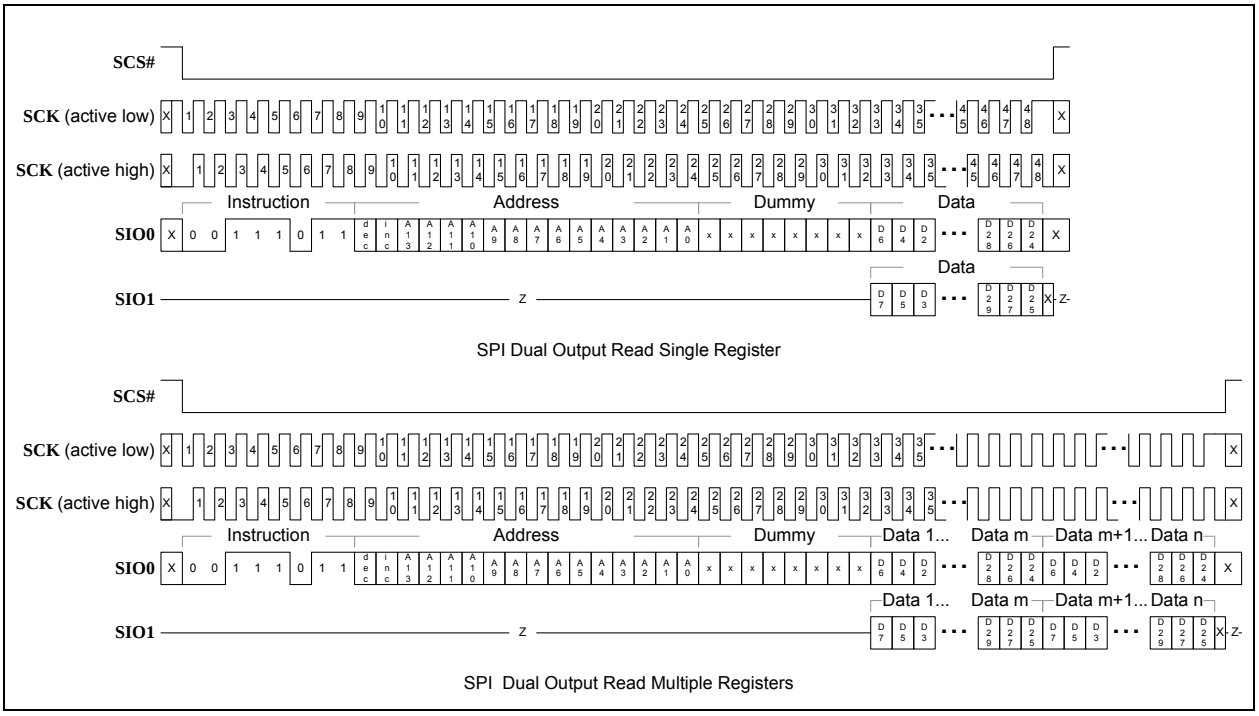
The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SDOR instruction, 3Bh, is input into the **SIO[0]** pin, followed by the two address bytes and 1 dummy byte. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last dummy di-bit, the **SIO[1:0]** pins are driven starting with the msbs of the LSB of the selected register. The remaining register di-bits are shifted out on subsequent falling clock edges.

The **SCS#** input is brought inactive to conclude the cycle. The **SIO[1:0]** pins are three-stated at this time.

Figure 10-7 illustrates a typical single and multiple register dual output read.

FIGURE 10-7: SPI DUAL OUTPUT READ



10.2.5 QUAD OUTPUT READ

The SPI Quad Output Read instruction inputs the instruction code and the address and dummy bytes one bit per clock and outputs the data four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

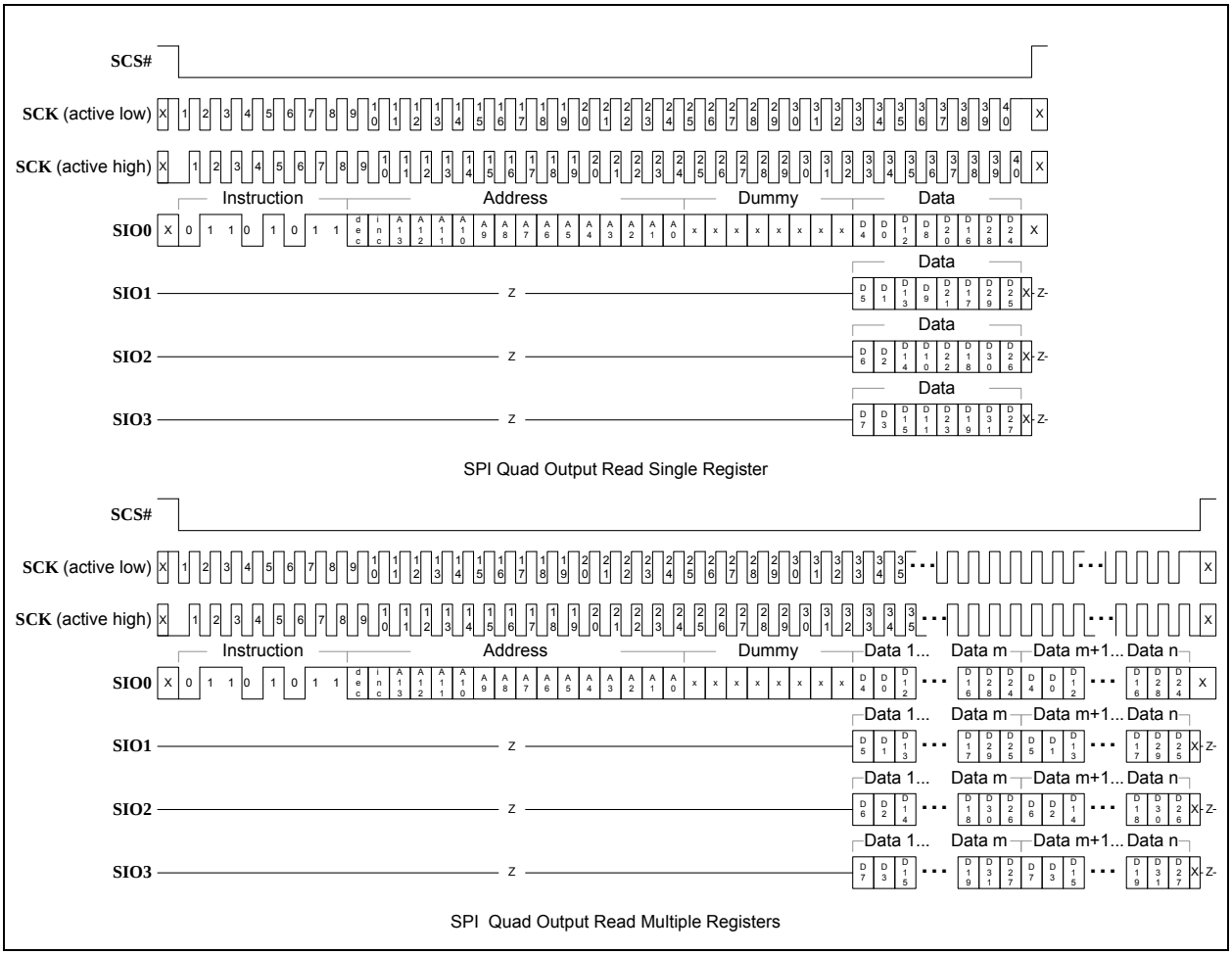
The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SQOR instruction, 6Bh, is input into the **SIO[0]** pin, followed by the two address bytes and 1 dummy byte. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last dummy bit, the **SIO[3:0]** pins are driven starting with the msn of the LSB of the selected register. The remaining register nibbles are shifted out.

The **SCS#** input is brought inactive to conclude the cycle. The **SIO[3:0]** pins are three-stated at this time.

Figure 10-8 illustrates a typical single and multiple register quad output read.

FIGURE 10-8: SPI QUAD OUTPUT READ



10.2.5.1 Dual I/O Read

The SPI Dual I/O Read instruction inputs the instruction code one bit per clock and the address and dummy bytes two bits per clock and outputs the data two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

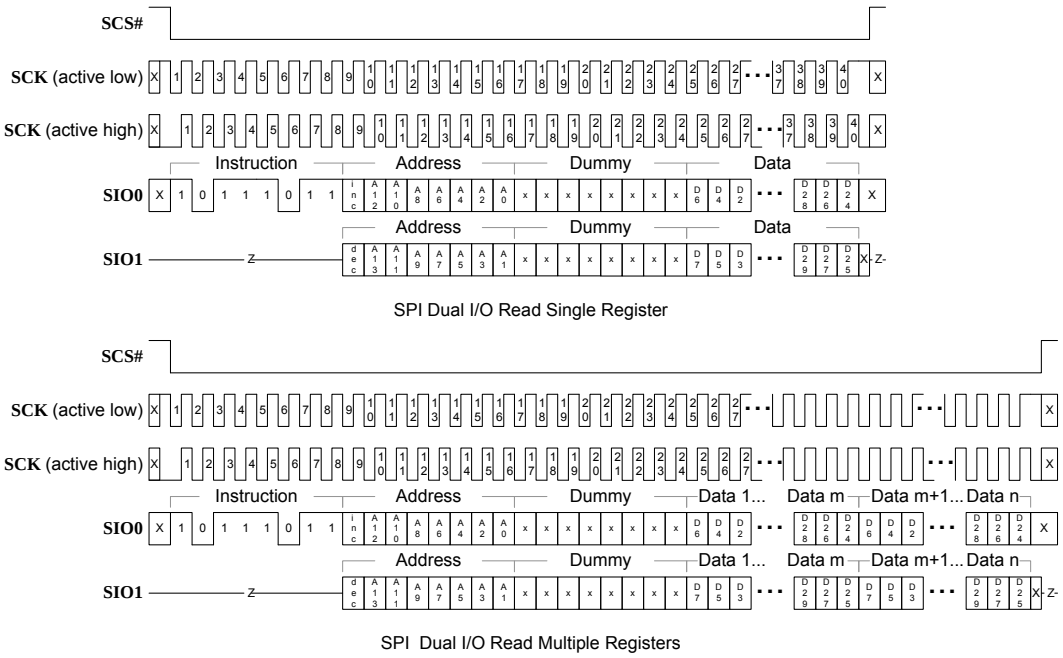
The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SDIOR instruction, BBh, is input into the **SIO[0]** pin, followed by the two address bytes and 2 dummy bytes into the **SIO[1:0]** pins. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last dummy di-bit, the **SIO[1:0]** pins are driven starting with the msbs of the LSB of the selected register. The remaining register di-bits are shifted out on subsequent falling clock edges.

The **SCS#** input is brought inactive to conclude the cycle. The **SIO[1:0]** pins are three-stated at this time.

Figure 10-9 illustrates a typical single and multiple register dual I/O read.

FIGURE 10-9: SPI DUAL I/O READ



10.2.5.2 Quad I/O Read

The SPI Quad I/O Read instruction inputs the instruction code one bit per clock and the address and dummy bytes four bits per clock and outputs the data four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

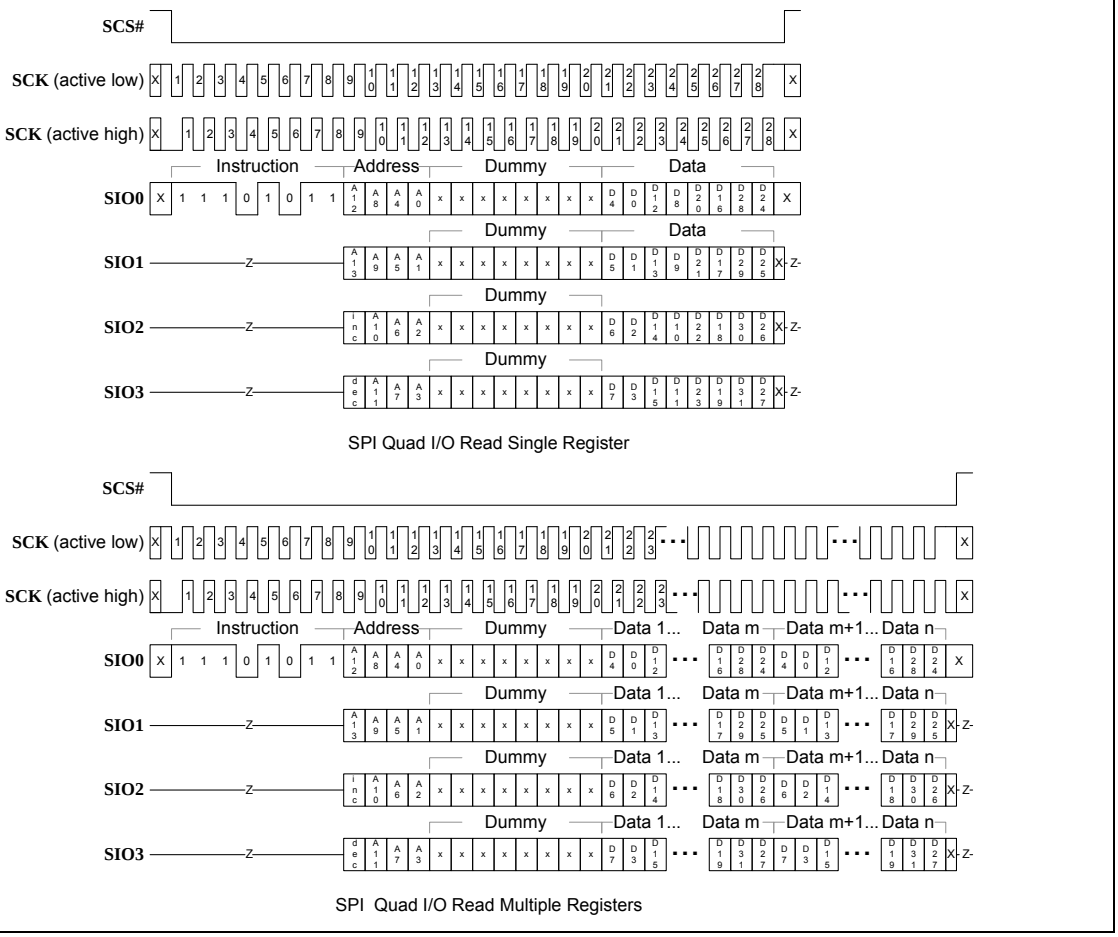
The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SQIOR instruction, EBh, is input into the **SIO[0]** pin, followed by the two address bytes and 4 dummy bytes into the **SIO[3:0]** pins. The address bytes specify a BYTE address within the device.

On the falling clock edge following the rising edge of the last dummy nibble, the **SIO[3:0]** pins are driven starting with the msn of the LSB of the selected register. The remaining register nibbles are shifted out on subsequent falling clock edges.

The **SCS#** input is brought inactive to conclude the cycle. The **SIO[3:0]** pins are three-stated at this time.

Figure 10-10 illustrates a typical single and multiple register quad I/O read.

FIGURE 10-10: SPI QUAD I/O READ



10.2.6 SPI WRITE COMMANDS

Multiple write commands are support by the SPI/SQI slave. The following applies to all write commands.

MULTIPLE WRITES

Multiple reads are performed by continuing the clock pulses and input data while **SCS#** is active. The upper two bits of the address specify auto-incrementing (address[15:14]=01b) or auto-decrementing (address[15:14]=10b). The internal DWORD address is incremented, decremented, or maintained based on these bits. Maintaining a fixed internal address may be useful for register “bit-banging” or other repeated writes.

10.2.6.1 Write

The Write instruction inputs the instruction code and address and data bytes one bit per clock. In SQI mode, the instruction code and the address and data bytes are input four bits per clock. This instruction is supported in SPI and SQI bus protocols with clock frequencies up to 80 MHz.

The SPI/SQI slave interface is selected by first bringing **SCS#** active. For SPI mode, the 8-bit WRITE instruction, 02h, is input into the **SI/SIO[0]** pin, followed by the two address bytes. For SQI mode, the 8-bit WRITE instruction, 02h, is input into the **SIO[3:0]** pins, followed by the two address bytes. The address bytes specify a BYTE address within the device.

The data follows the address bytes. For SPI mode, the data is input into the **SI/SIO[0]** pin starting with the msb of the LSB. For SQI mode the data is input nibble wide using **SIO[3:0]** starting with the msn of the LSB. The remaining bits/nibbles are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

The **SCS#** input is brought inactive to conclude the cycle.

Figure 10-11 illustrates a typical single and multiple register write for SPI mode. Figure 10-12 illustrates a typical single and multiple register write for SQI mode.

FIGURE 10-11: SPI WRITE

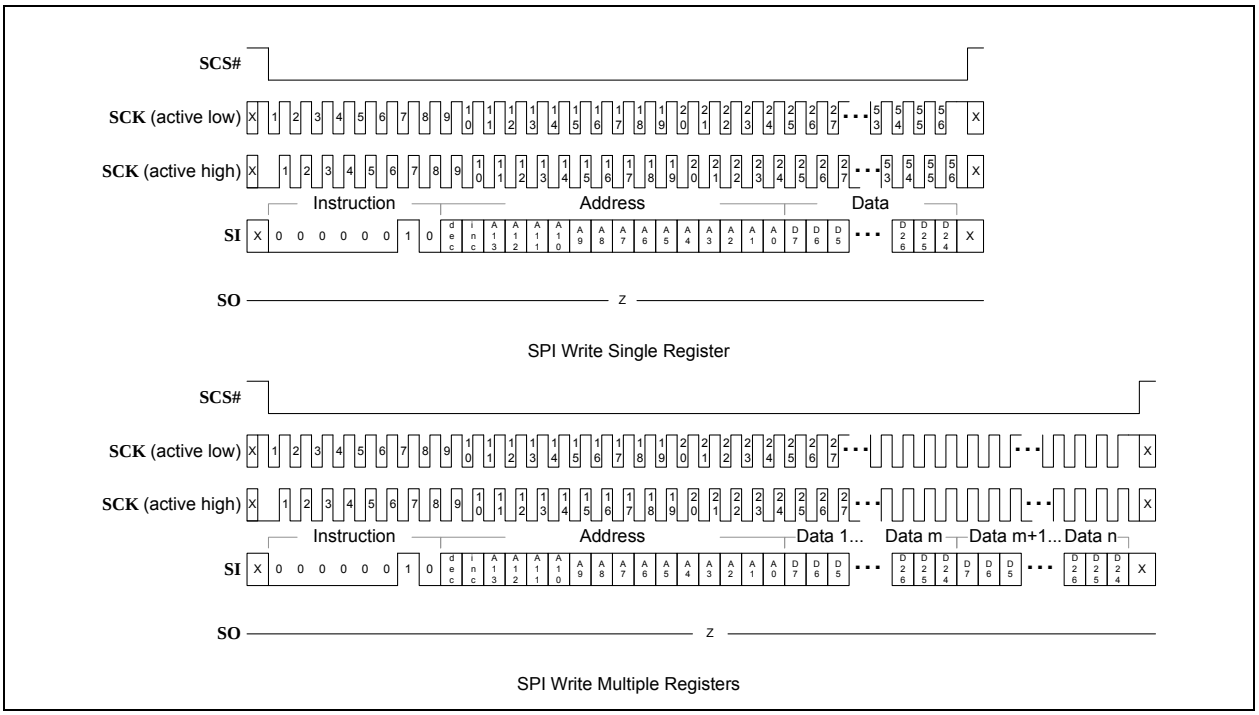
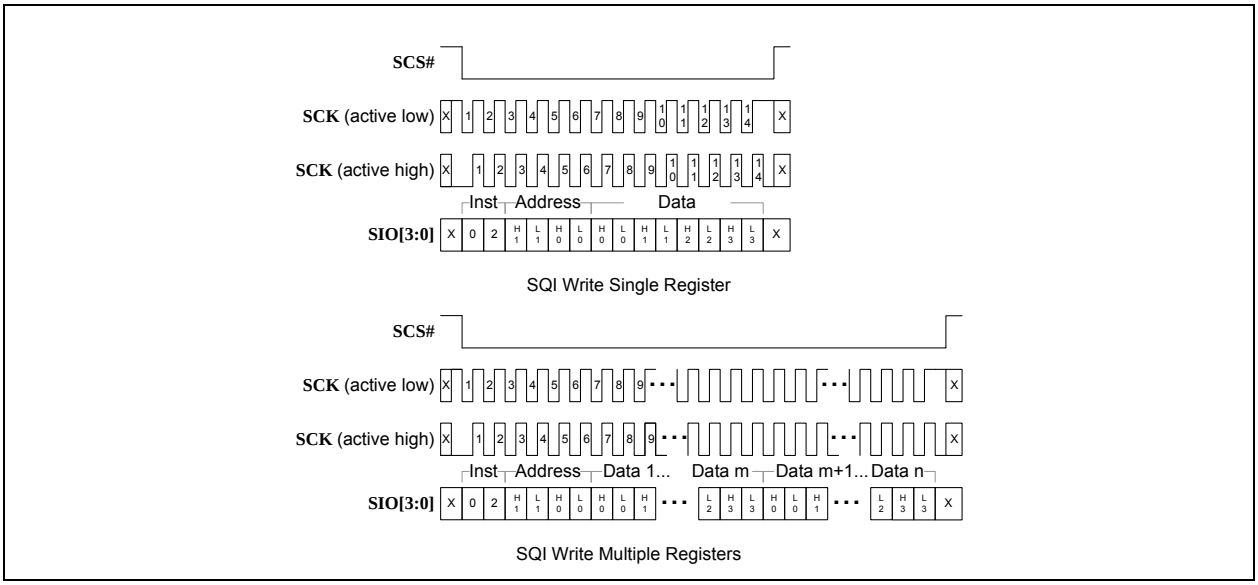


FIGURE 10-12: SQI WRITE



10.2.6.2 Dual Data Write

The SPI Dual Data Write instruction inputs the instruction code and address bytes one bit per clock and inputs the data two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

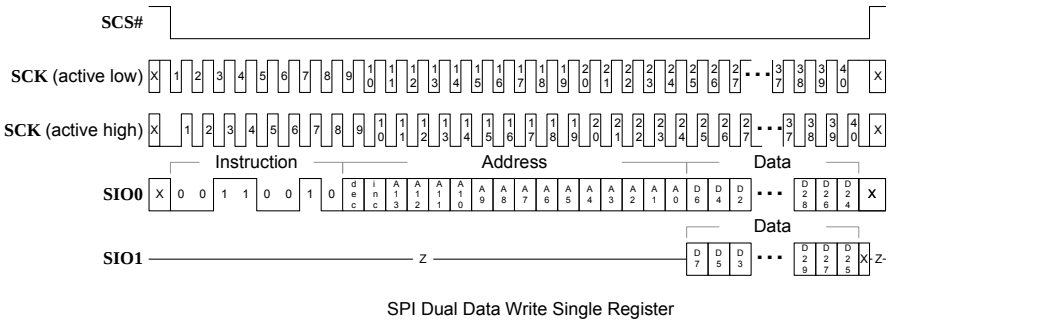
The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SDDW instruction, 32h, is input into the **SIO[0]** pin, followed by the two address bytes. The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the **SIO[1:0]** pins starting with the msbs of the LSB. The remaining di-bits are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

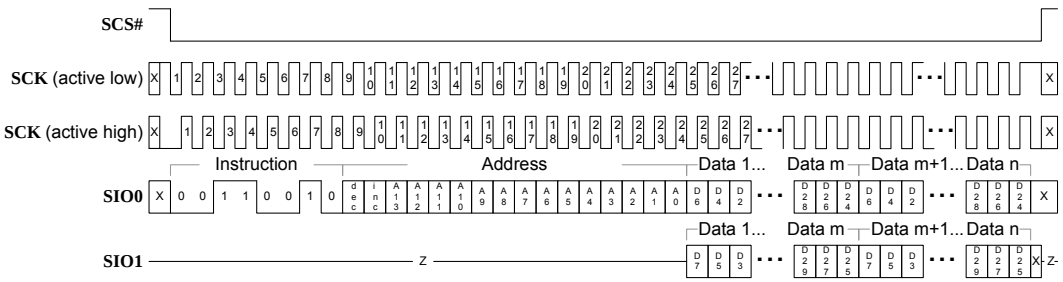
The **SCS#** input is brought inactive to conclude the cycle.

Figure 10-13 illustrates a typical single and multiple register dual data write.

FIGURE 10-13: SPI DUAL DATA WRITE



SPI Dual Data Write Single Register



SPI Dual Data Write Multiple Registers

10.2.6.3 Quad Data Write

The SPI Quad Data Write instruction inputs the instruction code and address bytes one bit per clock and inputs the data four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

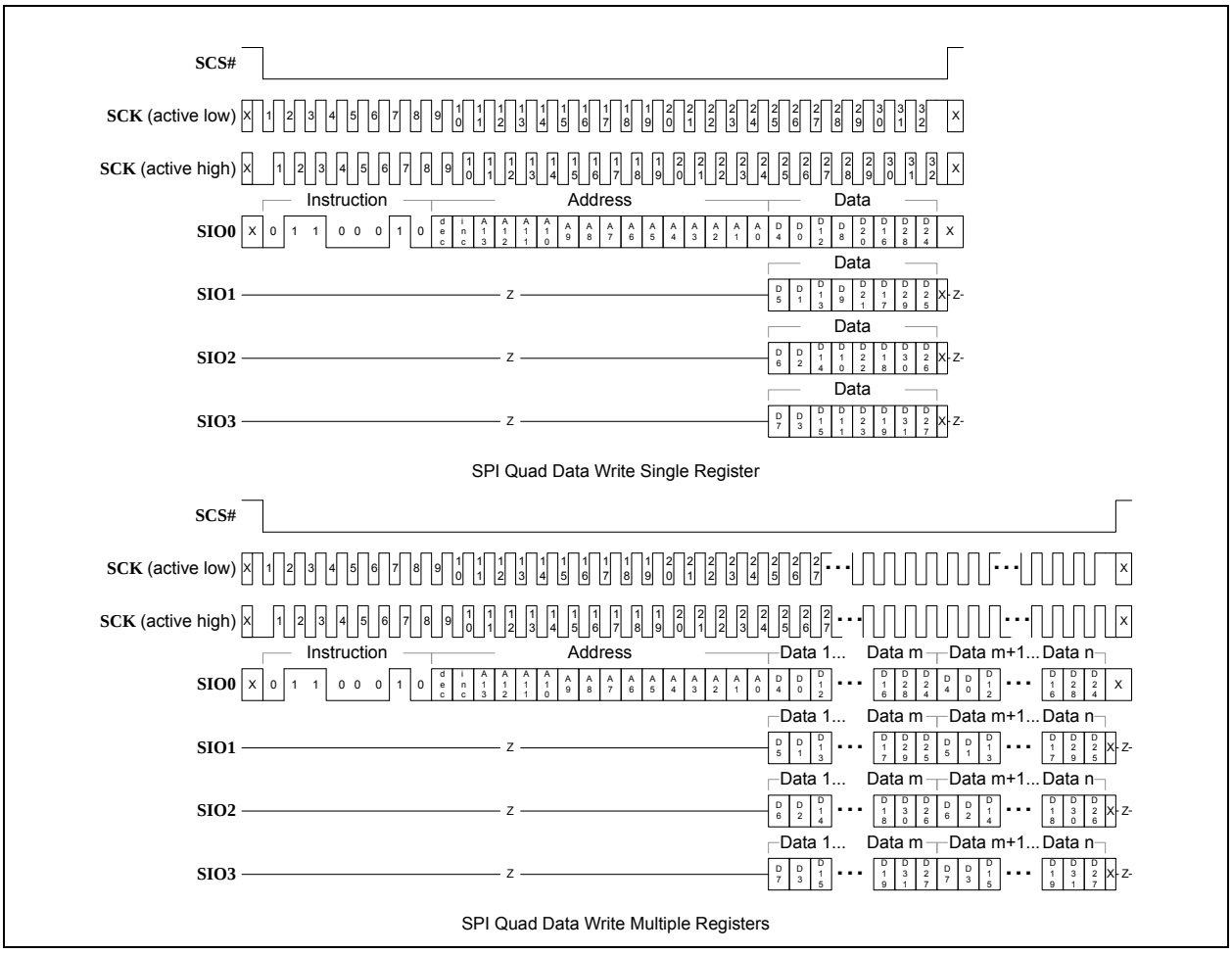
The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SQDW instruction, 62h, is input into the **SIO[0]** pin, followed by the two address bytes. The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the **SIO[3:0]** pins starting with the msn of the LSB. The remaining nibbles are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

The **SCS#** input is brought inactive to conclude the cycle.

Figure 10-14 illustrates a typical single and multiple register quad data write.

FIGURE 10-14: SPI QUAD DATA WRITE



10.2.6.4 Dual Address / Data Write

The SPI Dual Address / Data Write instruction inputs the instruction code one bit per clock and the address and data bytes two bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

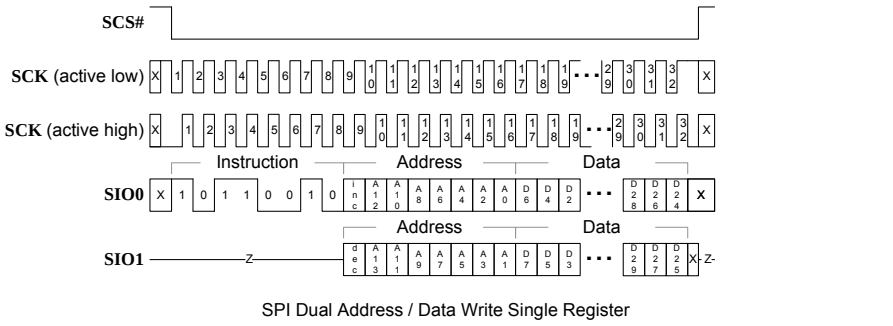
The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SDADW instruction, B2h, is input into the SIO[0] pin, followed by the two address bytes into the **SIO[1:0]** pins. The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the **SIO[1:0]** pins starting with the msbs of the LSB. The remaining di-bits are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

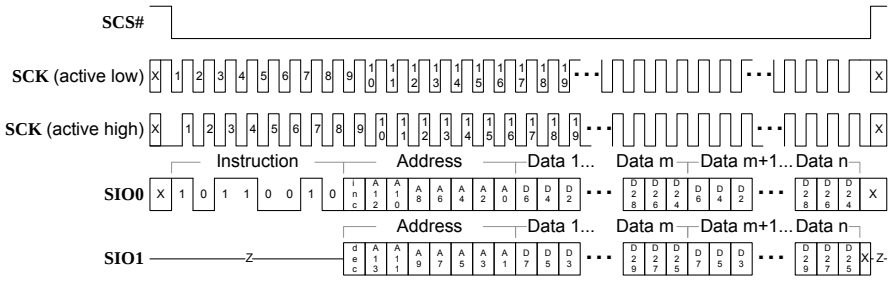
The **SCS#** input is brought inactive to conclude the cycle.

Figure 10-15 illustrates a typical single and multiple register dual address / data write.

FIGURE 10-15: SPI DUAL ADDRESS / DATA WRITE



SPI Dual Address / Data Write Single Register



SPI Dual Address / Data Write Multiple Registers

10.2.6.5 Quad Address / Data Write

The SPI Quad Address / Data Write instruction inputs the instruction code one bit per clock and the address and data bytes four bits per clock. This instruction is supported in SPI bus protocol only with clock frequencies up to 80 MHz. This instruction is not supported in SQI bus protocol.

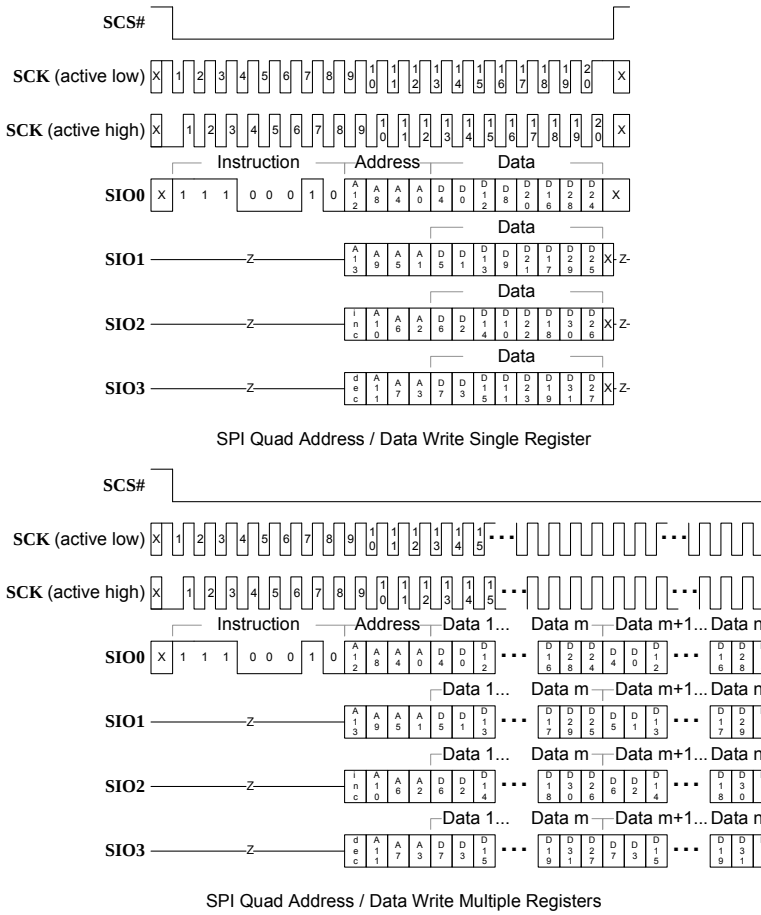
The SPI slave interface is selected by first bringing **SCS#** active. The 8-bit SQADW instruction, E2h, is input into the **SIO[0]** pin, followed by the two address bytes into the **SIO[3:0]** pins. The address bytes specify a BYTE address within the device.

The data follows the address bytes. The data is input into the **SIO[3:0]** pins starting with the msn of the LSB. The remaining nibbles are shifted in on subsequent clock edges. The data write to the register occurs after the 32-bits are input. In the event that 32-bits are not written when the **SCS#** is returned high, the write is considered invalid and the register is not affected.

The **SCS#** input is brought inactive to conclude the cycle.

Figure 10-16 illustrates a typical single and multiple register dual address / data write.

FIGURE 10-16: SPI QUAD ADDRESS / DATA WRITE



10.3 TX and RX FIFO Access

10.3.0.1 TX and RX Status FIFO Peek Address Access

Normal read access to the TX or RX Status FIFO causes the FIFO to advance to its next entry.

For access to the TX and RX Status FIFO Peek addresses, the FIFO does not advance to its next entry.

10.4 SPI/SQI Timing Requirements

FIGURE 10-17: SPI/SQI INPUT TIMING

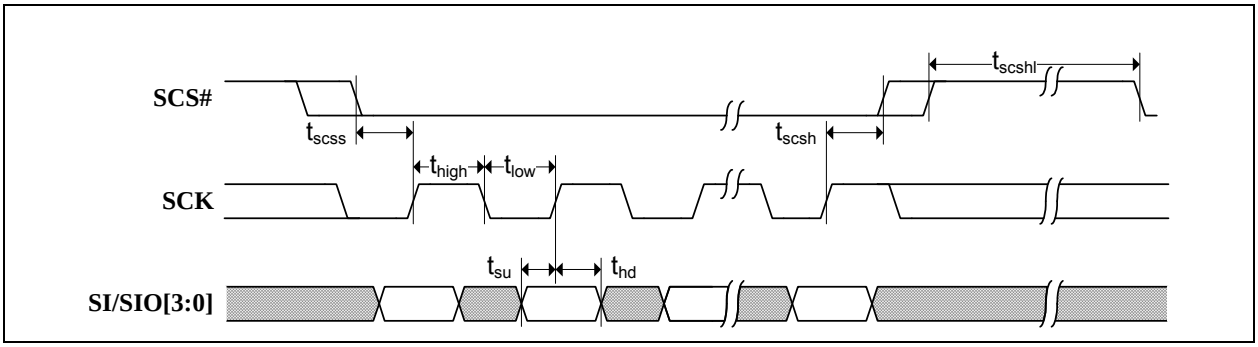


FIGURE 10-18: SPI/SQI OUTPUT TIMING

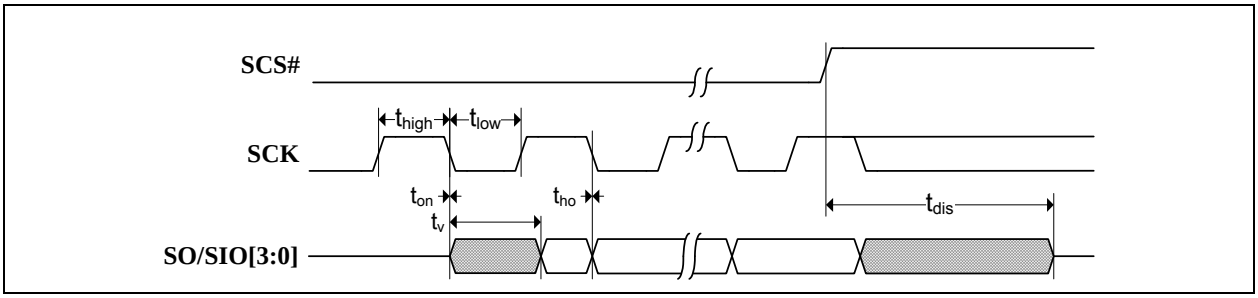


TABLE 10-3: SPI/SQI TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
f_{sck}	SCK clock frequency Note 3			30 / 80	MHz
t_{high}	SCK high time	5.5			ns
t_{low}	SCK low time	5.5			ns
t_{scss}	SCS# setup time to SCK	5			ns
t_{scsh}	SCS# hold time from SCK	5			ns
t_{scshl}	SCS# inactive time	50			ns
t_{su}	Data input setup time to SCK	3			ns
t_{hd}	Data input hold time from SCK	4			ns
t_{on}	Data output turn on time from SCK	0			ns
t_v	Data output valid time from SCK Note 4, Note 5			11.0/9.0	ns
t_{ho}	Data output hold time from SCK	0			ns
t_{dis}	Data output disable time from SCS# inactive			20	ns

Note 3: The Read instruction is limited to 30 MHz maximum

Note 4: Depends on loading of 30 pF or 10 pF

Note 5: Depending on the clock frequency and pulse width, data may not be valid until following the next rising edge of SCK. The host SPI controller may need to delay the sampling of the data by either a fixed time or by using the falling edge of SCK.

11.0 HOST MAC

11.1 Functional Overview

The Host MAC incorporates the essential protocol requirements for operating an Ethernet/IEEE 802.3-compliant node and provides an interface between the Host and the Ethernet PHY. On the front end, the Host MAC interfaces to the Host via 2 sets of FIFO's (TX Data FIFO, TX Status FIFO, RX Data FIFO, RX Status FIFO). An additional bus is used to access the Host MAC CSRs via the Host MAC CSR Interface Command Register (MAC_CSR_CMD) and Host MAC CSR Interface Data Register (MAC_CSR_DATA) system registers.

The receive and transmit FIFO's allow increased packet buffer storage to the Host MAC. The FIFOs are a conduit between the Host and the Host MAC through which all transmitted and received data and status information is passed. Deep FIFOs allow a high degree of latency tolerance relative to the various transport and OS software stacks reducing and minimizing overrun conditions. Both the Host MAC and the TX/RX FIFOs have separate receive and transmit data paths.

The Host MAC can store up to 250 Ethernet packets utilizing FIFOs, totaling 16KB, with a packet granularity of 4 bytes. This memory is shared by the RX and TX blocks and is configurable in terms of allocation via the Hardware Configuration Register (HW_CFG) register to the ranges described in Section 11.10.3, "FIFO Memory Allocation Configuration". This depth of buffer storage minimizes or eliminates receive overruns.

On the back end, the Host MAC interfaces with the 10/100 Ethernet PHY via an internal SMI (Serial Management Interface) bus. This allows the Host MAC access to the PHY's internal registers via the Host MAC MII Access Register (HMAC_MII_ACC) and Host MAC MII Data Register (HMAC_MII_DATA). The Host MAC interfaces to the PHY via an internal MII (Media Independent Interface) connection allowing for incoming and outgoing Ethernet packet transfers.

The Host MAC can operate at either 100Mbps or 10Mbps in both half-duplex or full-duplex modes. When operating in half-duplex mode, the Host MAC complies fully with Section 4 of ISO/IEC 8802-3 (ANSI/IEEE standard) and ANSI/IEEE 802.3 standards. When operating in full-duplex mode, the Host MAC complies with IEEE 802.3 full-duplex operation standard.

The Host MAC provides programmable enhanced features designed to minimize host supervision, bus utilization, and pre- or post-message processing. These features include the ability to disable retries after a collision, dynamic Frame Check Sequence (FCS) generation on a frame-by-frame basis, automatic pad field insertion and deletion to enforce minimum frame size attributes, and automatic retransmission and detection of collision frames. The Host MAC can sustain transmission or reception of minimally-sized back-to-back packets at full line speed with an interpacket gap (IPG) of 9.6 microseconds for 10 Mbps and 0.96 microseconds for 100 Mbps.

The primary attributes of the Host MAC are:

- Transmit and receive message data encapsulation
- Framing (frame boundary delimitation, frame synchronization)
- Error detection (physical medium transmission errors)
- Media access management
- Medium allocation (collision detection, except in full-duplex operation)
- Contention resolution (collision handling, except in full-duplex operation)
- Flow control during full-duplex mode
- Decoding of control frames (PAUSE command) and disabling the transmitter
- Generation of control frames
- Interface between the Host Bus Interface and the Ethernet PHY.

11.2 Flow Control

The Host MAC supports full-duplex flow control using the pause operation and control frame. Half-duplex flow control using back pressure is also supported. The Host MAC flow control is configured via the memory mapped Host MAC Automatic Flow Control Configuration Register (AFC_CFG) located in the System CSR space and the Host MAC Flow Control Register (HMAC_FLOW) located in the Host MAC CSR space.

11.2.1 FULL-DUPLEX FLOW CONTROL

The pause operation inhibits transmission of data frames for a specified period of time. A pause operation consists of a frame containing the globally assigned multicast address (01-80-C2-00-00-01), the PAUSE opcode, and a parameter indicating the quantum of slot time (512 bit times) to inhibit data transmissions. The PAUSE parameter may range from

0 to 65,535 slot times. The Host MAC logic, upon receiving a frame with the reserved multicast address and PAUSE opcode, inhibits data frame transmissions for the length of time indicated. If a Pause request is received while a transmission is in progress, then the pause will take effect after the transmission is complete. Control frames are received, processed by the Host MAC, and passed on.

The device will automatically transmit pause frames based on the settings of the Host MAC Automatic Flow Control Configuration Register (AFC_CFG) and the Host MAC Flow Control Register (HMAC_FLOW). When the RX FIFO reaches the level set in the Automatic Flow Control High Level (AFC_HI) field of Host MAC Automatic Flow Control Configuration Register (AFC_CFG), the device will transmit a pause frame. The pause time field that is transmitted is set in the Pause Time (FCPT) field of the Host MAC Flow Control Register (HMAC_FLOW) register. When the RX FIFO drops below the level set in the Automatic Flow Control Low Level (AFC_LO) field of Host MAC Automatic Flow Control Configuration Register (AFC_CFG), the device will automatically transmit a pause frame with a pause time of zero. The device will only send another pause frame when the RX FIFO level falls below Automatic Flow Control Low Level (AFC_LO) and then exceeds Automatic Flow Control High Level (AFC_HI) again.

11.2.2 HALF-DUPLEX FLOW CONTROL (BACKPRESSURE)

In half-duplex mode, backpressure is used for flow control. When the RX FIFO reaches the level set in the Automatic Flow Control High Level (AFC_HI) field of Host MAC Automatic Flow Control Configuration Register (AFC_CFG), the Host MAC will be enabled to collide with incoming frames. The Host MAC transmit logic enters a state at the end of current transmission (if any), where it waits for the beginning of a receive frame. Based on the settings in Host MAC Automatic Flow Control Configuration Register (AFC_CFG), backpressure can be enabled on any frame, a broadcast frame, any multicast frame or frames that match the stations address decoding logic.

In order to avoid any late collisions the Host MAC only generates collision-based backpressure at the start of a new frame. Once a new receive frame starts, the Host MAC intentional transmits which will result in a collision. Upon sensing the collision, the remote station will back off its transmission. Following the transmission of the intentional collision, the Host MAC waits for the next receive frame.

This pattern continues until either the RX FIFO drops below the level set in the Automatic Flow Control High Level (AFC_HI) or until the duration specified by Backpressure Duration (BACK_DUR) in field of Host MAC Automatic Flow Control Configuration Register (AFC_CFG) is reached. In either case, the Host MAC will allow one frame to be received before returning to backpressure operation. Note that the Backpressure Duration is timed from when the RX FIFO reaches the level set in Automatic Flow Control High Level (AFC_HI), regardless when or if actual backpressure occurs.

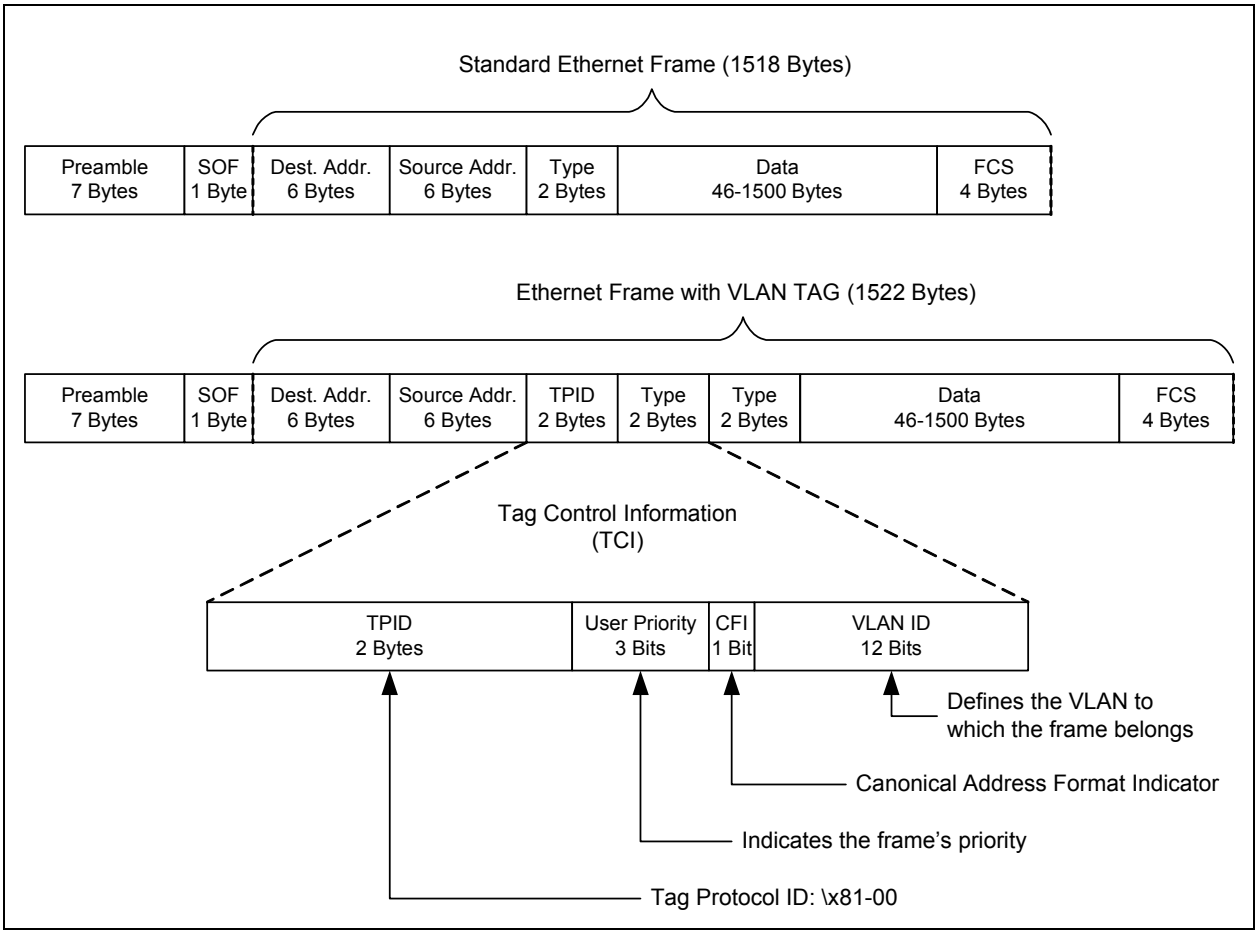
11.3 Virtual Local Area Network (VLAN) Support

Virtual Local Area Networks (VLANs), as defined within the IEEE 802.3 standard, provide network administrators a means of grouping nodes within a larger network into broadcast domains. To implement a VLAN, four extra bytes are added to the basic Ethernet packet. As shown in Figure 11-1, the four bytes are inserted after the Source Address Field and before the Type/Length field. The first two bytes of the VLAN tag identify the tag, and by convention are set to the value 0x8100. The last two bytes identify the specific VLAN associated with the packet and provide a priority field.

The device supports VLAN-tagged packets and provides two Host MAC registers, Host MAC VLAN1 Tag Register (HMAC_VLAN1) and Host MAC VLAN2 Tag Register (HMAC_VLAN2), which are used to identify VLAN-tagged packets. The HMAC_VLAN1 register is used to specify the VLAN1 tag which will increase the legal frame length from 1518 to 1522 bytes. The HMAC_VLAN2 register is used to specify the VLAN2 tag which will increase the legal frame length from 1518 to 1538 bytes. If a packet arrives bearing either of these tags in the two bytes succeeding the Source Address

field, the controller will recognize the packet as a VLAN-tagged packet, allowing the packet to be received and processed by the host software. If both VLAN1 and VLAN2 tag Identifiers are used, each should be unique. If both are set to the same value, VLAN1 is given higher precedence and the maximum legal frame length is set to 1522.

FIGURE 11-1: VLAN FRAME



11.4 Address Filtering

The Ethernet address fields of an Ethernet packet consist of two 6-byte fields: one for the destination address and one for the source address. The first bit of the destination address signifies whether it is a physical address or a multicast address.

The Host MAC address check logic filters the frame based on the Ethernet receive filter mode that has been enabled. The various filter modes of the Host MAC are specified based on the state of the control bits in the Host MAC Control Register (HMAC_CR), as shown in TABLE 11-1:. Please refer to the Section 11.15.1, "Host MAC Control Register (HMAC_CR)," on page 193 for more information on this register.

Frames that fail the address filtering are accepted only if the Receive All Mode (RXALL) bit in the Receive Configuration Register (RX_CFG) is set. The Filtering Fail bit in the RX Status will be set for these frames.

TABLE 11-1: ADDRESS FILTERING MODES

MCPAS	PRMS	INVFILT	HO	HPFILT	Description
0	0	0	0	0	**Perfect** - MAC address perfect filtering only for all addresses. Broadcast frames accepted if BCAST is low.
0	0	0	0	1	**Hash Perfect** - MAC address perfect filtering for physical address and hash filtering for multicast addresses. Broadcast frames accepted if BCAST is low.
0	0	0	1	1	**Hash Only** - Hash Filtering for physical and multicast addresses. Broadcast frames accepted if BCAST is low.
0	0	1	0	0	**Inverse Filtering**
X	1	0	0	X	**Promiscuous**
0			X	1	
1	0	0	0	X	**Perfect all Multicast** - Pass all multicast frames including broadcasts. Frames with physical addresses are perfect-filtered
1	0	0	1	1	**Hash Only all Multicast** - Pass all multicast frames including broadcasts. Frames with physical addresses are hash-filtered

11.4.1 PERFECT FILTERING

This filtering mode passes only incoming frames whose destination address field exactly matches the value programmed into the Host MAC Address High Register (HMAC_ADDRH) and the Host MAC Address Low Register (HMAC_ADDRL). The MAC address is formed by the concatenation of these two registers.

Broadcast frames are also accepted if Disable Broadcast Frames (BCAST) is low.

Note: If the HMAC_ADDRH and HMAC_ADDRL registers are set to the Broadcast address, Broadcast frames will be accepted regardless of the setting of the BCAST bit.

11.4.2 HASH ONLY FILTERING

This type of filtering checks for incoming receive packets with either multicast or physical destination addresses, and executes an imperfect address filtering against the hash table. The hash table is formed by merging the values in the Host MAC Multicast Hash Table High Register (HMAC_HASHH) and the Host MAC Multicast Hash Table Low Register (HMAC_HASHL) to form a 64-bit hash table.

During imperfect hash filtering, the destination address in the incoming frame is passed through the CRC logic and the upper 6-bits of the CRC register are used to index the contents of the hash table. The most significant bit determines the register to be used (HMAC_HASHH or HMAC_HASHL), while the other five bits determine the bit within the register. A value of 00000 selects Bit 0 of the HMAC_HASHL register and a value of 11111 selects Bit 31 of the HMAC_HASHH register.

Broadcast frames are also accepted if Disable Broadcast Frames (BCAST) is low.

Note:	If bit 31 of HMAC_HASHH is set, the Broadcast address will cause a hash match and Broadcast frames will be accepted regardless of the setting of the BCAST bit.

11.4.3 HASH PERFECT FILTERING

In hash perfect filtering, if the received frame is a physical address, the Host MAC packet filter will perfect-filter the incoming frame's destination field with the value programmed into the Host MAC Address High Register (HMAC_ADDRH) and the Host MAC Address Low Register (HMAC_ADDRL). However, if the incoming frame is a multicast frame, the Host MAC packet filter function performs an imperfect address filtering against the hash table.

The imperfect filtering against the hash table is the same imperfect filtering process described in Section 11.4.2, "Hash Only Filtering".

Broadcast frames are also accepted if Disable Broadcast Frames (BCAST) is low.

Note: If bit 31 of HMAC_HASHH is set, the Broadcast address will cause a hash match and Broadcast frames will be accepted regardless of the setting of the BCAST bit.

11.4.4 INVERSE FILTERING

In inverse filtering, the Host MAC packet filter accepts incoming frames with a destination address not matching the perfect address (i.e., the value programmed into the Host MAC Address High Register (HMAC_ADDRH) and the Host MAC Address Low Register (HMAC_ADDRL)) and rejects frames with destination addresses matching the perfect address.

Note: If the HMAC_ADDRH and HMAC_ADDRL registers are set to the Broadcast address, Broadcast frames will be filtered regardless of the setting of the BCAST bit.

11.4.5 PROMISCUOUS

When the Promiscuous Mode (PRMS) bit is set, all frames are accepted regardless of their destination address.

Note: Broadcast frames will be accepted regardless of the setting of the BCAST bit.

11.4.6 PERFECT FILTERING ALL MULTICAST

If the received frame is a physical address, the Host MAC packet filter will perfect-filter the incoming frame's destination field with the value programmed into the Host MAC Address High Register (HMAC_ADDRH) and the Host MAC Address Low Register (HMAC_ADDRL).

With the Pass All Multicast (MCPAS) bit of the Host MAC Control Register (HMAC_CR) set, all multicast frames are accepted. This includes all broadcast frames as well.

11.4.7 HASH ONLY FILTERING ALL MULTICAST

If the received frame is a physical address, the Host MAC packet filter will execute an imperfect address filtering against the hash table. The imperfect filtering against the hash table is the same imperfect filtering process described in Section 11.4.2, "Hash Only Filtering".

With the Pass All Multicast (MCPAS) bit of the Host MAC Control Register (HMAC_CR) set, all multicast frames are accepted. This includes all broadcast frames as well.

11.5 Frame Filtering

Following the address filtering, frames are accepted or rejected according to the following:

- Good frames that pass the address filtering are accepted. In the RX Status for these frames, the Filtering Fail bit will be clear (since the frame passed the address filter). The Packet Filter bit will be set.
- Good frames that fail the address filtering are accepted if the Receive All Mode (RXALL) bit in the Receive Configuration Register (RX_CFG) is set. In the RX Status for these frames, the Filtering Fail bit will be set (since the frame failed the address filter). The Packet Filter bit will also be set (since the RXALL bit allowed the acceptance of the frame).
- A good Broadcast frame is accepted if it passes the address filtering or if the RXALL bit is set. The Disable Broadcast Frames (BCAST) bit in the Receive Configuration Register (RX_CFG) determines if the Packet Filter bit in the RX Status is set (BCAST=0) or cleared (BCAST=1). Note that Disable Broadcast Frames (BCAST) doesn't cause the frame to be dropped if it passes the address filtering or if the RXALL bit is set. The Filtering Fail bit will indicate if the address filtering passed or failed.
- A good Control frame is accepted if it passes the address filtering or if the RXALL bit is set. The Pass Control Frames (FCPASS) bit in the Host MAC Flow Control Register (HMAC_FLOW) determines if the Packet Filter bit in the RX Status is set (FCPASS=1) or cleared (FCPASS=0). Note that Pass Control Frames (FCPASS) being low doesn't cause the frame to be dropped if it passes the address filtering or if the RXALL bit is set. The Filtering Fail bit will indicate if the address filtering passed or failed.
- A frame that has an error (runt, collision, CRC, too long) and is greater than 60 bytes in length is accepted if it passes the address filtering or if the RXALL bit is set. The Pass Bad Frames (PASSBAD) bit in the Receive Configuration Register (RX_CFG) determines if the Packet Filter bit in the RX Status is set (PASSBAD=1) or cleared (PASSBAD=0). The Filtering Fail bit will indicate if the address filtering passed or failed.
- A frame that has an error (runt, collision, CRC) and is 60 bytes or under in length is accepted if it passes the address filtering or if the RXALL bit is set and the Pass Bad Frames (PASSBAD) bit is set. The Packet Filter bit in the RX Status is set for these frames. The Filtering Fail bit will indicate if the address filtering passed or failed.

11.6 Wake-On-LAN (WOL)

The following bits of the Host MAC Wake-up Control and Status Register (HMAC_WUCSR), when enabled, may allow a WOL event to be asserted:

- Perfect DA Wakeup Enable (PFDA_EN)
- Broadcast Wakeup Enable (BCST_EN)
- Wake-Up Frame Enable (WUEN)
- Magic Packet Enable (MPEN)

The WoL Wait for Sleep (WOL_WAIT_SLEEP) bit in Host MAC Wake-up Control and Status Register (HMAC_WUCSR) will delay the WoL functions until the host has put the device into a power down mode.

WOL events may be indicated to the power management block via the Wake On Status (WOL_STS) bit of the Power Management Control Register (PMT_CTRL). Each WOL event type is detailed in the following sub-sections.

The WoL feature is part of the broader power management features of the device and can be used to trigger the power management event output pin (**PME**) or general interrupt request pin (**IRQ**). This is accomplished by enabling the desired WoL feature and setting the Wake-On-Enable (WOL_EN) bit of the Power Management Control Register (PMT_CTRL). Refer to Section 6.3, "Power Management," on page 44 for additional information.

11.6.1 PERFECT DA DETECTION

Setting the Perfect DA Wakeup Enable (PFDA_EN) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) places the MAC in the Perfect DA detection mode. In this mode, normal data reception is disabled, and detection logic within the MAC examines the destination address of each received frame.

When a frame whose destination address matches that specified by the Host MAC Address High Register (HMAC_ADDRH) and Host MAC Address Low Register (HMAC_ADDRL) is received, the Perfect DA Frame Received (PFDA_FR) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) is set. When the host clears the PFDA_EN bit, the Host MAC will resume normal receive operation.

11.6.2 BROADCAST DETECTION

Setting the Broadcast Wakeup Enable (BCST_EN) bit in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) places the MAC in the Broadcast detection mode. In this mode, normal data reception is disabled, and detection logic within the MAC examines the destination address of each received frame.

When a frame whose destination address is FF FF FF FF FF FF is received, the Broadcast Frame Received (BCAST_FR) bit in the WUCSR is set. When the host clears the BCST_EN bit, the Host MAC will resume normal receive operation.

11.6.3 WAKE-UP FRAME DETECTION

Eight programmable wakeup frame filters are supported. Each filter has a 128-bit byte mask that indicates which bytes of the frame should be compared by the MAC. A CRC-16 is calculated over these bytes. The result is then compared with the filter's respective CRC-16 to determine if a match exists.

Setting the Wake-Up Frame Enable (WUEN) in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR), places the Host MAC in the wake-up frame detection mode. In this mode, normal data reception is disabled, and detection logic within the Host MAC examines received data for the pre-programmed wake-up frame patterns.

Upon detection, the Remote Wake-Up Frame Received (WUFR) in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) register is set. When the host clears the WUEN bit, the Host MAC will resume normal receive operation.

Before putting the Host MAC into the wake-up frame detection state, the host must provide the detection logic with a list of sample frames and their corresponding byte masks. This information must be written into the Host MAC Wake-up Frame Filter Register (HMAC_WUFF). The wake-up frame filter is configured through this register using an index mechanism. After power-on reset, hardware reset, or soft reset, the Host MAC loads the first value written to the HMAC_WUFF register to the first DWORD in the wake-up frame filter (filter 0 byte mask 0). The second value written to this register is loaded to the second DWORD in the wake-up frame filter (filter 0 byte mask 1) and so on for all 40 DWORDs. The wake-up frame filter functionally is described below.

The Host MAC supports eight programmable 128-bit wake-up filters that support many different receive packet patterns. If remote wake-up mode is enabled, the remote wake-up function receives all frames addressed to the Host MAC. It then checks each frame against the enabled filter and recognizes the frame as a remote wake-up frame if it passes the wakeup frame filter register's address filtering and CRC value match.

In order to determine which bytes of the frames should be checked by the CRC module, the Host MAC uses a programmable byte mask and a programmable pattern offset for each of the eight supported filters.

The pattern's offset defines the location of the first byte that should be checked in the frame. Since the destination address is checked by the address filtering function, the pattern offset is always greater than 12.

The byte mask is a 128-bit field that specifies whether or not each of the 128 contiguous bytes within the frame, beginning in the pattern offset, should be checked. If bit j in the byte mask is set, the detection logic checks byte offset +j in the frame. In order to load the wake-up frame filter, the host must perform 40 writes to the Host MAC Wake-up Frame Filter Register (HMAC_WUFF). The contents of the Host MAC Wake-up Frame Filter Register (HMAC_WUFF) may be obtained by reading all 40 DWORDs. Table 11-2 shows the wake-up frame filter register's structure.

At the completion of the CRC-16 checking process, the CRC-16 calculated using the pattern offset and byte mask is compared to the expected CRC-16 value associated with the filter. If a match occurs, a remote wakeup event is signaled.

TABLE 11-2: WAKEUP FRAME FILTER REGISTER STRUCTURE

Filter 0 Byte Mask 0
Filter 0 Byte Mask 1
Filter 0 Byte Mask 2
Filter 0 Byte Mask 3
Filter 1 Byte Mask 0
Filter 1 Byte Mask 1

TABLE 11-2: WAKEUP FRAME FILTER REGISTER STRUCTURE (CONTINUED)

Filter 1 Byte Mask 2							
Filter 1 Byte Mask 3							
Filter 2 Byte Mask 0							
Filter 2 Byte Mask 1							
Filter 2 Byte Mask 2							
Filter 2 Byte Mask 3							
Filter 3 Byte Mask 0							
Filter 3 Byte Mask 1							
Filter 3 Byte Mask 2							
Filter 3 Byte Mask 3							
Filter 4 Byte Mask 0							
Filter 4 Byte Mask 1							
Filter 4 Byte Mask 2							
Filter 4 Byte Mask 3							
Filter 5 Byte Mask 0							
Filter 5 Byte Mask 1							
Filter 5 Byte Mask 2							
Filter 5 Byte Mask 3							
Filter 6 Byte Mask 0							
Filter 6 Byte Mask 1							
Filter 6 Byte Mask 2							
Filter 6 Byte Mask 3							
Filter 7 Byte Mask 0							
Filter 7 Byte Mask 1							
Filter 7 Byte Mask 2							
Filter 7 Byte Mask 3							
Reserved	Filter 3 Command	Reserved	Filter 2 Command	Reserved	Filter 1 Command	Reserved	Filter 0 Command
Reserved	Filter 7 Command	Reserved	Filter 6 Command	Reserved	Filter 5 Command	Reserved	Filter 4 Command
Filter 3 Offset		Filter 2 Offset		Filter 1Offset		Filter 0 Offset	
Filter 7 Offset		Filter 6 Offset		Filter 5 Offset		Filter 4 Offset	
Filter 1 CRC-16				Filter 0 CRC-16			
Filter 3 CRC-16				Filter 2 CRC-16			

TABLE 11-2: WAKEUP FRAME FILTER REGISTER STRUCTURE (CONTINUED)

Filter 5 CRC-16	Filter 4 CRC-16
Filter 7 CRC-16	Filter 6 CRC-16

The Filter i Byte Mask defines which incoming frame bytes Filter i will examine to determine whether or not this is a Wakeup Frame. Table 11-3, describes the byte mask's bit fields.

Filter x Mask 0 corresponds to bits [31:0]. Where the lsb corresponds to the first byte on the wire.

Filter x Mask 1 corresponds to bits [63:32]. Where the lsb corresponds to the first byte on the wire.

Filter x Mask 2 corresponds to bits [95:64]. Where the lsb corresponds to the first byte on the wire.

Filter x Mask 3 corresponds to bits [127:96]. Where the lsb corresponds to the first byte on the wire.

The following tables define elements common to both WUFF register structures.

TABLE 11-3: FILTER I BYTE MASK BIT DEFINITIONS

Filter i Byte Mask Description	
Bits	**Description**
127:0	**Byte Mask:** If bit j of the byte mask is set, the CRC machine processes byte *pattern-offset + j* of the incoming frame. Otherwise, byte *pattern-offset + j* is ignored.

The Filter i command register controls Filter i operation. Table 11-4 shows the Filter I command register.

TABLE 11-4: FILTER I COMMAND BIT DEFINITIONS

Filter i Commands	
Bits	**Description**
3:2	**Address Type:** Defines the destination address type of the pattern. 00 = Pattern applies only to unicast frames. 10 = Pattern applies only to multicast frames. X1 = Pattern applies to all frames that have passed the regular receive filter.
1	**RESERVED**
0	**Enable Filter:** When bit is set, Filter i is enabled, otherwise, Filter i is disabled.

The Filter i Offset register defines the offset in the frame's destination address field from which the frames are examined by Filter i. Table 11-5 describes the Filter i Offset bit fields.

TABLE 11-5: FILTER I OFFSET BIT DEFINITIONS

Filter i Offset Description	
Bits	**Description**
7:0	**Pattern Offset:** The offset of the first byte in the frame on which CRC is checked for Wakeup Frame recognition. The MAC checks the first offset byte of the frame for CRC and checks to determine whether the frame is a Wakeup Frame. Offset 0 is the first byte of the incoming frame's destination address.

The Filter i CRC-16 register contains the CRC-16 result of the frame that should pass Filter i.

TABLE 11-6: describes the Filter i CRC-16 bit fields.

The CRC-16 is calculated as follows:

At the start of a frame, CRC-16 is initialized with the value FFFFh. CRC-16 is updated when the pattern offset and mask indicate the received byte is part of the checksum calculation. The following algorithm is used to update the CRC-16 at that time:

Let:

^ denote the exclusive or operator.

Data [7:0] be the received data byte to be included in the checksum.

CRC[15:0] contain the calculated CRC-16 checksum.

F0 … F7 be intermediate results, calculated when a data byte is determined to be part of the CRC-16.

Calculate:

F0 = CRC[15] ^ Data[0]

F1 = CRC[14] ^ F0 ^ Data[1]

F2 = CRC[13] ^ F1 ^ Data[2]

F3 = CRC[12] ^ F2 ^ Data[3]

F4 = CRC[11] ^ F3 ^ Data[4]

F5 = CRC[10] ^ F4 ^ Data[5]

F6 = CRC[09] ^ F5 ^ Data[6]

F7 = CRC[08] ^ F6 ^ Data[7]

The CRC-16 is updated as follows:

CRC[15] = CRC[7] ^ F7

CRC[14] = CRC[6]

CRC[13] = CRC[5]

CRC[12] = CRC[4]

CRC[11] = CRC[3]

CRC[10] = CRC[2]

CRC[9] = CRC[1] ^ F0

CRC[8] = CRC[0] ^ F1

CRC[7] = F0 ^ F2

CRC[6] = F1 ^ F3

CRC[5] = F2 ^ F4

CRC[4] = F3 ^ F5

CRC[3] = F4 ^ F6

CRC[2] = F5 ^ F7

CRC[1] = F6

CRC[0] = F7

TABLE 11-6: FILTER I CRC-16 BIT DEFINITIONS

Filter i CRC-16 Description	
Bits	**Description**
15:0	**Pattern CRC-16:** This field contains the 16-bit CRC value from the pattern and the byte mask programmed to the Wakeup Filter register function. This value is compared against the CRC calculated on the incoming frame, and a match indicates the reception of a Wakeup Frame.

Table 11-7 indicates the cases that produce a wake when the Wake-Up Frame Enable (WUEN) bit of the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) is set. All other cases do not generate a wake.

TABLE 11-7: WAKEUP GENERATION CASES

Filter Enabled (Note 1)	Frame Type	CRC Match (Note 2)	Global Unicast Enabled (Note 3)	Pass Regular Receive Filter	Address Type (Note 4)
Yes	Unicast	Yes	Yes	x	x
Yes	Unicast	Yes	x	Yes	Unicast (=00)
Yes	Multicast	Yes	x	Yes	Multicast (=10)
Yes	Broadcast (Note 5)	Yes	x	x	x
Yes	x	Yes	x	Yes	Passed Receive Filter (=x1b)

Note 1: As determined by bit 0 of Filter i Command.

Note 2: CRC matches Filter i CRC-16 field.

Note 3: As determined by bit 9 of WUCSR.

Note 4: As determined by bits 3:2 of Filter i Command.

Note 5: When wake-up frame detection is enabled via the Wake-Up Frame Enable (WUEN) bit of the Host MAC Wake-up Control and Status Register (HMAC_WUCSR), a broadcast wake-up frame will wake-up the device despite the state of the Disable Broadcast Frames (BCAST) bit in the Host MAC Control Register (HMAC_CR).

Note: x indicates “don’t care”.

11.6.4 MAGIC PACKET DETECTION

Setting the Magic Packet Enable bit (MPEN) in the Host MAC Wake-up Control and Status Register (HMAC_WUCSR) places the Host MAC in the “Magic Packet” detection mode. In this mode, normal data reception is disabled, and detection logic within the Host MAC examines received data for a Magic Packet.

Upon detection, the Magic Packet Received bit (MPR) in the HMAC_WUCSR register is set. When the host clears the MPEN bit, the Host MAC will resume normal receive operation.

In Magic Packet mode, the Host MAC constantly monitors each frame addressed to the node for a specific Magic Packet pattern. Only packets passing the Address Filtering check of Section 11.4 (see Note 6) are checked for the Magic Packet requirements. Once the address requirement has been met, the Host MAC checks the received frame for the pattern 48’hFF_FF_FF_FF_FF_FF after the destination and source address field. The Host MAC then looks in the frame for 16 repetitions of the Host MAC address without any breaks or interruptions. In case of a break in the 16 address repetitions, the Host MAC again scans for the 48'hFF_FF_FF_FF_FF_FF pattern in the incoming frame. The 16 repetitions may be anywhere in the frame but must be preceded by the synchronization stream. The device will also accept a multicast frame, as long as it detects the 16 duplications of the Host MAC address.

For example, if the Host MAC address is 00h 11h 22h 33h 44h 55h, then the MAC scans for the following data sequence in an Ethernet frame:

Destination Address Source Address ……FF FF FF FF FF FF

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55

…CRC

Note 6: Normally, for Magic Packet Detection, address filtering should be set for Perfect Filtering or Hash Perfect Filtering.

11.7 Receive Checksum Offload Engine (RXCOE)

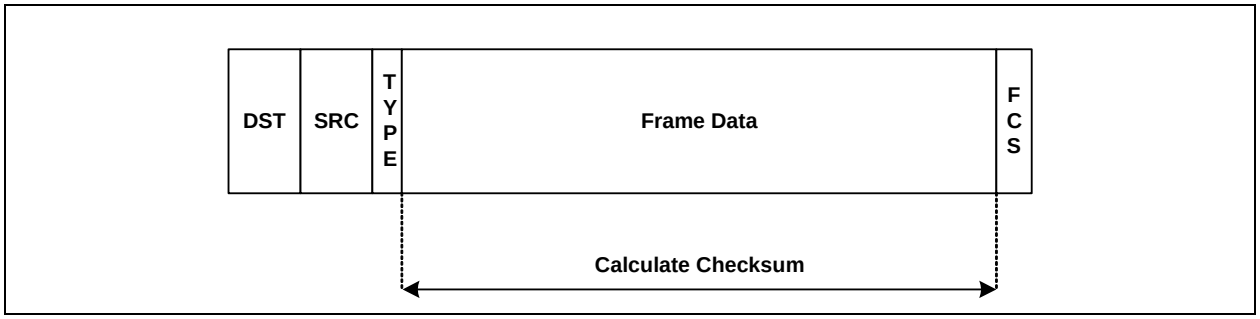
The receive checksum offload engine provides assistance to the Host by calculating a 16-bit checksum for a received Ethernet frame. The RXCOE readily supports the following IEEE802.3 frame formats:

- Type II Ethernet frames
- SNAP encapsulated frames
- Support for up to 2, 802.1q VLAN tags

The resulting checksum value can also be modified by software to support other frame formats.

