Data Sheet
April 2018

MAX24288

IEEE 1588 Packet Timestamper and Clock and 1Gbps Parallel-to-Serial MII Converter

General Description

The MAX24288 is a flexible, low-cost IEEE 1588 clock and timestamper with an SGMII or 1000BASE-X serial interface and a parallel MII interface that can be configured for GMII, RGMII, or 10/100 MII. The device provides all required hardware support for high-accuracy time and frequency synchronization using the IEEE 1588 Precision Time Protocol. In both the transmit and receive directions 1588 packets are identified and timestamped with high precision. System software makes use of these timestamps to determine the time offset between the system and its timing master. Software can then correct any time error by steering the device's 1588 clock subsystem appropriately. The device provides the necessary I/O to time-synchronize with a 1588 master elsewhere in the same system or to be the master to which slave components can synchronize.

In addition, the MAX24288 is a full-featured, gigabit parallel-to-serial MII converter. It provides full SGMII revision 1.8 compliance and also interfaces directly to 1Gbps 1000BASE-X SFP optical modules.

Applications

1588-Enabled Equipment with 1G Ethernet Ports
Wireless Base Stations and Controllers
Switches, Routers, DSLAMs, PON Equipment
Pseudowire Circuit Emulation Equipment
Test and Measurement Systems
Industrial and Factory Automation Equipment
Medical Equipment

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX24288ETK2	-40°C to +85°C	68 TQFN-EP* trays
MAX24288ETK2T	-40°C to +85°C	68 TQFN-EP* tape & reel

Suffix 2 denotes a lead(Pb)-free/RoHS-compliant package.
**EP = Exposed pad.*

Block Diagram appears on page 8.

Register Map appears on page 64.

Highlighted Features

- **Complete Hardware Support for IEEE 1588**
- **Ordinary, Boundary, and Transparent Clocks**
- **Flexible Block for Any 1588 Architecture**
- **1588 Clock Hardware**
  - Steerable by Software with 2^{-8}ns Time Resolution and 2^{-32}ns Period Resolution
  - 1ns Input Timestamp Accuracy and Output Edge Placement Accuracy
  - Three Time/Frequency Controls: Direct Time Write, Time Adjustment, and High-Resolution Frequency Adjustment
  - Programmable Clock and Time-Alignment I/O
  - Input Event Timestamper Detects Incoming Time Alignment (e.g., 1 PPS) or Clock Edges
  - Output Event Generator Provides Output Clock Signal or Time Alignment Signal
  - Built-In Support for Telecom Equipment Timing Architecture with Dual Redundant Timing Cards
- **1588 Timestamping Hardware**
  - 1588 v1 and v2 Packets, Transmit and Receive
  - Packet Classifier Supports 1588 Over Ethernet, IPv4/UDP, IPv6/UDP ,or MPLS, and Is Programmable for More Complex Stacks
  - Supports 802.1Q VLAN Tags and MAC-in-MAC
  - One-Step Operation: On-the-Fly Timestamp Insertion or Transparent Clock Corrections; No Need for Follow-Up Packets
  - Can Insert All Timestamps, Receive **and** Transmit, Into Packets for Easy Software Access
  - Optional Two-Step Operation
- **Parallel-to-Serial MII Conversion**
  - Bidirectional Wire-Speed Interface Conversion
  - Serial: 1000BASE-X or SGMII v1.8 (4, 6, or 8 Pin)
  - Parallel: GMII, RGMII, or 10/100 MII
  - Translates Link Speed and Duplex Mode Negotiation Between MDIO and SGMII PCS
- **Full Support for 1588 + Synchronous Ethernet**
- **MDIO and SPI™ Interfaces**
- **1.2V Operation with 3.3V I/O**

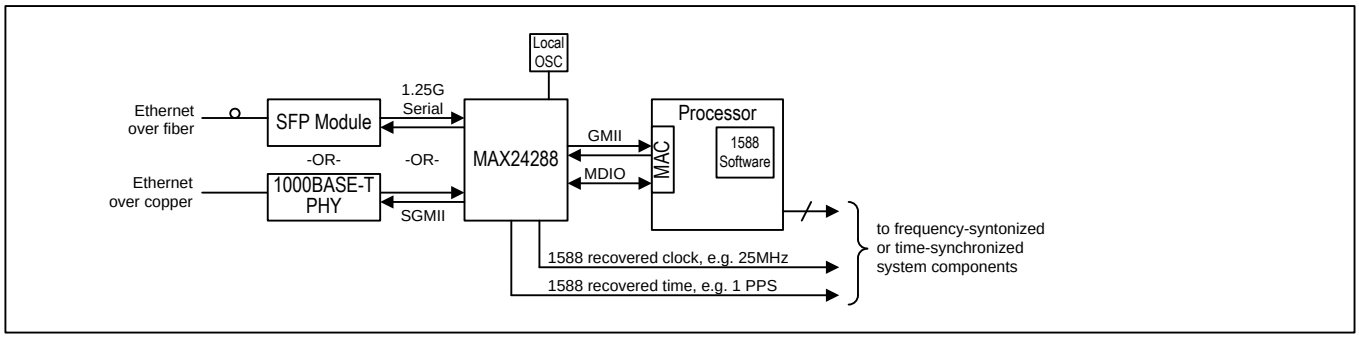
TABLE OF CONTENTS

TABLE OF FIGURES

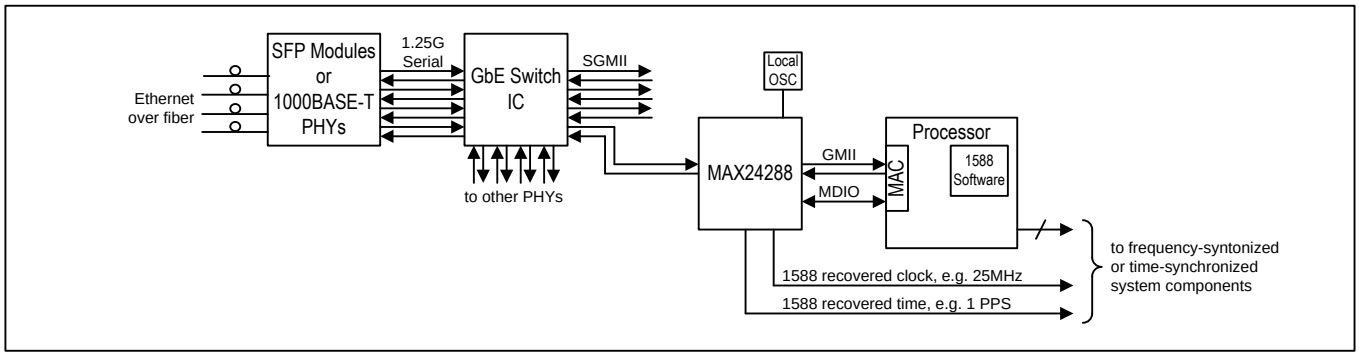
TABLE OF TABLES

1. Application Examples

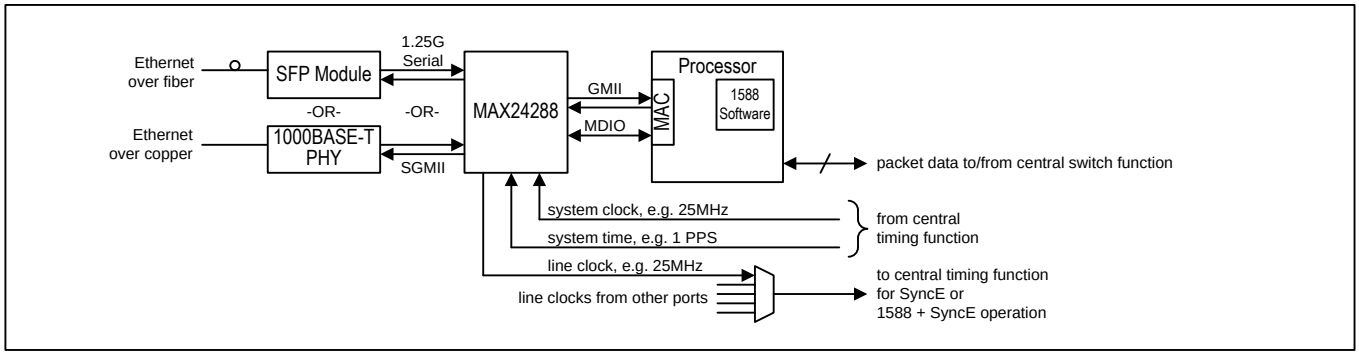
Example 1: Single-Port 1588 Slave Node



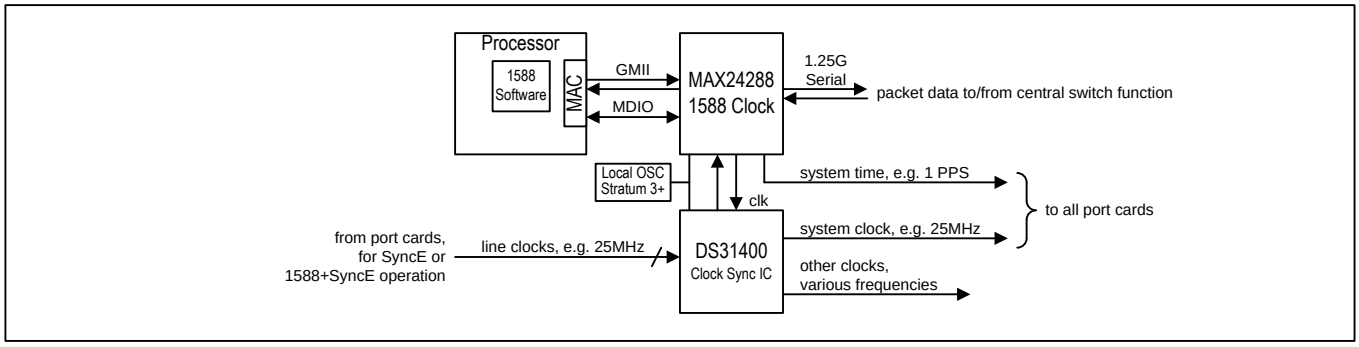
Example 2: Multiport System with Switch-Connected 1588 Slave Node



Example 3: Multiport System, Boundary or Transparent Clock, Port Card Logic

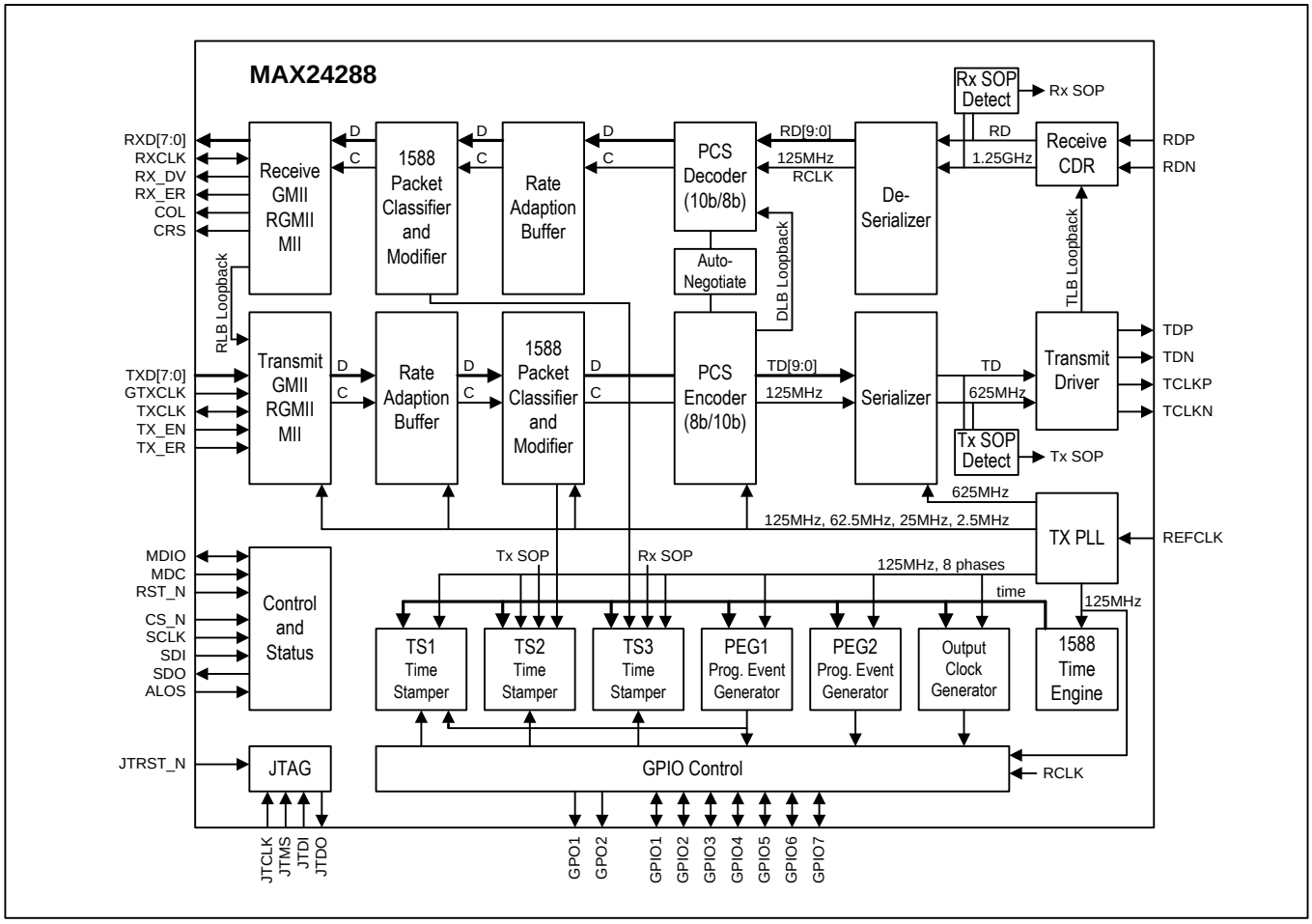


Example 4: Multiport System, Boundary or Transparent Clock, Central Timing Function



2. Block Diagram

Figure 2-1. Block Diagram



3. Detailed Features

General Features

- Control and status through MDIO interface or SPI interface
- High-speed MDIO interface (12.5MHz slave only) with optional preamble suppression
- Optional SPI 4-wire serial microprocessor interface (25MHz, slave only)
- Operates from a 10, 12.8, 25 or 125MHz reference clock
- Optional 125MHz output clock for MAC to use as GTXCLK

Parallel-Serial MII Conversion Features

- Bidirectional wire-speed interface conversion
- Serial Interface: 1000BASE-X or SGMII revision 1.8 (4-, 6- or 8-Pin)
- Parallel Interface: GMII, RGMII (10, 100 and 1000Mbps) or 10/100 MII (DTE or DCE)
- 8-pin source-clocked SGMII mode
- 4-pin 1000BASE-X SERDES mode to interface with optical modules
- Connects processors with parallel MII interfaces to 1000BASE-X SFP optical modules
- Connects processors with parallel MII interfaces to PHY or switch ICs with SGMII interfaces
- Interface conversion is transparent to MAC layer and higher layers

- Translates link speed and duplex mode between GMII/MII MDIO and SGMII PCS

1588 Clock Features

- Steerable by software with 2^{-8}ns time resolution and 2^{-32}ns period resolution
- 1ns input timestamp accuracy and output edge placement accuracy
- Initialized and steered by software on an external processor to follow an external 1588 master
- Three time/frequency controls: direct time write, time adjustment, and high-resolution frequency adjustment
- Programmable clock and time-alignment I/O to synchronize all boards in large systems
  - Can frequency-lock to an input clock signal from elsewhere in the system
  - Can timestamp an input time alignment signal to time-lock to a master elsewhere in the system (e.g. 1 PPS)
  - Can provide an output clock signal to slave components elsewhere in the system (125MHz / N , 1≤N≤255)
  - Can provide an output time alignment signal to slave components elsewhere in the system (e.g. 1 PPS)
- Input signal timestamper can stamp rising edges, falling edges or both
- Flexible programmable event generator (PEG) can output 1 PPS, one pulse per period, and a wide variety of clock signals
- Full support for dual redundant timing cards to match architecture used in SONET/SDH
- Full support for switches and routers as transparent clocks or boundary clocks
- Compatible with a wide variety of 1588 system architectures

1588 Timestamper Features

- Identifies and timestamps 1588 v1 and v2 packets in both transmit and receive directions
- Programmable packet classifier can identify packets transported by a variety of protocol stacks
  - 1588 over Ethernet
  - 1588 over IPv4/UDP
  - 1588 over IPv6/UDP
  - 1588 over MPLS
  - Configurable for more complex stacks as well
  - Recognizes 802.1Q VLAN tags and 802.1ah MAC-in-MAC
  - Can be configured to identify CESoP or SAToP for timing over adaptive-mode circuit emulation
- Transmit and receive timestamping with 1ns resolution
- One-step operation minimizes network bandwidth consumption
  - On-the-fly timestamp insertion
  - On-the-fly corrections in transparent clocks
  - No need for follow-up packets
- Can insert ALL timestamps (receive and transmit) into packets for easy software access
  - Three insert methods: direct overwrite, read-add-write, and read-subtract-write
  - Eliminates reads from timestamp FIFOs
  - Minimizes processor bus traffic
- Optional two-step operation
- Optional 8-entry timestamp FIFOs

Synchronous Ethernet Features

- Full support for 1588 over Synchronous Ethernet
- Receive path bit clock can be output on a GPIO pin to line-time the system from the Ethernet port
- Transmit path can be frequency-locked to a system clock signal connected to the REFCLK pin

4. Acronyms, Abbreviations, and Glossary

- BC Boundary Clock
- DCE Data Communication Equipment
- DDR Dual Data Rate (data driven and latched on both clock edges)
- DTE Data Terminating Equipment
- E2E End to End
- OC Ordinary Clock
- P2P Peer to Peer
- PCB Printed Circuit Board
- PHY Physical. Refers to either a transceiver device or a protocol layer
- PTP Precision Time Protocol – IEEE1588
- TC Transparent Clock

- Ingress The serial (SGMII) to parallel (GMII) direction
- Egress The parallel (GMII) to serial (SGMII) direction
- Receive The serial (SGMII) to parallel (GMII) direction
- Transmit The parallel (GMII) to serial (SGMII) direction

5. Pin Descriptions

Note that some pins have different pin names and functions under different configurations.

Table 5-1. Pin Type Descriptions

Type	Definition
I	Input
Idiff	Input differential
IO	Bi-directional
IOr	Bi-directional, sampled at reset
IOz	Bi-directional, can go high impedance
O	Output
Odiff	Output, differential (CML)
Oz	Output, can go hi impedance

Table 5-2. Detailed Pin Descriptions – Global Pins (2 Pins)

Pin Name	PIN #	Type	Pin Description
RST_N	67	I	**Reset** (active low, asynchronous) This signal resets all logic, state machines and registers in the device. Pin states are sampled and used to set the default values of several register fields as described in 6.1. RST_N should be held low for at least 100µs. See section 6.3.1.

Pin Name	PIN #	Type	Pin Description
REFCLK	68	I	**Reference Clock** This signal is the reference clock for the device. The frequency can be 10MHz, 12.8MHz, 25MHz or 125MHz ± 100 ppm. At reset the frequency is specified using the RXD[3:2] pins (see section 6.1). The REFCLK signal is the input clock to the TX PLL. See section 6.10. Note: REFCLK frequency cannot be changed dynamically among the frequencies listed above. To change REFCLK frequency, (1) power down MAX24288, (2) change REFCLK frequency, then (3) power up MAX24288. REFCLK is an analog input that is internally biased with a 10kΩ resistor to 1.2V. This support AC-coupling if desired.

Table 5-3. Detailed Pin Descriptions – MDIO Interface (2 Pins)

Pin Name	PIN #	Type	Pin Description
MDC	41	I	**MDIO Clock.** MDC is the clock signal of the 2-wire MDIO interface. It can be any frequency up to 12.5MHz. See section 6.5.
MDIO	42	IOz	**MDIO Data.** This is the bidirectional, half-duplex data signal of the MDIO interface. It is sampled and updated on positive edges of MDC. IEEE 802.3 requires a 2kΩ±5% pulldown resistor on this signal at the MAC. See section 6.5.

Table 5-4. Detailed Pin Descriptions – SPI Interface (4 pins)

Pin Name	PIN #	Type	Pin Description
SCLK	64	I	**SPI Clock Input.** SCLK can be any frequency up to 25MHz. By default, SDI and CS_N are sampled on the rising edge of SCLK, and SDO is updated on the falling edge of SCLK. The edge polarity and phase can be changed using PAGESEL.CPHA and CPOL. See section 6.4.
CS_N	45	I	**SPI Chip Select.** This signal must be asserted (low) to read or write internal registers using the SPI interface. See section 6.4.
SDI	63	I	**SPI Data Input.** The SPI bus master transmits data to the device on this pin. See section 6.4.
SDO	62	Oz	**SPI Data Output.** The device transmits data to the SPI bus master on this pin. SDO is high impedance until a read command is clocked into the device on the SDI pin. SDO then outputs the data values and returns to high impedance. See section 6.4.

Table 5-5. Detailed Pin Descriptions – JTAG Interface (5 pins)

Pin Name	PIN #	Type	Pin Description
JTRST_N	43	I	**JTAG Test Reset** (active low). Asynchronously resets the test access port (TAP) controller. JTRST_N should be held low during device power-up. If not used, JTRST_N can be held low or high after power-up. See section 8.
JTCLK	21	I	**JTAG Test Clock.** This clock signal can be any frequency up to 10MHz. JTDI and JTMS are sampled on the rising edge of JTCLK, and JTDO is updated on the falling edge of JTCLK. If not used, connect to DVDD33 or DVSS. See section 8.
JTMS	22	I	**JTAG Test Mode Select.** Sampled on the rising edge of JTCLK. Used to place the port into the various defined IEEE 1149.1 states. If not used, connect to DVDD33. See section 8.

Pin Name	PIN #	Type	Pin Description
JTDI	23	I	**JTAG Test Data Input.** Test instructions and data are clocked in on this pin on the rising edge of JTCLK. If not used, connect to DVDD33. See section 8.
JTDO	44	Oz	**JTAG Test Data Output.** Test instructions and data are clocked out on this pin on the falling edge of JTCLK. If not used leave unconnected. See section 8.

Table 5-6. Detailed Pin Descriptions – GPIO signals (5 dedicated pins, 4 shared pins)

Pin Name	PIN #	Type	Pin Description
GPO1	24	IOr	**General Purpose Output 1.** After reset, default behavior is to output a signal that indicates link status, 0=link down, 1=link up. The function can be changed after reset. See section 6.2.
GPO2	25	IOr	**General Purpose Output 2.** After reset, default behavior is to output the CRS (carrier sense) signal. The function can be changed after reset. See section 6.2.
GPIO1	61	IOz	**General Purpose Input or Output 1.** After reset this pin can be either high impedance or generating a 125MHz clock signal. GPO1=0 at reset: After reset, GPIO1 is high impedance. GPO1=1 at reset: After reset, GPIO1 is 125MHz clock out The function can be changed after reset. See section 6.2.
GPIO2	60	IOz	**General Purpose Input or Output 2.** After reset this pin is high impedance. The function can be changed after reset. See section 6.2.
GPIO3	59	IOz	**General Purpose Input or Output 3.** After reset this pin is high impedance. The function can be changed after reset. See section 6.2.
GPIO4/TXD[4]	52	IOz	**General Purpose Input or Output 4.** Available for use as a GPIO pin when the parallel interface is configured for MII or RGMII modes. After reset this pin is high impedance. The function can be changed after reset. See section 6.2.
GPIO5/TXD[5]	53	IOz	**General Purpose Input or Output 5.** Available for use as a GPIO pin when the parallel interface is configured for MII or RGMII modes. After reset this pin is high impedance. The function can be changed after reset. See section 6.2.
GPIO6/TXD[6]	54	IOz	**General Purpose Input or Output 6.** Available for use as a GPIO pin when the parallel interface is configured for MII or RGMII modes. After reset this pin is high impedance. The function can be changed after reset. See section 6.2.
GPIO7/TXD[7]	55	IOz	**General Purpose Input or Output 7.** Available for use as a GPIO pin when the parallel interface is configured for MII or RGMII modes. After reset this pin is high impedance. The function can be changed after reset. See section 6.2.

Table 5-7. Detailed Pin Descriptions – SGMII/1000BASE-X Serial Interface (7 pins)

Pin Name	PIN #	Type	Pin Description
TDP, TDN	9 8	Odiff	**Transmit Data Output** These pins form a differential CML output for the 1.25Gbaud SGMII transmit signal to a neighboring 1000BASE-X optical module (SFP, etc.) or PHY with SGMII interface. See section 6.6.
TCLKP, TCLKN	6 5	Odiff	**Transmit Clock Output** These pins form a differential CML output for an optional 625MHz clock for the SGMII transmit signal on TDP/TDN. This output is disabled at reset but is enabled by setting CR.TCLK_EN=1. See section 6.6.
RDP, RDN	13 14	Idiff	**Receive Data Input** These pins form a differential input for the 1.25Gbaud SGMII receive signal from a neighboring 1000BASE-X optical module (SFP, etc.) or PHY with SGMII interface. A receive clock signal is not necessary because the device uses a built-in CDR to recover the receive clock from the signal on RDP/RDN. See section 6.6.
ALOS	19	I	**Analog Loss of Signal** This pin receives analog loss-of-signal from a neighboring optical transceiver module. If the optical module does not have an ALOS output, this pin should be connected to DVSS for proper operation. See section 6.6. 0 = ALOS not detected or not required, normal operation 1 = ALOS detected, loss of signal

Table 5-8. Detailed Pin Descriptions – Parallel Interface (25 pins)

Pin Name	PIN #	Type	Pin Description
RXCLK	40	IO	**Receive Clock** In all modes the frequency tolerance is ± 100 ppm. GMII Mode: RXCLK is the 125MHz receive clock. RGMII Modes: RXCLK is the 125MHz (RGMII-1000), 25MHz (RGMII-100) or 2.5MHz (RGMII-10) receive clock (DDR). MII Mode: RXCLK is the 25MHz (100Mbps MII) or 2.5MHz (10Mbps MII) receive clock. In DTE mode (DTE_DCE)=1, RXCLK is an input. In DCE mode (DTE_DCE)=0, RXCLK is an output.
RXD[0] RXD[1] RXD[2] RXD[3] RXD[4] RXD[5] RXD[6] RXD[7]	38 37 36 35 34 33 32 31	IOr	**Receive Data Outputs** During reset these pins are configuration inputs. See section 6.1. After reset they are driven as outputs. GMII Mode: receive_data[7:0] is output on RXD[7:0] on the rising edge of RXCLK. MII Mode, RGMII-10 and RGMII-100 Modes: receive_data[3:0] is output on RXD[3:0] on the rising edge of RXCLK. RXD[7:4] are high impedance. RGMII-1000 Mode: receive_data[3:0] is output on RXD[3:0] on the rising edge of RXCLK, and receive_data[7:4] is output on the falling edge of RXCLK. RXD[7:4] are high impedance.

Pin Name	PIN #	Type	Pin Description
RX_DV	29	IOr	**Receive Data Valid** During reset this pin is a configuration input. See section 6.1. After reset it is driven as an output. MII Mode and GMII Mode: RX_DV is output on the rising edge of RXCLK. RGMII Modes: The RX_CTL signal is output on RX_DV on both edges of RXCLK.
RX_ER	28	IOr	**Receive Error** During reset this pin is a configuration input. See section 6.1. After reset it is driven as an output. MII Mode and GMII Mode: RX_ER is output on the rising edge of RXCLK. RGMII Mode: RX_ER pin is high impedance.
COL	27	IOr	**Collision Detect** During reset this pin is a configuration input. See section 6.1. After reset it is driven as an output. COL indicates that a Tx/Rx collision is occurring. It is meaningful only in half duplex operation. It is asynchronous to any of the clocks. COL is driven low at all times when BMCR.DLB=1 and BMCR.COL_TEST=0. When BMCR.DLB=1 and BMCR.COL_TEST=1, COL behaves as described in the COL_TEST bit description. 1 = Collision is occurring 0 = Collision is not occurring
CRS	26	IOr	**Carrier Sense** During reset this pin is a configuration input. See section 6.1. After reset it is driven as an output. CRS is asserted by the device when either the transmit data path or the receive data path is active. This signal is asynchronous to any of the clocks.
TXCLK	46	IO	**MII Transmit Clock** When TXCLK is an input, frequency tolerance is ±100ppm. MII Mode: TXCLK is the 25MHz (100Mbps MII) or 2.5MHz 10Mbps MII) transmit clock. In DTE mode (DTE_DCE)=1, TXCLK is an input. In DCE mode (DTE_DCE)=0, TXCLK is an output. GMII Mode and RGMII Mode: TXCLK can output a 125MHz clock for use by neighboring components (e.g. a MAC) when GMIICR.TXCLK_EN=1 (or TXCLK=1 at reset).
GTXCLK	66	I	**GMII/RGMII Transmit Clock** In all modes the frequency tolerance is ± 100ppm. GMII Mode: GTXCLK is the 125MHz transmit clock. RGMII Modes: GTXCLK is the 125MHz (RGMII-1000), 25MHz (RGMII-100) or 2.5MHz (RGMII-10) transmit clock (DDR). MII Mode: This pin is not used and should be pulled low. See the TXCLK pin description.