The RXCOE has two modes of operation. In mode 0, the RXCOE calculates the checksum between the first 14 bytes of the Ethernet frame and the FCS. This is illustrated in Figure 11-2.

FIGURE 11-2: RXCOE CHECKSUM CALCULATION



In mode 1, the RXCOE supports VLAN tags and a SNAP header. In this mode, the RXCOE calculates the checksum at the start of L3 packet. The VLAN1 tag register is used by the RXCOE to indicate what protocol type is to be used to indicate the existence of a VLAN tag. This value is typically 8100h.

Example frame configurations:

FIGURE 11-3: TYPE II ETHERNET FRAMES

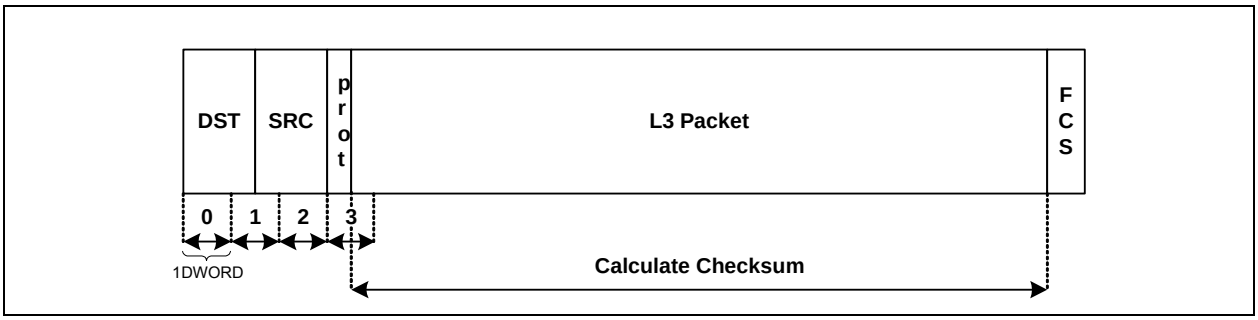


FIGURE 11-4: ETHERNET FRAME WITH VLAN TAG

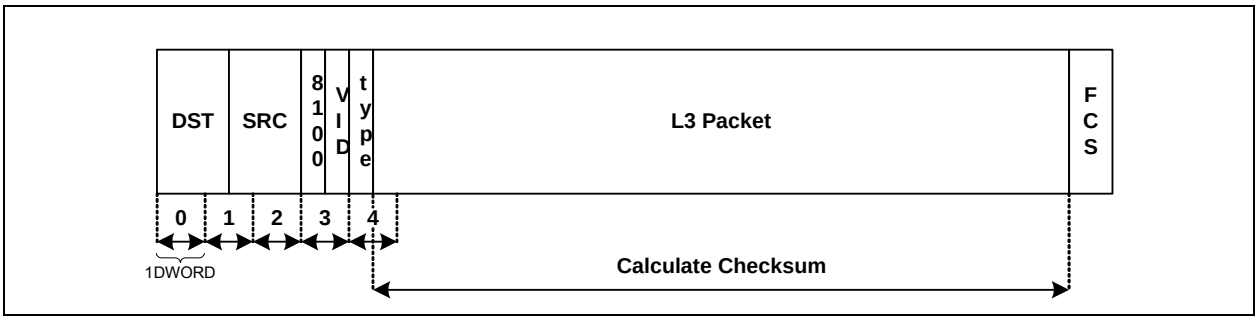


FIGURE 11-5: ETHERNET FRAME WITH LENGTH FIELD AND SNAP HEADER

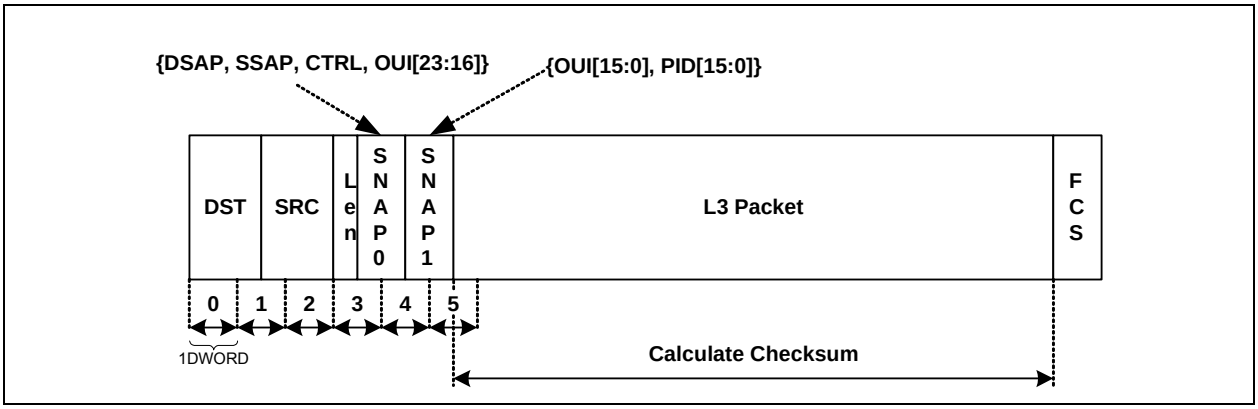


FIGURE 11-6: ETHERNET FRAME WITH VLAN TAG AND SNAP HEADER

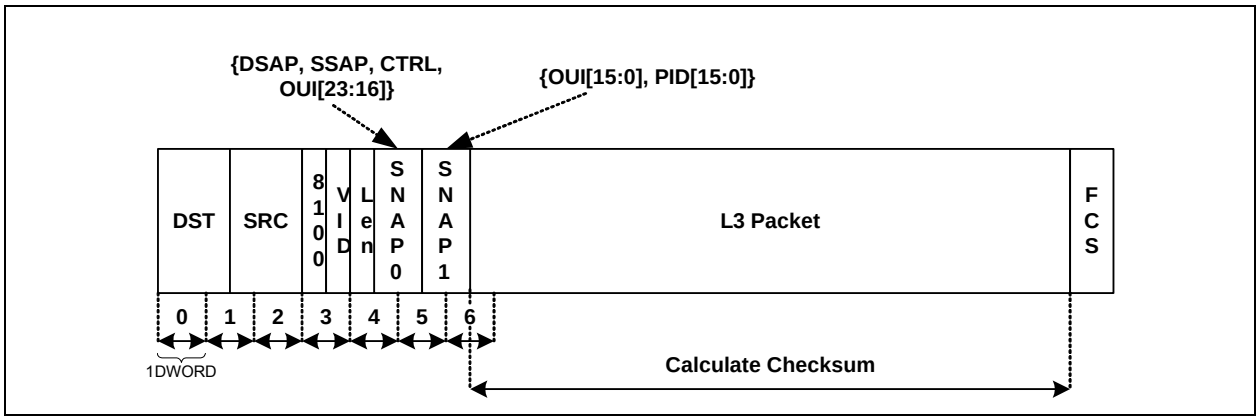
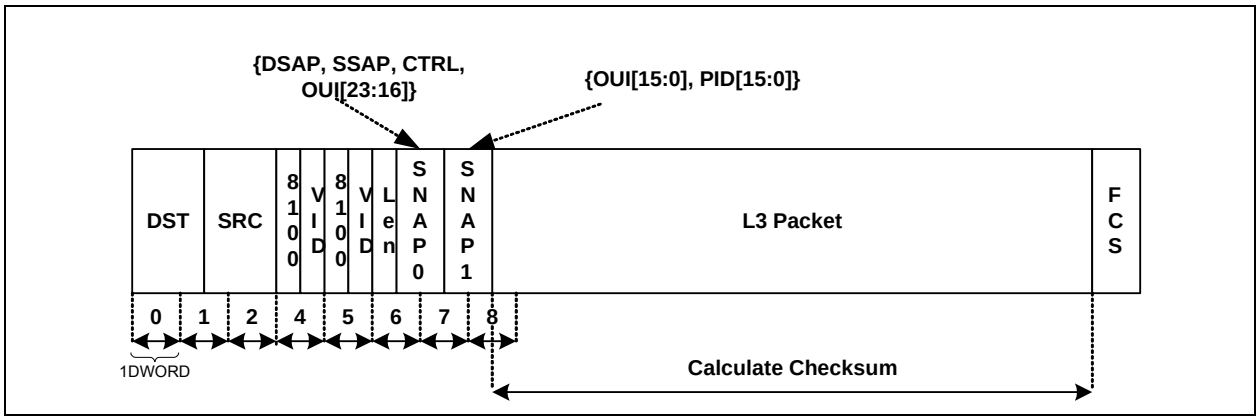


FIGURE 11-7: ETHERNET FRAME WITH MULTIPLE VLAN TAGS AND SNAP HEADER



The RXCOE supports a maximum of two VLAN tags. If there are more than two VLAN tags, the VLAN protocol identifier for the third tag is treated as an Ethernet type field. The checksum calculation will begin immediately after the type field.

Note that in all cases, the checksum calculation ends just before the frames FCS field. In the case where padding is added to meet the minimum frame length requirement, the checksum calculation will also include the pads byte(s). This may lead to unexpected results if the pad byte(s) are not zero.

The RXCOE resides in the RX path within the MAC. As the RXCOE receives an Ethernet frame, it calculates the 16-bit checksum. The RXCOE passes the Ethernet frame to the RX FIFO with the checksum appended to the end of the frame. The RXCOE inserts the checksum immediately after the last byte of the Ethernet frame and before it transmits the status word. The packet length field in the RX Status Word (refer to Section 11.12.3) will indicate that the frame size has increased by two bytes to accommodate the checksum.

Note: When enabled, the RXCOE calculates a checksum for every received frame.

Setting the RX Checksum Offload Engine Enable (RX_COE_EN) bit in the Host MAC Checksum Offload Engine Control Register (HMAC_COE_CR) enables the RXCOE, while the RX Checksum Offload Engine Mode (RX_COE_MODE) bit selects the operating mode. When the RXCOE is disabled, the received data is simply passed through the RXCOE unmodified.

Note: Software applications must stop the receiver and flush the RX data path before changing the state of the RX Checksum Offload Engine Enable (RX_COE_EN) or RX Checksum Offload Engine Mode (RX_COE_-MODE) bits.

Note: When the RXCOE is enabled, automatic pad stripping must be disabled (Automatic Pad Stripping (PADSTR) bit of the Host MAC Control Register (HMAC_CR)) and vice versa. These functions cannot be enabled simultaneously.

11.7.1 RX CHECKSUM CALCULATION

The checksum is calculated 16 bits at a time. In the case of an odd sized frame, an extra byte of zero is used to pad up to 16 bits.

Consider the following packet: DA, SA, Type, B0, B1, B2 … BN, FCS

Let [A, B] = A*256 + B;

If the packet has an even number of octets then

checksum = [B1, B0] + C0 + [B3, B2] + C1 + … + [BN, BN-1] + CN-1

Where C0, C1, ... CN-1 are the carry out results of the intermediate sums.

If the packet has an odd number of octets then

checksum = [B1, B0] + C0 + [B3, B2] + C1 + … + [0, BN] + CN-1

11.8 Transmit Checksum Offload Engine (TXCOE)

The transmit checksum offload engine provides assistance to the CPU by calculating a 16-bit checksum, typically for TCP, for a transmit Ethernet frame. The TXCOE calculates the checksum and inserts the results back into the data stream as it is transferred to the MAC.

To activate the TXCOE and perform a checksum calculation, the Host must first set the TX Checksum Offload Engine Enable (TX_COE_EN) bit in the Host MAC Checksum Offload Engine Control Register (HMAC_COE_CR). The Host then pre-pends a 3 DWORD buffer to the data that will be transmitted. The prepended buffer includes a TX Command A, TX Command B, and a 32-bit TX checksum preamble (refer to Table 11-8). When the CK bit of the TX Command ‘B’ is set in conjunction with the FS bit of TX Command ‘A’ and the TX Checksum Offload Engine Enable (TX_COE_EN) bit of the Host MAC Checksum Offload Engine Control Register (HMAC_COE_CR) register, the TXCOE will perform a checksum calculation on the associated packet. The TX checksum preamble instructs the TXCOE on the handling of the associated packet. The TXCSSP - TX Checksum Start Pointer field of the TX checksum preamble defines the byte offset at which the data checksum calculation will begin. The checksum calculation will begin at this offset and will continue until the end of the packet. The data checksum calculation must not begin in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet. When the calculation is complete, the checksum will be inserted into the packet at the byte offset defined by the TXCSLOC - TX Checksum Location field of the TX checksum preamble. The TX checksum cannot be inserted in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet. If the CK bit is not set in the first TX Command ‘B’ of a packet, the packet is passed directly through the TXCOE without modification, regardless if the TXCOE_EN is set. An example of a TX packet with a prepended TX checksum preamble can be found in Section 11.11.6.3, "TX Example 3". In this example, the Host provides the Ethernet frame to the Ethernet controller in four fragments, the first containing the TX Checksum Preamble. Figure 11-8 shows how these fragments are loaded into the TX Data FIFO. For more information on the TX Command ‘A’ and TX Command ‘B’, refer to Section 11.11.2, "TX Command Format," on page 161.

If the TX packet already includes a partial checksum calculation (perhaps inserted by an upper layer protocol), this checksum can be included in the hardware checksum calculation by setting the TXCSSP field in the TX checksum preamble to include the partial checksum. The partial checksum can be replaced by the completed checksum calculation by setting the TXCSLOC pointer to point to the location of the partial checksum.

TABLE 11-8: TX CHECKSUM PREAMBLE

Field	Description
31	**TXCSUDP - TX Checksum UDP Frame** This bit specifies if a checksum result of 0x0000 should be changed to 0xFFFF.
30:28	**RESERVED**
27:16	**TXCSLOC - TX Checksum Location** This field specifies the byte offset where the TX checksum will be inserted in the TX packet. The checksum will replace two bytes of data starting at this offset. The TX checksum cannot be inserted in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet.
15:12	**RESERVED**
11:0	**TXCSSP - TX Checksum Start Pointer** This field indicates start offset, in bytes, where the checksum calculation will begin in the associated TX packet. The data checksum calculation must not begin in the MAC header (first 14 bytes) or in the last 4 bytes of the TX packet.

Note: When the TXCOE is enabled, the third DWORD of the prepended packet is not transmitted. However, 4 bytes must be added to the packet length field in TX Command B.

Note: Software applications must stop the transmitter and flush the TX data path before changing the state of the TXCOE_EN bit. However, the CK bit of TX Command B can be set or cleared on a per-packet basis.

Note: The TXCOE_MODE may only be changed if the TX path is disabled. If it is desired to change this value during run time, it is safe to do so only after the TX Ethernet path is disabled and the TLI is empty.

Note: The TX checksum preamble must be DWORD-aligned.

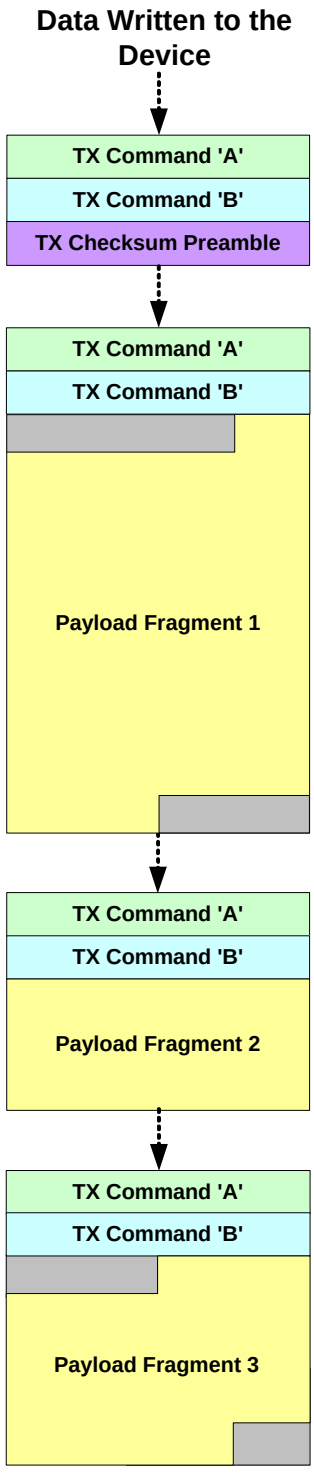
Note: TX preamble size is accounted for in both the buffer length and packet length.

Note: The first buffer, which contains the TX preamble, may not contain any Ethernet frame data

Figure 11-8 illustrates the use of a prepended checksum preamble when transmitting an Ethernet frame consisting of 3 payload buffers.

FIGURE 11-8: TX EXAMPLE ILLUSTRATING A PREPENDED TX CHECKSUM PREAMBLE

NOTE: The TX Checksum Preamble is pre-pended to data to be transmitted. FS is set in TX Command 'A' and CK is set in TX Command 'B'. No start offset may be added. FS must not be set for subsequent fragments of the same packet.



11.8.1 TX CHECKSUM CALCULATION

The TX checksum calculation is performed using the same operation as the RX checksum shown in Section 11.7.1, with the exception that the calculation starts as indicated by the TX checksum preamble and the transmitted checksum is the one's-compliment of the calculated value.

UDP checksums are optional under IPv4, and a checksum value of zero indicates to the receiver that no checksum was calculated. Under IPv6, however, according to RFC 2460, the UDP checksum is not optional. A calculated checksum that yields a result of zero must be changed to FFFFh for insertion into the UDP header. IPv6 receivers discard UDP packets containing a zero checksum. Bit 31 of the checksum preamble specifies if a result of 0x0000 should be changed to 0xFFFF. This allows the choice of checksum usage for UDP and other purposes.

11.9 Host MAC Address

The Host MAC address is configured via the Host MAC Address Low Register (HMAC_ADDRL) and Host MAC Address High Register (HMAC_ADDRH). These registers contain the 48-bit physical address of the Host MAC. The contents of these registers may be loaded directly by the host, or optionally, by the EEPROM Loader from EEPROM at power-on (if a programmed EEPROM is detected).

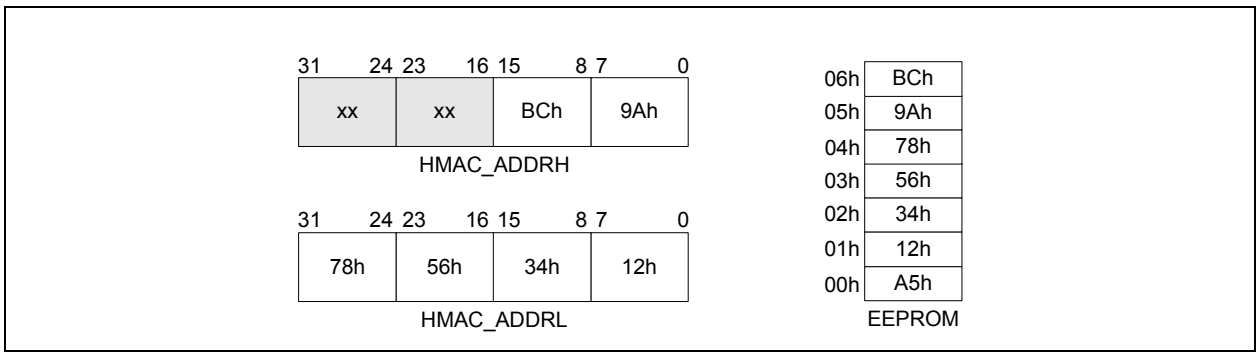
Table 11-9 below illustrates the byte ordering of the HMAC_ADDRL and HMAC_ADDRH registers with respect to the reception of the Ethernet physical address. Also shown is the correlation between the EEPROM addresses and HMAC_ADDRL and HMAC_ADDRH registers.

TABLE 11-9: EEPROM BYTE ORDERING AND REGISTER CORRELATION

EEPROM Address	Register Locations Written	Order of Reception on Ethernet
01h	HMAC_ADDRL[7:0]	1st
02h	HMAC_ADDRL[15:8]	2nd
03h	HMAC_ADDRL[23:16]	3rd
04h	HMAC_ADDRL[31:24]	4th
05h	HMAC_ADDRH[7:0]	5th
06h	HMAC_ADDRH[15:8]	6th

For example, if the desired Ethernet physical address is 12-34-56-78-9A-BC, the HMAC_ADDRL and HMAC_ADDRH registers would be programmed as shown in Figure 11-9. The values required to automatically load this configuration from the EEPROM are also shown.

FIGURE 11-9: EXAMPLE EEPROM MAC ADDRESS SETUP



Note: By convention, the right nibble of the left most byte of the Ethernet address (in this example, the 2 of the 12h) is the most significant nibble and is transmitted/received first.

For more information on the EEPROM and EEPROM Loader, refer to Section 13.0, "I2C Master EEPROM Controller," on page 282.

11.10 FIFOs

The device contains four host-accessible FIFOs (TX Status, RX Status, TX Data, and RX Data) and two internal inaccessible Host MAC TX/RX MIL FIFO's (TX MIL FIFO, RX MIL FIFO).

11.10.1 TX/RX FIFOS

The TX/RX Data and Status FIFOs store the incoming and outgoing address and data information, acting as a conduit between the host bus interface (HBI) and the Host MAC. The sizes of these FIFOs are configurable via the Hardware Configuration Register (HW_CFG) register to the ranges described in Table 11-10. Refer to Section 11.10.3, "FIFO Memory Allocation Configuration" for additional information. The the RX and TX FIFOs related register definitions can be found in section Section 11.14, "Host MAC & FIFO Interface Registers".

The TX and RX Data FIFOs have the base address of 20h and 00h respectively. However, each FIFO is also accessible at seven additional contiguous memory locations, as can be seen in FIGURE 5-1: Register Address Map on page 30. The Host may access the TX or RX Data FIFOs at any of these alias port locations, as they all function identically and contain the same data. This alias port addressing is implemented to allow hosts to burst through sequential addresses.

For HBI access, the TX and RX Data FIFOs may also be accessed using FIFO Direct Selection mode. In this mode, the address input is ignored and all read access are directed to the RX Data FIFO while all write accesses are directed to the TX Data FIFO. See Section 9.4.3.2, "FIFO Direct Select Access," on page 80 and Section 9.5.5.3, "FIFO Direct Select Access," on page 102.

The TX and RX Status FIFOs can each be read from two register locations; the Status FIFO Port, and the Status FIFO PEEK. The TX and RX Status FIFO Ports (48h and 40h respectively) will perform a destructive read, popping the data from the TX or RX Status FIFO. The TX and RX Status FIFO PEEK register locations (4Ch and 44h respectively) allow a non-destructive read of the top (oldest) location of the FIFOs.

Proper use of the The TX/RX Data and Status FIFOs, including the correct data formatting is described in detail in Section 11.11, "TX Data Path Operation," on page 159 and Section 11.12, "RX Data Path Operation," on page 170.

11.10.2 MIL FIFOS

The MAC Interface Layer (MIL), within the Host MAC, contains a 2KB transmit and a 128 Byte receive FIFO which are separate from the TX and RX FIFOs. These MIL FIFOs are not directly accessible from the HBI. The differentiation between the TX/RX FIFOs and the TX/RX MIL FIFOs is that once the transmit or receive packets are in the MIL FIFOs, the host no longer can control or access the TX or RX data. The MIL FIFOs are essentially the working buffers of the Host MAC logic. In the case of reception, the data must be moved into the RX FIFOs before the host can access the data. For TX operations, the MIL operates in store-and-forward mode and will queue an entire frame before beginning transmission.

As space in the TX MIL FIFO frees, data is moved into it from the TX Data FIFO. Depending on the size of the frames to be transmitted, the Host MAC can hold up to two Ethernet frames. This is in addition to any TX data that may be queued in the TX Data FIFO.

Conversely, as data is received, it is moved from the Host MAC to the RX MIL FIFO, and then into the RX Data FIFO. When the RX Data FIFO fills up, the current or subsequent RX frames will be lost until room is made in the RX Data FIFO. For each frame of data that is lost, the Host MAC RX Dropped Frames Counter Register (RX_DROP) is incremented.

RX and TX MIL FIFO levels are not visible to the host processor and operate independent of the TX/RX FIFOs. FIFO levels set for the TX/RX Data and Status FIFOs do not take into consideration the MIL FIFOs.

11.10.3 FIFO MEMORY ALLOCATION CONFIGURATION

TX and RX FIFO space is configurable through the Hardware Configuration Register (HW_CFG). The user must select the FIFO allocation by setting the TX FIFO Size (TX_FIF_SZ) field in the Hardware Configuration Register (HW_CFG). The TX_FIF_SZ field selects the total allocation for the TX data path, including the TX Status FIFO size. The TX Status FIFO size is fixed at 512 Bytes (128 TX Status DWORDs). The TX Status FIFO length is subtracted from the total TX FIFO size with the remainder being the TX Data FIFO Size. The minimum size of the TX FIFOs is 2KB (TX Data and TX Status FIFOs combined). Note that TX Data FIFO space includes both commands and payload data.

RX FIFO Size is the remainder of the unallocated FIFO space (16384 bytes – TX FIFO Size). The RX Status FIFO size is always equal to 1/16 of the RX FIFO size. The RX Status FIFO length is subtracted from the total RX FIFO size with the remainder being the RX Data FIFO Size.

For example, if TX_FIF_SZ = 6 then:

Total TX FIFO Size = 6144 Bytes (6KB)

TX Status FIFO Size = 512 Bytes (Fixed)

TX Data FIFO Size = 6144 – 512 = 5632 Bytes

RX FIFO Size = 16384 – 6144 = 10240 Bytes (10KB)

RX Status FIFO Size = 10240 / 16 = 640 Bytes (160 RX Status DWORDs)

RX Data FIFO Size = 10240 – 640 = 9600 Bytes

Table 11-10 contains an overview of the configurable TX/RX FIFO sizes and defaults. TABLE 11-11: shows every valid setting for the TX_FIF_SZ field and the resulting FIFO sizes. Note that settings not shown in this table are reserved and should not be used.

Note: The RX Data FIFO is considered full 4 DWORDs before the length that is specified in the HW_CFG register.

TABLE 11-10: TX/RX FIFO CONFIGURABLE SIZES

FIFO	Size Range	Default
TX Status	512	512
RX Status	128-892	704
TX Data	1536-13824	4608
RX Data	1920-13440	10560

TABLE 11-11: VALID TX/RX FIFO ALLOCATIONS

TX_FIF_SZ	TX DATA FIFO SIZE (bytes)	TX STATUS FIFO SIZE (bytes)	RX DATA FIFO SIZE (bytes)	RX STATUS FIFO SIZE (bytes)
2	1536	512	13440	896
3	2560	512	12480	832
4	3584	512	11520	768
5	4608	512	10560	704
6	5632	512	9600	640
7	6656	512	8640	576
8	7680	512	7680	512
9	8704	512	6720	448
10	9728	512	5760	384
11	10752	512	4800	320
12	11776	512	3840	256
13	12800	512	2880	192
14	13824	512	1920	128

11.11 TX Data Path Operation

Data is queued for transmission by writing it into the TX Data FIFO. Each packet to be transmitted may be divided among multiple buffers. Each buffer starts with a two DWORD TX command (TX command 'A' and TX command 'B'). The TX command instructs the device on the handling of the associated buffer. Packet boundaries are delineated using control bits within the TX command.

The host provides a 16-bit Packet Tag field in the TX command. The Packet Tag value is appended to the corresponding TX status DWORD. All Packet Tag fields must have the same value for all buffers in a given packet. If tags differ between buffers in the same packet the TXE error will be asserted. Any value may be chosen for a Packet Tag as long as all tags in the same Packet are identical. Packet Tags also provide a method of synchronization between transmitted packets and their associated status. Software can use unique Packet Tags to assist with validating matching status completions.

Note: The use of Packet Tags is not required by the hardware. This field can be used by the LAN software driver for any application. Packet Tags is only one application example.

The Packet Length field in the TX command specifies the number of bytes in the associated packet. All Packet Length fields must have the same value for all buffers in a given packet. Hardware compares the Packet Length field and the actual amount of data received by the Ethernet controller. If the actual packet length count does not match the Packet Length field as defined in the TX command, the Transmitter Error (TXE) flag is asserted.

The device can be programmed to start payload transmission of a buffer on a byte boundary by setting the "Data Start Offset" field in the TX command. The "Data Start Offset" field points to the actual start of the payload data within the first 8 DWORDs of the buffer. Data before the "Data Start Offset" pointer will be ignored. When a packet is split into multiple buffers, each successive buffer may begin on any arbitrary byte.

The device can be programmed to strip padding from the end of a transmit packet in the event that the end of the packet does not align with the host burst boundary. This feature is necessary when the device is operating in a system that always performs multi-word bursts. In such cases the device must guarantee that it can accept data in multiples of the Burst length regardless of the actual packet length. When configured to do so, the device will accept extra data at the end of the packet and will remove the extra padding before transmitting the packet. The device automatically removes data up to the boundary specified in the Buffer End Alignment field specified in each TX command.

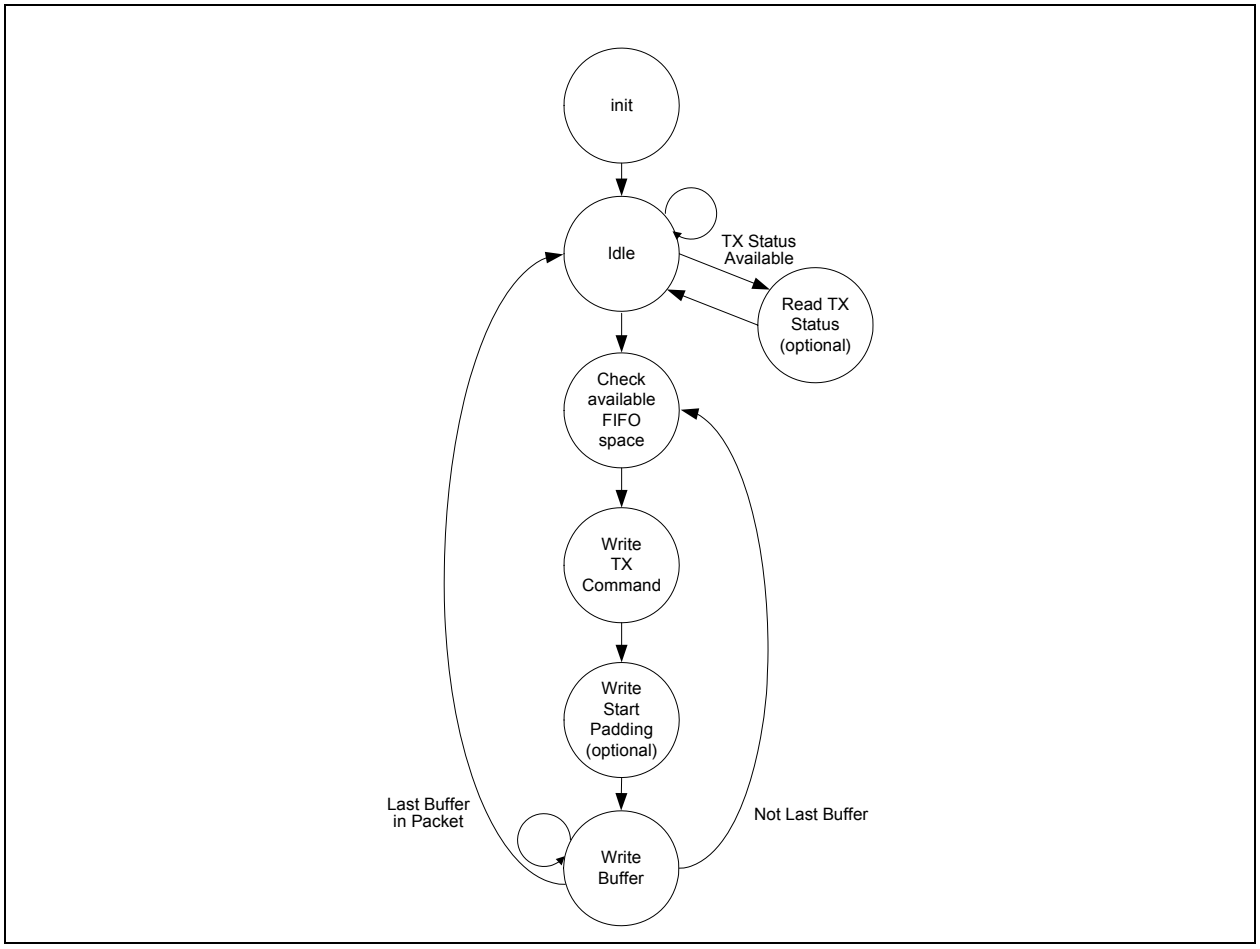
The host can instruct the device to issue an interrupt when the buffer has been fully loaded into the TX FIFO contained in the device and transmitted. This feature is enabled through the TX command 'Interrupt on Completion' field.

Upon completion of transmission, irrespective of success or failure, the status of the transmission is written to the TX Status FIFO. TX status is available to the host and may be read using PIO operations. An interrupt can be optionally enabled by the host to indicate the availability of a programmable number TX status DWORDS.

Before writing the TX command and payload data to the TX FIFO, the host must check the available TX FIFO space by performing a PIO read of the TX FIFO Information Register (TX_FIFO_INF). The host must ensure that it does not over-fill the TX FIFO or the TX Error (TXE) flag will be asserted.

The host proceeds to write the TX command by first writing TX command ‘A’, then TX command ‘B’. After writing the command, the host can then move the payload data into the TX FIFO. TX status DWORDs are stored in the TX Status FIFO to be read by the host at a later time upon completion of the data transmission onto the wire.

FIGURE 11-10: SIMPLIFIED HOST TX FLOW DIAGRAM



11.11.1 TX BUFFER FORMAT

TX buffers exist in the host's memory in a given format. The host writes a TX command word into the TX data buffer before moving the Ethernet packet data. The TX command A and command B are 32-bit values that are used by the device in the handling and processing of the associated Ethernet packet data buffer. Buffer alignment, segmentation and other packet processing parameters are included in the command structure. The buffer format is illustrated in Figure 11-11.

FIGURE 11-11: TX BUFFER FORMAT

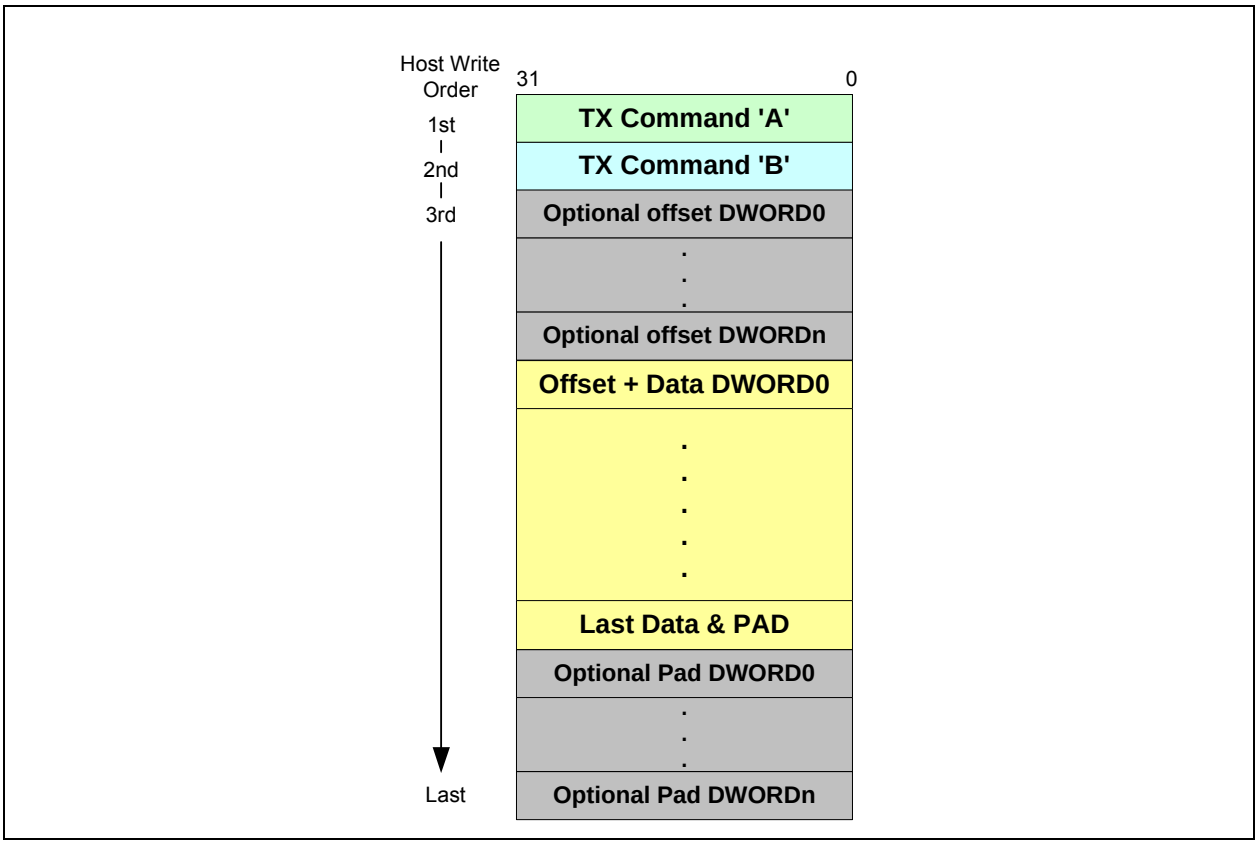


Figure 11-11 shows the TX Buffer as it is written into the device. It should be noted that not all of the data shown in this diagram is actually stored in the TX Data FIFO. This must be taken into account when calculating the actual TX Data FIFO usage. Please refer to Section 11.11.5, "Calculating Actual TX Data FIFO Usage" for a detailed explanation on calculating the actual TX Data FIFO usage.

11.11.2 TX COMMAND FORMAT

The TX command instructs the TX FIFO controller on handling the subsequent buffer. The command precedes the data to be transmitted. The TX command is divided into two, 32-bit words; TX command 'A' and TX command 'B'.

There is a 16-bit Packet Tag in the TX command 'B' command word. Packet Tags may, if host software desires, be unique for each packet (i.e., an incrementing count). The value of the tag will be returned in the TX status word for the associated packet. The Packet tag can be used by host software to uniquely identify each status word as it is returned to the host.

Both TX command 'A' and TX command 'B' are required for each buffer in a given packet. TX command 'B' must be identical for every buffer in a given packet. If the TX command 'B' words do not match, the Ethernet controller will assert the Transmitter Error (TXE) flag.

11.11.2.1 TX Command 'A'

TABLE 11-12: TX COMMAND 'A' FORMAT

Bits	Description
31	**Interrupt on Completion (IOC).** When set, the TX_IOC bit will be asserted in the Interrupt Status Register (INT_STS) when the current buffer has been fully loaded into the TX FIFO.
30:26	**Reserved.** These bits are reserved. Always write zeros to this field to guarantee future compatibility.
25:24	**Buffer End Alignment.** This field specifies the alignment that must be maintained on the last data transfer of a buffer. The host will add extra DWORDs of data up to the alignment specified in the table below. The device will remove the extra DWORDs. This mechanism can be used to maintain cache line alignment on host processors. [25] / [24] / End Alignment 0 / 0 / 4-byte alignment 0 / 1 / 16-byte alignment 1 / 0 / 32-byte alignment 1 / 1 / Reserved
23:21	**Reserved.** These bits are reserved. Always write zeros to this field to guarantee future compatibility
20:16	**Data Start Offset (bytes).** This field specifies the offset of the first byte of TX data. The offset value can be anywhere from 0 bytes to a 31 byte offset.
15:14	**Reserved.** These bits are reserved. Always write zeros to this field to guarantee future compatibility
13	**First Segment (FS).** When set, this bit indicates that the associated buffer is the first segment of the packet.
12	**Last Segment.** When set, this bit indicates that the associated buffer is the last segment of the packet
11	**Reserved.** These bits are reserved. Always write zeros to this field to guarantee future compatibility.
10:0	**Buffer Size (bytes).** This field indicates the number of bytes contained in the buffer following this command. This value, along with the Buffer End Alignment field, is read and checked by the device and used to determine how many extra DWORDs were added to the end of the Buffer. A running count is also maintained in the device of the cumulative buffer sizes for a given packet. This cumulative value is compared against the Packet Length field in the TX command 'B' word and if they do not correlate, the TXE flag is set. The buffer size specified does not include the buffer end alignment padding or data start offset added to a buffer.

11.11.2.2 TX Command 'B'

TABLE 11-13: TX COMMAND 'B' FORMAT

Bits	Description
31:16	**Packet Tag.** The host should write a unique packet identifier to this field. This identifier is added to the corresponding TX status word and can be used by the host to correlate TX status words with their corresponding packets. The use of packet tags is not required by the hardware. This field can be used by the LAN software driver for any application. Packet Tags is one application example.
15	**Reserved.** This bit is reserved. Always write zero to this bit to guarantee future compatibility.
14	**TX Checksum Enable (CK).** When this bit is set in conjunction with the first segment (FS) bit in TX Command 'A' and the TX Checksum Offload Engine Enable (TX_COE_EN) bit in the Host MAC Checksum Offload Engine Control Register (HMAC_COE_CR), the TX checksum offload engine (TXCOE) will calculate a L3 checksum for the associated frame.
13	**Add CRC Disable.** When set, the automatic addition of the CRC is disabled.
12	**Disable Ethernet Frame Padding.** When set, this bit prevents the automatic addition of padding to an Ethernet frame of less than 64 bytes. The CRC field is also added despite the state of the Add CRC Disable field.
11	**Reserved.** These bits are reserved. Always write zeros to this field to guarantee future compatibility.
10:0	**Packet Length (bytes).** This field indicates the total number of bytes in the current packet. This length does not include the offset or padding. If the Packet Length field does not match the actual number of bytes in the packet the Transmitter Error (TXE) flag will be set.

11.11.3 TX DATA FORMAT

The TX data section begins at the third DWORD in the TX buffer (after TX command 'A' and TX command 'B'). The location of the first byte of valid buffer data to be transmitted is specified in the "Data Start Offset" field of the TX command 'A' word. Table 11-14, "TX DATA Start Offset", shows the correlation between the setting of the LSBs in the "Data Start Offset" field and the byte location of the first valid data byte. Additionally, transmit buffer data can be offset by up to 7 additional DWORDS as indicated by the upper three MSBs (5:2) in the "Data Start Offset" field.

TABLE 11-14: TX DATA START OFFSET

Data Start Offset [1:0]:	11	10	01	00
First TX Data Byte:	D[31:24]	D[23:16]	D[15:8]	D[7:0]

TX data is contiguous until the end of the buffer. The buffer may end on a byte boundary. Unused bytes at the end of the packet will not be sent to the Host MAC Interface Layer for transmission.

The Buffer End Alignment field in TX command 'A' specifies the alignment that must be maintained for the associated buffer. End alignment may be specified as 4-, 16-, or 32-byte. The host processor is responsible for adding the additional data to the end of the buffer. The hardware will automatically remove this extra data.

11.11.3.1 TX Buffer Fragmentation Rules

Transmit buffers must adhere to the following rules:

- Each buffer can start and end on any arbitrary byte alignment
- The first buffer of any transmit packet can be any length
- Middle buffers (i.e., those with First Segment = Last Segment = 0) must be greater than, or equal to 4 bytes in length
- The final buffer of any transmit packet can be any length

The MIL operates in store-and-forward mode and has specific rules with respect to fragmented packets. The total space consumed in the TX MIL FIFO must be limited to no more than 2KB - 3 DWORDs (2,036 bytes total). Any transmit packet that is so highly fragmented that it takes more space than this must be un-fragmented (by copying to a driver-supplied buffer) before the transmit packet can be sent to the device.

One approach to determine whether a packet is too fragmented is to calculate the actual amount of space that it will consume, and check it against 2,036 bytes. Another approach is to check the number of buffers against a worst-case limit of 86 (see explanation below).

For the best case alignment scenario (full DWORDs at the start and the end of each buffer), the absolute largest frame size is 2040 bytes (plus the automatically added FCS if enable).

11.11.3.2 Calculating Worst-Case TX MIL FIFO Usage

The actual space consumed by a buffer in the TX MIL FIFO consists only of any partial DWORD offsets in the first/last DWORD of the buffer, plus all of the whole DWORDs in between. Any whole DWORD offsets and/or alignments are stripped off before the buffer is loaded into the TX Data FIFO, and TX command words are stripped off before the buffer is written to the TX MIL FIFO, so none of those DWORDs count as space consumed. The worst-case overhead for a TX buffer is 6 bytes, which assumes that it started on the high byte of a DWORD and ended on the low byte of a DWORD. A TX packet consisting of 86 such fragments would have an overhead of 516 bytes (6 * 86) which, when added to a 1514-byte max-size transmit packet (1516 bytes, rounded up to the next whole DWORD), would give a total space consumption of 2,032 bytes, leaving 4 bytes to spare; this is the basis for the “86 fragment” rule mentioned above. For more information on the MIL FIFO’s refer to Section 11.10.2, "MIL FIFOs," on page 157.

11.11.4 TX STATUS FORMAT

TX status is passed to the host CPU through a separate FIFO mechanism. A status word is returned for each packet transmitted. Data transmission is suspended if the TX Status FIFO becomes full. Data transmission will resume when the host reads the TX status and there is room in the FIFO for more “TX Status” data.

The host can optionally choose to not read the TX status. The TX status can be ignored by setting the “TX Status Discard Allow Overrun Enable” (TXSAO) bit in the Transmit Configuration Register (TX_CFG). Setting this bit high allows the transmitter to continue operation with a full TX Status FIFO. In this mode the status information is still available in the TX Status FIFO, and TX status interrupts still function. In the case of a full FIFO, the TXSUSED counter will stay at its maximum value and no further TX status will be written to the TX Status FIFO, preventing an overrun, until the host frees space by reading TX status. In this mode the host is responsible for re-synchronizing TX status in the case of an overrun.

Bits	Description
31:16	**Packet TAG.** Unique identifier written by the host into the Packet Tag field of the TX command ‘B’ word. This field can be used by the host to correlate TX status words with the associated TX packets.
15	**Error Status (ES).** When set, this bit indicates that the Ethernet controller has reported an error. This bit is the logical OR of bits 11, 10, 9, 8, 2, 1 in this status word.
14:12	**Reserved.** These bits are reserved. Always write zeros to this field to guarantee future compatibility.
11	**Loss of Carrier.** When set, this bit indicates the loss of carrier during transmission.
10	**No Carrier.** When set, this bit indicates that the carrier signal from the transceiver was not present during transmission.
9	**Late Collision.** When set, indicates that the packet transmission was aborted after the collision window of 64 bytes.
8	**Excessive Collisions.** When set, this bit indicates that the transmission was aborted after 16 collisions while attempting to transmit the current packet.
7	**Reserved.** This bit is reserved. Always write zeros to this field to guarantee future compatibility.
6:3	**Collision Count.** This counter indicates the number of collisions that occurred before the packet was transmitted. It is not valid when excessive collisions (bit 8) is also set.

Bits	Description
2	**Excessive Deferral.** If the deferred bit is set in the control register, the setting of the excessive deferral bit indicates that the transmission has ended because of a deferral of over 24288 bit times during transmission.
1	**Reserved.** This bit is reserved.
0	**Deferred.** When set, this bit indicates that the current packet transmission was deferred.

11.11.5 CALCULATING ACTUAL TX DATA FIFO USAGE

The following rules are used to calculate the actual TX Data FIFO space consumed by a TX Packet:

- TX command 'A' is stored in the TX Data FIFO for every TX buffer
- TX command 'B' is written into the TX Data FIFO when the First Segment (FS) bit is set in TX command 'A'
- Any DWORD-long data added as part of the “Data Start Offset” is removed from each buffer before the data is written to the TX Data FIFO. Any data that is less than 1 DWORD is passed to the TX Data FIFO.
- Payload from each buffer within a Packet is written into the TX Data FIFO.
- Any DWORD-long data added as part of the End Padding is removed from each buffer before the data is written to the TX Data FIFO. Any end padding that is less than 1 DWORD is passed to the TX Data FIFO

11.11.6 TRANSMIT EXAMPLES

11.11.6.1 TX Example 1

In this example a single, 111-Byte Ethernet packet will be transmitted. This packet is divided into three buffers. The three buffers are as follows:

Buffer 0:

- 7-Byte “Data Start Offset”
- 79-Bytes of payload data
- 16-Byte “Buffer End Alignment”

Buffer 1:

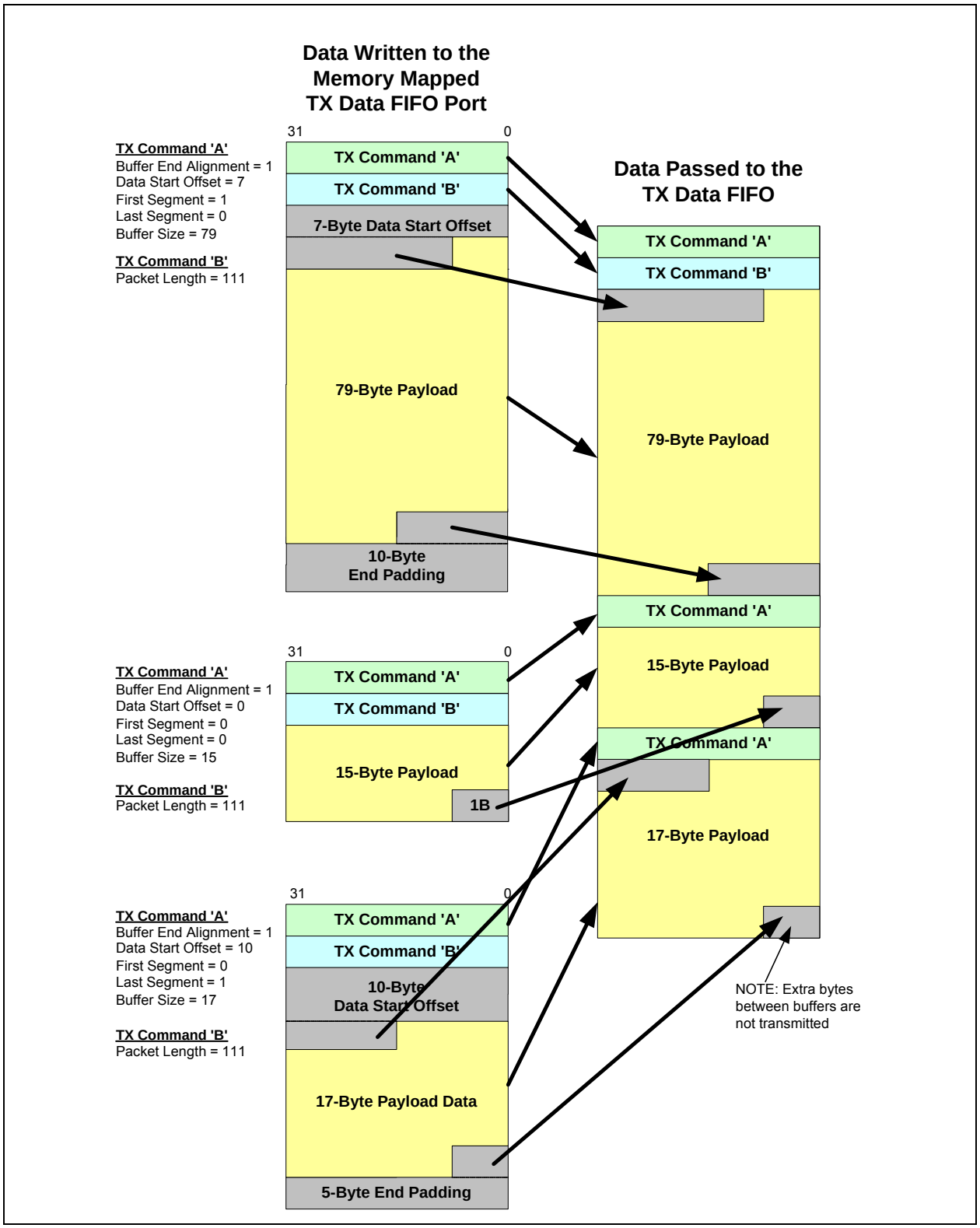
- 0-Byte “Data Start Offset”
- 15-Bytes of payload data
- 16-Byte “Buffer End Alignment”

Buffer 2:

- 10-Byte “Data Start Offset”
- 17-Bytes of payload data
- 16-Byte “Buffer End Alignment”

Figure 11-12 illustrates the TX command structure for this example, and also shows how data is passed to the TX Data FIFO.

FIGURE 11-12: TX EXAMPLE 1



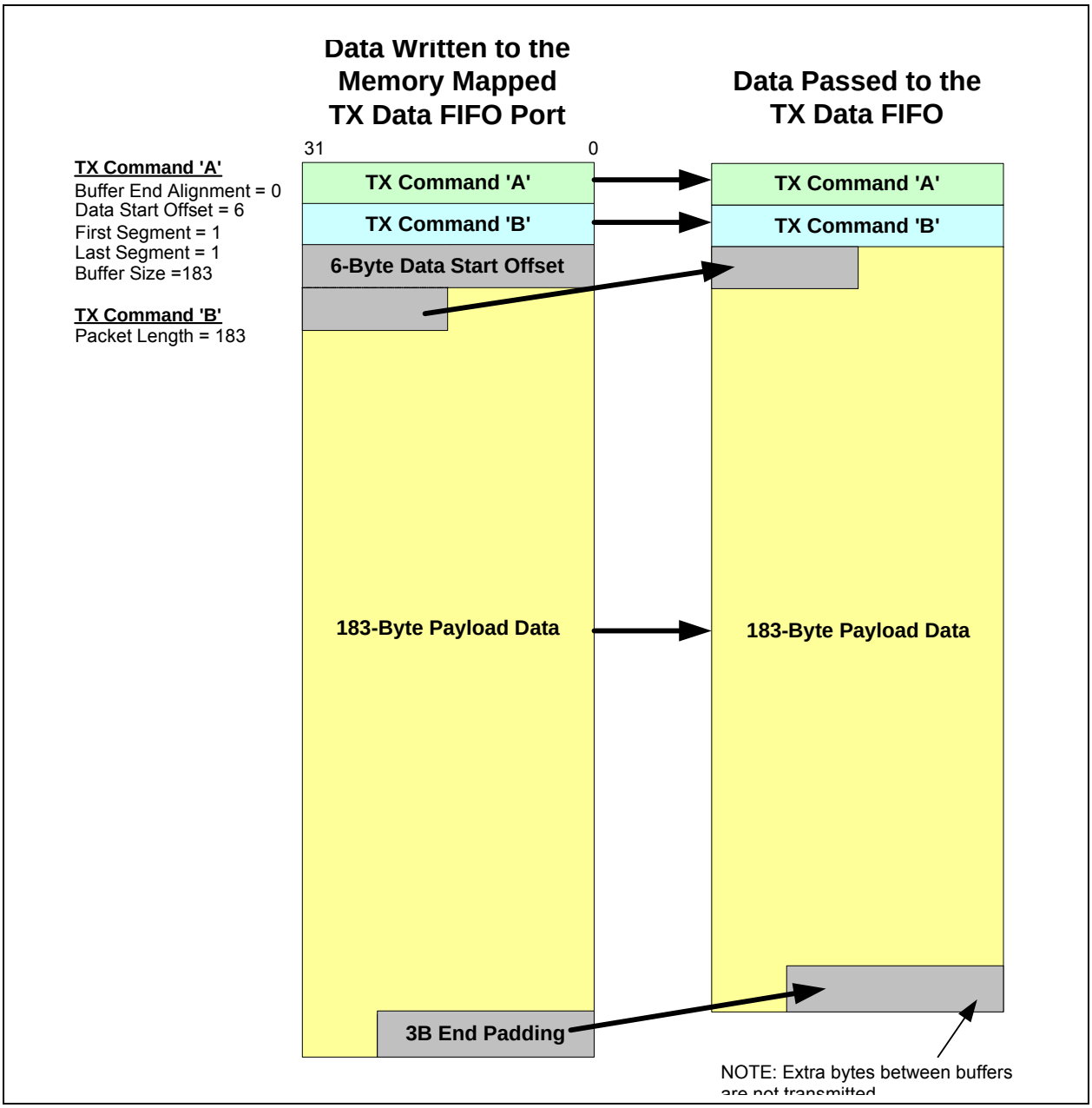
11.11.6.2 TX Example 2

In this example, a single 183-Byte Ethernet packet will be transmitted. This packet is in a single buffer as follows:

- 2-Byte “Data Start Offset”
- 183-Bytes of payload data
- 4-Byte “Buffer End Alignment”

Figure 11-13 illustrates the TX command structure for this example, and also shows how data is passed to the TX Data FIFO. Note that the packet resides in a single TX Buffer, therefore both the FS and LS bits are set in TX command ‘A’.

FIGURE 11-13: TX EXAMPLE 2



11.11.6.3 TX Example 3

In this example a single, 111-Byte Ethernet packet will be transmitted with a TX checksum. This packet is divided into four buffers. The four buffers are as follows:

Buffer 0:

- 4-Byte “Data Start Offset”
- 4-Byte Checksum Preamble
- 16-Byte “Buffer End Alignment”

Buffer 1:

- 7-Byte “Data Start Offset”
- 79-Bytes of payload data
- 16-Byte “Buffer End Alignment”

Buffer 2:

- 0-Byte “Data Start Offset”
- 15-Bytes of payload data
- 16-Byte “Buffer End Alignment”

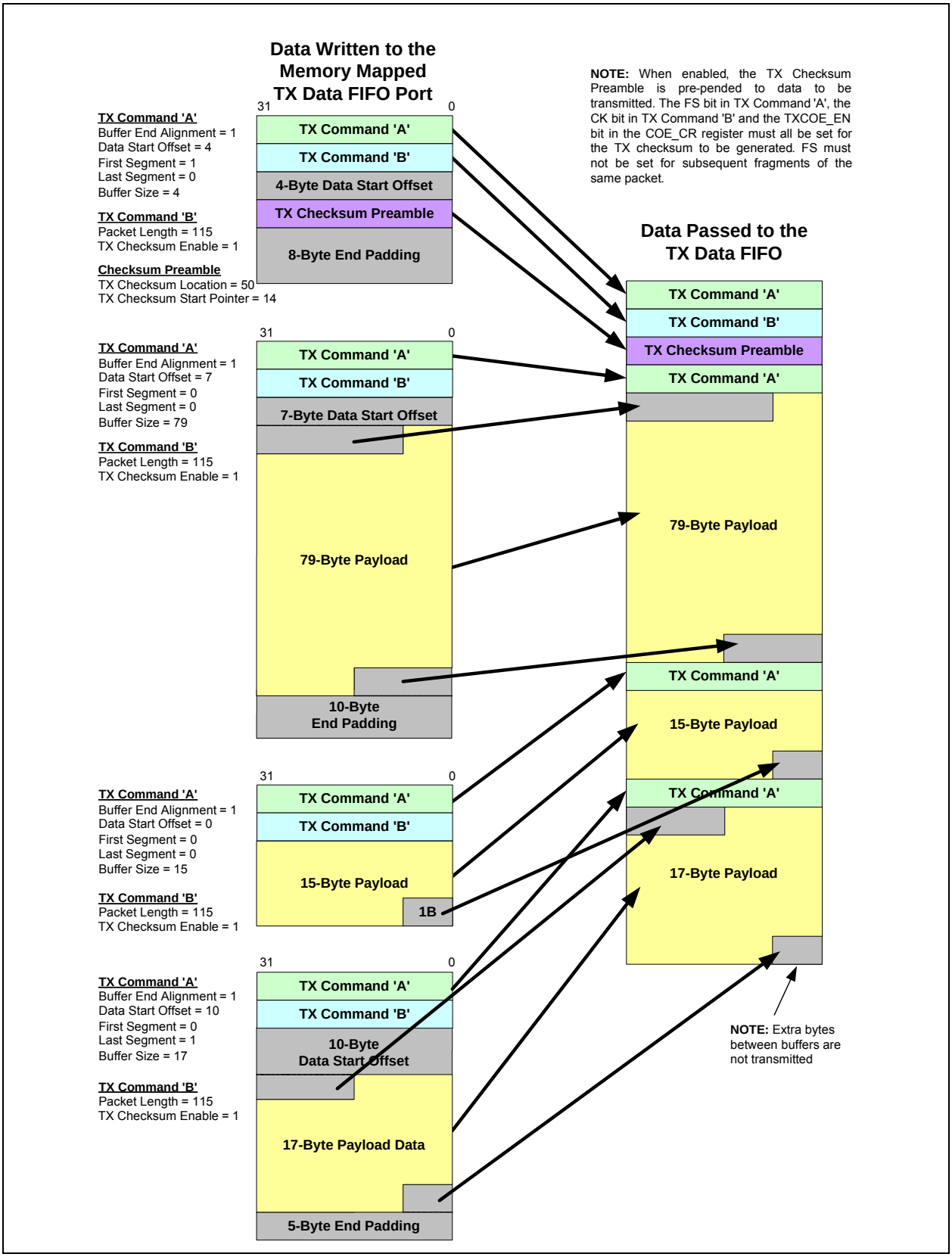
Buffer 3:

- 10-Byte “Data Start Offset”
- 17-Bytes of payload data
- 16-Byte “Buffer End Alignment”

Figure 11-12, "TX Example 1" illustrates the TX command structure for this example, and also shows how data is passed to the TX data FIFO.

Note: In order to perform a TX checksum calculation on the associated packet, bit 14 (CK) of the TX Command ‘B’ must be set in conjunction with bit 13 (FS) of TX Command ‘A’ and the TX Checksum Offload Engine Enable (TX_COE_EN) bit of the Host MAC Checksum Offload Engine Control Register (HMAC_COE_CR). For more information, refer to Section 11.8, "Transmit Checksum Offload Engine (TXCOE)".

FIGURE 11-14: TX EXAMPLE 3



11.11.7 TRANSMITTER ERRORS

If the Transmitter Error (TXE) flag is asserted for any reason, the transmitter will continue operation. TX Error (TXE) will be asserted under the following conditions:

- If the actual packet length count does not match the Packet Length field as defined in the TX command.
- Both TX command 'A' and TX command 'B' are required for each buffer in a given packet. TX command 'B' must be identical for every buffer in a given packet. If the TX command 'B' words do not match, the Ethernet controller will assert the Transmitter Error (TXE) flag.
- Host overrun of the TX Data FIFO.

11.11.8 STOPPING AND STARTING THE TRANSMITTER

To halt the transmitter, the host must set the STOP_TX bit in the Transmit Configuration Register (TX_CFG). The transmitter will finish sending the current frame (if there is a frame transmission in progress). When the transmitter has received the TX status for this frame, it will clear the STOP_TX and TX_ON bits, and will pulse the TXSTOP_INT in the Interrupt Status Register (INT_STS).

Once stopped, the host can optionally clear the TX Status and TX Data FIFOs. The host must re-enable the transmitter by setting the TX_ON bit. If the there are frames pending in the TX Data FIFO (i.e., TX Data FIFO was not purged), the transmission will resume with this data.

11.12 RX Data Path Operation

When an Ethernet Packet is received, the Host MAC Interface Layer (MIL) first begins to transfer the RX data. This data is loaded into the RX Data FIFO. The RX Data FIFO pointers are updated as data is written into the FIFO.

The last transfer from the MIL is the RX status word. The device implements a separate FIFO for the RX status words. The total available RX data and status queued in the RX FIFO can be read from the RX FIFO Information Register (RX_FIFO_INF). The host may read any number of available RX status words before reading the RX Data FIFO.

The host must use caution when reading the RX data and status. The host must never read more data than what is available in the FIFO's. If this is attempted an underrun condition will occur. If this error occurs, the Ethernet controller will assert the Receiver Error (RXE) interrupt. If an underrun condition occurs, a soft reset is required to regain host synchronization.

A configurable beginning offset is supported in the device. The RX data Offset field in the Receive Configuration Register (RX_CFG) controls the number of bytes that the beginning of the RX data buffer is shifted. The host can set an offset from 0-31 bytes. The offset may be changed in between RX packets, but it must not be changed during an RX packet read.

The device can be programmed to add padding at the end of a receive packet in the event that the end of the packet does not align with the host burst boundary. This feature is necessary when the device is operating in a system that always performs multi-DWORD bursts. In such cases the device must guarantee that it can transfer data in multiples of the Burst length regardless of the actual packet length. When configured to do so, the device will add extra data at the end of the packet to allow the host to perform the necessary number of reads so that the Burst length is not cut short. Once a packet has been padded by the H/W, it is the responsibility of the host to interrogate the packet length field in the RX status and determine how much padding to discard at the end of the packet.

It is possible to read multiple packets out of the RX Data FIFO in one continuous stream. It should be noted that the programmed Offset and Padding will be added to each individual packet in the stream, since packet boundaries are maintained.

11.12.1 RX SLAVE PIO OPERATION

Using PIO mode, the host can either implement a polling or interrupt scheme to empty the received packet out of the RX Data FIFO. The host will remain in the idle state until it receives an indication (interrupt or polling) that data is available in the RX Data FIFO. The host will then read the RX Status FIFO to get the packet status, which will contain the packet length and any other status information. The host should perform the proper number of reads, as indicated by the packet length *plus* the start offset *and* the amount of optional padding added to the end of the frame, from the RX Data FIFO. A typical host receive routine using interrupts can be seen in Figure 11-15, while a typical host receive routine using polling can be seen in Figure 11-16.

FIGURE 11-15: HOST RECEIVE ROUTINE USING INTERRUPTS

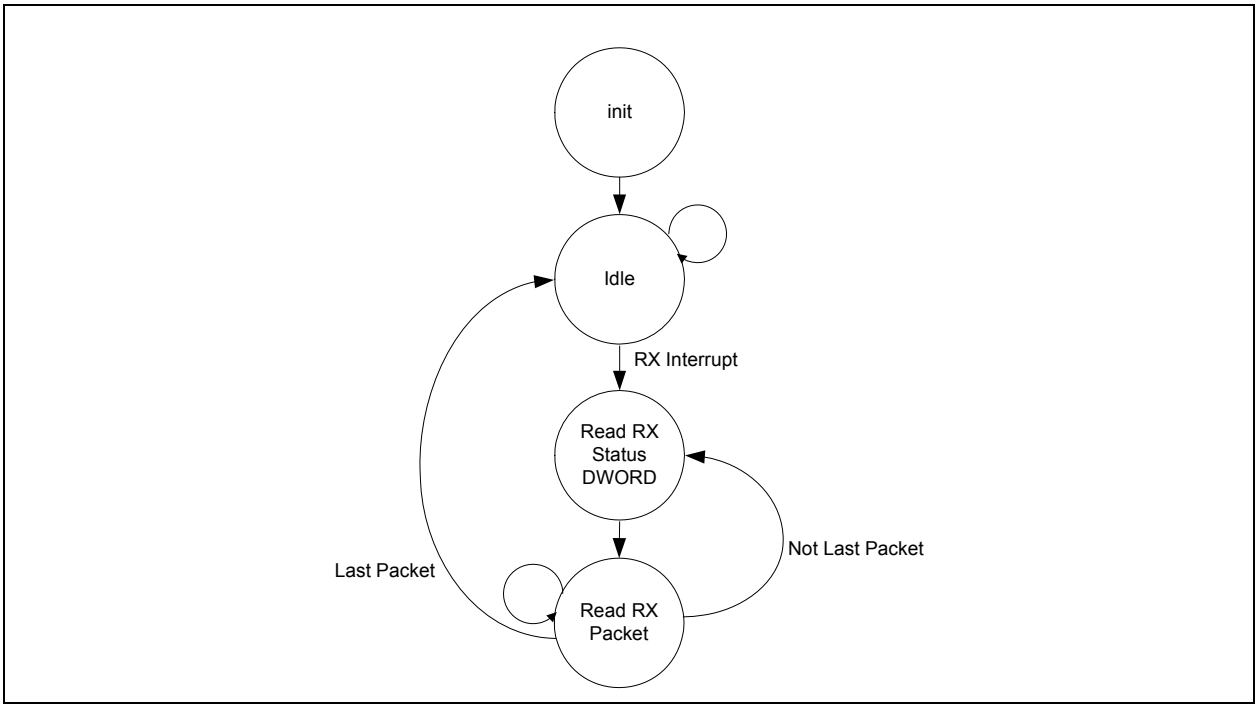
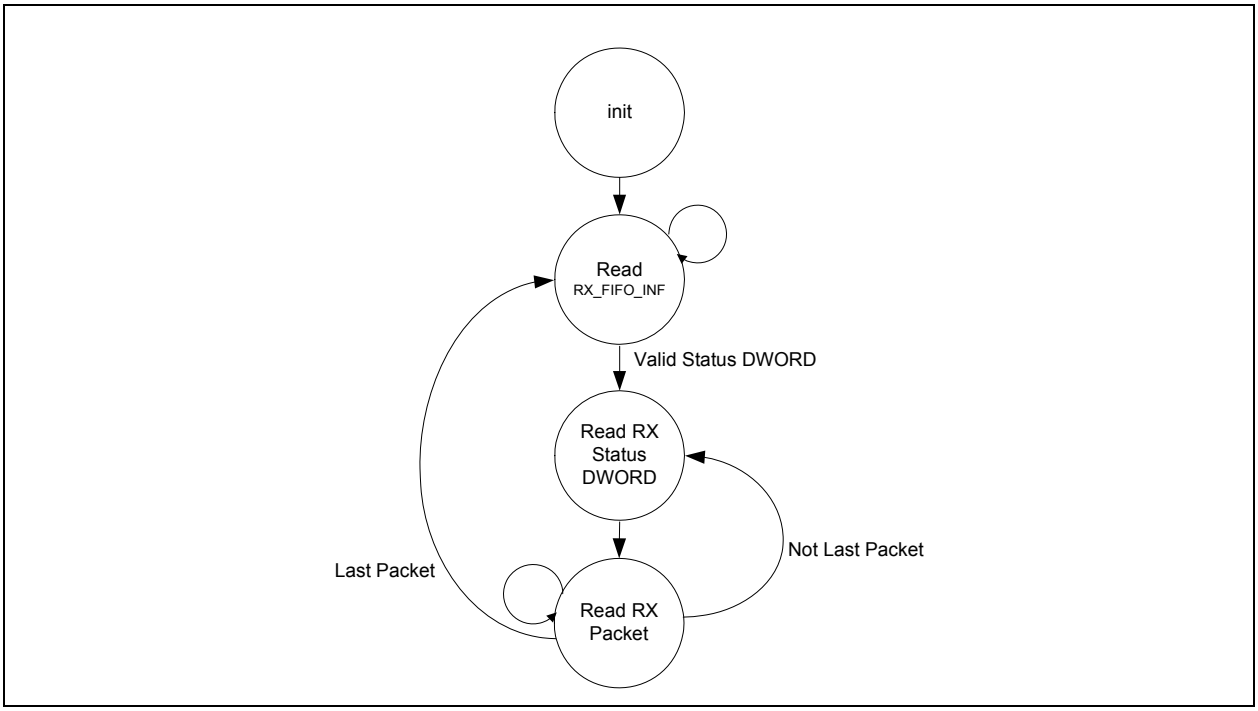


FIGURE 11-16: HOST RECEIVE ROUTINE USING POLLING



11.12.1.1 Receive Data FIFO Fast Forward

The RX data path implements an automatic data discard function. Using the RX Data FIFO Fast Forward bit (RX_FFWD) in the Receive Datapath Control Register (RX_DP_CTRL), the host can instruct the device to skip the packet at the head of the RX Data FIFO. The RX Data FIFO pointers are automatically incremented to the beginning of the next RX packet.

When performing a fast-forward, there must be at least 4 DWORDs of data in the RX Data FIFO for the packet being discarded. For cases with less than 4 DWORDs, do not use RX_FFWD. In this case data must be read from the RX Data FIFO and discarded using standard PIO read operations.

After initiating a fast-forward operation, do not perform any reads of the RX Data FIFO until the RX_FFWD bit is cleared. Other resources can be accessed during this time (i.e., any registers and/or the other three FIFO's). Also note that the RX_FFWD will only fast-forward the RX Data FIFO, not the RX Status FIFO. After an RX fast-forward operation the RX status must still be read from the RX Status FIFO.

The receiver does not have to be stopped to perform a fast-forward operation.