Pin Name	PIN #	Type	Pin Description
TXD[0] TXD[1] TXD[2] TXD[3] TXD[4]/GPIO4 TXD[5]/GPIO5 TXD[6]/GPIO6 TXD[7]/GPIO7	48 49 50 51 52 53 54 55	I I I I IOz IOz IOz IOz	**Transmit Data Inputs** Depending on the parallel MII interface mode, four or eight of these pins are used to accept transmit data from a neighboring component. GMII Mode: The rising edge of GTXCLK latches transmit_data[7:0] from TXD[7:0]. MII Mode, RGMII-10 and RGMII-100 Modes: The rising edge of TXCLK (MII) or GTXCLK (RGMII) latches transmit_data[3:0] from TXD[3:0]. TXD[7:4] become GPIO7 – GPIO4. RGMII-1000 Mode: The rising edge of GTXCLK latches transmit_data[3:0] from TXD[3:0]. The falling edge of GTXCLK latches transmit_data[7:4] from TXD[3:0]. TXD[7:4] become GPIO7 – GPIO4.
TX_EN	57	I	**Transmit Enable** MII Mode and GMII Mode: The rising edge of TXCLK (MII) or GTXCLK (GMII) latches the TX_EN signal from this pin. RGMII Modes: Both edges of GTXCLK latch the TX_CTL signal from this pin.
TX_ER	58	I	**Transmit Error** MII Mode and GMII Mode: The rising edge of TXCLK (MII) or GTXCLK (GMII) latches the TX_ER signal from this pin. RGMII Modes: This pin is not used.

Table 5-9. Detailed Pin Descriptions – Power and Ground Pins (15 pins)

Pin Name	PIN #	Pin Description
DVDD12	30, 56	Digital Power Supply, 1.2V (2 pins)
DVDD33	20, 39, 65	Digital Power Supply, 3.3V
DVSS	47	Return for DVDD12 and DVDD33
RVDD12	16	1.25G Receiver Analog Power Supply, 1.2V
RVDD33	12	1.25G Receiver Analog Power Supply, 3.3V
RVSS	15	Return for RVDD12 and RVDD33
TVDD12	11	1.25G Transmitter Analog Power Supply, 1.2V
TVDD33	7	1.25G Transmitter Analog Power Supply, 3.3V
TVSS	10	Return for TVDD12 and TVDD33
CVDD12	3	TX PLL Analog Power Supply, 1.2V
CVDD33	2	TX PLL Analog Power Supply, 3.3V
CVSS	4	Return for CVDD12 and CVDD33
GVDD12	18	Analog Power Supply, 1.2V
GVSS	1	Return for GVDD12.
Exposed Pad	EP	Exposed pad (die paddle). Connect to ground plane. EP also functions as a heatsink. Solder to the circuit-board ground plane to maximize thermal dissipation.

6. Functional Description

6.1 Pin Configuration During Reset

The MAX24288 initial configuration is determined by pins that are sampled at reset. The values on these pins are used to set the reset values of several register bits.

The pins that are sampled at reset to pin-configure the device are listed described in Table 6-1. During reset these pins are high-impedance inputs and require 10kΩ pullup or pulldown resistors to set pin-configuration values. After reset, the pins can become outputs if configured to do so and operate as configured. There are two pin configuration modes: 15-pin mode and 3-pin mode.

In 15-pin mode (COL=0 during reset, see Table 6-1) all major settings associated with the PCS block are configurable. In addition, the input reference clock frequency on the REFCLK pin is configured during reset using the RXD[3:2] pins.

Table 6-1. Reset Configuration Pins, 15-Pin Mode (COL=0)

Pin	Function	Register Bit Affected	Notes
CRS	Double Date Rate	GMIICR.DDR=CRS	See Table 6-2.
GPO2	10/100 MII: DTE or DCE	10/100 MII: GMIICR.DTE_DCE	0=DCE, 1=DTE (serial interface is configured for SGMII mode, PCSCR.BASEX=0)
	Other: Serial Interface	Other: PCSCR.BASEX	0=SGMII, 1=1000BASE=X
GPO1	GPIO1 Configuration	GPIOCR1.GPIO1_SEL[2]	0=high impedance 1=125MHz from TX PLL
RXD[1:0]	Parallel Interface Speed	GMIICR.SPD[1:0]	See Table 6-2.
RXD[3:2]	REFCLK Frequency	None	00=10MHz, 01=12.8MHz, 10=25MHz, 11=125MHz
RXD[7:4]	MDIO PHYAD[3:0].	Internal MDIO PHYAD register (device address on MDIO bus).	Note: PHYAD[4:0]=11111 enables factory test mode. Do not use.
RX_ER	MDIO PHYAD[4].		
RX_DV	Other: Auto-negotiation	BMCR.AN_EN	0=Disable, 1=Enable
TXCLK	TXCLK Enable	GMIICR.TXCLK_EN	0=high impedance 1=125MHz from TX PLL Ignored in MII mode

Table 6-2. Parallel Interface Configuration

SPD[1]	SPD[0]	Speed	DDR=0	DDR=1
0	0	10Mbps	MII	RGMII-10
0	1	100Mbps	MII	RGMII-100
1	0	1000Mbps	GMII	RGMII-1000
1	1	reserved		

In 3-pin mode (COL=1 during reset, see Table 6-3) the device is configured for a 1000Mbps RGMII or GMII parallel interface. This mode is targeted to the application of connecting an ASIC, FPGA or processor with an RGMII or GMII interface to a switch device with an SGMII interface or to a 1000BASE-X optical interface. In 3-pin mode, the REFCLK pin is configured for 25MHz, the MDIO interface is enabled (with PHY address set to 0x04), the SPI interface is enabled, 1000BASE-X auto-negotiation (or automatic transmission of SGMII control information) is enabled, TXCLK is configured to output a 125MHz clock, and the TCLKP/TCLKN differential pair is disabled. Note: if RX_ER and RXD[7:4] are all high when the device exits reset then the device enters factory test mode; for normal operation set these pins to any other combination of values.

Table 6-3. Reset Configuration Pins, 3-Pin Mode (COL=1)

Pin	Function	Register Bit Affected	Notes
CRS	Double Date Rate	GMIICR.DDR=CRS	0=GMII, 1=RGMII
GPO2	Serial Interface	PCSCR.BASEX	0=SGMII, 1=1000BASE=X

Note: In 3-pin mode register fields are automatically set as follows: REFCLK clock rate to 25MHz, GMIICR.SPD[1:0]=10, MDIO PHYAD is set to 0x04, BMCR.AN_EN=1, GMIICR.TXCLK_EN=1, GPIOCR1=0 and GPIOCR2=0. All other registers are reset to normal defaults listed in the register descriptions.

6.2 General-Purpose I/O

The MAX24288 has two general-purpose output pins, GPO1, GPO2, and seven general-purpose input/output pins, GPIO1 through GPIO7. Each pin can be configured to drive low or high or be in a high-impedance state. Other uses for the GPO and GPIO pins are listed in Table 6-4 through Table 6-6. The GPO and GPIO pins are each configured using a GPxx_SEL field in registers GPIOCR1 or GPIOCR2 with values as indicated in the tables below.

When a GPIO pin is configured as high impedance it can be used as an input. The real-time state of GPIOx can be read from GPIOSR.GPIOx. In addition, a latched status bit GPIOSR.GPIOxL is available for each GPIO pin. This latched status bit is set when the transition specified by GPIOCR2.GPIO13_LSC (for GPIO1 through GPIO3) or by GPIOCR2.GPIO47_LSC (for GPIO4 through GPIO7) occurs on the pin.

Note that GPIO4 through GPIO7 are alternate pin functions to TXD[7:4] and therefore are only available when the parallel MII is configured for MII or RGMII.

Table 6-4. GPO1, GPIO1 and GPIO3 Configuration Options

GPxx_SEL	Description
000	High impedance, not driven, can be used as an input
001	Drive logic 0
010	Drive logic 1
011	Interrupt output, active low. GPO1 drives low and high, GPIO1 and GPIO3 are open-drain.
100	Output 125MHz from the reference clock PLL
101	Output 25MHz or 125MHz from receive clock recovery PLL. Not squelched. Frequency specified by CR.RCFREQ.
110	Output real-time link status, 0=link down, 1=link up
111	Output PEG1 signal from 1588 event generator

Table 6-5. GPO2 and GPIO2 Configuration Options

GPxx_SEL	Description
000	High impedance, not driven, can be used as an input
001	Drive logic 0
010	Drive logic 1
011	Output the PTP_CLKO signal from 1588 time engine
100	Output 125MHz from reference clock PLL
101	Output 25MHz or 125MHz from receive clock recovery PLL. The frequency is specified by CR.RCFREQ. Signal is automatically squelched (driven low) when CR.RCSQL=1 and any of several conditions occur. See section 6.2.1.
110	Output CRS (carrier sense) status
111	Output PEG2 signal

Table 6-6. GPIO4, GPIO5, GPIO6 and GPIO7 Configuration Options

GPxx_SEL	Description
000	High impedance, not driven, can be used as an input
001	Drive logic 0
010	Drive logic 1
011	Output the PTP_CLKO signal from 1588 time engine
100	Output 125MHz from reference clock PLL
101	Output 25MHz or 125MHz from receive clock recovery PLL. The frequency is specified by CR.RCFREQ. Signal is automatically squelched (driven low) when CR.RCSQL=1 and any of several conditions occur. See section 6.2.1.
110	Output PEG1 signal
111	Output PEG2 signal

6.2.1 Receive Recovered Clock Squelch Criteria

A 25MHz or 125MHz clock from the receive clock recovery PLL can be output on any of GPO2, GPIO2 and GPIO4-7. When CR.RCSQL=1, this clock is squelched (driven low) when any of the following conditions occur:

- IR.ALOS=1 (analog loss-of-signal occurred)
- IR.RLOS=1 (CDR loss-of-signal occurred))
- IR.RLOL=1 (CDR PLL loss-of-lock occurred)
- IR.LINK_ST=0 (auto-negotiation link down occurred, latched low)

Since each of these criteria is a latched status bit, the output clock signal remains squelched until all of these latched status bits go inactive (as described in section 7.2).

6.3 Reset, Power Down and Processor Interrupt

6.3.1 Reset

The following reset functions are available in the device:

1. Hardware reset pin (RST_N): This pin asynchronously resets all logic, state machines and registers in the device except the JTAG logic. When the RST_N pin is low, all internal registers are reset to their default values. Pin states are sampled and used to set the default values of several register fields as described in section 6.1. RST_N should be asserted for at least 100µs.

2. Global reset bit, GPIOCR1.RST: Setting this bit is equivalent to asserting the RST_N pin. This bit is self-clearing.

3. (MDIO interface only) Datapath reset bit, BMCR.DP_RST. This bit resets the entire datapath from parallel MII interface through PCS encoder and decoder including the packet classifier and modifier blocks. It also resets the deserializer and transmit and receive start-of-packet detectors. It does not reset any registers, GPIO logic, the TX PLL or any block reset by PTPCR1.TE_RST. The DP_RST bit is self-clearing.

4. Time engine reset bit, PTPCR1.TE_RST. This bit resets the logic of the 1588 time engine, output clock generator, programmable event generators, timestampers and GPIO. It does not reset any registers, GPIO logic, the TX PLL or any block reset by BMCR.DP_RST. The TE_RST bit is self-clearing.

5. JTAG reset pin JTRST_N. This pin resets the JTAG logic. See section 8 for details about JTAG operation.

TE_RST does not affect the datapath or packet traffic.

6.3.2 Power Down

When sections of the MAX24288 are not used, they can be powered down to reduce power consumption.

The transmit serializer and the TDP/TDN and TCLKP/TCLN output drivers can be powered down by setting PTPCR1.TX_PWDN=1. In this mode, the output drivers are placed in a high-impedance state, and the pins are pulled up to 3.3V by their internal 50Ω termination resistors. See section 6.6.

The RDP/RDN inputs, the clock and data recovery PLL, and the de-serializer can be powered down by setting PTPCR1.RX_PWDN=1.

The parallel MII (section 6.7), the PCS encoder and decoder and all other parallel datapath logic, both receive and transmit, except the packet classifiers and packet modifiers can be powered down by setting PTPCR1.DP_PWDN=1.

The packet classifiers (section 6.13.6) and packet modifiers (6.13.7) can be powered down by setting PTPCR1.PKT_PWDN=1.

The time engine (section 6.13.1), output clock generator (6.13.2), PEGs (6.13.3) and timestampers (6.13.4) can be disabled by setting PTPCR1.TE_PWDN=1.

Finally, the TX PLL (section 6.10.2) can be powered down and bypassed by setting PTPCR1.PLL_PWDN=1. Because the serializer, transmit driver, receive CDR and deserializer do not get the high-speed clocks they need when the TX PLL is disabled, those blocks must be disabled when PLL_PWDN=1 by setting PTPCR1.TX_PWDN=1 and PTPCR1.RX_PWDN=1. In addition, if the frequency of the REFCLK signal is less than 125MHz, all internal logic is clocked at a slower rate, including the MDIO and SPI interfaces. The maximum clock rates for MDIO and SPI are reduced by a factor of (REFCLK_freq / 125MHz).

In addition, when the TX PLL is powered down, the time engine accumulator (Figure 6-13) is clocked directly from the REFCLK signal. Therefore, the uncertainty of timestamping and PEG edge placement is half a REFCLK cycle (vs. ~1ns when using the TX PLL).

Deasserting a PWDN bit causes the affected circuitry to be reset as described in section 6.3.1.

6.3.3 Processor Interrupts

Any of pins GPO1, GPIO1 and GPIO3 can be configured as an active low interrupt output by setting the appropriate field in GPIOCR1 to 011. GPO1 drives high and low while GPIO1 and GPIO3 are open-drain and require pullup resistors.

Status bits than can cause an interrupt are located in the IR and PTP_IR registers. The corresponding interrupt enable bits are located in the IR and PTP_IE registers. Both the PAGESEL register and the PTP_IR register have top-level IR AND PTP_IR status bits to indicate which registers have active interrupt sources. The PAGESEL register is available on all pages through the MDIO interface, allowing the interrupt routine to read the register without changing the MDIO page.

6.4 SPI - Serial Processor Interface

The MAX24288's SPI interface consists of four signals: serial clock (SCLK), serial data in (SDI), serial data out (SDO), and chip select (CS_N, active low). SPI is a widely-used master/slave bus protocol that allows a master device and one or more slave devices to communicate using only four wires. The MAX24288 is always a slave device. Masters are typically microprocessors, ASICs or FPGAs. Data transfers are always initiated by the master device, which also generates the SCLK signal. The MAX24288 receives serial data on the SDI pin and transmits serial data on the SDO pin. SDO is high impedance except when the MAX24288 is transmitting data to the bus master. At the maximum SPI clock frequency of 25MHz each non-burst read or write access takes approximately 1µs.

The SPI interface is enabled at reset but can be disabled by setting the PAGESEL.SPI_DIS register bit. (Note: the PAGESEL register can only be accessed using the MDIO interface). When the SPI bus is enabled, all of the IEEE1588 registers are mapped to the SPI register space as shown in Table 7-2. When the SPI bus is disabled, all of the IEEE1588 registers are mapped to pages 1, 2 and 3 of the MDIO register space as shown in Table 7-2. The MAX24288 accepts SPI commands with a 6-bit address field and therefore its SPI register space is 0 to 0x3F. Registers are 16 bits wide.

Clock Polarity and Phase. SCLK polarity and phase can be changed using the CPOL and CPHA bits of the PAGESEL register. The CPOL bit defines the polarity of SCLK. When CPOL=0, SCLK is normally low and pulses high during bus transactions. When CPOL = 1, SCLK is normally high and pulses low during bus transactions. The CPHA bit sets the phase (active edge) of SCLK. When CPHA = 0, data is latched in on SDI on the leading edge of the SCLK pulse and updated on SDO on the trailing edge. When CPHA = 1, data is latched in on SDI on the trailing edge of the SCLK pulse and updated on SDO on the following leading edge. SCLK does not have to toggle between accesses, i.e., when CS_N is high. See Figure 6-1.

Device Selection. Normally each SPI device has its own chip-select line. The MAX24288 is selected when its CS_N pin is low. The MAX24288 also supports an alternate device selection method where multiple MAX24288 devices share the same chip-select line. See *Design Option: Shared Chip Select* below for details. When CS_N is de-asserted the SDO signal is high impedance, and any incomplete transfer cycle is aborted. This behavior is asynchronous to the SCLK signal. The CS_N signal can stay asserted for the duration of multiple read and write cycles. The transition of CS_N from de-asserted to asserted defines the start of a cycle or multiple cycles.

Control Word. After CS_N is pulled low, the bus master transmits the control word during the first eight SCLK cycles. By default the 8-bit control word is sent with address MSb first: R/W A5 A4 A3 A2 A1 A0 BURST. When PAGESEL.SPISWAP=1 the control word is sent with address LSb first: R/W A0 A1 A2 A3 A4 A5 BURST where A[5:0] is the register address, R/W is the data direction bit (1=read, 0=write), and BURST is the burst bit (1=burst access, 0=single-word access). In the discussion that follows, a control word with R/W = 1 is a read control word, while a control word with R/W = 0 is a write control word.

Data Word. By default, 16-bit data words are sent MSb first. When PAGESEL.SPISWAP=1 data words are sent LSb first.

Single-Word Writes. See Figure 6-2. After CS_N goes low, the bus master transmits a write control word with BURST = 0 followed by the 16-bit word to be written. The data word is transferred to the register after the last data bit is sampled. If CS_N stays asserted the next word must be a control word.

Single-Word Reads. See Figure 6-2. After CS_N goes low, the bus master transmits a read control word with BURST = 0. The MAX24288 then responds with the requested 16-bit data word. When CS_N stays asserted the next word must be a control word.

Burst Writes. See Figure 6-2. After CS_N goes low, the bus master transmits a write control word with BURST = 1 followed by the first 16-bit data word to be written. The MAX24288 receives the first data word on SDI, writes it to the specified register, increments its internal address register, and prepares to receive the next data word. If the master continues to transmit, the MAX24288 continues to write the data received and increment its address counter. After the address counter reaches 1Fh it rolls over to address 00h and continues to increment. The bus master must terminate the transaction by pulling CS_N high after the last data word.

Burst Reads. See Figure 6-2. After CS_N goes low, the bus master transmits a read control word with BURST = 1. The MAX24288 then responds with the requested data word on SDO, increments its address counter, and prefetches the next data word. If the bus master continues to demand data, the MAX24288 continues to provide the data on SDO, increment its address counter, and prefetch the following word. After the address counter reaches 1Fh it rolls over to address 00h and continues to increment. The bus master must terminate the transaction by pulling CS_N high after the last data word. NOTE: The prefetch mentioned above can have the unintended effect of clearing latched status bits. Care should be taken to not terminate a burst read on the address prior to a register with latched status bits.

Early Termination of Bus Transactions. The bus master can terminate SPI bus transactions at any time by pulling CS_N high. In response to early terminations, the MAX24288 resets its SPI interface logic and waits for the

start of the next transaction. If a write transaction is terminated prior to the SCLK edge that latches the LSb of a data word, the word is not written.

Design Option: Wiring SDI and SDO Together. Because communication between the bus master and the MAX24288 is half-duplex, the SDI and SDO pins can be wired together externally to reduce wire count. To support this option, the bus master must not drive the SDI/SDO line when the MAX24288 is driving SDI/SDO. When SDI and SDO are tied together the CS_N signal must be de-asserted between commands.

Design Option: Shared Chip Select. Multiple MAX24288 devices can be configured to use a shared chip-select pin. In this mode of operation, each MAX24288 uses its MDIO PHY address (latched at reset as described in section 6.1) as its unique SPI device address. To read or write just one of the MAX24288 devices, the SPI bus master asserts the common CS_N pin and writes the device address and ENABLE=1 to the PHY_MATCH register. Each MAX24288 device performs this write to its PHY_MATCH register simultaneously and compares its device address to the PHY_MATCH address. When PHY_MATCH.ENABLE=1, only the MAX24288 whose device address matches the PHY_MATCH address responds to read and write commands on the SPI bus.

To access all MAX24288 devices simultaneously, the SPI bus master writes ENABLE=0 to the PHY_MATCH register. This causes all MAX24288 devices on the bus to respond to read and write commands on the SPI bus. Reading all devices at the same time is not useful, but writing all devices at the same time can be useful when all need the same configuration settings.

AC Timing. See Table 9-21 and Figure 9-4 for AC timing specifications for the SPI interface.

Figure 6-1. SPI Clock Polarity and Phase Options

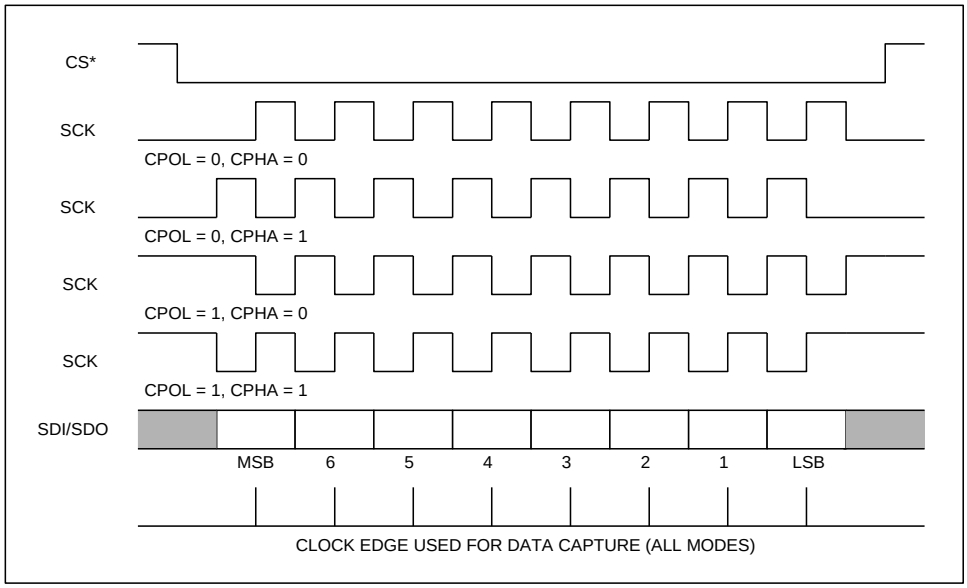
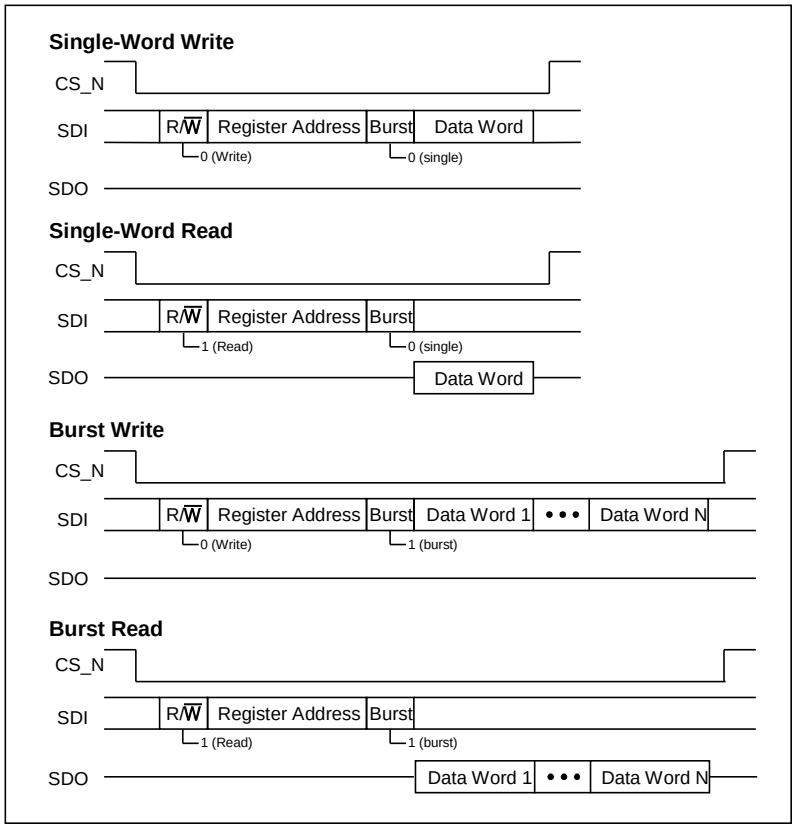


Figure 6-2. SPI Bus Transactions



6.5 MDIO Interface

6.5.1 MDIO Overview

The MAX24288's MDIO interface is compliant to IEEE 802.3 clause 22. MAX24288 always behaves as a PHY on the MDIO bus. Because MAX24288 is not a complete PHY but rather a device that sits between a MAC and a PHY, it implements only a subset of the registers and register fields specified in 802.3 clause 22 as shown in the table below.

MDIO Address	802.3 Name	MAX24288 Name
0	Control	BMCR
1	Status	BMSR
2, 3	PHY Identifier	ID1, ID2
4	Auto-Negotiation Advertisement	AN_ADV
5	Auto-Negotiation Link Partner Base Page Ability	AN_RX
6	Auto-Negotiation Expansion	AN_EXP
15	Extended Status	EXT_STAT

The MDIO consists of a bidirectional, half-duplex serial data signal (MDIO) and a ≤12.5MHz clock signal (MDC) driven by a bus master, usually a MAC. The format of management frames transmitted over the MDIO interface is shown below (see IEEE 802.3 clause 22.2.4.5 for more information). MDIO DC electrical characteristics are listed in section 9.2.1. AC electrical characteristics are listed in section 9.3.6. The MAX24288's MDIO slave state machine is shown in Figure 6-3.