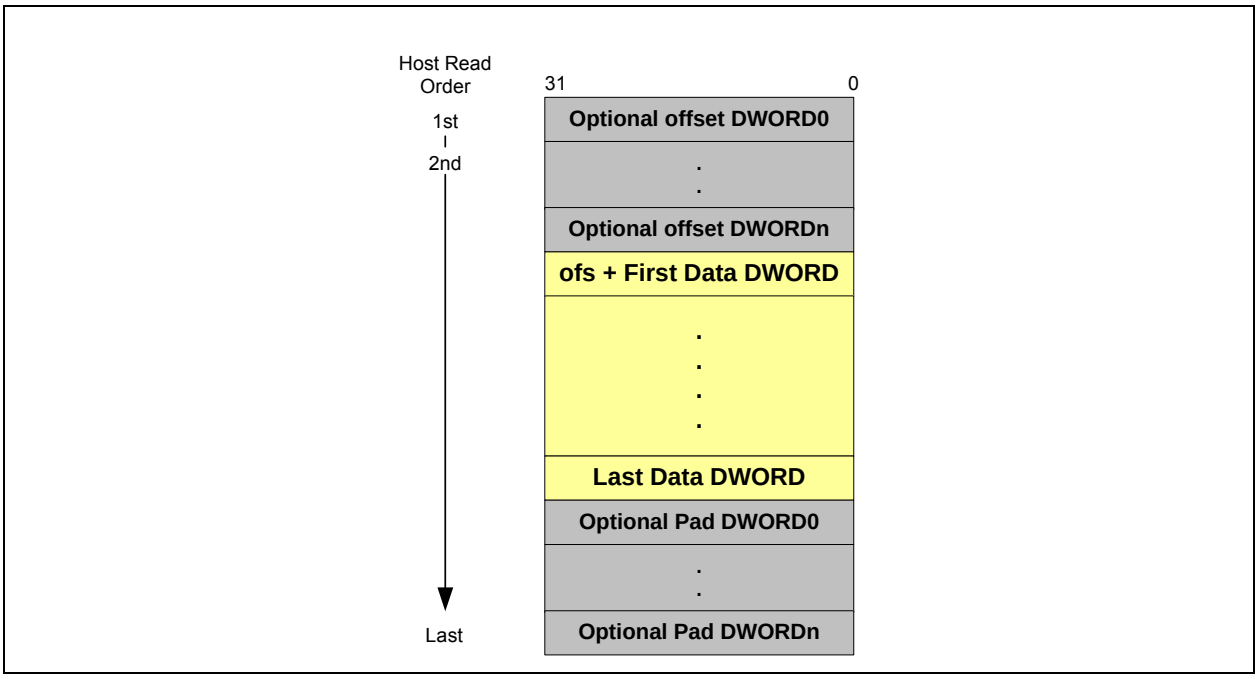
11.12.1.2 Force Receiver Discard (Receiver Dump)

In addition to the Receive data Fast Forward feature, device also implements a receiver "dump" feature. This feature allows the host processor to flush the entire contents of the RX Data and RX Status FIFOs. When activated, the read and write pointers for the RX Data and Status FIFO's will be returned to their reset state. To perform a receiver dump, the device receiver must be halted. Once the receiver stop completion is confirmed, the RX_DUMP bit can be set in the Receive Configuration Register (RX_CFG). The RX_DUMP bit is cleared when the dump is complete. For more information on stopping the receiver, please refer to Section 11.12.4, "Stopping and Starting the Receiver". For more information on the RX_DUMP bit, please refer to Section 11.14.2, "Receive Configuration Register (RX_CFG)," on page 178.

11.12.2 RX PACKET FORMAT

The RX status words can be read from the RX Status FIFO port, while the RX data packets can be read from the RX Data FIFO. RX data packets are formatted in a specific manner before the host can read them as shown in Figure 11-17. It is assumed that the host has previously read the associated status word from the RX Status FIFO, to ascertain the data size and any error conditions.

FIGURE 11-17: RX PACKET FORMAT



11.12.3 RX STATUS FORMAT

BITS	DESCRIPTION
31	**Packet Filter.** When set, this bit indicates that the associated frame passed the frame filtering described in Section 11.5.
30	**Filtering Fail.** When set, this bit indicates that the associated frame failed the address recognizing filtering described in Section 11.4.
29:16	**Packet Length.** The size, in bytes, of the corresponding received frame.
15	**Error Status (ES).** When set this bit indicates that the Host MAC Interface Layer (MIL) has reported an error. This bit is the Internal logical “or” of bits 11,7,6 and 1.
14	**Reserved.** These bits are reserved. Reads 0.
13	**Broadcast Frame.** When set, this bit indicates that the received frame has a Broadcast address.
12	**Length Error (LE).** When set, this bit indicates that the actual length does not match with the length/type field of the received frame.
11	**Runt Frame.** When set, this bit indicates that frame was prematurely terminated before the collision window (64 bytes). Runt frames are passed on to the host only if the Pass Bad Frames bit (PASSBAD) of the Host MAC Control Register (HMAC_CR) is set.
10	**Multicast Frame.** When set, this bit indicates that the received frame has a Multicast address.
9:8	**Reserved.** These bits are reserved. Reads 0.
7	**Frame Too Long.** When set, this bit indicates that the frame length exceeds the maximum Ethernet specification of 1518 bytes. This is only a frame too long indication and will not cause the frame reception to be truncated.
6	**Collision Seen.** When set, this bit indicates that the frame has seen a collision after the collision window. This indicates that a late collision has occurred.
5	**Frame Type.** When set, this bit indicates that the frame is an Ethernet-type frame (Length/Type field in the frame is greater than 1500). When reset, it indicates the incoming frame was an 802.3 type frame. This bit is not set for Runt frames less than 14 bytes.
4	**Receive Watchdog time-out.** When set, this bit indicates that the incoming frame was greater than or equal to 2048 bytes, therefore expiring the Receive Watchdog Timer. Frames greater than or equal to 2049 bytes are truncated to 2048 bytes and would most likely have a CRC error as a result.
3	**MII Error.** When set, this bit indicates that a receive error was detected during frame reception.
2	**Dribbling Bit.** When set, this bit indicates that the frame contained a non-integer multiple of 8 bits. This error is reported only if the number of dribbling bits in the last byte is at least 3 in the 10 Mbps operating mode. This bit will not be set when the collision seen bit[6] is set. If set and the CRC error bit is [1] reset, then the packet is considered to be valid.
1	**CRC Error.** When set, this bit indicates that a CRC error was detected. This bit is also set when the RX_ER pin is asserted during the reception of a frame even though the CRC may be correct. This bit is not valid if the received frame is a Runt frame, or a late collision was detected.
0	**Reserved.** These bits are reserved. Reads 0

11.12.4 STOPPING AND STARTING THE RECEIVER

To stop the receiver, the host must clear the RXEN bit in the Host MAC Control Register (HMAC_CR). When the receiver is halted, the RXSTOP_INT will be pulsed and reflected in the Interrupt Status Register (INT_STS). Once stopped, the host can optionally clear the RX Status and RX Data FIFOs. The host must re-enable the receiver by setting the RXEN bit.

11.12.5 RECEIVER ERRORS

If the Receiver Error (RXE) flag is asserted in the Interrupt Status Register (INT_STS) for any reason, the receiver will continue operation. RX Error (RXE) will be asserted under the following conditions:

- A host underrun of RX Data FIFO
- A host underrun of the RX Status FIFO
- An overrun of the RX Status FIFO (RX Status FIFO Full Interrupt (RSFF))

It is the duty of the host to identify and resolve any error conditions.

11.13 IEEE 802.3az Energy Efficient Ethernet

The device supports Energy Efficient Ethernet (EEE) in 100 Mbps mode as defined in the most recent version of the IEEE 802.3az standard.

Energy Efficient Ethernet is enabled via Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Host MAC Control Register (HMAC_CR).

11.13.1 TX LPI GENERATION

The process of when the MAC should indicate LPI requests to the PHY is divided into two sections:

- Client LPI Requests to MAC
- MAC LPI Request to PHY

11.13.1.1 Client LPI Requests to MAC

When the TX FIFO is empty for a time (in us) specified in Host MAC EEE TX LPI Request Delay Register (HMAC_EEE_TX_LPI_REQ_DELAY) a TX LPI request is asserted to the MAC. A setting of 0 us is possible for this time. If the TX FIFO becomes not empty while the timer is running, the timer is reset (i.e. empty time is not cumulative). Once TX LPI is requested and the TX FIFO becomes not empty, the TX LPI request is negated.

The TX FIFO empty timer is reset if Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) in the Host MAC Control Register (HMAC_CR) is cleared.

TX LPI requests are asserted only if the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit is set, and if the current speed is 100 Mbps, the current duplex is full (as indicated by the Full Duplex Mode (FDPX) bit of the Host MAC Control Register (HMAC_CR)) and the auto-negotiation result indicates that both the local and partner device support EEE 100 Mbps. In order to prevent an unstable link condition, the PHY link status also must indicate “up” for one second before LPI is requested.

TX LPI requests are asserted even if the Transmitter Enable (TX_ON) bit in the Transmit Configuration Register (TX_CFG) is cleared.

11.13.1.2 MAC LPI Request to PHY

Lower Power Idle (LPI) is requested by the Host MAC to the PHY using the MII value of TXEN=0, TXER=1, TXD[3:0]=4’b0001. The MAC always finishes the current packet before signaling TX LPI to the PHY. The MAC will generate TX LPI requests to the PHY even if the Transmitter Enable (TX_ON) bit in the Transmit Configuration Register (TX_CFG) is cleared.

802.3az specifies the usage of a simplified full duplex MAC with carrier sense deferral. Basically this means that once the TX LPI request to the PHY is de-asserted, the Host MAC will defer the time specified in Host MAC EEE Time Wait TX System Register (HMAC_EEE_TW_TX_SYS) in addition to the normal IPG before sending a frame.

TX LPI COUNTERS

The Host MAC maintains a counter, EEE TX LPI Transitions, that counts the number of times that TX LPI request to the PHY changes from de-asserted to asserted. The counter is not writable and does not clear on read. The counter is reset if the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Host MAC Control Register (HMAC_CR) is low.

The Host MAC maintains a counter, EEE TX LPI Time, that counts (in us) the amount of time that TX LPI request to the PHY is asserted. Note that this counter does not include the time specified in the Host MAC EEE Time Wait TX System Register (HMAC_EEE_TW_TX_SYS). The counter is not writable and does not clear on read. The counter is reset if the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Host MAC Control Register (HMAC_CR) is low.

11.13.2 RX LPI DETECTION

Receive Lower Power Idle (LPI) is indicated by the PHY to the MAC using the MII value of RXDV=0, RXER=1, TXD[3:0]=4'b0001.

11.13.2.1 Decoding LPI

The MAC will decode the LPI indication only when Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) is set in the Host MAC Control Register (HMAC_CR), and if the current speed is 100Mbs, the current duplex is full (as indicated by the Full Duplex Mode (FDPX) bit of the Host MAC Control Register (HMAC_CR)) and the auto-negotiation result indicates that both the local and partner device supports EEE at 100Mbs. In order to prevent an unstable link condition, the PHY link status also must indicate "up" for one second is set.) before LPI is decoded.

11.13.2.2 RX LPI Counters

The Host MAC maintains a counter, EEE RX LPI Transitions, that counts the number of times that the LPI indication from the PHY changes from de-asserted to asserted. The counter is not writable and does not clear on read. The counter is reset if the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Host MAC Control Register (HMAC_CR) is low.

The Host MAC maintains a counter, EEE RX LPI Time, that counts (in us) the amount of time that the PHY indicates LPI. The counter is not writable and does not clear on read. The counter is reset if the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Host MAC Control Register (HMAC_CR) is low.

11.14 Host MAC & FIFO Interface Registers

This section details the directly addressable Host MAC and TX/RX FIFO related System CSRs. These registers allow for the configuration of the TX/RX FIFO's, Host MAC and indirect access to the complete set of Host MAC CSRs. The Host MAC CSRs are accessible through via the Host MAC CSR Interface Command Register (MAC_CSR_CMD) and Host MAC CSR Interface Data Register (MAC_CSR_DATA).

Note: For more information on the TX/RX FIFO's, refer to Section 11.10, "FIFOs".

Note: The full list of indirectly addressable Host MAC CSRs are described in Section 11.15, "Host MAC Control and Status Registers," on page 192.

TABLE 11-15: HOST MAC & FIFO INTERFACE LOGIC REGISTERS

Address	Register Name (SYMBOL)
068h	FIFO Level Interrupt Register (FIFO_INT)
06Ch	Receive Configuration Register (RX_CFG)
070h	Transmit Configuration Register (TX_CFG)
078h	Receive Datapath Control Register (RX_DP_CTRL)
07Ch	RX FIFO Information Register (RX_FIFO_INF)
080h	TX FIFO Information Register (TX_FIFO_INF)
0A0h	Host MAC RX Dropped Frames Counter Register (RX_DROP)
0A4h	Host MAC CSR Interface Command Register (MAC_CSR_CMD)
0A8h	Host MAC CSR Interface Data Register (MAC_CSR_DATA)
0ACh	Host MAC Automatic Flow Control Configuration Register (AFC_CFG)
0B0h	Host MAC RX LPI Transitions Register (HMAC_RX_LPI_TRANSITION)
0B4h	Host MAC RX LPI Time Register (HMAC_RX_LPI_TIME)
0B8h	Host MAC TX LPI Transitions Register (HMAC_TX_LPI_TRANSITION)
0BCh	Host MAC TX LPI Time Register (HMAC_TX_LPI_TIME)

11.14.1 FIFO LEVEL INTERRUPT REGISTER (FIFO_INT)

Offset: 068h Size: 32 bits

This read/write register configures the limits where the RX/TX Data and Status FIFO's will generate system interrupts.

Bits	Description	Type	Default
31:24	**TX Data Available Level** The value in this field sets the level, in number of 64 Byte blocks, at which the TX Data FIFO Available Interrupt (TDFA) will be generated. When the TX Data FIFO free space is greater than this value, a TX Data FIFO Available Interrupt (TDFA) will be generated in the Interrupt Status Register (INT_STS).	R/W	48h
23:16	**TX Status Level** The value in this field sets the level, in number of DWORDs, at which the TX Status FIFO Level Interrupt (TSFL) will be generated. When the TX Status FIFO used space is greater than this value, a TX Status FIFO Level Interrupt (TSFL) will be generated in the Interrupt Status Register (INT_STS).	R/W	00h
15:8	**RESERVED**	RO	-
7:0	**RX Status Level** The value in this field sets the level, in number of DWORDs, at which the RX Status FIFO Level Interrupt (RSFL) will be generated. When the RX Status FIFO used space is greater than this value, a RX Status FIFO Level Interrupt (RSFL) will be generated in the Interrupt Status Register (INT_STS).	R/W	00h

11.14.2 RECEIVE CONFIGURATION REGISTER (RX_CFG)

Offset: 06Ch Size: 32 bits

This register controls the Host MAC receive engine.

Bits	Description	Type	Default
31:30	**RX End Alignment (RX_EA)** This field specifies the alignment that must be maintained on the last data transfer of a buffer. The device will add extra DWORDs of data up to the alignment specified in the table below. The host is responsible for removing these extra DWORDs. This mechanism can be used to maintain cache line alignment on host processors. **Bit Values [31:30] / End Alignment** 00 – 4-Byte Alignment 01 – 16-Byte Alignment 10 – 32-Byte Alignment 11 – RESERVED **Note:** The desired RX End Alignment must be set before reading a packet. The RX End Alignment can be changed between reading receive packets, but must not be changed if the packet is partially read.	R/W	00b
29:28	**RESERVED**	RO	-
27:16	**RX DMA Count (RX_DMA_CNT)** This 12-bit field indicates the amount of data, in DWORDs, to be transferred out of the RX Data FIFO before asserting the RX DMA Interrupt (RXD_INT). After being set, this field is decremented for each DWORD of data that is read from the RX Data FIFO. This field can be overwritten with a new value before it reaches zero.	R/W	000h
15	**Force RX Discard (RX_DUMP)** When a 1 is written to this bit, the RX Data and Status FIFO's are cleared of all pending data and the RX data and status pointers are cleared to zero. **Note:** Please refer to Section 11.12.1.2, "Force Receiver Discard (Receiver Dump)," on page 172 for a detailed description regarding the use of RX_DUMP.	WO SC	0b
14:13	**RESERVED**	RO	-

Bits	Description	Type	Default
12:8	**RX Data Offset (RXDOFF)** This field controls the offset value, in bytes, that is added to the beginning of an RX data packet. The start of the valid data will be shifted by the number of bytes specified in this field. An offset of 0-31 bytes is a valid number of offset bytes. **Note:** The two LSBs of this field (D[9:8]) must not be modified while the RX is running. The receiver must be halted, and all data purged before these two bits can be modified. The upper three bits (DWORD offset) may be modified while the receiver is running. Modifications to the upper bits will take affect on the next DWORD read.	R/W	00000b
7:0	**RESERVED**	RO	-

11.14.3 TRANSMIT CONFIGURATION REGISTER (TX_CFG)

Offset: 070h Size: 32 bits

This register controls the Host MAC transmit functions.

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15	**Force TX Status Discard (TXS_DUMP)** When a 1 is written to this bit, the TX Status FIFO is cleared of all pending status DWORDs and the TX status pointers are cleared to zero.	WO SC	0b
14	**Force TX Data Discard (TXD_DUMP)** When a 1 is written to this bit, the TX Data FIFO is cleared of all pending data and the TX data pointers are cleared to zero.	WO SC	0b
13:3	**RESERVED**	RO	-
2	**TX Status Allow Overrun (TXSAO)** When this bit is cleared, Host MAC data transmission is suspended if the TX Status FIFO becomes full. Setting this bit high allows the transmitter to continue operation with a full TX Status FIFO. **Note:** This bit does not affect the operation of the TX Status FIFO Full Interrupt (TSFF).	R/W	0b
1	**Transmitter Enable (TX_ON)** When this bit is set, the Host MAC transmitter is enabled. Any data in the TX Data FIFO will be sent. This bit is cleared automatically when the STOP_TX bit is set and the transmitter is halted.	R/W	0b
0	**Stop Transmitter (STOP_TX)** When this bit is set, the Host MAC transmitter will finish the current frame, and will then stop transmitting. When the transmitter has stopped this bit will clear. All writes to this bit are ignored while this bit is high.	R/W SC	0b

11.14.4 RECEIVE DATAPATH CONTROL REGISTER (RX_DP_CTRL)

Offset: 078h Size: 32 bits

This register is used to discard unwanted receive frames.

Bits	Description	Type	Default
31	**RX Data FIFO Fast Forward (RX_FFWD)** Writing a 1 to this bit causes the RX Data FIFO to fast-forward to the start of the next frame. This bit will remain high until the RX Data FIFO fast-forward operation has completed. No reads should be issued to the RX Data FIFO while this bit is high. **Note:** Please refer to section Section 11.12.1.1, "Receive Data FIFO Fast Forward," on page 172 for detailed information regarding the use of RX_FFWD.	R/W SC	0h
30:0	**RESERVED**	RO	-

11.14.5 RX FIFO INFORMATION REGISTER (RX_FIFO_INF)

Offset: 07Ch Size: 32 bits

This register contains the indication of used space in the RX FIFO's.

Bits	Description	Type	Default
31:24	**RESERVED**	RO	-
23:16	**RX Status FIFO Used Space (RXSUSED)** This field indicates the amount of space, in DWORDs, currently used in the RX Status FIFO.	RO	0b
15:0	**RX Data FIFO Used Space (RXDUSED)** This field indicates the amount of space, in bytes, used in the RX Data FIFO. For each receive frame, the field is incremented by the length of the receive data. In cases where the payload does not end on a DWORD boundary, the total will be rounded up to the nearest DWORD.	RO	0b

11.14.6 TX FIFO INFORMATION REGISTER (TX_FIFO_INF)

Offset: 080h Size: 32 bits

This register contains the indication of free space in the TX Data FIFO and the used space in the TX Status FIFO.

Bits	Description	Type	Default
31:24	**RESERVED**	RO	-
23:16	**TX Status FIFO Used Space (TXSUSED)** This field indicates the amount of space, in DWORDs, currently used in the TX Status FIFO.	RO	0b
15:0	**TX Data FIFO Free Space (TXFREE)** This field indicates the amount of space, in bytes, available in the TX Data FIFO. The application should never write more than is available, as indicated by this value.	RO	1200h

11.14.7 HOST MAC RX DROPPED FRAMES COUNTER REGISTER (RX_DROP)

Offset: 0A0h Size: 32 bits

This register indicates the number of receive frames that have been dropped by the Host MAC.

Bits	Description	Type	Default
31:0	**RX Dropped Frame Counter (RX_DFC)** This counter is incremented every time a receive frame is dropped by the Host MAC. RX_DFC is cleared on any read of this register. **Note:** The interrupt RXDFH_INT (bit 23 of the Interrupt Status Register (INT_STS)) can be issued when this counter passes through its halfway point (7FFFFFFFh to 80000000h).	RC	00000000h

11.14.8 HOST MAC CSR INTERFACE COMMAND REGISTER (MAC_CSR_CMD)

Offset: 0A4h Size: 32 bits

This read-write register is used to control the read and write operations to/from the Host MAC. This register in used in conjunction with the Host MAC CSR Interface Data Register (MAC_CSR_DATA) to indirectly access the Host MAC CSRs.

Note: The full list of Host MAC CSRs are described in Section 11.15, "Host MAC Control and Status Registers," on page 192.

Bits	Description	Type	Default
31	**Host MAC CSR Busy (HMAC_CSR_BUSY)** When a 1 is written into this bit, the read or write operation is performed to the specified Host MAC CSR. This bit will remain set until the operation is complete. In the case of a read, this indicates that the host can read valid data from the Host MAC CSR Interface Data Register (MAC_CSR_DATA). **Note:** The MAC_CSR_CMD and MAC_CSR_DATA registers must not be modified until this bit is cleared.	R/W SC	0b
30	**R/nW** When set, this bit indicates that the host is requesting a read operation. When clear, the host is performing a write. **0: Host MAC CSR Write Operation** **1: Host MAC CSR Read Operation**	R/W	0b
29:8	**RESERVED**	RO	-
7:0	**CSR Address** The 8-bit value in this field selects which Host MAC CSR will be accessed by the read or write operation. The index of each Host MAC CSR is defined in Section 11.15, "Host MAC Control and Status Registers," on page 192.	R/W	00h

11.14.9 HOST MAC CSR INTERFACE DATA REGISTER (MAC_CSR_DATA)

Offset: 0A8h Size: 32 bits

This read-write register is used in conjunction with the Host MAC CSR Interface Command Register (MAC_CSR_CMD) to indirectly access the Host MAC CSRs.

Note: The full list of Host MAC CSRs are described in Section 11.15, "Host MAC Control and Status Registers," on page 192.

Bits	Description	Type	Default
31:0	**Host MAC CSR Data** This field contains the value read from or written to the Host MAC CSR as specified in the Host MAC CSR Interface Command Register (MAC_CSR_CMD). Upon a read, the value returned depends on the R/nW bit in the MAC_CSR_CMD register. If R/nW is a 1, the data in this register is from the Host MAC. If R/nW is 0, the data is the value that was last written into this register. **Note:** The MAC_CSR_CMD and MAC_CSR_DATA registers must not be modified until the CSR Busy bit is cleared in the MAC_CSR_CMD register.	R/W	00000000h

11.14.10 HOST MAC AUTOMATIC FLOW CONTROL CONFIGURATION REGISTER (AFC_CFG)

Offset: 0ACh Size: 32 bits

This read/write register configures the mechanism that controls the automatic and software-initiated transmission of pause frames and back pressure from the Host MAC to the network. This register is used in conjunction with the Host MAC Flow Control Register (HMAC_FLOW) in the Host MAC CSR space. Pause frames and backpressure are sent to the network to stop it from sending packets to the Host MAC.

Note: The Host MAC will not transmit pause frames or assert back pressure if the transmitter is disabled.

Refer to section Section 11.2, "Flow Control," on page 139 for additional information.

Bits	Description	Type	Default
31:24	**RESERVED**	RO	-
23:16	**Automatic Flow Control High Level (AFC_HI)** This field specifies, in multiples of 64 bytes, the level at which flow control will trigger. When this limit is reached, the chip will apply back pressure or will transmit a pause frame as programmed in bits [3:0] of this register. During full-duplex operation only a single pause frame is transmitted when this level is reached. The pause time transmitted in this frame is programmed in the FCPT field of the Host MAC Flow Control Register (HMAC_FLOW) in the Host MAC CSR space. During half-duplex operation each incoming frame that matches the criteria in bits [3:0] of this register will be jammed for the period set in the BACK_DUR field.	R/W	00h
15:8	**Automatic Flow Control Low Level (AFC_LO)** This field specifies, in multiples of 64 bytes, the level at which a pause frame is transmitted with a pause time setting of zero. When the amount of data in the RX Data FIFO falls below this level the pause frame is transmitted. A pause time value of zero instructs the other transmitting device to immediately resume transmission. The zero time pause frame will only be transmitted if the RX Data FIFO had reached the AFC_HI level and a pause frame was sent. A zero pause time frame is sent whenever automatic flow control in enabled in bits [3:0] of this register. **Note:** When automatic flow control is enabled the AFC_LO setting must always be less than the AFC_HI setting.	R/W	00h
7:4	**Backpressure Duration (BACK_DUR)** When the Host MAC automatically asserts back pressure, it will be asserted for this period of time. In full-duplex mode, this field has no function and is not used. Please refer to Table 11-16, describing Backpressure Duration bit mapping for more information.	R/W	0h
3	**Flow Control on Multicast Frame (FCMULT)** When this bit is set, the Host MAC will assert back pressure when the AFC level is reached and a multicast frame is received. This field has no function in full-duplex mode. **0: Flow Control on Multicast Frame Disabled** **1: Flow Control on Multicast Frame Enabled**	R/W	0b

Bits	Description	Type	Default
2	**Flow Control on Broadcast Frame (FCBRD)** When this bit is set, the Host MAC will assert back pressure when the AFC level is reached and a broadcast frame is received. This field has no function in full-duplex mode. **0: Flow Control on Broadcast Frame Disabled** **1: Flow Control on Broadcast Frame Enabled**	R/W	0b
1	**Flow Control on Address Decode (FCADD)** When this bit is set, the Host MAC will assert back pressure when the AFC level is reached and a frame addressed to the Host MAC is received. This field has no function in full-duplex mode. **0: Flow Control on Address Decode Disabled** **1: Flow Control on Address Decode Enabled**	R/W	0b
0	**Flow Control on Any Frame (FCANY)** When this bit is set, the Host MAC will assert back pressure, or transmit a pause frame when the AFC level is reached and any frame is received. Setting this bit enables full-duplex flow control when the Host MAC is operating in full-duplex mode. When this mode is enabled during half-duplex operation, the Flow Controller does not decode the Host MAC address and will send a JAM upon receipt of a valid preamble (i.e., immediately at the beginning of the next frame after the RX Data FIFO level is reached). When this mode is enabled during full-duplex operation, the Flow Controller will immediately instruct the Host MAC to send a pause frame when the RX Data FIFO level is reached. The MAC will queue the pause frame transmission for the next available window. Setting this bit overrides bits [3:1] of this register.	R/W	0b

TABLE 11-16: BACKPRESSURE DURATION BIT MAPPING

	Backpressure Duration	
[7:4]	**100Mbs Mode**	**10Mbs Mode**
0h	5 us	7.2 us
1h	10 us	12.2 us
2h	15 us	17.2 us
3h	25 us	27.2 us
4h	50 us	52.2 us
5h	100 us	102.2 us
6h	150 us	152.2 us
7h	200 us	202.2 us
8h	250 us	252.2 us
9h	300 us	302.2 us
Ah	350 us	352.2 us
Bh	400 us	402.2 us
Ch	450 us	452.2 us
Dh	500 us	502.2 us
Eh	550 us	552.2 us
Fh	600 us	602.2 us

Note: Backpressure Duration is timed from when the RX FIFO reaches the level set in Automatic Flow Control High Level (AFC_HI), regardless when actual backpressure occurs.

11.14.11 HOST MAC RX LPI TRANSITIONS REGISTER (HMAC_RX_LPI_TRANSITION)

Offset: 0B0h Size: 32 bits

This register indicates the number of times that the RX LPI indication from the PHY changed from de-asserted to asserted.

Bits	Description	Type	Default
31:0	**EEE RX LPI Transitions** Count of total number of times that the RX LPI indication from the PHY changed from de-asserted to asserted. The counter is reset if the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Host MAC Control Register (HMAC_CR) is low.	RO	00000000h

11.14.12 HOST MAC RX LPI TIME REGISTER (HMAC_RX_LPI_TIME)

Offset: 0B4h Size: 32 bits

This register shows the total duration that the PHY has indicated RX LPI.

Bits	Description	Type	Default
31:0	**EEE RX LPI Time** This field shows the total duration, in us, that the PHY has indicated RX LPI. The counter is reset if the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Host MAC Control Register (HMAC_CR) is low.	RO	00000000h

11.14.13 HOST MAC TX LPI TRANSITIONS REGISTER (HMAC_TX_LPI_TRANSITION)

Offset: 0B8h Size: 32 bits

This register indicates the total number of times TX LPI request to the PHY changed from de-asserted to asserted.

Bits	Description	Type	Default
31:0	**EEE TX LPI Transitions** Count of total number of times the TX LPI request to the PHY changed from de-asserted to asserted. The counter is reset if the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Host MAC Control Register (HMAC_CR) is low.	RO	00000000h

11.14.14 HOST MAC TX LPI TIME REGISTER (HMAC_TX_LPI_TIME)

Offset: 0BCh Size: 32 bits

This register shows the total duration that TX LPI request to the PHY has been asserted.

Bits	Description	Type	Default
31:0	**EEE TX LPI Time** This field shows the total duration, in us, that TX LPI request to the PHY has been asserted. The counter is reset if the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit in the Host MAC Control Register (HMAC_CR) is low.	RO	00000000h

11.15 Host MAC Control and Status Registers

This section details the indirectly addressable Host MAC System CSRs. These registers are located in the Host MAC and are accessed indirectly via the system CSRs. Table 11-17 lists Host MAC registers that are accessible through the indexing method using the Host MAC CSR Interface Command Register (MAC_CSR_CMD) and Host MAC CSR Interface Data Register (MAC_CSR_DATA).

The Host MAC registers allow configuration of the various Host MAC parameters including the Host MAC address, flow control, multicast hash table, and wake-up configuration. The Host MAC CSRs also provide serial access to the PHY via the registers HMAC_MII_ACC and HMAC_MII_DATA. These registers allow access to the 10/100 Ethernet PHY registers.

TABLE 11-17: HOST MAC ADDRESSABLE REGISTERS

Address (Indirect)	Register Name (SYMBOL)
00h	Reserved for Future Use (RESERVED)
01h	Host MAC Control Register (HMAC_CR)
02h	Host MAC Address High Register (HMAC_ADDRH)
03h	Host MAC Address Low Register (HMAC_ADDRL)
04h	Host MAC Multicast Hash Table High Register (HMAC_HASHH)
05h	Host MAC Multicast Hash Table Low Register (HMAC_HASHL)
06h	Host MAC MII Access Register (HMAC_MII_ACC)
07h	Host MAC MII Data Register (HMAC_MII_DATA)
08h	Host MAC Flow Control Register (HMAC_FLOW)
09h	Host MAC VLAN1 Tag Register (HMAC_VLAN1)
0Ah	Host MAC VLAN2 Tag Register (HMAC_VLAN2)
0Bh	Host MAC Wake-up Frame Filter Register (HMAC_WUFF)
0Ch	Host MAC Wake-up Control and Status Register (HMAC_WUCSR)
0Dh	Host MAC Checksum Offload Engine Control Register (HMAC_COE_CR)
0Eh	Host MAC EEE Time Wait TX System Register (HMAC_EEE_TW_TX_SYS)
0Fh	Host MAC EEE TX LPI Request Delay Register (HMAC_EEE_TX_LPI_REQ_DELAY)
10h-FFh	Reserved for Future Use (RESERVED)

11.15.1 HOST MAC CONTROL REGISTER (HMAC_CR)

Offset: 01h Size: 32 bits

This read/write register establishes the RX and TX operation modes and controls for address filtering and packet filtering.

Bits	Description	Type	Default
31	**Receive All Mode (RXALL)** When set, all incoming packets will be received and passed on to the address filtering function for processing of the selected filtering mode on the received frame. Address filtering then occurs and is reported in Receive Status. When cleared, only frames that pass Destination Address filtering will be sent to the application.	R/W	0b
30:26	**RESERVED**	RO	-
25	**Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE)** When set, this bit enables EEE operation (both TX LPI and RX LPI)	R/W	Note 7
24	**RESERVED**	RO	0b
23	**Disable Receive Own (RCVOWN)** When set, the Host MAC disables the reception of frames when it is transmitting (TXEN output is asserted). When cleared, the Host MAC receives all packets, including those transmitted by the Host MAC. This bit has no effect when the Full Duplex Mode (FDPX) bit is set.	R/W	0b
22	**RESERVED**	RO	-
21	**Loopback operation Mode (LOOPBK)** Selects the loop back operation modes for the Host MAC. This field is only valid for full duplex mode. In internal loopback mode, the TX frame is received by the internal MII interface, and sent back to the Host MAC without being sent to the network. **0: Normal Operation. Loopback disabled.** **1: Loopback enabled** **Note:** When enabling or disabling the loopback mode it can take up to 10 μs for the mode change to occur. The transmitter and receiver must be stopped and disabled when modifying the LOOPBK bit. The transmitter or receiver should not be enabled within10μs of modifying the LOOPBK bit.	R/W	0b
20	**Full Duplex Mode (FDPX)** When set, the Host MAC operates in Full-Duplex mode, in which it can transmit and receive simultaneously.	R/W	0b
19	**Pass All Multicast (MCPAS)** When set, indicates that all incoming frames with a Multicast destination address (first bit in the destination address field is 1) are received. Incoming frames with physical address (Individual Address/Unicast) destinations are filtered and received only if the address matches the Host MAC Address.	R/W	0b
18	**Promiscuous Mode (PRMS)** When set, indicates that any incoming frame is received regardless of its destination address.	R/W	1b

Bits	Description	Type	Default
17	**Inverse filtering (INVFILT)** When set, the address check function operates in inverse filtering mode. This is valid only during Perfect filtering mode. Refer to Section 11.4.4, "Inverse Filtering," on page 143 for additional information.	R/W	0b
16	**Pass Bad Frames (PASSBAD)** When set, all incoming frames that passed address filtering are received, including runt frames and collided frames. Refer to Section 11.4, "Address Filtering," on page 142 for additional information.	R/W	0b
15	**Hash Only Filtering mode (HO)** When set, the address check Function operates in the Imperfect address filtering mode for both physical and multicast addresses. Refer to Section 11.4.2, "Hash Only Filtering," on page 143 for additional information.	R/W	0b
14	**RESERVED**	RO	-
13	**Hash/Perfect Filtering Mode (HPFILT)** When cleared (0), the device will implement a perfect address filter on incoming frames according the address specified in the Host MAC address registers (Host MAC Address High Register (HMAC_ADDRH) and Host MAC Address Low Register (HMAC_ADDRL)). When set (1), the address check function performs imperfect address filtering of multicast incoming frames according to the hash table specified in the multicast hash table register. If the Hash Only Filtering mode (HO) bit 15 is set, then the physical (IA) addresses are also imperfect filtered. If the Hash Only Filtering mode (HO) bit is cleared, then the IA addresses are perfect address filtered according to the MAC Address register Refer to Section 11.4.3, "Hash Perfect Filtering," on page 143 for additional information.	R/W	0b
12	**RESERVED**	RO	0b
11	**Disable Broadcast Frames (BCAST)** When set, disables the reception of broadcast frames. When cleared, forwards all broadcast frames to the application. **Note:** When wake-up frame detection is enabled via the WUEN bit of the Host MAC Wake-up Control and Status Register (HMAC_WUCSR), a broadcast wake-up frame will wake-up the device despite the state of this bit.	R/W	0b
10	**Disable Retry (DISRTY)** When set, the Host MAC attempts only one transmission. When a collision is seen on the network, the Host MAC ignores the current frame and goes to the next frame and a retry error is reported in the Transmit status. When reset, the Host MAC attempts 16 transmissions before signaling a retry error.	R/W	0b
9	**RESERVED**	RO	-
8	**Automatic Pad Stripping (PADSTR)** When set, the Host MAC strips the pad field on all incoming frames, if the length field is less than 46 bytes. The FCS field is also stripped, since it is computed at the transmitting station based on the data and pad field characters, and is invalid for a received frame that has had the pad characters stripped. Receive frames with a 46-byte or greater length field are passed to the application unmodified (FCS is not stripped). When cleared, the Host MAC passes all incoming frames to the host unmodified.	R/W	0b

<table>
<tr><th>Bits</th><th>Description</th><th>Type</th><th>Default</th></tr>
<tr><td>7:6</td><td>BackOff Limit (BOLMT)
The BOLMT bits allow the user to set the back-off limit in a relaxed or aggressive mode. According to IEEE 802.3, the Host MAC has to wait for a random number [r] of slot-times (see note) after it detects a collision, where:
(eq.1)0 < r < $_2$K
The exponent K is dependent on how many times the current frame to be transmitted has been retried, as follows:
(eq.2)K = min (n, 10) where n is the current number of retries.
If a frame has been retried three times, then K = 3 and r= 8 slot-times maximum. If it has been retried 12 times, then K = 10, and r = 1024 slot-times maximum.
An LFSR (linear feedback shift register) 20-bit counter emulates a 20 bit random number generator, from which r is obtained. Once a collision is detected, the number of the current retry of the current frame is used to obtain K (eq.2). This value of K translates into the number of bits to use from the LFSR counter. If the value of K is 3, the Host MAC takes the value in the first three bits of the LFSR counter and uses it to count down to zero on every slot-time. This effectively causes the Host MAC to wait eight slot-times. To give the user more flexibility, the BOLMT value forces the number of bits to be used from the LFSR counter to a predetermined value as in the table below.
<table>
<tr><th>BOLMT Value</th><th># Bits Used from LFSR Counter</th></tr>
<tr><td>00b</td><td>10</td></tr>
<tr><td>01b</td><td>8</td></tr>
<tr><td>10b</td><td>4</td></tr>
<tr><td>11b</td><td>1</td></tr>
</table>
Thus, if the value of K = 10, the Host MAC will look at the BOLMT if it is 00b, then use the lower ten bits of the LFSR counter for the wait countdown. If the BOLMT is 10b, then it will only use the value in the first four bits for the wait countdown, etc.
Note: Slot-time = 512 bit times. (See IEEE 802.3 Spec., sections 4.2.3.25 and 4.4.2.1)</td><td>R/W</td><td>0b</td></tr>
<tr><td>5</td><td>Deferral Check (DFCHK)
When set, enables the deferral check in the Host MAC. The Host MAC will abort the transmission attempt if it has deferred for more than 24,288 bit times. Deferral starts when the transmitter is ready to transmit, but is prevented from doing so because the CRS is active. Deferral time is not cumulative. If the transmitter defers for 10,000 bit times, then transmits, collides, backs off, and then has to defer again after completion of back-off, the deferral timer resets to 0 and restarts. When this bit is cleared, the deferral check is disabled in the Host MAC and the Host MAC defers indefinitely.</td><td>R/W</td><td>0b</td></tr>
<tr><td>4</td><td>RESERVED</td><td>RO</td><td>-</td></tr>
<tr><td>3</td><td>Transmitter enable (TXEN)
When set, the Host MAC’s transmitter is enabled and it will transmit frames from the buffer.
When cleared, the Host MAC’s transmitter is disabled and will not transmit any frames.</td><td>R/W</td><td>0b</td></tr>
<tr><td>2</td><td>Receiver Enable (RXEN)
When set, the Host MAC’s receiver is enabled and will receive frames.
When cleared, the MAC’s receiver is disabled and will not receive any frames.</td><td>R/W</td><td>0b</td></tr>
</table>

Bits	Description	Type	Default
1:0	**RESERVED**	RO	-

Note 7: The value of this field is determined by the EEE_enable_strap_1 (default 1b).

11.15.2 HOST MAC ADDRESS HIGH REGISTER (HMAC_ADDRH)

Offset: 02h Size: 32 bits

This read/write register contains the upper 16-bits of the physical address of the Host MAC. The contents of this register are optionally loaded from the EEPROM at power-on through the EEPROM Loader if a programmed EEPROM is detected. The least significant byte of this register (bits [7:0]) is loaded from address 05h of the EEPROM. The second byte (bits [15:8]) is loaded from address 06h of the EEPROM. Section 11.9, "Host MAC Address," on page 156 details the byte ordering of the HMAC_ADDRL and HMAC_ADDRH registers with respect to the reception of the Ethernet physical address. Please refer to Section 13.4, "EEPROM Loader," on page 290 for more information on the EEPROM Loader.

This register used to specify the address used for Perfect DA, Magic Packet and Wakeup frames, the unicast destination address for received pause frames, and the source address for transmitted pause frames. This register is not used for packet filtering.

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:0	**Physical Address [47:32]** This field contains the upper 16-bits (47:32) of the Physical Address of the Host MAC. The content of this field is undefined until loaded from the EEPROM at power-on. The host can update the contents of this field after the initialization process has completed.	R/W	FFFFh

11.15.3 HOST MAC ADDRESS LOW REGISTER (HMAC_ADDRL)

Offset: 03h Size: 32 bits

This read/write register contains the lower 32-bits of the physical address of the Host MAC. The contents of this register are optionally loaded from the EEPROM at power-on through the EEPROM Loader if a programmed EEPROM is detected. The least significant byte of this register (bits [7:0]) is loaded from address 01h of the EEPROM. The most significant byte of this register is loaded from address 04h of the EEPROM. Section 11.9, "Host MAC Address," on page 156 details the byte ordering of the HMAC_ADDRL and HMAC_ADDRH registers with respect to the reception of the Ethernet physical address. Please refer to Section 13.4, "EEPROM Loader," on page 290 for more information on the EEPROM Loader.

This register used to specify the address used for Perfect DA, Magic Packet and Wakeup frames, the unicast destination address for received pause frames, and the source address for transmitted pause frames. This register is not used for packet filtering.

BITS	DESCRIPTION	TYPE	DEFAULT
31:0	**Physical Address [31:0]** This field contains the lower 32-bits (31:0) of the Physical Address of the Host MAC. The content of this field is undefined until loaded from the EEPROM at power-on. The host can update the contents of this field after the initialization process has completed.	R/W	FFFFFFFFh

11.15.4 HOST MAC MULTICAST HASH TABLE HIGH REGISTER (HMAC_HASHH)

Offset: 04h Size: 32 bits

The 64-bit Multicast table is used for group address filtering. For hash filtering, the contents of the destination address in the incoming frame is used to index the contents of the Hash table. The most significant bit determines the register to be used (Hi/Low), while the other five bits determine the bit within the register. A value of 00000 selects Bit 0 of the Multicast Hash Table Lo register and a value of 11111 selects the Bit 31 of the Multicast Hash Table Hi register.

If the corresponding bit is 1, then the multicast frame is accepted. Otherwise, it is rejected. If the “Pass All Multicast” (MCPAS) bit of the Host MAC Control Register (HMAC_CR) is set, then all multicast frames are accepted regardless of the multicast hash values.

The Multicast Hash Table High register contains the higher 32 bits of the hash table and the Multicast Hash Table Low register contains the lower 32 bits of the hash table. Refer to Section 11.4, "Address Filtering," on page 142 for more information on address filtering.

Bits	Description	Type	Default
31:0	**Upper 32-bits of the 64-bit Hash Table**	R/W	00000000h

11.15.5 HOST MAC MULTICAST HASH TABLE LOW REGISTER (HMAC_HASHL)

Offset: 05h Size: 32 bits

This read/write register defines the lower 32-bits of the Multicast Hash Table. Please refer to the Host MAC Multicast Hash Table High Register (HMAC_HASHH) and Section 11.4, "Address Filtering," on page 142 for more information.

Bits	Description	Type	Default
31:0	**Lower 32-bits of the 64-bit Hash Table**	R/W	00000000h

11.15.6 HOST MAC MII ACCESS REGISTER (HMAC_MII_ACC)

Offset: 06h Size: 32 bits

This read/write register is used in conjunction with the Host MAC MII Data Register (HMAC_MII_DATA) to access the internal PHY registers. Refer to Section 12.2.18, "PHY Registers" for a list of accessible PHY registers and PHY address information.

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:11	**PHY Address (PHY_ADDR)** This field must be loaded with the PHY address that the MII access is intended for. Refer to Section 12.1.1, "PHY Addressing," on page 210 for additional information on PHY addressing.	R/W	00000b
10:6	**MII Register Index (MIIRINDA)** These bits select the desired MII register in the PHY.	R/W	00000b
5:2	**RESERVED**	RO	-
1	**MII Write (MIIWnR)** Setting this bit tells the PHY that this will be a write operation using the Host MAC MII Data Register (HMAC_MII_DATA). If this bit is cleared, a read operation will occur, packing the data in the Host MAC MII Data Register (HMAC_MII_DATA).	R/W	0b
0	**MII Busy (MIIBZY)** This bit must be polled to determine when the MII register access is complete. This bit must read a logical 0 before writing to this register or the Host MAC MII Data Register (HMAC_MII_DATA). The LAN driver software must set this bit in order for the device to read or write any of the MII PHY registers. During a MII register access, this bit will be set, signifying a read or write access is in progress. The MII data register must be kept valid until the Host MAC clears this bit during a PHY write operation. The MII data register is invalid until the Host MAC has cleared this bit during a PHY read operation.	RO SC	0b

11.15.7 HOST MAC MII DATA REGISTER (HMAC_MII_DATA)

Offset: 07h Size: 32 bits

This read/write register is used in conjunction with the Host MAC MII Access Register (HMAC_MII_ACC) to access the internal PHY registers. This register contains either the data to be written to the PHY register specified in the HMAC_MII_ACC Register, or the read data from the PHY register whose index is specified in the HMAC_MII_ACC Register.

The MII Busy (MIIBZY) bit in the Host MAC MII Access Register (HMAC_MII_ACC) must be cleared when writing to this register.

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:0	**MII Data** This field contains the 16-bit value read from the PHY read operation or the 16-bit data value to be written to the PHY before an MII write operation.	R/W	0000h

11.15.8 HOST MAC FLOW CONTROL REGISTER (HMAC_FLOW)

Offset: 08h Size: 32 bits

This read/write register controls the generation and reception of the Control (Pause command) frames by the Host MAC's flow control block. The control frame fields are selected as specified in the 802.3 Specification and the Pause-Time value from this register is used in the "Pause Time" field of the control frame. In full-duplex mode the FCBSY bit is set until the control frame is completely transferred. The host has to make sure that the FCBSY bit is cleared before writing the register. The Pass Control Frame bit (FCPASS) does not affect the sending of the frames, including Control Frames, to the host. The Flow Control Enable (FCEN) bit enables the receive portion of the Flow Control block.

This register is used in conjunction with the Host MAC Automatic Flow Control Configuration Register (AFC_CFG) in the System CSRs to configure flow control. Software flow control is initiated using the AFC_CFG register.

The Host MAC will not transmit pause frames or assert back pressure if the transmitter is disabled.

Bits	Description	Type	Default
31:16	**Pause Time (FCPT)** This field indicates the value to be used in the PAUSE TIME field in the control frame. This field must be initialized before full-duplex automatic flow control is enabled.	R/W	0000h
15:3	**RESERVED**	RO	-
2	**Pass Control Frames (FCPASS)** When set, the Host MAC sets the packet filter bit in the receive packet status to indicate to the application that a valid pause frame has been received. The application must accept or discard a received frame based on the packet filter control bit. The Host MAC receives, decodes and performs the Pause function when a valid Pause frame is received in Full-Duplex mode and when flow control is enabled (FCE bit set). When this bit is cleared, the Host MAC resets the Packet Filter bit in the Receive packet status. The Host MAC always passes the data of all frames it receives (including flow control frames) to the application. Frames that do not pass address filtering, as well as frames with errors, are passed to the application. The application must discard or retain the received frame's data based on the received frame's STATUS field. Filtering modes (promiscuous mode, for example) take precedence over the FCPASS bit.	R/W	0b
1	**Flow Control Enable (FCEN)** When set, enables the Host MAC flow control function. The Host MAC decodes all incoming frames for control frames; if it receives a valid control frame (PAUSE command), it disables the transmitter for a specified time (Decoded pause time x slot time). When this bit is cleared, the Host MAC flow control function is disabled; the MAC does not decode frames for control frames. **Note:** Flow Control is applicable when the Host MAC is set in full duplex mode. In half-duplex mode, this bit enables the backpressure function to control the flow of received frames to the Host MAC.	R/W	0b

Bits	Description	Type	Default
0	**Flow Control Busy (FCBSY)** In full-duplex mode, this bit indicates that the Host MAC is in the process of sending a PAUSE control frame. This bit is set by the automatic flow control function. During the transmission of the control frame, this bit continues to be set, signifying that a frame transmission is in progress. After the PAUSE control frame's transmission is complete, the Host MAC resets the bit to 0. The host software should read a logical 0 from this bit before writing to the Host MAC Flow Control (HMAC_FLOW) register.	R/W SC	0b

11.15.9 HOST MAC VLAN1 TAG REGISTER (HMAC_VLAN1)

Offset: 09h Size: 32 bits

This read/write register contains the VLAN tag field to identify VLAN1 frames. When a VLAN1 frame is detected, the legal frame length is increased from 1518 bytes to 1522 bytes. Refer to Section 11.3, "Virtual Local Area Network (VLAN) Support," on page 140 for additional information.

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:0	**VLAN1 Tag Identifier (VTI1)** This field contains the VLAN Tag used to identify VLAN1 frames. This field is compared with the 13th and 14th bytes of the incoming frames for VLAN1 frame detection. **Note:** If used, this register is typically set to the standard VLAN value of 8100h.	R/W	FFFFh

11.15.10 HOST MAC VLAN2 TAG REGISTER (HMAC_VLAN2)

Offset: 0Ah Size: 32 bits

This read/write register contains the VLAN tag field to identify VLAN2 frames. When a VLAN2 frame is detected, the legal frame length is increased from 1518 bytes to 1538 bytes. Refer to Section 11.3, "Virtual Local Area Network (VLAN) Support," on page 140 for additional information.

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:0	**VLAN2 Tag Identifier (VTI2)** This field contains the VLAN Tag used to identify VLAN2 frames. This field is compared with the 13th and 14th bytes of the incoming frames for VLAN2 frame detection. **Note:** If used, this register is typically set to the standard VLAN value of 8100h. If both VLAN1 and VLAN2 Tag Identifiers are used, they should be unique. If both are set to the same value, VLAN1 is given higher precedence and the maximum legal frame length is set to 1522.	R/W	FFFFh

11.15.11 HOST MAC WAKE-UP FRAME FILTER REGISTER (HMAC_WUFF)

Offset: 0Bh Size: 32 bits

This write-only register is used to configure the wake-up frame filter. Refer to Section 11.6.3, "Wake-up Frame Detection," on page 145 for additional information.

Bits	Description	Type	Default
31:0	**Wake-Up Frame Filter (WFF)** The Wake-up frame filter is configured through this register using an indexing mechanism. After power-on reset, digital reset, or Host MAC reset, the Host MAC loads the first value written to this location to the first DWORD in the Wake-up frame filter (filter 0 byte mask 0). The second value written to this location is loaded to the second DWORD in the wake-up frame filter (filter 0 byte mask 1) and so on. Once all 40 DWORDs have been written, the internal pointer will once again point to the first entry and the filter entries can be modified in the same manner. Similarly, 40 DWORDs can be read sequentially to obtain the values stored in the WFF. Please refer to Section 11.6.3, "Wake-up Frame Detection," on page 145 for further information. **Note:** This register should be read and written using 40 consecutive DWORD operations. Failure to read or write the entire contents of the WFF may cause the internal read/write pointers to be left in a position other than pointing to the first entry. A mechanism for resetting the internal pointers to the beginning of the WFF is available via the WFF Pointer Reset (WFF_PTR_RST) bit of the Host MAC Wake-up Control and Status Register (HMAC_WUCSR). This mechanism enables the application program to re-synchronize with the internal WFF pointers if it has not previously read/written the complete contents of the WFF.	R/W	-

11.15.12 HOST MAC WAKE-UP CONTROL AND STATUS REGISTER (HMAC_WUCSR)

Offset: 0Ch Size: 32 bits

This read/write register contains data and control settings pertaining to the Host MAC's remote wake-up status and capabilities. It is used in conjunction with the Host MAC Wake-up Frame Filter Register (HMAC_WUFF) to fully configure the wake-up frame filter. Refer to Section 11.6.3, "Wake-up Frame Detection," on page 145 for additional information.

Bits	Description	Type	Default
31	**WFF Pointer Reset (WFF_PTR_RST)** This self-clearing bit resets the Wakeup Frame Filter (WFF) internal read and write pointers to the beginning of the WFF.	SC	0b
30:10	**RESERVED**	RO	-
9	**Global Unicast Enable (GUE)** When set, the Host MAC wakes up from power-saving mode on receipt of a global unicast frame. This is accomplished by enabling global unicasts as a wakeup frame qualifier. A global unicast frame has the MAC Address [0] bits set to 0. **Note:** The Wake-Up Frame Enable (WUEN) bit of this register must also be set to enable wakeup.	R/W	0b
8	**WoL Wait for Sleep (WOL_WAIT_SLEEP)** When set, the WoL functions are not active until the device has entered a sleep state. When clear, WoL functions are active immediately.	R/W	0b
7	**Perfect DA Frame Received (PFDA_FR)** The MAC sets this bit upon receiving a valid frame with a destination address that matches the physical address.	R/WC	0b
6	**Remote Wake-Up Frame Received (WUFR)** The Host MAC sets this bit upon receiving a valid Remote Wake-up frame.	R/WC	0b
5	**Magic Packet Received (MPR)** The Host MAC sets this bit upon receiving a valid Magic Packet	R/WC	0b
4	**Broadcast Frame Received (BCAST_FR)** The MAC sets this bit upon receiving a valid broadcast frame.	R/WC	0b
3	**Perfect DA Wakeup Enable (PFDA_EN)** When set, remote wakeup mode is enabled and the MAC is capable of waking up on receipt of a frame with a destination address that matches the physical address of the device. The physical address is stored in the Host MAC Address High Register (HMAC_ADDRH) and Host MAC Address Low Register (HMAC_ADDRL).	R/W	0b
2	**Wake-Up Frame Enable (WUEN)** When set, Remote Wake-Up mode is enabled and the Host MAC is capable of detecting wake-up frames as programmed in the Host MAC Wake-up Frame Filter Register (HMAC_WUFF).	R/W	0b
1	**Magic Packet Enable (MPEN)** When set, Magic Packet Wake-up mode is enabled.	R/W	0b
0	**Broadcast Wakeup Enable (BCST_EN)** When set, remote wakeup mode is enabled and the MAC is capable of waking up from a broadcast frame.	R/W	0b

11.15.13 HOST MAC CHECKSUM OFFLOAD ENGINE CONTROL REGISTER (HMAC_COE_CR)

Offset: 0Dh Size: 32 bits

This register controls the RX and TX checksum offload engines.

Bits	Description	Type	Default
31:17	**RESERVED**	RO	-
16	**TX Checksum Offload Engine Enable (TX_COE_EN)** TX_COE_EN may only be changed if the TX path is disabled. If it is desired to change this value during run time, it is safe to do so only after the Host MAC is disabled and the TLI is empty. **0 = The TXCOE is bypassed** **1 = The TXCOE is enabled**	R/W	0b
15:2	**RESERVED**	RO	-
1	**RX Checksum Offload Engine Mode (RX_COE_MODE)** This register indicates whether the COE will check for VLAN tags or a SNAP header prior to beginning its checksum calculation. In its default mode, the calculation will always begin 14 bytes into the frame. RX_COE_MODE may only be changed if the RX path is disabled. If it is desired to change this value during run time, it is safe to do so only after the Host MAC is disabled and the TLI is empty. **0 = Begin checksum calculation after first 14 bytes of Ethernet Frame** **1 = Begin checksum calculation at start of L3 packet by adjusting for VLAN tags and/or SNAP header.**	R/W	0b
0	**RX Checksum Offload Engine Enable (RX_COE_EN)** RX_COE_EN may only be changed if the RX path is disabled. If it is desired to change this value during run time, it is safe to do so only after the Host MAC is disabled and the TLI is empty. **0 = The RXCOE is bypassed** **1 = The RXCOE is enabled**	R/W	0b

11.15.14 HOST MAC EEE TIME WAIT TX SYSTEM REGISTER (HMAC_EEE_TW_TX_SYS)

Offset: 0Eh Size: 32 bits

This register configures the time to wait before starting packet transmission after TX LPI removal.

BITS	DESCRIPTION	TYPE	DEFAULT
31:24	**RESERVED**	RO	-
23:0	**TX Delay After TX LPI Removal** This field configures the time to wait, in uS, before starting packet transmission after TX LPI removal. Software should only change this field when the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit is cleared. **Note:** In order to meet the IEEE 802.3 specified requirement, the minimum value of this field should be 00001Eh.	R/W	00001Eh

11.15.15 HOST MAC EEE TX LPI REQUEST DELAY REGISTER (HMAC_EEE_TX_LPI_REQ_DELAY)

Offset: 0Fh Size: 32 bits

This register contains the amount of time, in microseconds, the Host MAC must wait after the TX FIFO is empty before invoking the LPI protocol.

Note: The actual time can be up to 1 us longer than specified.

Note: A value of zero is valid and will cause no delay to occur.

Note: If the TX FIFO becomes non-empty, the timer is restarted

Bits	Description	Type	Default
31:0	**EEE TX LPI Request Delay** This field contains the time to wait, in microseconds, before invoking the LPI protocol. Software should only change this field when the Host MAC Energy Efficient Ethernet (HMAC_EEE_ENABLE) bit is cleared.	R/W	00000000h

12.0 ETHERNET PHY

12.1 Functional Overview

The device contains a PHY which connects to the Host MAC.

The PHY complies with the IEEE 802.3 Physical Layer for Twisted Pair Ethernet and can be configured for full/half duplex 100 Mbps (100BASE-TX / 100BASE-FX) or 10 Mbps (10BASE-T) Ethernet operation. All PHY registers follow the IEEE 802.3 (clause 22.2.4) specified MII management register set and are fully configurable.

12.1.1 PHY ADDRESSING

The default address for the PHY is fixed to 1.

In addition, the address for the PHY can be changed via the PHY Address (PHYADD) field in the PHY Special Modes Register (PHY_SPECIAL_MODES).

12.2 PHY

The device integrates two IEEE 802.3 PHY functions. The PHY can be configured for either 100 Mbps copper (100BASE-TX), 100 Mbps fiber (100BASE-FX) or 10 Mbps copper (10BASE-T) Ethernet operation and includes Auto-Negotiation and HP Auto-MDIX.

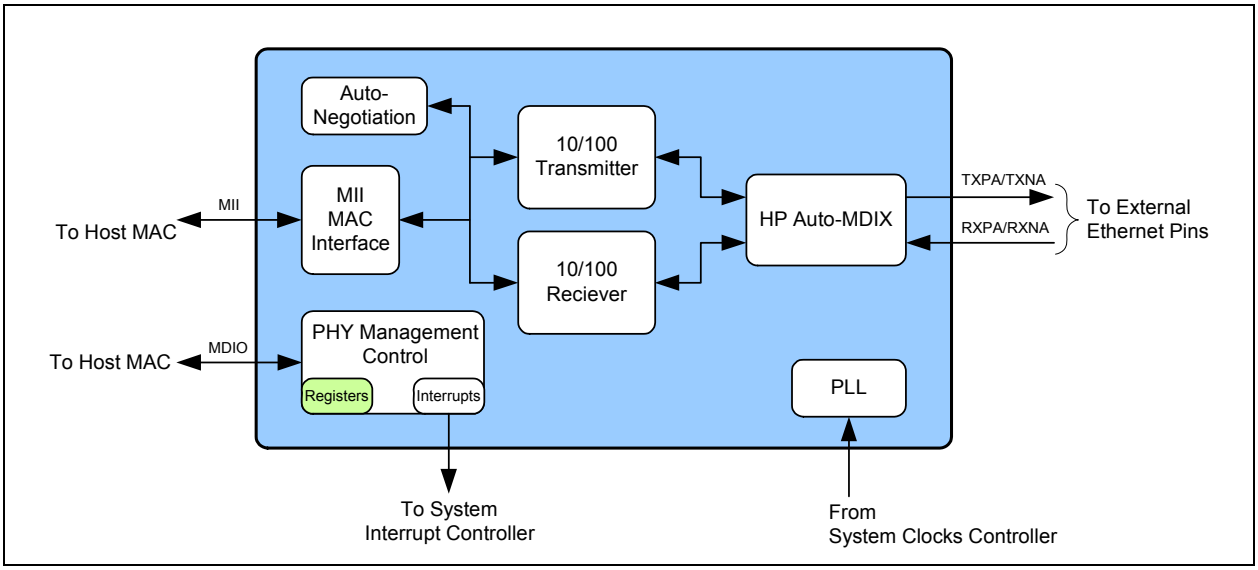
12.2.1 FUNCTIONAL DESCRIPTION

Functionally, the PHY can be divided into the following sections:

- 100BASE-TX Transmit and 100BASE-TX Receive
- 10BASE-T Transmit and 10BASE-T Receive
- Auto-Negotiation
- HP Auto-MDIX
- PHY Management Control and PHY Interrupts
- PHY Power-Down Modes and Energy Efficient Ethernet
- Resets
- Link Integrity Test
- Cable Diagnostics
- Loopback Operation
- 100BASE-FX Far End Fault Indication

A block diagram of the main components of the PHY can be seen in Figure 12-1.

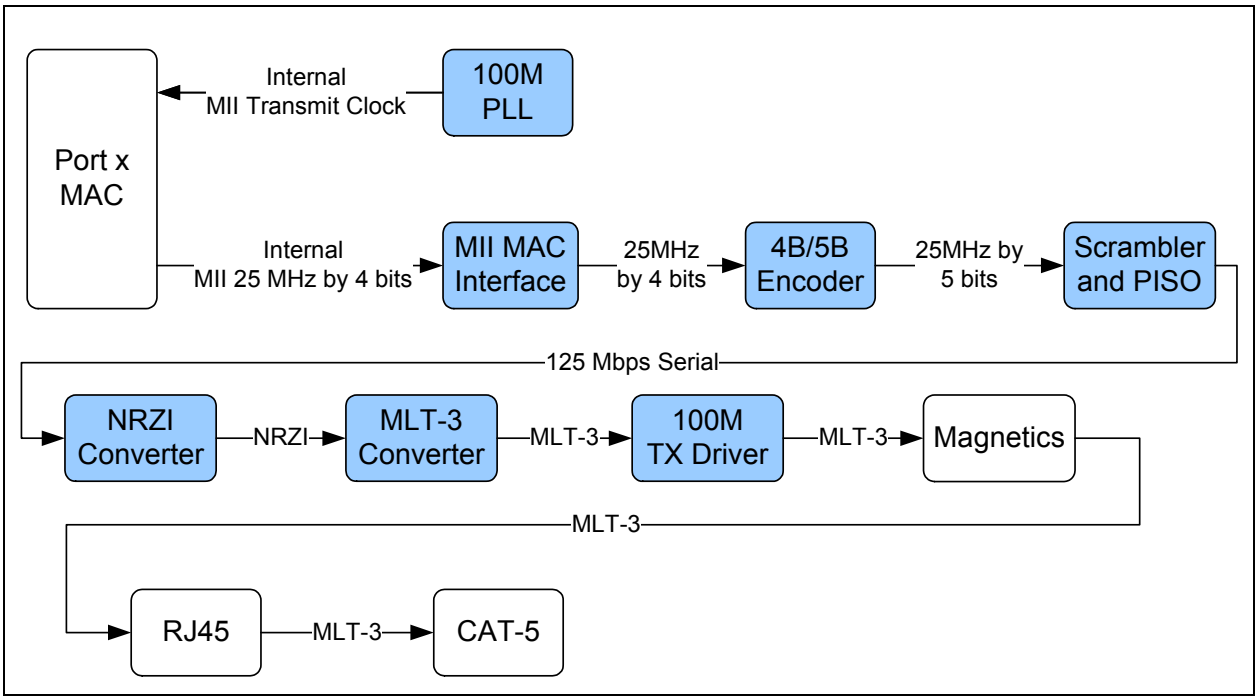
FIGURE 12-1: PHY BLOCK DIAGRAM



12.2.2 100BASE-TX TRANSMIT

The 100BASE-TX transmit data path is shown in Figure 12-2. Shaded blocks are those which are internal to the PHY. Each major block is explained in the following sections.

FIGURE 12-2: 100BASE-TX TRANSMIT DATA PATH



12.2.2.1 100BASE-TX Transmit Data Across the Internal MII Interface

For a transmission, the Host MAC drives the transmit data onto the internal MII TXD bus and asserts the internal MII TXEN to indicate valid data. The data is in the form of 4-bit wide 25 MHz data.

12.2.2.2 4B/5B Encoder

The transmit data passes from the MII block to the 4B/5B Encoder. This block encodes the data from 4-bit nibbles to 5-bit symbols (known as “code-groups”) according to Table 12-1. Each 4-bit data-nibble is mapped to 16 of the 32 possible code-groups. The remaining 16 code-groups are either used for control information or are not valid.

The first 16 code-groups are referred to by the hexadecimal values of their corresponding data nibbles, 0 through F. The remaining code-groups are given letter designations with slashes on either side. For example, an IDLE code-group is /I/, a transmit error code-group is /H/, etc.

TABLE 12-1: 4B/5B CODE TABLE

Code Group	Sym	Receiver Interpretation			Transmitter Interpretation		
11110	0	0	0000	DATA	0	0000	DATA
01001	1	1	0001		1	0001	
10100	2	2	0010		2	0010	
10101	3	3	0011		3	0011	
01010	4	4	0100		4	0100	
01011	5	5	0101		5	0101	
01110	6	6	0110		6	0110	
01111	7	7	0111		7	0111	
10010	8	8	1000		8	1000	
10011	9	9	1001		9	1001	
10110	A	A	1010		A	1010	
10111	B	B	1011		B	1011	
11010	C	C	1100		C	1100	
11011	D	D	1101		D	1101	
11100	E	E	1110		E	1110	
11101	F	F	1111		F	1111	
11111	/I/	IDLE			Sent after /T/R/ until the MII Transmitter Enable signal (TXEN) is received		
11000	/J/	First nibble of SSD, translated to “0101” following IDLE, else MII Receive Error (RXER)			Sent for rising MII Transmitter Enable signal (TXEN)		
10001	/K/	Second nibble of SSD, translated to “0101” following J, else MII Receive Error (RXER)			Sent for rising MII Transmitter Enable signal (TXEN)		
01101	/T/	First nibble of ESD, causes de-assertion of CRS if followed by /R/, else assertion of MII Receive Error (RXER)			Sent for falling MII Transmitter Enable signal (TXEN)		
00111	/R/	Second nibble of ESD, causes de-assertion of CRS if following /T/, else assertion of MII Receive Error (RXER)			Sent for falling MII Transmitter Enable signal (TXEN)		
00100	/H/	Transmit Error Symbol			Sent for rising MII Transmit Error (TXER)		

TABLE 12-1: 4B/5B CODE TABLE (CONTINUED)

Code Group	Sym	Receiver Interpretation	Transmitter Interpretation
00110	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
11001	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
00000	/P/	SLEEP, Indicates to receiver that the transmitter will be going to LPI	Sent due to LPI. Used to tell receiver before transmitter goes to LPI. Also used for refresh cycles during LPI.
00001	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
00010	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
00011	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
00101	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
01000	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
01100	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID
10000	/V/	INVALID, MII Receive Error (RXER) if during MII Receive Data Valid (RXDV)	INVALID

12.2.2.3 Scrambler and PISO

Repeated data patterns (especially the IDLE code-group) can have power spectral densities with large narrow-band peaks. Scrambling the data helps eliminate these peaks and spread the signal power more uniformly over the entire channel bandwidth. This uniform spectral density is required by FCC regulations to prevent excessive EMI from being radiated by the physical wiring.

The seed for the scrambler is generated from the PHY address, ensuring that each PHY will have its own scrambler sequence. For more information on PHY addressing, refer to Section 12.1.1, "PHY Addressing".

The scrambler also performs the Parallel In Serial Out conversion (PISO) of the data.

12.2.2.4 NRZI and MLT-3 Encoding

The scrambler block passes the 5-bit wide parallel data to the NRZI converter where it becomes a serial 125MHz NRZI data stream. The NRZI is then encoded to MLT-3. MLT-3 is a tri-level code where a change in the logic level represents a code bit “1” and the logic output remaining at the same level represents a code bit “0”.

12.2.2.5 100M Transmit Driver

The MLT-3 data is then passed to the analog transmitter, which drives the differential MLT-3 signal on output pins TXPx and TXNx, to the twisted pair media across a 1:1 ratio isolation transformer. The 10BASE-T and 100BASE-TX signals pass through the same transformer so that common “magnetics” can be used for both. The transmitter drives into the 100 Ω impedance of the CAT-5 cable. Cable termination and impedance matching require external components.

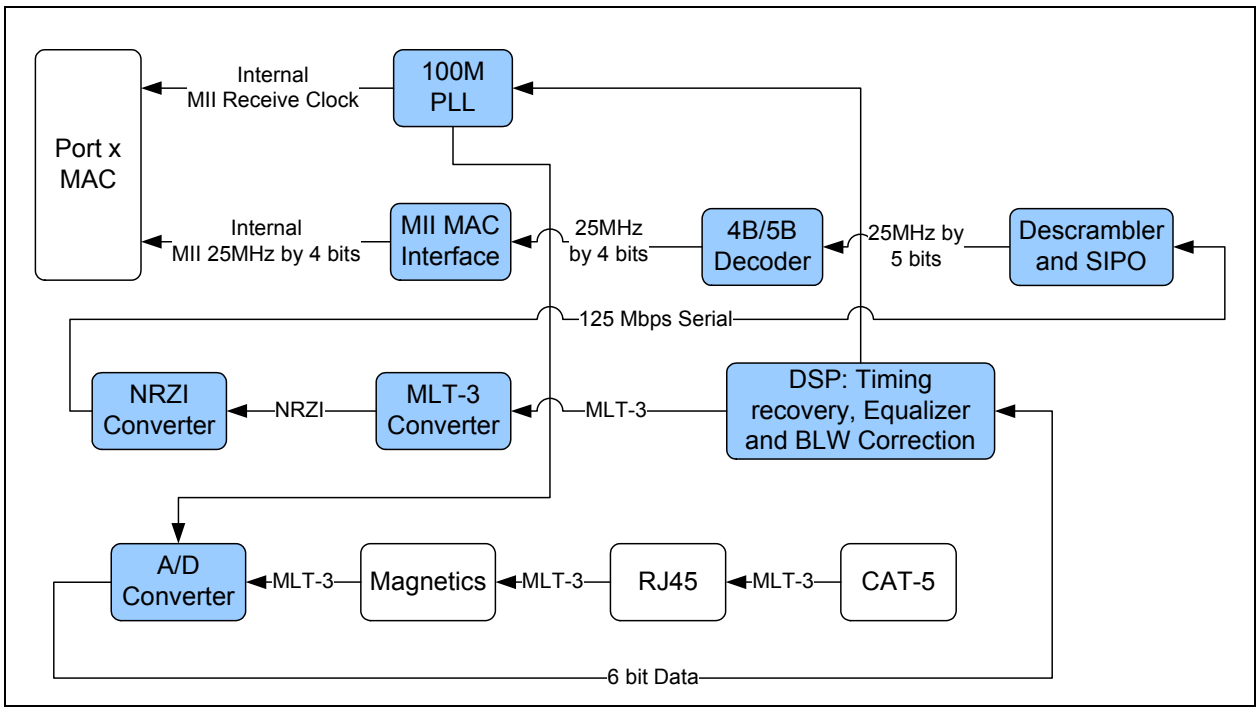
12.2.2.6 100M Phase Lock Loop (PLL)

The 100M PLL locks onto the reference clock and generates the 125 MHz clock used to drive the 125 MHz logic and the 100BASE-TX Transmitter.

12.2.3 100BASE-TX RECEIVE

The 100BASE-TX receive data path is shown in Figure 12-3. Shaded blocks are those which are internal to the PHY. Each major block is explained in the following sections.

FIGURE 12-3: 100BASE-TX RECEIVE DATA PATH



12.2.3.1 100M Receive Input

The MLT-3 data from the cable is fed into the PHY on inputs RXPx and RXNx via a 1:1 ratio transformer. The ADC samples the incoming differential signal at a rate of 125M samples per second. Using a 64-level quantizer, 6 digital bits are generated to represent each sample. The DSP adjusts the gain of the ADC according to the observed signal levels such that the full dynamic range of the ADC can be used.

12.2.3.2 Equalizer, BLW Correction and Clock/Data Recovery

The 6 bits from the ADC are fed into the DSP block. The equalizer in the DSP section compensates for phase and amplitude distortion caused by the physical channel consisting of magnetics, connectors, and CAT- 5 cable. The equalizer can restore the signal for any good-quality CAT-5 cable between 1m and 100m.

If the DC content of the signal is such that the low-frequency components fall below the low frequency pole of the isolation transformer, then the droop characteristics of the transformer will become significant and Baseline Wander (BLW) on the received signal will result. To prevent corruption of the received data, the PHY corrects for BLW and can receive the ANSI X3.263-1995 FDDI TP-PMD defined “killer packet” with no bit errors.

The 100M PLL generates multiple phases of the 125MHz clock. A multiplexer, controlled by the timing unit of the DSP, selects the optimum phase for sampling the data. This is used as the received recovered clock. This clock is used to extract the serial data from the received signal.

12.2.3.3 NRZI and MLT-3 Decoding

The DSP generates the MLT-3 recovered levels that are fed to the MLT-3 converter. The MLT-3 is then converted to an NRZI data stream.

12.2.3.4 Descrambler

The descrambler performs an inverse function to the scrambler in the transmitter and also performs the Serial In Parallel Out (SIPO) conversion of the data.

During reception of IDLE (/I/) symbols. the descrambler synchronizes its descrambler key to the incoming stream. Once synchronization is achieved, the descrambler locks on this key and is able to descramble incoming data.

Special logic in the descrambler ensures synchronization with the remote transceiver by searching for IDLE symbols within a window of 4000 bytes (40 us). This window ensures that a maximum packet size of 1514 bytes, allowed by the IEEE 802.3 standard, can be received with no interference. If no IDLE-symbols are detected within this time-period, receive operation is aborted and the descrambler re-starts the synchronization process.

The de-scrambled signal is then aligned into 5-bit code-groups by recognizing the /J/K/ Start-of-Stream Delimiter (SSD) pair at the start of a packet. Once the code-word alignment is determined, it is stored and utilized until the next start of frame.

12.2.3.5 5B/4B Decoding

The 5-bit code-groups are translated into 4-bit data nibbles according to the 4B/5B table. The translated data is presented on the internal MII RXD[3:0] signal lines. The SSD, /J/K/, is translated to “0101 0101” as the first 2 nibbles of the MAC preamble. Reception of the SSD causes the transceiver to assert the receive data valid signal, indicating that valid data is available on the RXD bus. Successive valid code-groups are translated to data nibbles. Reception of either the End of Stream Delimiter (ESD) consisting of the /T/R/ symbols, or at least two /I/ symbols causes the transceiver to de-assert carrier sense and receive data valid signal.

Note: These symbols are not translated into data.

12.2.3.6 Receive Data Valid Signal

The internal MII’s Receive Data Valid signal (RXDV) indicates that recovered and decoded nibbles are being presented on the RXD[3:0] outputs synchronous to RXCLK. RXDV becomes active after the /J/K/ delimiter has been recognized and RXD is aligned to nibble boundaries. It remains active until either the /T/R/ delimiter is recognized or link test indicates failure or SIGDET becomes false.

RXDV is asserted when the first nibble of translated /J/K/ is ready for transfer over the Media Independent Interface.

12.2.3.7 Receiver Errors

During a frame, unexpected code-groups are considered receive errors. Expected code groups are the DATA set (0 through F), and the /T/R/ (ESD) symbol pair. When a receive error occurs, the internal MII’s RXER signal is asserted and arbitrary data is driven onto the internal MII’s RXD[3:0] lines. Should an error be detected during the time that the /J/K/ delimiter is being decoded (bad SSD error), RXER is asserted true and the value 1110b is driven onto the RXD[3:0] lines. Note that the internal MII’s data valid signal (RXDV) is not yet asserted when the bad SSD occurs.

12.2.3.8 100M Receive Data Across the Internal MII Interface

For reception, the 4-bit data nibbles are sent to the MII MAC Interface block. These data nibbles are clocked to the controller at a rate of 25 MHz. RXCLK is the output clock for the internal MII bus. It is recovered from the received data to clock the RXD bus. If there is no received signal, it is derived from the system reference clock.

12.2.4 10BASE-T TRANSMIT

The 10BASE-T transmitter receives 4-bit nibbles from the internal MII at a rate of 2.5 MHz and converts them to a 10 Mbps serial data stream. The data stream is then Manchester-encoded and sent to the analog transmitter, which drives a signal onto the twisted pair via the external magnetics.

10BASE-T transmissions use the following blocks:

- MII (digital)
- TX 10M (digital)
- 10M Transmitter (analog)
- 10M PLL (analog)

12.2.4.1 10M Transmit Data Across the Internal MII Interface

For a transmission, the Host MAC drives the transmit data onto the internal MII TXD bus and asserts the internal MII TXEN to indicate valid data. The data is in the form of 4-bit wide 2.5 MHz data.

In half-duplex mode the transceiver loops back the transmitted data, on the receive path. This does not confuse the MAC/Controller since the COL signal is not asserted during this time. The transceiver also supports the SQE (Heartbeat) signal.

12.2.4.2 Manchester Encoding

The 4-bit wide data is sent to the 10M TX block. The nibbles are converted to a 10Mbps serial NRZI data stream. The 10M PLL produces a 20MHz clock. This is used to Manchester encode the NRZ data stream. When no data is being transmitted (internal MII TXEN is low), the 10M TX Driver block outputs Normal Link Pulses (NLPs) to maintain communications with the remote link partner.

12.2.4.3 10M Transmit Drivers

The Manchester encoded data is sent to the analog transmitter where it is shaped and filtered before being driven out as a differential signal across the TXPx and TXNx outputs.

12.2.5 10BASE-T RECEIVE

The 10BASE-T receiver gets the Manchester-encoded analog signal from the cable via the magnetics. It recovers the receive clock from the signal and uses this clock to recover the NRZI data stream. This 10M serial data is converted to 4-bit data nibbles which are passed to the controller across the internal MII at a rate of 2.5MHz.

10BASE-T reception uses the following blocks:

- Filter and SQUELCH (analog)
- 10M PLL (analog)
- RX 10M (digital)
- MII (digital)

12.2.5.1 10M Receive Input and Squelch

The Manchester signal from the cable is fed into the transceiver (on inputs RXPx and RXNx) via 1:1 ratio magnetics. It is first filtered to reduce any out-of-band noise. It then passes through a SQUELCH circuit. The SQUELCH is a set of amplitude and timing comparators that normally reject differential voltage levels below 300mV and detect and recognize differential voltages above 585mV.

12.2.5.2 Manchester Decoding

The output of the SQUELCH goes to the 10M RX block where it is validated as Manchester encoded data. The polarity of the signal is also checked. If the polarity is reversed (local RXP is connected to RXN of the remote partner and vice versa), the condition is identified and corrected. The reversed condition is indicated by the 10Base-T Polarity State (XPOL) bit in PHY Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND). The 10M PLL is locked onto the received Manchester signal, from which the 20MHz clock is generated. Using this clock, the Manchester encoded data is extracted and converted to a 10MHz NRZI data stream. It is then converted from serial to 4-bit wide parallel data.

The RX10M block also detects valid 10BASE-T IDLE signals - Normal Link Pulses (NLPs) - to maintain the link.

12.2.5.3 10M Receive Data Across the Internal MII Interface

For reception, the 4-bit data nibbles are sent to the MII MAC Interface block. These data nibbles are clocked to the controller at a rate of 2.5 MHz.

12.2.5.4 Jabber Detection

Jabber is a condition in which a station transmits for a period of time longer than the maximum permissible packet length, usually due to a fault condition, which results in holding the internal MII TXEN input for a long period. Special logic is used to detect the jabber state and abort the transmission to the line, within 45 ms. Once TXEN is deasserted, the logic resets the jabber condition.

The Jabber Detect bit in the PHY Basic Status Register (PHY_BASIC_STATUS) indicates that a jabber condition was detected.

12.2.6 AUTO-NEGOTIATION

The purpose of the Auto-Negotiation function is to automatically configure the transceiver to the optimum link parameters based on the capabilities of its link partner. Auto-Negotiation is a mechanism for exchanging configuration information between two link-partners and automatically selecting the highest performance mode of operation supported by both sides. Auto-Negotiation is fully defined in clause 28 of the IEEE 802.3 specification and is enabled by setting the Auto-Negotiation Enable (PHY_AN) of the PHY Basic Control Register (PHY_BASIC_CONTROL).

Note: Auto-Negotiation is not used for 100BASE-FX mode.

The advertised capabilities of the PHY are stored in the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV). The PHY contains the ability to advertise 100BASE-TX and 10BASE-T in both full or half-duplex modes. Besides the connection speed, the PHY can advertise remote fault indication and symmetric or asymmetric pause flow control as defined in the IEEE 802.3 specification. The transceiver supports “Next Page” capability which is used to negotiate Energy Efficient Ethernet functionality as well as to support software controlled pages. Many of the default advertised capabilities of the PHY are determined via configuration straps as shown in Section 12.2.18.5, "PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV)," on page 238. Refer to Section 7.0, "Configuration Straps," on page 54 for additional details on how to use the device configuration straps.

Once Auto-Negotiation has completed, information about the resolved link and the results of the negotiation process are reflected in the Speed Indication bits in the PHY Special Control/Status Register (PHY_SPECIAL_CONTROL_STATUS), as well as the PHY Auto-Negotiation Link Partner Base Page Ability Register (PHY_AN_LP_BASE_ABILITY). The Auto-Negotiation protocol is a purely physical layer activity and proceeds independently of the MAC controller.

The following blocks are activated during an Auto-Negotiation session:

- Auto-Negotiation (digital)
- 100M ADC (analog)
- 100M PLL (analog)
- 100M equalizer/BLW/clock recovery (DSP)
- 10M SQUELCH (analog)
- 10M PLL (analog)
- 10M Transmitter (analog)

When enabled, Auto-Negotiation is started by the occurrence of any of the following events:

- Power-On Reset (POR)
- Hardware reset (RST#)
- PHY Software reset (via Reset Control Register (RESET_CTL), or bit 15 of the PHY Basic Control Register (PHY_BASIC_CONTROL))
- PHY Power-down reset (Section 12.2.10, "PHY Power-Down Modes," on page 222)
- PHY Link status down (bit 2 of the PHY Basic Status Register (PHY_BASIC_STATUS) is cleared)
- Setting the PHY Basic Control Register (PHY_BASIC_CONTROL), bit 9 high (auto-neg restart)
- Digital Reset (via bit 0 of the Reset Control Register (RESET_CTL))
- Issuing an EEPROM Loader RELOAD command (Section 13.4, "EEPROM Loader," on page 290) via EEPROM Loader run sequence

Note: Refer to Section 6.2, "Resets," on page 38 for information on these and other system resets.

On detection of one of these events, the transceiver begins Auto-Negotiation by transmitting bursts of Fast Link Pulses (FLP). These are bursts of link pulses from the 10M TX Driver. They are shaped as Normal Link Pulses and can pass uncorrupted down CAT-3 or CAT-5 cable. A Fast Link Pulse Burst consists of up to 33 pulses. The 17 odd-numbered pulses, which are always present, frame the FLP burst. The 16 even-numbered pulses, which may be present or absent, contain the data word being transmitted. Presence of a data pulse represents a “1”, while absence represents a “0”.

The data transmitted by an FLP burst is known as a “Link Code Word.” These are defined fully in IEEE 802.3 clause 28. In summary, the transceiver advertises 802.3 compliance in its selector field (the first 5 bits of the Link Code Word). It advertises its technology ability according to the bits set in the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV).

There are 4 possible matches of the technology abilities. In the order of priority these are:

- 100M Full Duplex (Highest priority)
- 100M Half Duplex
- 10M Full Duplex
- 10M Half Duplex (Lowest priority)

If the full capabilities of the transceiver are advertised (100M, full-duplex), and if the link partner is capable of 10M and 100M, then Auto-Negotiation selects 100M as the highest performance mode. If the link partner is capable of half and full-duplex modes, then Auto-Negotiation selects full-duplex as the highest performance mode.

Once a capability match has been determined, the link code words are repeated with the acknowledge bit set. Any difference in the main content of the link code words at this time will cause Auto-Negotiation to re-start. Auto-Negotiation will also re-start if not all of the required FLP bursts are received.

Writing the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV) bits [8:5] allows software control of the capabilities advertised by the transceiver. Writing the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV) does not automatically re-start Auto-Negotiation. The Restart Auto-Negotiation (PHY_RST_AN) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) must be set before the new abilities will be advertised. Auto-Negotiation can also be disabled via software by clearing the Auto-Negotiation Enable (PHY_AN) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL).

12.2.6.1 Pause Flow Control

The Host MAC is capable of generating and receiving pause flow control frames per the IEEE 802.3 specification. The PHY's advertised pause flow control abilities are set via the Asymmetric Pause and Symmetric Pause bits of the PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV). This allows the PHY to advertise its flow control abilities and Auto-Negotiate the flow control settings with its link partner. The default values of these bits are determined via configuration straps as defined in Section 12.2.18.5, "PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV)," on page 238.

12.2.6.2 Parallel Detection

If the device is connected to a device lacking the ability to Auto-Negotiate (i.e. no FLPs are detected), it is able to determine the speed of the link based on either 100M MLT-3 symbols or 10M Normal Link Pulses. In this case the link is presumed to be half-duplex per the IEEE 802.3 standard. This ability is known as "Parallel Detection." This feature ensures interoperability with legacy link partners. If a link is formed via parallel detection, then the Link Partner Auto-Negotiation Able bit of the PHY Auto-Negotiation Expansion Register (PHY_AN_EXP) is cleared to indicate that the link partner is not capable of Auto-Negotiation. If a fault occurs during parallel detection, the Parallel Detection Fault bit of the PHY Auto-Negotiation Expansion Register (PHY_AN_EXP) is set.

The PHY Auto-Negotiation Link Partner Base Page Ability Register (PHY_AN_LP_BASE_ABILITY) is used to store the Link Partner Ability information, which is coded in the received FLPs. If the link partner is not Auto-Negotiation capable, then this register is updated after completion of parallel detection to reflect the speed capability of the link partner.

12.2.6.3 Restarting Auto-Negotiation

Auto-Negotiation can be re-started at any time by setting the Restart Auto-Negotiation (PHY_RST_AN) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL). Auto-Negotiation will also re-start if the link is broken at any time. A broken link is caused by signal loss. This may occur because of a cable break, or because of an interruption in the signal transmitted by the Link Partner. Auto-Negotiation resumes in an attempt to determine the new link configuration.

If the management entity re-starts Auto-Negotiation by setting the Restart Auto-Negotiation (PHY_RST_AN) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL), the device will respond by stopping all transmission/receiving operations. Once the internal break_link_time is completed in the Auto-Negotiation state-machine (approximately 1200ms), Auto-Negotiation will re-start. In this case, the link partner will have also dropped the link due to lack of a received signal, so it too will resume Auto-Negotiation.

Auto-Negotiation is also restarted after the EEPROM Loader updates the straps.

12.2.6.4 Disabling Auto-Negotiation

Auto-Negotiation can be disabled by clearing the Auto-Negotiation Enable (PHY_AN) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL). The transceiver will then force its speed of operation to reflect the information in the PHY Basic Control Register (PHY_BASIC_CONTROL) (Speed Select LSB (PHY_SPEED_SEL_LSB) and Duplex Mode (PHY_DUPLEX)). These bits are ignored when Auto-Negotiation is enabled.

12.2.6.5 Half Vs. Full-Duplex

Half-duplex operation relies on the CSMA/CD (Carrier Sense Multiple Access / Collision Detect) protocol to handle network traffic and collisions. In this mode, the carrier sense signal, CRS, responds to both transmit and receive activity. If data is received while the transceiver is transmitting, a collision results.

In full-duplex mode, the transceiver is able to transmit and receive data simultaneously. In this mode, CRS responds only to receive activity. The CSMA/CD protocol does not apply and collision detection is disabled.

12.2.7 HP AUTO-MDIX

HP Auto-MDIX facilitates the use of CAT-3 (10 BASE-T) or CAT-5 (100 BASE-T) media UTP interconnect cable without consideration of interface wiring scheme. If a user plugs in either a direct connect LAN cable or a cross-over patch cable, as shown in Figure 12-4, the transceiver is capable of configuring the TXPx/TXNx and RXPx/RXNx twisted pair pins for correct transceiver operation.

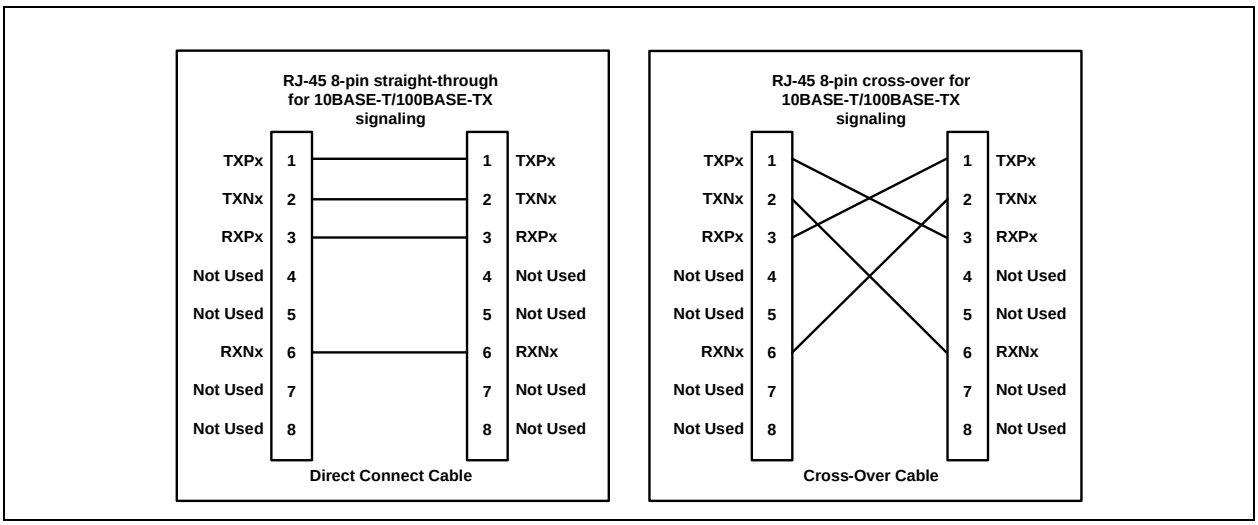
Note:	Auto-MDIX is not used for 100BASE-FX mode.

The internal logic of the device detects the TX and RX pins of the connecting device. Since the RX and TX line pairs are interchangeable, special PCB design considerations are needed to accommodate the symmetrical magnetics and termination of an Auto-MDIX design.

The Auto-MDIX function is enabled using the auto_mdix_strap_1 configuration strap. Manual selection of the cross-over can be set using the manual_mdix_strap_1 configuration strap. Software based control of the Auto-MDIX function may be performed using the Auto-MDIX Control (AMDIXCTRL) bit of the PHY Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND). When AMDIXCTRL is set to 1, the Auto-MDIX capability is determined by the Auto-MDIX Enable (AMDIXEN) and Auto-MDIX State (AMDIXSTATE) bits of the PHY Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND).

Note:	When operating in 10BASE-T or 100BASE-TX manual modes, the Auto-MDIX crossover time can be extended via the Extend Manual 10/100 Auto-MDIX Crossover Time bit of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG). Refer to Section 12.2.18.12, on page 247 for additional information. When Energy Detect Power-Down is enabled, the Auto-MDIX crossover time can be extended via the EDPD Extend Crossover bit of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG). Refer to Section 12.2.18.12, on page 247 for additional information

FIGURE 12-4: DIRECT CABLE CONNECTION VS. CROSS-OVER CABLE CONNECTION



12.2.8 PHY MANAGEMENT CONTROL

The PHY Management Control block is responsible for the management functions of the PHY, including register access and interrupt generation. A Serial Management Interface (SMI) is used to support registers as required by the IEEE 802.3 (Clause 22), as well as the vendor specific registers allowed by the specification. The SMI interface consists of

the MII Management Data (MDIO) signal and the MII Management Clock (MDC) signal. These signals allow access to all PHY registers. Refer to Section 12.2.18, "PHY Registers," on page 230 for a list of all supported registers and register descriptions. Non-supported registers will be read as FFFFh.

12.2.9 PHY INTERRUPTS

The PHY contains the ability to generate various interrupt events. Reading the PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE) shows the source of the interrupt. The PHY Interrupt Mask Register (PHY_INTERRUPT_MASK) enables or disables each PHY interrupt.

The PHY Management Control block aggregates the enabled interrupts status into an internal signal which is sent to the System Interrupt Controller and is reflected via the PHY Interrupt Event (PHY_INT) bit of the Interrupt Status Register (INT_STS). For more information on the device interrupts, refer to Section 8.0, "System Interrupts," on page 62.

The PHY interrupt system provides two modes, a Primary interrupt mode and an Alternative interrupt mode. Both modes will assert the internal interrupt signal sent to the System Interrupt Controller when the corresponding mask bit is set. These modes differ only in how they de-assert the internal interrupt signal. These modes are detailed in the following subsections.

Note: The Primary interrupt mode is the default interrupt mode after a power-up or hard reset. The Alternative interrupt mode requires setup after a power-up or hard reset.

12.2.9.1 Primary Interrupt Mode

The Primary interrupt mode is the default interrupt mode. The Primary interrupt mode is always selected after power-up or hard reset. In this mode, to enable an interrupt, set the corresponding mask bit in the PHY Interrupt Mask Register (PHY_INTERRUPT_MASK) (see Table 12-2). When the event to assert an interrupt is true, the internal interrupt signal will be asserted. When the corresponding event to de-assert the interrupt is true, the internal interrupt signal will be de-asserted.

TABLE 12-2: INTERRUPT MANAGEMENT TABLE

Mask	Interrupt Source Flag		Interrupt Source		Event to Assert interrupt	Event to De-assert interrupt
30.9	29.9	Link Up	LINKSTAT See Note 1	Link Status	Rising LINKSTAT	Falling LINKSAT or Reading register 29
30.7	29.7	ENERGYON	17.1	ENERGYON	Rising 17.1 (Note 3)	Falling 17.1 or Reading register 29
30.6	29.6	Auto-Negotiation complete	1.5	Auto-Negotiate Complete	Rising 1.5	Falling 1.5 or Reading register 29

TABLE 12-2: INTERRUPT MANAGEMENT TABLE (CONTINUED)

30.5	29.5	Remote Fault Detected	1.4	Remote Fault	Rising 1.4	Falling 1.4, or Reading register 1 or Reading register 29
30.4	29.4	Link Down	1.2	Link Status	Falling 1.2	Reading register 1 or Reading register 29
30.3	29.3	Auto-Negotiation LP Acknowledge	5.14	Acknowledge	Rising 5.14	Falling 5.14 or Reading register 29
30.2	29.2	Parallel Detection Fault	6.4	Parallel Detection Fault	Rising 6.4	Falling 6.4 or Reading register 6, or Reading register 29, or Re-Auto Negotiate or Link down
30.1	29.1	Auto-Negotiation Page Received	6.1	Page Received	Rising 6.1	Falling 6.1 or Reading register 6, or Reading register 29, or Re-Auto Negotiate, or Link down.

Note 1: LINKSTAT is the internal link status and is not directly available in any register bit.

Note 2: If the mask bit is enabled and the internal interrupt signal has been de-asserted while ENERGYON is still high, the internal interrupt signal will assert for 256 ms, approximately one second after ENERGYON goes low when the Cable is unplugged. To prevent an unexpected assertion of the internal interrupt signal, the ENERGYON interrupt mask should always be cleared as part of the ENERGYON interrupt service routine.

Note: The Energy On (ENERGYON) bit in the PHY Mode Control/Status Register (PHY_MODE_CONTROL_STATUS) is defaulted to a ‘1’ at the start of the signal acquisition process, therefore the INT7 bit in the PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE) will also read as a ‘1’ at power-up. If no signal is present, then both Energy On (ENERGYON) and INT7 will clear within a few milliseconds.

12.2.9.2 Alternate Interrupt Mode

The Alternate interrupt mode is enabled by setting the ALTINT bit of the PHY Mode Control/Status Register (PHY_MODE_CONTROL_STATUS) to “1”. In this mode, to enable an interrupt, set the corresponding bit of the in the PHY Interrupt Mask Register (PHY_INTERRUPT_MASK) (see Table 12-3). To clear an interrupt, clear the interrupt source and write a ‘1’ to the corresponding Interrupt Source Flag. Writing a ‘1’ to the Interrupt Source Flag will cause the state machine to check the Interrupt Source to determine if the Interrupt Source Flag should clear or stay as a ‘1’. If the con-

dition to de-assert is true, then the Interrupt Source Flag is cleared and the internal interrupt signal is also deasserted. If the condition to de-assert is false, then the Interrupt Source Flag remains set, and the internal interrupt signal remains asserted.

TABLE 12-3: ALTERNATIVE INTERRUPT MODE MANAGEMENT TABLE

Mask	Interrupt Source Flag		Interrupt Source		Event to Assert interrupt	Condition to De-assert	Bit to Clear interrupt
30.9	29.9	Link Up	LINKSTAT See Note 3	Link Status	Rising LINK-STAT	LINKSTAT low	29.9
30.7	29.7	ENERGYON	17.1	ENERGYON	Rising 17.1	17.1 low	29.7
30.6	29.6	Auto-Negotia-tion complete	1.5	Auto-Negoti-ate Com-plete	Rising 1.5	1.5 low	29.6
30.5	29.5	Remote Fault Detected	1.4	Remote Fault	Rising 1.4	1.4 low	29.5
30.4	29.4	Link Down	1.2	Link Status	Falling 1.2	1.2 high	29.4
30.3	29.3	Auto-Negotia-tion LP Acknowl-edge	5.14	Acknowl-edge	Rising 5.14	5.14 low	29.3
30.2	29.2	Parallel Detec-tion Fault	6.4	Parallel Detection Fault	Rising 6.4	6.4 low	29.2
30.1	29.1	Auto-Negotia-tion Page Received	6.1	Page Received	Rising 6.1	6.1 low	29.1

Note 3: LINKSTAT is the internal link status and is not directly available in any register bit.

Note: The Energy On (ENERGYON) bit in the PHY Mode Control/Status Register (PHY_MODE_CONTROL_STATUS) is defaulted to a ‘1’ at the start of the signal acquisition process, therefore the INT7 bit in the PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE) will also read as a ‘1’ at power-up. If no signal is present, then both Energy On (ENERGYON) and INT7 will clear within a few milliseconds.

12.2.10 PHY POWER-DOWN MODES

There are two PHY power-down modes: General Power-Down Mode and Energy Detect Power-Down Mode. These modes are described in the following subsections.

Note: For more information on the various power management features of the device, refer to Section 6.3, "Power Management," on page 44.

The PHY power-down modes do not reload or reset the PHY registers.

12.2.10.1 General Power-Down

This power-down mode is controlled by the Power Down (PHY_PWR_DWN) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL). In this mode the entire transceiver, except the PHY management control interface, is powered down. The transceiver will remain in this power-down state as long as the Power Down (PHY_PWR_DWN) bit is set. When the Power Down (PHY_PWR_DWN) bit is cleared, the transceiver powers up and is automatically reset.

12.2.10.2 Energy Detect Power-Down

This power-down mode is enabled by setting the Energy Detect Power-Down (EDPWRDOWN) bit of the PHY Mode Control/Status Register (PHY_MODE_CONTROL_STATUS). In this mode, when no energy is present on the line, the entire transceiver is powered down (except for the PHY management control interface, the SQUELCH circuit and the ENERGYON logic). The ENERGYON logic is used to detect the presence of valid energy from 100BASE-TX, 10BASE-T, or Auto-Negotiation signals.

In this mode, when the Energy On (ENERGYON) bit in the PHY Mode Control/Status Register (PHY_MODE_CONTROL_STATUS) signal is low, the transceiver is powered down and nothing is transmitted. When energy is received, via link pulses or packets, the Energy On (ENERGYON) bit goes high, and the transceiver powers up. The transceiver automatically resets itself into the state prior to power-down, and asserts the INT7 bit of the PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE). The first and possibly second packet to activate ENERGYON may be lost.

When the Energy Detect Power-Down (EDPWRDOWN) bit of the PHY Mode Control/Status Register (PHY_MODE_-CONTROL_STATUS) is low, energy detect power-down is disabled.

When in EDPD mode, the device's NLP characteristics may be modified. The device can be configured to transmit NLPs in EDPD via the EDPD TX NLP Enable bit of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG). When enabled, the TX NLP time interval is configurable via the EDPD TX NLP Interval Timer Select field of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG). When in EDPD mode, the device can also be configured to wake on the reception of one or two NLPs. Setting the EDPD RX Single NLP Wake Enable bit of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG) will enable the device to wake on reception of a single NLP. If the EDPD RX Single NLP Wake Enable bit is cleared, the maximum interval for detecting reception of two NLPs to wake from EDPD is configurable via the EDPD RX NLP Max Interval Detect Select field of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG).

The energy detect power down feature is part of the broader power management features of the device and can be used to trigger the power management event output pin (**PME**) or general interrupt request pin (**IRQ**). This is accomplished by enabling the energy detect power-down feature of the PHY as described above, and setting the energy detect enable of the Power Management Control Register (PMT_CTRL). Refer to Power Management for additional information.

12.2.11 ENERGY EFFICIENT ETHERNET

The PHY supports IEEE 802.3az Energy Efficient Ethernet (EEE). The EEE functionality is enabled/disabled via the PHY Energy Efficient Ethernet Enable (PHYEEEEN) bit of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG). Energy Efficient Ethernet is enabled or disabled by default via the EEE_enable_strap_1 configuration strap. In order for EEE to be utilized, the following conditions must be met:

- EEE functionality must be enabled via the PHY Energy Efficient Ethernet Enable (PHYEEEEN) bit of the PHY-EDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG)
- The 100BASE-TX EEE bit of the MMD PHY EEE Advertisement Register (PHY_EEE_ADV) must be set
- The MAC and link-partner must support and be configured for EEE operation
- The device and link-partner must link in 100BASE-TX full-duplex mode

The value of the PHY Energy Efficient Ethernet Enable (PHYEEEEN) bit affects the default values of the following register bits:

- 100BASE-TX EEE bit of the MMD PHY EEE Capability Register (PHY_EEE_CAP)
- 100BASE-TX EEE bit of the MMD PHY EEE Advertisement Register (PHY_EEE_ADV)

Note: Energy Efficient Ethernet is not used for 100BASE-FX mode.

12.2.12 RESETS

In addition to the chip-level hardware reset (**RST#**) and Power-On Reset (POR), the PHY supports three block specific resets. These are discussed in the following sections. For detailed information on all device resets and the reset sequence refer to Section 6.2, "Resets," on page 38.

Note:	Only a hardware reset (**RST#**) or Power-On Reset (POR) will automatically reload the configuration strap values into the PHY registers. The Digital Reset (DIGITAL_RST) bit in the Reset Control Register (RESET_CTL) does not reset the PHY. The Digital Reset (DIGITAL_RST) bit will cause the EEPROM Loader to reload the configuration strap values into the PHY registers and to reset all other PHY registers to their default values. An EEPROM RELOAD command via the EEPROM Command Register (E2P_CMD) also has the same effect. For all other PHY resets, PHY registers will need to be manually configured via software.

12.2.12.1 PHY Software Reset via RESET_CTL

The PHY can be reset via the Reset Control Register (RESET_CTL). This bit is self clearing after approximately 102 us. This reset does not reload the configuration strap values into the PHY registers.

12.2.12.2 PHY Software Reset via PHY_BASIC_CTRL

The PHY can also be reset by setting the Soft Reset (PHY_SRST) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL). This bit is self clearing and will return to 0 after the reset is complete. This reset does not reload the configuration strap values into the PHY registers.

12.2.12.3 PHY Power-Down Reset

After the PHY has returned from a power-down state, a reset of the PHY is automatically generated. The PHY power-down modes do not reload or reset the PHY registers. Refer to Section 12.2.10, "PHY Power-Down Modes," on page 222 for additional information.

12.2.13 LINK INTEGRITY TEST

The device performs the link integrity test as outlined in the IEEE 802.3u (clause 24-15) Link Monitor state diagram. The link status is multiplexed with the 10 Mbps link status to form the Link Status bit in the PHY Basic Status Register (PHY_BASIC_STATUS) and to drive the LINK LED functions.

The DSP indicates a valid MLT-3 waveform present on the RXPx and RXNx signals as defined by the ANSI X3.263 TP-PMD standard, to the Link Monitor state-machine, using the internal DATA_VALID signal. When DATA_VALID is asserted, the control logic moves into a Link-Ready state and waits for an enable from the auto-negotiation block. When received, the Link-Up state is entered, and the Transmit and Receive logic blocks become active. Should auto-negotiation be disabled, the link integrity logic moves immediately to the Link-Up state when the DATA_VALID is asserted.

To allow the line to stabilize, the link integrity logic will wait a minimum of 330 ms from the time DATA_VALID is asserted until the Link-Ready state is entered. Should the DATA_VALID input be negated at any time, this logic will immediately negate the Link signal and enter the Link-Down state.

When the 10/100 digital block is in 10BASE-T mode, the link status is derived from the 10BASE-T receiver logic.

12.2.14 CABLE DIAGNOSTICS

The PHY provides cable diagnostics which allow for open/short and length detection of the Ethernet cable. The cable diagnostics consist of two primary modes of operation:

- Time Domain Reflectometry (TDR) Cable Diagnostics
  TDR cable diagnostics enable the detection of open or shorted cabling on the TX or RX pair, as well as cable length estimation to the open/short fault.
- Matched Cable Diagnostics
  Matched cable diagnostics enable cable length estimation on 100 Mbps-linked cables.

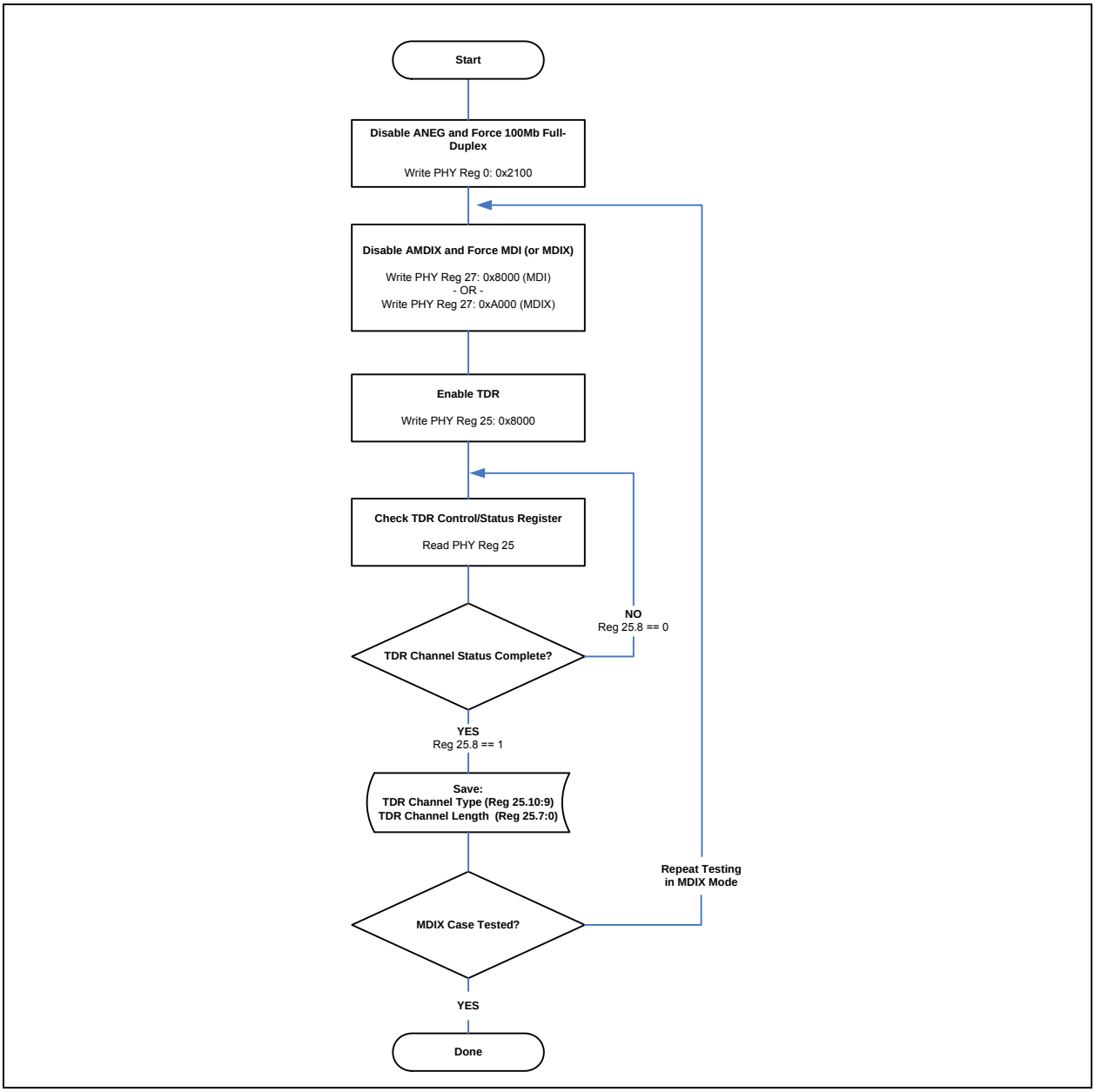
Refer to the following sub-sections for details on proper operation of each cable diagnostics mode.

Note:	Cable diagnostics are not used for 100BASE-FX mode.

12.2.14.1 Time Domain Reflectometry (TDR) Cable Diagnostics

The PHY provides TDR cable diagnostics which enable the detection of open or shorted cabling on the TX or RX pair, as well as cable length estimation to the open/short fault. To utilize the TDR cable diagnostics, Auto-MDIX and Auto Negotiation must be disabled, and the PHY must be forced to 100 Mbps full-duplex mode. These actions must be performed before setting the TDR Enable bit in the PHY TDR Control/Status Register (PHY_TDR_CONTROL_STAT). With Auto-MDIX disabled, the TDR will test the TX or RX pair selected by register bit 27.13 (Auto-MDIX State (AMDIX-STATE)). Proper cable testing should include a test of each pair. TDR cable diagnostics is not appropriate for 100BASE-FX mode. When TDR testing is complete, prior register settings may be restored. Figure 12-5 provides a flow diagram of proper TDR usage.

FIGURE 12-5: TDR USAGE FLOW DIAGRAM



The TDR operates by transmitting pulses on the selected twisted pair within the Ethernet cable (TX in MDI mode, RX in MDIX mode). If the pair being tested is open or shorted, the resulting impedance discontinuity results in a reflected signal that can be detected by the PHY. The PHY measures the time between the transmitted signal and received reflection and indicates the results in the TDR Channel Length field of the PHY TDR Control/Status Register (PHY_TDR_CONTROL_STAT). The TDR Channel Length field indicates the "electrical" length of the cable, and can be multiplied by the appropriate propagation constant in Table 12-4 to determine the approximate physical distance to the fault.

Note: The TDR function is typically used when the link is inoperable. However, an active link will drop when operating the TDR.

Since the TDR relies on the reflected signal of an improperly terminated cable, there are several factors that can affect the accuracy of the physical length estimate. These include:

1. ***Cable Type (CAT 5, CAT5e, CAT6):*** The electrical length of each cable type is slightly different due to the twists-per-meter of the internal signal pairs and differences in signal propagation speeds. If the cable type is known, the length estimate can be calculated more accurately by using the propagation constant appropriate for the cable type (see Table 12-4). In many real-world applications the cable type is unknown, or may be a mix of different cable types and lengths. In this case, use the propagation constant for the "unknown" cable type.
2. ***TX and RX Pair:*** For each cable type, the EIA standards specify different twist rates (twists-per-meter) for each signal pair within the Ethernet cable. This results in different measurements for the RX and TX pair.
3. ***Actual Cable Length:*** The difference between the estimated cable length and actual cable length grows as the physical cable length increases, with the most accurate results at less than approximately 100 m.
4. ***Open/Short Case:*** The Open and Shorted cases will return different TDR Channel Length values (electrical lengths) for the same physical distance to the fault. Compensation for this is achieved by using different propagation constants to calculate the physical length of the cable.

   For the Open case, the estimated distance to the fault can be calculated as follows:

   Distance to Open fault in meters ≅ TDR Channel Length * P_{OPEN}
   Where: P_{OPEN} is the propagation constant selected from Table 12-4

   For the Shorted case, the estimated distance to the fault can be calculated as follows:

   Distance to Open fault in meters ≅ TDR Channel Length * P_{SHORT}
   Where: P_{SHORT} is the propagation constant selected from Table 12-4

TABLE 12-4: TDR PROPAGATION CONSTANTS

TDR Propagation Constant	Cable Type			
	Unknown	CAT 6	CAT 5E	CAT 5
P_{OPEN}	0.769	0.745	0.76	0.85
P_{SHORT}	0.793	0.759	0.788	0.873

The typical cable length measurement margin of error for Open and Shorted cases is dependent on the selected cable type and the distance of the open/short from the device. Table 12-5 and Table 12-6 detail the typical measurement error for Open and Shorted cases, respectively.

TABLE 12-5: TYPICAL MEASUREMENT ERROR FOR OPEN CABLE (+/- METERS)

Physical Distance to Fault	Selected Propagation Constant			
	P_{OPEN} = Unknown	P_{OPEN} = CAT 6	P_{OPEN} = CAT 5E	P_{OPEN} = CAT 5
CAT 6 Cable, 0-100 m	9	6		
CAT 5E Cable, 0-100 m	5		5	
CAT 5 Cable, 0-100 m	13			3

TABLE 12-5: TYPICAL MEASUREMENT ERROR FOR OPEN CABLE (+/- METERS)

CAT 6 Cable, 101-160 m	14	6		
CAT 5E Cable, 101-160 m	8		6	
CAT 5 Cable, 101-160 m	20			6

TABLE 12-6: TYPICAL MEASUREMENT ERROR FOR SHORTED CABLE (+/- METERS)

PHYSICAL DISTANCE TO FAULT	SELECTED PROPAGATION CONSTANT			
	P_{SHORT} = Unknown	P_{SHORT} = CAT 6	P_{SHORT} = CAT 5E	P_{SHORT} = CAT 5
CAT 6 Cable, 0-100 m	8	5		
CAT 5E Cable, 0-100 m	5		5	
CAT 5 Cable, 0-100 m	11			2
CAT 6 Cable, 101-160 m	14	6		
CAT 5E Cable, 101-160 m	7		6	
CAT 5 Cable, 101-160 m	11			3

12.2.14.2 Matched Cable Diagnostics

Matched cable diagnostics enable cable length estimation on 100 Mbps-linked cables of up to 120 meters. If there is an active 100 Mb link, the approximate distance to the link partner can be estimated using the PHY Cable Length Register (PHY_CABLE_LEN). If the cable is properly terminated, but there is no active 100 Mb link (the link partner is disabled, nonfunctional, the link is at 10 Mb, etc.), the cable length cannot be estimated and the PHY Cable Length Register (PHY_CABLE_LEN) should be ignored. The estimated distance to the link partner can be determined via the Cable Length (CBLN) field of the PHY Cable Length Register (PHY_CABLE_LEN) using the lookup table provided in Table 12-7. The typical cable length measurement margin of error for a matched cable case is +/- 20 m. The matched cable length margin of error is consistent for all cable types from 0 to 120 m.

TABLE 12-7: MATCH CASE ESTIMATED CABLE LENGTH (CBLN) LOOKUP

CBLN Field Value	Estimated Cable Length
0 - 3	0
4	6
5	17
6	27
7	38
8	49
9	59
10	70
11	81
12	91

TABLE 12-7: MATCH CASE ESTIMATED CABLE LENGTH (CBLN) LOOKUP

13	102
14	113
15	123

Note: For a properly terminated cable (Match case), there is no reflected signal. In this case, the TDR Channel Length field is invalid and should be ignored.

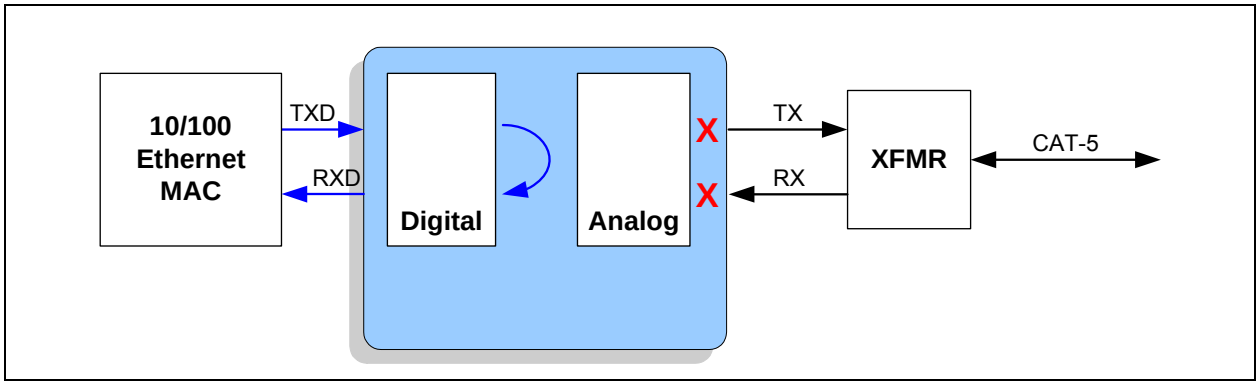
12.2.15 LOOPBACK OPERATION

The PHY may be configured for near-end loopback and connector loopback. These loopback modes are detailed in the following subsections.

12.2.15.1 Near-end Loopback

Near-end loopback mode sends the digital transmit data back out the receive data signals for testing purposes, as indicated by the blue arrows in Figure 12-6. The near-end loopback mode is enabled by setting the Loopback (PHY_LOOPBACK) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) to “1”. A large percentage of the digital circuitry is operational in near-end loopback mode because data is routed through the PCS and PMA layers into the PMD sublayer before it is looped back. The COL signal will be inactive in this mode, unless Collision Test Mode (PHY_COL_TEST) is enabled in the PHY Basic Control Register (PHY_BASIC_CONTROL). The transmitters are powered down regardless of the state of the internal MII TXEN signal.

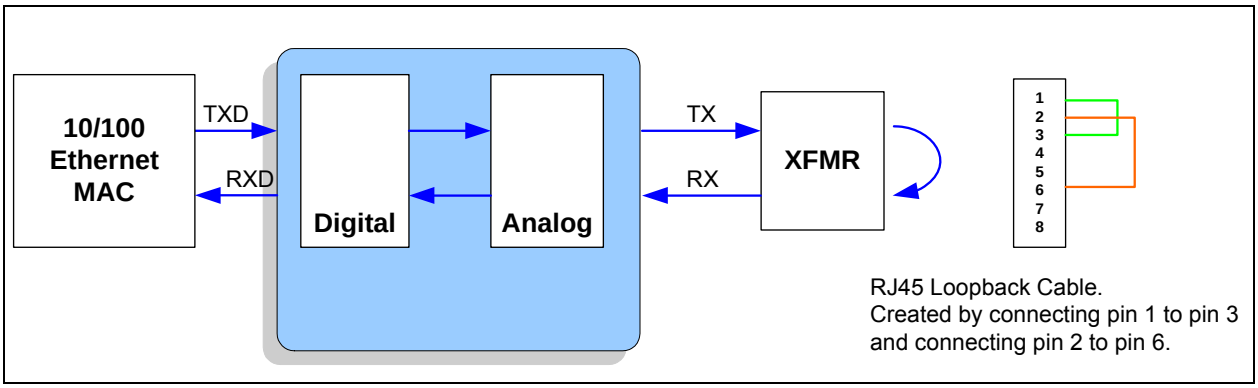
FIGURE 12-6: NEAR-END LOOPBACK BLOCK DIAGRAM



12.2.15.2 Connector Loopback

The device maintains reliable transmission over very short cables and can be tested in a connector loopback as shown in Figure 12-7. An RJ45 loopback cable can be used to route the transmit signals from the output of the transformer back to the receiver inputs. The loopback works at both 10 and 100 Mbps.

FIGURE 12-7: CONNECTION LOOPBACK BLOCK DIAGRAM



12.2.16 100BASE-FX OPERATION

When set for 100BASE-FX operation, the scrambler and MTL-3 blocks are disable and the analog RX and TX pins are changed to differential LVPECL pins and connect through external terminations to the external Fiber transceiver. The differential LVPECL pins support a signal voltage range compatible with SFF (LVPECL) and SFP (reduced LVPECL) type transceivers.

While in 100BASE-FX operation, the quality of the receive signal is provided by the external transceiver as either an open-drain, CMOS level, Loss of Signal (SFP) or a LVPECL Signal Detect (SFF).

12.2.16.1 100BASE-FX Far End Fault Indication

Since Auto-Negotiation is not specified for 100BASE-FX, its Remote Fault capability is unavailable. Instead, 100BASE-FX provides an optional Far-End Fault function.

When no signal is being received, the Far-End Fault feature transmits a special Far-End Fault Indication to its far-end peer. The Far-End Fault Indication is sent only when a physical error condition is sensed on the receive channel.

The Far-End Fault Indication is comprised of three or more repeating cycles, each of 84 ONEs followed by a single ZERO. This signal is sent in-band and is readily detectable but is constructed so as to not satisfy the 100BASE-X carrier sense criterion.

Far-End Fault is implemented through the Far-End Fault Generate, Far-End Fault Detect, and the Link Monitor processes. The Far-End Fault Generate process is responsible for sensing a receive channel failure (signal_status=OFF) and transmitting the Far-End Fault Indication in response. The transmission of the Far-End Fault Indication may start or stop at any time depending only on signal_status. The Far-End Fault Detect process continuously monitors the RX process for the Far-End Fault Indication. Detection of the Far-End Fault Indication disables the station by causing the Link Monitor process to de-assert link_status, which in turn causes the station to source IDLEs.

Far-End Fault is enabled by default while in 100BASE-FX mode via the Far End Fault Indication Enable (FEFI_EN) of the PHY Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND).

12.2.16.2 100BASE-FX Enable and LOS/SD Selection

100BASE-FX operation is enabled by the use of the FX mode strap (fx_mode_strap_1) and is reflected in the 100BASE-FX Mode (FX_MODE) bit in the PHY Special Modes Register (PHY_SPECIAL_MODES).

Loss of Signal mode is selected by the **FXLOSEN** strap input pin.

If Loss of Signal mode is not selected, then Signal Detect mode is selected by the **FXSDENA** strap input pin. When greater than 1 V (typ.), Signal Detect mode is enabled, when less than 1 V (typ.), copper twisted pair is enabled.

Note: The **FXSDENA** strap input pin is shared with the **FXSDA** pin. As such, the LVPECL levels ensure that the input is greater than 1 V (typ.) and that Signal Detect mode is selected. When TP copper is desired, the Signal Detect input function is not required and the pin should be set to 0 V.

Care must be taken such that an non-powered or disabled transceiver does not load the Signal Detect input below the valid LVPECL level.

Table 12-8 summarizes the selection.

TABLE 12-8: 100BASE-FX LOS, SD AND TP COPPER SELECTION

FXLOSEN	FXSDENA	PHY Mode
<1 V (typ.)	<1 V (typ.)	TP copper
	>1 V (typ.)	100BASE-FX Signal Detect
>1 V (typ.)	n/a	100BASE-FX LOS

12.2.17 REQUIRED ETHERNET MAGNETICS (100BASE-TX AND 10BASE-T)

The magnetics selected for use with the device should be an Auto-MDIX style magnetic, which is widely available from several vendors. Please review the SMSC/Microchip Application note 8.13 “Suggested Magnetics” for the latest qualified and suggested magnetics. A list of vendors and part numbers are provided within the application note.

12.2.18 PHY REGISTERS

The PHY registers are indirectly accessed through the Host MAC MII Access Register (HMAC_MII_ACC) and Host MAC MII Data Register (HMAC_MII_DATA).

A list of the MII serial accessible Control and Status registers and their corresponding register index numbers is included in Table 12-9. Each individual PHY is assigned a unique PHY address as detailed in Section 12.1.1, "PHY Addressing," on page 210.

In addition to the MII serial accessible Control and Status registers, a set of indirectly accessible registers provides support for the *IEEE 802.3 Section 45.2 MDIO Manageable Device (MMD) Registers*. A list of these registers and their corresponding register index numbers is included in Table 12-15.

Note: The Digital Reset (DIGITAL_RST) bit will cause the EEPROM Loader to reload the configuration strap values into the PHY registers and to reset all other PHY registers to their default values. An EEPROM RELOAD command via the EEPROM Command Register (E2P_CMD) also has the same effect.

Control and Status Registers

Table 12-9 provides a list of supported registers. Register details, including bit definitions, are provided in the following subsections.

Unless otherwise specified, reserved fields must be written with zeros if the register is written.

TABLE 12-9: PHY MII SERIALLY ACCESSIBLE CONTROL AND STATUS REGISTERS

Index	Register Name (SYMBOL)	Group
0	PHY Basic Control Register (PHY_BASIC_CONTROL)	Basic
1	PHY Basic Status Register (PHY_BASIC_STATUS)	Basic
2	PHY Identification MSB Register (PHY_ID_MSB)	Extended
3	PHY Identification LSB Register (PHY_ID_LSB)	Extended
4	PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV)	Extended

TABLE 12-9: PHY MII SERIALLY ACCESSIBLE CONTROL AND STATUS REGISTERS

Index	Register Name (SYMBOL)	Group
5	PHY Auto-Negotiation Link Partner Base Page Ability Register (PHY_AN_LP_BASE_ABILITY)	Extended
6	PHY Auto-Negotiation Expansion Register (PHY_AN_EXP)	Extended
7	PHY Auto Negotiation Next Page TX Register (PHY_AN_NP_TX)	Extended
8	PHY Auto Negotiation Next Page RX Register (PHY_AN_NP_RX)	Extended
13	PHY MMD Access Control Register (PHY_MMD_ACCESS)	Extended
14	PHY MMD Access Address/Data Register (PHY_MMD_ADDR_DATA)	Extended
16	PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG)	Vendor-specific
17	PHY Mode Control/Status Register (PHY_MODE_CONTROL_STATUS)	Vendor-specific
18	PHY Special Modes Register (PHY_SPECIAL_MODES)	Vendor-specific
24	PHY TDR Patterns/Delay Control Register (PHY_TDR_PAT_DELAY)	Vendor-specific
25	PHY TDR Control/Status Register (PHY_TDR_CONTROL_STAT)	Vendor-specific
26	PHY Symbol Error Counter Register	Vendor-specific
27	PHY Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND)	Vendor-specific
28	PHY Cable Length Register (PHY_CABLE_LEN)	Vendor-specific
29	PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE)	Vendor-specific
30	PHY Interrupt Mask Register (PHY_INTERRUPT_MASK)	Vendor-specific
31	PHY Special Control/Status Register (PHY_SPECIAL_CONTROL_STATUS)	Vendor-specific

12.2.18.1 PHY Basic Control Register (PHY_BASIC_CONTROL)

Index (decimal): 0 Size: 16 bits

This read/write register is used to configure the PHY.

Bits	Description	Type	Default
15	**Soft Reset (PHY_SRST)** When set, this bit resets all the PHY registers to their default state, except those marked as NASR type. This bit is self clearing. **0: Normal operation** **1: Reset**	R/W SC	0b
14	**Loopback (PHY_LOOPBACK)** This bit enables/disables the loopback mode. When enabled, transmissions are not sent to network. Instead, they are looped back into the PHY. **0: Loopback mode disabled (normal operation)** **1: Loopback mode enabled**	R/W	0b
13	**Speed Select LSB (PHY_SPEED_SEL_LSB)** This bit is used to set the speed of the PHY when the Auto-Negotiation Enable (PHY_AN) bit is disabled. **0: 10 Mbps** **1: 100 Mbps**	R/W	Note 4
12	**Auto-Negotiation Enable (PHY_AN)** This bit enables/disables Auto-Negotiation. When enabled, the Speed Select LSB (PHY_SPEED_SEL_LSB) and Duplex Mode (PHY_DUPLEX) bits are overridden. This bit is forced to a 0 if the 100BASE-FX Mode (FX_MODE) bit of the PHY Special Modes Register (PHY_SPECIAL_MODES) is a high. **0: Auto-Negotiation disabled** **1: Auto-Negotiation enabled**	R/W	Note 5
11	**Power Down (PHY_PWR_DWN)** This bit controls the power down mode of the PHY. **0: Normal operation** **1: General power down mode**	R/W	0b
10	**RESERVED**	RO	-
9	**Restart Auto-Negotiation (PHY_RST_AN)** When set, this bit restarts the Auto-Negotiation process. **0: Normal operation** **1: Auto-Negotiation restarted**	R/W SC	0b

Bits	Description	Type	Default
8	**Duplex Mode (PHY_DUPLEX)** This bit is used to set the duplex when the Auto-Negotiation Enable (PHY_AN) bit is disabled. **0: Half Duplex** **1: Full Duplex**	R/W	Note 6
7	**Collision Test Mode (PHY_COL_TEST)** This bit enables/disables the collision test mode of the PHY. When set, the collision signal is active during transmission. It is recommended that this feature be used only in loopback mode. **0: Collision test mode disabled** **1: Collision test mode enabled**	R/W	0b
6:0	**RESERVED**	RO	-

Note 4: The default value of this bit is determined by the logical OR of the Auto-Negotiation strap (autoneg_strap_1) and the speed select strap (speed_strap_1). Essentially, if the Auto-Negotiation strap is set, the default value is 1, otherwise the default is determined by the value of the speed select strap. Refer to Section 7.0, "Configuration Straps," on page 54 for more information. In 100BASE-FX mode, the default value of this bit is a 1.

Note 5: The default is the value of the Auto-Negotiation strap (autoneg_strap_1). Refer to Section 7.0, "Configuration Straps," on page 54 for more information. In 100BASE-FX mode, the default value of this bit is a 0.

Note 6: The default value of this bit is determined by the logical AND of the negation of the Auto-Negotiation strap (autoneg_strap_1) and the duplex select strap (duplex_strap_1). Essentially, if the Auto-Negotiation strap is set, the default value is 0, otherwise the default is determined by the value of the duplex select strap. Refer to Section 7.0, "Configuration Straps," on page 54 for more information.

In 100BASE-FX mode, the Auto-Negotiation strap is not considered and the default of this bit is the value of the duplex select strap.

12.2.18.2 PHY Basic Status Register (PHY_BASIC_STATUS)

Index (decimal): 1 Size: 16 bits

This register is used to monitor the status of the PHY.

Bits	Description	Type	Default
15	**100BASE-T4** This bit displays the status of 100BASE-T4 compatibility. **0: PHY not able to perform 100BASE-T4** **1: PHY able to perform 100BASE-T4**	RO	0b
14	**100BASE-X Full Duplex** This bit displays the status of 100BASE-X full duplex compatibility. **0: PHY not able to perform 100BASE-X full duplex** **1: PHY able to perform 100BASE-X full duplex**	RO	1b
13	**100BASE-X Half Duplex** This bit displays the status of 100BASE-X half duplex compatibility. **0: PHY not able to perform 100BASE-X half duplex** **1: PHY able to perform 100BASE-X half duplex**	RO	1b
12	**10BASE-T Full Duplex** This bit displays the status of 10BASE-T full duplex compatibility. **0: PHY not able to perform 10BASE-T full duplex** **1: PHY able to perform 10BASE-T full duplex**	RO	1b
11	**10BASE-T Half Duplex** (typ.) This bit displays the status of 10BASE-T half duplex compatibility. **0: PHY not able to perform 10BASE-T half duplex** **1: PHY able to perform 10BASE-T half duplex**	RO	1b
10	**100BASE-T2 Full Duplex** This bit displays the status of 100BASE-T2 full duplex compatibility. **0: PHY not able to perform 100BASE-T2 full duplex** **1: PHY able to perform 100BASE-T2 full duplex**	RO	0b
9	**100BASE-T2 Half Duplex** This bit displays the status of 100BASE-T2 half duplex compatibility. **0: PHY not able to perform 100BASE-T2 half duplex** **1: PHY able to perform 100BASE-T2 half duplex**	RO	0b
8	**Extended Status** This bit displays whether extended status information is in register 15 (per IEEE 802.3 clause 22.2.4). **0: No extended status information in Register 15** **1: Extended status information in Register 15**	RO	0b

Bits	Description	Type	Default
7	**Unidirectional Ability** This bit indicates whether the PHY is able to transmit regardless of whether the PHY has determined that a valid link has been established. **0: Can only transmit when a valid link has been established** **1: Can transmit regardless**	RO	0b
6	**MF Preamble Suppression** This bit indicates whether the PHY accepts management frames with the preamble suppressed. **0: Management frames with preamble suppressed not accepted** **1: Management frames with preamble suppressed accepted**	RO	0b
5	**Auto-Negotiation Complete** This bit indicates the status of the Auto-Negotiation process. **0: Auto-Negotiation process not completed** **1: Auto-Negotiation process completed**	RO	0b
4	**Remote Fault** This bit indicates if a remote fault condition has been detected. **0: No remote fault condition detected** **1: Remote fault condition detected**	RO/LH	0b
3	**Auto-Negotiation Ability** This bit indicates the PHY's Auto-Negotiation ability. **0: PHY is unable to perform Auto-Negotiation** **1: PHY is able to perform Auto-Negotiation**	RO	1b
2	**Link Status** This bit indicates the status of the link. **0: Link is down** **1: Link is up**	RO/LL	0b
1	**Jabber Detect** This bit indicates the status of the jabber condition. **0: No jabber condition detected** **1: Jabber condition detected**	RO/LH	0b
0	**Extended Capability** This bit indicates whether extended register capability is supported. **0: Basic register set capabilities only** **1: Extended register set capabilities**	RO	1b

12.2.18.3 PHY Identification MSB Register (PHY_ID_MSB)

Index (decimal): 2 Size: 16 bits

This read/write register contains the MSB of the Organizationally Unique Identifier (OUI) for the PHY. The LSB of the PHY OUI is contained in the PHY Identification LSB Register (PHY_ID_LSB).

Bits	Description	Type	Default
15:0	**PHY ID** This field is assigned to the 3rd through 18th bits of the OUI, respectively (OUI = 00800Fh).	R/W	0007h

12.2.18.4 PHY Identification LSB Register (PHY_ID_LSB)

Index (decimal): 3 Size: 16 bits

This read/write register contains the LSB of the Organizationally Unique Identifier (OUI) for the PHY. The MSB of the PHY OUI is contained in the PHY Identification MSB Register (PHY_ID_MSB).

Bits	Description	Type	Default
15:10	**PHY ID** This field is assigned to the 19th through 24th bits of the PHY OUI, respectively. (OUI = 00800Fh).	R/W	C140h
9:4	**Model Number** This field contains the 6-bit manufacturer's model number of the PHY.	R/W	
3:0	**Revision Number** This field contain the 4-bit manufacturer's revision number of the PHY.	R/W	

Note: The default value of the Revision Number field may vary dependent on the silicon revision number.

12.2.18.5 PHY Auto-Negotiation Advertisement Register (PHY_AN_ADV)

Index (decimal): 4 Size: 16 bits

This read/write register contains the advertised ability of the PHY and is used in the Auto-Negotiation process with the link partner.

Bits	Description	Type	Default
15	**Next Page** **0 = No next page ability** **1 = Next page capable**	R/W	0b
14	**RESERVED**	RO	-
13	**Remote Fault** This bit determines if remote fault indication will be advertised to the link partner. **0: Remote fault indication not advertised** **1: Remote fault indication advertised**	R/W	0b
12	**Extended Next Page** **Note:** This bit should be written as 0.	R/W	0b
11	**Asymmetric Pause** This bit determines the advertised asymmetric pause capability. **0: No Asymmetric PAUSE toward link partner advertised** **1: Asymmetric PAUSE toward link partner advertised**	R/W	Note 7
10	**Symmetric Pause** This bit determines the advertised symmetric pause capability. **0: No Symmetric PAUSE toward link partner advertised** **1: Symmetric PAUSE toward link partner advertised**	R/W	Note 7
9	**RESERVED**	RO	-
8	**100BASE-X Full Duplex** This bit determines the advertised 100BASE-X full duplex capability. **0: 100BASE-X full duplex ability not advertised** **1: 100BASE-X full duplex ability advertised**	R/W	1b
7	**100BASE-X Half Duplex** This bit determines the advertised 100BASE-X half duplex capability. **0: 100BASE-X half duplex ability not advertised** **1: 100BASE-X half duplex ability advertised**	R/W	1b
6	**10BASE-T Full Duplex** This bit determines the advertised 10BASE-T full duplex capability. **0: 10BASE-T full duplex ability not advertised** **1: 10BASE-T full duplex ability advertised**	R/W	Note 8 Table 12-10

Bits	Description	Type	Default
5	**10BASE-T Half Duplex** This bit determines the advertised 10BASE-T half duplex capability. **0: 10BASE-T half duplex ability not advertised** **1: 10BASE-T half duplex ability advertised**	R/W	Note 9 Table 12-11
4:0	**Selector Field** This field identifies the type of message being sent by Auto-Negotiation. **00001: IEEE 802.3**	R/W	00001b

Note 7: The default values of the Asymmetric Pause and Symmetric Pause bits are determined by the Manual Flow Control Enable Strap (manual_FC_strap_1). When the Manual Flow Control Enable Strap is 0, the Symmetric Pause bit defaults to 1 and the Asymmetric Pause bit defaults to the setting of the Full-Duplex Flow Control Enable Strap (FD_FC_strap_1). When the Manual Flow Control Enable Strap is 1, both bits default to 0. Refer to Section 7.0, "Configuration Straps," on page 54 for more information. In 100BASE-FX mode, the default value of these bits is 0.

Note 8: The default value of this bit is determined by the logical OR of the Auto-Negotiation Enable strap (autoneg_strap_1) with the logical AND of the negated Speed Select strap (speed_strap_1) and the Duplex Select Strap (duplex_strap_1). Table 12-10 defines the default behavior of this bit. Refer to Section 7.0, "Configuration Straps," on page 54 for more information. In 100BASE-FX mode, the default value of this bit is a 0.

TABLE 12-10: 10BASE-T FULL DUPLEX ADVERTISEMENT DEFAULT VALUE

autoneg_strap_1	speed_strap_1	duplex_strap_1	Default 10BASE-T Full Duplex (Bit 6) Value
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	x	x	1

Note 9: The default value of this bit is determined by the logical OR of the Auto-Negotiation strap (autoneg_strap_1) and the negated Speed Select strap (speed_strap_1). Table 12-11 defines the default behavior of this bit. Refer to Section 7.0, "Configuration Straps," on page 54 for more information. In 100BASE-FX mode, the default value of this bit is a 0.

TABLE 12-11: 10BASE-T HALF DUPLEX ADVERTISEMENT BIT DEFAULT VALUE

autoneg_strap_1	speed_strap_1	Default 10BASE-T Half Duplex (Bit 5) Value
0	0	1
0	1	0
1	0	1
1	1	1

12.2.18.6 PHY Auto-Negotiation Link Partner Base Page Ability Register (PHY_AN_LP_BASE_ABILITY)

Index (decimal): 5 Size: 16 bits

This read-only register contains the advertised ability of the link partner's PHY and is used in the Auto-Negotiation process between the link partner and the PHY.

Bits	Description	Type	Default
15	**Next Page** This bit indicates the link partner PHY page capability. **0: Link partner PHY does not advertise next page capability** **1: Link partner PHY advertises next page capability**	RO	0b
14	**Acknowledge** This bit indicates whether the link code word has been received from the partner. **0: Link code word not yet received from partner** **1: Link code word received from partner**	RO	0b
13	**Remote Fault** This bit indicates whether a remote fault has been detected. **0: No remote fault** **1: Remote fault detected**	RO	0b
12	**Extended Next Page** **0: Link partner PHY does not advertise extended next page capability** **1: Link partner PHY advertises extended next page capability**	RO	0b
11	**Asymmetric Pause** This bit indicates the link partner PHY asymmetric pause capability. **0: No Asymmetric PAUSE toward link partner** **1: Asymmetric PAUSE toward link partner**	RO	0b
10	**Pause** This bit indicates the link partner PHY symmetric pause capability. **0: No Symmetric PAUSE toward link partner** **1: Symmetric PAUSE toward link partner**	RO	0b
9	**100BASE-T4** This bit indicates the link partner PHY 100BASE-T4 capability. **0: 100BASE-T4 ability not supported** **1: 100BASE-T4 ability supported**	RO	0b
8	**100BASE-X Full Duplex** This bit indicates the link partner PHY 100BASE-X full duplex capability. **0: 100BASE-X full duplex ability not supported** **1: 100BASE-X full duplex ability supported**	RO	0b

Bits	Description	Type	Default
7	**100BASE-X Half Duplex** This bit indicates the link partner PHY 100BASE-X half duplex capability. **0: 100BASE-X half duplex ability not supported** **1: 100BASE-X half duplex ability supported**	RO	0b
6	**10BASE-T Full Duplex** This bit indicates the link partner PHY 10BASE-T full duplex capability. **0: 10BASE-T full duplex ability not supported** **1: 10BASE-T full duplex ability supported**	RO	0b
5	**10BASE-T Half Duplex** This bit indicates the link partner PHY 10BASE-T half duplex capability. **0: 10BASE-T half duplex ability not supported** **1: 10BASE-T half duplex ability supported**	RO	0b
4:0	**Selector Field** This field identifies the type of message being sent by Auto-Negotiation. **00001: IEEE 802.3**	RO	00001b

12.2.18.7 PHY Auto-Negotiation Expansion Register (PHY_AN_EXP)

Index (decimal): 6 Size: 16 bits

This read/write register is used in the Auto-Negotiation process between the link partner and the PHY.

Bits	Description	Type	Default
15:7	**RESERVED**	RO	-
6	**Receive Next Page Location Able** 0 = Received next page storage location is not specified by bit 6.5 1 = Received next page storage location is specified by bit 6.5	RO	1b
5	**Received Next Page Storage Location** **0 = Link partner next pages are stored in the** PHY Auto-Negotiation Link Partner Base Page Ability Register (PHY_AN_LP_BASE_ABILITY) **(PHY register 5)** **1 = Link partner next pages are stored in the** PHY Auto Negotiation Next Page RX Register (PHY_AN_NP_RX) **(PHY register 8)**	RO	1b
4	**Parallel Detection Fault** This bit indicates whether a Parallel Detection Fault has been detected. **0: A fault hasn't been detected via the Parallel Detection function** **1: A fault has been detected via the Parallel Detection function**	RO/LH	0b
3	**Link Partner Next Page Able** This bit indicates whether the link partner has next page ability. **0: Link partner does not contain next page capability** **1: Link partner contains next page capability**	RO	0b
2	**Next Page Able** This bit indicates whether the local device has next page ability. **0: Local device does not contain next page capability** **1: Local device contains next page capability**	RO	1b
1	**Page Received** This bit indicates the reception of a new page. **0: A new page has not been received** **1: A new page has been received**	RO/LH	0b
0	**Link Partner Auto-Negotiation Able** This bit indicates the Auto-Negotiation ability of the link partner. **0: Link partner is not Auto-Negotiation able** **1: Link partner is Auto-Negotiation able**	RO	0b

12.2.18.8 PHY Auto Negotiation Next Page TX Register (PHY_AN_NP_TX)

Index (In Decimal): 7 Size: 16 bits

Bits	Description	Type	Default
15	**Next Page** 0 = No next page ability 1 = Next page capable	R/W	0b
14	**RESERVED**	RO	-
13	**Message Page** 0 = Unformatted page 1 = Message page	R/W	1b
12	**Acknowledge 2** 0 = Device cannot comply with message. 1 = Device will comply with message.	R/W	0b
11	**Toggle** 0 = Previous value was HIGH. 1 = Previous value was LOW.	RO	0b
10:0	**Message Code** Message/Unformatted Code Field	R/W	000 0000 0001b

12.2.18.9 PHY Auto Negotiation Next Page RX Register (PHY_AN_NP_RX)

Index (In Decimal): 8 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15	**Next Page** **0 = No next page ability** **1 = Next page capable**	RO	0b
14	**Acknowledge** **0 = Link code word not yet received from partner** **1 = Link code word received from partner**	RO	0b
13	**Message Page** **0 = Unformatted page** **1 = Message page**	RO	0b
12	**Acknowledge 2** **0 = Device cannot comply with message.** **1 = Device will comply with message.**	RO	0b
11	**Toggle** **0 = Previous value was HIGH.** **1 = Previous value was LOW.**	RO	0b
10:0	**Message Code** Message/Unformatted Code Field	RO	000 0000 0000b

12.2.18.10 PHY MMD Access Control Register (PHY_MMD_ACCESS)

Index (In Decimal): 13 Size: 16 bits

This register in conjunction with the PHY MMD Access Address/Data Register (PHY_MMD_ADDR_DATA) provides indirect access to the MDIO Manageable Device (MMD) registers. Refer to the MDIO Manageable Device (MMD) Registers on page 263 for additional details.

Bits	Description	Type	Default
15:14	**MMD Function** This field is used to select the desired MMD function: **00 = Address** **01 = Data, no post increment** **10 = RESERVED** **11 = RESERVED**	R/W	00b
13:5	**RESERVED**	RO	-
4:0	**MMD Device Address (DEVAD)** This field is used to select the desired MMD device address. (3 = PCS, 7 = auto-negotiation)	R/W	0h

12.2.18.11 PHY MMD Access Address/Data Register (PHY_MMD_ADDR_DATA)

Index (In Decimal): 14 Size: 16 bits

This register in conjunction with the PHY MMD Access Control Register (PHY_MMD_ACCESS) provides indirect access to the MDIO Manageable Device (MMD) registers. Refer to the MDIO Manageable Device (MMD) Registers on page 263 for additional details.