	Management Frame Fields							
	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
READ Command	32 ‘1’s	01	10	AAAAA	RRRRR	Z0	16-bit	Z
WRITE Command	32 ‘1’s	01	01	AAAAA	RRRRR	10	16-bit	Z

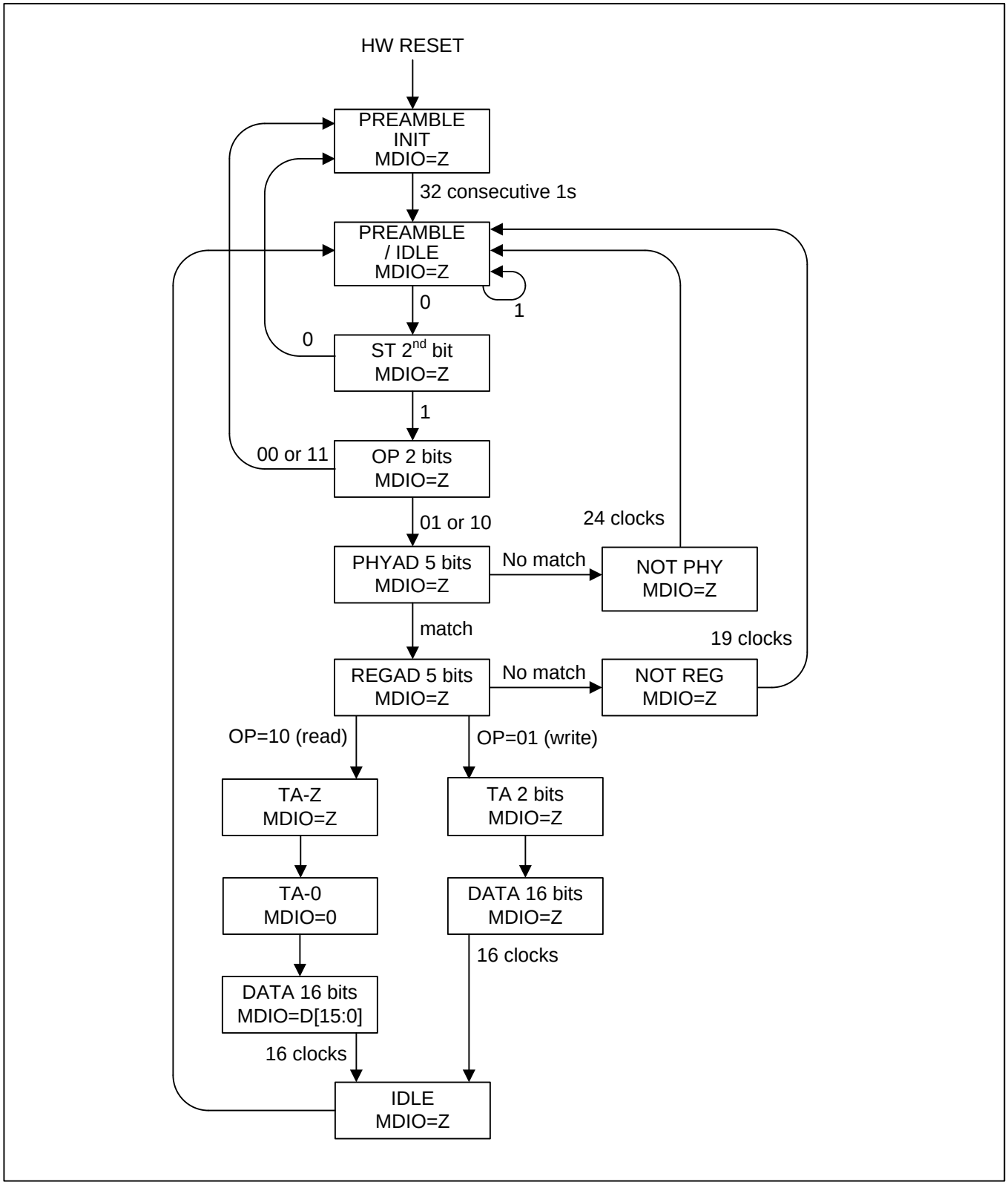
The transmission and reception bit order is MSB first for the PHYAD, REGAD and DATA fields

MAX24288 supports preamble suppression. This allows quicker bursts of read and write transfers to occur by shortening the minimum transfer cycle time from 65 clock periods to 33 clock periods. There must be at least a 32-bit preamble on the first transfer after reset, but on subsequent transfers the preamble can be suppressed or shortened. When the preamble is completely suppressed the 0 in the ST symbol follows the single IDLE Z, which is one clock period duration.

Like any MDIO slave, MAX24288 only performs the read or write operation specified if the PHYAD bits of the MDIO command match the device PHY address. The device PHY address is latched during device reset from the RXD[7:4] and RX_ER pins. See section 6.1.

The MAX24288 does not support the 802.3 clause 45 MDIO extensions. Management frames with ST bits other than 01 or OP bits other than 01 or 10 are ignored and put the device in a state where it ignores the MDIO traffic until it sees a full preamble (32 ones). If Clause 45 ICs and the MAX24288 are connected to the same MDIO management interface, the station management entity must put a full preamble on the bus after communicating with clause 45 ICs before communicating with the MAX24288.

Figure 6-3. MDIO Slave State Machine



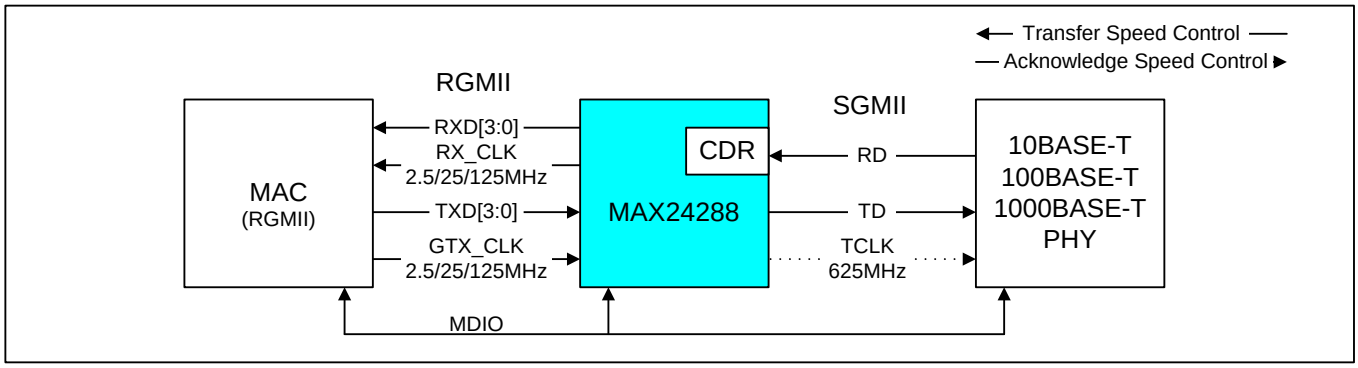
6.5.2 Examples of MAX24288 and PHY Management Using MDIO

The MDIO interface can control both the Ethernet PHY functions and the IEEE1588 functions of the MAX24288. The Ethernet functions (PCS, PMA, auto-negotiation, parallel and serial MIIs) can only be controlled using the MDIO interface. The IEEE1588 functions can be controlled using either the MDIO interface or the SPI interface (section 6.4). The PAGESEL.SPI_DIS bit selects which interface controls the IEEE1588 functions.

The MDIO interface is typically provided by the MAC function within a neighboring processor, ASIC or FPGA component. It can be used to configure the registers in the MAX24288 and/or the registers in a PHY or switch chip connected to the MAX24288 via the SGMII interface.

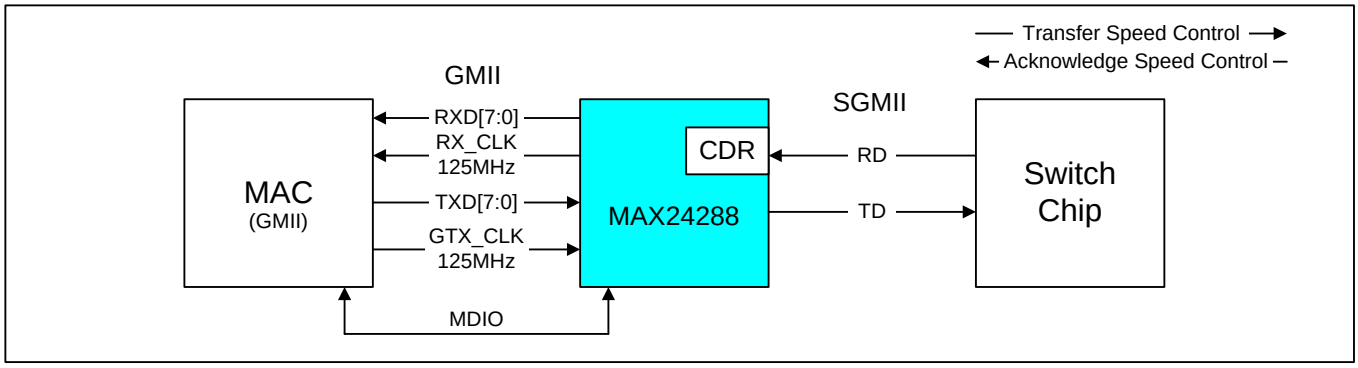
Case 1 in Figure 6-4 shows a typical application where the MAX24288 connects a MAC with a 3-speed RGMII interface to a 3-speed PHY with an SGMII interface. Through the MDIO interface, system software configures the MAX24288 and optionally the PHY. (The PHY may not need to be configured if it is operating in a hardware-only auto-negotiation 1000BASE-T mode). After initial configuration and after the PHY auto-negotiates link details with its 1000BASE-T link partner, the speed and mode are transferred to the MAX24288 over the SGMII interface as specified in the SGMII specification and are available in the MAX24288 AN_RX register. The processor reads this information and configures the MAC and the MAX24288 to match the mode the PHY is in.

Figure 6-4. Management Information Flow Options, Case 1,Tri-Mode PHY



Case 2 in Figure 6-5 shows a typical application where the MAX24288 connects a MAC with a GMII interface to an SGMII switch chip. Through the MDIO interface, system software configures the MAX24288 to match the MAC mode and writes the MAX24288's AN_ADV register to also match the MAC mode. The MAX24288 then transfers the speed and mode over the SGMII interface as specified in the SGMII specification. The switch chip receives this information and configures its port to match.

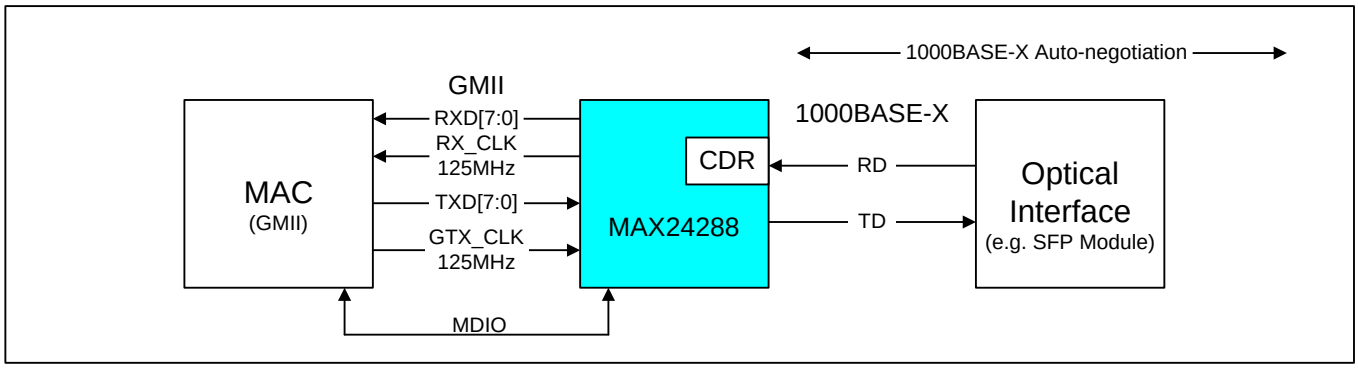
Figure 6-5. Management Information Flow Options, Case 2, SGMII Switch Chip



Case 3 in Figure 6-6 shows a typical application where the MAX24288 connects a MAC with a GMII interface to an optical interface. In this case the MAX24288 provides the 1000BASE-X PCS and PMA functions for the optical interface. Through the MDIO interface, system software configures the MAX24288 to match the MAC mode, both

of which need to be 1000 Mbps speed. The MAX24288 then auto-negotiates with its link partner. This 1000BASE-X auto-negotiation is primarily to establish the pause functionality of the link. The MAX24288's auto-negotiation support is described in section 6.8.

Figure 6-6. Management Information Flow Options, Case 3, 1000BASE-X Interface



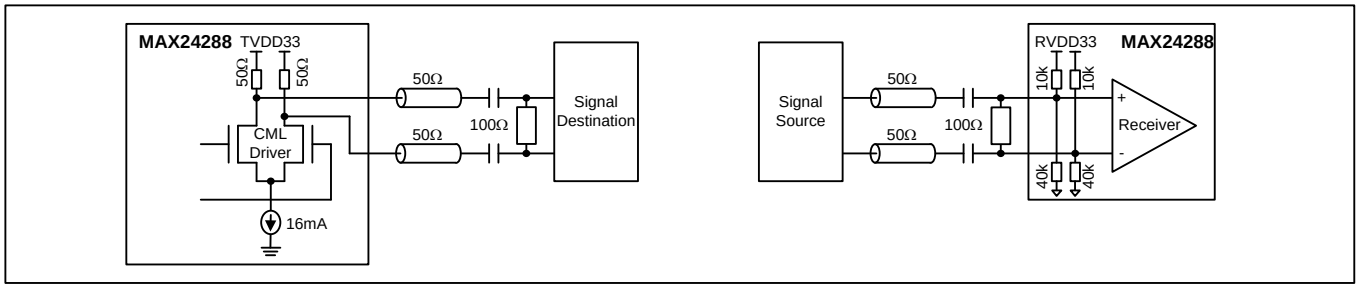
6.6 Serial Interface – 1000BASE-X or SGMII

The high-speed serial interface is compatible with the specification of the 1000BASE-CX PMD service interface TP1 as defined in 802.3 clause 39. It is also compatible with the specification of the SGMII interface and can connect to optical PMD modules in 1000BASE-SX/LX interfaces.

On this interface the MAX24288 transmits a 1250Mbaud differential signal on the TDP/TDN output pins. DDR clocking is used, and the transmit interface outputs a 625MHz differential clock signal on the TCLKP/TCLKN output pins. In the receive direction the clock and data recovery (CDR) block recovers both clock and data from the incoming 1250Mbaud signal on RDP/RDN. A separate receive clock signal is not needed.

Signal Format, Coupling, Termination. The serial interface passes data at 1.25 Gbaud using a CML differential output and an any-format differential input. The CML TDP/TDN outputs have internal 50Ω pullup resistors to TVDD33. The differential input RDP/RDN does not have internal termination, and an external 100Ω termination resistor between RDP and RDN is recommended. The high-speed serial interface pins are typically connected with neighboring components using AC coupling as shown in Figure 6-7.

Figure 6-7. Recommended External Components for High-Speed Serial Interface



Receive Loss-of-Signal. The device's receiver logic has an ALOS input pin through which analog loss-of-signal (ALOS) can be received from a neighboring optical transceiver module, if the high-speed serial signal is transmitted/received optically. The IR.ALOS bit is set when the ALOS pin goes high. ALOS can cause an interrupt if enabled by IR.ALOS_IE.

In addition, the clock-and-data recovery block (CDR) indicates loss-of-signal when it does not detect any transitions in 24 bit times. The IR.RLOS latched status bit is set when the CDR indicates loss-of-signal. RLOS can cause an interrupt if enabled by IR.RLOS_IE.

Receive Loss-of-Lock. The receive clock PLL in the CDR locks to the recovered clock from the RDP/RDN pins and produces several receive-side clock signals. If the receive clock PLL loses lock, it sets IR.RLOL, which can cause an interrupt if enabled by IR.RLOL_IE.

Transmit Clock. The TCLKP/TCLKN differential output can be enabled and disabled using CR.TCLK_EN. Disabled means the output drivers for TCLKP and TCLKN are disabled (high impedance) and the internal 50Ω termination resistors pull both TCLKP and TCLKN up to 3.3V.

Transmit Power Down. The serial interface transmitter is powered down by setting PTPCR1.TX_PWDN=1. See section 6.3.2.

DC Electrical Characteristics. See section 9.2.2.

AC Electrical Characteristics. See section 9.3.3.

6.7 Parallel Interface – GMII, RGMII, MII

The parallel interface can be configured as GMII or MII compliant to IEEE 802.3 clauses 35 and 22, respectively. It can also be configured as reduced pin count RGMII compliant to the HP document RGMII Version 1.3 12/10/2000. A summary of the parallel interface modes is shown in Table 6-7 below.

Table 6-7. Parallel Interface Modes

Mode	Baud Rate, Mbps	Data Transfer Per Cycle, # of Wires Per Direction	Transmit Clock	Receive Clock	Full Duplex	Half Duplex
GMII	1000	8-bit data, 8 wires	Input, 125MHz	Output, 125MHz	Yes	No
RGMII-1000	1000	8-bit data, 4 wires, DDR	Input, 125MHz	Output, 125MHz	Yes	No
RGMII-100	100	4-bit data, 4 wires	Input, 25MHz	Output 25MHz	Yes	Yes
RGMII-10	10	4-bit data, 4 wires	Input, 2.5MHz	Output, 2.5MHz	Yes	Yes
MII-100 DCE	100	4-bit data, 4 wires	Output, 25MHz	Output, 25MHz	Yes	Yes
MII-10 DCE	10	4-bit data, 4 wires	Output, 2.5MHz	Output, 2.5MHz	Yes	Yes
MII-100 DTE	100	4-bit data, 4 wires	Input, 25MHz	Input, 25MHz	Yes	Yes
MII-10 DTE	10	4-bit data, 4 wires	Input ,2.5MHz	Input, 2.5MHz	Yes	Yes

The parallel interface mode is controlled by GMIICR.SPD[1:0]. MII options are specified by GMIICR.DTE_DCE.

6.7.1 GMII Mode

The MAX24288's GMII interface is compliant to IEEE 802.3 clause 35 but only operates full duplex. Half duplex operation is not supported, and the TX_ER pin is ignored. The PHY therefore does not receive the following from the MAC: carrier extend, carrier extend error, and transmit error propagation as described in 802.3 section 35.2.1.6, section 35.2.2.5 and Table 35-1. These features are not needed for full duplex operation.

The parallel interface can be configured for GMII mode using software configuration or pin configuration at reset. For pin configuration (see section 6.1) one of the following combinations of pin states must be present during device reset:

- COL=0, RXD[1:0]=10, CRS=0
- COL=1, CRS=0

For software configuration, the following register fields must be set: GMIICR.SPD[1:0]=10 and GMIICR.DDR=0.

See IEEE 802.3 clause 35 for functional timing diagrams. GMII DC electrical characteristics are listed in section 9.2.1. AC electrical characteristics are listed in section 9.3.4 and 9.3.5.

Table 6-8. GMII Parallel Bus Pin Naming

Pin Name	802.3 Pin Name	Function
RXCLK	RX_CLK	Receive 125MHz clock output
RXD[7:0]	RXD[7:0]	Receive data output
RX_DV	RX_DV	Receive data valid output
RX_ER	RX_ER	Receive data error output
CRS	CRS	Receive carrier sense
COL	COL	Receive collision (held low in GMII mode)
TXCLK	---	Outputs 125MHz from the TX PLL for use by the MAC when GMIICR.TXCLK_EN=1.
GTXCLK	GTX_CLK	Transmit 125MHz clock input
TXD[7:0]	TXD[7:0]	Transmit data input
TX_EN	TX_EN	Transmit data enable input
TX_ER	TX_ER	Transmit data error input (not used - ignored)

6.7.2 RGMII Mode

The RGMII interface has three modes of operation to support three Ethernet speeds: 10, 100 and 1000Mbps. This document refers to these three modes as RGMII-1000 for 1000Mbps operation, RGMII-100 for 100Mbps operation and RGMII-10 for 10Mbps operation. RGMII is specified to support speed changes among the three rates as needed. The RGMII specification document can be downloaded from http://www.hp.com/rnd/pdfs/RGMIIv1_3.pdf or can be found by a web search for "RGMII 1.3".

On the receive RGMII interface the MAX24288 does not report in-band status for link state, clock speed and duplex during normal inter-frame. This status indication is defined as optional in the RGMII specification.

The parallel interface can be configured for RGMII modes using software configuration or pin configuration at reset. For pin configuration (see section 6.1) one of the following combinations of pin states must be present during device reset:

- COL=0, RXD[1:0]=xx, CRS=1 (xx: 00=RGMII-10, 01=RGMII-100, 10=RGMII-1000)
- COL=1, CRS=1 (RGMII-1000 only)

For software configuration, the following register fields must be set: GMIICR.SPD[1:0]=xx and GMIICR.DDR=1 (where xx values are the same as shown above for RXD[1:0]).

Table 6-9. RGMII Parallel Bus Pin Naming

Pin Name	RGMII Pin Name	Function
RXCLK	RXC	Receive 125MHz clock output
RXD[3:0]	RD[3:0]	Receive data bits 3 to 0 and 7 to 4 output
RX_DV	RX_CTL	Receive data valid output
RX_ER	---	Not used
CRS	---	Outputs carrier sense signal
COL	---	Outputs collision signal
TXCLK	---	Outputs 125MHz from the TX PLL for use by the MAC when GMIICR.TXCLK_EN=1.
GTXCLK	TXC	Transmit 125MHz clock input
TXD[3:0]	TD[3:0]	Transmit data bits 3 to 0 and 7 to 4 input
TX_EN	TX_CTL	Transmit data enable input
TX_ER	---	Not used

6.7.2.1 RGMII-1000 Mode

RGMII-1000 is a reduced pin count alternative to the GMII interface. Pin count is reduced by sampling and updating data and control signals on both clock edges. For data, only four data lines are used, as shown in Table 6-9. Data bits 3:0 are latched on the rising edge of the clock, and data bits 7:4 are latched on the falling edge. The transmit control signals TX_EN and TX_ER are multiplexed onto a single TX_CTL signal. TX_EN is latched on the rising edge of the transmit clock TXC while a modified TX_ER signal is latched on the falling edge of TXC. The receive control signals RX_DV and RX_ER are multiplexed onto a single RX_CTL signal. RX_DV is latched on the rising edge of the receive clock RXC while a modified RX_ER signal is latched on the falling edge of RXC.

The modified TX_ER signal = (GMII TX_ER) XOR (GMII TX_EN). The modified RX_ER signal = (GMII_RX_ER) XOR (GMII_RX_DV). These modifications are done to reduce power by eliminating control signal toggling at 125MHz during normal data transmission and during idle.

On the transmit side of the parallel interface the MAX24288 ignores TX_ER, as it does in all 1000Mbps interface modes, since it only operates full-duplex at 1000Mbps. Therefore the 0,1 TX_CTL encoding is interpreted as 0,0 (idle, normal interframe). Similarly, the 1,0 TX_CTL encoding is interpreted as 1,1 (normal data transmission).

See the RGMII v1.3 specification document for functional timing diagrams. DC electrical characteristics are listed in section 9.2.1. AC electrical characteristics are listed in section 9.3.4 and 9.3.5.

6.7.2.2 RGMII-10 and RGMII-100 Modes

The RGMII interface can be used to convey 10 and 100Mbps Ethernet data. The theory of operation at these lower speeds is similar to RGMII-1000 mode as described above but with a 2.5MHz clock for 10Mbps operation, a 25MHz clock for 100Mbps operation, and data latched and updated only on the rising edge of the clock. The RXD[3:0] pins maintain their values during the negative edge of RXCLK. The control signals are conveyed on both clock edges exactly as described for RGMII-1000. See the RGMII v1.3 specification document for functional timing diagrams. DC electrical characteristics are listed in section 9.2.1. AC characteristics are listed in section 9.3.4 and 9.3.5.

6.7.2.3 Clocks

The RXCLK clock output is 125, 25 or 2.5MHz, depending on RGMII mode. It is derived from the recovered clock from the receive serial data.

The GTXCLK clock input must be 125, 25 or 2.5MHz ±100 ppm. The GTXCLK clock is not used as the source of the serial data transmit clock.

6.7.3 MII Mode

The MAX24288's MII interface is compliant to IEEE 802.3 clause 22 except that TX_ER is ignored (and therefore the MAX24288 doesn't receive transmit error propagation from the MAC).

The parallel interface can be configured for MII mode using software configuration or pin configuration at reset. For pin configuration (see section 6.1) device pins must be set as follows during device reset: COL=0, RXD[1:0]=0x, CRS=0 (x=0 for 10Mbps, x=1 for 100Mbps). For software configuration, the following register fields must be set: GMIICR.SPD[1:0]=0x and GMIICR.DDR=0.

Since the MAX24288 can be used in a variety of applications, it can be configured to source the TXCLK and RXCLK as a PHY normally does or to accept TXCLK and RXCLK from a neighboring component. The former case is called DCE mode; the latter is called DTE mode. DTE/DCE selection is controlled by the GMIICR.DTE_DCE register bit.

In DTE mode the MAX24288 is configured for operation on the MAC side of the MII. Both TXCLK and RXCLK are inputs at 2.5MHz (10Mbps MII) or 25MHz (100Mbps MII) ±100 ppm. TXCLK is not used as the serial data transmit clock (which is derived from the REFCLK input instead).

In DCE mode the MAX24288 is configured for operation on the PHY side of the MII. Both TXCLK and RXCLK are outputs at 2.5MHz or 25MHz. The TXCLK output clock is derived from the REFCLK input. The RXCLK output clock is derived from the recovered clock from the receive serial data when a receive signal is present or from REFCLK when no receive signal is present.

See 802.3 clause 22 for functional timing diagrams. MII DC electrical characteristics are listed in section 9.2.1. AC electrical characteristics are listed in section 9.3.4 and 9.3.5.

On the transmit MII interface the MAX24288 requires that the preamble including the SFD must be an even number of nibbles (i.e. number of nibbles divided by 2 is an integer). On the receive MII interface the MAX24288 always outputs an even number of nibbles of preamble.

Table 6-10. MII Parallel Bus Pin Naming

Pin Name	802.3 Pin Name	Function
RXCLK	RX_CLK	Receive 2.5 or 25MHz clock, can be input or output
RXD[3:0]	RXD[3L0]	Receive data nibble output
RX_DV	RX_DV	Receive data valid output
RX_ER	RX_ER	Receive data error output
CRS	CRS	Receive carrier sense
COL	COL	Receive collision
TXCLK	TX_CLK	Transmit 2.5 or 25MHz clock, can be input or output
GTXCLK	---	Not used
TXD[3:0]	TXD[3:0]	Transmit data nibble input
TX_EN	TX_EN	Transmit data enable input
TX_ER	TX_ER	Transmit data error input (not used - ignored)

6.8 Auto-Negotiation (AN)

In the MAX24288 the auto-negotiation mechanism described in IEEE 802.3 Clause 37 is used for auto-negotiation between IEEE 802.3 1000BASE-X link partners as well as the transfer of SGMII PHY status to a neighboring MAC as described in the Cisco Serial-SGMII document ENG-46158. The 802.3 1000BASE-X next page functionality is not supported. The PCSCR.BASEX register bit controls the link timer time-out mode of the auto-negotiation protocol.

The auto-negotiation mechanism between link partners uses a special code set that passes a 16-bit value called tx_Config_Reg[15:0] as defined in IEEE 802.3. The auto-negotiation transfer starts when BMCR.AN_START is set (self clearing) and BMCR.AN_EN is set. The tx_Config_Reg value is continuously sent with the ACK bit (bit 14) clear and the other bits sourced from the AN_ADV register. When the device receives a non-zero rx_Config_Reg value it sets the ACK bit in the tx_Config_Reg, saves the rx_Config_Reg bits to the AN_RX register, and sets the AN_EXP.PAGE and IR.PAGE bits to tell system software that a new page has arrived. The tx_Config_Reg transmission stops when the link timer times out after 10ms in 1000BASE-X mode or after 1.6ms in SGMII mode.

System software must complete the auto-negotiation function by processing the AN_RX register and configuring the hardware appropriately.

The fields of the auto-negotiation registers AN_ADV and AN_RX have different meanings when used in 1000BASE-X or SGMII mode. See section 6.8.1 for 1000BASE-X and section 6.8.2. for SGMII.

By default, an internal watchdog timer monitors the auto-negotiation process. If the link is not up within 5 seconds after auto-negotiation was started, the watchdog timer restarts the auto-negotiation. This watchdog timer can be disabled by setting PCSCR.WD_DIS=1.