Bits	Description	Type	Default
15:0	**MMD Register Address/Data** If the MMD Function field of the PHY MMD Access Control Register (PHY_MMD_ACCESS) is “00”, this field is used to indicate the MMD register address to read/write of the device specified in the MMD Device Address (DEVAD) field. Otherwise, this register is used to read/write data from/to the previously specified MMD address.	R/W	0000h

12.2.18.12 PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG)

Index (decimal): 16 Size: 16 bits

This register is used to Enable EEE functionality and control NLP pulse generation and the Auto-MDIX Crossover Time of the PHY.

Bits	Description	Type	Default
15	**EDPD TX NLP Enable** Enables the generation of a Normal Link Pulse (NLP) with a selectable interval while in Energy Detect Power-Down. 0=disabled, 1=enabled. The Energy Detect Power-Down (EDPWRDOWN) bit in the PHY Mode Control/Status Register (PHY_MODE_CONTROL_STATUS) needs to be set in order to enter Energy Detect Power-Down mode and the PHY needs to be in the Energy Detect Power-Down state in order for this bit to generate the NLP. Bit 3 of this register also needs to be set when setting this bit.	R/W NASR Note 10	0b
14:13	**EDPD TX NLP Interval Timer Select** Specifies how often a NLP is transmitted while in the Energy Detect Power-Down state. **00b: 1 s** **01b: 768 ms** **10b: 512 ms** **11b: 256 ms**	R/W NASR Note 10	00b
12	**EDPD RX Single NLP Wake Enable** When set, the PHY will wake upon the reception of a single Normal Link Pulse. When clear, the PHY requires two link pluses, within the interval specified below, in order to wake up. Single NLP Wake Mode is recommended when connecting to “Green” network devices.	R/W NASR Note 10	0b
11:10	**EDPD RX NLP Max Interval Detect Select** These bits specify the maximum time between two consecutive Normal Link Pulses in order for them to be considered a valid wake up signal. **00b: 64 ms** **01b: 256 ms** **10b: 512 ms** **11b: 1 s**	R/W NASR Note 10	00b
9:3	**RESERVED**	RO	-
2	**PHY Energy Efficient Ethernet Enable (PHYEEEEN)** When set, enables Energy Efficient Ethernet (EEE) operation in the PHY. When cleared, EEE operation is disabled. Refer to Section 12.2.11, "Energy Efficient Ethernet," on page 223 for additional information.	R/W NASR Note 10	Note 11

Bits	Description	Type	Default
1	**EDPD Extend Crossover** When in Energy Detect Power-Down (EDPD) mode (Energy Detect Power-Down (EDPWRDOWN) = 1), setting this bit to 1 extends the crossover time by 2976 ms. **0 = Crossover time extension disabled** **1 = Crossover time extension enabled (2976 ms)**	R/W NASR Note 10	0b
0	**Extend Manual 10/100 Auto-MDIX Crossover Time** When Auto-Negotiation is disabled, setting this bit extends the Auto-MDIX crossover time by 32 sample times (32 * 62 ms = 1984 ms). This allows the link to be established with a partner PHY that has Auto-Negotiation enabled. When Auto-Negotiation is enabled, this bit has no affect. It is recommended that this bit is set when disabling AN with Auto-MDIX enabled.	R/W NASR Note 10	1b

Note 10: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) is set.

Note 11: The default value of this bit is a 0 if in 100BASE-FX mode, otherwise the default value of this bit is determined by the Energy Efficient Ethernet Enable Strap (EEE_enable_strap_1). Refer to Section 7.0, "Configuration Straps," on page 54 for more information.

12.2.18.13 PHY Mode Control/Status Register (PHY_MODE_CONTROL_STATUS)

Index (decimal): 17 Size: 16 bits

This read/write register is used to control and monitor various PHY configuration options.

Bits	Description	Type	Default
15:14	**RESERVED**	RO	-
13	**Energy Detect Power-Down (EDPWRDOWN)** This bit controls the Energy Detect Power-Down mode. **0: Energy Detect Power-Down is disabled** **1: Energy Detect Power-Down is enabled** **Note:** When in EDPD mode, the device's NLP characteristics can be modified via the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG).	R/W	0b
12:7	**RESERVED**	RO	-
6	**ALTINT** Alternate Interrupt Mode: **0 = Primary interrupt system enabled (Default)** **1 = Alternate interrupt system enabled** Refer to Section 12.2.9, "PHY Interrupts," on page 220 for additional information.	R/W NASR Note 12	0b
5:2	**RESERVED**	RO	-
1	**Energy On (ENERGYON)** Indicates whether energy is detected. This bit transitions to "0" if no valid energy is detected within 256 ms (1500 ms if auto-negotiation is enabled). It is reset to "1" by a hardware reset and by a software reset if auto-negotiation was enabled or will be enabled via strapping. Refer to Section 12.2.10.2, "Energy Detect Power-Down," on page 223 for additional information.	RO	1b
0	**RESERVED**	RO	-

Note 12: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) is set.

12.2.18.14 PHY Special Modes Register (PHY_SPECIAL_MODES)

Index (decimal): 18 Size: 16 bits

This read/write register is used to control the special modes of the PHY.

Bits	Description	Type	Default
15:11	**RESERVED**	RO	-
10	**100BASE-FX Mode (FX_MODE)** This bit enables 100BASE-FX Mode **Note:** FX_MODE cannot properly be changed with this bit. This bit must always be written with its current value. Device strapping must be used to set the desired mode.	R/W NASR Note 13	Note 14
9:8	**RESERVED**	RO	-
7:5	**PHY Mode (MODE[2:0])** This field controls the PHY mode of operation. Refer to Table 12-12 for a definition of each mode. **Note:** This field should be written with its read value.	R/W NASR Note 13	Note 15
4:0	**PHY Address (PHYADD)** The PHY Address field determines the MMI address to which the PHY will respond and is also used for initialization of the cipher (scrambler) key. Refer to Section 12.1.1, "PHY Addressing," on page 210 for additional information.	R/W NASR Note 13	Note 16

Note 13: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) is set.

Note 14: The default value of this bit is determined by the Fiber Enable strap (fx_mode_strap_1).

Note 15: The default value of this field is determined by a combination of the configuration straps autoneg_strap_1, speed_strap_1, and duplex_strap_1. If the autoneg_strap_1 is 1, then the default MODE[2:0] value is 111b. Else, the default value of this field is determined by the remaining straps. MODE[2]=0, MODE[1]=(speed_strap_1), and MODE[0]=(duplex_strap_1). Refer to Section 7.0, "Configuration Straps," on page 54 for more information. In 100BASE-FX mode, the default value of these bits is 010b or 011b. depending on the duplex configuration strap.

Note 16: The default value of this field is determined per Section 12.1.1, "PHY Addressing," on page 210.

TABLE 12-12: MODE[2:0] DEFINITIONS

MODE[2:0]	Mode Definitions
000	10BASE-T Half Duplex. Auto-Negotiation disabled.
001	10BASE-T Full Duplex. Auto-Negotiation disabled.
010	100BASE-TX or 100BASE-FX Half Duplex. Auto-Negotiation disabled. CRS is active during Transmit & Receive.
011	100BASE-TX or 100BASE-FX Full Duplex. Auto-Negotiation disabled. CRS is active during Receive.

TABLE 12-12: MODE[2:0] DEFINITIONS (CONTINUED)

MODE[2:0]	Mode Definitions
100	100BASE-TX Full Duplex is advertised. Auto-Negotiation enabled. CRS is active during Receive.
101	RESERVED
110	Power Down mode.
111	All capable. Auto-Negotiation enabled.

12.2.18.15 PHY TDR Patterns/Delay Control Register (PHY_TDR_PAT_DELAY)

Index (In Decimal): 24 Size: 16 bits

Bits	Description	Type	Default
15	**TDR Delay In** **0 = Line break time is 2 ms.** **1 = The device uses TDR Line Break Counter to increase the line break time before starting TDR.**	R/W NASR Note 17	1b
14:12	**TDR Line Break Counter** When TDR Delay In is 1, this field specifies the increase in line break time in increments of 256 ms, up to 2 seconds.	R/W NASR Note 17	001b
11:6	**TDR Pattern High** This field specifies the data pattern sent in TDR mode for the high cycle.	R/W NASR Note 17	101110b
5:0	**TDR Pattern Low** This field specifies the data pattern sent in TDR mode for the low cycle.	R/W NASR Note 17	011101b

Note 17: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) is set.

12.2.18.16 PHY TDR Control/Status Register (PHY_TDR_CONTROL_STAT)

Index (In Decimal): 25 Size: 16 bits

Bits	Description	Type	Default
15	**TDR Enable** **0 = TDR mode disabled** **1 = TDR mode enabled** **Note:** This bit self clears when TDR completes (TDR Channel Status goes high)	R/W NASR SC Note 18	0b
14	**TDR Analog to Digital Filter Enable** **0 = TDR analog to digital filter disabled** **1 = TDR analog to digital filter enabled (reduces noise spikes during TDR pulses)**	R/W NASR Note 18	0b
13:11	**RESERVED**	RO	-
10:9	**TDR Channel Cable Type** Indicates the cable type determined by the TDR test. **00 = Default** **01 = Shorted cable condition** **10 = Open cable condition** **11 = Match cable condition**	R/W NASR Note 18	00b
8	**TDR Channel Status** When high, this bit indicates that the TDR operation has completed. This bit will stay high until reset or the TDR operation is restarted (TDR Enable = 1)	R/W NASR Note 18	0b
7:0	**TDR Channel Length** This eight bit value indicates the TDR channel length during a short or open cable condition. Refer to Section 12.2.14.1, "Time Domain Reflectometry (TDR) Cable Diagnostics," on page 225 for additional information on the usage of this field. **Note:** This field is not valid during a match cable condition. The PHY Cable Length Register (PHY_CABLE_LEN) must be used to determine cable length during a non-open/short (match) condition. Refer to Section 12.2.14, "Cable Diagnostics," on page 224 for additional information.	R/W NASR Note 18	00h

Note 18: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) is set.

12.2.18.17 PHY Symbol Error Counter Register

Index (In Decimal): 26 Size: 16 bits

Bits	Description	Type	Default
15:0	**Symbol Error Counter (SYM_ERR_CNT)** This 100BASE-TX receiver-based error counter increments when an invalid code symbol is received, including IDLE symbols. The counter is incremented only once per packet, even when the received packet contains more than one symbol error. This field counts up to 65,536 and rolls over to 0 if incremented beyond its maximum value. **Note:** This register is cleared on reset, but is not cleared by reading the register. It does not increment in 10BASE-T mode.	RO	0000h

12.2.18.18 PHY Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND)

Index (decimal): 27 Size: 16 bits

This read/write register is used to control various options of the PHY.

Bits	Description	Type	Default
15	**Auto-MDIX Control (AMDIXCTRL)** This bit is responsible for determining the source of Auto-MDIX control for the port. When set, the Manual MDIX and Auto MDIX straps (manual_mdix-_strap_1/auto_mdix_strap_1) are overridden, and Auto-MDIX functions are controlled using the AMDIXEN and AMDIXSTATE bits of this register. When cleared, Auto-MDIX functionality is controlled by the Manual MDIX and Auto MDIX straps by default. Refer to Section 7.0, "Configuration Straps," on page 54 for configuration strap definitions. **0: Port Auto-MDIX determined by strap inputs (Table 12-14)** **1: Port Auto-MDIX determined by bits 14 and 13** **Note:** The value of auto_mdix_strap_1 is indicated in the AMDIX_EN Strap State bit of the Hardware Configuration Register (HW_CFG).	R/W NASR Note 19	0b
14	**Auto-MDIX Enable (AMDIXEN)** When the AMDIXCTRL bit of this register is set, this bit is used in conjunction with the AMDIXSTATE bit to control the Port Auto-MDIX functionality as shown in Table 12-13. Auto-MDIX is not appropriate and should not be enabled for 100BASE-FX mode.	R/W NASR Note 19	0b
13	**Auto-MDIX State (AMDIXSTATE)** When the AMDIXCTRL bit of this register is set, this bit is used in conjunction with the AMDIXEN bit to control the Port Auto-MDIX functionality as shown in Table 12-13.	R/W NASR Note 19	0b
12	**RESERVED**	RO	-
11	**SQE Test Disable (SQEOFF)** This bit controls the disabling of the SQE test (Heartbeat). SQE test is enabled by default. **0: SQE test enabled** **1: SQE test disabled**	R/W NASR Note 19	0b
10:6	**RESERVED**	RO	-
5	**Far End Fault Indication Enable (FEFI_EN)** This bit enables Far End Fault Generation and Detection. See Section 12.2.16.1, "100BASE-FX Far End Fault Indication," on page 229 for more information.	R/W	Note 20
4	**10Base-T Polarity State (XPOL)** This bit shows the polarity state of the 10Base-T. **0: Normal Polarity** **1: Reversed Polarity**	RO	0b

Bits	Description	Type	Default
3:0	**RESERVED**	RO	-

Note 19: Register bits designated as NASR are reset when the PHY Reset is generated via the Reset Control Register (RESET_CTL). The NASR designation is only applicable when the Soft Reset (PHY_SRST) bit of the PHY Basic Control Register (PHY_BASIC_CONTROL) is set.

Note 20: The default value of this bit is a 1 if in 100BASE-FX mode, otherwise the default is a 0.

TABLE 12-13: AUTO-MDIX ENABLE AND AUTO-MDIX STATE BIT FUNCTIONALITY

Auto-MDIX Enable	Auto-MDIX State	Mode
0	0	Manual mode, no crossover
0	1	Manual mode, crossover
1	0	Auto-MDIX mode
1	1	RESERVED (do not use this state)

TABLE 12-14: MDIX STRAP FUNCTIONALITY

auto_mdix_strap_1	manual_mdix_strap_1	Mode
0	0	Manual mode, no crossover
0	1	Manual mode, crossover
1	x	Auto-MDIX mode

12.2.18.19 PHY Cable Length Register (PHY_CABLE_LEN)

Index (In Decimal): 28 Size: 16 bits

Bits	Description	Type	Default
15:12	**Cable Length (CBLN)** This four bit value indicates the cable length. Refer to Section 12.2.14.2, "Matched Cable Diagnostics," on page 227 for additional information on the usage of this field. **Note:** This field indicates cable length for 100BASE-TX linked devices that do not have an open/short on the cable. To determine the open/short status of the cable, the PHY TDR Patterns/Delay Control Register (PHY_TDR_PAT_DELAY) and PHY TDR Control/ Status Register (PHY_TDR_CONTROL_STAT) must be used. Cable length is not supported for 10BASE-T links. Refer to Section 12.2.14, "Cable Diagnostics," on page 224 for additional information.	RO	0000b
11:0	**RESERVED - Write as 100000000000b, ignore on read**	R/W	-

12.2.18.20 PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE)

Index (decimal): 29 Size: 16 bits

This read-only register is used to determine to source of various PHY interrupts. All interrupt source bits in this register are read-only and latch high upon detection of the corresponding interrupt (if enabled). A read of this register clears the interrupts. These interrupts are enabled or masked via the PHY Interrupt Mask Register (PHY_INTERRUPT_MASK).

Bits	Description	Type	Default
15:9	**RESERVED**	RO	-
9	**INT9** This interrupt source bit indicates a Link Up (link status asserted). **0: Not source of interrupt** **1: Link Up (link status asserted)**	RO/LH	0b
8	**RESERVED**	RO	-
7	**INT7** This interrupt source bit indicates when the Energy On (ENERGYON) bit of the PHY Mode Control/Status Register (PHY_MODE_CONTROL_STATUS) has been set. **0: Not source of interrupt** **1: ENERGYON generated**	RO/LH	0b
6	**INT6** This interrupt source bit indicates Auto-Negotiation is complete. **0: Not source of interrupt** **1: Auto-Negotiation complete**	RO/LH	0b
5	**INT5** This interrupt source bit indicates a remote fault has been detected. **0: Not source of interrupt** **1: Remote fault detected**	RO/LH	0b
4	**INT4** This interrupt source bit indicates a Link Down (link status negated). **0: Not source of interrupt** **1: Link Down (link status negated)**	RO/LH	0b
3	**INT3** This interrupt source bit indicates an Auto-Negotiation LP acknowledge. **0: Not source of interrupt** **1: Auto-Negotiation LP acknowledge**	RO/LH	0b
2	**INT2** This interrupt source bit indicates a Parallel Detection fault. **0: Not source of interrupt** **1: Parallel Detection fault**	RO/LH	0b

Bits	Description	Type	Default
1	**INT1** This interrupt source bit indicates an Auto-Negotiation page received. **0: Not source of interrupt** **1: Auto-Negotiation page received**	RO/LH	0b
0	**RESERVED**	RO	-

12.2.18.21 PHY Interrupt Mask Register (PHY_INTERRUPT_MASK)

Index (decimal): 30 Size: 16 bits

This read/write register is used to enable or mask the various PHY interrupts and is used in conjunction with the PHY Interrupt Source Flags Register (PHY_INTERRUPT_SOURCE).

Bits	Description	Type	Default
15:10	**RESERVED**	RO	-
9	**INT9_MASK** This interrupt mask bit enables/masks the Link Up (link status asserted) interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
8	**RESERVED**	RO	-
7	**INT7_MASK** This interrupt mask bit enables/masks the ENERGYON interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
6	**INT6_MASK** This interrupt mask bit enables/masks the Auto-Negotiation interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
5	**INT5_MASK** This interrupt mask bit enables/masks the remote fault interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
4	**INT4_MASK** This interrupt mask bit enables/masks the Link Down (link status negated) interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
3	**INT3_MASK** This interrupt mask bit enables/masks the Auto-Negotiation LP acknowledge interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
2	**INT2_MASK** This interrupt mask bit enables/masks the Parallel Detection fault interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b

Bits	Description	Type	Default
1	**INT1_MASK** This interrupt mask bit enables/masks the Auto-Negotiation page received interrupt. **0: Interrupt source is masked** **1: Interrupt source is enabled**	R/W	0b
0	**RESERVED**	RO	-

12.2.18.22 PHY Special Control/Status Register (PHY_SPECIAL_CONTROL_STATUS)

Index (decimal): 31 Size: 16 bits

This read/write register is used to control and monitor various options of the PHY.

Bits	Description	Type	Default
15:13	**RESERVED**	RO	-
12	**Autodone** This bit indicates the status of the Auto-Negotiation on the PHY. **0: Auto-Negotiation is not completed, is disabled, or is not active** **1: Auto-Negotiation is completed**	RO	0b
11:5	**RESERVED - Write as 0000010b, ignore on read**	R/W	0000010b
4:2	**Speed Indication** This field indicates the current PHY speed configuration. STATE / DESCRIPTION 000 RESERVED 001 10BASE-T Half-duplex 010 100BASE-TX Half-duplex 011 RESERVED 100 RESERVED 101 10BASE-T Full-duplex 110 100BASE-TX Full-duplex 111 RESERVED	RO	XXXb
1:0	**RESERVED**	RO	0b

MDIO Manageable Device (MMD) Registers

The device MMD registers adhere to the *IEEE 802.3-2008 45.2 MDIO Interface Registers* specification. The MMD registers are not memory mapped. These registers are accessed indirectly via the PHY MMD Access Control Register (PHY_MMD_ACCESS) and PHY MMD Access Address/Data Register (PHY_MMD_ADDR_DATA). The supported MMD device addresses are 3 (PCS), 7 (Auto-Negotiation), and 30 (Vendor Specific). Table 12-15, "MMD Registers" details the supported registers within each MMD device.

TABLE 12-15: MMD REGISTERS

MMD DEVICE ADDRESS (IN DECIMAL)	INDEX (IN DECIMAL)	REGISTER NAME
3 (PCS)	0	PHY PCS Control 1 Register (PHY_PCS_CTL1)
	1	PHY PCS Status 1 Register (PHY_PCS_STAT1)
	5	PHY PCS MMD Devices Present 1 Register (PHY_PCS_MMD_PRESENT1)
	6	PHY PCS MMD Devices Present 2 Register (PHY_PCS_MMD_PRESENT2)
	20	PHY EEE Capability Register (PHY_EEE_CAP)
	22	PHY EEE Wake Error Register (PHY_EEE_WAKE_ERR)
7 (Auto-Negotiation)	5	PHY Auto-Negotiation MMD Devices Present 1 Register (PHY_AN_MMD_PRESENT1)
	6	PHY Auto-Negotiation MMD Devices Present 2 Register (PHY_AN_MMD_PRESENT2)
	60	PHY EEE Advertisement Register (PHY_EEE_ADV)
	61	PHY EEE Link Partner Advertisement Register (PHY_EEE_LP_ADV)
30 (Vendor Specific)	2	PHY Vendor Specific MMD 1 Device ID 1 Register (PHY_VEND_SPEC_MMD1_DEVID1)
	3	PHY Vendor Specific MMD 1 Device ID 2 Register (PHY_VEND_SPEC_MMD1_DEVID2)
	5	PHY Vendor Specific MMD 1 Devices Present 1 Register (PHY_VEND_SPEC_MMD1_PRESENT1)
	6	PHY Vendor Specific MMD 1 Devices Present 2 Register (PHY_VEND_SPEC_MMD1_PRESENT2)
	8	PHY Vendor Specific MMD 1 Status Register (PHY_VEND_SPEC_MMD1_STAT)
	14	PHY Vendor Specific MMD 1 Package ID 1 Register (PHY_VEND_SPEC_MMD1_PKG_ID1)
	15	PHY Vendor Specific MMD 1 package ID 2 Register (PHY_VEND_SPEC_MMD1_PKG_ID2)

To read or write an MMD register, the following procedure must be observed:

1. Write the PHY MMD Access Control Register (PHY_MMD_ACCESS) with 00b (address) for the MMD Function field and the desired MMD device (3 for PCS, 7 for Auto-Negotiation) for the MMD Device Address (DEVAD) field.
2. Write the PHY MMD Access Address/Data Register (PHY_MMD_ADDR_DATA) with the 16-bit address of the desired MMD register to read/write within the previously selected MMD device (PCS or Auto-Negotiation).

3. Write the PHY MMD Access Control Register (PHY_MMD_ACCESS) with 01b (data) for the MMD Function field and choose the previously selected MMD device (3 for PCS, 7 for Auto-Negotiation) for the MMD Device Address (DEVAD) field.
4. If reading, read the PHY MMD Access Address/Data Register (PHY_MMD_ADDR_DATA), which contains the selected MMD register contents. If writing, write the PHY MMD Access Address/Data Register (PHY_MMD_ADDR_DATA) with the register contents intended for the previously selected MMD register.

Unless otherwise specified, reserved fields must be written with zeros if the register is written.

12.2.18.23 PHY PCS Control 1 Register (PHY_PCS_CTL1)

Index (In Decimal): 3.0 Size: 16 bits

Bits	Description	Type	Default
15:11	**RESERVED**	RO	-
10	**Clock Stop Enable** **0 = The PHY cannot stop the clock during Low Power Idle (LPI)** **1 = The PHY may stop the clock during LPI** **Note:** This bit has no affect since the device does not support this mode.	R/W	0b
9:0	**RESERVED**	RO	-

12.2.18.24 PHY PCS Status 1 Register (PHY_PCS_STAT1)

Index (In Decimal): 3.1 Size: 16 bits

Bits	Description	Type	Default
15:12	**RESERVED**	RO	-
11	**TX LPI Received** **0 = TX PCS has not received LPI** **1 = TX PCS has received LPI**	RO/LH	0b
10	**RX LPI Received** **0 = RX PCS has not received LPI** **1 = RX PCS has received LPI**	RO/LH	0b
9	**TX LPI Indication** **0 = TX PCS is not currently receiving LPI** **1 = TX PCS is currently receiving LPI**	RO	0b
8	**RX LPI Indication** **0 = RX PCS is not currently receiving LPI** **1 = RX PCS is currently receiving LPI**	RO	0b
7	**RESERVED**	RO	-
6	**Clock Stop Capable** **0 = The MAC cannot stop the clock during Low Power Idle (LPI)** **1 = The MAC may stop the clock during LPI** **Note:** The device does not support this mode.	RO	0b
5:0	**RESERVED**	RO	-

12.2.18.25 PHY PCS MMD Devices Present 1 Register (PHY_PCS_MMD_PRESENT1)

Index (In Decimal): 3.5 Size: 16 bits

Bits	Description	Type	Default
15:8	**RESERVED**	RO	-
7	**Auto-Negotiation Present** **0 = Auto-negotiation not present in package** **1 = Auto-negotiation present in package**	RO	1b
6	**TC Present** **0 = TC not present in package** **1 = TC present in package**	RO	0b
5	**DTE XS Present** **0 = DTE XS not present in package** **1 = DTE XS present in package**	RO	0b
4	**PHY XS Present** **0 = PHY XS not present in package** **1 = PHY XS present in package**	RO	0b
3	**PCS Present** **0 = PCS not present in package** **1 = PCS present in package**	RO	1b
2	**WIS Present** **0 = WIS not present in package** **1 = WIS present in package**	RO	0b
1	**PMD/PMA Present** **0 = PMD/PMA not present in package** **1 = PMD/PMA present in package**	RO	0b
0	**Clause 22 Registers Present** **0 = Clause 22 registers not present in package** **1 = Clause 22 registers present in package**	RO	0b

12.2.18.26 PHY PCS MMD Devices Present 2 Register (PHY_PCS_MMD_PRESENT2)

Index (In Decimal): 3.6 Size: 16 bits

Bits	Description	Type	Default
15	**Vendor Specific Device 2 Present** **0 = Vendor specific device 2 not present in package** **1 = Vendor specific device 2 present in package**	RO	0b
14	**Vendor Specific Device 1 Present** **0 = Vendor specific device 1 not present in package** **1 = Vendor specific device 1 present in package**	RO	1b
13	**Clause 22 Extension Present** **0 = Clause 22 extension not present in package** **1 = Clause 22 extension present in package**	RO	0b
12:0	**RESERVED**	RO	-

12.2.18.27 PHY EEE Capability Register (PHY_EEE_CAP)

Index (In Decimal): 3.20 Size: 16 bits

Bits	Description	Type	Default
15:7	**RESERVED**	RO	-
6	**10GBASE-KR EEE** **0 = EEE is not supported for 10GBASE-KR** **1 = EEE is supported for 10GBASE-KR** **Note:** The device does not support this mode.	RO	0b
5	**10GBASE-KX4 EEE** **0 = EEE is not supported for 10GBASE-KX4** **1 = EEE is supported for 10GBASE-KX4** **Note:** The device does not support this mode.	RO	0b
4	**10GBASE-KX EEE** **0 = EEE is not supported for 10GBASE-KX** **1 = EEE is supported for 10GBASE-KX** **Note:** The device does not support this mode.	RO	0b
3	**10GBASE-T EEE** **0 = EEE is not supported for 10GBASE-T** **1 = EEE is supported for 10GBASE-T** **Note:** The device does not support this mode.	RO	0b
2	**1000BASE-T EEE** **0 = EEE is not supported for 1000BASE-T** **1 = EEE is supported for 1000BASE-T** **Note:** The device does not support this mode.	RO	0b
1	**100BASE-TX EEE** **0 = EEE is not supported for 100BASE-TX** **1 = EEE is supported for 100BASE-TX**	RO	Note 21
0	**RESERVED**	RO	-

Note 21: The default value of this field is determined by the value of the PHY Energy Efficient Ethernet Enable (PHYEEEEN) of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG) on page 247. If PHY Energy Efficient Ethernet Enable (PHYEEEEN) is 0b, this field is 0b and 100BASE-TX EEE capability is not supported. If PHY Energy Efficient Ethernet Enable (PHYEEEEN) is 1b, then this field is 1b and 100BASE-TX EEE capability is supported.

12.2.18.28 PHY EEE Wake Error Register (PHY_EEE_WAKE_ERR)

Index (In Decimal): 3.22 Size: 16 bits

Bits	Description	Type	Default
15:0	**EEE Wake Error Counter** This counter is cleared to zeros on read and is held to all ones on overflow.	RO/RC	0000h

12.2.18.29 PHY Auto-Negotiation MMD Devices Present 1 Register (PHY_AN_MMD_PRESENT1)

Index (In Decimal): 7.5 Size: 16 bits

Bits	Description	Type	Default
15:8	**RESERVED**	RO	-
7	**Auto-Negotiation Present** **0 = Auto-negotiation not present in package** **1 = Auto-negotiation present in package**	RO	1b
6	**TC Present** **0 = TC not present in package** **1 = TC present in package**	RO	0b
5	**DTE XS Present** **0 = DTE XS not present in package** **1 = DTE XS present in package**	RO	0b
4	**PHY XS Present** **0 = PHY XS not present in package** **1 = PHY XS present in package**	RO	0b
3	**PCS Present** **0 = PCS not present in package** **1 = PCS present in package**	RO	1b
2	**WIS Present** **0 = WIS not present in package** **1 = WIS present in package**	RO	0b
1	**PMD/PMA Present** **0 = PMD/PMA not present in package** **1 = PMD/PMA present in package**	RO	0b
0	**Clause 22 Registers Present** **0 = Clause 22 registers not present in package** **1 = Clause 22 registers present in package**	RO	0b

12.2.18.30 PHY Auto-Negotiation MMD Devices Present 2 Register (PHY_AN_MMD_PRESENT2)

Index (In Decimal): 7.6 Size: 16 bits

Bits	Description	Type	Default
15	**Vendor Specific Device 2 Present** **0 = Vendor specific device 2 not present in package** **1 = Vendor specific device 2 present in package**	RO	0b
14	**Vendor Specific Device 1 Present** **0 = Vendor specific device 1 not present in package** **1 = Vendor specific device 1 present in package**	RO	1b
13	**Clause 22 Extension Present** 0 = Clause 22 extension not present in package 1 = Clause 22 extension present in package	RO	0b
12:0	**RESERVED**	RO	-

12.2.18.31 PHY EEE Advertisement Register (PHY_EEE_ADV)

Index (In Decimal): 7.60 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:2	**RESERVED**	RO	-
1	**100BASE-TX EEE** **0 = Do not advertise EEE capability for 100BASE-TX.** **1 = Advertise EEE capability for 100BASE-TX.**	Note 22	Note 23
0	**RESERVED**	RO	-

Note 22: This bit is read/write (R/W). However, the user must not set this bit if EEE is disabled.

Note 23: The default value of this field is determined by the value of the PHY Energy Efficient Ethernet Enable (PHYEEEEN) of the PHYEDPD NLP / Crossover Time / EEE Configuration Register (PHY_EDPD_CFG) on page 247. If PHY Energy Efficient Ethernet Enable (PHYEEEEN) is 0b, this field is 0b and 100BASE-TX EEE capability is not advertised. If PHY Energy Efficient Ethernet Enable (PHYEEEEN) is 1b, then this field is 1b and 100BASE-TX EEE capability is advertised.

12.2.18.32 PHY EEE Link Partner Advertisement Register (PHY_EEE_LP_ADV)

Index (In Decimal): 7.61 Size: 16 bits

BITS	DESCRIPTION	TYPE	DEFAULT
15:7	**RESERVED**	RO	-
6	**10GBASE-KR EEE** **0 = Link partner does not advertise EEE capability for 10GBASE-KR.** **1 = Link partner advertises EEE capability for 10GBASE-KR.** **Note:** This device does not support this mode.	RO	0b
5	**10GBASE-KX4 EEE** **0 = Link partner does not advertise EEE capability for 10GBASE-KX4.** **1 = Link partner advertises EEE capability for 10GBASE-KX4.** **Note:** This device does not support this mode.	RO	0b
4	**10GBASE-KX EEE** **0 = Link partner does not advertise EEE capability for 10GBASE-KX.** **1 = Link partner advertises EEE capability for 10GBASE-KX.** **Note:** This device does not support this mode.	RO	0b
3	**10GBASE-T EEE** **0 = Link partner does not advertise EEE capability for 10GBASE-T.** **1 = Link partner advertises EEE capability for 10GBASE-T.** **Note:** This device does not support this mode.	RO	0b
2	**1000BASE-T EEE** **0 = Link partner does not advertise EEE capability for 1000BASE-T.** **1 = Link partner advertises EEE capability for 1000BASE-T.** **Note:** This device does not support this mode.	RO	0b
1	**100BASE-TX EEE** **0 = Link partner does not advertise EEE capability for 100BASE-TX.** **1 = Link partner advertises EEE capability for 100BASE-TX.**	RO	0b
0	**RESERVED**	RO	-

12.2.18.33 PHY Vendor Specific MMD 1 Device ID 1 Register (PHY_VEND_SPEC_MMD1_DEVID1)

Index (In Decimal): 30.2 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	0000h

12.2.18.34 PHY Vendor Specific MMD 1 Device ID 2 Register (PHY_VEND_SPEC_MMD1_DEVID2)

Index (In Decimal): 30.3 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	0000h

12.2.18.35 PHY Vendor Specific MMD 1 Devices Present 1 Register (PHY_VEND_SPEC_MMD1_PRESENT1)

Index (In Decimal): 30.5 Size: 16 bits

Bits	Description	Type	Default
15:8	**RESERVED**	RO	-
7	**Auto-Negotiation Present** **0 = Auto-negotiation not present in package** **1 = Auto-negotiation present in package**	RO	1b
6	**TC Present** **0 = TC not present in package** **1 = TC present in package**	RO	0b
5	**DTE XS Present** **0 = DTE XS not present in package** **1 = DTE XS present in package**	RO	0b
4	**PHY XS Present** **0 = PHY XS not present in package** **1 = PHY XS present in package**	RO	0b
3	**PCS Present** **0 = PCS not present in package** **1 = PCS present in package**	RO	1b
2	**WIS Present** **0 = WIS not present in package** **1 = WIS present in package**	RO	0b
1	**PMD/PMA Present** **0 = PMD/PMA not present in package** **1 = PMD/PMA present in package**	RO	0b
0	**Clause 22 Registers Present** **0 = Clause 22 registers not present in package** **1 = Clause 22 registers present in package**	RO	0b

12.2.18.36 PHY Vendor Specific MMD 1 Devices Present 2 Register (PHY_VEND_SPEC_MMD1_PRESENT2)

Index (In Decimal): 30.6 Size: 16 bits

Bits	Description	Type	Default
15	**Vendor Specific Device 2 Present** **0 = Vendor specific device 2 not present in package** **1 = Vendor specific device 2 present in package**	RO	0b
14	**Vendor Specific Device 1 Present** **0 = Vendor specific device 1 not present in package** **1 = Vendor specific device 1 present in package**	RO	1b
13	**Clause 22 Extension Present** **0 = Clause 22 extension not present in package** **1 = Clause 22 extension present in package**	RO	0b
12:0	**RESERVED**	RO	-

12.2.18.37 PHY Vendor Specific MMD 1 Status Register (PHY_VEND_SPEC_MMD1_STAT)

Index (In Decimal): 30.8 Size: 16 bits

Bits	Description	Type	Default
15:14	**Device Present** **00 = No device responding at this address** **01 = No device responding at this address** **10 = Device responding at this address** **11 = No device responding at this address**	RO	10b
13:0	**RESERVED**	RO	-

12.2.18.38 PHY Vendor Specific MMD 1 Package ID 1 Register (PHY_VEND_SPEC_MMD1_PKG_ID1)

Index (In Decimal): 30.14 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	0000h

12.2.18.39 PHY Vendor Specific MMD 1 package ID 2 Register (PHY_VEND_SPEC_MMD1_PKG_ID2)

Index (In Decimal): 30.15 Size: 16 bits

Bits	Description	Type	Default
15:0	**RESERVED**	RO	0000h

.

13.0 I²C MASTER EEPROM CONTROLLER

13.1 Functional Overview

This chapter details the EEPROM I²C master and EEPROM Loader provided by the device. The I²C EEPROM controller is an I²C master module which interfaces an optional external EEPROM with the system register bus and the EEPROM Loader. Multiple sizes of external EEPROMs are supported. Configuration of the EEPROM size is accomplished via the eeprom_size_strap configuration strap. Various commands are supported for EEPROM access, allowing for the storage and retrieval of static data. The I²C interface conforms to the NXP *I²C-Bus Specification*.

The EEPROM Loader provides the automatic loading of configuration settings from the EEPROM into the device at reset. The EEPROM Loader module interfaces to the EEPROM Controller, Ethernet PHYs and the system CSRs.

13.2 I²C Overview

I²C is a bi-directional 2-wire data protocol. A device that sends data is defined as a transmitter and a device that receives data is defined as a receiver. The bus is controlled by a master which generates the **EESCL** clock, controls bus access and generates the start and stop conditions. Either a master or slave may operate as a transmitter or receiver as determined by the master.

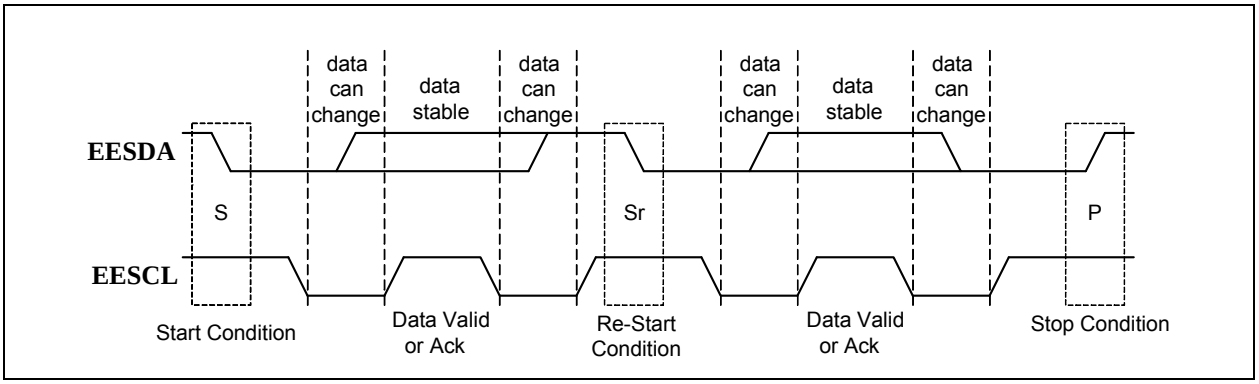
Both the clock (**EESCL**) and data (**EESDA**) signals have digital input filters that reject pulses that are less than 100 ns. The data pin is driven low when either interface sends a low, emulating the wired-AND function of the I²C bus.

The following bus states exist:

- **Idle:** Both **EESDA** and **EESCL** are high when the bus is idle.
- **Start & Stop Conditions:** A start condition is defined as a high to low transition on the **EESDA** line while **EESCL** is high. A stop condition is defined as a low to high transition on the **EESDA** line while **EESCL** is high. The bus is considered to be busy following a start condition and is considered free 4.7 µs/1.3 µs (for 100 kHz and 400 kHz operation, respectively) following a stop condition. The bus stays busy following a repeated start condition (instead of a stop condition). Starts and repeated starts are otherwise functionally equivalent.
- **Data Valid:** Data is valid, following the start condition, when **EESDA** is stable while **EESCL** is high. Data can only be changed while the clock is low. There is one valid bit per clock pulse. Every byte must be 8 bits long and is transmitted MSB first.
- **Acknowledge:** Each byte of data is followed by an acknowledge bit. The master generates a ninth clock pulse for the acknowledge bit. The transmitter releases **EESDA** (high). The receiver drives **EESDA** low so that it remains valid during the high period of the clock, taking into account the setup and hold times. The receiver may be the master or the slave depending on the direction of the data. Typically the receiver acknowledges each byte. If the master is the receiver, it does not generate an acknowledge on the last byte of a transfer. This informs the slave to not drive the next byte of data so that the master may generate a stop or repeated start condition.

Figure 13-1 displays the various bus states of a typical I²C cycle.

FIGURE 13-1: I²C CYCLE



13.3 I²C Master EEPROM Controller

The I²C EEPROM controller supports I²C compatible EEPROMs.

Note: When the EEPROM Loader is running, it has exclusive use of the I²C EEPROM controller. Refer to Section 13.4, "EEPROM Loader" for more information.

The I²C master implements a low level serial interface (start and stop condition generation, data bit transmission and reception, acknowledge generation and reception) for connection to I²C EEPROMs and consists of a data wire (**EESDA**) and a serial clock (**EESCL**). The serial clock is driven by the master, while the data wire is bi-directional. Both signals are open-drain and require external pull-up resistors.

The I²C master interface runs at the standard-mode rate of 100 kHz. I²C master interface timing information is detailed in Figure 13-2 and Table 13-1.

FIGURE 13-2: I²C MASTER TIMING

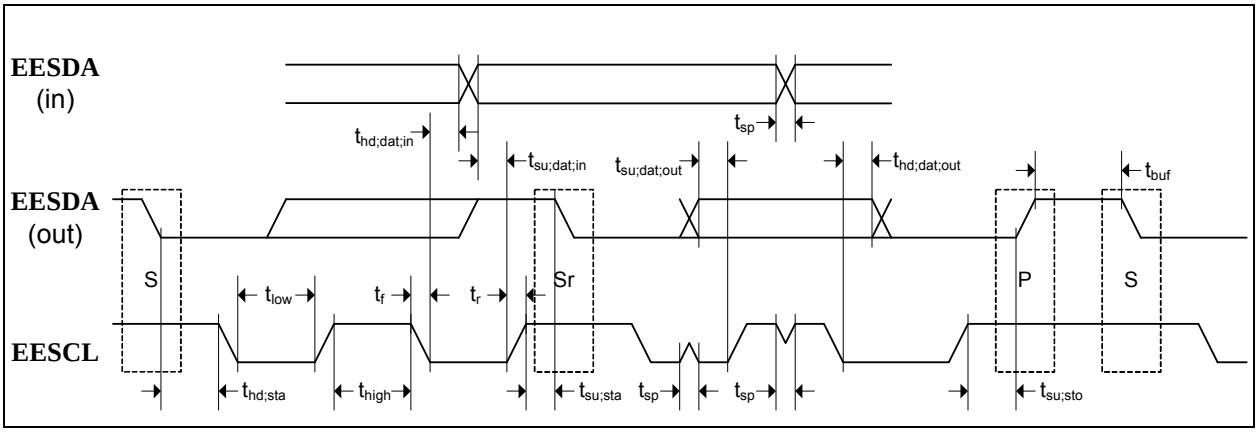


TABLE 13-1: I²C MASTER TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
f_{scl}	**EESCL** clock frequency			100	kHz
t_{high}	**EESCL** high time	4.0			µs
t_{low}	**EESCL** low time	4.7			µs
t_r	Rise time of **EESDA** and **EESCL**			1000	ns
t_f	Fall time of **EESDA** and **EESCL**			300	ns
$t_{su;sta}$	Setup time (provided to slave) of **EESCL** high before **EESDA** output falling for repeated start condition	5.2 Note 1			µs
$t_{hd;sta}$	Hold time (provided to slave) of **EESCL** after **EESDA** output falling for start or repeated start condition	4.5 Note 1			µs
$t_{su;dat;in}$	Setup time (from slave) **EESDA** input before **EESCL** rising	200 Note 2			ns
$t_{hd;dat;in}$	Hold time (from slave) of **EESDA** input after **EESCL** falling	0			ns
$t_{su;dat;out}$	Setup time (provided to slave) **EESDA** output before **EESCL** rising	1250 Note 3			ns

TABLE 13-1: I²C MASTER TIMING VALUES (CONTINUED)

Symbol	Description	Min	Typ	Max	Units
$t_{hd;dat;out}$	Hold time (provided to slave) of **EESDA** output after **EESCL** falling	1000 Note 3			ns
$t_{su;sto}$	Setup time (provided to slave) of **EESCL** high before **EESDA** output rising for stop condition	4.5 Note 1			µs
t_{buf}	Bus free time	4.7			µs
t_{sp}	Input spike suppression on **EESCL** and **EESDA**			100	ns

Note 1: These values provide 500 ns of margin compared to the I²C specification.

Note 2: This value provides 50 ns of margin compared to the I²C specification.

Note 3: These values provide 1000 ns of margin compared to the I²C specification.

Based on the eeprom_size_strap configuration strap, various sized I²C EEPROMs are supported. The varying size ranges are supported by additional bits in the EEPROM Controller Address (EPC_ADDRESS) field of the EEPROM Command Register (E2P_CMD). Within each size range, the largest EEPROM uses all the address bits, while the smaller EEPROMs treat the upper address bits as don't cares. The EEPROM controller drives all the address bits as requested regardless of the actual size of the EEPROM. The supported size ranges for I²C operation are shown in Table 13-2.

TABLE 13-2: I²C EEPROM SIZE RANGES

eeprom_size_strap	# of Address Bytes	EEPROM Size	EEPROM Types
0	1 (Note 4)	128 x 8 through 2048 x 8	24xx01, 24xx02, 24xx04, 24xx08, 24xx16
1	2	4096 x 8 through 65536 x 8	24xx32, 24xx64, 24xx128, 24xx256, 24xx512

Note 4: Bits in the control byte are used as the upper address bits.

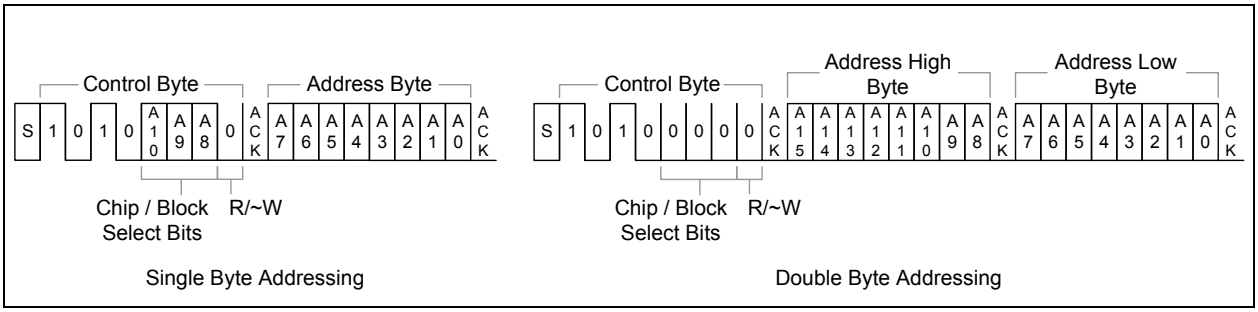
13.3.1 I²C EEPROM DEVICE ADDRESSING

The I²C EEPROM is addressed for a read or write operation by first sending a control byte followed by the address byte or bytes. The control byte is preceded by a start condition. The control byte and address byte(s) are each acknowledged by the EEPROM slave. If the EEPROM slave fails to send an acknowledge, then the sequence is aborted (a start condition and a stop condition are sent) and the EEPROM Controller Timeout (EPC_TIMEOUT) bit of the EEPROM Command Register (E2P_CMD) is set.

The control byte consists of a 4 bit control code, 3 bits of chip/block select and one direction bit. The control code is 1010b. For single byte addressing EEPROMs, the chip/block select bits are used for address bits 10, 9 and 8. For double byte addressing EEPROMs, the chip/block select bits are set low. The direction bit is set low to indicate the address is being written.

Figure 13-3 illustrates a typical I2C EEPROM addressing bit order for single and double byte addressing.

FIGURE 13-3: I2C EEPROM ADDRESSING

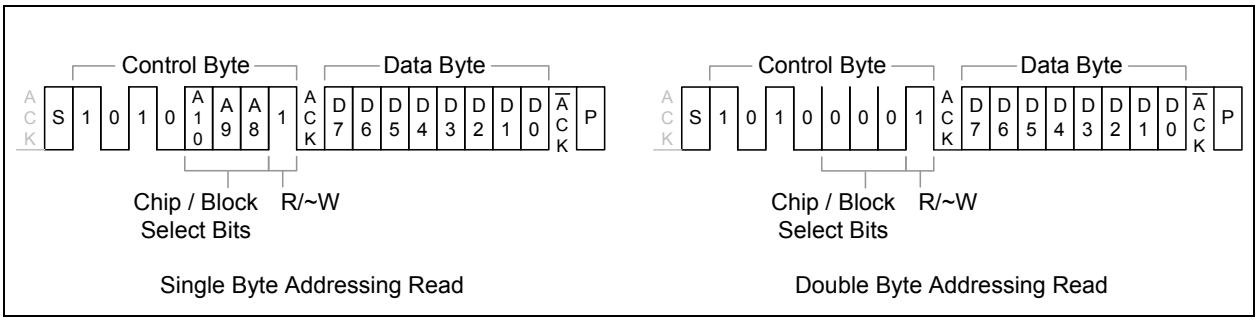


13.3.2 I2C EEPROM BYTE READ

Following the device addressing, a data byte may be read from the EEPROM by outputting a start condition and control byte with a control code of 1010b, chip/block select bits as described in Section 13.3.1 and the R/~W bit high. The EEPROM will respond with an acknowledge, followed by 8 bits of data. If the EEPROM slave fails to send an acknowledge, then the sequence is aborted (a start condition and a stop condition are sent) and the EEPROM Controller Timeout (EPC_TIMEOUT) bit in the EEPROM Command Register (E2P_CMD) is set. The I2C master then sends a no-acknowledge, followed by a stop condition.

Figure 13-4 illustrates a typical I2C EEPROM byte read for single and double byte addressing.

FIGURE 13-4: I2C EEPROM BYTE READ



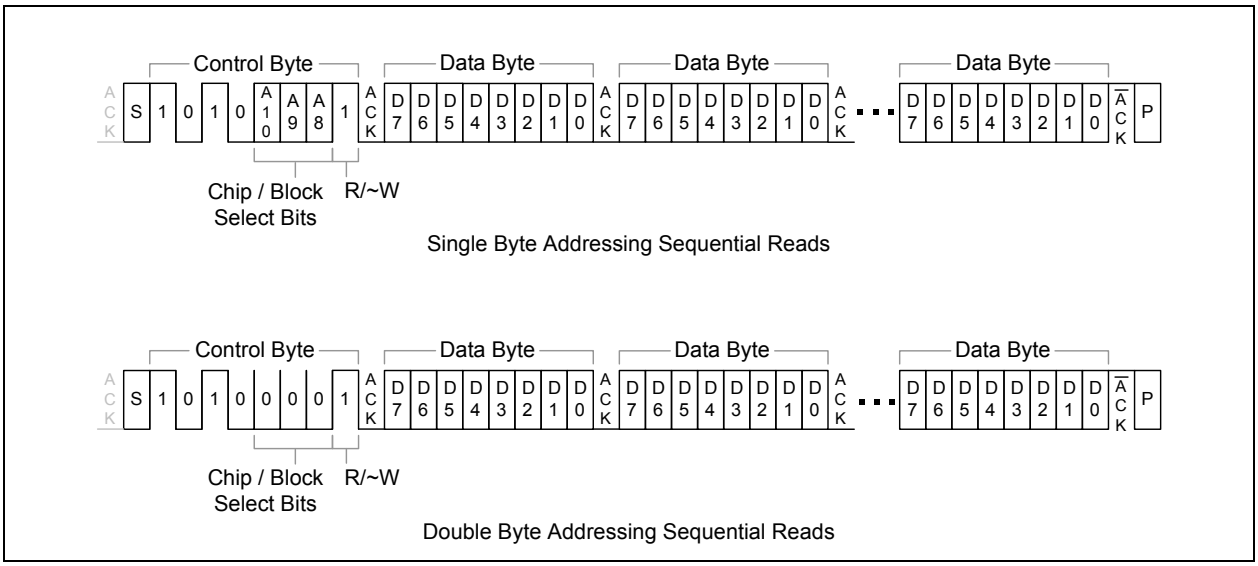
For a register level description of a read operation, refer to Section 13.3.7, "I2C Master EEPROM Controller Operation," on page 288.

13.3.3 I2C EEPROM SEQUENTIAL BYTE READS

Following the device addressing, data bytes may be read sequentially from the EEPROM by outputting a start condition and control byte with a control code of 1010b, chip/block select bits as described in Section 13.3.1 and the R/~W bit high. The EEPROM will respond with an acknowledge, followed by 8 bits of data. If the EEPROM slave fails to send an acknowledge, then the sequence is aborted (a start condition and a stop condition are sent) and the EEPROM Controller Timeout (EPC_TIMEOUT) bit in the EEPROM Command Register (E2P_CMD) is set. The I2C master then sends an acknowledge and the EEPROM responds with the next 8 bits of data. This continues until the last desired byte is read, at which point the I2C master sends a no-acknowledge (instead of the acknowledge), followed by a stop condition.

Figure 13-5 illustrates a typical I2C EEPROM sequential byte reads for single and double byte addressing.

FIGURE 13-5: I2C EEPROM SEQUENTIAL BYTE READS



Sequential reads are used by the EEPROM Loader. Refer to Section 13.4, "EEPROM Loader" for additional information.

For a register level description of a read operation, refer to Section 13.3.7, "I2C Master EEPROM Controller Operation," on page 288.

13.3.4 I2C EEPROM BYTE WRITES

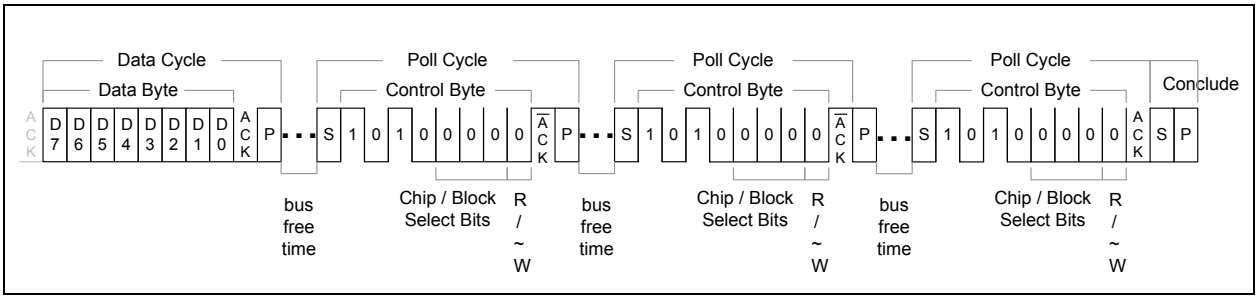
Following the device addressing, a data byte may be written to the EEPROM by outputting the data after receiving the acknowledge from the EEPROM. The data byte is acknowledged by the EEPROM slave and the I2C master finishes the write cycle with a stop condition. If the EEPROM slave fails to send an acknowledge, then the sequence is aborted (a start condition and a stop condition are sent) and the EEPROM Controller Timeout (EPC_TIMEOUT) bit in the EEPROM Command Register (E2P_CMD) is set.

Following the data byte write cycle, the I2C master will poll the EEPROM to determine when the byte write is finished. After meeting the minimum bus free time, a start condition is sent followed by a control byte with a control code of 1010b, chip/block select bits low (since they are don't cares) and the R/~W bit low. If the EEPROM is finished with the byte write, it will respond with an acknowledge. Otherwise, it will respond with a no-acknowledge and the I2C master will issue a stop and repeat the poll. If the acknowledge does not occur within 30 ms, a timeout occurs (a start condition and a stop condition are sent) and the EEPROM Controller Timeout (EPC_TIMEOUT) bit in the EEPROM Command Register (E2P_CMD) is set. The check for timeout is only performed following each no-acknowledge, since it may be possible that the EEPROM write finished before the timeout but the 30 ms expired before the poll was performed (due to the bus being used by another master).

Once the I2C master receives the acknowledge, it concludes by sending a start condition, followed by a stop condition, which will place the EEPROM into standby.

Figure 13-6 illustrates a typical I2C EEPROM byte write.

FIGURE 13-6: I2C EEPROM BYTE WRITE



For a register level description of a write operation, refer to Section 13.3.7, "I2C Master EEPROM Controller Operation," on page 288.

13.3.5 WAIT STATE GENERATION

The serial clock is also used as an input as it can be held low by the slave device in order to wait-state the data cycle. Once the slave has data available or is ready to receive, it will release the clock. Assuming the masters clock low time is also expired, the clock will rise and the cycle will continue. If the slave device holds the clock low for more than 30 ms, the current command sequence is aborted (a start condition and a stop condition are not sent since the clock is being held low, instead the clock and data lines are just released) and the EEPROM Controller Timeout (EPC_TIMEOUT) bit in the EEPROM Command Register (E2P_CMD) is set.

13.3.6 I^2C BUS ARBITRATION AND CLOCK SYNCHRONIZATION

Since the I^2C master and the I^2C slave serial interfaces share common pins, there are at least two master I^2C devices on the bus (the device and the Host). There exists the potential that both masters try to access the bus at the same time. The I^2C specification handles this situation with three mechanisms: bus busy, clock synchronization and bus arbitration.

Note: The timing parameters referred to in the following subsections refer to the detailed timing information presented in the NXP *I^2C-Bus Specification*.

13.3.6.1 Bus Busy

A master may start a transfer only if the bus is not busy. The bus is considered to be busy after the START condition and is considered to be free again t_{buf} time after the STOP condition. The standard mode value of 4.7 µs is used for t_{buf} since the EEPROM master runs at the standard mode rate. Following reset, it is unknown if the bus is actually busy, since the START condition may have been missed. Therefore, following reset, the bus is initially considered busy and is considered free t_{buf} time after the STOP condition or if clock and data are seen high for 4 ms.

13.3.6.2 Clock Synchronization

Clock synchronization is used, since both masters may be generating different clock frequencies. When the clock is driven low by one master, each other active master will restart its low timer and also drive the clock low. Each master will drive the clock low for its minimum low time and then release it. The clock line will not go high until all masters have released it. The slowest master therefore determines the actual low time. Devices with shorter low timers will wait. Once the clock goes high, each master will start its high timer. The first master to reach its high time will once again drive the clock low. The fastest master therefore determines the actual high time. The process then repeats. Clock synchronization is similar to the cycle stretching that can be done by a slave device, with the exception that a slave device can only extend the low time of the clock. It can not cause the falling edge of the clock.

13.3.6.3 Arbitration

Arbitration involves testing the input data vs. the output data, when the clock goes high, to see if they match. Since the data line is wired-AND'ed, a master transmitting a high value will see a mismatch if another master is transmitting a low value. The comparison is not done when receiving bits from the slave. Arbitration starts with the control byte and, if both masters are accessing the same slave, can continue into address and data bits (for writes) or acknowledge bits (for reads). If desired, a master that loses arbitration can continue to generate clock pulses until the end of the loosing byte (note that the ACK on a read is considered the end of the byte) but the losing master may no longer drive any data bits.

It is not permitted for another master to access the EEPROM while the device is using it during startup or due to an EEPROM command. The other master should wait sufficient time or poll the device to determine when the EEPROM is available. This restriction simplifies the arbitration and access process since arbitration will always be resolved when transmitting the 8 control bits during the device addressing or during the Poll Cycles.

If arbitration is lost during the device addressing, the I^2C master will return to the beginning of the device addressing sequence and wait for the bus to become free.

If arbitration is lost during a Poll Cycle, the I^2C master will return to the beginning of the Poll Cycle sequence and wait for the bus to become free. Note that in this case the 30 ms timeout-counter should not be reset. If the 30 ms timeout should expire while waiting for the bus to become free, the sequence should not abort without first completing a final poll (with the exception of the busy / arbitration timeout described in Section 13.3.6.4).

13.3.6.4 Timeout Due to Busy or Arbitration

It is possible for another master to monopolize the bus (due to a continual bus busy or more successful arbitration). If successful arbitration is not achieved within 1.92 s from the start of the read or write request or from the start of the Poll Cycle, the command sequence or Poll Cycle is aborted and the EEPROM Controller Timeout (EPC_TIMEOUT) bit in the EEPROM Command Register (E2P_CMD) is set. Note that this is a total timeout value and not the timeout for any one portion of the sequence.

13.3.7 I²C MASTER EEPROM CONTROLLER OPERATION

I²C master EEPROM operations are performed using the EEPROM Command Register (E2P_CMD) and EEPROM Data Register (E2P_DATA).

The following operations are supported:

- READ (Read Location)
- WRITE (Write Location)
- RELOAD (EEPROM Loader Reload - See Section 13.4, "EEPROM Loader")

Note: The EEPROM Loader uses the READ command only.

The supported commands are detailed in Section 13.5.1, "EEPROM Command Register (E2P_CMD)," on page 295. Details specific to each operational mode are explained in Section 13.2, "I2C Overview," on page 282 and Section 13.4, "EEPROM Loader", respectively.

When issuing a WRITE command, the desired data must first be written into the EEPROM Data Register (E2P_DATA). The WRITE command may then be issued by setting the EEPROM Controller Command (EPC_COMMAND) field of the EEPROM Command Register (E2P_CMD) to the desired command value. If the operation is a WRITE, the EEPROM Controller Address (EPC_ADDRESS) field in the EEPROM Command Register (E2P_CMD) must also be set to the desired location. The command is executed when the EEPROM Controller Busy (EPC_BUSY) bit of the EEPROM Command Register (E2P_CMD) is set. The completion of the operation is indicated when the EEPROM Controller Busy (EPC_BUSY) bit is cleared.

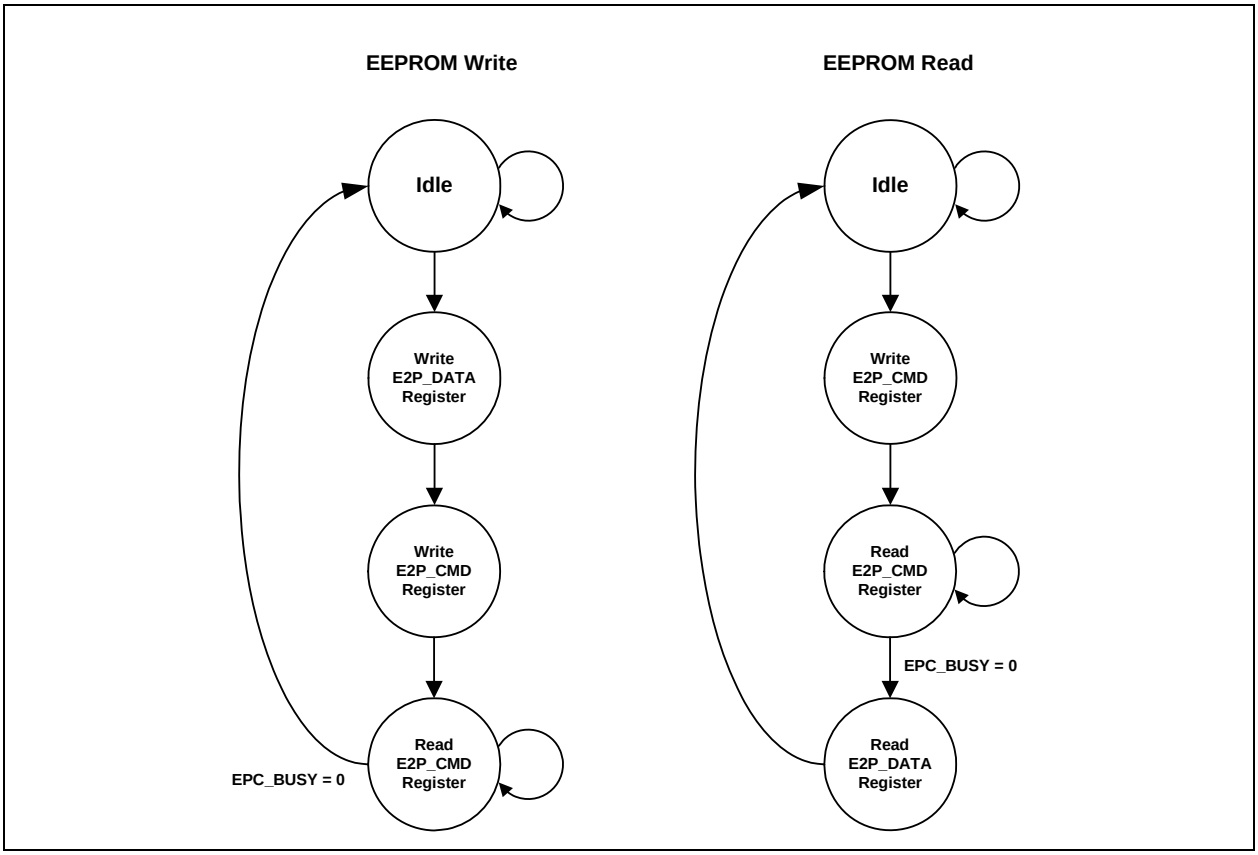
When issuing a READ command, the EEPROM Controller Command (EPC_COMMAND) and EEPROM Controller Address (EPC_ADDRESS) fields of the EEPROM Command Register (E2P_CMD) must be configured with the desired command value and the read address, respectively. The READ command is executed by setting the EEPROM Controller Busy (EPC_BUSY) bit of the EEPROM Command Register (E2P_CMD). The completion of the operation is indicated when the EEPROM Controller Busy (EPC_BUSY) bit is cleared, at which time the data from the EEPROM may be read from the EEPROM Data Register (E2P_DATA).

The RELOAD operation is performed by writing the RELOAD command into the EEPROM Controller Command (EPC_COMMAND) field of the EEPROM Command Register (E2P_CMD). The command is executed by setting the EEPROM Controller Busy (EPC_BUSY) bit of the EEPROM Command Register (E2P_CMD). In all cases, the software must wait for the EEPROM Controller Busy (EPC_BUSY) bit to clear before modifying the EEPROM Command Register (E2P_CMD).

If an operation is attempted and the EEPROM device does not respond within 30 ms, the device will timeout and the EEPROM Controller Timeout (EPC_TIMEOUT) bit of the EEPROM Command Register (E2P_CMD) will be set.

Figure 13-7 illustrates the process required to perform an EEPROM read or write operation.

FIGURE 13-7: EEPROM ACCESS FLOW DIAGRAM



13.4 EEPROM Loader

The EEPROM Loader interfaces to the I^2C EEPROM controller, the PHYs and to the system CSRs (via the Register Access MUX). All system CSRs are accessible to the EEPROM Loader.

The EEPROM Loader runs upon a pin reset (**RST#**), power-on reset (POR), digital reset (Digital Reset (DIGITAL_RST) bit in the Reset Control Register (RESET_CTL)) or upon the issuance of a RELOAD command via the EEPROM Command Register (E2P_CMD). Refer to Section 6.2, "Resets," on page 38 for additional information on resets.

The EEPROM contents must be loaded in a specific format for use with the EEPROM Loader. An overview of the EEPROM content format is shown in Table 13-3. Each section of EEPROM contents is discussed in detail in the following sections.

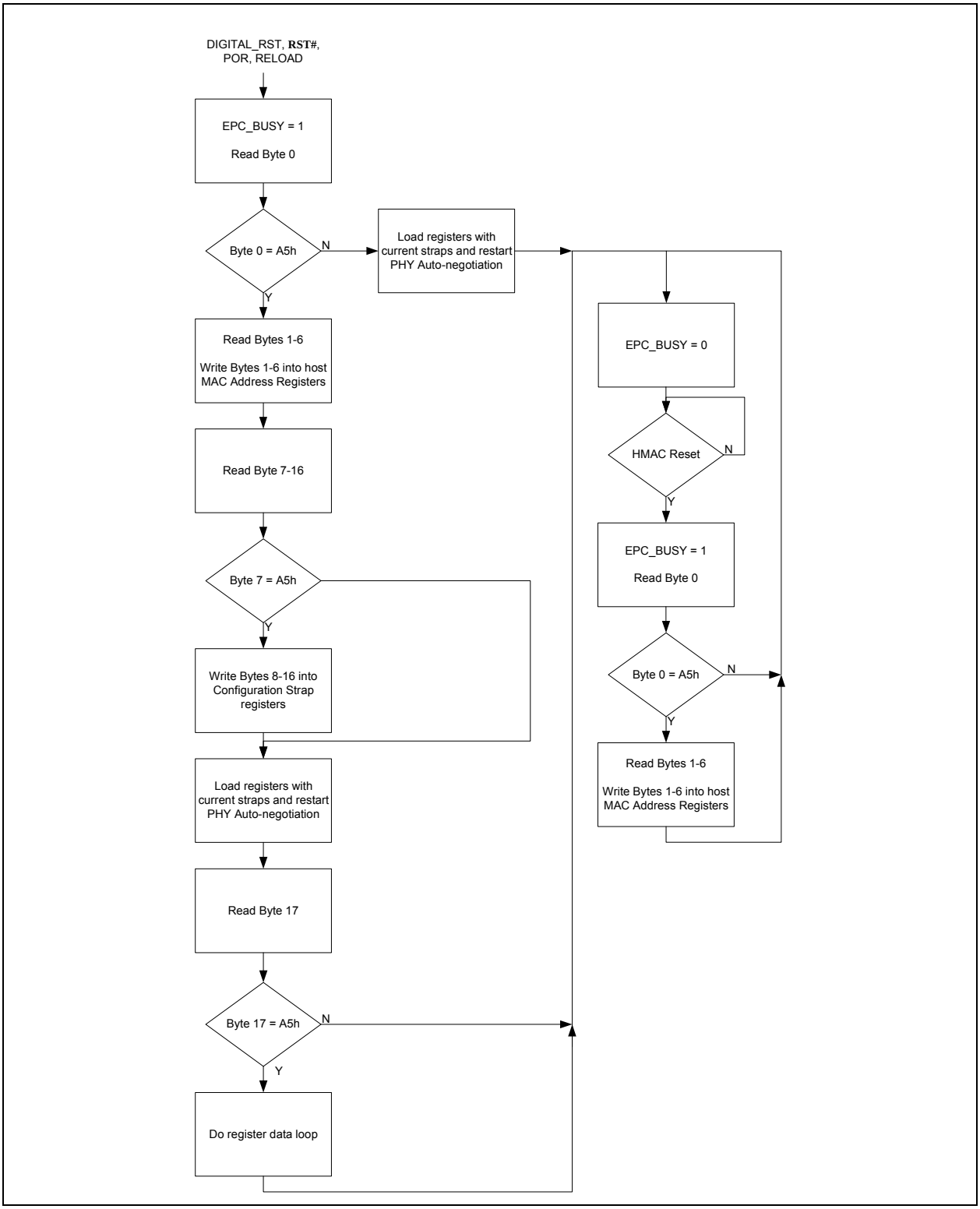
TABLE 13-3: EEPROM CONTENTS FORMAT OVERVIEW

EEPROM Address	Description	Value
0	EEPROM Valid Flag	A5h
1	MAC Address Low Word [7:0]	1st Byte on the Network
2	MAC Address Low Word [15:8]	2nd Byte on the Network
3	MAC Address Low Word [23:16]	3rd Byte on the Network
4	MAC Address Low Word [31:24]	4th Byte on the Network
5	MAC Address High Word [7:0]	5th Byte on the Network
6	MAC Address High Word [15:8]	6th Byte on the Network
7	Configuration Strap Values Valid Flag	A5h
8 - 16	Configuration Strap Values	See Table 13-4
17	Burst Sequence Valid Flag	A5h
18	Number of Bursts	See Section 13.4.5, "Register Data"
19 and above	Burst Data	See Section 13.4.5, "Register Data"

13.4.1 EEPROM LOADER OPERATION

Upon a pin reset ((**RST#**), power-on reset (POR), digital reset (Digital Reset (DIGITAL_RST) bit in the Reset Control Register (RESET_CTL)) or upon the issuance of a RELOAD command via the EEPROM Command Register (E2P_CMD), the EEPROM Controller Busy (EPC_BUSY) bit in the EEPROM Command Register (E2P_CMD) will be set. While the EEPROM Loader is active, the Device Ready (READY) bit of the Hardware Configuration Register (HW_CFG) is cleared and no writes to the device should be attempted. The operational flow of the EEPROM Loader can be seen in Figure 13-8.

FIGURE 13-8: EEPROM LOADER FLOW DIAGRAM



13.4.2 EEPROM VALID FLAG

Following the release of **RST#**, POR, DIGITAL_RST or a RELOAD command, the EEPROM Loader starts by reading the first byte of data from the EEPROM. If the value of A5h is not read from the first byte, the EEPROM Loader will load the current configuration strap values into the registers, restart PHY Auto-negotiation and then terminate, clearing the EEPROM Controller Busy (EPC_BUSY) bit in the EEPROM Command Register (E2P_CMD). Otherwise, the EEPROM Loader will continue reading sequential bytes from the EEPROM.

13.4.3 MAC ADDRESS

The next six bytes in the EEPROM, after the EEPROM Valid Flag, are written into the Host MAC Address High Register (HMAC_ADDRH) and Host MAC Address Low Register (HMAC_ADDRL) registers. The EEPROM bytes are written into the MAC address registers in the order specified in Table 13-3.

13.4.3.1 Host MAC Address Reload

While the EEPROM Loader is in the wait state, if a Host MAC reset is detected (via the Host MAC Reset (HMAC_RST) bit in the Reset Control Register (RESET_CTL)), the EEPROM Loader will read byte 0. If the byte 0 value is A5h, the EEPROM Loader will read bytes 1 through 6 from the EEPROM and reload the Host MAC Address High Register (HMAC_ADDRH) and Host MAC Address Low Register (HMAC_ADDRL). During this time, the EPC_BUSY bit in the EEPROM Command Register (E2P_CMD) is set and Device Ready (READY) bit of the Hardware Configuration Register (HW_CFG) is cleared.

13.4.4 SOFT-STRAPS

The 7th byte of data to be read from the EEPROM is the Configuration Strap Values Valid Flag. If this byte has a value of A5h, the next 9 bytes of data (8-16) are written into the configuration strap registers per the assignments detailed in Table 13-4.