6.8.1 1000BASE-X Auto-Negotiation

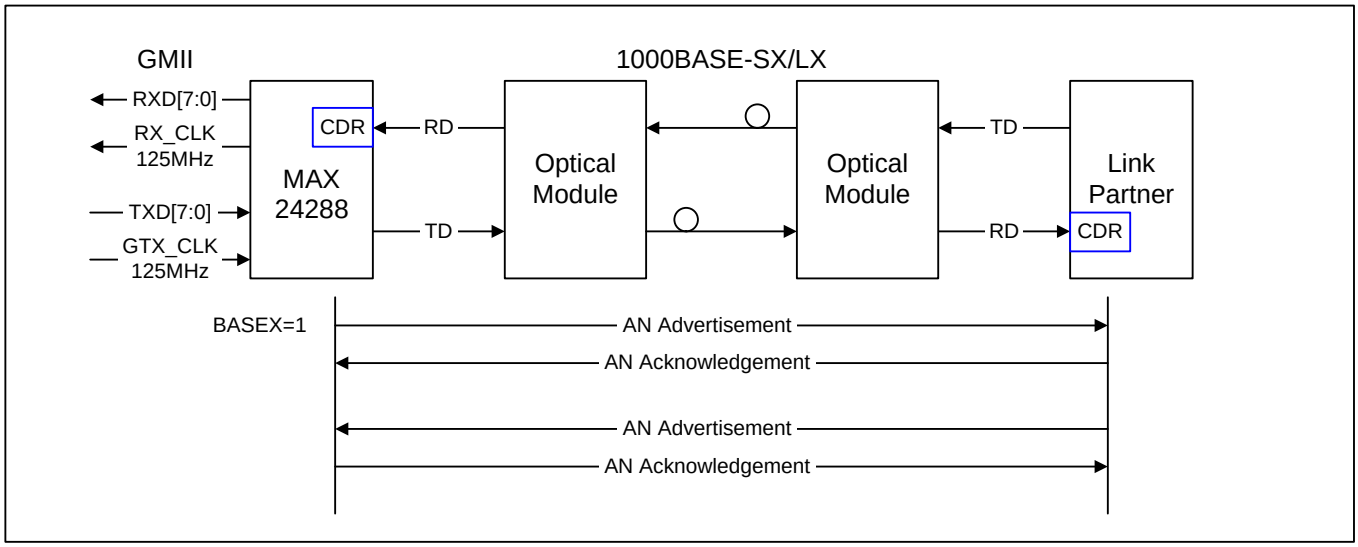
In 1000BASE-X auto-negotiation, two link partners send their specific capabilities to each other. Each link partner compares the capabilities that it advertises in its AN_ADV register against its link partner's advertised abilities, which are stored in the AN_RX register when received. Each link partner uses the same arbitration algorithm specified in IEEE 802.3 clause 37.2 to determine how it should configure its hardware, and, therefore, the two link partners' configurations should match. However, if there is no overlap of capabilities between the link partners, the RF bits (Remote Fault, see Table 6-11) are set to 11, and the auto-negotiation process is started again. An external processor configures the advertised abilities, reads the link partner’s abilities, performs the arbitration algorithm, and sets the RF bits as needed.

The MAX24288 AN block implements the auto-negotiation state machine shown in IEEE 802.3 Figure 37-6. It also generates an internal link-down status signal whenever the AN state machine is in any state other than AN_DISABLE_LINK_OK or LINK_OK. This link-down signal is used to clear the active-low BMSR.LINK_ST bit and is used to squelch output clock signals on GPIO pins when CR.RCSQL=1.

System software can configure the MAX24288 for 1000BASE-X auto-negotiation by setting PCSCR.BASEX=1 and BMCR.AN_EN=1.

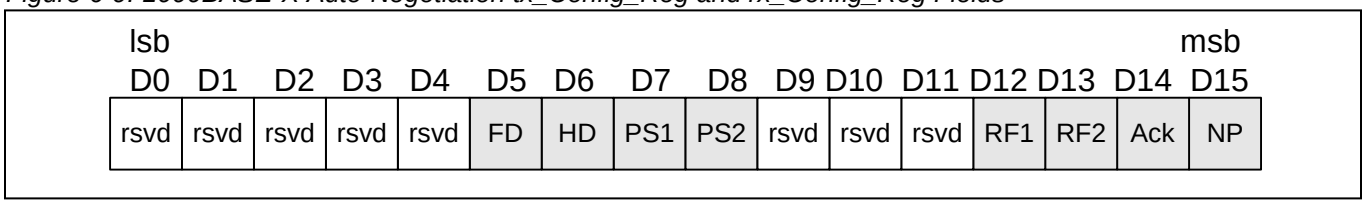
The MAX24288 can also be configured for 1000BASE-X auto-negotiation by pin settings at reset without software configuration. In 15-pin configuration mode (COL=0 at reset), if RXD[1:0]=10 and GPO2=1 and RX_DV=1 at reset then PCSCR.BASEX is set to 1 to indicate 1000BASE-X mode and BMCR.AN_EN is set to 1 to enable auto-negotiation. In 3-pin configuration (COL=1 at reset), if GPO2=1 then PCSCR.BASEX is set to 1 to indicate 1000BASE-X mode and auto-negotiation is automatically enabled. In both pin configuration modes the AN_ADV register's reset-default value causes device capabilities to be advertised as follows: full-duplex only, no pause support, no remote fault. See section 6.1 for details about pin configuration at reset.

Figure 6-8. Auto-Negotiation with a Link Partner over 1000BASE-X



The tx_Config_Reg and rx_Config_Reg format for 1000BASE-X auto-negotiation is shown in Figure 6-9. All reserved bits are set to 0. The ACK bit is set and detected by the PCS auto-negotiation hardware.

Figure 6-9. 1000BASE-X Auto-Negotiation tx_Config_Reg and rx_Config_Reg Fields



The AN_ADV register is the source of tx_Config_Reg. The received rx_Config_Reg is written to the AN_RX register. The auto-negotiation fields for AN_ADV and AN_RX are described in Table 6-11 and Table 6-12.

Table 6-11. AN_ADV 1000BASE-X Auto-Negotiation Ability Advertisement Register (MDIO 4)

Bit(s)	Name	Description	R/W	Reset
15	NP	Next Page capability is not supported. This bit should always be set to 0.	RW	0
14	Reserved	Ignore on Read	RO	0
13:12	RF	Remote Fault. Used to indicate to the link partner that a remote fault condition has been detected: 00 = No Error, Link OK 01 = Link Failure 10 = Off Line 11 = Auto-Negotiation Error	RW	00
11:9	ZERO1	Always write 000	RW	0

Bit(s)	Name	Description	R/W	Reset
8:7	PS	Pause. Used to indicate pause capabilities to the link partner. 00 = No Pause 01 = Symmetric Pause 10 = Asymmetric Pause 11 = Both Symmetric and Asymmetric Pause	RW	00
6	HD	Used to indicate ability to support half duplex to link partner. Since half duplex is not supported always write 0.	RW	0
5	FD	Used to indicate ability to support full duplex to link partner. 0 = Do not advertise full duplex capability 1 = Advertise full duplex capability	RW	1
4:0	ZERO2	Always write 00000	RW	00000

Table 6-12. AN_RX 1000BASE-X Auto-negotiation Ability Receive Register (MDIO 5)

Bit(s)	Name	Description	R/W	Reset
15	NP	Next Page. Used by link partner to indicate its PCS has a Next Page to exchange. 0 = No Next Page exchange request 1 = Next Page exchange request	RO	0
14	Acknowledge	Indicates link partner successfully received the previously transmitted base page.	RO	0
13:12	RF	Remote Fault. Link partner uses this field to indicate a remote fault condition has been detected. 00 = No Error, Link OK 01 = Link Failure 10 = Off Line 11 = Auto-Negotiation Error	RO	0
11:9	Reserved	Ignore on Read	RO	0
8:7	PS	Pause. Used by link partner to indicate its pause capabilities. 00 = No Pause 01 = Symmetric Pause 10 = Asymmetric Pause 11 = Both Symmetric and Asymmetric Pause	RO	0
6	HD	Used by link partner to indicate ability to support half duplex. 0 = Not able to support half duplex 1 = Able to support half duplex	RO	0
5	FD	Used by link partner to indicate ability to support full duplex. 0 = Not able to support full duplex 1 = Able to support full duplex	RO	0
4:0	Reserved	Ignore on Read	RO	0

6.8.2 SGMII Control Information Transfer

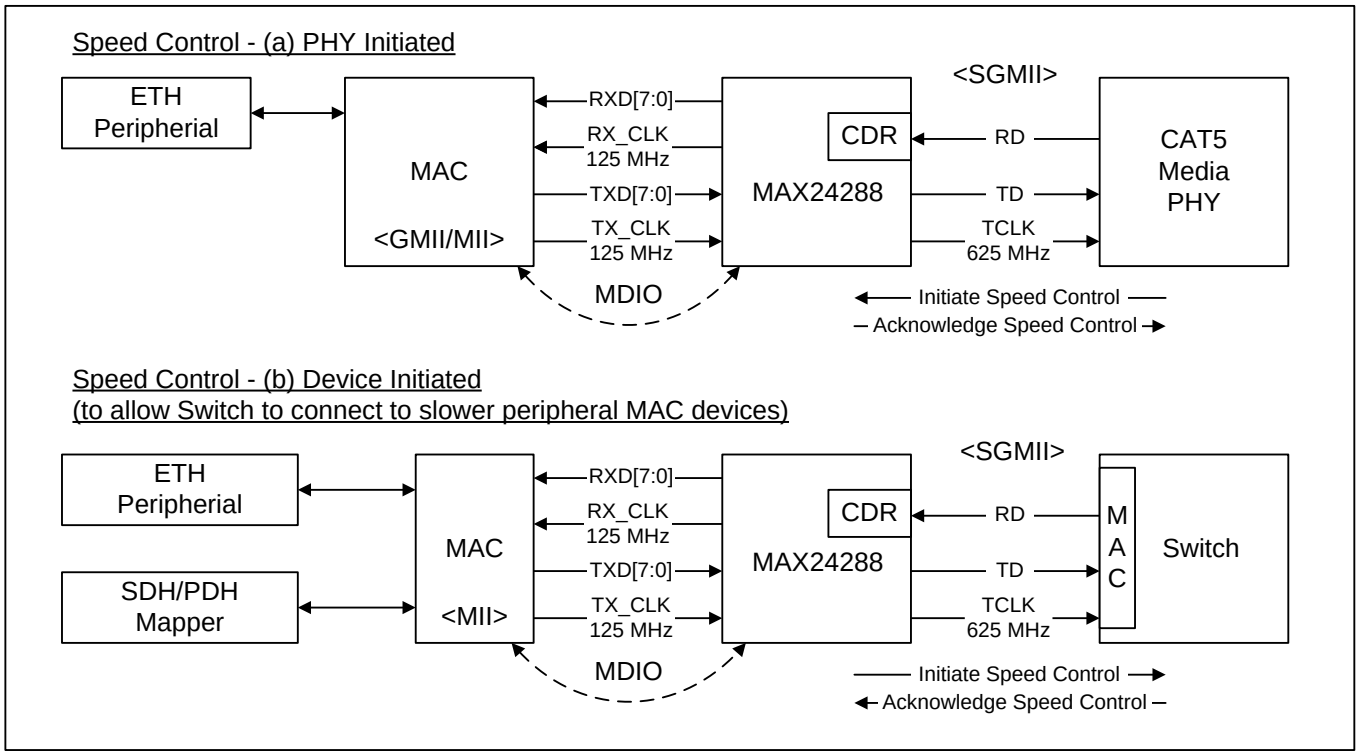
SGMII control information transfer mode is enabled by setting PCSCR.BASEX=0. According the SGMII specification, a PHY sends control information to the neighboring MAC using the same facilities used for 1000BASE-X auto-negotiation (AN_ADV, AN_RX, tx_Config_Reg, rx_Config_Reg, see section 6.8.1). Since the MAX24288 sits between a PHY and a MAC it can behave as the transmitter or receiver in the control information transfer process depending on how it is connected to neighboring components.

When the MAX24288 is connected to a 1000BASE-T PHY, for example, the PHY transfers control information to the MAX24288, which then acknowledges the information transfer. System software then reads the control information from the MAX24288 and configures the MAX24288 to match the PHY. This situation is shown in case (a) of Figure 6-10.

In other scenarios, such as when the MAX24288 is connected by SGMII to a switch IC, shown in case (b) of Figure 6-10, the MAX24288 transfers control information to the neighboring component, which then acknowledges the information transfer. System software then reads the control information from the neighboring component and configures that component to match the MAX24288.

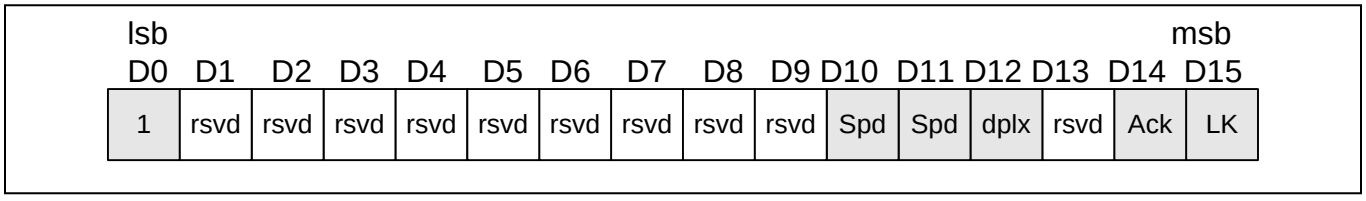
The MAX24288 AN block implements the auto-negotiation state machine shown in IEEE 802.3 Figure 37-6. It also generates an internal link-down status signal whenever the AN state machine is in any state other than AN_DISABLE_LINK_OK or LINK_OK. This link-down signal is used to clear the active-low BMSR.LINK_ST bit and is used to squelch output clock signals on GPIO pins when CR.RCSQL=1.

Figure 6-10. SGMII Control Information Generation, Reception and Acknowledgement



The control information fields carried on the SGMII are: link status (up/down), link speed (1000/100/10Mbps) and duplex mode (full/half). The tx_Config_Reg and rx_Config_Reg format for SGMII control information transfer is shown in Figure 6-11. All reserved bits are set to 0. The ACK bit is set and detected by the PCS hardware.

Figure 6-11. SGMII tx_Config_Reg and rx_Config_Reg Fields



When the MAX24288 is the control information receiver, its AN_ADV register must be set to 0x0001. The received control information is automatically stored in the AN_RX register.

When the MAX24288 is the control information transmitter, the information is sourced from its AN_ADV register.

The MAX24288 can be configured to automatically send SGMII control information by pin settings at reset without software configuration. In 15-pin configuration mode (COL=0 at reset), if RXD[1:0]=10 and GPO2=0 and RX_DV=1

at reset then PCSCR:BASEX is set to 0 to indicate SGMII mode, BMCR.AN_EN is set to 1 to enable auto-negotiation, and the AN_ADV SPD[1:0] bits are set by the reset values of the RXD[1:0] pins. In 3-pin configuration (COL=1 at reset), if GPO2=0 then PCSCR.BASEX is set to 0 to indicate SGMII mode, auto-negotiation is automatically enabled, and the AN_ADV SPD[1:0] bits are set to 10 (1000Mbps). In both pin configuration modes the AN_ADV register's reset-default value causes device capabilities to be advertised as follows: full-duplex only, link up. See section 6.1 for details about pin configuration at reset.

The AN_ADV register is the source of tx_Config_Reg value. The received rx_Config_Reg value is written to the AN_RX register. These meanings are described in Table 6-13 and Table 6-14.

Table 6-13. AN_ADV SGMII Configuration Information Register (MDIO 4)

Bit(s)	Name	Description	R/W	Reset
15	LK	Link Status. 0 = Link down 1 = Link up	RW	Note 1
14	Reserved	Ignore on Read	RO	0
13	ZERO1	Always write 0	RW	0
12	DPLX	Duplex mode 0 = Half duplex 1 = Full duplex	RW	1
11:10	SPD[1:0]	Link speed 00 = 10Mbps 01 = 100Mbps 10 = 1000Mbps 11 = Reserved	RW	Note 1
9:1	ZERO2	Always write 000000000	RW	0
0	ONE	Always write 1	RW	1

Note 1: See the AN_ADV register description.

Table 6-14. AN_RX SGMII Configuration Information Receive Register (MDIO 5)

Bit(s)	Name	Description	R/W	Reset
15	LK	Link partner link status. 0 = Link down 1 = Link up	RO	0
14:13	Reserved	Ignore on read	RO	0
12	DPLX	Link partner duplex mode 0 = Half duplex 1 = Full duplex	RO	0
11:10	SPD	Link partner link speed 00 = 10Mbps 01 = 100Mbps 10 = 1000Mbps 11 = Reserved	RO	0
9:0	Reserved	Ignore on read	RO	0

6.9 Data Paths

The MAX24288 data paths performs bidirectional conversion between a parallel interface (GMII, RGMII or MII) and a 1.25Gbps serial interface (1000BASE-X or SGMII).

In GMII, RGMII and MII modes, the data paths implement the 802.3 PCS and PMA sublayers including auto-negotiation. The parallel interface data is 8 bits wide. The PCS logic performs 8B/10B encoding and decoding. The PMA logic performs 10:1 serialization and deserialization.

6.9.1 Serial to Parallel Conversion and Decoding

Refer to the block diagram in Figure 2-1. Clock and data are recovered from the incoming 1.25Gbps serial signal on RDP/RDN. The serial data is then converted to 10-bit parallel data with 10-bit alignment determined by detecting commas. The 10-bit code groups are then 8B/10B decoded by the PCS decoder as specified in 802.3 clause 36. The 8-bit data stream then passes through a rate adaption buffer that accounts for phase and/or frequency differences between recovered clock and MII clock. Next the data passes through the 1588 packet classifier and modifier logic where 1588 packets are identified, timestamped and optionally modified on-the-fly. Finally, the 8-bit data is clocked out of the device to a neighboring MAC either 4 bits or 8 bits at a time. Half duplex is supported in 10 and 100 Mbps modes but not in 1000 Mbps modes. Carrier extend is not supported in 1000 Mbps modes.

6.9.2 Parallel to Serial Conversion and Encoding

Refer to the block diagram in Figure 2-1. Parallel data is received from the MAC clocked by a 125MHz, 25MHz or 2.5MHz clock, either 4 bits or 8 bits at a time. The data then passes through a rate adaption buffer that accounts for phase and/or frequency differences between MII clock and transmit clock. The 8-bit data stream then passes through the 1588 packet classifier and modifier logic where 1588 packets are identified, timestamped and optionally modified on-the-fly. The data is then 8B/10B encoded by the PCS encoder as specified in 802.3 clause 36. The 10-bit code-groups are then serialized. The resulting 1.25Gbps serial data is transmitted to a neighboring 1000BASE-X optical module or SGMII-interface copper PHY. Half duplex is supported in 10 and 100 Mbps modes but not in 1000 Mbps modes. Carrier extend is not supported in 1000 Mbps modes.

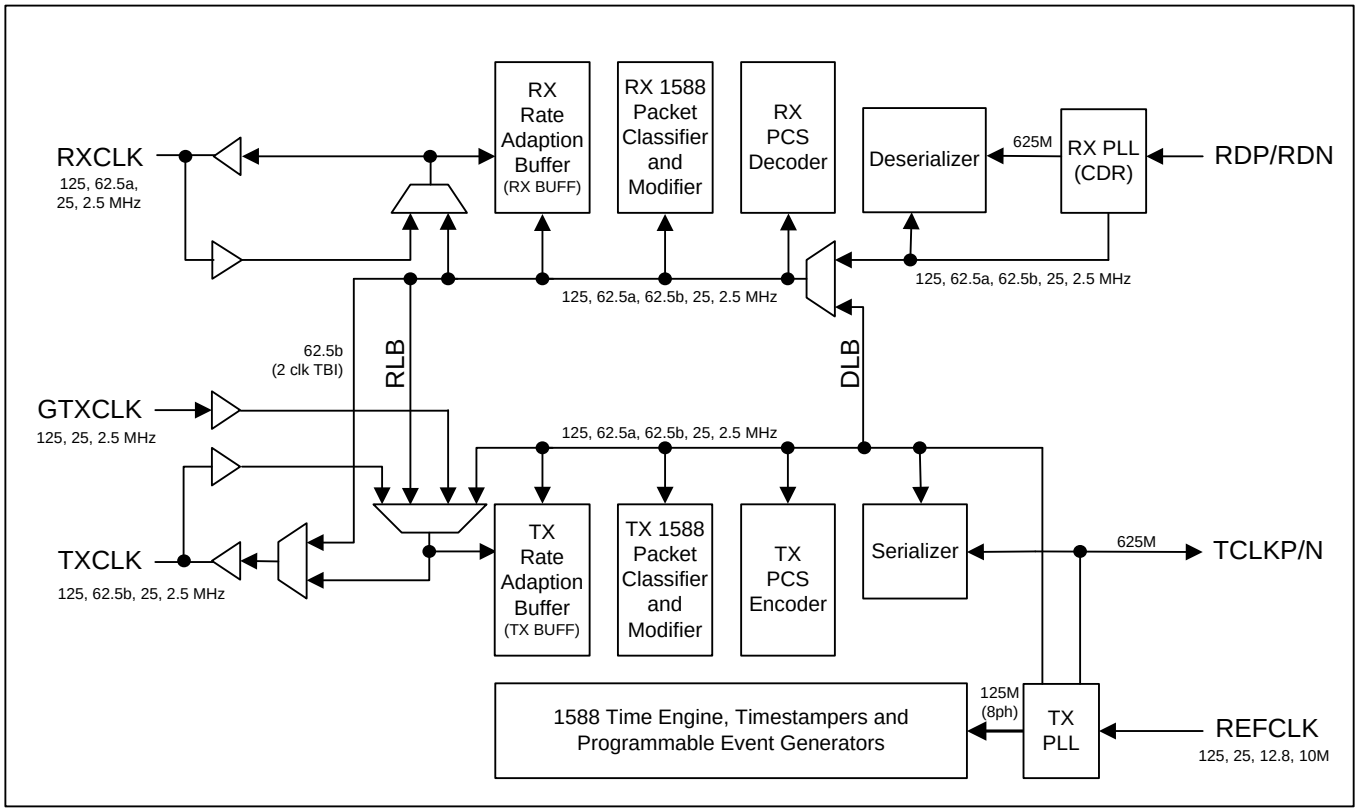
6.9.3 Rate Adaption Buffers, Jumbo Packets and Clock Frequency Differences

The MAX24288 can handle jumbo packets up to 9001 bytes long as long as the clock frequencies of each rate adaption buffer's write clock and the read clock are both within ±100ppm of nominal.

For example, in GMII mode the transmit rate adaption buffer is written by GTXCLK and read by a 125MHz clock frequency locked to the REFCLK signal. If GTXCLK and REFCLK are both maintained within ±100ppm of nominal frequency then jumbo packets up to 9001 bytes long can be accommodated by the transmit rate adaption buffer.

6.10 Timing Paths

Figure 6-12. Timing Path Diagram



Numbers in the tables below are clock frequencies in MHz.

Table 6-15. Timing Path Muxes – No Loopback

Mode	TX Rate Buffer Mux	TXCLK Mux/Driver	RX PCS Mux	RX Rate Buffer Mux	RXCLK Driver
GMII	GTXCLK 125	--	RX PLL 62.5	RX PLL 125	RX PLL 125
RGMII 1000	GTXCLK 125	--	RX PLL 62.5	RX PLL 125	RX PLL 125
RGMII 100	GTXCLK 25	--	RX PLL 62.5	RX PLL 25	RX PLL 25
RGMII 10	GTXCLK 2.5	--	RX PLL 62.5	RX PLL 2.5	RX PLL 2.5
MII 100 - DCE	TX PLL 25	TX PLL 25	RX PLL 62.5	RX PLL 25	RX PLL 25
MII 100 - DTE	TXCLK 25	---	RX PLL 62.5	RXCLK 25	---
MII 10 - DCE	TX PLL 2.5	TX PLL 2.5	RX PLL 62.5	RX PLL 2.5	RX PLL 2.5
MII 10 - DTE	TXCLK 2.5	--	RX PLL 62.5	RXCLK 2.5	---

Table 6-16. Timing Path Muxes – DLB Loopback

Mode	TX Rate Buffer Mux	TXCLK Mux/Driver	RX PCS Mux	RX Rate Buffer Mux	RXCLK Driver
GMII	GTXCLK 125	--	TX PLL 62.5	TX PLL 125	TX PLL 125
RGMII 1000	GTXCLK 125	--	TX PLL 62.5	TX PLL 125	TX PLL 125
RGMII 100	GTXCLK 25	--	TX PLL 62.5	TX PLL 25	TX PLL 25
RGMII 10	GTXCLK 2.5	--	TX PLL 62.5	TX PLL 2.5	TX PLL 2.5
MII 100 - DCE	TX PLL 25	TX PLL 25	TX PLL 62.5	TX PLL 25	TX PLL 25
MII 100 - DTE	TXCLK 25	---	TX PLL 62.5	TXCLK 25	---
MII 10 - DCE	TX PLL 2.5	TX PLL 2.5	TX PLL 62.5	TX PLL 2.5	TX PLL 2.5
MII 10 - DTE	TXCLK 2.5	---	TX PLL 62.5	TXCLK 2.5	---

Table 6-17. Timing Path Muxes – RLB Loopback

Mode	TX Rate Buffer Mux	TXCLK Mux/Driver	RX PCS Mux	RX Rate Buffer Mux	RXCLK Driver
GMII	RX PLL 125	--	RX PLL 62.5	RX PLL 125	RX PLL 125
RGMII 1000	RX PLL 125	--	RX PLL 62.5	RX PLL 125	RX PLL 125
RGMII 100	RX PLL 25	--	RX PLL 62.5	RX PLL 25	RX PLL 25
RGMII 10	RX PLL 2.5	--	RX PLL 62.5	RX PLL 2.5	RX PLL 2.5
MII 100 - DCE	RX PLL 25	TX PLL 25	RX PLL 62.5	RX PLL 25	RX PLL 25
MII 100 - DTE	RXCLK 25	---	RX PLL 62.5	RXCLK 25	---
MII 10 - DCE	RX PLL 2.5	TX PLL 2.5	RX PLL 62.5	RX PLL 2.5	RX PLL 2.5
MII 10 - DTE	RXCLK 2.5	--	RX PLL 62.5	RXCLK 2.5	--

6.10.1 RX PLL

The RX PLL is used to recover the clock from the RDP/RDN high-speed serial input signal. It generates 625MHz for the de-serializer and 125MHz, 62.5MHz, 25MHz and 2.5MHz for the receive-side PCS decoder, 1588 packet classifier and modifier, rate adaption buffer and parallel port logic. It also generates a loss of lock (RLOL) signal for the IR.RLOL latched status bit.

6.10.2 TX PLL

The TX PLL generates a low-jitter 625MHz clock signal for the serializer and the TCLKP/TCLKN differential output. This clock signal meets the jitter requirements of IEEE802.3. It also generates 125MHz, 62.5MHz, 25MHz and 2.5MHz for the transmit-side PCS decoder, 1588 packet classifier and modifier, rate adaption buffer and parallel port logic. In addition it generates an eight-phase 125MHz clock for the IEEE1588 output clock generator, programmable event generators and timestampers.

The TX PLL locks to the REFCLK signal, which can be 125MHz, 25MHz, 12.8MHz or 10MHz as specified by the RXD[3:2] pins during device reset. The 12.8MHz and 10MHz frequencies enable the device to share an oscillator with any clock synchronization ICs that may be on the same board.

6.10.3 Input Jitter Tolerance

The input jitter tolerance is specified at the receive serial input RDP/RDN. The MAX24288 CDR block accepts data with maximum total jitter up to 0.75 UI (or 600 ps) peak-to-peak as required by 802.3 Table 38-10 and as shown in Table 9-8. Total jitter is composed of both deterministic and random components. The allowed random jitter equals the allowed total jitter minus the actual deterministic jitter at RDP/RDN.

6.10.4 Output Jitter Generation

The output jitter generation limit is specified at the transmit serial output (TDP/TDN) of the serializer. According to Table 38-10 of IEEE 802.3, the maximum total jitter generated must be less than 0.24 UI (192 ps) peak-to-peak and the deterministic jitter should be less than 0.10 UI (80 ps) peak-to-peak. MAX24288 typical and maximum jitter generation specifications are shown in Table 9-10. Actual output jitter performance is a function of REFCLK signal jitter. Contact the factory for a tool to calculate output jitter from REFCLK phase noise.

6.10.5 TX PLL Jitter Transfer

The TX PLL has a bandwidth of approximately 200kHz and jitter transfer peaking of 0.1dB or less.

6.10.6 GPIO Pins as Clock Outputs

Reference Clock. A 125MHz clock from the TX PLL (locked to the REFCLK signal) can be output on one or more GPIO pins. See section 6.2 for configuration details. One use for this 125MHz output is to provide a 125MHz transmit clock to a MAC block on a neighboring component. The MAC then uses this signal to clock its transmit GMII/RGMII/MII pins.

Recovered Clock. A 25MHz or 125MHz clock signal from the receive clock and data recovery PLL can be output on one or more GPIO pins. This clock signal is typically used in synchronous Ethernet applications to send recovered Ethernet line timing to the system's central timing function. See section 6.2 for configuration details.

These output clock signals do not glitch during internal switching between frequencies or sources or when being squelched and unsquelched.

6.11 Loopbacks

Three loopbacks are available in the MAX24288. The loopback data paths are shown in the block diagram in Figure 2-1. The clocking paths are shown in section 6.10.

6.11.1 Diagnostic Loopback

Diagnostic loopback is enabled by setting BMCR.DLB=1. During diagnostic loopback the PCS decoder in the receive path takes 10-bit codes from the PCS encoder in the transmit path rather than from the deserializer. In these modes the rate adaption buffers are in the path, and therefore the receive clock can be different than the transmit clock.

During diagnostic loopback, if CR.DLBDO=0 the TDP/TDN output driver is placed in a high-impedance state and the TDP/TDN pins are both pulled up to 3.3V by their internal 50Ω termination resistors. The TCLKP/TCLKN output continues to toggle if enabled. If CR.DLBDO=1 then data is transmitted on TDP/TDN during diagnostic loopback while also being looped back to the receiver.

During diagnostic loopback, the COL signal remains deasserted at all times, unless BMCR.COL_TEST is set, in which case the COL signal behaves as described in 802.3 section 22.2.4.1.9.

6.11.2 Terminal Loopback

Terminal loopback is enabled by setting PCSCR.TLB=1. When this loopback is enabled, the receive CDR takes serial data from the transmit driver rather than from the RDP/RDN pins.

During terminal loopback, if CR.TLBDO=0 the TDP/TDN output driver is placed in a high-impedance state and the TDP/TDN pins are both pulled up to 3.3V by their internal 50Ω termination resistors. The TCLKP/TCLKN output continues to toggle if enabled. If CR.TLBDO=1 then data is transmitted on TDP/TDN during terminal loopback while also being looped back to the receiver..

6.11.3 Remote Loopback

Remote loopback is enable by setting GMIICR.RLB=1. When this loopback is enabled, the transmit parallel interface logic takes data from the receive parallel interface logic rather than from the transmit parallel interface pins. During remote loopback the rate adaption buffers, PCS encoder and decoder, and PCS auto-negotiation function all operate normally.

Loopback control bits BMCR.DLB and PCSCR.TLB must be set to zero for correct remote loopback operation.

During remote loopback, if CR.RLBDO=0 the receive parallel interface pins RXD, RX_DV, RX_ER, CRS and COL are all driven low. If CR.RLBDO=1 then receive data and control information are output on these pins while also being looped back to the transmitter.

6.12 Diagnostic and Test Functions

Transmit PCS Pattern Generator. The PCS encoder has a built-in pattern generator block. These patterns provide different types of jitter in order to test the performance of a downstream PHY receiver. When JIT_DIAG.JIT_EN=1, the PCS encoder outputs 10-bit codes from the jitter pattern generator rather than from the 8B/10B encoding logic. The pattern to be generated is specified by JIT_DIAG.JIT_PAT. Patterns include low-, mixed- and high-frequency test patterns from 802.3 Annex 36A as well as a custom 10-bit pattern from the JIT_DIAG.CUST_PAT register field.

When JIT_EN is set to 1, pattern generation starts and packet transmission stops immediately. This can cut off the tail end of the packet currently being transmitted and can also cause a running disparity error. When JIT_EN is set to 0, pattern generation stops and packet transmission starts immediately. This can result in the transmission of only the tail end of a packet and can also cause a running disparity error.

6.13 1588 Hardware

6.13.1 1588 Time Engine

The MAX24288 has a built-in real-time clock that can be controlled so that it is (1) syntonized or synchronized to a remote master using the IEEE1588 protocol over a packet network, (2) syntonized or synchronized to a master within the system using clock and/or time alignment signals, or (3) syntonized to a remote master using adaptive mode circuit emulation over a packet network.

A block diagram of the time engine is shown in Figure 6-13. As the figure shows, the time engine is an accumulator clocked by a reference clock derived from the REFCLK signal. By default the reference clock is 125MHz from the TX PLL, which is frequency locked to the signal on the REFCLK pin (see Figure 6-12).

The accumulator is a real-time clock with a 48-bit seconds field, a 30-bit nanoseconds field and an 8-bit fractional nanoseconds field. Negative time is not supported. During each reference clock cycle, a number (TIME_ADJ) is added to the accumulator to advance the time. Figure 6-14 shows the logic that generates the TIME_ADJ signal. In free-run operation, when the reference clock frequency is 125MHz, exactly 8.0ns (the period of a 125MHz clock) should be added to the accumulator to advance time by 8ns every 8ns reference clock period.

The TIME_ADJ value has a resolution of 2^{-32}ns while the accumulator has a resolution of 2^{-8}ns. The additional resolution of the TIME_ADJ field is maintained by the delta-sigma ($\Delta\Sigma$) block. When given a TIME_ADJ value that is not an integer multiple of 2^{-8}ns (i.e. a TIME_ADJ value such that N ≤ TIME_ADJ ≤ N+1 where N is an integer multiple of 2^{-8}ns), the delta-sigma block continually dithers between an output of N and an output of N+1 in a pattern and ratio that makes the average output value be exactly equal to the full-resolution TIME_ADJ value. This technique allows very high resolution such as a TIME_ADJ resolution of 2^-32ns in this design. The dithering done by the delta-sigma block does cause phase jitter on output signals generated by the time engine's programmable event generators (see section 6.13.3), but this phase jitter is very high frequency and can be easily filtered by downstream PLLs.

In some applications the time engine must be frequency-locked to a timing master without the conveyance of frequency over layer 1 (Synchronous Ethernet or SDH/SONET). In this situation, system software and the MAX24288's time engine are used to form a control loop similar to a PLL. In this PLL, software implements the phase detector and loop filter while the time engine is the controllable oscillator.

As shown in Figure 6-13 and Figure 6-14, the time engine has three controls that can be used by system software to time- and frequency-lock to a time master:

1. Directly write the time accumulator through the TIME register
2. Frequency adjustment: change the frequency by changing the value in the PERIOD register
3. Time adjustment: temporarily add an offset (PER_ADJ) to the period for a specific number of reference clock cycles (ADJ_CNT)

These controls allow the time engine to be synchronized quickly to its master by first writing the time directly, then doing an initial pull-in step using the time adjustment function, then completing the pull-in process and maintaining lock with frequency (period) adjustments. The time adjustment function is also used to quickly resynchronize time after a time interval spent in holdover without causing slave components to lose synchronization.

Figure 6-13. 1588 Time Engine

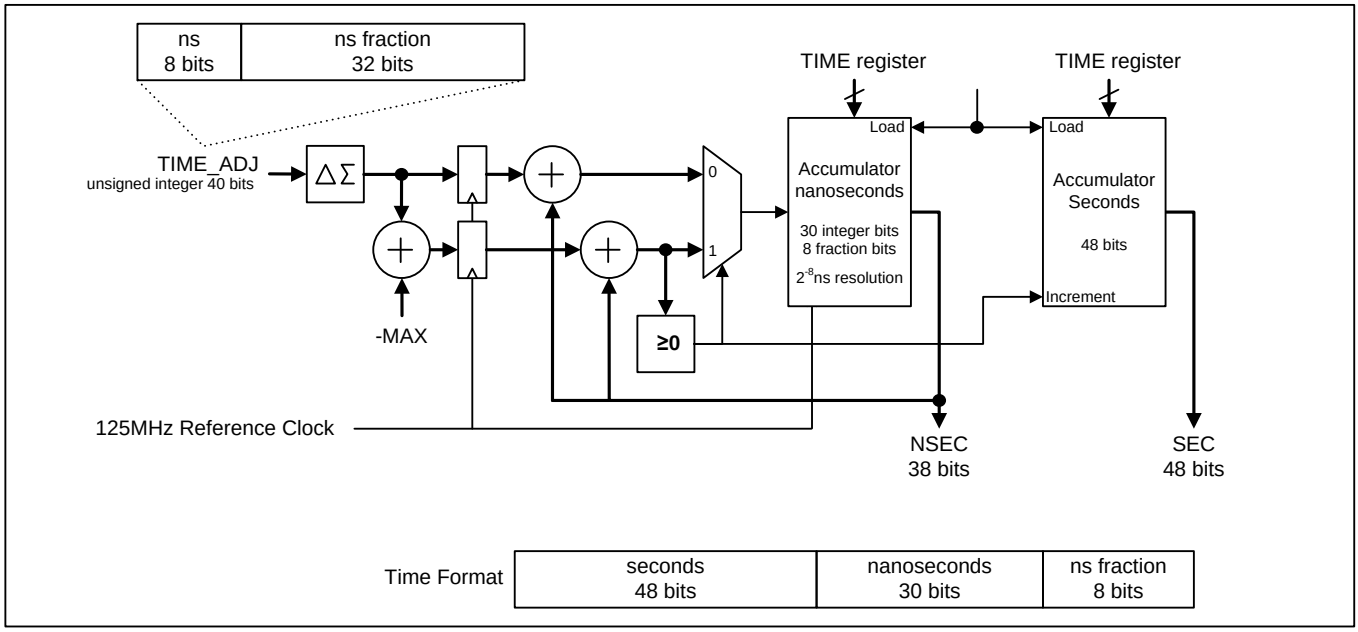
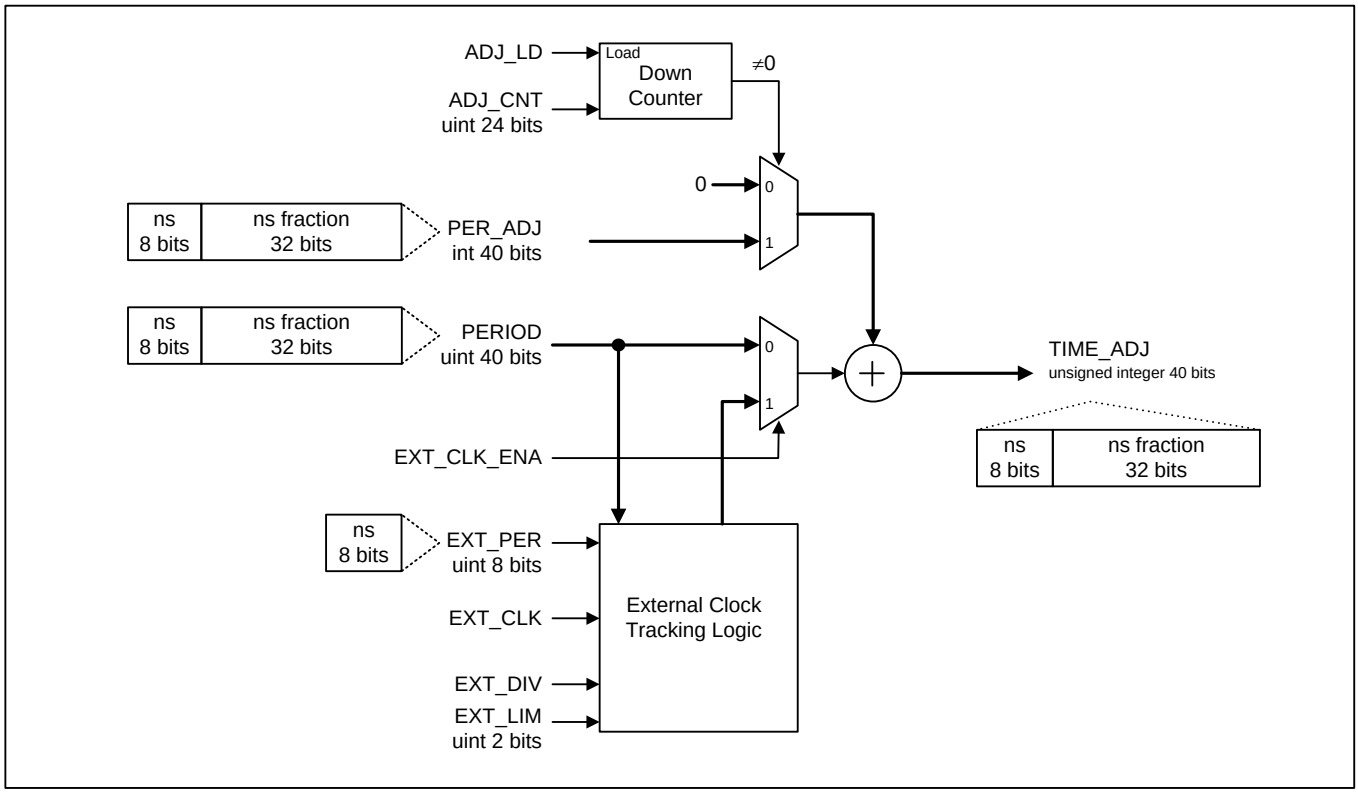


Figure 6-14. Time Engine Period Generator



6.13.1.1 Direct Time Write

The time engine accumulator can be written and read through the TIME register. The time is in 1588 standard format: 48 bits of seconds and 30 bits of nanoseconds. When such a write is done, the time engine instantaneously

jumps to the new time. (Note: this, in turn, can cause instantaneous phase changes in periodic signals generated by the programmable event generators.)

6.13.1.2 Frequency Control

The frequency of the time engine can be changed by writing the period to the PERIOD register. The period register is in units of ns and has 8 integer bits and 32 fractional bits (lsb is 2^{-32}ns).

6.13.1.3 Precise Time Adjustment

The time adjustment control provides a hardware-controlled method to change the time slowly over a large number of reference clock cycles. In a time adjustment operation, an offset is temporarily added to the period for a specific number of reference clock cycles. After that number of cycles, the period reverts back to the value stored in the PERIOD register and the latched status bit PTP_IR.TAC is set. Using this time adjustment control, a specific time change can be made slowly over hundreds, thousands or millions of reference clock cycles. A time adjustment operation is started by writing the period adjustment and cycle count to the PER_ADJ and ADJ_CNT registers, respectively. The period adjustment is in units of ns and has 8 integer bits and 32 fractional bits (lsb is 2^{-32}ns). The cycle count is a 24-bit unsigned integer. The magnitude of the period adjustment must be less than half of the PERIOD register setting for the time adjustment function to work reliably.

As an example of a time adjustment operation, if the period adjustment is set to +1.50 nanoseconds for 1,000,000 clock periods (8ms total duration) the period adjustment register would be set to 0x01,8000,0000, the cycle count register would be set to 0x0F4240 (1,000,000), and the resulting time shift would be -1.50 milliseconds.

6.13.1.4 External Clock Syntonization

If needed, the time engine can be syntonized with an external clock signal on one of the GPIO pins. When PTPCR3.EXT_CLK_ENA=1, PTPCR3.EXT_SRC specifies the GPIO pin on which the clock signal is applied, and PTPCR3.EXT_PER specifies the nominal period of the clock after the PTPCR3.EXT_DIV divider. The nominal period of the clock out of the EXT_DIV divider must be an integer number of nanoseconds and ≥8ns (i.e. frequency ≤125MHz). If the EXT_CLK frequency is greater than 125MHz at the GPIO pin, PTPCR3.EXT_DIV must be set to internally divide the frequency to ≤125MHz.

When the external clock mode is enabled, the PERIOD register should be set to the period of the reference clock (e.g. 8.0ns for a 125MHz external clock). The external clock logic then dynamically adjusts the time values being added to the time engine accumulator to cause time to advance in the accumulator (Figure 6-13) with a long-term fractional frequency offset (FFO) equal to the FFO of the EXT_CLK signal plus the FFO expressed in the PERIOD register (if any) plus the FFO of the REFCLK signal. For example, if the EXT_CLK signal is 1ppm faster than nominal (FFO=+1ppm) and the PERIOD register indicates 8.0ns (i.e. FFO=0ppm) then time advances in the time engine at +1ppm plus the FFO of the REFCLK signal.

Note that the external clock logic does not affect the frequency of the time engine's 125MHz reference clock signal; it only affects the rate that time advances in the time engine accumulator. It also doesn't affect the frequency of any of the clocks generated by the TX PLL. It does affect the output clock generator (section 6.13.2) and programmable event generators (section 6.13.3), and output signals from these blocks do have the same FFO as the time engine accumulator.

As the frequency of the EXT_CLK signal changes, the external clock logic dynamically adjusts the period value being added to the time accumulator to track the frequency changes. The PTPCR3.EXT_LIM field specifies the maximum number of nanoseconds to adjust the period value vs. the value in the PERIOD register. If larger offsets than the value specified by the EXT_LIM field are required, the required adjustments are accumulated in the external clock logic and then added to the time accumulator at the EXT_LIM rate. The effect of this behavior is that output signals derived from the time engine follow frequency changes on the EXT_CLK signal with a reaction speed limited by the EXT_LIM value.

The external clock tracking logic generates approximately 2ns of phase noise on MAX24288 output signals vs. approximately 1ns in other modes of operation.

Note that another way to syntonize the time engine to a clock signal is to use the clock signal as MAX24288's REFCLK signal, since the time engine's reference clock is derived from the REFCLK signal. Using this approach also syntonizes the transmit bit clock of the high-speed serial interface, since that clock is created by the same PLL (the TX PLL, see Figure 6-12).

6.13.2 Output Clock Generator

The primary frequency output from the 1588 time engine is the PTP_CLKO signal. PTP_CLKO can be configured to be 125/MHz ÷ n where n = 1 to 255 as set by PTPCR2.CLKO_DIV. Typical PTP_CLKO frequencies are 125MHz, 62.5MHz, 31.25MHz, 25MHz, 5MHz and 1MHz. PTP_CLKO can be inverted by setting PTPCR2.CLKO_INV=1.

Whenever the 1588 time engine is synchronized to a time or frequency master, PTP_CLKO is, by extension, syntonized to the frequency of the master. Note that the frequency of PTP_CLKO is immediately affected by all 1588 time engine controls: direct time write, period adjustment, and time adjustment.

The output jitter of PTP_CLKO is approximately 1ns. To achieve this level of jitter when clocking the 1588 time engine with a 125MHz (8ns) reference clock, eight phases of the reference clock are used, effectively giving a 1GHz reference clock from which to synthesize PTP_CLKO.

PTP_CLKO can be made available on any of general-purpose I/O pins GPO2, GPIO2, or GPIO4-7 by configuring GPIOCR1 and/or GPIOCR2 appropriately.

6.13.3 Programmable Event Generators

The MAX24288 has two identical programmable event generators (PEGs). Each of these PEGs can be configured to generate output signals with time-triggered rising or falling edges. PEG output signals can be non-periodic control signals, 50% duty cycle clock signals, or periodic pulses, such as a one pulse per second (1PPS) signal. For each PEG, one or more GPIO pins must be configured using the appropriate field in GPIOCR1 or GPIOCR2 to output the PEG signal.

Each PEG has a controller that accepts commands written to a 22-bit-wide, 16-word-deep FIFO that stores multiple event generation commands. As shown in Table 6-18, bits 15:0 of each 22-bit word are a 16-bit time field or repeat count. Bits 19:16 are a command code (see Table 6-19). Bit 20 specifies GPIO pin behavior after the event: continue to drive or go high impedance. Bit 21 marks the event command as one for which the PEG controller must set the PTP_IR.P1EC or P2EC latched status bit when it has completed the command.

Table 6-18. PEG Command FIFO Fields

FIFO Bits	Field	Description
21	Stat	When this bit is set, the PEG controller sets the P1EC or P2EC latched status bit after the event command is completed. This can be used, for example, to interrupt software on one-second boundaries.
20	Disable (After Event)	0 = Output pin continues to drive the last level (high or low) after the event 1 = Output pin goes high impedance one reference clock (8ns) after the event.
19:16	Command	4-bit event command code. See Table 6-19 for commands.
15:0	Data	16-bit data. See Table 6-19 for descriptions of what the data means for each command code.

Table 6-19. PEG Commands

FIFO[19:16] Command	Command Description	
0000	**Repeat.** Let N=FIFO[3:0] and M=FIFO[15:4]. Repeat the last N FIFO entries M times. M=0 indicates repeat forever (simple clock mode) until a new command is written to the PEG FIFO.	
0001	**Fractional Clock Synthesis Repeat.** Let M1=FIFO[15:8] and M2=FIFO[7:0]. Repeat last 4 FIFO entries forever. Two structures are possible here for the previous 4 FIFO entries: (1) Four 16-bit relative-time toggle commands (command=1111). Words 1 and 2 are repeated M1 times then words 3 and 4 are repeated M2 times. (2) Two 32-bit relative-time toggle commands (command=1110). The 32-bit command in words 1 and 2 is repeated 2*M1 times then the 32-bit command in words 3 and 4 is repeated 2*M2 times. In both cases the pattern is repeated forever until a new command is written to the PEG FIFO.	
0010	**Set Absolute Time Reference.** Set the PEG's absolute time reference point to the time engine's current time. The next relative time command will be relative to this reference point. FIFO[15:0] ignored.	
0011	Undefined	
01xx	**Create Positive Edge**	These three values of FIFO[19:18] are combined with the three values of FIFO[17:16]=01, 10 or 11 (described below) to create time-triggered edge placements in the PEG output signal.
10xx	**Create Negative Edge**	
11xx	**Toggle (Create Opposite Edge)**	
xx00 (xx≠00)	Undefined	
xx01 (xx≠00)	**Absolute time value.** First word of 5-word, 48-bit absolute time value. FIFO[15:0]: Word 1 = seconds[47:32], word 2 = seconds[31:16], word 3 = seconds[15:0], word 4 = ns[29:16], word 5 = ns[15:0]. The command field (FIFO[19:16]) of words 2 through 5 is don't-care (recommended value: same as word 1)	
xx10 (xx≠00)	**32-bit relative time value.** First word of 2-word relative time value. FIFO[15:0]: Word 1 = ns[29:16], word 2 =ns[15:0]. The command field (FIFO[19:16]) of word 2 is don't-care (recommended value: same as word 1).	
xx11 (xx≠00)	**16-bit relative time value.** Single-word relative time value, FIFO[15:0] is ns[15:0].	

An absolute time command (xx01) or the Set Absolute Time Reference command (0010) must be used before relative time commands (xx10 or xx11) can be used. Relative commands create an event at a time relative to the previous event (the previous event can be absolute or relative).

When PEGCR.P1RES or P2RES is set to 1, the resolution of the 32-bit and 16-bit relative time values is increased by a factor of 256, giving a resolution of 1/256 (i.e. 2^{-8}) nanoseconds. This is used to generate a signal with a period that is not an integer number of nanoseconds. The P1RES or P2RES bit must remain unchanged for the duration of a repeat or fractional synthesis repeat command.

The control bits PEGCR.P1RST and P2RST are used to reset the PEGs. When a PEG is reset, its command FIFO is emptied, its control logic is reset, and its output signal is driven low.

The control bits PEGCR.P1DIS and P2DIS prevent the PEG from moving to the next command in the PEG FIFO. When PnDIS=1, the PEG continues to execute any command or group of commands (grouped by repeat or fractional clock synthesis repeat commands) already being executed, but it cannot proceed to the next command until PnDIS is set to 0. This feature is valuable for ensuring that repeat groups are completely loaded into the FIFO before being executed by the PEG.

Real-time status bits PTP_SR.P1FF and P2FF indicate when the PEG command FIFOs are full. System software should monitor these bits to prevent FIFO overflow.

The PEG1 Command FIFO is written through the PEG1_FIFO register. The PEG2 Command FIFO is written through the PEG2_FIFO register.

The PEG Repeat command can be used to make periodic signals with periods that are integer multiples of 2ns (PEGCR.PnRES=0) or 2/256ns (PEGCR.PnRES=1). The basic idea is to first set an absolute reference and then repeat a toggle command at a relative time equal to half the period of the desired signal. As an example, to generate a 50% duty cycle 25MHz clock using PEG1, follow these steps:

1. Set PEGCR.P1DIS=1.
2. Write 0x20000 to the PEG1 FIFO (Set Absolute Time Reference command).
3. Write 0xF0014 to the PEG1 FIFO (Toggle command, 16-bit relative time, 20ns half cycle).
4. Write 0x00001 to the PEG1 FIFO (Repeat command, repeat previous 1 FIFO entry forever).
5. Set PEGCR.P1DIS=0.

As another example, to generate a one pulse per second (1PPS) signal with a 50ns wide pulse using PEG2, the basic idea is to first generate a rising edge at the next 1 second boundary (absolute time). Then repeat forever these two edge placements: a toggle command 50ns later followed by another toggle command 1 second minus 50ns later. Specifically, follow these steps:

1. Set PEGCR.P2DIS=1.
2. Write 0x50000 to the PEG2 FIFO (Create Positive Edge command, 48-bit absolute time, sec[47:32])
3. Write 0x50000 to the PEG2 FIFO (sec[31:16] for above command)
4. Write 0x50000 to the PEG2 FIFO (sec[15:0] for above command)
5. Write 0x50000 to the PEG2 FIFO (ns[31:16] for above command)
6. Write 0x50000 to the PEG2 FIFO (ns[15:0] for above command)
7. Write 0xF0032 to the PEG2 FIFO (Toggle command, 16-bit relative time, 50ns)
8. Write 0xE3B9A to the PEG2 FIFO (Toggle command, 32-bit relative time, 999,999,950ns)
9. Write 0x0C9CE to the PEG2 FIFO (ns[15:0] for above command)
10. Write 0x00003 to the PEG2 FIFO (Repeat command, repeat previous 3 FIFO entries, i.e. the two Toggle commands, forever)
11. Set PEGCR.P2DIS=0.

Note that in steps 2, 3 and 4 the seconds value would have to be set to a specific one-second period in the near future for the example to work correctly.

The PEG toggle and repeat commands for other common frequencies are shown in Table 6-20.

Table 6-20. Common Frequencies Using Repeat Command

Frequency	Toggle Command			Repeat Command	PnRES bit
	FIFO[19:16] Command	Half Period (ns)	FIFO[15:0] Hex		
1 Hz	0xE	500,000,000	0x1DCD 0x6500	0x00002	0
8 KHz	0xF	62,500	0xF424	0x00001	0
1MHz	0xF	500	0x01F4	0x00001	0
2.048MHz	0xF	62,500/256	0xF424	0x00001	1
10MHz	0xF	50	0x0032	0x00001	0
25MHz	0xF	20	0x0014	0x00001	0

The PEG Fractional Clock Synthesis Repeat command adds additional frequency capabilities. See the description of this command in Table 6-19. The generated jitter is minimized by the PEG internally repeating 1/16 of M1 then 1/16 of M2 in an alternating manner. This jitter is high frequency and therefore easily filtered by downstream PLLs. The numbers M1 and M2 are adhered to precisely to get an exact synthesis, even if M1 and M2 are not integer multiples of 16. The 1/256 resolution can be applied by setting PEGCR.P1RES or P2RES to 1 if needed.

Table 6-21 shows how to use the fractional clock synthesis repeat command to create common telecom frequencies (with P1RES or P2RES set to 1). FIFO Entries 1 through 4 are relative-time toggle commands.

Table 6-21. Common Frequencies Using Fractional Clock Synthesis Repeat Command

FREQ	M1	M2	FIFO Entry 1	FIFO Entry 2	FIFO Entry 3	FIFO Entry 4
30.720MHz	1	2	16+ 70/256 ns	16+ 70/256 ns	16+ 71/256 ns	16+ 71/256 ns
19.440MHz	155	88	25 + 184/256ns	25 + 184/256ns	25 + 185/256ns	25 + 185/256ns
1.544MHz	86	107	323 + 213/256 ns (32-bit Command)		323 + 214/256 ns (32-bit Command)	

Notes: The closest spacing of a relative edge to the previous event is 16ns (two 125MHz reference clock periods). Also, the highest frequency periodic signal that a PEG can produce is one fourth of the reference clock frequency (e.g. 31.25MHz for a 125MHz reference clock).

A PEG can also be used to encode a data value as a pulse of a specific width. A second device can then use input signal timestamping (section 6.13.4) to determine the width and decode the value. As an example, pulse width could be used to encode 8 bits of data per pulse using pulse_width =n*32 + 64 ns which creates pulse widths from 64ns to 8,224ns (255*32+64). Thirty-two bits of data can then be sent as four separate pulses. This method can be a useful way to convey the exact time at the one-second boundary from a time master to a time slave in the same system using the wire that already carries the 1 pulse per second signal or other time alignment signal.

6.13.4 Input Signal Timestamping

Any of the three timestampers in the MAX24288 (TS1, TS2 or TS3) can timestamp edges of an input signal (rising, falling or both). This feature can be used in a wide variety of applications to timestamp signals from sensors or other ICs in the system to note the precise time something important happens.

In addition, this feature can be used to time-align the MAX24288's time engine to another component in the system. This can be necessary, for instance, when the MAX24288 is a timestamper on one card in a multicard and/or multiport switch or router. Typically, such a system is required to perform as a 1588 boundary clock or transparent clock in which packets are timestamped at the MII interfaces of multiple ports. In such a system it is important that all the timestampers have a common understanding of the current time. The easiest, most accurate way to achieve this is to have each timestamper time-locked to a time alignment signal from a central 1588 clock.