If the flag byte is not A5h, these next 9 bytes are skipped (they are still read to maintain the data burst, but are discarded). However, the current configuration strap values are still loaded into the registers and the PHY Auto-negotiation is still restarted. Refer to Section 7.0, "Configuration Straps," on page 54 for more information on configuration straps.

Note: Bit locations in Table 13-4 that do not define a configuration strap must be written as 0.

TABLE 13-4: EEPROM CONFIGURATION BITS

Byte/Bit	7	6	5	4	3	2	1	0
Byte 8		FD_FC_strap_1	manual_FC_strap_1	manual_mdix_strap_1	auto_mdix_strap_1	speed_strap_1	duplex_strap_1	autoneg_strap_1
Byte 9								
Byte 10								
Byte 11			LED_fun_strap[2]	LED_fun_strap[1]	LED_fun_strap[0]		EEE_enable_strap_1	
Byte 12						LED_en_strap[2]	LED_en_strap[1]	LED_en_strap[0]
Byte 13								
Byte 14	HBI_ale_qualification_strap	HBI_rw_mode_strap	HBI_cs_polarity_strap	HBI_rd_rdwr_polarity_strap	HBI_wr_en_polarity_strap	HBI_ale_polarity_strap		
Byte 15								
Byte 16								

13.4.5 REGISTER DATA

Optionally following the configuration strap values, the EEPROM data may be formatted to allow access to the device's parallel, directly writable registers. Access to indirectly accessible registers is achievable with an appropriate sequence of writes (at the cost of EEPROM space).

This data is first preceded with a Burst Sequence Valid Flag (EEPROM byte 17). If this byte has a value of A5h, the data that follows is recognized as a sequence of bursts. Otherwise, the EEPROM Loader is finished, will go into a wait state and clear the EEPROM Controller Busy (EPC_BUSY) bit in the EEPROM Command Register (E2P_CMD). This can optionally generate an interrupt.

The data at EEPROM byte 18 and above should be formatted in a sequence of bursts. The first byte is the total number of bursts. Following this is a series of bursts, each consisting of a starting address, count and the count x 4 bytes of data. This results in the following formula for formatting register data:

8 bits number_of_bursts

repeat (number_of_bursts)

16 bits {starting_address[9:2] / count[7:0]}

repeat (count)

8 bits data[31:24], 8 bits data[23:16], 8 bits data[15:8], 8 bits data[7:0]

Note: The starting address is a DWORD address. Appending two 0 bits will form the register address.

As an example, the following is a 3 burst sequence, with 1, 2 and 3 DWORDs starting at register addresses 40h, 80h and C0h respectively:

A5h, (Burst Sequence Valid Flag)

3h, (number_of_bursts)

16{10h, 1h}, (starting_address1 divided by 4 / count1)

11h, 12h, 13h, 14h, (4 x count1 of data)

16{20h, 2h}, (starting_address2 divided by 4 / count2)

21h, 22h, 23h, 24h, 25h, 26h, 27h, 28h, (4 x count2 of data)

16{30h, 3h}, (starting_address3 divided by 4 / count3)

31h, 32h, 33h, 34h, 35h, 36h, 37h, 38h, 39h, 3Ah, 3Bh, 3Ch (4 x count3 of data)

In order to avoid overwriting the MAC CSR or MII Management interfaces, the EEPROM Loader waits until the following bits are cleared before performing any register write:

- Host MAC CSR Busy (HMAC_CSR_BUSY) bit of the Host MAC CSR Interface Command Register (MAC_CSR_CMD)
- MII Busy (MIIBZY) bit of the Host MAC MII Access Register (HMAC_MII_ACC)

The EEPROM Loader checks that the EEPROM address space is not exceeded. If so, it will stop and set the EEPROM Loader Address Overflow (LOADER_OVERFLOW) bit in the EEPROM Command Register (E2P_CMD). The address limit is based on the eeprom_size_strap which specifies a range of sizes. The address limit is set to the largest value of the specified range.

13.4.6 EEPROM LOADER FINISHED WAIT-STATE

Once finished with the last burst, the EEPROM Loader will go into a wait-state and the EEPROM Controller Busy (EPC_BUSY) bit of the EEPROM Command Register (E2P_CMD) will be cleared. This can optionally generate an interrupt.

13.5 I²C Master EEPROM Controller Registers

This section details the directly addressable I²C Master EEPROM Controller related System CSRs. These registers should only be used if an EEPROM has been connected to the device. For an overview of the entire directly addressable register map, refer to Section 5.0, "Register Map," on page 29.

TABLE 13-5: I²C MASTER EEPROM CONTROLLER REGISTERS

Address	Register Name (SYMBOL)
1B4h	EEPROM Command Register (E2P_CMD)
1B8h	EEPROM Data Register (E2P_DATA)

13.5.1 EEPROM COMMAND REGISTER (E2P_CMD)

Offset: 1B4h Size: 32 bits

This read/write register is used to control the read and write operations of the serial EEPROM.

Bits	Description	Type	Default
31	**EEPROM Controller Busy (EPC_BUSY)** When a 1 is written into this bit, the operation specified in the EPC_COMMAND field of this register is performed at the specified EEPROM address. This bit will remain set until the selected operation is complete. In the case of a read, this indicates that the Host can read valid data from the EEPROM Data Register (E2P_DATA). The E2P_CMD and E2P_DATA registers should not be modified until this bit is cleared. In the case where a write is attempted and an EEPROM is not present, the EPC_BUSY bit remains set until the EEPROM Controller Timeout (EPC_TIMEOUT) bit is set. At this time the EPC_BUSY bit is cleared. **Note:** EPC_BUSY is set immediately following power-up, or pin reset, or DIGITAL_RST reset. This bit is also set following the setting of the Host MAC Reset (HMAC_RST) bit in the Reset Control Register (RESET_CTL). After the EEPROM Loader has finished loading, the EPC_BUSY bit is cleared. Refer to chapter Section 13.4, "EEPROM Loader," on page 290 for more information.	R/W SC	0b

<table>
<tr><th>Bits</th><th>Description</th><th>Type</th><th>Default</th></tr>
<tr><td>30:28</td><td>EEPROM Controller Command (EPC_COMMAND)
This field is used to issue commands to the EEPROM controller. The EEPROM controller will execute a command when the EPC_BUSY bit is set. A new command must not be issued until the previous command completes. The field is encoded as follows:
<table><tr><th>[30]</th><th>[29]</th><th>[28]</th><th>Operation</th></tr><tr><td>0</td><td>0</td><td>0</td><td>READ</td></tr><tr><td>0</td><td>0</td><td>1</td><td>RESERVED</td></tr><tr><td>0</td><td>1</td><td>0</td><td>RESERVED</td></tr><tr><td>0</td><td>1</td><td>1</td><td>WRITE</td></tr><tr><td>1</td><td>0</td><td>0</td><td>RESERVED</td></tr><tr><td>1</td><td>0</td><td>1</td><td>RESERVED</td></tr><tr><td>1</td><td>1</td><td>0</td><td>RESERVED</td></tr><tr><td>1</td><td>1</td><td>1</td><td>RELOAD</td></tr></table>Note: Only the READ, WRITE and RELOAD commands are valid for I^2C mode. If an unsupported command is attempted, the EPC_BUSY bit will be cleared and EPC_TIMEOUT will be set.
The EEPROM operations are defined as follows:

READ (Read Location)
This command will cause a read of the EEPROM location pointed to by the EPC_ADDRESS bit field. The result of the read is available in the EEPROM Data Register (E2P_DATA).

WRITE (Write Location)
If erase/write operations are enabled in the EEPROM, this command will cause the contents of the EEPROM Data Register (E2P_DATA) to be written to the EEPROM location selected by the EPC_ADDRESS field.

RELOAD (EEPROM Loader Reload)
Instructs the EEPROM Loader to reload the device from the EEPROM. If a value of A5h is not found in the first address of the EEPROM, the EEPROM is assumed to be un-programmed and the RELOAD operation will fail. The CFG_LOADED bit indicates a successful load. Following this command, the device will enter the not ready state. The Device Ready (READY) bit in the Hardware Configuration Register (HW_CFG) should be polled to determine then the RELOAD is complete.</td><td>R/W</td><td>000b</td></tr>
<tr><td>27:19</td><td>RESERVED</td><td>RO</td><td>-</td></tr>
<tr><td>18</td><td>EEPROM Loader Address Overflow (LOADER_OVERFLOW)
This bit indicates that the EEPROM Loader tried to read past the end of the EEPROM address space. This indicates misconfigured EEPROM data.

This bit is cleared when the EEPROM Loader is restarted with a RELOAD command, Host MAC Reset (HMAC_RESET), or a Digital Reset (DIGITAL_RST).</td><td>RO</td><td>0b</td></tr>
</table>

Bits	Description	Type	Default
17	**EEPROM Controller Timeout (EPC_TIMEOUT)** This bit is set when a timeout occurs, indicating the last operation was unsuccessful. If an EEPROM WRITE operation is performed and no response is received from the EEPROM within 30 ms, the EEPROM controller will timeout and return to its idle state. This bit is also set if the EEPROM fails to respond with the appropriate ACKs, if the EEPROM slave device holds the clock low for more than 30 ms, if the I2C bus is not acquired within 1.92 seconds , or if an unsupported EPC_COMMAND is attempted. This bit is cleared when written high.	R/WC	0b
16	**Configuration Loaded (CFG_LOADED)** When set, this bit indicates that a valid EEPROM was found and the EEPROM Loader completed normally. This bit is set upon a successful load. It is cleared on power-up, pin and DIGITAL_RST resets, Host MAC Reset (HMAC_RESET), or at the start of a RELOAD. This bit is cleared when written high.	R/WC	0b
15:0	**EEPROM Controller Address (EPC_ADDRESS)** This field is used by the EEPROM Controller to address a specific memory location in the serial EEPROM. This address must be byte aligned.	R/W	0000h

13.5.2 EEPROM DATA REGISTER (E2P_DATA)

Offset: 1B8h Size: 32 bits

This read/write register is used in conjunction with the EEPROM Command Register (E2P_CMD) to perform read and write operations with the serial EEPROM.

Bits	Description	Type	Default
31:8	**RESERVED**	RO	-
7:0	**EEPROM Data (EEPROM_DATA)** This field contains the data read from or written to the EEPROM.	R/W	00h

14.0 IEEE 1588

14.1 Functional Overview

The device provides hardware support for the IEEE 1588-2008 Precision Time Protocol (PTP), allowing clock synchronization with remote Ethernet devices, packet time stamping, and time driven event generation.

Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.

The device may function as a master or a slave clock per the IEEE 1588-2008 specification. End-to-end and peer-to-peer link delay mechanisms are supported as are one-step and two-step operations.

A 32-bit seconds and 30-bit nanoseconds tunable clock is provided that is used as the time source for all PTP timestamp related functions. A 1588 Clock Events sub-module provides 1588 Clock comparison based interrupt generation and timestamp related GPIO event generation. GPIO pins can be used to trigger a timestamp capture when configured as an input, or output a signal based on a 1588 Clock Target compare event.

All features of the IEEE 1588 unit can be monitored and configured via their respective configuration and status registers. A detailed description of all 1588 CSRs is included in Section 14.8, "1588 Registers".

14.1.1 IEEE 1588-2008

IEEE 1588-2008 specifies a Precision Time Protocol (PTP) used by master and slave clock devices to pass time information in order to achieve clock synchronization. Ten network message types are defined:

- Sync
- Follow_Up
- Delay_Req
- Delay_Resp
- PDelay_Req
- PDelay_Resp
- PDelay_Resp_Follow_Up
- Announce
- Signaling
- Management

The first seven message types are used for clock synchronization. Using these messages, the protocol software may calculate the offset and network delay between timestamps, adjusting the slave clock frequency as needed. Refer to the IEEE 1588-2008 protocol for message definitions and proper usage.

A PTP domain is segmented into PTP sub-domains, which are then segmented into PTP communication paths. Within each PTP communication path there is a maximum of one master clock, which is the source of time for each slave clock. The determination of which clock is the master and which clock(s) is(are) the slave(s) is not fixed, but determined by the IEEE 1588-2008 protocol. Similarly, each PTP sub-domain may have only one master clock, referred to as the Grand Master Clock.

PTP communication paths are conceptually equivalent to Ethernet collision domains and may contain devices which extend the network. However, unlike Ethernet collision domains, the PTP communication path does not stop at a network switch, bridge, or router. This leads to a loss of precision when the network switch/bridge/router introduces a variable delay. Boundary clocks are defined which conceptually bypass the switch/bridge/router (either physically or via device integration). Essentially, a boundary clock acts as a slave to an upstream master, and as a master to a down stream slave. A boundary clock may contain multiple ports, but a maximum of one slave port is permitted.

Although boundary clocks solve the issue of the variable delay influencing the synchronization accuracy, they add clock jitter as each boundary clock tracks the clock of its upstream master. Another approach that is supported is the concept of transparent clocks. These devices measure the delay they have added when forwarding a message (the residence time) and report this additional delay either in the forwarded message (one-step) or in a subsequent message (two-step).

The PTP relies on the knowledge of the path delays between the master and the slave. With this information, and the knowledge of when the master has sent the packet, a slave can calculate its clock offset from the master and make appropriate adjustments. There are two methods of obtaining the network path delay. Using the end-to-end method, packets are exchanged between the slave and the master. Any intermediate variable bridge or switch delays are compensated by the transparent clock method described above. Using the round trip time and accounting for the residence time reported, the slave can calculate the mean delay from the master. Each slave sends and receives its own mes-

sages and calculates its own delay. While the end-to-end method is the simplest, it does add burden on the master since the master must process packets from each slave in the system. This is amplified when boundary clocks are replaced by transparent clocks. Also, the end-to-end delays must be recalculated if there is a change in the network topology. Using the peer-to-peer method, packets are exchanged only between adjacent master, slaves and transparent clocks. Each peer pair calculates the receive path delay. As time synchronization packets are forwarded between the master and the slave, the transparent clock adds the pre-measured receive path delay into the residence time. The final receiver adds its receive path delay. Using the peer-to-peer method, the full path delay is accounted for without the master having to service each slave. The peer-to-peer method better supports network topology changes since each path delay is kept up-to-date regardless of the port status.

The PTP implementation consists of the following major function blocks:

- PTP Timestamp
  This block provides time stamping and packet modification functions.
- 1588 Clock
  This block provides a tunable clock that is used as the time source for all PTP timestamp related functions.
- 1588 Clock Events
  This block provides clock comparison-based interrupt generation and timestamp related GPIO event generation.
- 1588 GPIOs
  This block provides for time stamping GPIO input events and for outputting clock comparison-based interrupt status.
- 1588 Interrupt
  This block provides interrupt generation, masking and status.
- 1588 Registers
  This block provides contains all configuration, control and status registers.

14.2 PTP Timestamp

This sub-module handles all PTP packet tasks related to recording timestamps of packets and inserting timestamps into packets.

Modes supported are:

- Ordinary Clock, master and slave, one-step and two-step, end-to-end or peer-to-peer delay
  - All 1588 packets are to and from the Host MAC
  - RX and TX timestamps saved in registers for S/W
  - RX timestamp stored in packet for ease of retrieval by S/W
  - Egress timestamp of Sync packet inserted on-the-fly for one-step
  - TX timestamp of Delay_Req packet stored in received Delay_Resp packet for ease of retrieval
  - Correction Field and ingress timestamp of Pdelay_Req packet saved in registers for one-step turnaround time
  - Correction Field of Pdelay_Resp packet automatically calculated and inserted on-the-fly for one-step
  - PTP checksums and Ethernet FCS updated on-the-fly
  - Ingress and egress timestamps corrected for latency
  - Asymmetry corrections
  - Peer delay correction on received Sync packets

Functions include:

- Detecting a PTP packet
  - 802.3/SNAP or Ethernet II encoding
  - Skipping over VLAN tags
  - Ethernet, IPv4 or IPv6 message formats
  - Skipping over IP extension headers
  - Checking the MAC and / or the IP addresses
- Recording the timestamp of received packets into registers
  - Accounting for the ingress latency
- Recording the timestamp of received packets into the packet and updating the layer 3 checksum and layer 2 FCS fields
  - Accounting for the ingress latency
- Forwarding or filtering PTP packets as needed to support ordinary clock mode
- Recording the timestamp of transmitted packets into registers
  - Accounting for the egress latency
- One-step on-the-fly timestamp insertion for Sync packets and updating the layer 3 checksum and layer 2 FCS
- One-step on-the-fly turnaround time insertion for Pdelay_Req packets and updating the layer 3 checksum and layer 2 FCS

Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.

14.2.1 RECEIVE FRAME PROCESSING

14.2.1.1 Ingress Time Snapshot

For each Ethernet frame, the receive frame processing detects the SFD field of the frame and temporarily saves the current 1588 Clock value.

INGRESS LATENCY

The ingress latency is the amount of time between the start of the frame's first symbol after the SFD on the network medium and the point when the 1588 clock value is internally captured. It is specified by the RX Latency (RX_LATENCY[15:0]) field in the 1588 Port Latency Register (1588_LATENCY) and is subtracted from the 1588 Clock value at the detection of the SFD. The setting is used to adjust the internally captured 1588 clock value such that the resultant timestamp more accurately corresponds to the start of the frame's first symbol after the SFD on the network medium.

The ingress latency consists of the receive latency of the PHY and the latency of the 1588 frame detection circuitry. The value depends on the port mode. Typical values are:

- 100BASE-TX: 285ns
- 100BASE-FX: 231ns plus the receive latency of the fiber transceiver
- 10BASE-T: 1674ns

14.2.1.2 1588 Receive Parsing

The 1588 Receive parsing block parses the incoming frame to identify 1588 PTP messages.

Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.

The Receive parsing block may be programmed to detect PTP messages encoded in UDP/IPv4, UDP/IPv6 and Layer 2 Ethernet formats via the RX IPv4 Enable (RX_IPV4_EN), RX IPv6 Enable (RX_IPV6_EN) and RX Layer 2 Enable (RX_LAYER2_EN) bits in the 1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG).

VLAN tagged and non-VLAN tagged frame formats are supported. Multiple VLAN tags are handled as long as they all use the standard type of 0x8100. Both Ethernet II (type field) and 802.3 (length field) w/ SNAP frame formats are supported.

The following tests are made to determine that the packet is a PTP message.

- MAC Destination Address checking is enabled via the RX MAC Address Enable (RX_MAC_ADDR_EN) in the 1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG).

  For the Layer 2 message format, the addresses of 01:1B:19:00:00:00 or 01:80:C2:00:00:0E may be enabled via the 1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG). Either address is allowed for Peer delay and non-Peer delay messages.

  For IPv4/UDP messages, any of the IANA assigned multicast IP destination addresses for IEEE 1588 (224.0.1.129 and 224.0.1.130 through .132), as well as the IP destination address for the Peer Delay Mechanism (224.0.0.107) may be enabled via the 1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG). These IP addresses map to the 802.3 MAC addresses of 01:00:5e:00:01:81 through 01:00:5e:00:01:84 and 01:00:5e:00:00:6B. Any of these addresses are allowed for Peer delay and non-Peer delay messages.

  For IPv6/UDP messages, any of the IANA assigned multicast IP destination addresses for IEEE 1588 (FF0X:0:0:0:0:0:0:181 and FF0X:0:0:0:0:0:0:182 through :184), as well as the IP destination address for the Peer Delay Mechanism (FF02:0:0:0:0:0:0:6B) may be enabled via the 1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG). These IP addresses map to the 802.3 MAC addresses of 33:33:00:00:01:81 through 33:33:00:00:01:84 and 33:33:00:00:00:6B. Any of these addresses are allowed for Peer delay and non-Peer delay messages.

  A user defined MAC address defined in the 1588 User MAC Address High-WORD Register (1588_USER_MAC_HI) and the 1588 User MAC Address Low-DWORD Register (1588_USER_MAC_LO) may also be individually enabled for the above formats.
- If the Type / Length Field indicates an EtherType then

  For the Layer 2 message format, the EtherType must equal 0x88F7.

  For IPv4/UDP messages, the EtherType must equal 0x0800.

  For IPv6/UDP messages, the EtherType must equal 0x86DD.
- If the Type / Length Field indicates a Length and the next 3 bytes equal 0xAAAA03 (indicating that a SNAP header is present) and the SNAP header has a OUI equal to 0x000000 then

  For the Layer 2 message format, the EtherType in the SNAP header must equal 0x88F7.

  For IPv4/UDP messages, the EtherType in the SNAP header must equal 0x0800.

  For IPv6/UDP messages, the EtherType in the SNAP header must equal 0x86DD.
- For IPv4/UDP messages, the Version field in the IPv4 header must equal 4, the IHL field must be 5 and the Protocol field must equal 17 (UDP) or 51 (AH). IPv4 options are not supported.
- For IPv6/UDP messages, the Version field in the IPv6 header must equal 6 and the Next Header field must equal 17 (UDP) or one of the IPv6 extension header values (0 - Hop-by-Hop Options, 60 - Destination Options, 43 -

Routing, 44 - Fragment, 51 - Authentication Header (AH)

- For IPv4/UDP messages, Destination IP Address checking is enabled via the RX IP Address Enable (RX_IP_ADDR_EN) in the 1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG). Any of the IANA assigned multicast IP destination addresses for IEEE 1588 (224.0.1.129 and 224.0.1.130 through .132), as well as the IP destination address for the Peer Delay Mechanism (224.0.0.107) may be enabled via the 1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG). Any of these addresses are allowed for Peer delay and non-Peer delay messages.
- For IPv6/UDP messages, Destination IP Address checking is enabled via the RX IP Address Enable (RX_IP_ADDR_EN) in the 1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG). Any of the IANA assigned multicast IP destination addresses for IEEE 1588 (FF0X:0:0:0:0:0:0:181 and FF0X:0:0:0:0:0:0:182 through :184), as well as the IP destination address for the Peer Delay Mechanism (FF02:0:0:0:0:0:0:6B) may be enabled via the 1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG). Any of these addresses are allowed for Peer delay and non-Peer delay messages.
- For IPv4/UDP if the Protocol field in the fixed header was 51 (AH), the Next Header field is checked for 17 (UDP) and the AH header is skipped.
- For IPv6/UDP if the Next Header field in the fixed header was one of the IPv6 extension header values, the Next Header field in the extension header is checked for 17 (UDP) or one of the IPv6 extension header values. If it is one of the IPv6 extension header values, the process repeats until either a value of 17 (UDP) or a value of 59 (No Next Header) are found or the packet ends.

14.2.1.3 Receive Message Ingress Time Recording

Following the determination of packet format and qualification of the packet as a PTP message above, the PTP header is checked for ALL of the following.

- The messageType field of the PTP header is checked and only those messages enabled via the RX PTP Message Type Enable (RX_PTP_MESSAGE_EN[15:0]) bits in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG) will be have their ingress times saved. Typically Sync, Delay_Req, PDelay_Req and PDelay_Resp messages are enabled.
- The versionPTP field of the PTP header is checked against the RX PTP Version (RX_PTP_VERSION[3:0]) field in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG). Only those messages with a matching version will be have their ingress times saved. A setting of 0 allows any PTP version.

  Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.
- If enabled via the RX PTP Domain Match Enable (RX_PTP_DOMAIN_EN) bit in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG), the domainNumber field of the PTP header is checked against the RX PTP Domain (RX_PTP_DOMAIN[7:0]) value in the same register. Only those messages with a matching domain will be have their ingress times saved.
- If enabled via the RX PTP Alternate Master Enable (RX_PTP_ALT_MASTER_EN) bit in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG), the alternateMasterFlag in the flagField of the PTP header is checked and only those messages with an alternateMasterFlag set to 0 will be have their ingress times saved.

At the end of the frame, the frame's FCS and the UDP checksum (for IPv4 and IPv6 formats) are verified. FCS checking can be disabled using the RX PTP FCS Check Disable (RX_PTP_FCS_DIS) bit in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG). UDP checksum checking can be disabled using the RX PTP UDP Checksum Check Disable (RX_PTP_UDP_CHKSUM_DIS) bit in the same register.

Note: A IPv4 UDP checksum value of 0x0000 indicates that the checksum is not included and is considered a pass. A IPv6 UDP checksum value of 0x0000 is invalid and is considered a fail.

Note: For IPv6, the UDP checksum calculation includes the IPv6 Pseudo header. Part of the IPv6 Pseudo header is the final IPv6 destination address.

If the IPv6 packet does not contain a Routing header, then the final IPv6 destination address is the destination address contained in the IPv6 header.

If the IPv6 packet does contain a Routing header, then the final IPv6 destination address is the address in the last element of the Routing header.

Note: The UDP checksum is calculated over the entire UDP payload as indicated by the UDP length field and not the assumed PTP packet length.

Note: The UDP checksum calculation does not included layer 2 pad bytes, if any.

If the FCS and checksum tests pass:

- The latency adjusted, 1588 Clock value, saved above at the start of the frame, is recorded into the 1588 Port RX Ingress Time Seconds Register (1588_RX_INGRESS_SEC) and 1588 Port RX Ingress Time NanoSeconds Register (1588_RX_INGRESS_NS).
- The messageType and sequenceId fields and 12-bit CRC of the portIdentity field of the PTP header are recorded into the Message Type (MSG_TYPE), Sequence ID (SEQ_ID) and Source Port Identity CRC (SRC_PRT_CRC) fields of the 1588 Port RX Message Header Register (1588_RX_MSG_HEADER).

  The 12-bit CRC of the portIdentity field is created by using the polynomial of $X^{12} + X^{11} + X^3 + X^2 + X + 1$.
- The corresponding maskable 1588 RX Timestamp Interrupt (1588_RX_TS_INT) is set in the 1588 Interrupt Status Register (1588_INT_STS).

Up to four receive events are saved per port with the count shown in the 1588 RX Timestamp Count (1588_RX_TS_CNT[2:0]) field in the 1588 Port Capture Information Register (1588_CAP_INFO). Additional events are not recorded. When the appropriate 1588 RX Timestamp Interrupt (1588_RX_TS_INT) bit is written as a one to clear, 1588 RX Timestamp Count (1588_RX_TS_CNT[2:0]) will decrement. If there are remaining events, the capture registers will update to the next event and the interrupt will set again.

PDELAY_REQ INGRESS TIME SAVING

One-step Pdelay_Resp messages sent by the Host, require their correctionField to be calculated on-the-fly to include the turnaround time between the ingress of the Pdelay_Req and the egress time of the Pdelay_Resp.

The 1588 Port RX Pdelay_Req Ingress Time Seconds Register (1588_RX_PDREQ_SEC) and the 1588 Port RX Pdelay_Req Ingress Time NanoSeconds Register (1588_RX_PDREQ_NS) hold the ingress time of the Pdelay_Req message.

The 1588 Port RX Pdelay_Req Ingress Correction Field High Register (1588_RX_PDREQ_CF_HI) and the 1588 Port RX Pdelay_Req Ingress Correction Field Low Register (1588_RX_PDREQ_CF_LOW) hold the correctionField of the Pdelay_Req message.

These registers can be set by S/W prior to sending the Pdelay_Resp message.

Alternatively, these registers can be updated by the H/W when the Pdelay_Req message is received. This function is enabled by the Auto Update (AUTO) bit in the 1588 Port RX Pdelay_Req Ingress Time NanoSeconds Register (1588_RX_PDREQ_NS) independent from the RX PTP Message Type Enable (RX_PTP_MESSAGE_EN[15:0]) bits.

As above (including all applicable notes):

- The versionPTP and domainNumber fields and alternateMasterFlag in the flagField of the PTP header are checked, if enabled.

  Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.
- At the end of the frame, the frame's FCS and the UDP checksum (for IPv4 and IPv6 formats) are verified, if enabled.

If all tests pass, then the Pdelay_Req message information is updated.

14.2.1.4 Ingress Packet Modifications

INGRESS TIME INSERTION INTO PACKETS

As an alternate to reading the receive time stamp from registers and matching it to the correct frame received in the Host MAC, the saved, latency adjusted, 1588 Clock value can be stored into the packet.

This function is enabled via the RX PTP Insert Timestamp Enable (RX_PTP_INSERT_TS_EN) and RX PTP Insert Timestamp Seconds Enable (RX_PTP_INSERT_TS_SEC_EN) bits in the 1588 Port RX Timestamp Insertion Configuration Register (1588_RX_TS_INSERT_CONFIG).

Note: Inserting the ingress time into the packet is an additional, separately enabled, feature verses the Ingress Time Recording described above. The capture registers are still updated as is the appropriate 1588 RX Timestamp Interrupt (1588_RX_TS_INT) bit and the 1588 RX Timestamp Count (1588_RX_TS_CNT[2:0]) field.

Following the determination of packet format and qualification of the packet as a PTP message above, the PTP header is checked for ALL of the following.

- The messageType field of the PTP header is checked and only those messages enabled via the RX PTP Mes-

sage Type Enable (RX_PTP_MESSAGE_EN[15:0]) bits in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG) will be have their ingress times inserted. Typically Sync, Delay_Req, PDelay_Req and PDelay_Resp messages are enabled.

- The versionPTP field of the PTP header is checked against the RX PTP Version (RX_PTP_VERSION[3:0]) field in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG). Only those messages with a matching version will be have their ingress times inserted. A setting of 0 allows any PTP version.

  Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.

  Note: The domainNumber field and alternateMasterFlag in the flagField of the PTP header are not tested for purpose of ingress time insertion.

The packet is modified as follows:

- The four bytes of nanoseconds are stored at an offset from the start of the PTP header

  The offset is specified in RX PTP Insert Timestamp Offset (RX_PTP_INSERT_TS_OFFSET[5:0]) field in the 1588 Port RX Timestamp Insertion Configuration Register (1588_RX_TS_INSERT_CONFIG).

  The lowest two bits of the seconds are stored into the upper 2 bits of the nanoseconds.
- If also enabled, bits 3:0 of the seconds are stored into bits 3:0 of a reserved byte in the PTP header. Bits 7:4 are set to zero.

  The offset of this reserved byte is specified by the RX PTP Insert Timestamp Seconds Offset (RX_PTP_INSERT_TS_SEC_OFFSET[5:0]) field in the 1588 Port RX Timestamp Insertion Configuration Register (1588_RX_TS_INSERT_CONFIG).

  Note: For version 2 of IEEE 1588, the four reserved bytes starting at offset 16 should be used for the nanoseconds. The reserved byte at offset 5 should be used for the seconds.

DELAY REQUEST EGRESS TIME INSERTION INTO DELAY REPONSE PACKET

Normally, in ordinary clock operation, the egress times of transmitted Delay_Req packets are saved and read by the Host S/W. To avoid the need to read these timestamps via register access, the egress time of the last transmitted Delay_Req packet on the port can be inserted into Delay_Resp packets received on the port.

This function is enabled via the RX PTP Insert Delay Request Egress in Delay Response Enable (RX_PTP_INSERT_DREQ_DRESP_EN) bit in the 1588 Port RX Timestamp Insertion Configuration Register (1588_RX_TS_INSERT_CONFIG).

As with any Ingress Time Insertion, Delay_Resp messages must be enable in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG) and the RX PTP Insert Timestamp Enable (RX_PTP_INSERT_TS_EN) must be set.

Note: Inserting the delay request egress time into the packet is an additional, separately enabled, feature verses the Egress Time Recording described above.

As with INGRESS TIME INSERTION INTO PACKETS, above:

- The versionPTP field of the PTP header is checked and the domainNumber field and alternateMasterFlag in the flagField of the PTP header are not checked.

  Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.
- The four bytes of nanoseconds / 2 bits of seconds are stored at the specified offset of the PTP header.
- Bits 3:0 of the seconds are stored at the specified offset in the PTP header, if enabled.

Effectively, this function is the same as the INGRESS TIME INSERTION INTO PACKETS except that the egress time of the Delay_Req is inserted instead of the ingress time of the Delay_Resp.

FRAME UPDATING

Frames are modified even if their original FCS or UDP checksum is invalid.

- For IPv4, the UDP checksum is set to 0.

  If the original UDP checksum was invalid, a receive symbol error is forced and the 1588 Port RX Checksum Dropped Count Register (1588_RX_CHKSUM_DROPPED_CNT) incremented.
  This can be disabled by the RX PTP Bad UDP Checksum Force Error Disable (RX_PTP_BAD_UDP_CHKSUM_-

FORCE_ERR_DIS) field in the 1588 Port RX Timestamp Insertion Configuration Register (1588_RX-_TS_INSERT_CONFIG).

Note: An original UDP checksum value of 0x0000 indicates that the checksum is not included and is considered a pass.

Note: The original UDP checksum is calculated over the entire UDP payload as indicated by the UDP length field and not the assumed PTP packet length.

Note: The original UDP checksum calculation does not included layer 2 pad bytes, if any.

- For IPv6, the two bytes beyond the end of the PTP message are modified so that the original UDP checksum is correct for the modified payload. These bytes are updated by accumulating the differences between the original frame data and the substituted data using the mechanism defined in IETF RFC 1624.

  If the original UDP checksum was invalid, a receive symbol error is forced and the 1588 Port RX Checksum Dropped Count Register (1588_RX_CHKSUM_DROPPED_CNT) incremented.
  This can be disabled by the RX PTP Bad UDP Checksum Force Error Disable (RX_PTP_BAD_UDP_CHKSUM_-FORCE_ERR_DIS) field in the 1588 Port RX Timestamp Insertion Configuration Register (1588_RX-_TS_INSERT_CONFIG).

  Note: Since the two bytes beyond the end of the PTP message are modified based on the differences between the original frame data and the substituted data, an invalid incoming checksum would always result in an outgoing checksum error.

  Note: An original UDP checksum value of 0x0000 is invalid and is considered a fail.

  Note: For IPv6, the UDP checksum calculation includes the IPv6 Pseudo header. Part of the IPv6 Pseudo header is the final IPv6 destination address.

  If the IPv6 packet does not contain a Routing header, then the final IPv6 destination address is the destination address contained in the IPv6 header.

  If the IPv6 packet does contain a Routing header, then the final IPv6 destination address is the address in the last element of the Routing header.

  Note: The original UDP checksum is calculated over the entire UDP payload as indicated by the UDP length field and not the assumed PTP packet length.

  Note: The original UDP checksum calculation does not included layer 2 pad bytes, if any.

  Note: The two bytes beyond the end of the PTP message are located by using the messageLength field from the PTP header.

- The frame FCS is recomputed.
  If the original FCS was invalid, a bad FCS is forced.
- If the frame has a receive symbol error(s), a receive symbol error indication will be propagated at the same nibble location(s).

  Note: FCS and UDP checksums are only updated if the frame was actually modified. If no modifications are done, the existing FCS and checksums are left unchanged.

14.2.1.5 Ingress Message Filtering

PTP messages can be filtered upon receive. Following the determination of packet format and qualification of the packet as a PTP message above, the PTP header is checked for ANY of the following.

- The messageType field of the PTP header is checked and those messages that have their RX PTP Message Type Filter Enable (RX_PTP_MSG_FLTR_EN[15:0]) bits in the 1588 Port RX Filter Configuration Register (1588_RX_-FILTER_CONFIG) set will be filtered. Typically Delay_Req and Delay_Resp messages are filtered in peer-to-peer transparent clocks.
- The versionPTP field of the PTP header is checked against the RX PTP Version (RX_PTP_VERSION[3:0]) field in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG). If the RX PTP Version Filter Enable (RX_PTP_VERSION_FLTR_EN) bit in the 1588 Port RX Filter Configuration Register (1588_RX_-FILTER_CONFIG) is set, messages with a non-matching version will be filtered. A version setting of 0 allows any PTP version and would not cause filtering.

  Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.

- If enabled via the RX PTP Domain Filter Enable (RX_PTP_DOMAIN_FLTR_EN) bit in the 1588 Port RX Filter Configuration Register (1588_RX_FILTER_CONFIG), messages whose domainNumber field in the PTP header does not match the RX PTP Domain (RX_PTP_DOMAIN[7:0]) value in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG) will be filtered.
- If enabled via the RX PTP Alternate Master Filter Enable (RX_PTP_ALT_MASTER_FLTR_EN) bit in the 1588 Port RX Filter Configuration Register (1588_RX_FILTER_CONFIG), messages whose alternateMasterFlag in the flagField of the PTP header is set will be filtered.

At the end of the frame, the frame's FCS and the UDP checksum (for IPv4 and IPv6 formats) are verified. FCS checking can be disabled using the RX PTP FCS Check Disable (RX_PTP_FCS_DIS) bit in the 1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG). UDP checksum checking can be disabled using the RX PTP UDP Checksum Check Disable (RX_PTP_UDP_CHKSUM_DIS) bit in the same register.

Note: A IPv4 UDP checksum value of 0x0000 indicates that the checksum is not included and is considered a pass. A IPv6 UDP checksum value of 0x0000 is invalid and is considered a fail.

Note: For IPv6, the UDP checksum calculation includes the IPv6 Pseudo header. Part of the IPv6 Pseudo header is the final IPv6 destination address.

If the IPv6 packet does not contain a Routing header, then the final IPv6 destination address is the destination address contained in the IPv6 header.

If the IPv6 packet does contain a Routing header, then the final IPv6 destination address is the address in the last element of the Routing header.

Note: The UDP checksum is calculated over the entire UDP payload as indicated by the UDP length field and not the assumed PTP packet length.

Note: The UDP checksum calculation does not included layer 2 pad bytes, if any.

If the FCS and checksum tests pass, the frame is filtered by inserting a receive symbol error and the 1588 Port RX Filtered Count Register (1588_RX_FILTERED_CNT) is incremented.

Note: The MAC will count this as an errored packet.

Note: Message filtering is an additional, separately enabled, feature verses any packet ingress time recording and packet modification. Although these functions typically would not be used together on the same message type.

14.2.2 TRANSMIT FRAME PROCESSING

14.2.2.1 Egress Time Snapshot

For each Ethernet frame, the transmit frame processing detects the SFD field of the frame and temporarily saves the current 1588 Clock value.

EGRESS LATENCY

The egress latency is the amount of time between the point when the 1588 clock value is internally captured and the start of the frame's first symbol after the SFD on the network medium. It is specified by the TX Latency (TX_LATENCY[15:0]) field in the 1588 Port Latency Register (1588_LATENCY) and is added to the 1588 Clock value at the detection of the SFD. The setting is used to adjust the internally captured 1588 clock value such that the resultant timestamp more accurately corresponds to the start of the frame's first symbol after the SFD on the network medium.

The egress latency consists of the transmit latency of the PHY and the latency of the 1588 frame detection circuitry. The value depends on the port mode. Typical values are:

- 100BASE-TX: 95ns
- 100BASE-FX: 68ns plus the transmit latency of the fiber transceiver
- 10BASE-T: 1139ns

14.2.2.2 1588 Transmit Parsing

The 1588 Transmit parsing block parses the outgoing frame to identify 1588 PTP messages.

Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.

The Transmit parsing block may be programmed to detect PTP messages encoded in UDP/IPv4, UDP/IPv6 and Layer 2 Ethernet formats via the TX IPv4 Enable (TX_IPV4_EN), TX IPv6 Enable (TX_IPV6_EN) and TX Layer 2 Enable (TX_LAYER2_EN) bits in the 1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG).

VLAN tagged and non-VLAN tagged frame formats are supported. Multiple VLAN tags are handled as long as they all use the standard type of 0x8100. Both Ethernet II (type field) and 802.3 (length field) w/ SNAP frame formats are supported.

The following tests are made to determine that the packet is a PTP message.

- MAC Destination Address checking is enabled via the TX MAC Address Enable (TX_MAC_ADDR_EN) in the 1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG).

  For the Layer 2 message format, the addresses of 01:1B:19:00:00:00 or 01:80:C2:00:00:0E may be enabled via the 1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG). Either address is allowed for Peer delay and non-Peer delay messages.

  For IPv4/UDP messages, any of the IANA assigned multicast IP destination addresses for IEEE 1588 (224.0.1.129 and 224.0.1.130 through .132), as well as the IP destination address for the Peer Delay Mechanism (224.0.0.107) may be enabled via the 1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG). These IP addresses map to the 802.3 MAC addresses of 01:00:5e:00:01:81 through 01:00:5e:00:01:84 and 01:00:5e:00:00:6B. Any of these addresses are allowed for Peer delay and non-Peer delay messages.

  For IPv6/UDP messages, any of the IANA assigned multicast IP destination addresses for IEEE 1588 (FF0X:0:0:0:0:0:0:181 and FF0X:0:0:0:0:0:0:182 through :184), as well as the IP destination address for the Peer Delay Mechanism (FF02:0:0:0:0:0:0:6B) may be enabled via the 1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG). These IP addresses map to the 802.3 MAC addresses of 33:33:00:00:01:81 through 33:33:00:00:01:84 and 33:33:00:00:00:6B. Any of these addresses are allowed for Peer delay and non-Peer delay messages.

  A user defined MAC address defined in the 1588 User MAC Address High-WORD Register (1588_USER_MAC_HI) and the 1588 User MAC Address Low-DWORD Register (1588_USER_MAC_LO) may also be individually enabled for the above formats.
- If the Type / Length Field indicates an EtherType then

  For the Layer 2 message format, the EtherType must equal 0x88F7.

  For IPv4/UDP messages, the EtherType must equal 0x0800.

  For IPv6/UDP messages, the EtherType must equal 0x86DD.
- If the Type / Length Field indicates a Length and the next 3 bytes equal 0xAAAA03 (indicating that a SNAP header is present) and the SNAP header has a OUI equal to 0x000000 then

  For the Layer 2 message format, the EtherType in the SNAP header must equal 0x88F7.

  For IPv4/UDP messages, the EtherType in the SNAP header must equal 0x0800.

  For IPv6/UDP messages, the EtherType in the SNAP header must equal 0x86DD.
- For IPv4/UDP messages, the Version field in the IPv4 header must equal 4, the IHL field must be 5 and the Protocol field must equal 17 (UDP) or 51 (AH). IPv4 options are not supported.
- For IPv6/UDP messages, the Version field in the IPv6 header must equal 6 and the Next Header field must equal 17 (UDP) or one of the IPv6 extension header values (0 - Hop-by-Hop Options, 60 - Destination Options, 43 - Routing, 44 - Fragment, 51 - Authentication Header (AH)
- For IPv4/UDP messages, Destination IP Address checking is enabled via the TX IP Address Enable (TX_IP_ADDR_EN) in the 1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG). Any of the IANA assigned multicast IP destination addresses for IEEE 1588 (224.0.1.129 and 224.0.1.130 through .132), as well as the IP destination address for the Peer Delay Mechanism (224.0.0.107) may be enabled via the 1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG). Any of these addresses are allowed for Peer delay and non-Peer delay messages.
- For IPv6/UDP messages, Destination IP Address checking is enabled via the TX IP Address Enable (TX_IP_ADDR_EN) in the 1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG). Any of the IANA assigned multicast IP destination addresses for IEEE 1588 (FF0X:0:0:0:0:0:0:181 and FF0X:0:0:0:0:0:0:182 through :184), as well as the IP destination address for the Peer Delay Mechanism (FF02:0:0:0:0:0:0:6B) may be enabled via the 1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG). Any of these

addresses are allowed for Peer delay and non-Peer delay messages.

- For IPv4/UDP if the Protocol field in the fixed header was 51 (AH), the Next Header field is checked for 17 (UDP) and the AH header is skipped.
- For IPv6/UDP if the Next Header field in the fixed header was one of the IPv6 extension header values, the Next Header field in the extension header is checked for 17 (UDP) or one of the IPv6 extension header values. If it is one of the IPv6 extension header values, the process repeats until either a value of 17 (UDP) or a value of 59 (No Next Header) are found or the packet ends.

14.2.2.3 Transmit Message Egress Time Recording

Following the determination of packet format and qualification of the packet as a PTP message above, the PTP header is checked for ALL of the following.

- The messageType field of the PTP header is checked and only those messages enabled via the TX PTP Message Type Enable (TX_PTP_MESSAGE_EN[15:0]) bits in the 1588 Port TX Timestamp Configuration Register (1588_TX_TIMESTAMP_CONFIG) will be have their egress times saved. Typically Sync, Delay_Req, PDelay_Req and PDelay_Resp messages are enabled.
- The versionPTP field of the PTP header is checked against the TX PTP Version (TX_PTP_VERSION[3:0]) field in the 1588 Port TX Timestamp Configuration Register (1588_TX_TIMESTAMP_CONFIG). Only those messages with a matching version will be have their egress times saved. A setting of 0 allows any PTP version.

  Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.
- If enabled via the TX PTP Domain Match Enable (TX_PTP_DOMAIN_EN) bit in the 1588 Port TX Timestamp Configuration Register (1588_TX_TIMESTAMP_CONFIG), the domainNumber field of the PTP header is checked against the TX PTP Domain (TX_PTP_DOMAIN[7:0]) value in the same register. Only those messages with a matching domain will be have their egress times saved.
- If enabled via the TX PTP Alternate Master Enable (TX_PTP_ALT_MASTER_EN) bit in the 1588 Port TX Timestamp Configuration Register (1588_TX_TIMESTAMP_CONFIG), the alternateMasterFlag in the flagField of the PTP header is checked and only those messages with an alternateMasterFlag set to 0 will be have their egress times saved.

At the end of the frame, the frame's FCS and the UDP checksum (for IPv4 and IPv6 formats) are verified. FCS checking can be disabled using the TX PTP FCS Check Disable (TX_PTP_FCS_DIS) bit in the 1588 Port TX Timestamp Configuration Register (1588_TX_TIMESTAMP_CONFIG). UDP checksum checking can be disabled using the TX PTP UDP Checksum Check Disable (TX_PTP_UDP_CHKSUM_DIS) bit in the same register.

Note: A IPv4 UDP checksum value of 0x0000 indicates that the checksum is not included and is considered a pass. A IPv6 UDP checksum value of 0x0000 is invalid and is considered a fail.

Note: For IPv6, the UDP checksum calculation includes the IPv6 Pseudo header. Part of the IPv6 Pseudo header is the final IPv6 destination address.

If the IPv6 packet does not contain a Routing header, then the final IPv6 destination address is the destination address contained in the IPv6 header.

If the IPv6 packet does contain a Routing header, then the final IPv6 destination address is the address in the last element of the Routing header.

Note: The UDP checksum is calculated over the entire UDP payload as indicated by the UDP length field and not the assumed PTP packet length.

Note: The UDP checksum calculation does not included layer 2 pad bytes, if any.

If the FCS and checksum tests pass:

- The latency adjusted, 1588 Clock value, saved above at the start of the frame, is recorded into the 1588 Port TX Egress Time Seconds Register (1588_TX_EGRESS_SEC) and 1588 Port TX Egress Time NanoSeconds Register (1588_TX_EGRESS_NS).
- The messageType and sequenceId fields and 12-bit CRC of the portIdentity field of the PTP header are recorded into the Message Type (MSG_TYPE), Sequence ID (SEQ_ID) and Source Port Identity CRC (SRC_PRT_CRC) fields of the 1588 Port TX Message Header Register (1588_TX_MSG_HEADER).

  The 12-bit CRC of the portIdentity field is created by using the polynomial of $X^{12} + X^{11} + X^3 + X^2 + X + 1$.
- The corresponding maskable 1588 TX Timestamp Interrupt (1588_TX_TS_INT) is set in the 1588 Interrupt Status Register (1588_INT_STS).

Up to four transmit events are saved per port with the count shown in the 1588 TX Timestamp Count (1588_TX_TS_CNT[2:0]) field in the 1588 Port Capture Information Register (1588_CAP_INFO). Additional events are not recorded. When the appropriate 1588 TX Timestamp Interrupt (1588_TX_TS_INT) bit is written as a one to clear, 1588 TX Timestamp Count (1588_TX_TS_CNT[2:0]) will decrement. If there are remaining events, the capture registers will update to the next event and the interrupt will set again.

DELAY_REQ EGRESS TIME SAVING

Normally, in ordinary clock operation, the egress time of transmitted Delay_Req packets are saved and read by the Host S/W. To avoid the need to read these timestamps via register access, the egress time of the last transmitted Delay_Req packet on the port can be inserted into Delay_Resp packets received on the port.

The 1588 Port TX Delay_Req Egress Time Seconds Register (1588_TX_DREQ_SEC) and the 1588 Port TX Delay_Req Egress Time NanoSeconds Register (1588_TX_DREQ_NS) hold the egress time of the Delay_Req message.

These registers are updated by the H/W when the Delay_Req message is transmitted independent of the settings in the TX PTP Message Type Enable (TX_PTP_MESSAGE_EN[15:0]) bits.

As above (including all applicable notes):

- The versionPTP and domainNumber fields and alternateMasterFlag in the flagField of the PTP header are checked, if enabled.

  Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.

- At the end of the frame, the frame's FCS and the UDP checksum (for IPv4 and IPv6 formats) are verified, if enabled.

If all tests pass, then the Delay_Req message information is updated and available for the receive function.

14.2.2.4 Egress Packet Modifications

EGRESS TIME INSERTION - SYNC MESSAGE FUNCTION

While functioning as an ordinary clock master, one-step transmission of Sync messages from the Host S/W requires the actual egress time to be inserted into the ten byte, originTimestamp field. The 32-bit nanoseconds portion and the lower 32 bits of the seconds portion come from the latency adjusted, 1588 Clock value, saved above at the start of the frame. The upper 16 bits of seconds are taken from the 1588 TX One-Step Sync Upper Seconds Register (1588_TX_ONE_STEP_SYNC_SEC). The Host software is responsible for maintaining this register if required.

Note: Inserting the egress time into the packet is an additional, separately enabled, feature verses the Egress Time Recording described above.

This function is enabled via the TX PTP Sync Message Egress Time Insertion (TX_PTP_SYNC_TS_INSERT) bit in the 1588 Port TX Modification Register (1588_TX_MOD) and is used only on frames which have bit 7 of the PTP header's reserved byte cleared.

Note: The offset of the reserved byte is specified by the TX PTP 1 Reserved Byte Offset (TX_PTP_1_RSVD_OFFSET[5:0]) field in the 1588 Port TX Modification Register (1588_TX_MOD).

Proper operation of the transmitter requires that the reserved byte resides after the versionPTP field and before the correctionField.

For version 2 of IEEE 1588, the reserved byte at offset 5 should be used.

Following the determination of packet format and qualification of the packet as a PTP message above, the PTP header is checked.

- The versionPTP field of the PTP header is checked against the TX PTP Version (TX_PTP_VERSION[3:0]) field in the 1588 Port TX Timestamp Configuration Register (1588_TX_TIMESTAMP_CONFIG). Only those messages with a matching version will have their egress time inserted. A setting of 0 allows any PTP version.

  Note: The domainNumber field and alternateMasterFlag in the flagField of the PTP header are not tested.

  Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.

EGRESS CORRECTION FIELD TURNAROUND TIME ADJUSTMENT - PDELAY_RESP MESSAGE FUNCTION

One-step Pdelay_Resp messages sent by the Host, require their correctionField to be calculated on-the-fly to include the turnaround time between the ingress of the Pdelay_Req and the egress time of the Pdelay_Resp.

Pdelay_Resp.CF = Pdelay_Req.CF + Pdelay_Resp.egress time - Pdelay_Req.ingress time.

Note: Adjusting the Correction Field in the packet is an additional, separately enabled, feature verses the Egress Time Recording described above.

Note: If the original correctionField contains a value of 7FFFFFFFFFFFFFFF, it is not modified.

If adjustment to the correctionField would result in a value that is larger than 7FFFFFFFFFFFFFFF, that value is used instead.

The 1588 Port RX Pdelay_Req Ingress Time Seconds Register (1588_RX_PDREQ_SEC) and the 1588 Port RX Pdelay_Req Ingress Time NanoSeconds Register (1588_RX_PDREQ_NS) hold the ingress time of the Pdelay_Req message.

The 1588 Port RX Pdelay_Req Ingress Correction Field High Register (1588_RX_PDREQ_CF_HI) and the 1588 Port RX Pdelay_Req Ingress Correction Field Low Register (1588_RX_PDREQ_CF_LOW) hold the correctionField of the Pdelay_Req message.

These registers are set by S/W prior to sending the Pdelay_Resp message or by the automatic updating described above in PDELAY_REQ INGRESS TIME SAVING.

The egress time is the latency adjusted, 1588 Clock value, saved above at the start of the Pdelay_Resp frame.

Note: Since only four bits worth of seconds of the Pdelay_Req ingress time are stored, the Host must send the Pdelay_Resp within 16 seconds.

This function is enabled via the TX PTP Pdelay_Resp Message Turnaround Time Insertion (TX_PTP_PDRESP_TA_INSERT) bit in the 1588 Port TX Modification Register (1588_TX_MOD) and is used only on frames which have bit 7 of the PTP header's reserved byte cleared.

As with Egress Time Insertion above:

- The versionPTP field of the PTP header is checked and the domainNumber field and alternateMasterFlag in the flagField of the PTP header are not checked

  Note: Support for the IEEE 1588-2002 (v1) packet format is not provided.

FRAME UPDATING

Frames are modified even if their original FCS or UDP checksum is invalid.

- For IPv4, the UDP checksum is set to 0 under the following conditions.

  If the TX PTP Clear UDP/IPv4 Checksum Enable (TX_PTP_CLR_UDPV4_CHKSUM) bit in the 1588 Port TX Modification Register 2 (1588_TX_MOD2) is set and either Sync Egress Time Insertion or Pdelay_Resp Correction Field Turnaround Time Adjustment is enabled, the UDP checksum is set to 0 for *all* PTP messages. The actual message type and the ptp_version field are not checked.
- For IPv6, the two bytes beyond the end of the PTP message are modified to correct for the UDP checksum. These bytes are updated by accumulating the differences between the original frame data and the substituted data using the mechanism defined in IETF RFC 1624.

  It is assumed that the existing two bytes are zero and are replaced.

It is assumed that the original UDP checksum is valid and is not checked.

Note: Since the two bytes beyond the end of the PTP message are modified based on the differences between the original frame data and the substituted data, an invalid incoming checksum would result in an outgoing checksum error.

Note: The two bytes beyond the end of the PTP message are located by using the messageLength field from the PTP header.

- The frame FCS is recomputed
  It is assumed that the original FCS is valid and is not checked.
- If the frame has a transmit symbol error(s), a transmit symbol error indication will be propagated at the same nibble location(s)

Note: The FCS and IPv6/UDP checksum are updated only if the frame was actually modified.

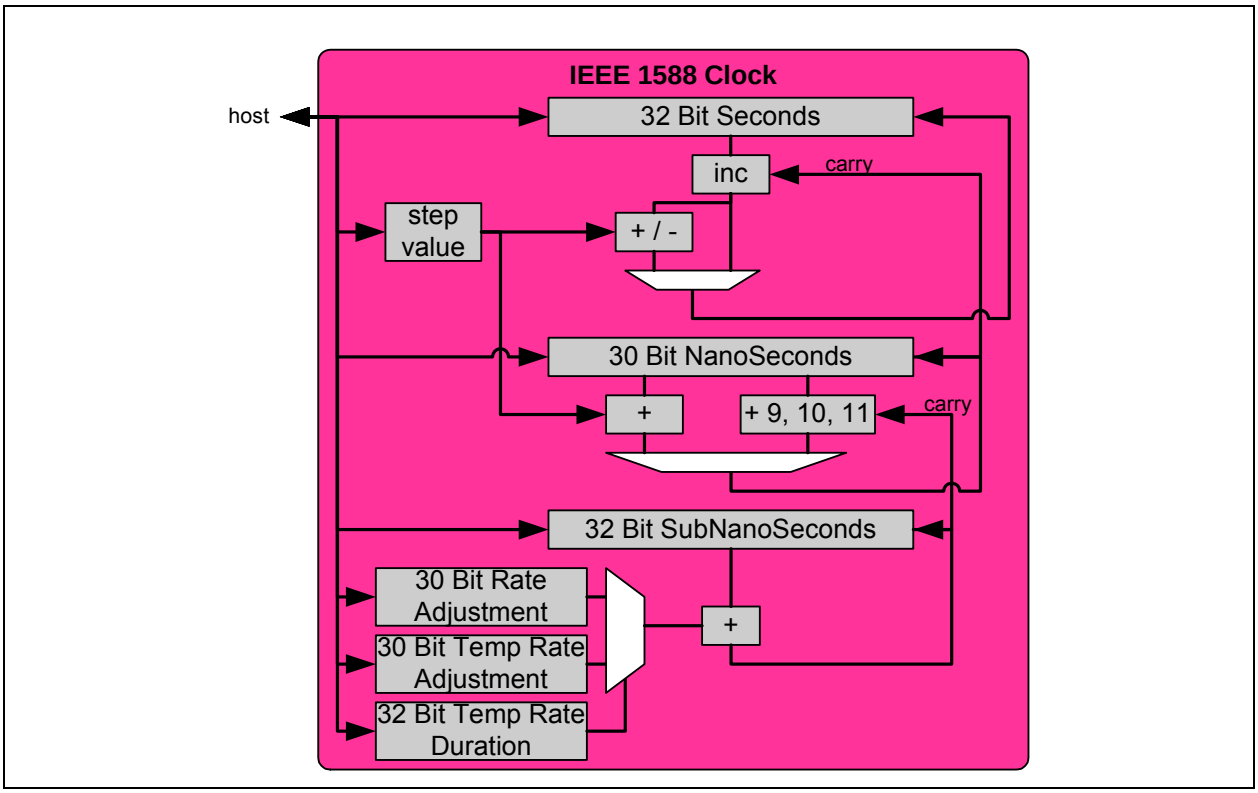
The IPv4/UDP checksum is cleared as indicated above and could be the only modification in the message. If the IPv4/UDP checksum is cleared, the FCS is recomputed.

If no modifications are done, the existing FCS and checksums are left unchanged.

14.3 1588 Clock

The tunable 1588 Clock is the time source for all PTP related functions of the device. The block diagram is shown in Figure 14-1.

FIGURE 14-1: 1588 CLOCK BLOCK DIAGRAM



The 1588 Clock consists of a 32-bit wide seconds portion and a 30-bit wide nanoseconds portion. Running at a nominal reference frequency of 100MHz, the nanoseconds portion is normally incremented by a value of 10 every reference clock period. Upon reaching or exceeding its maximum value of 10^9, the nanoseconds portion rolls over to or past zero and the seconds portion is incremented.

The 1588 Clock can be read by setting the Clock Read (1588_CLOCK_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL). This saves the current value of the 1588 Clock into the 1588 Clock Seconds Register (1588_CLOCK_SEC), 1588 Clock NanoSeconds Register (1588_CLOCK_NS) and 1588 Clock Sub-NanoSeconds Register (1588_CLOCK_SUBNS) where it can be read.

Although the IEEE 1588-2008 specification calls for a 48-bit seconds counter, the hardware only supports 32 bits. For purposes of event timestamping, residence time correction or other comparisons, the 136 year rollover time of 32 bits is sufficient. Rollover can be detected and corrected by comparing the two values of interest. To support one-step operations, the device can insert the Egress Timestamp into the origin Timestamp field of Sync messages. However, the Host must maintain the 1588 TX One-Step Sync Upper Seconds Register (1588_TX_ONE_STEP_SYNC_SEC). The Host should avoid sending a Sync message if there is a possibility that the 32-bit seconds counter will reach its rollover value before the message is transmitted.

A 32-bit sub-nanoseconds counter is used to precisely tune the rate of the 1588 Clock by accounting for the difference between the nominal 10ns and the actual rate of the master clock. Every reference clock period the sub-nanoseconds counter is incremented by the Clock Rate Adjustment Value (1588_CLOCK_RATE_ADJ_VALUE) in the 1588 Clock Rate Adjustment Register (1588_CLOCK_RATE_ADJ), specified in 2^{-32} nanoseconds. When the sub-nanoseconds counter rolls over past zero, the nanoseconds portion of the 1588 Clock is incremented by 9 or 11 instead of the normal value of 10. The choice to speed up or slow down is determined by the Clock Rate Adjustment Direction (1588_CLOCK_RATE_ADJ_DIR) bit. The ability to adjust for 1 ns approximately every 43 seconds allows for a tuning precision of approximately 2.3^{-9} percent. The maximum adjustment is 1 ns every 4 clocks (40 ns) or 2.5 percent.

In addition to adjusting the frequency of the 1588 Clock, the Host may directly set the 1588 Clock, make a one-time step adjustment of the 1588 Clock or specify a temporary rate. The choice of method depends on needed adjustment. For initial adjustments, direct or one-time step adjustments may be best. For on-going minor adjustments, the temporary rate adjustment may be best. Ideally, the frequency will be matched and once the 1588 Clock is synchronized, no further adjustments would be needed.

In order to perform a direct writing of the 1588 Clock, the desired value is written into the 1588 Clock Seconds Register (1588_CLOCK_SEC), 1588 Clock NanoSeconds Register (1588_CLOCK_NS) and 1588 Clock Sub-NanoSeconds Register (1588_CLOCK_SUBNS). The Clock Load (1588_CLOCK_LOAD) bit in the 1588 Command and Control Register (1588_CMD_CTL) is then set.

In order to perform a one-time positive or negative adjustment to the seconds portion of the 1588 Clock, the desired change and direction are written into the 1588 Clock Step Adjustment Register (1588_CLOCK_STEP_ADJ). The Clock Step Seconds (1588_CLOCK_STEP_SECONDS) bit in the 1588 Command and Control Register (1588_CMD_CTL) is then set. The internal sub-nanoseconds counter and the nanoseconds portion of the 1588 Clock are not affected. If a nanoseconds portion rollover coincides with the 1588 Clock adjustment, the 1588 Clock adjustment is applied in addition to the seconds increment.

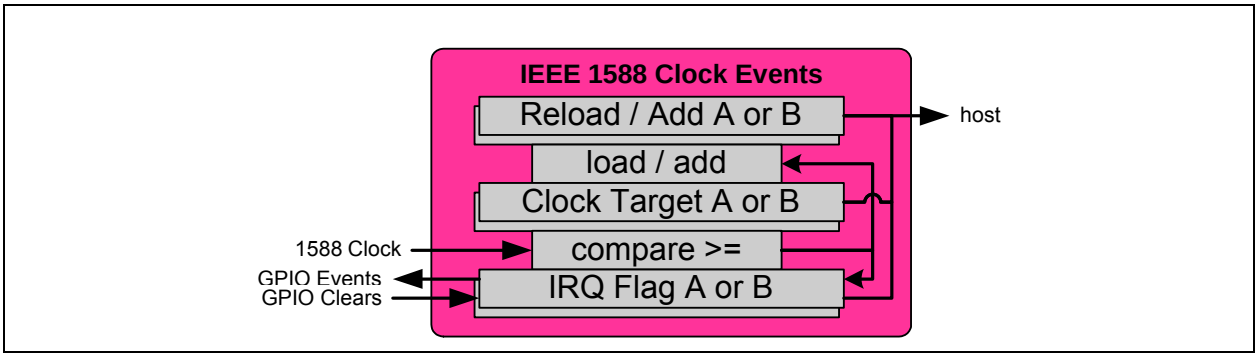
In order to perform a one-time positive adjustment to the nanoseconds portion of the 1588 Clock, the desired change is written into the 1588 Clock Step Adjustment Register (1588_CLOCK_STEP_ADJ). The Clock Step NanoSeconds (1588_CLOCK_STEP_NANOSECONDS) bit in the 1588 Command and Control Register (1588_CMD_CTL) is then set. If the addition to the nanoseconds portion results in a rollover past zero, then the seconds portion of the 1588 Clock is incremented. The normal (9, 10 or 11 ns) increment to the nanoseconds portion is suppressed for one clock. This can be compensated for by specifying an addition value 10ns higher. A side benefit is that using an addition value of 0 effectively pauses the 1588 Clock for 10ns while a value less than 10 slows the clock down just briefly. The internal sub-nanoseconds counter of the 1588 Clock is not affected by the adjustment, however, if a sub-nanoseconds counter rollover coincides with the 1588 Clock adjustment it will be missed.

In order to perform a temporary rate adjustment of the 1588 Clock, the desired temporary rate and direction are written into the 1588 Clock Temporary Rate Adjustment Register (1588_CLOCK_TEMP_RATE_ADJ) and the duration of the temporary rate, specified in reference clock cycles, is written into the 1588 Clock Temporary Rate Duration Register (1588_CLOCK_TEMP_RATE_DURATION)**.** The Clock Temporary Rate (1588_CLOCK_TEMP_RATE) bit in the 1588 Command and Control Register (1588_CMD_CTL) is then set. Once the temporary rate duration expires, the Clock Temporary Rate (1588_CLOCK_TEMP_RATE) bit will self-clear and the 1588 Clock Rate Adjustment Register (1588_CLOCK_RATE_ADJ) will once again control the 1588 Clock rate. This method of adjusting the 1588 Clock may be preferred since it avoids large discrete changes in the 1588 Clock value. For a maximum setting in both the 1588 Clock Temporary Rate Adjustment Register (1588_CLOCK_TEMP_RATE_ADJ) and 1588 Clock Temporary Rate Duration Register (1588_CLOCK_TEMP_RATE_DURATION), the 1588 Clock can be adjusted by approximately 1 second.

14.4 1588 Clock Events

The 1588 Clock Events block is responsible for generating and controlling all 1588 Clock related events. Two clock event channels, A and B, are available. The block diagram is shown in Figure 14-2.

FIGURE 14-2: 1588 CLOCK EVENT BLOCK DIAGRAM



For each clock event channel, a comparator compares the 1588 Clock with a Clock Target loaded in the 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) and 1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x).

The Clock Target Register pair requires two 32-bit write cycles, one to each half, before the register pair is affected. The writes may be in any order. There is a register pair for each clock event channel (A and B).

The Clock Target can be read by setting the Clock Target Read (1588_CLOCK_TARGET_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL). This saves the current value of the both Clock Targets (A and B) into the 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) and 1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) where they can be read.

When the 1588 Clock reaches or passes the Clock Target for a clock event channel, a clock event occurs which triggers the following:

- The maskable interrupt for that clock event channel (1588 Timer Interrupt A (1588_TIMER_INT_A) or 1588 Timer Interrupt B (1588_TIMER_INT_B)) is set in the 1588 Interrupt Status Register (1588_INT_STS).
- The Reload/Add A (RELOAD_ADD_A) or Reload/Add B (RELOAD_ADD_B) bit in the 1588 General Configuration Register (1588_GENERAL_CONFIG) is checked to determine the new Clock Target behavior:
  - –RELOAD_ADD = 1:
    The new Clock Target is loaded from the Reload / Add Registers (1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RELOAD_SEC_x) and 1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TARGET_RELOAD_NS_x)).
  - –RELOAD_ADD = 0:
    The Clock Target is incremented by the Reload / Add Registers (1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RELOAD_SEC_x) and 1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TARGET_RELOAD_NS_x)). The Clock Target NanoSeconds rolls over at 10^9 and the carry is added to the Clock Target Seconds.

The Clock Target Reload / Add Register pair requires two 32-bit write cycles, one to each half, before the register pair is affected. The writes may be in any order. There is a register pair for each clock event channel (A and B).

Note: Writing the 1588 Clock may cause the interrupt event to occur if the new 1588 Clock value is set equal to or greater than the current Clock Target.

The Clock Target reload function (RELOAD_ADD = 1) allows the Host to pre-load the next trigger time in advance. The add function (RELOAD_ADD = 0), allows for a automatic repeatable event.

14.5 1588 GPIOs

In addition to time stamping PTP packets, the 1588 Clock value can be saved into a set of clock capture registers based on the GPIO inputs. The GPIO inputs can also be used to clear the 1588 Clock Target compare event interrupt. When configured as outputs, GPIOs can be used to output a signal based on an 1588 Clock Target compare events.

Note: The IEEE 1588 Unit supports up to 3 GPIO signals.

14.5.1 1588 GPIO INPUTS

14.5.1.1 GPIO Event Clock Capture

When the GPIO pins are configured as inputs, and enabled with the GPIO Rising Edge Capture Enable 2-0 (GPIO_RE_CAPTURE_ENABLE[2:0]) or GPIO Falling Edge Capture Enable 2-0 (GPIO_FE_CAPTURE_ENABLE[2:0]) bits in the 1588 GPIO Capture Configuration Register (1588_GPIO_CAP_CONFIG), a rising or falling edge, respectively, will capture the 1588 Clock into the 1588 GPIO x Rising Edge Clock Seconds Capture Register (1588_GPIO_RE_CLOCK_SEC_CAP_x) and the 1588 GPIO x Rising Edge Clock NanoSeconds Capture Register (1588_GPIO_RE_CLOCK_NS_CAP_x) or 1588 GPIO x Falling Edge Clock Seconds Capture Register (1588_GPIO_FE_CLOCK_SEC_CAP_x) and the 1588 GPIO x Falling Edge Clock NanoSeconds Capture Register (1588_GPIO_FE_CLOCK_NS_CAP_x) where x equals the number of the active GPIO input.

GPIO inputs must be stable for greater than 40 ns to be recognized as capture events and are edge sensitive.

The GPIO inputs have a fixed capture latency of 65 ns that can be accounted for by the Host driver. The GPIO inputs have a capture latency uncertainty of +/-5 ns.

The corresponding, maskable, interrupt flags 1588 GPIO Rising Edge Interrupt (1588_GPIO_RE_INT[7:0]) or 1588 GPIO Falling Edge Interrupt (1588_GPIO_FE_INT[7:0]) in the 1588 Interrupt Status Register (1588_INT_STS) will also be set. This is in addition to the interrupts available in the General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN).

A lock enable bit is provided for each timestamp enabled GPIO, Lock Enable GPIO Rising Edge (LOCK_GPIO_RE) and Lock Enable GPIO Falling Edge (LOCK_GPIO_FE) in the 1588 GPIO Capture Configuration Register (1588_GPIO_CAP_CONFIG), which prevents the corresponding GPIO clock capture registers from being overwritten if the GPIO interrupt in 1588 Interrupt Status Register (1588_INT_STS) is already set.

14.5.1.2 GPIO Timer Interrupt Clear

The GPIO inputs can also be configured to clear the 1588 Timer Interrupt A (1588_TIMER_INT_A) or 1588 Timer Interrupt B (1588_TIMER_INT_B) in the 1588 Interrupt Status Register (1588_INT_STS) by setting the corresponding enable and select bits in the 1588 General Configuration Register (1588_GENERAL_CONFIG).

The polarity of the GPIO input is determined by the GPIO Interrupt/1588 Polarity 2-0 (GPIO_POL[2:0]) bits in the General Purpose I/O Configuration Register (GPIO_CFG).

GPIO inputs must be active for greater than 40 ns to be recognized as interrupt clear events and are edge sensitive.

14.5.2 1588 GPIO OUTPUTS

Upon detection of a Clock Target A or B compare event, the corresponding clock event channel can be configured to output a 100 ns pulse, toggle its output, or reflect its 1588 Timer Interrupt bit. The selection is made using the Clock Event Channel A Mode (CLOCK_EVENT_A) and Clock Event Channel B Mode (CLOCK_EVENT_B) bits of the 1588 General Configuration Register (1588_GENERAL_CONFIG).

A GPIO pin is configured as a 1588 event output by setting the corresponding 1588 GPIO Output Enable 2-0 (1588_GPIO_OE[2:0]) bits in the General Purpose I/O Configuration Register (GPIO_CFG). These bits override the GPIO Direction bits of the General Purpose I/O Data & Direction Register (GPIO_DATA_DIR) and allow for GPIO output generation based on the 1588 Clock Target compare event. The choice of the event channel is controlled by the 1588 GPIO Channel Select 2-0 (GPIO_CH_SEL[2:0]) bits in the General Purpose I/O Configuration Register (GPIO_CFG).

Note: The 1588 GPIO Output Enable 2-0 (1588_GPIO_OE[2:0]) bits do not override the GPIO Buffer Type 2-0 (GPIOBUF[2:0]) in the General Purpose I/O Configuration Register (GPIO_CFG).

The clock event polarity, which determines whether the 1588 GPIO output is active high or active low, is controlled by the GPIO Interrupt/1588 Polarity 2-0 (GPIO_POL[2:0]) bits in the General Purpose I/O Configuration Register (GPIO_CFG).

The GPIO outputs have a latency of approximately 40 ns when using “100 ns pulse” or “Interrupt bit” modes and 30 ns when using “toggle” mode. On chip delays contribute an uncertainty of +/-4ns to these values.

14.6 Software Triggered Clock Capture

As an alternative, the GPIO Capture registers can be used by Host software to recorded software events by specifying the GPIO register set in the 1588 Manual Capture Select 3-0 (1588_MANUAL_CAPTURE_SEL[3:0]) and setting the 1588 Manual Capture (1588_MANUAL_CAPTURE) bit in the 1588 Command and Control Register (1588_CMD_CTL).

This also causes the corresponding bit in the 1588 Interrupt Status Register (1588_INT_STS) to set.

Note: The interrupts available in the General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN) are not set by the using this method.

Note: The Lock Enable GPIO Rising Edge (LOCK_GPIO_RE) and Lock Enable GPIO Falling Edge (LOCK_GPIO_FE) bits do not apply to manual clock capture.

The full set of GPIO Capture registers is always available regardless of the number of GPIOs supported by the device.

14.7 1588 Interrupt

The IEEE 1588 unit provides multiple interrupt conditions. These include timestamp indication on the transmitter and receiver, individual GPIO input timestamp interrupts, and a clock comparison event interrupts. All 1588 interrupts are located in the 1588 Interrupt Status Register (1588_INT_STS) and are fully maskable via their respective enable bits in the 1588 Interrupt Enable Register (1588_INT_EN).