Aligning the time engine to a time alignment signal involves these concepts:

- Configure the source of the time alignment signal to output a signal with a rising or falling edge at an exact time boundary, such as a 1 pulse per second (1 PPS) signal that goes high at the start of each second.
- Apply that signal to the MAX24288 on the GPIO pin specified by TSCR.TS1SRC_SEL (this example presumes the use of TS1). Also, configure the GPIO pin as an input by setting GPIOCR1.GPIOn_SEL=0.
- Configure the timestamper to timestamp the significant edge of the time alignment signal (TSCR.TS1_EDGE=01 or 10).
- System software then looks for an input signal timestamp by polling the MAX24288's real-time status bit PTP_SR.TS1_NE or waiting for an interrupt generated by latched status register PTP_IR.TS1_NE.
- Software reads the timestamp from the FIFO as described in the TS1_FIFO register description.
- Finally, software implements a PLL phase detector by calculating the difference between the timestamp and the expected value, implements the desired PLL loop filter behavior, and controls the time engine using any of the three controls described in section 6.13.1 (typically frequency/period).
- This process is repeated continually for each significant edge of the input time alignment signal.

As shown in Table 7-3, when an input signal timestamp is read from a timestamp FIFO, bit 15 of TEIO5 indicates the polarity of the input signal edge: 0=falling, 1=rising.

Each timestamper has an eight entry FIFO. Software must be able to read timestamps out of the FIFO faster than the expected time alignment signal frequency to avoid FIFO overflow. Timestamper FIFO overflow is signaled by latched status bit PTP_IR.TSn_OF and can generate an interrupt if configured to do so. PTP_SR.TSn_NE provides real-time FIFO empty/not-empty status. A transition from FIFO empty to FIFO not-empty is signaled by latched status bit PTP_IR.TSn_NE, which can also generate an interrupt if configured.

For applications that need to use a timestamper to lock to an input clock signal, the signal going to the TS1 timestamper can be internally divided down by one or both of the TS1 dividers (configured by TS1_DIV1 and TSCR.TS1_DIV2. This frequency division reduces the number of edges that must be timestamped to avoid overflowing the timestamp FIFO.

The precision of input signal timestamps is 1ns (using eight phases of the 125MHz reference clock). The pulse width of the input time alignment signal can be very small (less than 10 ns) when only one edge is timestamped. When both edges are timestamped, it must be ≥24ns.

6.13.5 Packet Timestamping

Timestampers TS2 and TS3 can be configured to timestamp the start of egress and ingress packets when TSCR.TS2SRC_SEL=0 and TSCR.TS3SRC_SEL=0, respectively. In this mode, the start of every packet is timestamped, but only PTP packets that are qualified by the packet classifier (section 6.13.6) and have suitable event types (section 6.13.6.3) have their timestamps (1) saved to a timestamp FIFO for system software to read, (2) written into the packet itself (section 6.13.7) or (3) used to update the PTPv2 correction field (section 6.13.7). The precision of the timestamps is 1ns, which is achieved by looking for the start of packet in the 1.25Gbps serial domain and using eight phases of the 125MHz reference clock. Timestamps are formatted as specified by IEEE1588 with a 48-bit seconds field and a 30-bit nanoseconds field (see Table 7-3).

IEEE1588 specifies the timestamping instant for an ingress or egress packet to be the instant when the first bit after the Ethernet start-of-frame delimiter crosses the reference plane marking the boundary between the PTP node and the network. Since the MAX24288 timestampers are not ideally located at the reference plane, the timestamps they generate must be corrected for ingress and egress latencies as described in IEEE1588v2 section 7.3.4.2. These latencies are the sum of the latency of the electrical or optical PHY on the network side of the MAX24288 and the latency of the MAX24288 from the high-speed 1000BASE-X/SGMII pins to the transmit and receive start-of-packet (SOP) detectors. MAX24288 latencies are shown in section 9.3.10. Correction for these latencies can be done by system software or by using MAX24288 on-the-fly corrections as described in section 6.13.7.6.

When a packet timestamp is written to a timestamp FIFO, packet identification information is also written to the FIFO. This ID information consists of 4-bit message type, 16-bit sequence ID, and 12-bit identity code. The message type field is the lower 4 bits of the control field for a 1588v1 message, or the 4-bit messageType field for a 1588v2 message. The 16-bit sequence ID field is the sequenceId field from the PTP message header for both 1588v1 and 1588v2 messages.

By default (PTPCR1.VID2FIFO=0), the 12-bit identity code is the 12-bit sum of bytes 20 to 29 of the PTP message header. These bytes contain the sourcePortIdentity field in 1588v2 packets or the msgType, sourceCommunicationTechnology and sourceUuid fields in 1588v1 packets. This sum provides a simple and effective hash code that is unlikely to produce the same value for two or more values of the 10-byte field. When PTPCR1.VID2FIFO=1, the 12-bit identity code is the packet's outermost VLAN ID.

The message type, sequence ID and identity code fields can be read from the FIFO through the HDR_DAT1 and HDR_DAT2 registers. When timestamper TS2 or TS3 is configured to timestamp packets, HDR_DAT1 and HDR_DAT2 are updated at the same time that the TEIO registers are updated when the FIFO for that timestamper is read (TERW.RD=1 and TERW.RDSEL=010 or 011).

6.13.6 Packet Classification

The MAX24288 has two packet classifiers (PC), one ingress and one egress. Each packet classifier examines packet headers and qualifies suitable PTP packets. Each PC has built-in understanding of Ethernet headers, VLAN tags, 802.1ah MAC-in-MAC, MEF headers, MPLS tags, and IPv4/UDP and IPv6/UDP headers. The PC can be configured to identify PTP event packets carried by layer 2 Ethernet frames, MEF frames, MPLS or IP/UDP packets.

The PC can identify PTPv2 packets, PTPv1 packets or both and also supports future versions of PTP. Using the PKT_CLASS.PTP_VERSION and VER_EXACT fields, the PC can be configured to timestamp only packets with one specified version number or any version number less than or equal to a specified version number.

To handle MAC-in-MAC as defined in IEEE 802.1ah, when the I-TAG Ethertype 0x88E7 is detected the PC skips the following 4 bytes of I-TAG information and then examines the header fields of the inner Ethernet frame. The PC can navigate multiple levels of MAC-in-MAC nesting.

The PC also can traverse multiple levels of VLAN tags. A VLAN tag is indicated by any of the standard VLAN Ethertypes 0x8100, 0x88A8, 0x9100, 0x9200 and 0x9300 or the optional value set in the VLAN2_ID register. When a VLAN Ethertype is detected, the PC skips the next 2 bytes of VLAN information and then examines the next Ethertype location. The PC can navigate any combination of VLAN and MAC-in-MAC Ethertypes.

For MPLS, the PC can traverse stacked MPLS tags to find the innermost tag. The start of a 32-bit MPLS tag is indicated by the MPLS multicast Ethertype 0x8848 or the MPLS unicast Ethertype 0x8847. If S=0 in an MPLS tag (indicating it is not the innermost tag), then the tag is skipped and the next tag is examined. If S=1 in the MPLS tag (the innermost tag), then when PKT_CLASS.MPLS_MCAST=1 or MPLS_UCAST=1, the packet classifier compares the 20-bit MPLS label with the MPLS_LABEL registers as a qualification criterion.

The PC has both hardwired logic and configurable logic available to perform the packet classification process. These are discussed in sections 6.13.6.1 and 6.13.6.2 below.

Note: To minimize packet classification errors during configuration register changes, PTPCR1.PC_PM_DIS should be set before modifying PC configuration registers and then cleared after all registers changes have been made.

6.13.6.1 Hardwired Packet Classifier (HPC)

The hardwired packet classifier can be configured to look for and qualify PTP messages carried by layer 2 Ethernet frames, MEF frames, MPLS or IP/UDP packets using the bit fields in the PKT_CLASS register. The pseudocode below describes HPC operation. To keep the text below straightforward, VLAN tags and MAC-in-MAC are not discussed. The HPC, however, does navigate VLAN tags and MAC-in-MAC nesting to find the PTP message, as described above.

PTP over IEEE802.3 Ethernet, Standard Ethertype

```
IF PKT_CLASS.ENET=1 AND Ethertype = 0x88F7 (PTP Ethertype) THEN
    A PTP message starts at the byte following the Ethertype.
    IF versionPTP matches PKT_CLASS.PTP_VERSION THEN
            Packet Qualifies
```

PTP over IEEE802.3 Ethernet, Alternate Ethertype

```
IF PKT_CLASS.ENET_CFG=1 AND Ethertype = ETYPE_ALT THEN
    A PTP message starts at the byte following the Ethertype.
    IF versionPTP matches PKT_CLASS.PTP_VERSION THEN
            Packet Qualifies
```

PTP over UDP/IPv4/Ethernet

```
IF PKT_CLASS.UDP_IPv4=1 _
    AND Ethertype=0x0800 (IPv4) _
    AND IPv4 Protocol field = 0x11 (UDP)
    AND (UDP_SRC=0xFFFF OR UDP Source Port = UDP_SRC) _
    AND (UDP_DST=0xFFFF OR UDP Destination Port = UDP_DST) THEN
        A PTP message starts at the byte following the UDP header.
        IF versionPTP matches PKT_CLASS.PTP_VERSION THEN
```

Packet Qualifies

PTP over UDP/IPv6/Ethernet
IF PKT_CLASS.UDP_IPv6=1 _
AND Ethertype=0x86DD (IPv6) _
AND IPv6 Next Header field= 0x11 (UDP)
AND (UDP_SRC=0xFFFF OR UDP Source Port = UDP_SRC) _
AND (UDP_DST=0xFFFF OR UDP Destination Port = UDP_DST) THEN
A PTP message starts at the byte following the UDP header.
IF versionPTP matches PKT_CLASS.PTP_VERSION THEN
Packet Qualifies

PTP over MPLS Multicast over Ethernet
IF PKT_CLASS.MPLS_MCAST =1 _
AND Ethertype=0x8848 (MPLS Multicast) _
AND MPLS Inner Label=MPLS_LABEL THEN
A PTP message starts at the byte following the MPLS inner label.
IF versionPTP matches PKT_CLASS.PTP_VERSION THEN
Packet Qualifies

PTP over MPLS Unicast over Ethernet
IF PKT_CLASS.MPLS_UCAST =1 _
AND Ethertype=0x8847 (MPLS Multicast) _
AND MPLS Inner Label=MPLS_LABEL THEN
A PTP message starts at the byte following the MPLS inner label.
IF versionPTP matches PKT_CLASS.PTP_VERSION THEN
Packet Qualifies

PTP over MEF
IF PKT_CLASS.MEF_CFG =1 _
AND Ethertype=0x88D8 (MEF-8 CES) _
AND MEF ECID field = MEF_ECID THEN
A PTP message starts at the byte following the MEF header.
IF versionPTP matches PKT_CLASS.PTP_VERSION THEN
Packet Qualifies

6.13.6.2 Configurable Packet Classifier (CPC)

The configurable packet classifier logic can be used to set up qualification criteria other than those done by the HPC. There are two ways to use the CPC criteria:

- "AND" them with HPC criteria to add more criteria that have to be met to qualify a packet.
- "OR" them with HPC criteria starting at a specific point in the packet headers (the CPC start position) to provide an alternate set of qualification criteria.

The choice of AND mode vs. OR mode is set by the PKT_CLASS.CFG_OR bit. These modes are discussed in more detail below.

In both modes a key reference point is the CPC start position specified in PKT_CLASS.CFG_START. Table 6-22 lists the possible CPC start positions.

Table 6-22. Configurable Packet Classifier Start Positions

CFG_START	CPC Start Position
0	Disabled
1	Start at the beginning of the packet
2	Start after first Ethertype that is not a VLAN Ethertype and not an I-TAG Ethertype (i.e. not 0x8100, 0x88A8, 0x88E7, 0x9100, 0x9200, 0x9300)
3	Start after PTP Ethertype (0x88F7)
4	Start after the configurable Ethertype specified by ETYPE_ALT
5	Start after MEF Ethertype (0x88D8)
6	Start after either MPLS Ethertype (0x8847 or 0x8848)
7	Start after MPLS unicast Ethertype (0x8847)
8	Start after IPv4 Ethertype (0x0800)
9	Start after IPv6 Ethertype (0x86DD)
10	Start after MEF header
11	Start after MPLS inner label specified by MPLS_LABEL following either MPLS Ethertype (0x8847 or 0x8848)
12	Starts after MPLS inner label specified by MPLS_LABEL following MPLS unicast Ethertype (0x8847)
13	Start after IPv4/UDP header
14	Start after IPv6/UDP header
15	Start after either IPv4/UDP or IPv6/UDP header

The CPC has eight bit-maskable 16-bit matching criteria, any or all of which can be used to set up packet qualification criteria. Each of the criteria has a programmable offset relative to the CPC start position.

A CPC matching field is configured for use by following these steps:

1. Specify which bits should be compared in the CFG_MASK register.
2. Specify the required values of the bits to be compared in the CFG_MATCH register.
3. Specify the offset from the CPC start position to the comparison point in the CFG_OFFSET register and set CFG_OFFSET.ENABLE=1.
4. Write the CFG_WR register with the number of the CPC matching field (0-7) in the CFG_SEL field and WR=1.

6.13.6.2.1 AND Mode

In AND mode (PKT_CLASS.CFG_OR=0) a packet must meet HPC criteria AND must meet all enabled CPC criteria to be qualified. AND mode can be used, for example, to add criteria for acceptable Ethernet destination address and/or IP destination address (or ranges of addresses).

For CFG_START values 3 through 15 (see Table 6-22), the CPC criteria are tied to a particular protocol stack specified by the CFG_START value. The CPC criteria are, therefore, only applied to packets with that protocol stack. For example, if CFG_START=8 ("starts after IPv4 Ethertype") then only packets with an IPv4 Ethertype would be further qualified by the CPC criteria.

If CFG_START=1 ("starts at the beginning of the packet") then all packets are further qualified by the CPC criteria, which can be useful for Ethernet address criteria, for example.

Note that in AND mode the PTP_OFFSET indirect register (used in OR mode) is ignored because the start of the PTP header is determined by the HPC by examining packet header fields.

6.13.6.2.2 OR Mode

In OR mode (PKT_CLASS.CFG_OR=1) a packet is qualified if it meets any of the HPC criteria enabled in the PKT_CLASS register (i.e. normal HPC operation) OR it meets HPC criteria up to the CPC start position and then meets CPC criteria thereafter.

For CFG_START values 3 through 15 (see Table 6-22), the CPC criteria are tied to a particular Ethertype or packet header (specified by the CFG_START value) being present in the packet prior to the CPC start position. The CPC criteria are, therefore, only applied to packets that have that Ethertype or packet header prior to the CPC start position.

The following is an example CPC configuration to qualify packets with a PTP-over-IPv4/UDP-over-MPLS-over-Ethernet protocol stack. It first uses HPC criteria for MPLS-over-Ethernet and then uses three configurable matching criteria to look for IPVER=IPv4, IPv4 Protocol=UDP and UDP Destination Port=319 (PTP) after the MPLS inner label.

PKT_CLASS.CFG_OR=1
PKT_CLASS.CFG_START = 12 (CPC start position is after MPLS inner label for MPLS unicast)

CFG_MATCH = 0x4000 (IPVER = IPv4)
CFG_MASK = 0xF000

CFG_OFFSET = 0x8000 + 0
CFG_WR = 0x0080 (write to criterion 1)

CFG_MATCH = 0x0011 (IPv4 protocol = UDP)
CFG_MASK = 0x00FF

CFG_OFFSET = 0x8000 + 8
CFG_WR = 0x0081 (write to criterion 2)

CFG_MATCH = 0x013F (UDP destination port = 319)
CFG_MASK = 0xFFFF

CFG_OFFSET = 0x8000 + 22(decimal)
CFG_WR = 0x0082 (write to criterion 3)

CFG_OFFSET = 28
CFG_WR = 0x008F (write to PTP_OFFSET register)

In this example if the HPC finds a packet with MPLS unicast Ethertype and MPLS inner label equal to MPLS_LABEL then the HPC triggers the CPC to further qualify the packet. If the CPC finds that the packet matches all three configured criteria then it declares the packet qualified.

Note that if PKT_CLASS.MPLS_UCAST=1 and the MAX24288 is configured as described in the example above, then packets can have a PTP-over-MPLS-over-Ethernet protocol stack (qualified by the HPC only) OR a PTP-over-IPv4/UDP-over-MPLS-over-Ethernet protocol stack (qualified by a combination of the HPC and the CPC). The ability to qualify packets with one OR the other of these protocol stacks (OR other protocol stacks if enabled in PKT_CLASS) illustrates OR mode. If, in contrast, PKT_CLASS.MPLS_UCAST=0, then packets with the PTP-over-MPLS-over-Ethernet protocol stack are not qualified, but packets with the PTP-over-IPv4/UDP-over-MPLS-over-Ethernet protocol stack are still qualified.

If CFG_START=1 ("starts at the beginning of the packet") then the CPC criteria are applied starting at the beginning of the packet and therefore provide a qualification path that is entirely independent of the HPC. In this case, packets are qualified if they meet the CPC criteria OR they meet the HPC criteria enabled in the PKT_CLASS register.

One additional parameter must be set to use OR mode. While the HPC operating alone on supported protocol stacks can locate the start of the PTP header by examining fields in lower-layer protocols (e.g. IP Protocol field,

UDP Destination Port), the CPC does not have this ability. In OR mode, therefore, the offset from the CPC start position to the start of the PTP message header must be written to the PTP_OFFSET register. Note that PTP_OFFSET=0 is invalid. OR mode scenarios where PTP_OFFSET=0 is desired can be done in AND mode instead.

In OR mode packets that meet all HPC and CPC criteria up to the beginning of the PTP header must also meet this PTP header criteria to be qualified: versionPTP matches PKT_CLASS.PTP_VERSION.

6.13.6.3 PTP Message Type Checking, Timestamp FIFOs

A packet declared to be qualified by the HPC and/or CPC, as described in sections 6.13.6.1 and 6.13.6.2, is eligible for timestamping or on-the-fly modification. Which qualified packets actually have their timestamps written to the timestamp FIFO or modified on-the-fly is determined by the packet's PTP message type and by the settings of the TS_FIFO_EN and TS_INSERT_EN register fields.

For PTP version 1 packets (versionPTP field=1), the control field is examined to determine the PTP message type. Message types 0 (Sync) and 1 (Delay_Req) are PTP event messages that normally are timestamped.

For PTP version 2 packets (versionPTP field = 2), the messageType field is examined to determine the PTP message type. Message types 0 to 7 are classified as PTP event messages that normally are timestamped. In 1588v2, the event message types are as follows:

- 0: Sync
- 1: Delay_Req
- 2: Pdelay_Req
- 3: Pdelay_Resp
- 4 to 7: Reserved for future PTP event message types

Note that only the control field (1588v1) or the messageType field (1588v2) is examined by the timestamping logic to determine message type. The PTP message type is not determined from UDP port number.

The TS_FIFO_EN register has individual timestamp enable bits for each event message type 0-7 for both the egress and ingress packet-flow directions. For example, egress Sync messages (message type 0) have their timestamps saved to the TS2 FIFO if the TS_FIFO_EN.EM0_EN bit is set. As another example, ingress Delay_Req messages (message type 1) have their timestamps saved to the TS3 FIFO if the TS_FIFO_EN.IM1_EN bit is set.

The TS_INSERT_EN register has similar enable bits for on-the-fly timestamp insertion, which is described in detail in section 6.13.7.

Each timestamper has an eight-entry FIFO. If a FIFO is used, software must be able to read timestamps out of the FIFO faster than the expected packet arrival frequency to avoid FIFO overflow. Timestamper FIFO overflow is signaled by latched status bit PTP_IR.TSn_OF and can generate an interrupt if configured to do so. PTP_SR.TSn_NE provides real-time FIFO empty/not-empty status. A transition from FIFO empty to FIFO not-empty is signaled by latched status bit PTP_IR.TSn_NE, which can also generate an interrupt if configured.

Note that timestamps are written to the timestamp FIFOs without considering whether the packets have FCS errors. System software must read and discard timestamps for packets that are discarded due to FCS error.

6.13.6.4 Source Port Identity Checking

For each ingress PTP message a 12-bit identify code is computed from the 10-byte sourcePortIdentity field (1588v2) or the messageType, sourceCommunicationTechnology and sourceUuid fields (1588v1). When UID_CHK.CHK_EN=1, the computed identity code must match the value stored in UID_CHK.UID for the message timestamp to be stored in the TS3 FIFO (if writing to the TS3 FIFO is enabled as described in section 6.13.6.3).

6.13.7 On-the-Fly Packet Modification

The MAX24288 can be configured to perform a variety of 1588-related packet modifications on-the-fly in both ingress (serial interface to parallel interface) and egress (parallel to serial) directions:

- Writing Ingress Timestamps Into Ingress PTP Messages (OC, BC, TC)
- Clearing Ingress Timestamps In Egress Messages (BC, TC)
- On-the-Fly Timestamping for Egress Messages (OC or BC)
- One-Step Transparent Clock Residence Time Correction (TC)
- On-the-Fly P2P Transparent Clock Correction for Ingress Link Delay (TC)
- On-the-Fly Correction Field Modification for Asymmetry, Latency (OC, BC, TC)
- Hardwired Ethernet and IP Address Checking and Overwriting

These on-the-fly packet modifications can be employed in 1588 nodes as shown in Table 6-23.

Table 6-23. One-Step/On-the-Fly Selection Matrix

	OC or BC MASTER	OC or BC SLAVE	P2P TC	E2E TC
Sync egress one-step timestamping	V1, V2			
Delay_Req egress one-step timestamping		M		
Pdelay_Req egress one-step timestamping	M	M	M	
Pdelay_Resp egress one-step timestamping	V2	V2	V2	
Sync ingress timestamp write		M		
Delay_Req ingress timestamp write	M			
Pdelay_Req ingress timestamp write	M	M	M	
Pdelay_Resp ingress timestamp write	M	M	M	
correctionField modification for residence time			V2	V2
correctionField modification for ingress link delay			V2	
correctionField modification for asymmetry	V2	V2	V2	V2

V1=1588v1; V2=1588v2; M=Microsemi method (1588v2 only)

The on-the-fly modifications available in the MAX24288 are discussed in the subsections below. When a packet is modified by any of these methods, the Ethernet FCS field is also modified to make a valid packet. In addition, for PTP-over-IPv4/UDP packets, the UDP checksum is set to zero (see Note 4 below). For PTPv2-over-IPv6/UDP packets, the two bytes of the UDP payload after the end of the PTP message (specified by 1588v2 Annex D for this purpose) are modified to make the UDP checksum correct.

Notes About One-Step/On-the-Fly Corrections:

1. If the FCS of an Ethernet frame is incorrect before the contents are modified by the MAX24288 then the FCS value is left unchanged (i.e. it remains incorrect).

2. 1588v1 does not specify two extra bytes of UDP payload after the end of the PTP message as 1588v2 does. On-the-fly packet modification cannot be supported for PTPv1 over IPv6/UDP since these two bytes aren't guaranteed to be present and therefore cannot be modified to make the UDP checksum correct.

3. If a PTP message has a correctionField value of 0x7FFF FFFF FFFF FFFF before on-the-fly correction then the value is left unchanged. If a computed correctionField value has a magnitude larger than can be represented by the correctionField data type then the correctionField is set to 0x7FFF FFFF FFFF FFFF.

4. If any on-the-fly packet modifications are enabled as described in the subsections below then the packet modifier changes the UDP checksum to 0 in any packet that is classified to be PTP-over-UDP/IPv4. This happens to each and every PTP-over-UDP/IPv4 packet regardless of whether it is eventually declared to be a qualified packet or not. All such packets are affected because the UDP checksum field appears so early in the packet that the packet classifier typically has not determined whether the packet is a qualified packet or not

before the UDP checksum field is scheduled to exit the packet modification buffer. Therefore, rather than causing wrong UDP checksums in some packets, all UDP checksums are set to zero. This action is allowed in 1588v2 section D.1. Note that this paragraph also applies to circuit emulation packets (see section 6.13.8) when PTPCR1.TOP_MODE=1.

5. To minimize packet modification errors during configuration register changes, PTPCR1.PC_PM_DIS should be set before modifying packet modifier (PM) configuration registers and then cleared after all registers changes have been made.

6. Packets in 1588v1 do not have correction fields. Therefore on-the-fly correctionField modification described in this section are not made to 1588v1 packets.

6.13.7.1 Writing Ingress Timestamps Into Ingress PTP Messages (OC, BC, TC)

The MAX24288 can be configured to write ingress timestamps into qualified ingress PTP messages on-the-fly. This removes the need for 1588 software to read ingress timestamps from the TS3 FIFO and match them to received messages. It also reduces MDIO or SPI bus transactions accordingly.

This capability can be enabled selectively for each PTP event message type using the TS_INSERT_EN.IMx_EN bits. When any of these bits is set, a 40-byte packet modification buffer and a corresponding fixed 40*8ns=320ns delay are added to the ingress data path.

The seconds and/or nanoseconds fields of the timestamp can be written into each PTP event message. The TS_INSERT.SEC_ENA field enables/disables writing seconds information and specifies whether 8 lsbs, 32 lsbs or all 48 bits are written. The TS_INSERT.SEC_OFF field specifies the offset in bytes from the start of the PTP message header to the point at which the seconds information is to be written. Similarly, the TS_INSERT.NSEC_ENA field enables/disables writing the nanoseconds field into the lower 30 bits of a 4-byte section of the PTP message starting at the byte offset specified by the TS_INSERT.NSEC_OFF field. The seconds and nanoseconds field are each written into the PTP message with the most significant byte closest to the beginning of the message.

In a 1588v2 application, the most straightforward way to use this feature is to write the nanoseconds portion of the timestamp into the 4-byte reserved field in the PTP common header at byte offset 16 and write the 8 lsbs of the seconds field into the reserved byte in the common header at byte offset 5.

6.13.7.2 Clearing Ingress Timestamps In Egress Messages (BC, TC)

When TS_INSERT_EN.CLR_TS_INS=1, the MAX24288 clears the bytes of egress PTP messages that were written with timestamp information at ingress as described in section 6.13.7.1. It does this by writing 0x00 to the bytes specified by the fields of the TS_INSERT register for all PTP messages regardless of message type. This is useful for transparent clock and boundary clock applications where the ingress timestamp information should not be propagated with the packet to the next system.

6.13.7.3 On-the-Fly Timestamping for Egress Messages (OC or BC)

The MAX24288 supports one-step 1588v2 Sync and Pdelay_Resp messages in the egress direction as described in 1588v2. It also supports one-step 1588v1 Sync messages. In both cases, one-step operation means that the hardware-generated timestamp for a message is written into the message on-the-fly as it is transmitted. In one-step operation, Follow-Up and Pdelay_Resp_Follow_Up messages are not needed and not used, and the twoStepFlag field in the Sync and Pdelay_Resp messages is set to 0 by system software when the packets are created.

In addition, the device also supports an on-the-fly method for timestamping egress 1588v2 Delay_Req and Pdelay_Req messages. This behavior is not specified in 1588v2, but it can be enabled and used on a slave port of an OC or BC (for Delay_Req messages) or any port of an OC, BC or TC (for Pdelay_Req messages) without any changes to any other 1588 network nodes. When one-step/on-the-fly support is enabled for all four event message types in the egress direction, the need for system software tor read egress timestamps from the TS2 FIFO is eliminated. If the MAX24288 is also configured to write ingress timestamps into ingress PTP messages as

described in section 6.13.7.1 then the need for system software to read either of the timestamp FIFOs is eliminated. This greatly reduces MDIO or SPI bus transactions between microprocessor and MAX24288.

The one-step/on-the-fly egress timestamping capability can be enabled selectively for each PTP event message type using the four TS_INSERT_EN.EMx_EN bits. When any of these bits is set, a 40-byte packet modification buffer and a corresponding fixed 40*8ns=320ns delay are added to the egress data path.

6.13.7.3.1 One-Step/On-the-Fly Operations on Egress Messages

The following bullets describe what the MAX24288 does when one-step/on-the-fly operation is enabled for each PTP event message type in the egress direction.

Standard one-step capabilities:

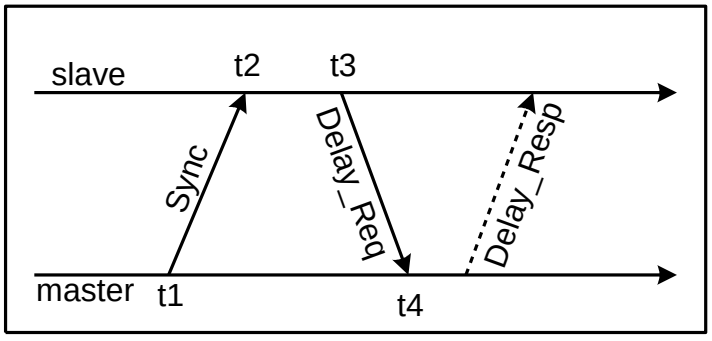
- **Sync: hardware timestamp → originTimestamp.** The egress hardware timestamp is written into the originTimestamp field in the Sync message. (1588v1 or v2)
- **Pdelay_Resp: correctionField + Pdelay_Resp_Egress_TS – Pdelay_Req_Ingress_TS → correctionField.** The most recent Pdelay_Req ingress hardware timestamp is subtracted from the egress Pdelay_Resp hardware timestamp and the result is added to the correctionField of the Pdelay_Resp message. The ingress hardware automatically saves the Pdelay_Req ingress timestamp to an internal register for this purpose. (1588v2 only)

Additional on-the-fly capabilities:

- **Delay_Req: correctionField + HW_TS lsbs → correctionField.** The MAX24288 generates a hardware timestamp (HW_TS) and adds the least significant four bits of seconds (converted to nanoseconds) and the nanoseconds of the HW_TS to the correctionField of the message. See section 6.13.7.3.2 below for a detailed explanation of how this works and how it can be used. (1588v2 only)
- **Pdelay_Req: correctionField + HW_TS lsbs → correctionField.** Same as Delay_Req above. See section 6.13.7.3.3 below for a detailed explanation of how this works and how it can be used. (1588v2 only)

6.13.7.3.2 On-the-Fly Delay_Req Mean Path Delay Calculation

Figure 6-15. Sync and Delay_Req Tmestamp Points



When on-the-fly egress Delay_Req timestamping is enabled, the overall concept is to convey the Delay_Req egress timestamp (t3 in Figure 6-15) from the MAX24288 to local 1588 software without storing t3 in a timestamp FIFO.

Normally, according to 1588v2, mean path delay is calculated as follows:

(1) <meanPathDelay> = [(t2 – t3) + (receiveTimestamp of Delay_Resp message(t4) – originTimestamp of Sync message(t1)) – correctionField of Sync message – correctionField of Delay_Resp message] / 2

After some rearrangement the calculation becomes:

(2) <meanPathDelay> = [t2 – originTimestamp of Sync message(t1) – correctionField of Sync message + receiveTimestamp of Delay_Resp message(t4) – (t3 + correctionField of Delay_Resp message)] / 2

Since t3 and "correctionField of Delay_Resp message" are added together in equation (2), and since the Delay_Resp correctionField is copied from the Delay_Req correctionField by the 1588 master, it is feasible to convey the t3 timestamp to the local 1588 software in the correctionField through the Delay_Req/Delay_Resp round trip. However, because the format of the correctionField is not large enough to hold an entire timestamp, the MAX24288 only adds the least significant four bits of seconds (converted to nanoseconds) and the nanoseconds of the t3 timestamp to the egress Delay_Req correctionField. Meanwhile, just before sending the Delay_Req message, the local 1588 software must save the upper 44 bits of t3 timestamp seconds as part of a software timestamp, SW_TS. This software timestamp must be accurate to within one second.

When the Delay_Resp message arrives, the local 1588 software performs these calculations:

convert the Delay_Resp correctionField into seconds (CF.sec) and nanoseconds (CF.ns)

seconds_diff =CF.sec – SW_TS.sec[3:0]

if seconds_diff < 0 then seconds_diff = seconds_diff + 16 /* to account for sec[3:0] roll-over */

t3_plus_Delay_Resp_CF.sec[47:0] = SW_TS.sec[47:0] + seconds_diff

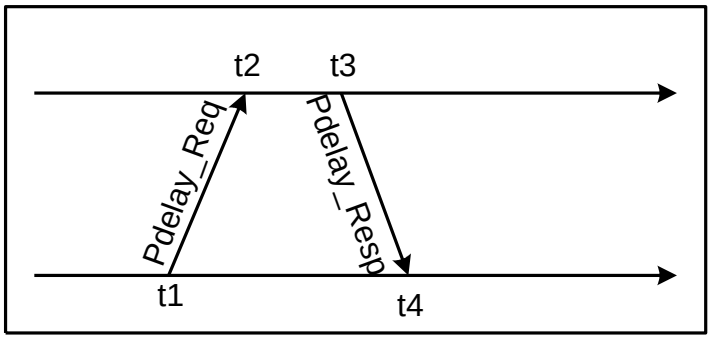
t3_plus_Delay_Resp_CF.ns = CF.ns

The value t3_plus_Delay_Resp_CF is then used in place of (t3 + correctionField of Delay_Resp message) in the meanPathDelay calculation in equation (2).

Note that using this method has no effect on the normal uses of the Delay_Req and Delay_Resp correctionFields. The 1588 slave can add values into the Delay_Req correctionField to correct for any known asymmetries. The remote 1588 master behaves normally as described in 1588v2 and copies the Delay_Req correctionField to the Delay_Resp correctionField and can add values into the Delay_Resp correctionField to correct for any known asymmetries in its data path as well. Furthermore, any transparent clocks in the round-trip path between the slave and master can add residence times and peer delays into the correctionField as required.

6.13.7.3.3 On-the-Fly PDelay_Req Mean Path Delay Calculation

Figure 6-16. Pdelay Timestamp Points



On-the-fly egress Pdelay_Req timestamping is similar to the on-the-fly egress Delay_Req timestamping described in section 6.13.7.3.2. In the discussion in this section, note that the timestamps are numbered differently than in section 6.13.7.3.2. Each section has numbering that matches numbering in the 1588v2 specification.

When on-the-fly egress Pdelay_Req timestamping is enabled, the overall concept is to convey the Pdelay_Req egress timestamp (t1 in Figure 6-16) from the MAX24288 to local 1588 software without storing t1 in a timestamp FIFO.

Normally, according to 1588v2, mean path delay is calculated as follows if the twoStepFlag=0 in the Pdelay_Resp message:

(1) <meanPathDelay> = [(t4 – t1) – correctionField of Pdelay_Resp message] / 2

After some rearrangement the calculation becomes:

(2) <meanPathDelay> = [t4 – (t1 + correctionField of Pdelay_Resp message)] / 2

Since t1 and "correctionField of Pdelay_Resp message" are added together in equation (2), and since the Pdelay_Resp correctionField is copied from the Pdelay_Req correctionField by the 1588 link partner, it is feasible to convey the t1 timestamp from the MAX24288 to the local 1588 software in the correctionField through the Pdelay_Req/Pdelay_Resp round trip. However, because the format of the correctionField is not large enough to hold an entire timestamp, the MAX24288 only adds the least significant four bits of seconds (converted to nanoseconds) and the nanoseconds of the t1 timestamp to the egress Pdelay_Req correctionField. Meanwhile, just before sending the Pdelay_Req message, the 1588 software must save the upper 44 bits of t1 timestamp seconds as part of a software timestamp, SW_TS. This software timestamp must be accurate to within one second.

When the Pdelay_Resp message arrives, the 1588 software performs these calculations:

```
convert the Pdelay_Resp correctionField into seconds (CF.sec) and nanoseconds (CF.ns)

seconds_diff =CF.sec – SW_TS.sec[3:0]

if seconds_diff < 0 then seconds_diff = seconds_diff + 16   /* to account for sec[3:0] roll-over */

t1_plus_Pdelay_Resp_CF.sec[47:0] = SW_TS.sec[47:0] + seconds_diff

t1_plus_Pdelay_Resp_CF.ns = CF.ns
```

The value t1_plus_Pdelay_Resp_CF is then used in place of (t1 + correctionField of Pdelay_Resp message) in the meanPathDelay calculation in equation (2).

Note that using this method has no effect on the normal uses of the Pdelay_Req and Pdelay_Resp correctionFields. The 1588 node can add values into the Pdelay_Req correctionField to correct for any known asymmetries. The 1588 link partner behaves normally as described in 1588v2 and copies the Pdelay_Req correctionField to the Pdelay_Resp correctionField and can add values into the Pdelay_Resp correctionField to correct for any known asymmetries in its data path as well.

The discussion above covers the case where twoStepFlag=0 in the Pdelay_Resp message. This method can be shown to work similarly when twoStepFlag=1 in the Pdelay_Resp message.

6.13.7.4 One-Step Transparent Clock Residence Time Correction (TC)

The residence time correction behavior described in this section supports residence times of less than or equal to 1 second.

Both end-to-end and peer-to-peer transparent clocks must timestamp PTP messages at ingress and egress in order to measure and report the residence time of the message in the transparent clock. This residence time is used by 1588 slaves (OC and BC) to cancel out the delay caused by the transparent clock. When each transparent clock in a 1588 path accurately reports the residence time of each packet, 1588 slaves can cancel out all queuing related packet delay for each packet and can therefore cancel out most packet delay variation (PDV). As a result, time transfer using 1588 can be much more accurate, slave PDV filtering algorithms can be simplified, and the

slave's master oscillator can be less expensive than would be the case if the intermediate nodes in the network were not transparent clocks.

Since residenceTime = egressTimestamp – ingressTimestamp, one straightforward way to report residence time is to manipulate the correctionField of each message so that hardware at the ingress port calculates correctionField = correctionField – ingressTimestamp on-the-fly, and hardware at the egress port calculates correctionField = correctionField + egressTimestamp on-the-fly.

The MAX24288 can be configured to do these on-the-fly corrections by setting TS_INSERT_EN.TC_CF_EN=1. By default MAX24288 only corrects Sync and Delay_Req event messages in this mode. These are the event message types that require residence time correction according to sections 11.5.2.1 and 11.5.3.2 of 1588v2. However, when TS_INSERT_EN.TC_CF_PD=1, MAX24288 also corrects Pdelay_Req and Pdelay_Resp event messages as described in section 11.5.4.2 of 1588v2. When TS_INSERT_EN.TC_CF_EN is set, a 40-byte packet modification buffer and a corresponding fixed 40*8ns=320ns delay are added to the egress data path and to the ingress data path.

Since a correctionField is not large enough to have an entire timestamp added in or subtracted out, the MAX24288 makes use of the nanoseconds and the least significant four bits of seconds from each timestamp. At ingress the nanoseconds field of the ingress timestamp is subtracted from the message's correctionField. At egress the nanoseconds field of the egress timestamp is added to the message's correctionField. To detect the case where the TC's local time crosses a one-second boundary after the generation of the ingress timestamp but before the generation of the egress timestamp, the ingress port hardware forwards four lsbs of ingress timestamp seconds to the egress port hardware. The egress port hardware compares the seconds lsbs of the ingress timestamp and the egress timestamp and modifies the correctionField appropriately when a one-second boundary crossing is detected.

The four lsbs of ingress timestamp seconds are conveyed in the four msbs of the correctionField. Since these four msbs represent corrections > 17,592 seconds, it is expected that they will never convey actual correction field information in real networks and therefore can be used temporarily intra-system for this purpose. The egress port hardware restores the four msbs of the correctionField by sign-extending the correctionField from bit 59 through bit 63.

The ingress and egress hardware have error checking mechanisms that work as follows:

- Ingress or egress: If the correctionField is maximum positive or negative value then it isn't modified.
- Ingress: If after subtracting ingress nanoseconds the upper 5 bits of the correctionField are not all ones or all zeroes then the correctionField is set to the maximum value.
- Egress: If residence time is greater than 1 second then the correctionField is set to the maximum value.
- Egress: If the modified correctionField is greater than maximum value then set it to the maximum value.

Downstream 1588 slaves should discard messages where the correctionField is set to the maximum value 0x7FFF FFFF FFFF FFFF (see 1588v2 section 13.3.2.7).

6.13.7.5 On-the-Fly P2P Transparent Clock Correction for Ingress Link Delay (TC)

A peer-to-peer transparent clock exchanges Pdelay_Req and Pdelay_Resp messages with a neighboring 1588 node to calculate the link delay (i.e. meanPathDelay) between the TC and its neighbor. A one-step P2P TC then has the responsibility to add the meanPathDelay to the correctionField of ingress Sync messages. This is done in MAX24288 hardware on-the-fly by adding the value in the

MEAN_PATH_DELAY register to the correctionField of ingress Sync packets. There is no enable/disable bit for this function. To disable this function when the system is not operating as a P2P TC, set the

MEAN_PATH_DELAY register value to 0. When

MEAN_PATH_DELAY≠0, a 40-byte packet modification buffer and a corresponding fixed 40*8ns=320ns delay are added to the ingress data path.

The mean path delay function can also be used in other ways. For example, in a network without P2P TCs where meanPathDelay is calculated from Delay_Req/Delay_Resp packet exchange, a 1588 slave node can write the calculated meanPathDelay to the

MEAN_PATH_DELAY register where it is automatically added to the correctionField of incoming Sync packets. The function is also available for other correctionField modification uses as needed.

6.13.7.6 On-the-Fly Correction Field Modification for Asymmetry, Latency (OC, BC, TC)

1588 nodes are required to correct for any known asymmetries between the egress direction and the ingress direction of a port (155v2 section 11.6). Nodes are also required to correct for egress latency and ingress latency as defined in 1588v2 section 7.3.4. These corrections can be done in MAX24288 hardware on-the-fly for any combination of event message types in the egress direction, ingress direction or both. To enable this capability, an adjustment value is written to one of the three CF_COR registers and then the appropriate fields of the CF_INGRESS and CF_EGRESS registers are configured to enable corrections for the desired event message types. The CF_COR registers are 2s-complement, and their resolution is 0.25ns. The CF_COR value is *added* to the correctionField for *ingress* PTP messages and is *subtracted* from the correctionField for *egress* PTP messages as called for in 1588v2 section 11.6. When any of the CF_INGRESS fields are nonzero, a 40-byte packet modification buffer and a corresponding fixed 40*8ns=320ns delay are added to the ingress data path..

Any of the CF_COR, CF_INGRESS and CF_EGRESS registers can also be used as needed to modify correctionFields on-the-fly for reasons other than asymmetry or latency.

6.13.7.7 Hardwired Ethernet and IP Address Checking and Overwriting

In some applications, in the ingress path there is a need to change multicast Ethernet and IP addresses to unicast so that simple layer 2 Ethernet switches can be used as PTP enabled switches, in particular peer-to-peer transparent clock switches.

Table 6-24 shows the Ethernet, IPv4 and IPv6 multicast addresses used for PTP and the corresponding overwrite enable bits in the PTPCR2 register. For the PTPOE or P2POE overwrite bits, If the bit is set and an ingress Ethernet destination address (DA) matches the corresponding Ethernet DA in Table 6-24 then the DA is overwritten with a unicast DA. When one or more of the PTPCR2 enable bits is nonzero, a 40-byte packet modification buffer and a corresponding fixed 40*8ns=320ns delay are added to the ingress data path.

For the five IP overwrite enable bits in Table 6-24, if the bit is set and an ingress Ethernet destination address (DA) matches the corresponding Ethernet DA in Table 6-24 then the Ethernet DA is overwritten with a unicast DA. In addition, if the packet is an IPv4 packet and PKT_CLASS.UDP_IPv4=1 and the ingress packet IPv4 DA corresponds with the Ethernet DA as shown in Table 6-24 then the IP DA is also overwritten and the IPv4 header checksum is recalculated. Similarly, if the packet is an IPv6 packet and PKT_CLASS.UDP_IPv6=1 and the ingress packet IPv6 DA corresponds with the Ethernet DA as shown in Table 6-24 then the IP DA is also overwritten.

The IETF mapping of IPv4 multicast addresses to Ethernet multicast addresses has 32 possible IPv4 multicast addresses mapped to each Ethernet multicast address. Therefore it is possible for a network to have another active IPv4 multicast address that maps to one of the Ethernet multicast addresses used for PTP. This situation must be avoided. To help detect it, the latched status bit PTP_SR.OVW_ERR is set when PKT_CLASS.UDP_IPv4=1 and either the Ethernet or IP address matches an enabled address in Table 6-24 but the other address does not correspond. When PKT_CLASS.UDP_IPv6=1, similar checking is done for IPv6 packets.

All 32 bits of the IPv4 DA are overwritten. For IPv6 only the upper 16 bits and lower 48 bits are overwritten. The upper 16 bits are overwritten with 0x0000 to make it a unicast IP address.

In MAC-in-MAC packets, only the innermost Ethernet DA is checked and overwritten as described above. Enabled IP DA overwrites are only done when the innermost Ethernet DA matches and is overwritten. Since this DA check/overwrite function would typically be used at network end points after all outer Ethernet frames have been removed, this function typically would be enabled in end-points but disabled in aggregation and transport equipment where MAC-in-MAC packets may be present.

The registers for the first three programmable matching criteria of the Configurable Protocol Classifier (section 6.13.6.2) are used as the source of the unicast Ethernet and IP addresses used to overwrite multicast addresses (see Table 6-25). These three CPC criteria must be disabled for use by the CPC (CFG_OFFSET.ENABLE=0) in order to be used instead for this overwrite function.

When a packet is modified, the Ethernet FCS is also modified to make a valid packet. For PTP-over-IPv4/UDP packets, the UDP checksum is always set to zero when any of the five IP overwrite enable bits in Table 6-24 are set to 1. For PTPv2-over-IPv6/UDP packets, the two bytes of the UDP payload after the end of the PTP message (specified by 1588v2 Annex D for this purpose) are modified to make the UDP checksum correct. See also Notes About One-Step/On-the-Fly Corrections in section 6.13.7.

Table 6-24. Ethernet and IP Multicast Addresses to Check

Enable Bit in PTPCR2	Ethernet DA	IPv4 DA	IPv6 DA	Description
PTPOE	01-1B-19-00-00-00	--	--	PTP over Ethernet
P2POE	01-1B-C2-00-00-0E	--	--	PTP Peer-to-Peer over Ethernet
PTPOIP	01-00-5E-00-01-81	224.0.1.129	FF0X:0:0:0:0:0:0:0181	PTP primary over IP
PTP1OIP	01-00-5E-00-01-82	224.0.1.130	FF0X:0:0:0:0:0:0:0182	PTP alternate-1 over IP
PTP2OIP	01-00-5E-00-01-83	224.0.1.131	FF0X:0:0:0:0:0:0:0183	PTP alternate-2 over IP
PTP3OIP	01-00-5E-00-01-84	224.0.1.132	FF0X:0:0:0:0:0:0:0184	PTP alternate-3 over IP
P2POIP	01-00-5E-00-00-6B	224.0.0.107	FF02:0:0:0:0:0:0:006B	PTP Peer-to-Peer over IP

Table 6-25. Source of Unicast Addresses to Overwrite Multicast

Bit to Overwrite	Source of Bits
Ethernet DA[15:0]	CPC Criterion 1 Mask[15:0]
Ethernet DA[31:16]	CPC Criterion 2 Mask[15:0]
Ethernet DA[47:32]	CPC Criterion 3 Mask[15:0]
IPv4 DA[15:0]	CPC Criterion 1 Match[15:0]
IPv4 DA[31:16]	CPC Criterion 2 Match[15:0]
IPv6 DA[15:0]	CPC Criterion 1 Match[15:0]
IPv6 DA[31:16]	CPC Criterion 2 Match[15:0]
IPv6 DA[47:32]	CPC Criterion 3 Match[15:0]

6.13.8 Circuit Emulation Timestamping for Adaptive Clock Recovery

The MAX24288 1588 hardware can be configured to support clock recovery over adaptive mode circuit emulation (see IETF RFCs 4553, 5086 and 5087 and Metro Ethernet Forum document MEF 8). Circuit emulation is typically used to carry PDH TDM signals such as DS1s and E1s over a packet network. The adaptive mode type of circuit emulation is used when a system that recovers TDM signals from the packet network does not have access to a high-quality network clock. When this is the case, adaptive clock recovery (ACR) is used to recover the clock of the TDM signal from the packets. The MAX24288 provides the necessary hardware to enable ACR.

To support ACR the MAX24288 can be configured to timestamp each ingress circuit emulation packet and capture the sequence number of the packet. In addition, the PEG1 programmable event generator can be configured to generate a feedback clock with a frequency equal to the nominal expected packet arrival rate (or an integer multiple thereof). This feedback clock is optionally divided down and then timestamped by timestamper TS1. The difference between packet timestamps and feedback timestamps can be used as error information analogous to the output of a phase-frequency-detector in a PLL. System software and the MAX24288's 1588 time engine (section 6.13.1) can be used to complete the PLL loop required to perform ACR. To do this, software implements a digital loop filter and any other DSP operations required and steers the frequency of the 1588 time engine, which performs the role of the VCO in the PLL.

Adaptive clock recovery (also known as Timing over Packet mode or TOP mode) is enabled in the MAX24288 by setting PTPCR1.TOP_MODE=1. In this mode, PEG1 (see section 6.13.3) is typically configured to generate a 1.544MHz (DS1) or 2.048MHz (E1) clock. If the PEG1 frequency is a multiple of the nominal expected packet

arrival rate, one or both of the TS1 dividers are configured (TS1_DIV1 and TSCR.TS1_DIV2) to divide down the feedback clock to the expected rate.

In TOP mode the packet classifier (section 6.13.6) must be configured to qualify circuit emulation packets. This is typically accomplished by using the hardwired packet classifier for transport layer packet headers and the configurable packet classifier for the circuit emulation messages. In TOP mode, the offset from the CPC start position to the start of the sequence number (in the circuit emulation control word) must be specified in the PTP_OFFSET register. PTP_OFFSET is configured by writing the offset to the CFG_OFFSET register and then writing the CFG_WR register with CFG_SEL=1111 and WR=1.

To convey the packet and PEG1 feedback timestamps to system software, the timestamps can be put into the TS3 and TS1 FIFOs, respectively, or they can be inserted into the ingress packet on-the-fly.

Writing Timestamps to FIFOs. The ingress packet timestamp and sequence number are written into the TS3 FIFO when TSCR.TS3SRC_SEL=0. The PEG1 feedback timestamp is written into the TS1 FIFO when TSCR.TS1SRC_SEL=0. By default every ingress timestamp is written to the TS3 FIFO. However, when PTPCR1.TS3_FIFO=1, only one entry can be written to the TS3 (ingress) FIFO after each timestamp is written to the TS1 (PEG1 feedback) FIFO. This reduces the number of FIFO entries that system software must process.

Writing Timestamps to the Ingress Packet. When TS_INSERT_EN.TOP_EN=1 and TS_INSERT. NSEC_ENA=1, the ingress packet timestamp and the most recent PEG1 feedback timestamp are written into each qualified TOP ingress packet on-the-fly. The data is written into the packet at the byte offset specified by TS_INSERT.NSEC_OFF. In total, 10 bytes are written into the packet: 4 bytes of packet timestamp nanoseconds, then 4 bytes of PEG1 feedback timestamp nanoseconds then a two-byte field to make the UDP checksum correct in IPv6/UDP packets. Each field is written most-significant byte first.

6.14 Data Path Latencies

The MAX24288 data path latencies are shown in Table 6-26 below. These latencies exceed the full-duplex delay constraints in 802.3 Table 36-9b. Therefore MAX24288 may not be compatible with PAUSE operation as specified in 802.3 Clause 31.

Table 6-26. GMII Data Path Latencies

Event	Without Packet Modification Buffer		W/ 40-Byte Packet Modification Buffer		802.3 Max, ns (#bit times)
	Min, ns	Max, ns	Min, ns	Max, ns	
TX_EN=1 sampled to 1st bit of /S/ on TDP/TDN	119	121	431	433	108.8 (136)
1st bit of /T/ on RDP/RDN to RX_DV deassert	211	216	531	536	153.6 (192)

6.15 Power Supply Considerations

Due to the multi-power-supply nature of the device, some I/Os have parasitic diodes between a 1.2V supply and a 3.3V supply. When ramping power supplies up or down, care must be taken to avoid forward-biasing these diodes because it could cause latchup. Two methods are available to prevent this. The first method is to place a Schottky diode external to the device between the 1.2V supply and the 3.3V supply to force the 3.3V supply to be within one parasitic diode drop of the 1.2V supply. The second method is to ramp up the 3.3V supply first and then ramp up the 1.2V supply.

6.16 Startup Procedure

MAX24288 requires the following start-up procedure after power-up for proper operation:

1. Assert RST_N (low) for at least 100μs after the power supplies have ramped up and are stable then deassert RST_N.
2. Write 0x0012 to the PAGESEL register. This disables the SPI interface and selects MDIO page 2.
3. Set the PTPCR1.RX_PWDN bit to power-down the receive CDR.
4. Wait 1ms.
5. Clear the PTPCR1.RX_PWDN bit.
6. Set the BMCR.DP_RST bit to reset the datapath. This bit is self-clearing.

The device can then be further configured as needed.

7. Register Descriptions

The PHY registers (Table 7-1) can be accessed only through the MDIO interface, which is part of the parallel MII interface. The IEEE1588 registers (Table 7-2) can be accessed either through the MDIO interface or through the SPI interface, depending on the setting of the PAGESEL.SPI_DIS register bit. When the SPI interface is enabled, the MDIO interface can access only the Table 7-1 registers, the SPI interface can access only the Table 7-2 registers, and accesses over these interfaces to these register spaces can be simultaneous. When the SPI interface is disabled, All registers are accessed through the MDIO interface.

Through the MDIO interface, registers at addresses 16 to 30 are paged using the PAGESEL.PAGE register field to allow register map expansion to more than 32 registers. MDIO register addresses 0 to 15 and 31 are not paged and remain the same regardless of the value of PAGESEL.PAGE. Through the SPI interface, all 1588 registers can be accessed at the SPI addresses shown in Table 7-2. Nonexistent registers are not writable and read back as high impedance through the MDIO port and 0x0000 through the SPI port.

7.1 Register Map

Table 7-1. PHY Register Map (MDIO Only)

MDIO Register Address	Register Name	R/W
0	BMCR	RW
1	BMSR	RO
2	ID1	RO
3	ID2	RO
4	AN_ADV	RW
5	AN_RX	RO
6	AN_EXP	RO
7	--	--
8	--	--
9	--	--
10	--	--
11	--	--
12	--	--
13	--	--
14	--	--
15	EXT_STAT	RO
P0.16	JIT_DIAG	RW
P0.17	PCSCR	RW
P0.18	GMIICR	RW
P0.19	CR	RW
P0.20	IR	RW
P0.21	--	--
P0.22	--	--
P0.23	--	--
P0.24	--	--
P0.25	--	--
P0.26	--	--
P0.27	--	--
P0.28	--	--
P0.29	--	--
P0.30	--	--
31	PAGESEL	RW

Table 7-2. 1588 Register Map (MDIO or SPI)

Address		Register Name	R/W
MDIO	SPI		
P1.16	0	ID	RO
P1.17	1	GPIOCR1	RW
P1.18	2	GPIOCR2	RW
P1.19	3	GPIOSR	RO
P1.20	4	PTP_IR	RO
P1.21	5	PTP_IE	RW
P1.22	6	PTP_SR	RO
P1.23	7	HDR_DAT1	RO
P1.24	8	HDR_DAT2	RO
P1.25	9	TEIO1	RW
P1.26	10	TEIO2	RW
P1.27	11	TEIO3	RW
P1.28	12	TEIO4	RW
P1.29	13	TEIO5	RW
P1.30	14	TERW	RW
---	15	--	--
P2.16	16	PTPCR1	RW
P2.17	17	PTPCR2	RW
P2.18	18	TSCR	RW
P2.19	19	PEGCR	RW
P2.20	20	TS1_DIV	RW
P2.21	21	TS_FIFO_EN	RW
P2.22	22	TS_INSERT_EN	RW
P2.23	23	TS_INSERT	RW
P2.24	24	CF_INGRESS	RW
P2.25	25	CF_EGRESS	RW
P2.26	26	PTPCR3	RW
P2.27	27	--	--
P2.28	28	UID_CHK	RW
P2.29	29	--	--
P2.30	30	--	--
---	31	--	--
P3.16	32	PKT_CLASS	RW
P3.17	33	VLAN2_ID	RW
P3.18	34	--	--
P3.19	35	--	--
P3.20	36	MEF_ECID_HI	RW
P3.21	37	MEF_ECID_LO	RW
P3.22	38	MPLS_LABEL_HI	RW
P3.23	39	MPLS_LABEL_LO	RW
P3.24	40	ETYPE_ALT	RW
P3.25	41	UDP_SRC	RW
P3.26	42	UDP_DST	RW
P3.27	43	CFG_MASK	RW
P3.28	44	CFG_MATCH	RW
P3.29	45	CFG_OFFSET	RW
P3.30	46	CFG_WR	RW
---	47	PHY_MATCH	RW

7.2 Direct Access Registers

The register operating type is described in the “R/W” column using the following codes:

Type	Description
RW	**Read-Write.** Register field can be written and read back.
RO	**Read Only**. Register field can only be read; writing it has no effect. Write 0 for future compatibility.
SC	**Self Clearing.** Register bit self clears to 0 after being written as 1
LH-E	**Latch High—Event.** Bit latches high when the internal event occurs and returns low when it is read.
LL-E	**Latch Low—Event.** Bit latches low when the internal event occurs and returns high when it is read.
LH-C	**Latch High—Condition.** Bit latches high when the internal condition is present. If the condition is still present when the bit is read then the bit stays high. If the condition is not present when the bit is read then the bit returns low.
LL-C	**Latch Low—Condition.** Bit latches low when the internal condition is present. If the condition is still present when the bit is read then the bit stays low. If the condition is not present when the bit is read then the bit returns high.