All 1588 interrupts are ANDed with their individual enables and then ORed, generating the 1588 Interrupt Event (1588_EVNT) bit of the Interrupt Status Register (INT_STS).

When configured as inputs, the GPIOs have the added functionality of clearing the 1588 Timer Interrupt A (1588_TIMER_INT_A) or 1588 Timer Interrupt B (1588_TIMER_INT_B) bits of the 1588 Interrupt Status Register (1588_INT_STS) as described in Section 14.5.1.2.

Refer to Section 8.0, "System Interrupts," on page 62 for additional information on the device interrupts.

14.8 1588 Registers

This section details the directly addressable PTP timestamp related registers.

For GPIO related registers, the wildcard “x” should be replaced with “0” through “7”.

Similarly, for Clock Compare events, the wildcard “x” should be replaced with “A” or “B”.

Port and GPIO registers share a common address space. Port vs. GPIO registers are selected by using the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL). The GPIO accessed (“x”) is set by the GPIO Select (GPIO_SEL[2:0]) field.

Note: The IEEE 1588 Unit supports 3 GPIO signals.

For an overview of the entire directly addressable register map, refer to Section 5.0, "Register Map," on page 29.

TABLE 14-1: 1588 CONTROL AND STATUS REGISTERS

BANK SELECT	ADDRESS OFFSET	Register Name (Symbol)
na	100h	1588 Command and Control Register (1588_CMD_CTL)
na	104h	1588 General Configuration Register (1588_GENERAL_CONFIG)
na	108h	1588 Interrupt Status Register (1588_INT_STS)
na	10Ch	1588 Interrupt Enable Register (1588_INT_EN)
na	110h	1588 Clock Seconds Register (1588_CLOCK_SEC)
na	114h	1588 Clock NanoSeconds Register (1588_CLOCK_NS)
na	118h	1588 Clock Sub-NanoSeconds Register (1588_CLOCK_SUBNS)
na	11Ch	1588 Clock Rate Adjustment Register (1588_CLOCK_RATE_ADJ)
na	120h	1588 Clock Temporary Rate Adjustment Register (1588_CLOCK_TEMP_RATE_-ADJ)
na	124h	1588 Clock Temporary Rate Duration Register (1588_CLOCK_TEMP_RATE_DU-RATION)
na	128h	1588 Clock Step Adjustment Register (1588_CLOCK_STEP_ADJ)
na	12Ch	1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) x=A
na	130h	1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) x=A
na	134h	1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RE-LOAD_SEC_x) x=A
na	138h	1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TAR-GET_RELOAD_NS_x) x=A
na	13Ch	1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) x=B
na	140h	1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) x=B
na	144h	1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RE-LOAD_SEC_x) x=B
na	148h	1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TAR-GET_RELOAD_NS_x) x=B
na	14Ch	1588 User MAC Address High-WORD Register (1588_USER_MAC_HI)

TABLE 14-1: 1588 CONTROL AND STATUS REGISTERS (CONTINUED)

BANK SELECT	ADDRESS OFFSET	Register Name (Symbol)
na	150h	1588 User MAC Address Low-DWORD Register (1588_USER_MAC_LO)
na	154h	1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL)
0	158h	1588 Port Latency Register (1588_LATENCY)
0	15Ch	1588 Port Asymmetry and Peer Delay Register (1588_ASYM_PEERDLY)
0	160h	1588 Port Capture Information Register (1588_CAP_INFO)
1	158h	1588 Port RX Parsing Configuration Register (1588_RX_PARSE_CONFIG)
1	15Ch	1588 Port RX Timestamp Configuration Register (1588_RX_TIMESTAMP_CONFIG)
1	160h	1588 Port RX Timestamp Insertion Configuration Register (1588_RX_TS_INSERT_CONFIG)
1	168h	1588 Port RX Filter Configuration Register (1588_RX_FILTER_CONFIG)
1	16Ch	1588 Port RX Ingress Time Seconds Register (1588_RX_INGRESS_SEC)
1	170h	1588 Port RX Ingress Time NanoSeconds Register (1588_RX_INGRESS_NS)
1	174h	1588 Port RX Message Header Register (1588_RX_MSG_HEADER)
1	178h	1588 Port RX Pdelay_Req Ingress Time Seconds Register (1588_RX_PDREQ_SEC)
1	17Ch	1588 Port RX Pdelay_Req Ingress Time NanoSeconds Register (1588_RX_PDREQ_NS)
1	180h	1588 Port RX Pdelay_Req Ingress Correction Field High Register (1588_RX_PDREQ_CF_HI)
1	184h	1588 Port RX Pdelay_Req Ingress Correction Field Low Register (1588_RX_PDREQ_CF_LOW)
1	188h	1588 Port RX Checksum Dropped Count Register (1588_RX_CHKSUM_DROPPED_CNT)
1	18Ch	1588 Port RX Filtered Count Register (1588_RX_FILTERED_CNT)
2	158h	1588 Port TX Parsing Configuration Register (1588_TX_PARSE_CONFIG)
2	15Ch	1588 Port TX Timestamp Configuration Register (1588_TX_TIMESTAMP_CONFIG)
2	164h	1588 Port TX Modification Register (1588_TX_MOD)
2	168h	1588 Port TX Modification Register 2 (1588_TX_MOD2)
2	16Ch	1588 Port TX Egress Time Seconds Register (1588_TX_EGRESS_SEC)
2	170h	1588 Port TX Egress Time NanoSeconds Register (1588_TX_EGRESS_NS)
2	174h	1588 Port TX Message Header Register (1588_TX_MSG_HEADER)
2	178h	1588 Port TX Delay_Req Egress Time Seconds Register (1588_TX_DREQ_SEC)
2	17Ch	1588 Port TX Delay_Req Egress Time NanoSeconds Register (1588_TX_DREQ_NS)

TABLE 14-1: 1588 CONTROL AND STATUS REGISTERS (CONTINUED)

BANK SELECT	ADDRESS OFFSET	Register Name (Symbol)
2	180h	1588 TX One-Step Sync Upper Seconds Register (1588_TX_ONE_STEP_SYNC_SEC)
3	15Ch	1588 GPIO Capture Configuration Register (1588_GPIO_CAP_CONFIG)
3	16Ch	1588 GPIO x Rising Edge Clock Seconds Capture Register (1588_GPIO_RE_CLOCK_SEC_CAP_x)
3	170h	1588 GPIO x Rising Edge Clock NanoSeconds Capture Register (1588_GPIO_RE_CLOCK_NS_CAP_x)
3	178h	1588 GPIO x Falling Edge Clock Seconds Capture Register (1588_GPIO_FE_CLOCK_SEC_CAP_x)
3	17Ch	1588 GPIO x Falling Edge Clock NanoSeconds Capture Register (1588_GPIO_FE_CLOCK_NS_CAP_x)

14.8.1 1588 COMMAND AND CONTROL REGISTER (1588_CMD_CTL)

Offset: 100h Size: 32 bits
Bank: na

Bits	Description	Type	Default
31:14	**RESERVED**	RO	-
13	**Clock Target Read (1588_CLOCK_TARGET_READ)** Writing a one to this bit causes the current values of both of the 1588 clock targets (A and B) to be saved into the 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) and the 1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) so they can be read. Writing a zero to this bit has no affect.	WO SC	0b
12:9	**1588 Manual Capture Select 3-0 (1588_MANUAL_CAPTURE_SEL[3:0])** These bits specify which GPIO 1588 Clock Capture Registers are used during a manual capture. Bit 3 selects the rising edge (0) or falling edge (1) registers. Bits 2-0 select the GPIO number. **Note:** All 8 GPIO register sets are available.	R/W	0000b
8	**1588 Manual Capture (1588_MANUAL_CAPTURE)** Writing a one to this bit causes the current value of the 1588 clock to be saved into the GPIO 1588 Clock Capture Registers specified above. The corresponding bit in the 1588 Interrupt Status Register (1588_INT_STS) is also set. Writing a zero to this bit has no affect.	WO SC	0b
7	**Clock Temporary Rate (1588_CLOCK_TEMP_RATE)** Writing a one to this bit enables the use of the temporary clock rate adjustment specified in the 1588 Clock Temporary Rate Adjustment Register (1588_CLOCK_TEMP_RATE_ADJ) for the duration specified in the 1588 Clock Temporary Rate Duration Register (1588_CLOCK_TEMP_RATE_DURATION). Writing a zero to this bit has no affect.	WO SC	0b
6	**Clock Step NanoSeconds (1588_CLOCK_STEP_NANOSECONDS)** Writing a one to this bit adds the value of the Clock Step Adjustment Value (1588_CLOCK_STEP_ADJ_VALUE) field in the 1588 Clock Step Adjustment Register (1588_CLOCK_STEP_ADJ) to the nanoseconds portion of the 1588 Clock. Writing a zero to this bit has no affect.	WO SC	0b

Bits	Description	Type	Default
5	**Clock Step Seconds (1588_CLOCK_STEP_SECONDS)** Writing a one to this bit adds or subtracts the value of the Clock Step Adjustment Value (1588_CLOCK_STEP_ADJ_VALUE) field in the 1588 Clock Step Adjustment Register (1588_CLOCK_STEP_ADJ) to or from the seconds portion of the 1588 Clock. The choice of adding or subtracting is set using the Clock Step Adjustment Direction (1588_CLOCK_STEP_ADJ_DIR) bit. Writing a zero to this bit has no affect.	WO SC	0b
4	**Clock Load (1588_CLOCK_LOAD)** Writing a one to this bit writes the value of the 1588 Clock Seconds Register (1588_CLOCK_SEC), the 1588 Clock NanoSeconds Register (1588_CLOCK_NS) and the 1588 Clock Sub-NanoSeconds Register (1588_CLOCK_SUBNS) into the 1588 Clock. Writing a zero to this bit has no affect.	WO SC	0b
3	**Clock Read (1588_CLOCK_READ)** Writing a one to this bit causes the current value of the 1588 clock to be saved into the 1588 Clock Seconds Register (1588_CLOCK_SEC), the 1588 Clock NanoSeconds Register (1588_CLOCK_NS) and the 1588 Clock Sub-NanoSeconds Register (1588_CLOCK_SUBNS) so it can be read. Writing a zero to this bit has no affect.	WO SC	0b
2	**1588 Enable (1588_ENABLE)** Writing a one to this bit will enable the 1588 unit. Reading this bit will return the current enabled value. Writing a zero to this bit has no affect.	R/W SC	Note 1:
1	**1588 Disable (1588_DISABLE)** Writing a one to this bit will cause the 1588 Enable (1588_ENABLE) to clear once all current frame processing is completed. No new frame processing will be started if this bit is set. Writing a zero to this bit has no affect.	WO SC	0b
0	**1588 Reset (1588_RESET)** Writing a one to this bit resets the 1588 H/W, state machines and registers and disables the 1588 unit. Any frame modifications in progress are halted at the risk of causing frame data or FCS errors. 1588_Reset should only be used once the 1588 unit is disabled as indicated by the 1588 Enable (1588_ENABLE) bit. **Note:** Writing a zero to this bit has no affect.	WO SC	0b

Note 1: The default value of this field is determined by the configuration strap 1588_enable_strap.

14.8.2 1588 GENERAL CONFIGURATION REGISTER (1588_GENERAL_CONFIG)

Offset: 104h Size: 32 bits
Bank: na

Bits	Description	Type	Default
31:19	**RESERVED**	RO	-
18:17	**RESERVED**	RO	-
16	**Time-Stamp Unit Enable (TSU_ENABLE)** This bit enables the receive and transmit functions of the time-stamp unit. The 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) bit must also be set. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
15	**GPIO 1588 Timer Interrupt B Clear Enable (GPIO_1588_TIMER_INT_B_CLEAR_EN)** This bit enables the selected GPIO to clear the 1588_TIMER_INT_B bit of the 1588 Interrupt Status Register (1588_INT_STS). The GPIO input is selected using the GPIO 1588 Timer Interrupt B Clear Select (GPIO_1588_TIMER_INT_B_CLEAR_SEL[2:0]) bits in this register. The polarity of the GPIO input is determined by GPIO Interrupt/1588 Polarity 2-0 (GPIO_POL[2:0]) in the General Purpose I/O Configuration Register (GPIO_CFG). **Note:** The GPIO must be configured as an input for this function to operate. For the clear function, GPIO inputs are edge sensitive and must be active for greater than 40 ns to be recognized.	R/W	0b
14:12	**GPIO 1588 Timer Interrupt B Clear Select (GPIO_1588_TIMER_INT_B_CLEAR_SEL[2:0])** These bits determine which GPIO is used to clear the 1588 Timer Interrupt B (1588_TIMER_INT_B) bit of the 1588 Interrupt Status Register (1588_INT_STS). **Note:** The IEEE 1588 Unit supports 3 GPIO signals.	R/W	000b

Bits	Description	Type	Default
11	**GPIO 1588 Timer Interrupt A Clear Enable (GPIO_1588_TIMER_INT_A_CLEAR_EN)** This bit enables the selected GPIO to clear the 1588 Timer Interrupt A (1588_TIMER_INT_A) bit of the 1588 Interrupt Status Register (1588_INT_STS). The GPIO input is selected using the GPIO 1588 Timer Interrupt A Clear Select (GPIO_1588_TIMER_INT_A_CLEAR_SEL[2:0]) bits in this register. The polarity of the GPIO input is determined by GPIO Interrupt/1588 Polarity 2-0 (GPIO_POL[2:0]) in the General Purpose I/O Configuration Register (GPIO_CFG). **Note:** The GPIO must be configured as an input for this function to operate. For the clear function, GPIO inputs are edge sensitive and must be active for greater than 40 ns to be recognized.	R/W	0b
10:8	**GPIO 1588 Timer Interrupt A Clear Select (GPIO_1588_TIMER_INT_A_CLEAR_SEL[2:0])** These bits determine which GPIO is used to clear the 1588_TIMER_INT_A bit of the 1588 Interrupt Status Register (1588_INT_STS). **Note:** The IEEE 1588 Unit supports 3 GPIO signals.	R/W	000b
7:6	**RESERVED**	RO	-
5:4	**Clock Event Channel B Mode (CLOCK_EVENT_B)** These bits determine the output on Clock Event Channel B when a Clock Target compare event occurs. **00: 100ns pulse output** **01: Toggle output** **10: 1588_TIMER_INT_B bit value in 1588_INT_STS_EN register output** **11: RESERVED** **Note:** The General Purpose I/O Configuration Register (GPIO_CFG) is used to enable the clock event onto the GPIO pins as well as to set the polarity and output buffer type.	R/W	00b
3:2	**Clock Event Channel A Mode (CLOCK_EVENT_A)** These bits determine the output on Clock Event Channel A when a Clock Target compare event occurs. **00: 100ns pulse output** **01: Toggle output** **10: 1588_TIMER_INT_A bit value in 1588_INT_STS_EN register output** **11: RESERVED** **Note:** The General Purpose I/O Configuration Register (GPIO_CFG) is used to enable the clock event onto the GPIO pins as well as to set the polarity and output buffer type.	R/W	00b

Bits	Description	Type	Default
1	**Reload/Add B (RELOAD_ADD_B)** This bit determines the course of action when a Clock Target compare event for Clock Event Channel B occurs. When set, the 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) and 1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) are loaded from the 1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RELOAD_SEC_x) and 1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TARGET_RELOAD_NS_x) x=B. When low, the Clock Target Registers are incremented by the Clock Target Reload Registers. **0: Increment upon a clock target compare event** **1: Reload upon a clock target compare event**	R/W	0b
0	**Reload/Add A (RELOAD_ADD_A)** This bit determines the course of action when a Clock Target compare event for Clock Event Channel A occurs. When set, the 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) and 1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) are loaded from the 1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RELOAD_SEC_x) and 1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TARGET_RELOAD_NS_x) x=A. When low, the Clock Target Registers are incremented by the Clock Target Reload Registers. **0: Increment upon a clock target compare event** **1: Reload upon a clock target compare event**	R/W	0b

14.8.3 1588 INTERRUPT STATUS REGISTER (1588_INT_STS)

Offset:	108h	Size:	32 bits
Bank:	na		

This read/write register contains the 1588 interrupt status bits.

Writing a 1 to a interrupt status bits acknowledges and clears the individual interrupt. If enabled in the 1588 Interrupt Enable Register (1588_INT_EN), these interrupt bits are cascaded into the 1588 Interrupt Event (1588_EVNT) bit of the Interrupt Status Register (INT_STS). Status bits will still reflect the status of the interrupt source regardless of whether the source is enabled as an interrupt. The 1588 Interrupt Event Enable (1588_EVNT_EN) bit of the Interrupt Enable Register (INT_EN) must also be set in order for an actual system level interrupt to occur. Refer to Section 8.0, "System Interrupts," on page 62 for additional information.

Bits	Description	Type	Default
31:24	**1588 GPIO Falling Edge Interrupt (1588_GPIO_FE_INT[7:0])** This interrupt indicates that a falling event occurred and the 1588 Clock was captured. **Note:** As 1588 capture inputs, GPIO inputs are edge sensitive and must be low for greater than 40 ns to be recognized as interrupt inputs. These bits can also be set due to a manual capture via 1588 Manual Capture (1588_MANUAL_CAPTURE).	R/WC	00h
23:16	**1588 GPIO Rising Edge Interrupt (1588_GPIO_RE_INT[7:0])** This interrupt indicates that a rising event occurred and the 1588 Clock was captured. **Note:** As 1588 capture inputs, GPIO inputs are edge sensitive and must be high for greater than 40 ns to be recognized as interrupt inputs. These bits can also be set due to a manual capture via 1588 Manual Capture (1588_MANUAL_CAPTURE).	R/WC	00h
15:13	**RESERVED**	RO	-
12	**1588 TX Timestamp Interrupt (1588_TX_TS_INT)** This interrupt indicates that a PTP packet was transmitted and its egress time stored. Up to four events, as indicated by the 1588 TX Timestamp Count (1588_TX_TS_CNT[2:0]) field in the 1588 Port Capture Information Register (1588_CAP_INFO), are buffered.	R/WC	0b
11:9	**RESERVED**	RO	-
8	**1588 RX Timestamp Interrupt (1588_RX_TS_INT)** This interrupt indicates that a PTP packet was received and its ingress time and associated data stored. Up to four events, as indicated by the 1588 RX Timestamp Count (1588_RX_TS_CNT[2:0]) field in the 1588 Port Capture Information Register (1588_CAP_INFO), are buffered.	R/WC	0b
7:2	**RESERVED**	RO	-

Bits	Description	Type	Default
1	**1588 Timer Interrupt B (1588_TIMER_INT_B)** This interrupt indicates that the 1588 clock equaled or passed the Clock Event Channel B Clock Target value in the 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) and 1588 Clock Target x Nano-Seconds Register (1588_CLOCK_TARGET_NS_x) x=B. **Note:** This bit is also cleared by an active edge on a GPIO if enabled. For the clear function, GPIO inputs are edge sensitive and must be active for greater than 40 ns to be recognized as a clear input.	R/WC	0b
0	**1588 Timer Interrupt A (1588_TIMER_INT_A)** This interrupt indicates that the 1588 clock equaled or passed the Clock Event Channel A Clock Target value in the 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) and 1588 Clock Target x Nano-Seconds Register (1588_CLOCK_TARGET_NS_x) x=A. **Note:** This bit is also cleared by an active edge on a GPIO if enabled. For the clear function, GPIO inputs are edge sensitive and must be active for greater than 40 ns to be recognized as a clear input.	R/WC	0b

14.8.4 1588 INTERRUPT ENABLE REGISTER (1588_INT_EN)

Offset:	10Ch	Size:	32 bits
Bank:	na		

This read/write register contains the 1588 interrupt enable bits.

If enabled, these interrupt bits are cascaded into the 1588 Interrupt Event (1588_EVNT) bit of the Interrupt Status Register (INT_STS). Writing a 1 to an interrupt enable bits will enable the corresponding interrupt as a source. Status bits will still reflect the status of the interrupt source regardless of whether the source is enabled as an interrupt in this register. The 1588 Interrupt Event Enable (1588_EVNT_EN) bit of the Interrupt Enable Register (INT_EN) must also be set in order for an actual system level interrupt to occur. Refer to Section 8.0, "System Interrupts," on page 62 for additional information.

Bits	Description	Type	Default
31:24	**1588 GPIO Falling Edge Interrupt Enable (1588_GPIO_FE_EN[7:0])**	R/W	00h
23:16	**1588 GPIO Rising Edge Interrupt Enable (1588_GPIO_RE_EN[7:0])**	R/W	00h
15:13	**RESERVED**	RO	-
12	**1588 TX Timestamp Enable (1588_TX_TS_EN)**	R/W	0b
11:9	**RESERVED**	RO	-
8	**1588 RX Timestamp Enable (1588_RX_TS_EN)**	R/W	0b
7:2	**RESERVED**	RO	-
1	**1588 Timer B Interrupt Enable (1588_TIMER_EN_B)**	R/W	0b
0	**1588 Timer A Interrupt Enable (1588_TIMER_EN_A)**	R/W	0b

14.8.5 1588 CLOCK SECONDS REGISTER (1588_CLOCK_SEC)

Offset: 110h Size: 32 bits
Bank: na

This register contains the seconds portion of the 1588 Clock. It is used to read the 1588 Clock following the setting of the Clock Read (1588_CLOCK_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL) and to directly change the 1588 Clock when the Clock Load (1588_CLOCK_LOAD) bit is set.

Bits	Description	Type	Default
31:0	**Clock Seconds (1588_CLOCK_SEC)** This field contains the seconds portion of the 1588 Clock.	R/W	00000000h

Note: The value read is the saved value of the 1588 Clock when the Clock Read (1588_CLOCK_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL) is set.

14.8.6 1588 CLOCK NANOSECONDS REGISTER (1588_CLOCK_NS)

Offset: 114h Size: 32 bits
Bank: na

This register contains the nanoseconds portion of the 1588 Clock. It is used to read the 1588 Clock following the setting of the Clock Read (1588_CLOCK_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL) and to directly change the 1588 Clock when the Clock Load (1588_CLOCK_LOAD) bit is set.

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29:0	**Clock NanoSeconds (1588_CLOCK_NS)** This field contains the nanoseconds portion of the 1588 Clock.	R/W	00000000h

Note: The value read is the saved value of the 1588 Clock when the Clock Read (1588_CLOCK_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL) is set.

14.8.7 1588 CLOCK SUB-NANOSECONDS REGISTER (1588_CLOCK_SUBNS)

Offset: 118h Size: 32 bits
Bank: na

This register contains the sub-nanoseconds portion of the 1588 Clock. It is used to read the 1588 Clock following the setting of the Clock Read (1588_CLOCK_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL) and to directly change the 1588 Clock when the Clock Load (1588_CLOCK_LOAD) bit is set.

Bits	Description	Type	Default
31:0	**Clock Sub-NanoSeconds (1588_CLOCK_SUBNS)** This field contains the sub-nanoseconds portion of the 1588 Clock.	R/W	00000000h

Note: The value read is the saved value of the 1588 Clock when the Clock Read (1588_CLOCK_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL) is set.

14.8.8 1588 CLOCK RATE ADJUSTMENT REGISTER (1588_CLOCK_RATE_ADJ)

Offset: 11Ch Size: 32 bits
Bank: na

This register is used to adjust the rate of the 1588 Clock. Every 10 ns, 1588 Clock is normally incremented by 10 ns. This register is used to occasionally change that increment to 9 or 11 ns.

Bits	Description	Type	Default
31	**Clock Rate Adjustment Direction (1588_CLOCK_RATE_ADJ_DIR)** This field specifies if the 1588 Rate Adjustment causes the 1588 Clock to be faster or slower than the reference clock. **0 = slower (1588 Clock increments by 9 ns)** **1 = faster (1588 Clock increments by 11 ns)**	R/W	0b
30	**RESERVED**	RO	-
29:0	**Clock Rate Adjustment Value (1588_CLOCK_RATE_ADJ_VALUE)** This field indicates an adjustment to the reference clock period of the 1588 Clock in units of 2^{-32}ns. On each 10 ns reference clock cycle, this value is added to the 32-bit sub-nanoseconds portion of the 1588 Clock. When the sub-nanoseconds portion wraps around to zero, the 1588 Clock will be adjusted by 1ns.	R/W	00000000h

14.8.9 1588 CLOCK TEMPORARY RATE ADJUSTMENT REGISTER (1588_CLOCK_TEMP_RATE_ADJ)

Offset: 120h Size: 32 bits
Bank: na

This register is used to temporarily adjust the rate of the 1588 Clock. Every 10 ns, 1588 Clock is normally incremented by 10 ns. This register is used to occasionally change that increment to 9 or 11 ns.

Bits	Description	Type	Default
31	**Clock Temporary Rate Adjustment Direction (1588_CLOCK_TEMP_RATE_ADJ_DIR)** This field specifies if the 1588 Temporary Rate Adjustment causes the 1588 Clock to be faster or slower than the reference clock. **0 = slower (1588 Clock increments by 9 ns)** **1 = faster (1588 Clock increments by 11 ns)**	R/W	0b
30	**RESERVED**	RO	-
29:0	**Clock Temporary Rate Adjustment Value (1588_CLOCK_TEMP_RATE_ADJ_VALUE)** This field indicates a temporary adjustment to the reference clock period of the 1588 Clock in units of 2^{-32}ns. On each 10ns reference clock cycle, this value is added to the 32-bit sub-nanoseconds portion of the 1588 Clock. When the sub-nanoseconds portion wraps around to zero, the 1588 Clock will be adjusted by 1ns (a 9 or 11 ns increment instead of the normal 10ns).	R/W	00000000h

14.8.10 1588 CLOCK TEMPORARY RATE DURATION REGISTER (1588_CLOCK_TEMP_RATE_DURATION)

Offset: 124h Size: 32 bits
Bank: na

This register specifies the active duration of the temporary clock rate adjustment.

Bits	Description	Type	Default
31:0	**Clock Temporary Rate Duration (1588_CLOCK_TEMP_RATE_DURATION)** This field specifies the duration of the temporary rate adjustment in reference clock cycles.	R/W	00000000h

14.8.11 1588 CLOCK STEP ADJUSTMENT REGISTER (1588_CLOCK_STEP_ADJ)

Offset: 128h Size: 32 bits
Bank: na

This register is used to perform a one-time adjustment to either the seconds portion or the nanoseconds portion of the 1588 Clock. The amount and direction can be specified.

Bits	Description	Type	Default
31	**Clock Step Adjustment Direction (1588_CLOCK_STEP_ADJ_DIR)** This field specifies if the Clock Step Adjustment Value (1588_CLOCK_-STEP_ADJ_VALUE) is added to or subtracted from the 1588 Clock. **0 = subtracted** **1 = added** **Note:** Only addition is supported for the nanoseconds portion of the 1588 Clock	R/W	0b
30	**RESERVED**	RO	-
29:0	**Clock Step Adjustment Value (1588_CLOCK_STEP_ADJ_VALUE)** When the nanoseconds portion of the 1588 Clock is being adjusted, this field specifies the amount to add. This is in lieu of the normal 9, 10 or 11 ns increment. When the seconds portion of the 1588 Clock is being adjusted, the lower 4 bits of this field specify the amount to add to or subtract.	R/W	00000000h

14.8.12 1588 CLOCK TARGET X SECONDS REGISTER (1588_CLOCK_TARGET_SEC_X)

Offset:	Channel A: 12Ch Channel B: 13Ch	Size:	32 bits
Bank:	Channel A: na Channel B: na		

This read/write register combined with 1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) form the 1588 Clock Target value. The 1588 Clock Target value is compared to the current 1588 Clock value and can be used to trigger an interrupt upon at match. Refer to Section 14.4, "1588 Clock Events" for additional information.

Bits	Description	Type	Default
31:0	**Clock Target Seconds (CLOCK_TARGET_SEC)** This field contains the seconds portion of the 1588 Clock Compare value.	R/W	00000000h

Note: Both this register and the 1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x) must be written for either to be affected.

Note: The value read is the saved value of the 1588 Clock Target when the Clock Target Read (1588_CLOCK_TARGET_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL) is set.

Note: When the Clock Target Read (1588_CLOCK_TARGET_READ) bit is set, the previous value written to this register is overwritten. Normally, a read command should not be requested in between writing this register and the 1588 Clock Target x NanoSeconds Register (1588_CLOCK_TARGET_NS_x).

14.8.13 1588 CLOCK TARGET X NANOSECONDS REGISTER (1588_CLOCK_TARGET_NS_X)

Offset:	Channel A: 130h Channel B: 140h	Size:	32 bits
Bank:	Channel A: na Channel B: na		

This read/write register combined with 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) form the 1588 Clock Target value. The 1588 Clock Target value is compared to the current 1588 Clock value and can be used to trigger an interrupt upon at match. Refer to Section 14.4, "1588 Clock Events" for additional information.

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29:0	**Clock Target NanoSeconds (CLOCK_TARGET_NS)** This field contains the nanoseconds portion of the 1588 Clock Compare value.	R/W	00000000h

Note: Both this register and the 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x) must be written for either to be affected.

Note: The value read is the saved value of the 1588 Clock Target when the Clock Target Read (1588_CLOCK_TARGET_READ) bit in the 1588 Command and Control Register (1588_CMD_CTL) is set.

Note: When the Clock Target Read (1588_CLOCK_TARGET_READ) bit is set, the previous value written to this register is overwritten. Normally, a read command should not be requested in between writing this register and the 1588 Clock Target x Seconds Register (1588_CLOCK_TARGET_SEC_x).

14.8.14 1588 CLOCK TARGET X RELOAD / ADD SECONDS REGISTER (1588_CLOCK_TARGET_RELOAD_SEC_X)

Offset:	Channel A: 134h Channel B: 144h	Size:	32 bits
Bank:	Channel A: na Channel B: na		

This read/write register combined with 1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TARGET_RELOAD_NS_x) form the 1588 Clock Target Reload value. The 1588 Clock Target Reload is the value that is reloaded or added to the 1588 Clock Compare value when a clock compare event occurs. Refer to Section 14.4, "1588 Clock Events" for additional information.

Bits	Description	Type	Default
31:0	**Clock Target Reload Seconds (CLOCK_TARGET_RELOAD_SEC)** This field contains the seconds portion of the 1588 Clock Target Reload value that is reloaded to the 1588 Clock Compare value.	R/W	00000000h

Note: Both this register and the 1588 Clock Target x Reload / Add NanoSeconds Register (1588_CLOCK_TARGET_RELOAD_NS_x) must be written for either to be affected.

14.8.15 1588 CLOCK TARGET X RELOAD / ADD NANOSECONDS REGISTER (1588_CLOCK_TARGET_RELOAD_NS_X)

Offset:	Channel A: 138h Channel B: 148h	Size:	32 bits
Bank:	Channel A: na Channel B: na		

This read/write register combined with 1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RELOAD_SEC_x) form the 1588 Clock Target Reload value. The 1588 Clock Target Reload is the value that is reloaded or added to the 1588 Clock Compare value when a clock compare event occurs. Refer to Section 14.4, "1588 Clock Events" for additional information.

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29:0	**Clock Target Reload NanoSeconds (CLOCK_TARGET_RELOAD_NS)** This field contains the nanoseconds portion of the 1588 Clock Target Reload value that is reloaded to the 1588 Clock Compare value.	R/W	00000000h

Note: Both this register and the 1588 Clock Target x Reload / Add Seconds Register (1588_CLOCK_TARGET_RELOAD_SEC_x) must be written for either to be affected.

14.8.16 1588 USER MAC ADDRESS HIGH-WORD REGISTER (1588_USER_MAC_HI)

Offset:	14Ch	Size:	32 bits
Bank:	na		

This read/write register combined with the 1588 User MAC Address Low-DWORD Register (1588_USER_MAC_LO) forms the 48-bit user defined MAC address. The Auxiliary MAC address can be enabled for each protocol via their respective User Defined MAC Address Enable bit in the 1588 Port RX Parsing Configuration Register (1588_RX-_PARSE_CONFIG).

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:0	**User MAC Address High (USER_MAC_HI)** This field contains the high 16 bits of the user defined MAC address used for PTP packet detection. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0000h

14.8.17 1588 USER MAC ADDRESS LOW-DWORD REGISTER (1588_USER_MAC_LO)

Offset: 150h Size: 32 bits
Bank: na

This read/write register combined with the 1588 User MAC Address High-WORD Register (1588_USER_MAC_HI) forms the 48-bit user defined MAC address. The Auxiliary MAC address can be enabled for each protocol via their respective User Defined MAC Address Enable bit in the 1588 Port RX Parsing Configuration Register (1588_RX-_PARSE_CONFIG).

Bits	Description	Type	Default
31:0	**User MAC Address Low (USER_MAC_LO)** This field contains the low 32 bits of the user defined MAC address used for PTP packet detection. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	00000000h

14.8.18 1588 BANK PORT GPIO SELECT REGISTER (1588_BANK_PORT_GPIO_SEL)

Offset: 154h Size: 32 bits
Bank: na

Bits	Description	Type	Default
31:11	**RESERVED**	RO	-
10:8	**GPIO Select (GPIO_SEL[2:0])** This field specifies which GPIO the various GPIO x registers will access.	R/W	000b
7:3	**RESERVED**	RO	-
2:0	**Bank Select (BANK_SEL[2:0]** This field specifies which bank of registers is accessed. **000: Port General** **001: Port RX** **010: Port TX** **011: GPIOs** **1xx: Reserved**	R/W	000b

14.8.19 1588 PORT LATENCY REGISTER (1588_LATENCY)

Offset: 158h Size: 32 bits
Bank: 0

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:16	**TX Latency (TX_LATENCY[15:0])** This field specifies the egress delay in nanoseconds between the PTP timestamp point and the network medium. The setting is used to adjust the internally captured 1588 clock value such that the resultant timestamp more accurately corresponds to the start of the frame's first symbol after the SFD on the network medium. The value depends on the port mode. Typical values are: • 100BASE-TX: 95ns • 100BASE-FX: 68ns plus the receive latency of the fiber transceiver • 10BASE-T: 1139ns **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	95 Note 2
15:0	**RX Latency (RX_LATENCY[15:0])** This field specifies the ingress delay in nanoseconds between the network medium and the PTP timestamp point. The setting is used to adjust the internally captured 1588 clock value such that the resultant timestamp more accurately corresponds to the start of the frame's first symbol after the SFD on the network medium. The value depends on the port mode. Typical values are: • 100BASE-TX: 285ns • 100BASE-FX: 231ns plus the receive latency of the fiber transceiver • 10BASE-T: 1674ns **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	285 Note 2

Note 2: The default value is appropriate for 100BASE-TX mode. For other modes (100BASE-FX or 10BASE-T) the proper value needs to be set via software or EEPROM.

14.8.20 1588 PORT ASYMMETRY AND PEER DELAY REGISTER (1588_ASYM_PEERDLY)

Offset: 15Ch Size: 32 bits
Bank: 0

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:16	**Port Delay Asymmetry (DELAY_ASYM[15:0])** This field specifies the previously known delay asymmetry in nanoseconds. This is a signed 2's complement number. Positive values occur when the master-to-slave or responder-to-requestor propagation time is longer than the slave-to-master or requestor-to-responder propagation time. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0000h
15:0	**RX Peer Delay (RX_PEER_DELAY[15:0])** This field specifies the measured peer delay in nanoseconds used during peer-to-peer mode.	R/W	0000h

14.8.21 1588 PORT CAPTURE INFORMATION REGISTER (1588_CAP_INFO)

Offset: 160h Size: 32 bits
Bank: 0

This read only register provides information about the receive and transmit capture buffers.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:7	**RESERVED**	RO	-
6:4	**1588 TX Timestamp Count (1588_TX_TS_CNT[2:0])** This field indicates how many transmit timestamps are available to be read. It is incremented when a PTP packet is transmitted and decremented when the 1588 TX Timestamp Interrupt (1588_TX_TS_INT) bit is written with a 1.	RO	000b
3	**RESERVED**	RO	-
2:0	**1588 RX Timestamp Count (1588_RX_TS_CNT[2:0])** This field indicates how many receive timestamps are available to be read. It is incremented when a PTP packet is received and decremented when the 1588 RX Timestamp Interrupt (1588_RX_TS_INT) bit is written with a 1.	RO	000b

14.8.22 1588 PORT RX PARSING CONFIGURATION REGISTER (1588_RX_PARSE_CONFIG)

Offset: 158h Size: 32 bits
Bank: 1

This register is used to configure the PTP receive message detection.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:15	**RESERVED**	RO	-
14	**RX Layer 2 Address 1 Enable (RX_LAYER2_ADD1_EN)** This bit enables the Layer 2 MAC address of 01:80:C2:00:00:0E for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
13	**RX Layer 2 Address 2 Enable (RX_LAYER2_ADD2_EN)** This bit enables the Layer 2 MAC address of 01:1B:19:00:00:00 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
12	**RX Address 1 Enable (RX_ADD1_EN)** This bit enables the IPv4 MAC address of 01:00:5E:00:01:81 and IPv4 destination address of 224.0.1.129 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:01:81 and IPv6 destination address of FF0X:0:0:0:0:0:0:181 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
11	**RX Address 2 Enable (RX_ADD2_EN)** This bit enables the IPv4 MAC address of 01:00:5E:00:01:82 and IPv4 destination address of 224.0.1.130 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:01:82 and IPv6 destination address of FF0X:0:0:0:0:0:0:182 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b

Bits	Description	Type	Default
10	**RX Address 3 Enable (RX_ADD3_EN)** This bit enables the IPv4 MAC address of 01:00:5E:00:01:83 and IPv4 destination address of 224.0.1.131 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:01:83 and IPv6 destination address of FF0X:0:0:0:0:0:0:183 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
9	**RX Address 4 Enable (RX_ADD4_EN)** This bit enables the IPv4 MAC address of 01:00:5E:00:01:84 and IPv4 destination address of 224.0.1.132 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:01:84 and IPv6 destination address of FF0X:0:0:0:0:0:0:184 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
8	**RX Address 5 Enable (RX_ADD5_EN)** This bit enables the IPv4 MAC address of 01:00:5e:00:00:6B and IPv4 destination address of 224.0.0.107 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:00:6B and IPv6 destination address of FF02:0:0:0:0:0:0:6B for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
7	**RX User Defined Layer 2 MAC Address Enable (RX_LAYER2_USER_MAC_EN)** This bit enables a user defined Layer 2 MAC address in PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
6	**RX User Defined IPv6 MAC Address Enable (RX_IPV6_USER_MAC_EN)** This bit enables a user defined IPv6 MAC address in PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
5	**RX User Defined IPv4 MAC Address Enable (RX_IPV4_USER_MAC_EN)** This bit enables the user defined IPv4 MAC address in PTP messages. The address is defined via the 1588 User MAC Address High-WORD Register (1588_USER_MAC_HI) and the 1588 User MAC Address Low-DWORD Register (1588_USER_MAC_LO). **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b

Bits	Description	Type	Default
4	**RX IP Address Enable (RX_IP_ADDR_EN)** This bit enables the checking of the IP destination address in PTP messages for both IPv4 and IPv6 formats. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
3	**RX MAC Address Enable (RX_MAC_ADDR_EN)** This bit enables the checking of the MAC destination address in PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
2	**RX Layer 2 Enable (RX_LAYER2_EN)** This bit enables the detection of the layer 2 formatted PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
1	**RX IPv6 Enable (RX_IPV6_EN)** This bit enables the detection of the UDP/IPv6 formatted PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
0	**RX IPv4 Enable (RX_IPV4_EN)** This bit enables the detection of the UDP/IPv4 formatted PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b

14.8.23 1588 PORT RX TIMESTAMP CONFIGURATION REGISTER (1588_RX_TIMESTAMP_CONFIG)

Offset: 15Ch Size: 32 bits
Bank: 1

This register is used to configure PTP receive message timestamping.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:24	**RX PTP Domain (RX_PTP_DOMAIN[7:0])** This field specifies the PTP domain in use. If RX PTP Domain Match Enable (RX_PTP_DOMAIN_EN) is set, the domainNumber in the PTP message must matches the value in this field in order to recorded the ingress time. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	00h
23	**RX PTP Domain Match Enable (RX_PTP_DOMAIN_EN)** When this bit is set, the domainNumber in the PTP message is checked against the value in RX PTP Domain (RX_PTP_DOMAIN[7:0]). **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
22	**RX PTP Alternate Master Enable (RX_PTP_ALT_MASTER_EN)** When this bit is set, the alternateMasterFlag in the PTP message is checked for a zero value. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
21	**RX PTP UDP Checksum Check Disable (RX_PTP_UDP_CHKSUM_DIS)** When this bit is cleared, ingress times are not saved and ingress messages are not filtered if the frame has an invalid UDP checksum. When this bit is set, the UDP checksum check is bypassed and the ingress time is saved and ingress messages are filtered regardless. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
20	**RX PTP FCS Check Disable (RX_PTP_FCS_DIS)** When this bit is cleared, ingress times are not saved and ingress messages are not filtered if the frame has an invalid FCS. When this bit is set, the FCS check is bypassed. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b

Bits	Description	Type	Default
19:16	**RX PTP Version (RX_PTP_VERSION[3:0])** This field specifies the PTP version in use. A setting of 0 allows any PTP version. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	2h
15:0	**RX PTP Message Type Enable (RX_PTP_MESSAGE_EN[15:0])** These bits individually enable timestamping of their respective message types. Bit 0 of this field corresponds to a message type value of 0 (Sync), bit 1 to message type value 1 (Delay_Req), etc. Typically Sync, Delay_Req, PDelay_Req and PDelay_Resp messages are enabled.	R/W	0000h

14.8.24 1588 PORT RX TIMESTAMP INSERTION CONFIGURATION REGISTER (1588_RX_TS_INSERT_CONFIG)

Offset: 160h Size: 32 bits
Bank: 1

This register is used to configure PTP message timestamp insertion.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:18	**RESERVED**	RO	-
17	**RX PTP Insert Delay Request Egress in Delay Response Enable (RX_PTP_INSERT_DREQ_DRESP_EN)** When this bit is set, the egress time of the last Delay_Req packet is inserted into received Delay_Resp packets. This bit has no affect if RX_PTP_INSERT_TS_EN is a low or if detection of the Delay_Resp message type is not enabled.	R/W	0b
16	**RX PTP Bad UDP Checksum Force Error Disable (RX_PTP_BAD_UDP_CHKSUM_FORCE_ERR_DIS)** When this bit is cleared, ingress packets that have an invalid UDP checksum will have a receive symbol error forced if the packet is modified for timestamp or correction field reasons. When this bit is set, the UDP checksum check is bypassed. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
15	**RX PTP Insert Timestamp Seconds Enable (RX_PTP_INSERT_TS_SEC_EN)** When RX_PTP_INSERT_TS_EN is set, this bit enables bits 3:0 of the seconds portion of the receive ingress time to be inserted into the PTP message. This bit has no affect if RX_PTP_INSERT_TS_EN is a low.	R/W	0b
14	**RESERVED**	RO	-
13:8	**RX PTP Insert Timestamp Seconds Offset (RX_PTP_INSERT_TS_SEC_OFFSET[5:0])** This field specifies the offset into the PTP header where the seconds portion of the receive ingress time is inserted. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	000101b
7	**RX PTP Insert Timestamp Enable (RX_PTP_INSERT_TS_EN)** When set, receive ingress times are inserted into the PTP message.	R/W	0b
6	**RESERVED**	RO	-

Bits	Description	Type	Default
5:0	**RX PTP Insert Timestamp Offset (RX_PTP_INSERT_TS_OFFSET[5:0])** This field specifies the offset into the PTP header where the receive ingress time is inserted. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	010000b

14.8.25 1588 PORT RX FILTER CONFIGURATION REGISTER (1588_RX_FILTER_CONFIG)

Offset:	168h	Size:	32 bits
Bank:	1		

This register is used to configure PTP message filtering.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:19	**RESERVED**	RO	-
18	**RX PTP Alternate Master Filter Enable (RX_PTP_ALT_MASTER_FLTR_EN)** This bit enables message filtering based on the alternateMasterFlag flagField bit. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
17	**RX PTP Domain Filter Enable (RX_PTP_DOMAIN_FLTR_EN)** This bit enables message filtering based on the PTP domain. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
16	**RX PTP Version Filter Enable (RX_PTP_VERSION_FLTR_EN)** This bit enables message filtering based on the PTP version. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
15:0	**RX PTP Message Type Filter Enable (RX_PTP_MSG_FLTR_EN[15:0])** These bits enable individual message filtering. Bit 0 of this field corresponds to a message type value of 0 (Sync), bit 1 to message type value 1 (Delay_Req), etc. Typically Delay_Req and Delay_Resp messages are filtered for peer-to-peer transparent clocks.	R/W	0000h

14.8.26 1588 PORT RX INGRESS TIME SECONDS REGISTER (1588_RX_INGRESS_SEC)

Offset:	16Ch	Size:	32 bits
Bank:	1		

This read only register combined with the 1588 Port RX Ingress Time NanoSeconds Register (1588_RX_INGRESS_NS) contains the RX timestamp captures. Up to four captures are buffered.

Note: Values are only valid if the 1588 RX Timestamp Count (1588_RX_TS_CNT[2:0]) field indicates that at least one timestamp is available.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:0	**Timestamp Seconds (TS_SEC)** This field contains the seconds portion of the receive ingress time.	RO	00000000h

14.8.27 1588 PORT RX INGRESS TIME NANOSECONDS REGISTER (1588_RX_INGRESS_NS)

Offset: 170h Size: 32 bits
Bank: 1

This read only register combined with the 1588 Port RX Ingress Time Seconds Register (1588_RX_INGRESS_SEC) contains the RX timestamp capture. Up to four captures are buffered.

Note: Values are only valid if the 1588 RX Timestamp Count (1588_RX_TS_CNT[2:0]) field indicates that at least one timestamp is available.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29:0	**Timestamp NanoSeconds (TS_NS)** This field contains the nanoseconds portion of the receive ingress time.	RO	00000000h

14.8.28 1588 PORT RX MESSAGE HEADER REGISTER (1588_RX_MSG_HEADER)

Offset:	174h	Size:	32 bits
Bank:	1		

This read only register contains the RX message header. Up to four captures are buffered.

Note: Values are only valid if the 1588 RX Timestamp Count (1588_RX_TS_CNT[2:0]) field indicates that at least one timestamp is available.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:20	**Source Port Identity CRC (SRC_PRT_CRC)** This field contains the 12-bit CRC of the sourcePortIdentity field of the received PTP packet.	RO	000h
19:16	**Message Type (MSG_TYPE)** This field contains the messageType field of the received PTP packet.	RO	0h
15:0	**Sequence ID (SEQ_ID)** This field contains the sequenceId field of the received PTP packet.	RO	0000h

14.8.29 1588 PORT RX PDELAY_REQ INGRESS TIME SECONDS REGISTER (1588_RX_PDREQ_SEC)

Offset: 178h Size: 32 bits
Bank: 1

This register combined with the 1588 Port RX Pdelay_Req Ingress Time NanoSeconds Register (1588_RX_PDREQ_NS) contains the ingress time of the last Pdelay_Req message. This register is automatically updated if the Auto Update (AUTO) bit is set.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:4	**RESERVED**	RO	-
3:0	**Timestamp Seconds (TS_SEC)** This field contains the seconds portion of the receive ingress time.	R/W	0h

14.8.30 1588 PORT RX PDELAY_REQ INGRESS TIME NANOSECONDS REGISTER (1588_RX_PDREQ_NS)

Offset:	17Ch	Size:	32 bits
Bank:	1		

This register combined with the 1588 Port RX Pdelay_Req Ingress Time Seconds Register (1588_RX_PDREQ_SEC) contains the ingress time of the last Pdelay_Req message. This register is automatically updated if the Auto Update (AUTO) bit is set.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31	**Auto Update (AUTO)** If this bit is set, the TS_NS field in this register, the TS_SEC field in 1588_RX_PDREQ_SEC and the CF field in 1588_RX_PDREQ_CF_HI / 1588_RX_PDREQ_CF_LO are updated when a PDelay_Req message is received. When cleared, S/W is responsible to maintain those fields. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
30	**RESERVED**	RO	-
29:0	**Timestamp NanoSeconds (TS_NS)** This field contains the nanoseconds portion of the receive ingress time.	R/W	00000000h

14.8.31 1588 PORT RX PDELAY_REQ INGRESS CORRECTION FIELD HIGH REGISTER (1588_RX_PDREQ_CF_HI)

Offset:	180h	Size:	32 bits
Bank:	1		

This register combined with the 1588 Port RX Pdelay_Req Ingress Correction Field Low Register (1588_RX_PDREQ_CF_LOW) contains the correction field from the last Pdelay_Req message. Only the nanoseconds portion is used.

This register is automatically updated if the Auto Update (AUTO) bit is set.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:0	**Correction Field (CF[63:32])** This field contains the upper 32 bits of the correction field.	R/W	00000000h

14.8.32 1588 PORT RX PDELAY_REQ INGRESS CORRECTION FIELD LOW REGISTER (1588_RX_PDREQ_CF_LOW)

Offset:	184h	Size:	32 bits
Bank:	1		

This register combined with the 1588 Port RX Pdelay_Req Ingress Correction Field High Register (1588_RX_PDREQ_CF_HI) contains the correction field from the last Pdelay_Req message. Only the nanoseconds portion is used.

This register is automatically updated if the Auto Update (AUTO) bit is set.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:16	**Correction Field (CF[31:16])** This field contains the low middle 16 bits of the correction field.	R/W	0000h
15:0	**RESERVED**	RO	-

14.8.33 1588 PORT RX CHECKSUM DROPPED COUNT REGISTER (1588_RX_CHKSUM_DROPPED_CNT)

Offset:	188h	Size:	32 bits
Bank:	1		

This register counts the number of packets dropped at ingress due to a bad UDP checksum. The packet will also be counted as an error by the receiving MAC.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:0	**Bad Checksum Dropped Count (BAD_CHKSUM_DROPPED_CNT[31:0])** This field is a count of packets dropped at ingress due to a bad UDP checksum. It can be cleared by writing a zero value at the risk of losing any previous count. **Note:** This counter will stop at its maximum value of FFFF_FFFFh. Minimum rollover time at 100 Mbps is approximately 481 hours.	R/W	00000000h

14.8.34 1588 PORT RX FILTERED COUNT REGISTER (1588_RX_FILTERED_CNT)

Offset:	18Ch	Size:	32 bits
Bank:	1		

This register counts the number of packets filtered at ingress due to Ingress Message Filtering. The packet will also be counted as an error by the receiving MAC.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:0	**Filtered Count (FILTERED_CNT[31:0])** This field is a count of packets dropped at ingress due to Ingress Message Filtering. It can be cleared by writing a zero value at the risk of losing any previous count. **Note:** This counter will stop at its maximum value of FFFF_FFFFh. Minimum rollover time at 100 Mbps is approximately 481 hours.	R/W	00000000h

14.8.35 1588 PORT TX PARSING CONFIGURATION REGISTER (1588_TX_PARSE_CONFIG)

Offset: 158h Size: 32 bits
Bank: 2

This register is used to configure the PTP transmit message detection.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:15	**RESERVED**	RO	-
14	**TX Layer 2 Address 1 Enable (TX_LAYER2_ADD1_EN)** This bit enables the Layer 2 MAC address of 01:80:C2:00:00:0E for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
13	**TX Layer 2 Address 2 Enable (TX_LAYER2_ADD2_EN)** This bit enables the Layer 2 MAC address of 01:1B:19:00:00:00 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
12	**TX Address 1 Enable (TX_ADD1_EN)** This bit enables the IPv4 MAC address of 01:00:5E:00:01:81 and IPv4 destination address of 224.0.1.129 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:01:81 and IPv6 destination address of FF0X:0:0:0:0:0:0:181 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
11	**TX Address 2 Enable (TX_ADD2_EN)** This bit enables the IPv4 MAC address of 01:00:5E:00:01:82 and IPv4 destination address of 224.0.1.130 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:01:82 and IPv6 destination address of FF0X:0:0:0:0:0:0:182 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b

Bits	Description	Type	Default
10	**TX Address 3 Enable (TX_ADD3_EN)** This bit enables the IPv4 MAC address of 01:00:5E:00:01:83 and IPv4 destination address of 224.0.1.131 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:01:83 and IPv6 destination address of FF0X:0:0:0:0:0:0:183 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
9	**TX Address 4 Enable (TX_ADD4_EN)** This bit enables the IPv4 MAC address of 01:00:5E:00:01:84 and IPv4 destination address of 224.0.1.132 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:01:84 and IPv6 destination address of FF0X:0:0:0:0:0:0:184 for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
8	**TX Address 5 Enable (TX_ADD5_EN)** This bit enables the IPv4 MAC address of 01:00:5e:00:00:6B and IPv4 destination address of 224.0.0.107 for PTP packets. This bit enables the IPv6 MAC address of 33:33:00:00:00:6B and IPv6 destination address of FF02:0:0:0:0:0:0:6B for PTP packets. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
7	**TX User Defined Layer 2 MAC Address Enable (TX_LAYER2_USER_MAC_EN)** This bit enables a user defined Layer 2 MAC address in PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
6	**TX User Defined IPv6 MAC Address Enable (TX_IPV6_USER_MAC_EN)** This bit enables a user defined IPv6 MAC address in PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
5	**TX User Defined IPv4 MAC Address Enable (TX_IPV4_USER_MAC_EN)** This bit enables the user defined IPv4 MAC address in PTP messages. The address is defined via the 1588 User MAC Address High-WORD Register (1588_USER_MAC_HI) and the 1588 User MAC Address Low-DWORD Register (1588_USER_MAC_LO). **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b

Bits	Description	Type	Default
4	**TX IP Address Enable (TX_IP_ADDR_EN)** This bit enables the checking of the IP destination address in PTP messages for both IPv4 and IPv6 formats. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
3	**TX MAC Address Enable (TX_MAC_ADDR_EN)** This bit enables the checking of the MAC destination address in PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
2	**TX Layer 2 Enable (TX_LAYER2_EN)** This bit enables the detection of the layer 2 formatted PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
1	**TX IPv6 Enable (TX_IPV6_EN)** This bit enables the detection of the UDP/IPv6 formatted PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b
0	**TX IPv4 Enable (TX_IPV4_EN)** This bit enables the detection of the UDP/IPv4 formatted PTP messages. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	1b

14.8.36 1588 PORT TX TIMESTAMP CONFIGURATION REGISTER (1588_TX_TIMESTAMP_CONFIG)

Offset:	15Ch	Size:	32 bits
Bank:	2		

This register is used to configure PTP transmit message timestamping.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:24	**TX PTP Domain (TX_PTP_DOMAIN[7:0])** This field specifies the PTP domain in use. If TX PTP Domain Match Enable (TX_PTP_DOMAIN_EN) is set, the domainNumber in the PTP message must match the value in this field in order to recorded the egress time. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	00h
23	**TX PTP Domain Match Enable (TX_PTP_DOMAIN_EN)** When this bit is set, the domainNumber in the PTP message is checked against the value in TX PTP Domain (TX_PTP_DOMAIN[7:0]). **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
22	**TX PTP Alternate Master Enable (TX_PTP_ALT_MASTER_EN)** When this bit is set, the alternateMasterFlag in the PTP message is checked for a zero value. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
21	**TX PTP UDP Checksum Check Disable (TX_PTP_UDP_CHKSUM_DIS)** When this bit is cleared, egress times are not saved if the frame has an invalid UDP checksum. When this bit is set, the UDP checksum check is bypassed and the egress time is saved regardless. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b
20	**TX PTP FCS Check Disable (TX_PTP_FCS_DIS)** When this bit is cleared, egress times are not saved if the frame has an invalid FCS. When this bit is set, the FCS check is bypassed. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b

Bits	Description	Type	Default
19:16	**TX PTP Version (TX_PTP_VERSION[3:0])** This field specifies the PTP version in use. A setting of 0 allows any PTP version. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	2h
15:0	**TX PTP Message Type Enable (TX_PTP_MESSAGE_EN[15:0])** These bits individually enable timestamping of their respective message types. Bit 0 of this field corresponds to a message type value of 0 (Sync), bit 1 to message type value 1 (Delay_Req), etc. Typically Sync, Delay_Req, PDelay_Req and PDelay_Resp messages are enabled	R/W	0000h

14.8.37 1588 PORT TX MODIFICATION REGISTER (1588_TX_MOD)

Offset: 164h Size: 32 bits
Bank: 2

This register is used to configure TX PTP message modifications.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29	**TX PTP Pdelay_Resp Message Turnaround Time Insertion (TX_PTP_PDRESP_TA_INSERT)** **Note:** This bit enables the turnaround time between the received Pdelay_Req and the transmitted Pdelay_Resp to be inserted into the correction field of Pdelay_Resp messages sent by the Host.	R/W	0b
28	**TX PTP Sync Message Egress Time Insertion (TX_PTP_SYNC_TS_INSERT)** This bit enables the egress time to be inserted into the originTimestamp field of Sync messages sent by the Host.	R/W	0b
27:22	**RESERVED**	RO	-
21:16	**TX PTP 1 Reserved Byte Offset (TX_PTP_1_RSVD_OFFSET[5:0])** This field specifies the offset into the PTP header. **Note:** The host S/W must not change this field while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	000101b
15:0	**RESERVED**	RO	-

14.8.38 1588 PORT TX MODIFICATION REGISTER 2 (1588_TX_MOD2)

Offset:	168h	Size:	32 bits
Bank:	2		

This register is used to configure TX PTP message modifications.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:1	**RESERVED**	RO	-
0	**TX PTP Clear UDP/IPv4 Checksum Enable (TX_PTP_CLR_UDPV4_CHKSUM)** This bit enables the clearing of the UDP/IPv4 checksum when Pdelay_Resp Message Turnaround Time Insertion or Sync Message Egress Time Insertion is enabled. **Note:** The host S/W must not change this bit while the 1588 Enable (1588_ENABLE) bit in 1588 Command and Control Register (1588_CMD_CTL) is set.	R/W	0b

14.8.39 1588 PORT TX EGRESS TIME SECONDS REGISTER (1588_TX_EGRESS_SEC)

Offset:	16Ch	Size:	32 bits
Bank:	2		

This read only register combined with the 1588 Port TX Egress Time NanoSeconds Register (1588_TX_EGRESS_NS) contains the TX timestamp captures. Up to four captures are buffered.

Note: Values are only valid if the 1588 TX Timestamp Count (1588_TX_TS_CNT[2:0]) field indicates that at least one timestamp is available.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:0	**Timestamp Seconds (TS_SEC)** This field contains the seconds portion of the transmit egress time.	RO	00000000h

14.8.40 1588 PORT TX EGRESS TIME NANOSECONDS REGISTER (1588_TX_EGRESS_NS)

Offset:	170h	Size:	32 bits
Bank:	2		

This read only register combined with the 1588 Port TX Egress Time Seconds Register (1588_TX_EGRESS_SEC) contains the TX timestamp capture. Up to four captures are buffered.

Note: Values are only valid if the 1588 TX Timestamp Count (1588_TX_TS_CNT[2:0]) field indicates that at least one timestamp is available.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29:0	**Timestamp NanoSeconds (TS_NS)** This field contains the nanoseconds portion of the transmit egress time.	RO	00000000h

14.8.41 1588 PORT TX MESSAGE HEADER REGISTER (1588_TX_MSG_HEADER)

Offset:	174h	Size:	32 bits
Bank:	2		

This read only register contains the TX message header. Up to four captures are buffered.

Note: Values are only valid if the 1588 TX Timestamp Count (1588_TX_TS_CNT[2:0]) field indicates that at least one timestamp is available.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:20	**Source Port Identity CRC (SRC_PRT_CRC)** This field contains the 12-bit CRC of the sourcePortIdentity field of the transmitted PTP packet.	RO	000h
19:16	**Message Type (MSG_TYPE)** This field contains the messageType field of the transmitted PTP packet.	RO	0h
15:0	**Sequence ID (SEQ_ID)** This field contains the sequenceId field of the transmitted PTP packet.	RO	0000h

14.8.42 1588 PORT TX DELAY_REQ EGRESS TIME SECONDS REGISTER (1588_TX_DREQ_SEC)

Offset:	178h	Size:	32 bits
Bank:	2		

This register combined with the 1588 Port TX Delay_Req Egress Time NanoSeconds Register (1588_TX_DREQ_NS) contains the egress time of the last Delay_Req message. The contents of this field are normally only used to insert the egress time into received Delay_Resp messages.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

BITS	DESCRIPTION	TYPE	DEFAULT
31:4	**RESERVED**	RO	-
3:0	**Timestamp Seconds (TS_SEC)** This field contains the seconds portion of the transmit egress time.	RO	0h

14.8.43 1588 PORT TX DELAY_REQ EGRESS TIME NANOSECONDS REGISTER (1588_TX_DREQ_NS)

Offset:	17Ch	Size:	32 bits
Bank:	2		

This register combined with the 1588 Port TX Delay_Req Egress Time Seconds Register (1588_TX_DREQ_SEC) contains the egress time of the last Delay_Req message. The contents of this field are normally only used to insert the egress time into received Delay_Resp messages.

Note: Port and GPIO registers share a common address space. Port registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29:0	**Timestamp NanoSeconds (TS_NS)** This field contains the nanoseconds portion of the transmit egress time.	RO	00000000h

14.8.44 1588 TX ONE-STEP SYNC UPPER SECONDS REGISTER (1588_TX_ONE_STEP_SYNC_SEC)

Offset: 180h Size: 32 bits
Bank: 2

This register contains the highest 16 bits of the originTimestamp which is inserted into Sync messages when one-step timestamp insertion is enabled.

Note: This is a static field that is maintained by the Host. It is not incremented when the lower 32 bits of the 1588 Clock rollover.

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:0	**Clock Seconds High (1588_CLOCK_SEC_HI)** This field contains the highest 16 bits of seconds of the 1588 Clock.	R/W	0000h

14.8.45 1588 GPIO CAPTURE CONFIGURATION REGISTER (1588_GPIO_CAP_CONFIG)

Offset: 15Ch Size: 32 bits
Bank: 3

Note: Port and GPIO registers share a common address space. GPIO registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL).

Note: The IEEE 1588 Unit supports 3 GPIO signals.

Bits	Description	Type	Default
31:27	**RESERVED**	RO	-
26:24	**Lock Enable GPIO Falling Edge (LOCK_GPIO_FE)** These bits enable/disables the GPIO falling edge lock. This lock prevents a 1588 capture from overwriting the Clock value if the GPIO interrupt in the 1588 Interrupt Status Register (1588_INT_STS) is already set due to a previous capture. **0: Disables GPIO falling edge lock** **1: Enables GPIO falling edge lock**	R/W	111b
23:19	**RESERVED**	RO	-
18:16	**Lock Enable GPIO Rising Edge (LOCK_GPIO_RE)** These bits enable/disables the GPIO rising edge lock. This lock prevents a 1588 capture from overwriting the Clock value if the GPIO interrupt in the 1588 Interrupt Status Register (1588_INT_STS) is already set due to a previous capture. **0: Disables GPIO rising edge lock** **1: Enables GPIO rising edge lock**	R/W	111b
15:11	**RESERVED**	RO	-
10:8	**GPIO Falling Edge Capture Enable 2-0 (GPIO_FE_CAPTURE_ENABLE[2:0])** These bits enable the falling edge of the respective GPIO input to capture the 1588 clock value and to set the respective 1588_GPIO interrupt in the 1588 Interrupt Status Register (1588_INT_STS). **0: Disables GPIO Capture** **1: Enables GPIO Capture** **Note:** The GPIO must be configured as an input for this function to operate. GPIO inputs are edge sensitive and must be low for greater than 40 ns to be recognized.	R/W	000b
7:3	**RESERVED**	RO	-

Bits	Description	Type	Default
2:0	**GPIO Rising Edge Capture Enable 2-0 (GPIO_RE_CAPTURE_ENABLE[2:0])** These bits enable the rising edge of the respective GPIO input to capture the 1588 clock value and to set the respective 1588_GPIO interrupt in the 1588 Interrupt Status Register (1588_INT_STS). **0: Disables GPIO Capture** **1: Enables GPIO Capture** **Note:** The GPIO must be configured as an input for this function to operate. GPIO inputs are edge sensitive and must be high for greater than 40 ns to be recognized.	R/W	000b

14.8.46 1588 GPIO X RISING EDGE CLOCK SECONDS CAPTURE REGISTER (1588_GPIO_RE_CLOCK_SEC_CAP_X)

Offset:	16Ch	Size:	32 bits
Bank:	3		

This read only register combined with the 1588 GPIO x Rising Edge Clock NanoSeconds Capture Register (1588_GPIO_RE_CLOCK_NS_CAP_x) forms the GPIO rising edge timestamp capture.

Note: Values are only valid if the appropriate 1588 GPIO Rising Edge Interrupt (1588_GPIO_RE_INT[7:0]) in the 1588 Interrupt Status Register (1588_INT_STS) indicates that a timestamp is available.

Note: Unless the corresponding Lock Enable GPIO Rising Edge (LOCK_GPIO_RE) bit is set, a new capture may occur between reads of this register and the 1588 GPIO x Rising Edge Clock NanoSeconds Capture Register (1588_GPIO_RE_CLOCK_NS_CAP_x). Software techniques are required to avoid reading intermediate values.

Note: Port and GPIO registers share a common address space. GPIO registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL). The GPIO accessed (“x”) is set by the GPIO Select (GPIO_SEL[2:0]) field.

Note: All 8 GPIO register sets are available.

Bits	Description	Type	Default
31:0	**Timestamp Seconds (TS_SEC)** This field contains the seconds portion of the timestamp upon the rising edge of a GPIO or upon a software commanded manual capture.	RO	00000000h

14.8.47 1588 GPIO X RISING EDGE CLOCK NANOSECONDS CAPTURE REGISTER (1588_GPIO_RE_CLOCK_NS_CAP_X)

Offset:	170h	Size:	32 bits
Bank:	3		

This read only register combined with the 1588 GPIO x Rising Edge Clock Seconds Capture Register (1588_GPIO_RE_CLOCK_SEC_CAP_x) forms the GPIO rising edge timestamp capture.

Note: Values are only valid if the appropriate 1588 GPIO Rising Edge Interrupt (1588_GPIO_RE_INT[7:0]) in the 1588 Interrupt Status Register (1588_INT_STS) indicates that a timestamp is available.

Note: Unless the corresponding Lock Enable GPIO Rising Edge (LOCK_GPIO_RE) bit is set, a new capture may occur between reads of this register and the 1588 GPIO x Rising Edge Clock Seconds Capture Register (1588_GPIO_RE_CLOCK_SEC_CAP_x). Software techniques are required to avoid reading intermediate values.

Note: Port and GPIO registers share a common address space. GPIO registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL). The GPIO accessed (“x”) is set by the GPIO Select (GPIO_SEL[2:0]) field.

Note: All 8 GPIO register sets are available.

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29:0	**Timestamp NanoSeconds (TS_NS)** This field contains the nanoseconds portion of the timestamp upon the rising edge of a GPIO or upon a software commanded manual capture.	RO	00000000h

14.8.48 1588 GPIO X FALLING EDGE CLOCK SECONDS CAPTURE REGISTER (1588_GPIO_FE_CLOCK_SEC_CAP_X)

Offset:	178h	Size:	32 bits
Bank:	3		

This read only register combined with the 1588 GPIO x Falling Edge Clock NanoSeconds Capture Register (1588_GPIO_FE_CLOCK_NS_CAP_x) forms the GPIO falling edge timestamp capture.

Note: Values are only valid if the appropriate 1588 GPIO Falling Edge Interrupt (1588_GPIO_FE_INT[7:0]) in the 1588 Interrupt Status Register (1588_INT_STS) indicates that a timestamp is available.

Note: Unless the corresponding Lock Enable GPIO Falling Edge (LOCK_GPIO_FE) bit is set, a new capture may occur between reads of this register and the 1588 GPIO x Falling Edge Clock NanoSeconds Capture Register (1588_GPIO_FE_CLOCK_NS_CAP_x). Software techniques are required to avoid reading intermediate values.

Note: Port and GPIO registers share a common address space. GPIO registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL). The GPIO accessed (“x”) is set by the GPIO Select (GPIO_SEL[2:0]) field.

Note: All 8 GPIO register sets are available.

Bits	Description	Type	Default
31:0	**Timestamp Seconds (TS_SEC)** This field contains the seconds portion of the timestamp upon the falling edge of a GPIO or upon a software commanded manual capture.	RO	00000000h

14.8.49 1588 GPIO X FALLING EDGE CLOCK NANOSECONDS CAPTURE REGISTER (1588_GPIO_FE_CLOCK_NS_CAP_X)

Offset:	17Ch	Size:	32 bits
Bank:	3		

This read only register combined with the 1588 GPIO x Falling Edge Clock Seconds Capture Register (1588_GPIO_FE_CLOCK_SEC_CAP_x) forms the GPIO falling edge timestamp capture.

Note: Values are only valid if the appropriate 1588 GPIO Falling Edge Interrupt (1588_GPIO_FE_INT[7:0]) in the 1588 Interrupt Status Register (1588_INT_STS) indicates that a timestamp is available.

Note: Unless the corresponding Lock Enable GPIO Falling Edge (LOCK_GPIO_FE) bit is set, a new capture may occur between reads of this register and the 1588 GPIO x Falling Edge Clock Seconds Capture Register (1588_GPIO_FE_CLOCK_SEC_CAP_x). Software techniques are required to avoid reading intermediate values.

Note: Port and GPIO registers share a common address space. GPIO registers are selected by the Bank Select (BANK_SEL[2:0] in the 1588 Bank Port GPIO Select Register (1588_BANK_PORT_GPIO_SEL). The GPIO accessed (“x”) is set by the GPIO Select (GPIO_SEL[2:0]) field.

Note: All 8 GPIO register sets are available.

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29:0	**Timestamp NanoSeconds (TS_NS)** This field contains the nanoseconds portion of the timestamp upon the falling edge of a GPIO or upon a software commanded manual capture.	RO	00000000h

15.0 GENERAL PURPOSE TIMER & FREE-RUNNING CLOCK

This chapter details the General Purpose Timer (GPT) and the Free-Running Clock.

15.1 General Purpose Timer

The device provides a 16-bit programmable General Purpose Timer that can be used to generate periodic system interrupts. The resolution of this timer is 100 µs.

The GPT loads the General Purpose Timer Count Register (GPT_CNT) with the value in the General Purpose Timer Pre-Load (GPT_LOAD) field of the General Purpose Timer Configuration Register (GPT_CFG) when the General Purpose Timer Enable (TIMER_EN) bit of the General Purpose Timer Configuration Register (GPT_CFG) is asserted (1). On a chip-level reset or when the General Purpose Timer Enable (TIMER_EN) bit changes from asserted (1) to de-asserted (0), the General Purpose Timer Pre-Load (GPT_LOAD) field is initialized to FFFFh. The General Purpose Timer Count Register (GPT_CNT) is also initialized to FFFFh on reset.

Once enabled, the GPT counts down until it reaches 0000h. At 0000h, the counter wraps around to FFFFh, asserts the GP Timer (GPT_INT) interrupt status bit in the Interrupt Status Register (INT_STS), asserts the IRQ interrupt (if GP Timer Interrupt Enable (GPT_INT_EN) is set in the Interrupt Enable Register (INT_EN)) and continues counting. GP Timer (GPT_INT) is a sticky bit. Once this bit is asserted, it can only be cleared by writing a 1 to the bit. Refer to Section 8.2.6, "General Purpose Timer Interrupt," on page 65 for additional information on the GPT interrupt.

Software can write a pre-load value into the General Purpose Timer Pre-Load (GPT_LOAD) field at any time (e.g., before or after the General Purpose Timer Enable (TIMER_EN) bit is asserted). The General Purpose Timer Count Register (GPT_CNT) will immediately be set to the new value and continue to count down (if enabled) from that value.

15.2 Free-Running Clock

The Free-Running Clock (FRC) is a simple 32-bit up-counter that operates from a fixed 25 MHz clock. The current FRC value can be read via the Free Running 25MHz Counter Register (FREE_RUN). On assertion of a chip-level reset, this counter is cleared to zero. On de-assertion of a reset, the counter is incremented once for every 25 MHz clock cycle. When the maximum count has been reached, the counter rolls over to zeros. The FRC does not generate interrupts.

Note: The free running counter can take up to 160 ns to clear after a reset event.

15.3 General Purpose Timer and Free-Running Clock Registers

This section details the directly addressable general purpose timer and free-running clock related System CSRs. For an overview of the entire directly addressable register map, refer to Section 5.0, "Register Map," on page 29.

TABLE 15-1: MISCELLANEOUS REGISTERS

ADDRESS	Register Name (SYMBOL)
08Ch	General Purpose Timer Configuration Register (GPT_CFG)
090h	General Purpose Timer Count Register (GPT_CNT)
09Ch	Free Running 25MHz Counter Register (FREE_RUN)

15.3.1 GENERAL PURPOSE TIMER CONFIGURATION REGISTER (GPT_CFG)

Offset: 08Ch Size: 32 bits

This read/write register configures the device's General Purpose Timer (GPT). The GPT can be configured to generate host interrupts at the interval defined in this register. The current value of the GPT can be monitored via the General Purpose Timer Count Register (GPT_CNT). Refer to Section 15.1, "General Purpose Timer," on page 380 for additional information.

Bits	Description	Type	Default
31:30	**RESERVED**	RO	-
29	**General Purpose Timer Enable (TIMER_EN)** This bit enables the GPT. When set, the GPT enters the run state. When cleared, the GPT is halted. On the 1 to 0 transition of this bit, the GPT_LOAD field of this register will be preset to FFFFh. **0: GPT Disabled** **1: GPT Enabled**	R/W	0b
28:16	**RESERVED**	RO	-
15:0	**General Purpose Timer Pre-Load (GPT_LOAD)** This value is pre-loaded into the GPT. This is the starting value of the GPT. The timer will begin decrementing from this value when enabled.	R/W	FFFFh

15.3.2 GENERAL PURPOSE TIMER COUNT REGISTER (GPT_CNT)

Offset: 090h Size: 32 bits

This read-only register reflects the current general purpose timer (GPT) value. The register should be used in conjunction with the General Purpose Timer Configuration Register (GPT_CFG) to configure and monitor the GPT. Refer to Section 15.1, "General Purpose Timer," on page 380 for additional information.

Bits	Description	Type	Default
31:16	**RESERVED**	RO	-
15:0	**General Purpose Timer Current Count (GPT_CNT)** This 16-bit field represents the current value of the GPT.	RO	FFFFh

15.3.3 FREE RUNNING 25MHZ COUNTER REGISTER (FREE_RUN)

Offset: 09Ch Size: 32 bits

This read-only register reflects the current value of the free-running 25MHz counter. Refer to Section 15.2, "Free-Running Clock," on page 380 for additional information.

Bits	Description	Type	Default
31:0	**Free Running Counter (FR_CNT)** This field reflects the current value of the free-running 32-bit counter. At reset, the counter starts at zero and is incremented by one every 25 MHz cycle. When the maximum count has been reached, the counter will rollover to zero and continue counting. **Note:** The free running counter can take up to 160nS to clear after a reset event.	RO	00000000h

16.0 GPIO/LED CONTROLLER

16.1 Functional Overview

The GPIO/LED Controller provides 3 configurable general purpose input/output pins, **GPIO[2:0]**. These pins can be individually configured to function as inputs, push-pull outputs or open drain outputs and each is capable of interrupt generation with configurable polarity. Alternatively, all 3 GPIO pins can be configured as LED outputs, enabling these pins to drive Ethernet status LEDs for external indication of various attributes of the port. All GPIOs also provide extended 1588 functionality. Refer to Section 14.5, "1588 GPIOs," on page 314 for additional details.

GPIO and LED functionality is configured via the GPIO/LED System Control and Status Registers (CSRs). These registers are defined in Section 16.4, "GPIO/LED Registers," on page 386.

16.2 GPIO Operation

The GPIO controller is comprised of 3 programmable input/output pins. These pins are individually configurable via the GPIO CSRs. On application of a chip-level reset:

- All GPIOs are set as inputs (GPIO Direction 2-0 (GPIODIR[2:0]) cleared in General Purpose I/O Data & Direction Register (GPIO_DATA_DIR))
- All GPIO interrupts are disabled (GPIO Interrupt Enable[2:0] (GPIO[2:0]_INT_EN) cleared in General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN)
- All GPIO interrupts are configured to low logic level triggering (GPIO Interrupt/1588 Polarity 2-0 (GPIO_POL[2:0]) cleared in General Purpose I/O Configuration Register (GPIO_CFG))

Note: **GPIO[2:0]** may be configured as LED outputs by default, dependent on the LED_en_strap[2:0] configuration straps. Refer to Section 16.3, "LED Operation" for additional information.

The direction and buffer type of all GPIOs are configured via the General Purpose I/O Configuration Register (GPIO_CFG) and General Purpose I/O Data & Direction Register (GPIO_DATA_DIR). The direction of each GPIO, input or output, should be configured first via its respective GPIO Direction 2-0 (GPIODIR[2:0]) bit in the General Purpose I/O Data & Direction Register (GPIO_DATA_DIR). When configured as an output, the output buffer type for each GPIO is selected by the GPIO Buffer Type 2-0 (GPIOBUF[2:0]) bits in the General Purpose I/O Configuration Register (GPIO_CFG). Push/pull and open-drain output buffers are supported for each GPIO. When functioning as an open-drain driver, the GPIO output pin is driven low when the corresponding GPIO Data 2-0 (GPIOD[2:0]) bit in the General Purpose I/O Data & Direction Register (GPIO_DATA_DIR) is cleared to 0 and is not driven when set to 1.

When a GPIO is enabled as a push/pull output, the value output to the GPIO pin is set via the corresponding GPIO Data 2-0 (GPIOD[2:0]) bit in the General Purpose I/O Data & Direction Register (GPIO_DATA_DIR). For GPIOs configured as inputs, the corresponding GPIO Data 2-0 (GPIOD[2:0]) bit reflects the current state of the GPIO input.

In GPIO mode, the input buffers are disabled when the pin is set to an output and the pull-ups are normally enabled.

Note: Upon reset, GPIOs that were outputs may generate an active interrupt status as the system settles - typically when a low GPIO pin slowly rises due to the internal pull-up. The interrupt status bits within the General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN) should be cleared as part of the device initialization software routine.

16.2.1 GPIO INTERRUPTS

Each GPIO provides the ability to trigger a unique GPIO interrupt in the General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN). Reading the GPIO Interrupt[2:0] (GPIO[2:0]_INT) bits of this register provides the current status of the corresponding interrupt and each interrupt is enabled by setting the corresponding GPIO Interrupt Enable[2:0] (GPIO[2:0]_INT_EN) bit. The GPIO/LED Controller aggregates the enabled interrupt values into an internal signal that is sent to the System Interrupt Controller and is reflected via the Interrupt Status Register (INT_STS) GPIO Interrupt Event (GPIO) bit. For more information on interrupts, refer to Section 8.0, "System Interrupts," on page 62.

As interrupts, GPIO inputs are level sensitive and must be active for greater than 40 ns to be recognized.

16.2.1.1 GPIO Interrupt Polarity

The interrupt polarity can be set for each individual GPIO via the GPIO Interrupt/1588 Polarity 2-0 (GPIO_POL[2:0]) bits in the General Purpose I/O Configuration Register (GPIO_CFG). When set, a high logic level on the GPIO pin will set the corresponding interrupt bit in the General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN). When cleared, a low logic level on the GPIO pin will set the corresponding interrupt bit.

16.3 LED Operation

GPIO[2:0] can be individually selected to function as a LED. These pins are configured as LED outputs by setting the corresponding LED Enable 2-0 (LED_EN[2:0]) bit in the LED Configuration Register (LED_CFG). When configured as an LED, the pin is either a push-pull or open-drain / open-source output and the GPIO related input buffer and pull-up are disabled. The default configuration, including polarity, is determined by input straps or EEPROM entries. Refer to Section 7.0, "Configuration Straps," on page 54 for additional information.

The functions associated with each LED pin are configurable via the LED Function 2-0 (LED_FUN[2:0]) bits of the LED Configuration Register (LED_CFG). These bits allow the configuration of each LED pin to indicate various port related functions. The behaviors of each LED for each LED Function 2-0 (LED_FUN[2:0]) configuration are described in the following tables. Detailed definitions for each LED indication type are provided in Section 16.3.1.

The default values of the LED Function 2-0 (LED_FUN[2:0]) and LED Enable 2-0 (LED_EN[2:0]) bits of the LED Configuration Register (LED_CFG) are determined by the LED_fun_strap[2:0] and LED_en_strap[2:0] configuration straps. For more information on the LED Configuration Register (LED_CFG) and its related straps, refer to Section 16.4.1, "LED Configuration Register (LED_CFG)," on page 387.

All LED outputs may be disabled by setting the LED Disable (LED_DIS) bit in the Power Management Control Register (PMT_CTRL). Open-drain / open-source LEDs are un-driven. Push-pull LEDs are still driven but are set to their inactive state.

TABLE 16-1: LED OPERATION AS A FUNCTION OF LED_FUN[2:0] = 000B - 011B

	000b	001b	010b	011b
LED2 (GPIO2)	Link / Activity	100Link / Activity	Activity	Activity
LED1 (GPIO1)	Full-duplex / Collision	Full-duplex / Collision	Link	Link
LED0 (GPIO0)	Speed	10Link / Activity	Speed	Full-duplex / Collision

TABLE 16-2: LED OPERATION AS A FUNCTION OF LED_FUN[2:0] = 100B - 111B

	100b	101b	110b	111b
LED2 (GPIO2)	Activity	Reserved	Reserved	Reserved
LED1 (GPIO1)	10Link			
LED0 (GPIO0)	100Link			

The various LED indication functions listed in the previous tables are described in the following section.

16.3.1 LED FUNCTION DEFINITIONS

The following LED rules apply :

- “Active” is defined as the pin being driven to the opposite value latched at reset on the related hard-straps. The LED polarity cannot be modified via soft-straps.
- “Inactive” is defined as the pin not being driven.
- The input buffers and pull-ups are disabled on the shared GPIO/LED pins.

The following LED function definitions apply :

- **Activity -** The signal is pulsed active for 80mS to indicate transmit or receive activity on the port. The signal is

then made inactive for a minimum of 80mS, after which the process will repeat if RX or TX activity is again detected.

Note: The idle condition is *inactive* in contrast to that of the Link / Activity function.

Note: The signal will be held inactive if the PHY does not have a valid link.

- **Link** - A steady active output indicates that the port has a valid link (10Mbps or 100Mbps), while a steady inactive output indicates no link on the port.
- **Link / Activity** - A steady active output indicates that the port has a valid link, while a steady inactive output indicates no link on the port. When the port has a valid link, the signal is pulsed inactive for 80 ms to indicate transmit or receive activity on the port. The signal is then made active for a minimum of 80 ms, after which the process will repeat if RX or TX activity is again detected.
- **100Link** - A steady active output indicates the port has a valid link and the speed is 100 Mbps. The signal will be held inactive if the port does not have a valid link or the speed is not 100 Mbps.
- **100Link / Activity** - A steady active output indicates the port has a valid link and the speed is 100 Mbps. The signal is pulsed inactive for 80 ms to indicate TX or RX activity on the port. The signal is then driven active for a minimum of 80 ms, after which the process will repeat if RX or TX activity is again detected. The signal will be held inactive if the port does not have a valid link or the speed is not 100 Mbps.
- **10Link** - A steady active output indicates the port has a valid link and the speed is 10 Mbps. This signal will be held inactive if the port does not have a valid link or the speed is not 10 Mbps.
- **10Link / Activity** - A steady active output indicates the port has a valid link and the speed is 10 Mbps. The signal is pulsed inactive for 80 ms to indicate transmit or receive activity on the port. The signal is then driven active for a minimum of 80 ms, after which the process will repeat if RX or TX activity is again detected. This signal will be held inactive if the port does not have a valid link or the speed is not 10 Mbps.
- **Full-duplex / Collision** - A steady active output indicates the port is in full-duplex mode. In half-duplex mode, the signal is pulsed active for 80 ms to indicate a network collision. The signal is then made inactive for a minimum of 80 ms, after which the process will repeat if another collision is detected. The signal will be held inactive if the port does not have a valid link.
- **Speed** - A steady active output indicates a valid link with a speed of 100 Mbps. A steady inactive output indicates a speed of 10 Mbps. The signal will be held inactive if the port does not have a valid link.

16.4 GPIO/LED Registers

This section details the directly addressable General Purpose I/O (GPIO) and LED related System CSRs. For an overview of the entire directly addressable register map, refer to Section 5.0, "Register Map," on page 29.

TABLE 16-3: GPIO/LED REGISTERS

ADDRESS	REGISTER NAME (SYMBOL)
1BCh	LED Configuration Register (LED_CFG)
1E0h	General Purpose I/O Configuration Register (GPIO_CFG)
1E4h	General Purpose I/O Data & Direction Register (GPIO_DATA_DIR)
1E8h	General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN)

16.4.1 LED CONFIGURATION REGISTER (LED_CFG)

Offset: 1BCh Size: 32 bits

This read/write register configures the **GPIO[2:0]** pins as LED pins and sets their functionality.

Bits	Description	Type	Default
31:11	**RESERVED**	RO	-
10:8	**LED Function 2-0 (LED_FUN[2:0])** These bits control the function associated with each LED pin as shown in Section 16.3, "LED Operation," on page 385. **Note:** In order for these assignments to be valid, the particular pin must be enabled as an LED output pin via the LED_EN bits of this register.	R/W	Note 1
7:3	**RESERVED**	RO	-
2:0	**LED Enable 2-0 (LED_EN[2:0])** This field toggles the functionality of the **GPIO[2:0]** pins between GPIO and LED. **0: Enables the associated pin as a GPIO signal** **1: Enables the associated pin as a LED output** When configured as LED outputs, the pins are open-drain/open-source outputs and the pull-ups and input buffers are disabled. When open-drain/open-source, the polarity of the pins depends upon the strap value sampled at reset. If a high is sampled at reset, then this signal is active low. **Note:** The polarity is determined by the strap value sampled on reset (a hard-strap) and not the soft-strap value (of the shared strap) set via EEPROM. When configured as a GPIO output, the pins are configured per the General Purpose I/O Configuration Register (GPIO_CFG) and the General Purpose I/O Data & Direction Register (GPIO_DATA_DIR). The polarity of the pins does not depend upon the strap value sampled at reset.	R/W	Note 2

Note 1: The default value of this field is determined by the configuration strap LED_fun_strap[2:0].

Note 2: The default value of this field is determined by the configuration strap LED_en_strap[2:0].

16.4.2 GENERAL PURPOSE I/O CONFIGURATION REGISTER (GPIO_CFG)

Offset: 1E0h Size: 32 bits

This read/write register configures the GPIO input and output pins. The polarity of the GPIO pins is configured here as well as the IEEE 1588 timestamping and clock compare event output properties. Refer to Section 14.5, "1588 GPIOs," on page 314 for additional 1588 information.

Bits	Description	Type	Default
31:27	**RESERVED**	RO	-
26:24	**1588 GPIO Channel Select 2-0 (GPIO_CH_SEL[2:0])** These bits select the 1588 channel to be output on the corresponding GPIO[2:0]. Refer to Section 14.5, "1588 GPIOs," on page 314 for additional information. **0: Sets 1588 channel A as the output for the corresponding GPIO pin** **1: Sets 1588 channel B as the output for the corresponding GPIO pin**	R/W	000b
23:19	**RESERVED**	RO	-
18:16	**GPIO Interrupt/1588 Polarity 2-0 (GPIO_POL[2:0])** These bits set the interrupt input polarity and 1588 clock event output polarity of the 3 GPIO pins. The configured level (high/low) will set the corresponding GPIO_INT bit in the General Purpose I/O Interrupt Status and Enable Register (GPIO_INT_STS_EN). 1588 clock events will be output active at the configured level (high/low). These bits also determine the polarity of the GPIO 1588 Timer Interrupt Clear inputs. Refer to Section 14.5, "1588 GPIOs," on page 314 for additional information. **0: Sets low logic level trigger on corresponding GPIO pin** **1: Sets high logic level trigger on corresponding GPIO pin**	R/W	000b
15:11	**RESERVED**	RO	-
10:8	**1588 GPIO Output Enable 2-0 (1588_GPIO_OE[2:0])** These bits configure the 3 GPIO pins to output 1588 clock compare events. **0: Disables the output of 1588 clock compare events** **1: Enables the output of 1588 clock compare events** **Note:** These bits override the direction bits in the General Purpose I/O Data & Direction Register (GPIO_DATA_DIR) register. However, the GPIO Buffer Type 2-0 (GPIOBUF[2:0]) in the General Purpose I/O Configuration Register (GPIO_CFG) is not overridden.	R/W	000b
7:3	**RESERVED**	RO	-

<table>
<tr><th>Bits</th><th>Description</th><th>Type</th><th>Default</th></tr>
<tr><td>2:0</td><td>GPIO Buffer Type 2-0 (GPIOBUF[2:0])
This field sets the buffer types of the 3 GPIO pins.

0: Corresponding GPIO pin configured as an open-drain driver
1: Corresponding GPIO pin configured as a push/pull driver

As an open-drain driver, the output pin is driven low when the corresponding data register is cleared, and is not driven when the corresponding data register is set.

As an open-drain driver used for 1588 Clock Events, the corresponding GPIO_POL_x bit determines when the corresponding pin is driven per the following table:

<table>
<tr><th>GPIOx Clock Event Polarity</th><th>1588 Clock Event</th><th>Pin State</th></tr>
<tr><td>0</td><td>no</td><td>not driven</td></tr>
<tr><td>0</td><td>yes</td><td>driven low</td></tr>
<tr><td>1</td><td>no</td><td>driven low</td></tr>
<tr><td>1</td><td>yes</td><td>not driven</td></tr>
</table>
</td><td>R/W</td><td>000b</td></tr>
</table>

16.4.3 GENERAL PURPOSE I/O DATA & DIRECTION REGISTER (GPIO_DATA_DIR)

Offset: 1E4h Size: 32 bits

This read/write register configures the direction of the GPIO pins and contains the GPIO input and output data bits.

Bits	Description	Type	Default
31:19	**RESERVED**	RO	-
18:16	**GPIO Direction 2-0 (GPIODIR[2:0])** These bits set the input/output direction of the 3 GPIO pins. **0: GPIO pin is configured as an input** **1: GPIO pin is configured as an output**	R/W	000b
15:3	**RESERVED**	RO	-
2:0	**GPIO Data 2-0 (GPIOD[2:0])** When a GPIO pin is enabled as an output, the value written to this field is output on the corresponding GPIO pin. Upon a read, the value returned depends on the current direction of the pin. If the pin is an input, the data reflects the current state of the corresponding GPIO pin. If the pin is an output, the data is the value that was last written into this register. The pin direction is determined by the GPIODIR bits of this register and the 1588_GPIO_OE bits in the General Purpose I/O Configuration Register (GPIO_CFG).	R/W	000b

16.4.4 GENERAL PURPOSE I/O INTERRUPT STATUS AND ENABLE REGISTER (GPIO_INT_STS_EN)

Offset: 1E8h Size: 32 bits

This read/write register contains the GPIO interrupt status bits.

Writing a 1 to any of the interrupt status bits acknowledges and clears the interrupt. If enabled, these interrupt bits are cascaded into the GPIO Interrupt Event (GPIO) bit of the Interrupt Status Register (INT_STS). Writing a 1 to any of the interrupt enable bits will enable the corresponding interrupt as a source. Status bits will still reflect the status of the interrupt source regardless of whether the source is enabled as an interrupt in this register. The GPIO Interrupt Event Enable (GPIO_EN) bit of the Interrupt Enable Register (INT_EN) must also be set in order for an actual system level interrupt to occur. Refer to Section 8.0, "System Interrupts," on page 62 for additional information.

BITS	DESCRIPTION	TYPE	DEFAULT
31:19	**RESERVED**	RO	-
18:16	**GPIO Interrupt Enable[2:0] (GPIO[2:0]_INT_EN)** When set, these bits enable the corresponding GPIO interrupt. **Note:** The GPIO interrupts must also be enabled via the GPIO Interrupt Event Enable (GPIO_EN) bit of the Interrupt Enable Register (INT_EN) in order to cause the interrupt pin (**IRQ**) to be asserted.	R/W	000b
15:3	**RESERVED**	RO	-
2:0	**GPIO Interrupt[2:0] (GPIO[2:0]_INT)** These signals reflect the interrupt status as generated by the GPIOs. These interrupts are configured through the General Purpose I/O Configuration Register (GPIO_CFG). **Note:** As GPIO interrupts, GPIO inputs are level sensitive and must be active greater than 40 ns to be recognized as interrupt inputs.	R/WC	000b

17.0 MISCELLANEOUS

This chapter describes miscellaneous functions and registers that are present in the device.

17.1 Miscellaneous System Configuration & Status Registers

This section details the remainder of the directly addressable System CSRs. These registers allow for monitoring and configuration of various device functions such as the Chip ID/revision, byte order testing, and hardware configuration.

For an overview of the entire directly addressable register map, refer to Section 5.0, "Register Map," on page 29.

TABLE 17-1: MISCELLANEOUS REGISTERS

ADDRESS	Register Name (SYMBOL)
050h	Chip ID and Revision (ID_REV)
064h	Byte Order Test Register (BYTE_TEST)
074h	Hardware Configuration Register (HW_CFG)

17.1.1 CHIP ID AND REVISION (ID_REV)

Offset: 050h Size: 32 bits

This read-only register contains the ID and Revision fields for the device.

Bits	Description	Type	Default
31:16	**Chip ID** This field indicates the chip ID.	RO	9250
15:0	**Chip Revision** This field indicates the design revision.	RO	Note 1

Note 1: Default value is dependent on device revision.

17.1.2 BYTE ORDER TEST REGISTER (BYTE_TEST)

Offset: 064h Size: 32 bits

This read-only register can be used to determine the byte ordering of the current configuration. Byte ordering is a function of the host data bus width and endianess. Refer to Section 9.0, "Host Bus Interface," on page 74 for additional information on byte ordering.

The BYTE_TEST register can optionally be used as a dummy read register when assuring minimum write-to-read or read-to-read timing. Refer to Section 9.0, "Host Bus Interface," on page 74 for additional information.

For host interfaces that are disabled during the reset state, the BYTE_TEST register can be used to determine when the device has exited the reset state.

Note: This register can be read while the device is in the reset or not ready / power savings states without leaving the host interface in an intermediate state. If the host interface is in a reset state, returned data may be invalid. However, during reset, the returned data will not match the normal valid data pattern.

Note: It is not necessary to read all fours BYTEs of this register. DWORD access rules do not apply to this register.

Bits	Description	Type	Default
31:0	**Byte Test (BYTE_TEST)** This field reflects the current byte ordering	RO	87654321h

17.1.3 HARDWARE CONFIGURATION REGISTER (HW_CFG)

Offset: 074h Size: 32 bits

This register allows the configuration of various hardware features including TX/RX FIFO sizes and Host MAC transmit threshold properties. A detailed explanation of the allowable settings for FIFO memory allocation can be found in Section 11.10.3, "FIFO Memory Allocation Configuration," on page 157.

Note: This register can be read while the device is in the reset or not ready / power savings states without leaving the host interface in an intermediate state. If the host interface is in a reset state, returned data may be invalid.

Note: It is not necessary to read all fours BYTEs of this register. DWORD access rules do not apply to this register.

Bits	Description	Type	Default
31:28	**RESERVED**	RO	-
27	**Device Ready (READY)** When set, this bit indicates that the device is ready to be accessed. Upon power-up, RST# reset, return from power savings states, Host MAC module level reset or digital reset, the host processor may interrogate this field as an indication that the device has stabilized and is fully active. This rising edge of this bit will assert the Device Ready (READY) bit in the Interrupt Status Register (INT_STS) and can cause an interrupt if enabled. **Note:** With the exception of the HW_CFG, PMT_CTRL, BYTE_TEST, and RESET_CTL registers, read access to any internal resources is forbidden while the READY bit is cleared. Writes to any address are invalid until this bit is set. **Note:** This bit is identical to bit 0 of the Power Management Control Register (PMT_CTRL).	RO	0b
26	**RESERVED**	RO	-
25	**AMDIX_EN Strap State** This bit reflects the state of the auto_mdix_strap_1 strap that connects to the PHY. The strap value is loaded with the level of the auto_mdix_strap_1 during reset and can be re-written by the EEPROM Loader. The strap value can be overridden by bit 15 and 13 of the PHY Special Control/Status Indication Register (PHY_SPECIAL_CONTROL_STAT_IND).	RO	Note 2
24:22	**RESERVED**	RO	-
21	**RESERVED - This bit must be written with 0b for proper operation.**	R/W	0b
20	**Must Be One (MBO).** This bit must be set to ‘1’ for normal device operation.	R/W	0b

Bits	Description	Type	Default
19:16	**TX FIFO Size (TX_FIF_SZ)** This field sets the size of the TX FIFOs in 1KB values to a maximum of 14KB. The TX Status FIFO consumes 512 bytes of the space allocated by TX_FIF_-SIZ, and the TX Data FIFO consumes the remaining space specified by TX_FIF_SZ. The minimum size of the TX FIFOs is 2KB (TX Data FIFO and Status FIFO combined). The TX Data FIFO is used for both TX data and TX commands. The RX Status and Data FIFOs consume the remaining space, which is equal to 16KB minus TX_FIF_SIZ. See section Section 11.10.3, "FIFO Memory Allocation Configuration," on page 157 for more information.	R/W	5h
15:14	**RESERVED**	RO	-
13:12	**RESERVED - This field must be written with 00b for proper operation.**	R/W	00b
11:0	**RESERVED**	RO	-

Note 2: The default value of this field is determined by the configuration strap auto_mdix_strap_1. See Section 6.3, "Power Management," on page 44 for more information.

18.0 JTAG

18.1 JTAG

A IEEE 1149.1 compliant TAP Controller supports boundary scan and various test modes.

The device includes an integrated JTAG boundary-scan test port for board-level testing. The interface consists of four pins (**TDO**, **TDI**, **TCK** and **TMS**) and includes a state machine, data register array, and an instruction register. The JTAG pins are described in Table 3-9, “JTAG Pin Descriptions,” on page 25. The JTAG interface conforms to the IEEE Standard 1149.1 - 2001 *Standard Test Access Port (TAP) and Boundary-Scan Architecture*.

All input and output data is synchronous to the **TCK** test clock input. TAP input signals **TMS** and **TDI** are clocked into the test logic on the rising edge of **TCK**, while the output signal **TDO** is clocked on the falling edge.

JTAG pins are multiplexed with the GPIO/LED and EEPROM pins. The JTAG functionality is selected when the **TESTMODE** pin is asserted.

The implemented IEEE 1149.1 instructions and their op codes are shown in Table 18-1.

TABLE 18-1: IEEE 1149.1 OP CODES

INSTRUCTION	OP CODE	COMMENT
BYPASS 0	16'h0000	Mandatory Instruction
BYPASS 1	16'hFFFF	Mandatory Instruction
SAMPLE/PRELOAD	16'hFFF8	Mandatory Instruction
EXTEST	16'hFFE8	Mandatory Instruction
CLAMP	16'hFFEF	Optional Instruction
ID_CODE	16'hFFFE	Optional Instruction
HIGHZ	16'hFFCF	Optional Instruction
INT_DR_SEL	16'hFFFD	Private Instruction

Note: The JTAG device ID is 000F1445h

Note: All digital I/O pins support IEEE 1149.1 operation. Analog pins and the **OSCI** / **OSCO** pins do not support IEEE 1149.1 operation.

18.1.1 JTAG TIMING REQUIREMENTS

This section specifies the JTAG timing of the device.

FIGURE 18-1: JTAG TIMING

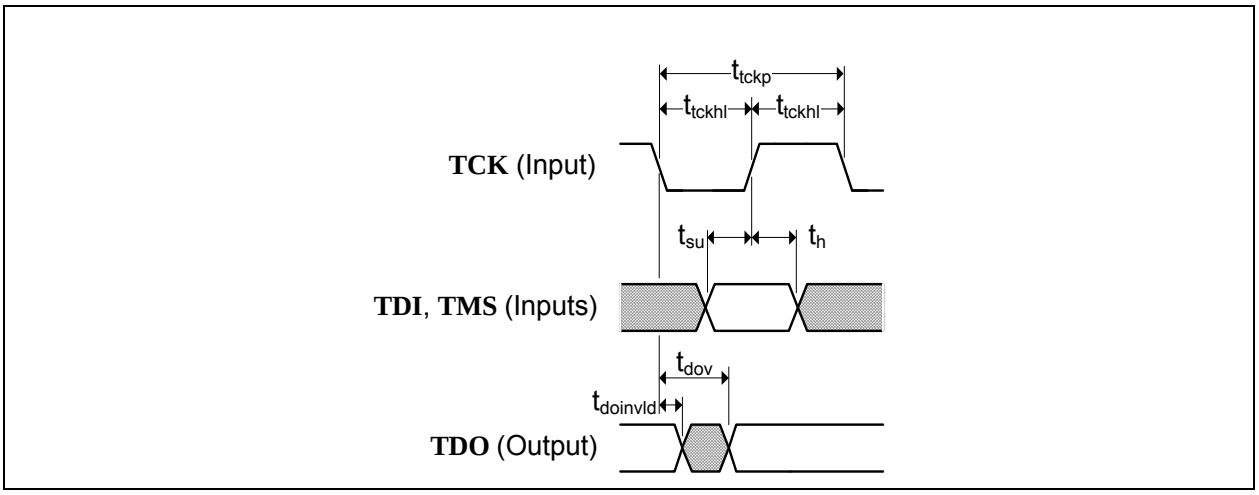


TABLE 18-2: JTAG TIMING VALUES

Symbol	Description	Min	Max	Units	Notes
t_{tckp}	**TCK** clock period	40		ns	
t_{tckhl}	**TCK** clock high/low time	t_{tckp}*0.4	t_{tckp}*0.6	ns	
t_{su}	**TDI**, **TMS** setup to **TCK** rising edge	5		ns	
t_h	**TDI**, **TMS** hold from **TCK** rising edge	5		ns	
t_{dov}	**TDO** output valid from **TCK** falling edge		15	ns	
$t_{doinvld}$	**TDO** output invalid from **TCK** falling edge	0		ns	

Note: Timing values are with respect to an equivalent test load of 25 pF.

19.0 OPERATIONAL CHARACTERISTICS

19.1 Absolute Maximum Ratings*

Supply Voltage (VDD12TX1, VDD12TX2, OSCVDD12, VDDCR) (Note 1) . . . . . 0 V to +1.5 V

Supply Voltage (VDD33TXRX1, VDD33TXRX2, VDD33BIAS, VDD33, VDDIO) (Note 1) . . . . . 0 V to +3.6 V

Ethernet Magnetics Supply Voltage . . . . . -0.5 V to +3.6 V

Positive voltage on input signal pins, with respect to ground (Note 2) . . . . . **VDDIO** + 2.0 V

Negative voltage on input signal pins, with respect to ground (Note 3) . . . . . -0.5 V

Positive voltage on OSCI, with respect to ground . . . . . +3.6 V

Storage Temperature . . . . . -55°C to +150°C

Junction Temperature . . . . . +150°C

Lead Temperature Range . . . . . Refer to JEDEC Spec. J-STD-020

HBM ESD Performance . . . . . JEDEC Class 3A

Note 1: When powering this device from laboratory or system power supplies, it is important that the absolute maximum ratings not be exceeded or device failure can result. Some power supplies exhibit voltage spikes on their outputs when AC power is switched on or off. In addition, voltage transients on the AC power line may appear on the DC output. If this possibility exists, it is suggested to use a clamp circuit.

Note 2: This rating does not apply to the following pins: OSCI, RBIAS

Note 3: This rating does not apply to the following pins: RBIAS

*Stresses exceeding those listed in this section could cause permanent damage to the device. This is a stress rating only. Exposure to absolute maximum rating conditions for extended periods may affect device reliability. Functional operation of the device at any condition exceeding those indicated in Section 19.2, "Operating Conditions**", Section 19.5, "DC Specifications", or any other applicable section of this specification is not implied. Note, device signals are *NOT* 5 volt tolerant.

19.2 Operating Conditions**

Supply Voltage (VDD12TX1, VDD12TX2, OSCVDD12, VDDCR) . . . . . +1.14 V to +1.26 V

Analog Port Supply Voltage (VDD33TXRX1, VDD33TXRX2, VDD33BIAS, VDD33) . . . . . +3.0 V to +3.6 V

I/O Supply Voltage (VDDIO) (Note 1) . . . . . +1.62 V to +3.6 V

Ethernet Magnetics Supply Voltage . . . . . +2.25 V to +3.6 V

Ambient Operating Temperature in Still Air (T_A) . . . . . Note 4

Note 4: 0°C to +70°C for commercial version, -40°C to +85°C for industrial version, -40°C to +105°C for extended industrial version.

Extended industrial temperature range is supported with the following restrictions:
- 64-QFN package: External regulator required (Internal regulator disabled) and 2.5 V (typ) Ethernet magnetics voltage.

Proper operation of the device is guaranteed only within the ranges specified in this section. After the device has completed power-up, **VDDIO and the magnetics power supply must maintain their voltage level with ±10%. Varying the voltage greater than ±10% after the device has completed power-up can cause errors in device operation.

Note: Do not drive input signals without power supplied to the device.

19.3 Package Thermal Specifications

TABLE 19-1: 64-PIN QFN PACKAGE THERMAL PARAMETERS

Parameter	Symbol	Value	Units	Comments
Thermal Resistance Junction to Ambient	Θ_{JA}	23.6	°C/W	Measured in still air
Thermal Resistance Junction to Bottom of Case	Ψ_{JT}	0.1	°C/W	Measured in still air
Thermal Resistance Junction to Top of Case	Θ_{JC}	1.8	°C/W	Airflow 1 m/s

TABLE 19-2: 64-PIN TQFP-EP PACKAGE THERMAL PARAMETERS

Parameter	Symbol	Value	Units	Comments
Thermal Resistance Junction to Ambient	Θ_{JA}	29.0	°C/W	Measured in still air
Thermal Resistance Junction to Bottom of Case	Ψ_{JT}	0.3	°C/W	Measured in still air
Thermal Resistance Junction to Top of Case	Θ_{JC}	12.8	°C/W	Airflow 1 m/s

Note: Thermal parameters are measured or estimated for devices in a multi-layer 2S2P PCB per JESDN51.

TABLE 19-3: MAXIMUM POWER DISSIPATION

Mode	Maximum Power (mW)
Internal Regulator Disabled, 2.5 V Ethernet Magnetics	460
Internal Regulator Disabled, 3.3 V Ethernet Magnetics	550
Internal Regulator Enabled, 2.5 V Ethernet Magnetics	617
Internal Regulator Enabled, 3.3 V Ethernet Magnetics	706

19.4 Current Consumption and Power Consumption

This section details the device's typical supply current consumption and power dissipation for 10BASE-T, 100BASE-TX and power management modes of operation with the internal regulator enabled and disabled.

19.4.1 INTERNAL REGULATOR DISABLED

TABLE 19-4: CURRENT CONSUMPTION AND POWER DISSIPATION (REGS. DISABLED)

		3.3 V Device Current (mA) (A) Note 5, Note 7	1.2 V Device Current (mA) (B) Note 6, Note 7	TX Magnetics Current (mA) (C) Note 8	Device Power with 2.5 V Magnetics (mW) Note 9, Note 10	Device Power with 3.3 V Magnetics (mW) Note 9, Note 11
Reset (RST#)	Typ.	23.2	37.0	0.0	121	121
D0, 100BASE-TX with Traffic (No EEE)	Typ.	44.2	51.7	41.0	311	344
D0, 100BASE-TX Idle (w/o EEE)	Typ.	44.2	51.0	41.0	310	343
D0, 100BASE-TX Idle (with EEE)	Typ.	36.2	46.0	0.0	175	175
D0, 10BASE-T with Traffic	Typ.	21.7	42.8	101.0	376	457
D0, 10BASE-T Idle	Typ.	22.8	43.0	101.0	380	461
D0, PHY Energy Detect Power Down	Typ.	8.9	40.2	0.0	78	78
D0, PHY General Power Down	Typ.	4.9	40.5	0.0	65	65
D1, 100BASE-TX Idle (w/o EEE)	Typ.	38.1	28.6	41.0	263	296
D1, 100BASE-TX Idle (with EEE)	Typ.	36.6	23.3	0.0	149	149
D1, 10BASE-T Idle	Typ.	16.7	19.9	101.0	332	413
D1, PHY Energy Detect Power Down	Typ.	9.0	17.5	0.0	51	51
D1, PHY General Power Down	Typ.	5.8	17.5	0.0	41	41
D2, 100BASE-TX Idle (w/o EEE)	Typ.	38.0	28.7	41.0	263	296
D2, 100BASE-TX Idle (with EEE)	Typ.	36.6	23.4	0.0	149	149
D2, 10BASE-T Idle	Typ.	16.8	19.9	101.0	332	413

TABLE 19-4: CURRENT CONSUMPTION AND POWER DISSIPATION (REGS. DISABLED)

D2, PHY Energy Detect Power Down	Typ.	8.7	6.1	0.0	36	36
D2, PHY General Power Down	Typ.	5.9	6.1	0.0	27	27
D3, PHY General Power Down	Typ.	5.7	2.7	0.0	23	23

Note 5: VDD33TXRX1, VDD33TXRX2, VDD33BIAS, VDD33, VDDIO

Note 6: VDD12TX1, VDD12TX2, OSCVDD12, VDDCR

Note 7: Current measurements do not include power applied to the magnetics or the optional external LEDs.

Note 8: The Ethernet component current is independent of the supply rail voltage (2.5V or 3.3V) of the transformer. Copper TP operation is assumed. Current is zero if using 100BASE-FX mode.

Note 9: This includes the power dissipated by the transmitter by way of the current through the transformer.

Note 10: 3.3*(A) + 1.2*(B) + (2.5)*(C) @ Typ

Note 11: 3.3*(A) + 1.2*(B) + (3.3)*(C) @ Typ

19.4.2 INTERNAL REGULATOR ENABLED

TABLE 19-5: CURRENT CONSUMPTION AND POWER DISSIPATION (REGS. ENABLED)

		3.3 V Device Current (mA) (A) Note 12, Note 13, Note 14	TX Magnetics Current (mA) (C) Note 15	Device Power with 2.5 V Magnetics (mW) Note 16, Note 17	Device Power with 3.3 V Magnetics (mW) Note 16, Note 18
Reset (**RST#**)	Typ.	61.5	0.0	203	203
D0, 100BASE-TX with Traffic (No EEE)	Typ.	95.9	41.0	419	452
D0, 100BASE-TX Idle (w/o EEE)	Typ.	96.4	41.0	421	454
D0, 100BASE-TX Idle (with EEE)	Typ.	82.2	0.0	272	272
D0, 10BASE-T with Traffic	Typ.	67.5	101.0	476	557
D0, 10BASE-T Idle	Typ.	67.7	101.0	476	557
D0, PHY Energy Detect Power Down	Typ.	51.4	0.0	170	170

TABLE 19-5: CURRENT CONSUMPTION AND POWER DISSIPATION (REGS. ENABLED)

D0, PHY General Power Down	Typ.	48.4	0.0	160	160
D1, 100BASE-TX Idle (w/o EEE)	Typ.	66.6	41.0	323	356
D1, 100BASE-TX Idle (with EEE)	Typ.	59.2	0.0	196	196
D1, 10BASE-T Idle	Typ.	38.5	101.0	380	461
D1, PHY Energy Detect Power Down	Typ.	27.5	0.0	91	91
D1, PHY General Power Down	Typ.	24.3	0.0	81	81
D2, 100BASE-TX Idle (w/o EEE)	Typ.	66.4	41.0	322	355
D2, 100BASE-TX Idle (with EEE)	Typ.	59.4	0.0	196	196
D2, 10BASE-T Idle	Typ.	38.4	101.0	380	460
D2, PHY Energy Detect Power Down	Typ.	16.2	0.0	54	54
D2, PHY General Power Down	Typ.	13.3	0.0	44	44
D3, PHY General Power Down	Typ.	9.4	0.0	32	32

Note 12: VDD33TXRX1, VDD33TXRX2, VDD33BIAS, VDD33, VDDIO

Note 13: VDD12TX1 and VDD12TX2, are driven by the internal regulator via the PCB. The current is accounted for via VDD33.

Note 14: Current measurements do not include power applied to the magnetics or the optional external LEDs.

Note 15: The Ethernet component current is independent of the supply rail voltage (2.5V or 3.3V) of the transformer. Copper TP operation is assumed. Current is zero if using 100BASE-FX mode.

Note 16: This includes the power dissipated by the transmitter by way of the current through the transformer.

Note 17: 3.3*(A) + (2.5)*(C) @ Typ

Note 18: 3.3*(A) + (3.3)*(C) @ Typ

19.5 DC Specifications

TABLE 19-6: NON-VARIABLE I/O DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Units	Notes
IS Type Input Buffer						
Low Input Level	V_{ILI}	-0.3		0.8	V	
High Input Level	V_{IHI}	2.0		3.6	V	
Schmitt Trigger Hysteresis (V_{IHT} - V_{ILT})	V_{HYS}	121		151	mV	
Input Leakage (V_{IN} = VSS or VDD33)	I_{IH}	-10		10	µA	Note 19
Input Capacitance	C_{IN}			3	pF	
Pull-Up Impedance (V_{IN} = VSS)	R_{DPU}	6		8.9	KΩ	
Pull-Down Impedance (V_{IN} = VDD33)	R_{DPD}	52		79	KΩ	
AI Type Input Buffer (FXSDENA)						
Low Input Level	V_{IL}	-0.3		0.8	V	
High Input Level	V_{IH}	1.2		VDD33+0.3	V	
AI Type Input Buffer (RXPA/RXNA)						
Differential Input Level	$V_{IN-DIFF}$	0.1		VDD33TXRX1	V	
Common Mode Voltage	V_{CM}	1.0	VDD33TXRX1-1.3		V	
Input Capacitance	C_{IN}			5	pF	
AI Type Input Buffer (FXLOSEN Input)						
State A Threshold	V_{THA}	-0.3		0.8	V	
State B Threshold	V_{THB}	1.2		VDD33+0.3	V	
ICLK Type Input Buffer (OSCI Input)						Note 20
Low Input Level	V_{ILI}	-0.3		0.35	V	
High Input Level	V_{IHI}	OSCVDD12-0.35		3.6	V	
Input Leakage	I_{ILCK}	-10		10	µA	
ILVPECL Input Buffer						
Low Input Level	V_{IL}-VDD33TXRX1	VDD33TXRX1+0.3		-1.48	V	Note 21
High Input Level	V_{IH}-VDD33TXRX1	-1.14		0.3	V	Note 21

TABLE 19-6: NON-VARIABLE I/O DC ELECTRICAL CHARACTERISTICS (CONTINUED)

Parameter	Symbol	Min	Typ	Max	Units	Notes
OLVPECL Output Buffer						
Low Output Level	V_{OL}			VDD33TXRX1-1.62	V	
High Output Level	V_{OH}	VDD33TXRX1-1.025			V	
Peak-to-Peak Differential (SFF mode)	$V_{DIFF-SFF}$	1.2	1.6	2.0	V	
Peak-to-Peak Differential (SFP mode)	$V_{DIFF-SFP}$	0.6	0.8	1.0	V	
Common Mode Voltage	V_{CM}	1.0	VDD33TXRX1-1.3		V	
Offset Voltage	V_{OFFSET}		40		mV	Note 22
Load Capacitance	C_{LOAD}			10	pF	

Note 19: This specification applies to all inputs and tri-stated bi-directional pins. Internal pull-down and pull-up resistors add +/- 50 µA per-pin (typical).

Note 20: OSCI can optionally be driven from a 25 MHz singled-ended clock oscillator.

Note 21: LVPECL compatible.

Note 22: V_{OFFSET} is a function of the external resistor network configuration. The listed value is recommended to prevent issues due to crosstalk.

TABLE 19-7: VARIABLE I/O DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	Min	1.8 V Typ	3.3 V Typ	Max	Units	Notes
VIS Type Input Buffer							
Low Input Level	V_{ILI}	-0.3				V	
High Input Level	V_{IHI}				3.6	V	
Negative-Going Threshold	V_{ILT}	0.64	0.83	1.41	1.76	V	Schmitt trigger
Positive-Going Threshold	V_{IHT}	0.81	0.99	1.65	1.90	V	Schmitt trigger
Schmitt Trigger Hysteresis (V_{IHT} - V_{ILT})	V_{HYS}	102	158	138	288	mV	
Input Leakage (V_{IN} = **VSS** or **VDDIO**)	I_{IH}	-10			10	µA	Note 23
Input Capacitance	C_{IN}				2	pF	
Pull-Up Impedance (V_{IN} = VSS)	R_{DPU}	54	68	82		KΩ	
Pull-Up Current (V_{IN} = VSS)	I_{DPU}	20	27	67		µA	
Pull-Down Impedance (V_{IN} = VDD33)	R_{DPD}	54	68	85		KΩ	
Pull-Down Current (V_{IN} = VDD33)	I_{DPD}	19	26	66		µA	
VO8 Type Buffers							
Low Output Level	V_{OL}				0.4	V	I_{OL} = 8 mA
High Output Level	V_{OH}	VDDIO - 0.4				V	I_{OH} = -8 mA
VOD8 Type Buffer							
Low Output Level	V_{OL}				0.4	V	I_{OL} = 8 mA
VO12 Type Buffers							
Low Output Level	V_{OL}				0.4	V	I_{OL} = 12 mA
High Output Level	V_{OH}	VDDIO - 0.4				V	I_{OH} = -12 mA
VOD12 Type Buffer							
Low Output Level	V_{OL}				0.4	V	I_{OL} = 12 mA
VOS12 Type Buffers							
High Output Level	V_{OH}	VDDIO - 0.4				V	I_{OH} = -12 mA
VO16 Type Buffers							
Low Output Level	V_{OL}				0.4	V	I_{OL} = 16 mA
High Output Level	V_{OH}	VDDIO - 0.4				V	I_{OH} = -16 mA

Note 23: This specification applies to all inputs and tri-stated bi-directional pins. Internal pull-down and pull-up resistors add ±50 µA per-pin (typical).

TABLE 19-8: 100BASE-TX TRANSCEIVER CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Peak Differential Output Voltage High	V_{PPH}	950	-	1050	mVpk	Note 24
Peak Differential Output Voltage Low	V_{PPL}	-950	-	-1050	mVpk	Note 24
Signal Amplitude Symmetry	V_{SS}	98	-	102	%	Note 24
Signal Rise and Fall Time	T_{RF}	3.0	-	5.0	ns	Note 24
Rise and Fall Symmetry	T_{RFS}	-	-	0.5	ns	Note 24
Duty Cycle Distortion	D_{CD}	35	50	65	%	Note 25
Overshoot and Undershoot	V_{OS}	-	-	5	%	
Jitter	-	-	-	1.4	ns	Note 26

Note 24: Measured at line side of transformer, line replaced by 100 Ω (+/- 1%) resistor.

Note 25: Offset from 16 ns pulse width at 50% of pulse peak.

Note 26: Measured differentially.

TABLE 19-9: 10BASE-T TRANSCEIVER CHARACTERISTICS

Parameter	Symbol	Min	Typ	Max	Units	Notes
Transmitter Peak Differential Output Voltage	V_{OUT}	2.2	2.5	2.8	V	Note 27
Receiver Differential Squelch Threshold	V_{DS}	300	420	585	mV	

Note 27: Min/max voltages guaranteed as measured with 100 Ω resistive load.

19.6 AC Specifications

This section details the various AC timing specifications of the device.

Note: The I^2C timing adheres to the NXP *I^2C-Bus Specification*. Refer to the NXP *I^2C-Bus Specification* for detailed I^2C timing information.

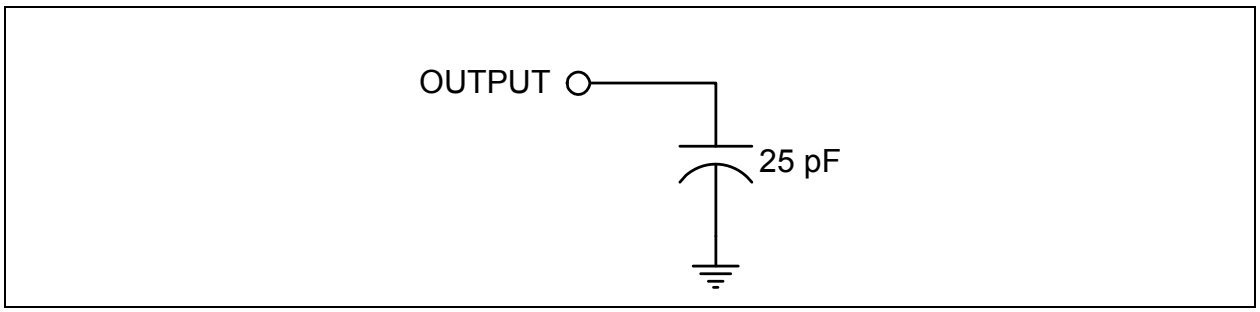
Note: The MII/SMI timing adheres to the *IEEE 802.3 Specification*.

Note: The RMII timing adheres to the RMII Consortium *RMII Specification R1.2*.

19.6.1 EQUIVALENT TEST LOAD

Output timing specifications assume the 25 pF equivalent test load, unless otherwise noted, as illustrated in Figure 19-1.

FIGURE 19-1: OUTPUT EQUIVALENT TEST LOAD



19.6.2 POWER SEQUENCING TIMING

These diagrams illustrates the device power sequencing requirements. The **VDDIO**, **VDD33**, **VDD33TXRX1**, **VDD33TXRX2**, **VDD33BIAS** and magnetics power supplies must all reach operational levels within the specified time period t_{pon}. When operating with the internal regulators disabled, **VDDCR**, **OSCVDD12**, **VDD12TX1** and **VDD12TX2** are also included into this requirement.

In addition, once the **VDDIO** power supply reaches 1.0 V, it must reach 80% of its operating voltage level (1.44 V when operating at 1.8 V, 2.0 V when operating at 2.5 V, 2.64 V when operating at 3.3 V) within an additional 15ms. This requirement can be safely ignored if using an external reset as shown in Section 19.6.3, "Reset and Configuration Strap Timing".

Device power supplies can turn off in any order provided they all reach 0 volts within the specified time period t_{poff}.

FIGURE 19-2: POWER SEQUENCE TIMING - INTERNAL REGULATORS

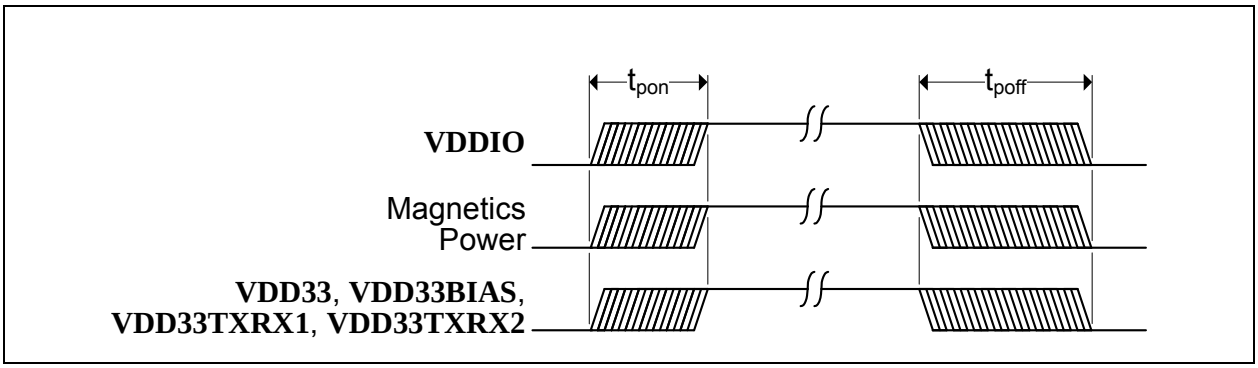


FIGURE 19-3: POWER SEQUENCE TIMING - EXTERNAL REGULATORS

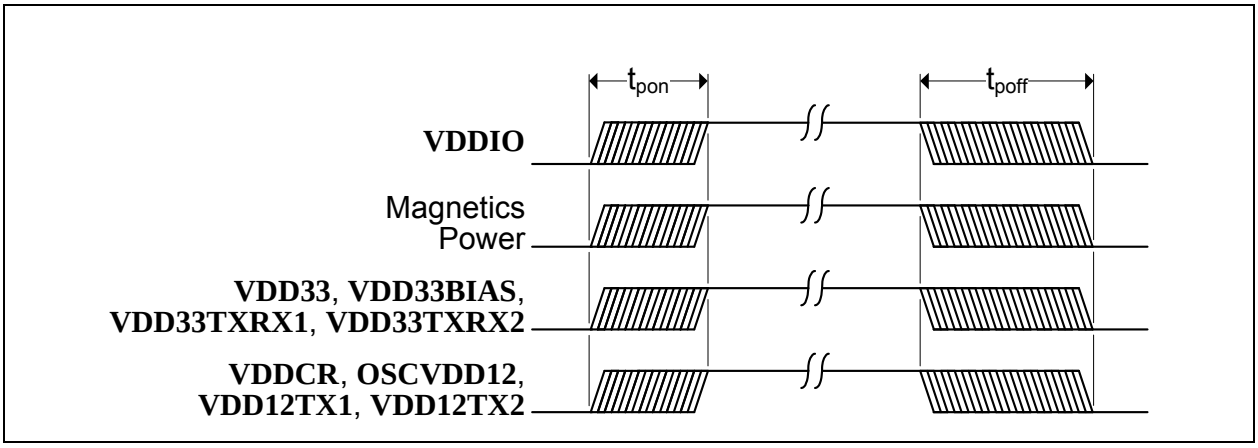


TABLE 19-10: POWER SEQUENCING TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{pon}	Power supply turn on time	-	-	50	ms
t_{poff}	Power supply turn off time	-	-	500	ms

19.6.3 RESET AND CONFIGURATION STRAP TIMING

This diagram illustrates the RST# pin timing requirements and its relation to the configuration strap pins and output drive. Assertion of RST# is not a requirement. However, if used, it must be asserted for the minimum period specified. The RST# pin can be asserted at any time, but must not be deasserted until t_{purstd} after all external power supplies have reached operational levels. Refer to Section 6.2, "Resets," on page 38 for additional information.

FIGURE 19-4: RST# PIN CONFIGURATION STRAP LATCHING TIMING

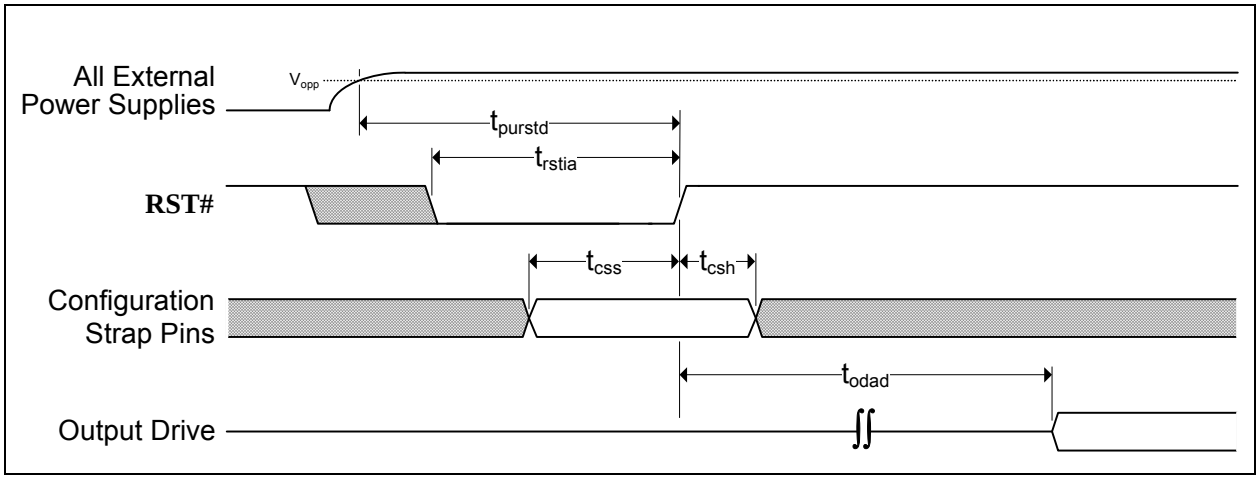


TABLE 19-11: RST# PIN CONFIGURATION STRAP LATCHING TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{purstd}	External power supplies at operational level to RST# deassertion	25			ms
t_{rstia}	RST# input assertion time	200	-	-	µs
t_{css}	Configuration strap pins setup to RST# deassertion	200	-	-	ns
t_{csh}	Configuration strap pins hold after RST# deassertion	10	-	-	ns
t_{odad}	Output drive after deassertion	3	-	-	us

Note: The clock input must be stable prior to RST# deassertion.

Note: Device configuration straps are latched as a result of RST# assertion. Refer to Section 6.2.1, "Chip-Level Resets," on page 39 for details.

Note: Configuration strap latching and output drive timings shown assume that the Power-On reset has finished first otherwise the timings in Section 19.6.4, "Power-On and Configuration Strap Timing" apply.

19.6.4 POWER-ON AND CONFIGURATION STRAP TIMING

This diagram illustrates the configuration strap valid timing requirements in relation to power-on. In order for valid configuration strap values to be read at power-on, the following timing requirements must be met.

FIGURE 19-5: POWER-ON CONFIGURATION STRAP LATCHING TIMING

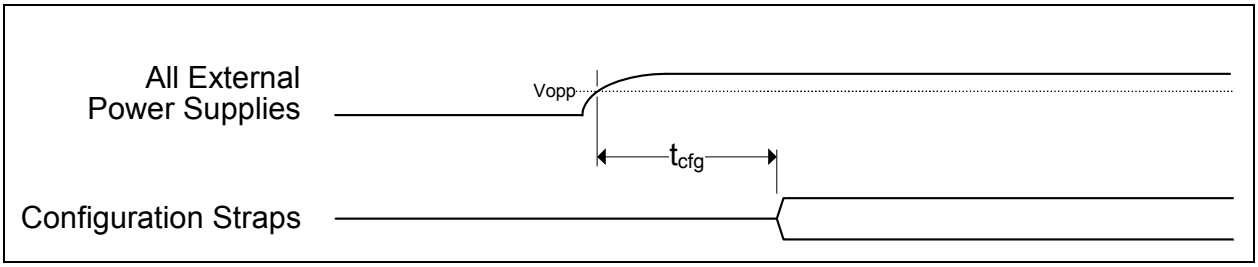


TABLE 19-12: POWER-ON CONFIGURATION STRAP LATCHING TIMING VALUES

Symbol	Description	Min	Typ	Max	Units
t_{cfg}	Configuration strap valid time	-	-	15	ms

Note: Configuration straps must only be pulled high or low. Configuration straps must not be driven as inputs.

Device configuration straps are also latched as a result of RST# assertion. Refer to Section 19.6.3, "Reset and Configuration Strap Timing" and Section 6.2.1, "Chip-Level Resets," on page 39 for additional details.

19.6.5 HOST BUS INTERFACE I/O TIMING

Timing specifications for the Host Bus Interface are given in Section 9.4.5, "Multiplexed Addressing Mode Timing Requirements," on page 90 and Section 9.5.7, "Indexed Addressing Mode Timing Requirements," on page 115.

19.6.6 SPI/SQI SLAVE INTERFACE I/O TIMING

Timing specifications for the SPI/SQI Slave Bus Interface are given in Section 10.4, "SPI/SQI Timing Requirements," on page 138.

19.6.7 I^2C EEPROM I/O TIMING

Timing specifications for I^2C EEPROM access are given in Section 13.3, "I2C Master EEPROM Controller," on page 283.

19.6.8 JTAG TIMING

Timing specifications for the JTAG interface are given in Table 18.1.1, “JTAG Timing Requirements,” on page 398.

19.7 Clock Circuit

The device can accept either a 25 MHz crystal or a 25 MHz single-ended clock oscillator (±50 ppm) input. If the single-ended clock oscillator method is implemented, **OSCO** should be left unconnected and **OSCI** should be driven with a clock signal that adheres to the specifications outlined throughout Section 19.0, "Operational Characteristics". See Table 19-13 for the recommended crystal specifications.

TABLE 19-13: CRYSTAL SPECIFICATIONS

PARAMETER	SYMBOL	MIN	NOM	MAX	UNITS	NOTES
Crystal Cut	AT, typ					
Crystal Oscillation Mode	Fundamental Mode					
Crystal Calibration Mode	Parallel Resonant Mode					
Frequency	F_{fund}	-	25.000	-	MHz	
802.3 Frequency Tolerance at 25°C	F_{tol}	-	-	±40	ppm	Note 28
802.3 Frequency Stability Over Temp	F_{temp}	-	-	±40	ppm	Note 28
802.3 Frequency Deviation Over Time	F_{age}	-	±3 to 5	-	ppm	Note 29
802.3 Total Allowable PPM Budget		-	-	±50	ppm	Note 30
Shunt Capacitance	C_O	-	-	7	pF	
Load Capacitance	C_L	-	-	18	pF	
Drive Level	P_W	300 Note 31	-	-	µW	
Equivalent Series Resistance	R_1	-	-	100	Ω	
Operating Temperature Range		Note 32	-	Note 33	°C	
OSCI Pin Capacitance		-	3 typ	-	pF	Note 34
OSCO Pin Capacitance		-	3 typ	-	pF	Note 34

Note 28: The maximum allowable values for frequency tolerance and frequency stability are application dependent. Since any particular application must meet the IEEE ±50 ppm Total PPM Budget, the combination of these two values must be approximately ±45 ppm (allowing for aging).

Note 29: Frequency Deviation Over Time is also referred to as Aging.

Note 30: The total deviation for 100BASE-TX is ±50 ppm.

Note 31: The minimum drive level requirement P_W is reduced to 100 uW with the addition of a 500 Ω series resistor, if $C_O \leq 5$ pF, $C_L \leq 12$ pF and $R1 \leq 80$ Ω

Note 32: 0 °C for commercial version, -40 °C for industrial and extended industrial versions

Note 33: +70 °C for commercial version, +85 °C for industrial version, +105 °C for extended industrial version

Note 34: This number includes the pad, the bond wire and the lead frame. PCB capacitance is not included in this value. The **OSCI** pin, **OSCO** pin and PCB capacitance values are required to accurately calculate the value of the two external load capacitors. The total load capacitance must be equivalent to what the crystal expects to see in the circuit so that the crystal oscillator will operate at 25.000 MHz.

20.0 PACKAGE OUTLINES

20.1 64-QFN

FIGURE 20-1: 64-QFN PACKAGE

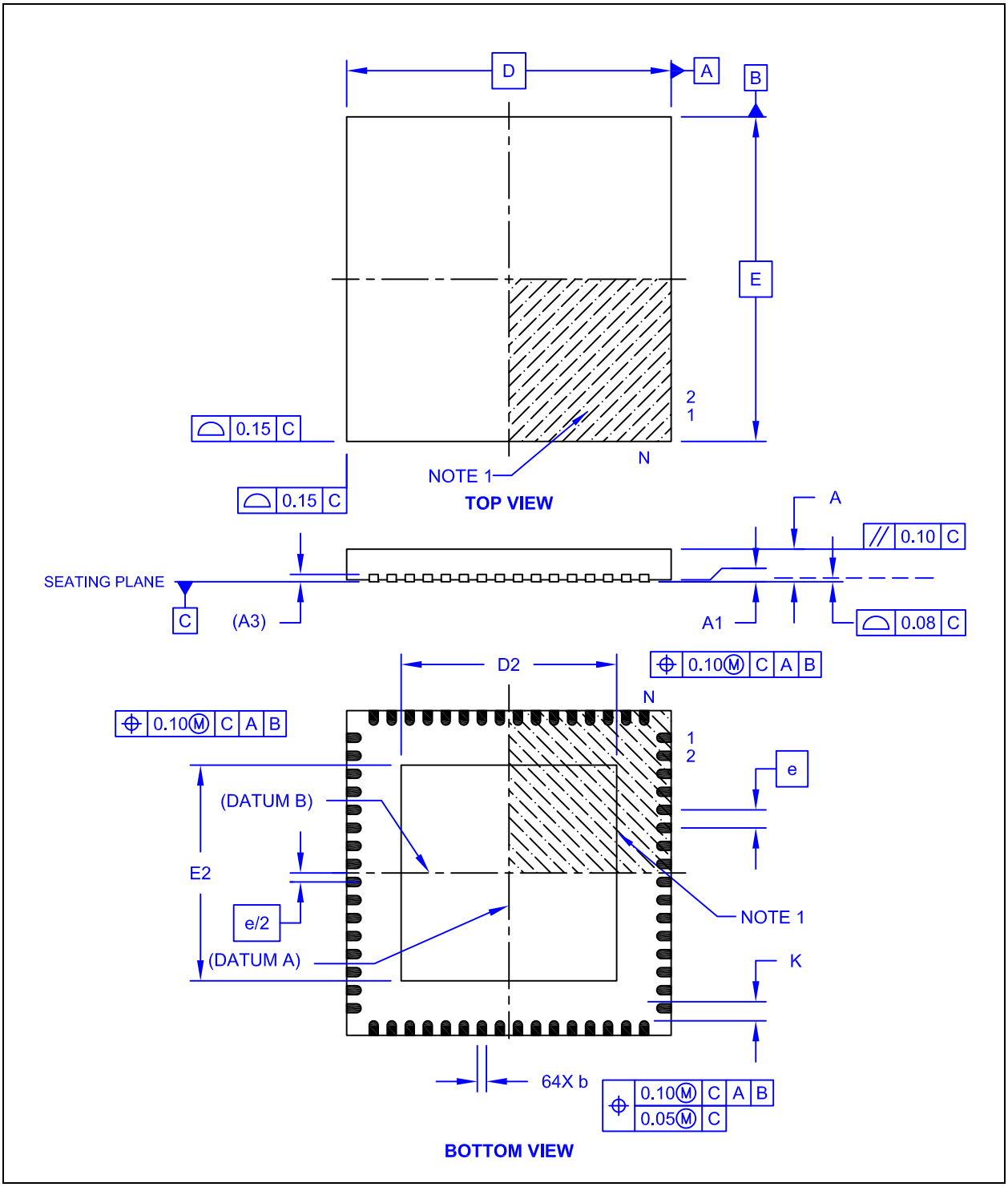
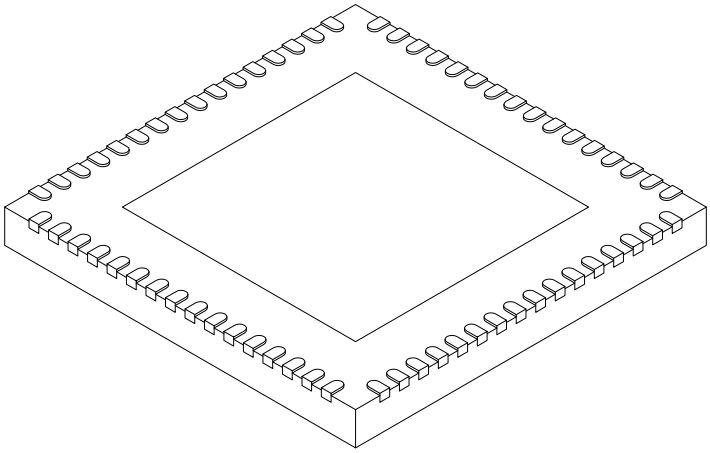


FIGURE 20-2: 64-QFN PACKAGE DIMENSIONS



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Pins	N		64	
Pitch	e		0.50 BSC	
Overall Height	A	0.80	0.90	1.00
Standoff	A1	0.00	0.02	0.05
Contact Thickness	A3		0.20 REF	
Overall Width	E		9.00 BSC	
Exposed Pad Width	E2	5.90	6.00	6.10
Overall Length	D		9.00 BSC	
Exposed Pad Length	D2	5.90	6.00	6.10
Contact Width	b	0.18	0.25	0.30
Contact Length	L	0.30	0.40	0.50
Contact-to-Exposed Pad	K	0.90	1.10	-

Notes:

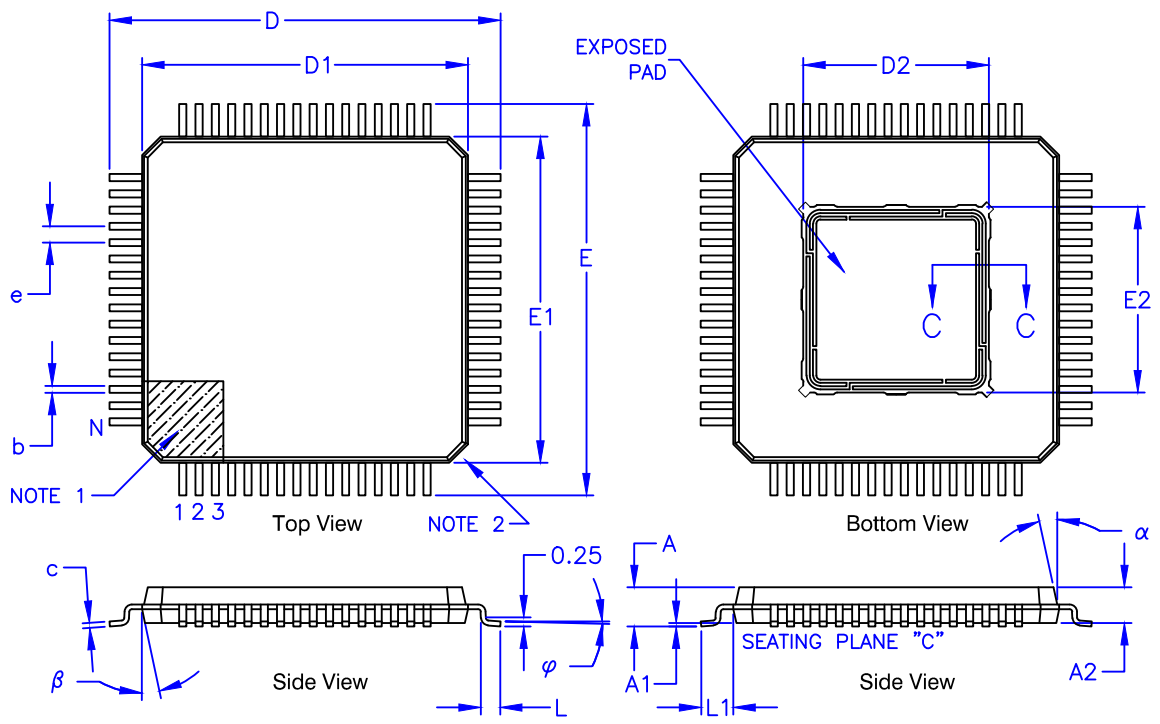
1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Package is saw singulated.
3. Dimensioning and tolerancing per ASME Y14.5M.

   BSC: Basic Dimension. Theoretically exact value shown without tolerances.

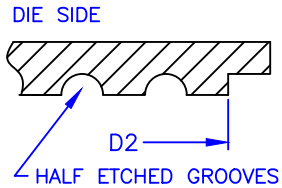
   REF: Reference Dimension, usually without tolerance, for information purposes only.

20.2 64-TQFP-EP

FIGURE 20-3: 64-TQFP-EP PACKAGE



Units		MILLIMETERS		
Dimension Limits		MIN	NOM	MAX
Number of Leads	N		64	
Lead Pitch	e		0.50 BSC	
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1		1.00 REF	
Foot Angle	φ	0°	3.5°	7°
Overall PKG Length/Width	D/E		12.00 BSC	
PKG Body Length/Width	D1/E1		10.00 BSC	
Exposed Pad Length/Width	D2/E2	5.65	5.70	5.75
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°



SECTION C - C

Lead Position Tolerance (dim "b")

⌖	0.08Ⓜ	C

Lead Coplanarity Tolerance

⌓	0.08	C

Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.
2. Chamfers at corners are optional; size may vary.
3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25mm per side.
4. Dimensioning and tolerancing per ASME Y14.5M
   BSC: Basic Dimension. Theoretically exact value shown without tolerances.
   REF: Reference Dimension, usually without tolerance, for information purposes only.

21.0 REVISION HISTORY

TABLE 21-1: REVISION HISTORY

Revision Level	Section/Figure/Entry	Correction
DS00001913A (06-30-15)		Initial Release

THE MICROCHIP WEB SITE

Microchip provides online support via our WWW site at www.microchip.com. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

CUSTOMER CHANGE NOTIFICATION SERVICE

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at www.microchip.com. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

CUSTOMER SUPPORT

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or field application engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: http://microchip.com/support

PRODUCT IDENTIFICATION SYSTEM

To order or obtain information, e.g., on pricing or delivery, refer to the factory or the listed sales office.

PART NO.	[X]	[X]	/	XX
Device	Tape and Reel Option	Temperature Range		Package

Device:	LAN9250	
Tape and Reel Option:	Blank = Standard packaging (tray) T = Tape and Reel(Note 1)	
Temperature Range:	Blank = 0°C to +70°C I = -40°C to +85°C V = -40°C to +105°C	(Commercial) (Industrial) (Extended Industrial)(Note 2)
Package:	ML = 64-pin QFN PT = 64-pin TQFP-EP	

Examples:

a) LAN9250/ML
Standard Packaging (Tray),
Commercial Temperature,
64-pin QFN

b) LAN9250TI/PT
Tape and Reel
Industrial Temperature,
64-pin TQFP-EP

Note 1: Tape and Reel identifier only appears in the catalog part number description. This identifier is used for ordering purposes and is not printed on the device package. Check with your Microchip Sales Office for package availability with the Tape and Reel option.

2: Extended industrial temp. support (105°C) in the 64-QFN only

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

ISBN: 9781632772152

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
═ ISO/TS 16949 ═

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
http://www.microchip.com/support
Web Address:
www.microchip.com

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983
Indianapolis

Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2401-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9588
Fax: 86-23-8980-9500

China - Dongguan
Tel: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2401-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-27-5980-5300
Fax: 86-27-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC

China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4123

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880- 3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Germany - Pforzheim
Tel: 49-7231-424750

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

01/27/15