In the register definitions below, the register addresses (MDIO and SPI) are provided at the end of the table title. Addresses 16 to 30 are bank-switched by the PAGESEL.PAGE field as shown in Table 7-1. Addresses 0 to 15 and 31 are not bank-switched. The SPI register space is mapped to the MDIO register space using this formula; SPI address = (MDIO page – 1) * 16 + MDIO address – 16. For example an MDIO register mapped to page 2 address 17 would have the address (MDIO 2.17, SPI 17).

7.2.1 BMCR

Basic Mode Control Register (MDIO 0)

Bit(s)	Name	Description	R/W	Reset
15	DP_RST	Datapath Reset. This bit resets the entire datapath from parallel MII through PCS. Self-clearing. See section 6.3.1. 0 = normal operation 1 = reset	RW, SC	0
14	DLB	Diagnostic Loopback. Transmit data is looped back to the receive path. See block diagram in Figure 2-1 for the location of this loopback and see section 6.11 for additional details. 0 = Disable diagnostic loopback (normal operation) 1 = Enable diagnostic loopback	RW	0
13	Reserved	Ignore on Read	RO	0
12	AN_EN	Auto-negotiation enable. See section 6.8. 0 = Disable 1 = Enable	RW	Note 1
11:10	Reserved	Ignore on Read	RO	0
9	AN_START	Setting this bit causes a restart of the Auto-negotiation process. Self clears. See section 6.8.	RW, SC	0
8	Reserved	Ignore on Read	RO	0
7	COL_TEST	Collision test. When this bit is set, MAX24288 asserts the COL signal within 64 parallel interface transmit clock cycles after TX_EN is asserted and deasserts the COL signal within one parallel interface transmit clock cycle after TX_EN is deasserted. See 802.3 section 22.2.4.1.9. See section 6.11.1.	RW	0
6:0	Reserved	Ignore on Read	RO	0

Note 1: At reset when COL=1 or AN_EN is set to 1 else it is set to 0.

7.2.2 BMSR

Basic Mode Status Register (MDIO 1)

Bit(s)	Name	Description	R/W	Reset
15	Reserved	Ignore on read	RO	0
14	SPD100FD	100BASE-X Full Duplex capability. Always indicates 1 = able to do 100BASE-X full duplex	RO	1
13	SPD100HD	100BASE-X Half Duplex capability Always indicates 1 = able to do 100BASE-X half duplex	RO	1
12	SPD10FD	10Mb/s Full Duplex capability Always indicates 1 = able to do 10Mb/s full duplex	RO	1
11	SPD10HD	10Mb/s Half Duplex capability Always indicates 1 = able to do 10Mb/s half duplex	RO	1
10:9	Reserved	Ignore on read	RO	0
8	EXT_STAT	Extended Status information available Always indicates 1 = extended status information in EXT_STAT register	RO	1
7	Reserved	Ignore on Read	RO	0
6	MF_PRE	Management preamble suppression supported Always indicates 1 = MDIO preamble suppression is supported.	RO	1
5	AN_COMP	Auto-negotiation Complete 0 = Auto-negotiation not completed or not in progress 1 = Auto-negotiation has completed	RO	0
4	RFAULT	Remote Fault – Indicates presence of remote fault on link partner. For SGMII, remote fault is latched high when the ALOS input pin goes high or when the CDR indicates receive loss-of-lock. For 1000BASE-X, remote fault is RF≠00 in AN_RX. RFAULT is latched high when a remote fault is detected. If no remote fault is detected when RFAULT is read then RFAULT goes low. Otherwise RFAULT remains unchanged. IR.RFAULT is a read-only copy of this bit that can cause an interrupt when enabled. 0 = no remote fault has been detected since this bit was last read 1 = remote fault has occurred since this bit was last read	RO, LH-C	0
3	AN_ABIL	Auto-negotiation Ability Always indicates 1 = able to perform Auto-negotiation	RO	1
2	LINK_ST	Link Status – Indicates the status of the physical connection to the link partner. LINK_ST is latched low when the link goes down. If the PCS state machine is in the link-up state when LINK_ST is read then LINK_ST goes high. Otherwise LINK_ST remains unchanged. IR.LINK_ST is a read-only copy of this bit that can cause an interrupt when enabled. 0 = link down has occurred since this bit was last read 1 = link has been up continuously since this bit was last read	RO, LL-C	0
1	Reserved	Ignore on Read	RO	0
0	EXT_CAP	Extended Register Capability Always indicates 1 = the extended registers exist	RO	1

7.2.3 ID1 and ID2

Registers ID1 and ID2 are set to all zeroes as allowed by clause 22.2.4.3.1 of IEEE 802.3. The actual device ID can be read from the ID register.

Device ID 1 Register (MDIO 2)

Bit(s)	Name	Description	R/W	Reset
15:0	OUI_HI	OUI[3:18]	RO	0

Device ID 2 Register (MDIO 3)

Bit(s)	Name	Description	R/W	Reset
15:10	OUI_LI	OUI[19:24]	RO	0
9:4	MODEL	MODEL[5:0] Model Number	RO	0
3:0	REV	REV[3:0] Revision Number	RO	0

7.2.4 AN_ADV

The register contents are transmitted to the link partner's AN_RX register.

The fields of this register have different functions depending on whether the device is in 1000BASE-X auto-negotiation mode or in SGMII control information transfer mode. See section 6.8 for details.

Auto-Negotiation Advertisement Register (MDIO 4)

Bit(s)	Name	Description	R/W	Reset
15	AN_ADV[15] (NP_LK)	See section 6.8 for details.	RW	Note 1
14	AN_ADV[14]	Ignore on read	RO	0
13:0	AN_ADV[13:0]	See section 6.8 for details.	RW	Note 1

Note 1: The reset value of the bits of this register depend on the values of configuration pins at reset, as shown below. See section 6.1.

Configuration Pin Settings at Reset	Configuration Description	AN_ADV[15:0] Reset Value
COL=0, RX_DV=0	No auto-negotiation	0000 0000 0000 0000
COL=0, RX_DV=1, RXD[1]=0	15-pin config mode, SGMII 10 or 100Mbps	1001 0R00 0000 0001 (R = RXD[0] pin value)
COL=0, RX_DV=1, RXD[1]=1, GPO2=0	15-pin config mode, SGMII 1000Mbps	1001 1000 0000 0001
COL=0, RX_DV=1, RXD[1]=1, GPO2=1	15-pin config mode, 1000BASE-X	0000 0000 0010 0000
COL=1, GPO2=0	3-pin config mode, SGMII 1000Mbps	1001 1000 0000 0001
COL=1, GPO2=1	3-pin config mode, 1000BASE-X	0000 0000 0010 0000

7.2.5 AN_RX

The rx_Config_Reg[15:0] value received from the link partner is stored in this register.

This fields of this register have different functions depending on whether the device is in 1000BASE-X auto-negotiation mode or in SGMII control information transfer mode. See section 6.8 for details.

Auto-Negotiation Link Partner Ability Receive Register (MDIO 5)

Bit(s)	Name	Description	R/W	Reset
15	NP	See section 6.8 for details	RO	0
14	Acknowledge	1= link partner successfully received the transmitted base page.	RO	0
13:0	ABILITY	See section 6.8 for details	RO	0

7.2.6 AN_EXP

This register is used to indicate that a new link partner abilities page has been received.

Auto-negotiation Extended Status Register (MDIO 6)

Bit(s)	Name	Description	R/W	Reset
15:3	Reserved	Ignore on Read	RO	0
2	NP	Next Page capability - This device does not support next pages, it always reads as 0.	RO	0
1	PAGE	Page received, clears when read. See section 6.8. 0 = No new AN_RX page from link partner 1 = New AN_RX page from Link partner is ready	RO, LH-E	0
0	Reserved	Ignore on Read	RO	0

7.2.7 EXT_STAT

Extended Status Register (MDIO 15)

Bit(s)	Name	Description	R/W	Reset
15	1000X_FDX	1000BASE-X Full Duplex capability. Always indicates 1 = able to do 100BASE-X full duplex	RO	1
14	1000X_HDX	1000BASE-X Half Duplex capability. Always indicates 0 = not able to do 100BASE-X half duplex	RO	0
13:0	Reserved	Ignore on Read	RO	0

7.2.8 JIT_DIAG

Jitter Diagnostics Register (MDIO 0.16)

Bit(s)	Name	Description	R/W	Reset
15	JIT_EN	Jitter Pattern Enable. Enables jitter pattern to be transmitted instead of normal data on TDP/TDN. 0 = Transmit normal data patterns from PCS encoder 1 = Transmit jitter test pattern specified by JIT_PAT	RW	0
14:12	JIT_PAT	Jitter Pattern. Specifies the pattern to transmit. Includes standard patterns specified in IEEE 802.3 Annex 36A. **JIT_PAT** / **Pattern** 000 — Custom pattern defined in CUST_PAT[9:0] 001 — 802.3 Annex 36A.1 high frequency test pattern 1010101010… 010 — 802.3 Annex 36A.3 mixed frequency test pattern 1111101011 0000010100… 011 — Low frequency pattern 1111100000… 100 — "Random" jitter pattern 0011111010 1001001100 1010011100 1100010101… 101 — 802.3 Annex 36A.2 low frequency test pattern 1111100000… 110 — Reserved 111 — Reserved .	RW	0
11:10	Reserved	Write as 0, Ignore on Read	RO	0
9:0	CUST_PAT	CUST_PAT[9:0] Custom 10-bit repeating pattern used when JIT_PAT=000. The transmission order is LSB(0) to MSB(9).	RW	0

7.2.9 PCSCR

PCS Control Register (MDIO 0.17)

Bit(s)	Name	Description	R/W	Reset
15	Reserved	Ignore on read	RO	0
14	TIM_SHRT	When PCSCR.BASEX=1 (1000BASE-X mode), this bit can be used to shorten the link timer timeout from 10ms to 1.6ms. The normal 10ms setting is called for in the 802.3 standard. When BASEX=0 this bit is ignored. 0 = PCS link timer has normal timeout 1 = PCS link timer has 1.6ms timeout	RW	0
13	DYRX_DIS	Disable Receive PCS Running Disparity. 0 = Enable PCS receive running disparity (normal) 1 = Disable PCS receive running disparity	RW	0
12	DYTX_DIS	Disable Transmit PCS Running Disparity. 0 = Enable PCS transmit running disparity (normal) 1 = Disable PCS transmit running disparity	RW	0
11:7	Reserved	Write as 0, Ignore on Read	RO	0
6	WD_DIS	Disable auto-negotiation watchdog timer. See section 6.8. 0 = Restart auto-negotiation after 5 seconds if link not up after previous start or restart 1 = Disable auto-negotiation watchdog restart	RW	0
5	Reserved	Ignore on Read	RO	0
4	BASEX	Specifies 1000BASE-X PCS mode or SGMII PCS mode. See section 6.8. 0 = SGMII PCS mode selected, link timer = 1.6 ms 1 = 1000BASE-X PCS mode selected, link timer = 10 ms (or 1.6ms when PCSCR.TIM_SHRT=1)	RW	Note 1
3:2	Reserved	Write as 0, Ignore on Read	RO	0
1	TLB	Terminal Loopback. Transmit data is looped back to the receive path at the high-speed serial interface (TDP/TDN to RDP/RDN). See block diagram in Figure 2-1 for the location of this loopback and see section 6.11 for additional details. 0 = Disable terminal loopback (normal operation) 1 = Enable terminal loopback	RW	0
0	Reserved	Write as 1, Ignore on Read	RW	1

Note 1: At reset, if COL=1 or RXD[1] = 1 then BASEX is set to the value of the GPO2 pin, else BASEX is set to 0. In other words, in 3-pin configuration mode OR (15-pin configuration mode AND parallel interface is set to 1000Mbps) BASEX is set to the value of the GPO2 pin. Otherwise BASEX is set to 0.

7.2.10 GMIICR

GMII Interface Control Register (MDIO 0.18)

<table>
<tr><th>Bit(s)</th><th>Name</th><th>Description</th><th>R/W</th><th>Reset</th></tr>
<tr><td>15:14</td><td>SPD[1:0]</td><td>Selects parallel MII interface mode. See section 6.7.
<table>
<tr><th rowspan="2">SPD</th><th rowspan="2">Speed</th><th colspan="2">Bus mode</th></tr>
<tr><th>DDR=0</th><th>DDR=1</th></tr>
<tr><td>00</td><td>10 Mbps</td><td>MII</td><td>RGMII-10</td></tr>
<tr><td>01</td><td>100 Mbps</td><td>MII</td><td>RGMII-100</td></tr>
<tr><td>10</td><td>1000 Mbps</td><td>GMII</td><td>RGMII-1000</td></tr>
<tr><td>11</td><td>1000 Mbps</td><td colspan="2">reserved</td></tr>
</table></td><td>RW</td><td>Note 1</td></tr>
<tr><td>13</td><td>Reserved</td><td>Write as 0, ignore on Read</td><td>RO</td><td>0</td></tr>
<tr><td>12</td><td>DTE_DCE</td><td>DTE-DCE mode selection for MII bus mode. Used when SPD[1:0]=0x and DDR=0; ignored otherwise. See section 6.7.3.
1 = MII-DTE (MAX24288 on MAC side of MII, both RXCLK and TXCLK are inputs)
0 = MII-DCE (MAX24288 on PHY side of MII, both RXCLK and TXCLK are outputs)</td><td>RW</td><td>Note 2</td></tr>
<tr><td>11</td><td>DDR</td><td>Reduced pin count (RGMII) using double data rate, i.e. data sampling/updating on both edges of the clock. See the SPD[1:0] description above and section 6.7.
0 = MII or GMII bus mode
1 = RGMII bus mode</td><td>RW</td><td>Note 3</td></tr>
<tr><td>10</td><td>TXCLK_EN</td><td>In GMII and RGMII modes, the TXCLK pin is not used for parallel interface operation. The TXCLK_EN bit enables TXCLK to output a 125MHz clock from the TX PLL in those modes. TXCLK_EN is ignored in MII mode. See section 6.7.
0 = TXCLK pin is high impedance
1 = TXCLK pin outputs 125MHz clock</td><td>RW</td><td>Note 4</td></tr>
<tr><td>9:8</td><td>Reserved</td><td>Write as 0, ignore on Read</td><td>RO</td><td>0</td></tr>
<tr><td>7</td><td>Reserved</td><td>Write as 1, ignore on Read</td><td>RW</td><td>1</td></tr>
<tr><td>6:4</td><td>Reserved</td><td>Write as 0, ignore on Read</td><td>RO</td><td>0</td></tr>
<tr><td>3</td><td>REF_INV</td><td>REFCLK Invert control.
0 = Noninverted
1 = Inverted</td><td>RW</td><td>0</td></tr>
<tr><td>2:1</td><td>Reserved</td><td>Write as 0, ignore on Read</td><td>RO</td><td>0</td></tr>
<tr><td>0</td><td>RLB</td><td>Remote Loopback. Receive data is looped back to the transmit path. See block diagram in Figure 2-1 for the location of this loopback and see section 6.11 for additional details.
0 = Disable remote loopback (normal operation)
1 = Enable remote loopback</td><td>RW</td><td>0</td></tr>
</table>

Note 1: At reset if COL=0 then SPD[1:0] is set to the value on the RXD[1:0] pins, else SPD[1:0] is set to 10. In other words, in 15-pin configuration mode SPD[1:0] is set to the value on the RXD[1:0] pins. In 3-pin configuration mode SPD[1:0] is set to 10 (1000Mbps).

Note 2: At reset if COL=0 and RXD[1] = 0 the DTE_DCE bit is set to the value on the GPO2 pin, else DTE_DCE is set to 0. In other words, in 15-pin configuration mode when the parallel interface is configured for 10Mbps or 100Mbps the DTE_DCE bit is set to the value on the GPO2 pin. Otherwise the DTE_DCE bit is set to 0.

Note 3: At reset the DDR bit is set to the value on the CRS pin.

Note 4: At reset if COL=0 the TXCLK_EN bit is set to the value on the TXCLK pin, else TXCLK_EN is set to 1. In other words, in 15-pin configuration mode the TXCLK_EN bit is set to the value on the TXCLK pin. In 3-pin configuration mode TXCLK_EN is set to 1.

7.2.11 CR

Control Register (MDIO 0.19)

Bit(s)	Name	Description	R/W	Reset
15:13	Reserved	Write as 0, Ignore on Read	RO	0
12	DLBDO	Diagnostic Loopback Data Out. Set this bit to enable transmit data to be output on the serial interface during diagnostic loopback (i.e. when BMCR.DLB=1). See section 6.11.	RW	0
11	RLBDO	Remote Loopback Data Out. Set this bit to enable receive data to be output on the parallel interface during remote loopback (i.e. when GMIICR.RLB=1). See section 6.11.	RW	0
10	TLBDO	Terminal Loopback Data Out. Set this bit to enable transmit data (rather than zeros) to be output on the serial interface during terminal loopback (i.e. when PCSCR.TLB=1). See section 6.11.	RW	0
9	RCFREQ	Specifies which recovered clock frequency to output on a GPIO pin. See section 6.2. 0 = 25MHz 1 = 125MHz	RW	0
8	RCSQL	Set this bit to squelch the recovered clock on GPO2, GPIO2 and GPIO4-7 when any of several squelch conditions occur. See sections 6.2.1. 0 = Recovered clock output not squelched 1 = Recovered clock squelched when a squelch condition occurs	RW	0
7:6	Reserved	Write as 0, Ignore on Read	RO	0
5	TCLK_EN	Serial interface transmit clock output enable. See section 6.6. 0 = Disable TCLKP/TCLKN 1 = Enable TCLKP/TCLKN	RW	0
4:0	Reserved	Write as 0, Ignore on Read	RO	0

7.2.12 IR

This register contains both latched status bits and interrupt enable bits. When the latched status bit is active and the associated interrupt enable bit is set an interrupt signal can be driven onto one of the GPIO pins by configuring the GPIOCR1 register.

Interrupt Register 1 (MDIO 0.20)

Bit(s)	Name	Description	R/W	Reset
15:14	Reserved	Write as 0, Ignore on Read	RO	0
13	RFAULT_IE	Interrupt Enable for RFAULT. 0 = interrupt disabled 1 = interrupt enabled	RW	0
12	LINK_ST_IE	Interrupt Enable for LINK_ST. 0 = interrupt disabled 1 = interrupt enabled	RW	0
11	ALOS_IE	Interrupt Enable for ALOS. See section 6.6. 0 = interrupt disabled 1 = interrupt enabled	RW	0
10	PAGE _IE	Interrupt Enable for PAGE. See section 6.8. 0 = interrupt disabled 1 = interrupt enabled	RW	0
9	RLOL_IE	Interrupt Enable for RLOL. See section 6.6. 0 = interrupt disabled 1 = interrupt enabled	RW	0
8	RLOS_IE	Interrupt Enable for RLOS. See section 6.6. 0 = interrupt disabled 1 = interrupt enabled	RW	0
7:6	Reserved	Write as 0, Ignore on Read	RO	0
5	RFAULT	Remote Fault. This is a read-only copy of BMSR.RFAULT. An interrupt is generated when RFAULT=1 and RFAULT_IE=1.	RO	0
4	LINK_ST	Link Status. This is a read-only copy of BMSR.LINK_ST (active low). An interrupt is generated when LINK_ST=0 and LINK_ST_IE=1.	RO	0
3	ALOS	Analog Loss-of-Signal latched status bit. Set when ALOS input pin goes high. An interrupt is generated when ALOS=1 and ALOS_IE=1. See section 6.6. 0 = Defect not detected since last read 1 = Defect detected since last read	RO, LH-C	0
2	PAGE	This is a read-only copy of AN_EXP.PAGE. An interrupt is generated when PAGE=1 and PAGE_IE=1. See section 6.8. 0 = Defect not detected since last read 1 = Defect detected since last read	RO	0
1	RLOL	Receive CDR Loss-of-Lock latched status bit. Set when the CDR PLL loses lock. An interrupt is generated when RLOL=1 and RLOL_IE=1. See section 6.10.1. 0 = Defect not detected since last read 1 = Defect detected since last read	RO, LH-C	0
0	RLOS	Receive CDR Loss-of-Signal latched status bit. Set when the CDR block sees no transitions in the incoming signal for 24 consecutive bit times. See section 6.6. 0 = Defect not detected since last read 1 = Defect detected since last read	RO, LH-C	0

7.2.13 PAGESEL

This page select register is used to extend the MDIO register space by mapping 1 of 4 pages of 15 registers into the MDIO register addresses 16 to 30. PAGESEL also has configuration bits that set the SPI bus timing modes. It also has global interrupt source status bits. This register is available on all pages at MDIO register address 31. This register is not available through the SPI interface.

Page Register (MDIO 31, on all pages)

Bit(s)	Name	Description	R/W	Reset
15	TEST	Factory Test. Always write 0.	RW	0
14	PTP_IR	Interrupt from PTP_IR register status, set if any latched status and its associated enable bit are both active. See section 6.3.3. 0 = interrupt source not active 1 = interrupt source is active	RO	0
13	IR	Interrupt from IR register. This bit is set if any latched status bit and its associated interrupt enable bit are both active in the IR register. See section 6.3.3. 0 = interrupt source not active 1 = interrupt source is active	RO	0
12:8	Reserved	Ignore on Read	RO	0
7	SPISWAP	SPI address and data transmit/receive bit order. See section 6.4. 0 = Most significant bit first 1 = Least significant bit first	RW	0
6	CPHA	SPI data phase select. See section 6.4.	RW	0
5	CPOL	SPI polarity select. See section 6.4.	RW	0
4	SPI_DIS	SPI interface disable. See section 6.4. 0 = Enable SPI. Maps the IEEE1588 registers to SPI. 1 = Disable SPI. Maps the IEEE1588 registers to MDIO.	RW	0
3:2	Reserved	Ignore on Read	RO	0
1:0	PAGE[1:0]	Page selection for MDIO register addresses 16 to 30. See section 7. 00 = PHY extended register page 0 01 = 1588 extended register page 1 10 = 1588 extended register page 2 11 = 1588 extended register page 3	RW	00

7.2.14 ID

The ID register matches the JTAG device ID (lower 12 bits) and revision (all 4 bits).

Device ID Register (MDIO 1.16, SPI 0)

Bit(s)	Name	Description	R/W	Reset
15:12	REV	REV[3:0] Device revision number. Contact factory for value.	RO	Note 1
11:0	DEVICE	DEVICE[11:0] Device ID	RO	Note 1

Note 1: See Device Code in Table 8-2.

7.2.15 GPIOCR1

GPIO Control Register 1 (MDIO 1.17, SPI 1)

Bit(s)	Name	Description	R/W	Reset
15	RST	Global device reset. Pin states are sampled and used to set the default values of several register fields. See section 6.3.1. 0 = normal operation 1 = reset	RW, SC	0
14:12	GPO1_SEL[2:0]	GPO1 output pin mode selection. See Table 6-4.	RW	110
11:9	GPO2_SEL[2:0]	GPO2 output pin mode selection. See Table 6-5.	RW	110
8:6	GPIO1_SEL[2:0]	GPIO1 output pin mode selection. See Table 6-4.	RW	Note 1
5:3	GPIO2_SEL[2:0]	GPIO2 output pin mode selection. See Table 6-5.	RW	000
2:0	GPIO3_SEL[2:0]	GPIO3 output pin mode selection. See Table 6-4.	RW	000

Note 1: At reset if COL=0 and GPO1=1 the GPIO1_SEL bits are set to 100 (125MHz), else the bits are set to 000 (high impedance).

7.2.16 GPIOCR2

GPIO Control Register 2 (MDIO 1.18, SPI 2)

Bit(s)	Name	Description	R/W	Reset
15:14	Reserved	Ignore on Read	RO	0
13	GPIO47_LSC	GPIO4-7 Latched Status Control. This bit controls the behavior of latched status bits GPIO4L through GPIO7L in GPIOSR. See section 6.2. 0 = Set latched status bit when input goes low 1 = Set latched status bit when input goes high	RW	0
12	GPIO13_LSC	GPIO1-3 Latched Status Control. This bit controls the behavior of latched status bits GPIO1L through GPIO3L in GPIOSR. See section 6.2. 0 = Set latched status bit when input goes low 1 = Set latched status bit when input goes high	RW	0
11:9	GPIO7_SEL[2:0]	GPIO7 output pin mode selection. See Table 6-6.	RW	000
8:6	GPIO6_SEL[2:0]	GPIO6 output pin mode selection. See Table 6-6.	RW	000
5:3	GPIO5_SEL[2:0]	GPIO5 output pin mode selection. See Table 6-6.	RW	000
2:0	GPIO4_SEL[2:0]	GPIO4 output pin mode selection. See Table 6-6.	RW	000

7.2.17 GPIOSR

GPIO Status Register (MDIO 1.19, SPI 3)

Bit(s)	Name	Description	R/W	Reset
15	Reserved	Ignore on Read	RO	0
14	GPIO7L	GPIO7 latched status.. Set when the transition specified by GPIOCR2.GPIO47_LSC occurs on the GPIO7 pin. 0 = Transition did not occur since this bit was last read 1 = Transition did occur since this bit was last read	RO, LH-E	0
13	GPIO6L	GPIO6 latched status.. Set when the transition specified by GPIOCR2.GPIO47_LSC occurs on the GPIO6 pin. 0 = Transition did not occur since this bit was last read 1 = Transition did occur since this bit was last read	RO, LH-E	0
12	GPIO5L	GPIO5 latched status.. Set when the transition specified by GPIOCR2.GPIO47_LSC occurs on the GPIO5 pin. 0 = Transition did not occur since this bit was last read 1 = Transition did occur since this bit was last read	RO, LH-E	0
11	GPIO4L	GPIO4 latched status.. Set when the transition specified by GPIOCR2.GPIO47_LSC occurs on the GPIO4 pin. 0 = Transition did not occur since this bit was last read 1 = Transition did occur since this bit was last read	RO, LH-E	0
10	GPIO3L	GPIO3 latched status.. Set when the transition specified by GPIOCR2.GPIO13_LSC occurs on the GPIO3 pin. 0 = Transition did not occur since this bit was last read 1 = Transition did occur since this bit was last read	RO, LH-E	0
9	GPIO2L	GPIO2 latched status.. Set when the transition specified by GPIOCR2.GPIO13_LSC occurs on the GPIO2 pin. 0 = Transition did not occur since this bit was last read 1 = Transition did occur since this bit was last read	RO, LH-E	0
8	GPIO1L	GPIO1 latched status.. Set when the transition specified by GPIOCR2.GPIO13_LSC occurs on the GPIO1 pin. 0 = Transition did not occur since this bit was last read 1 = Transition did occur since this bit was last read	RO, LH-E	0
7	Reserved	Ignore on Read	RO	0
6	GPIO7	GPIO7 pin real time status. See section 6.2. 0 = Pin low 1 = Pin high	RO	0
5	GPIO6	GPIO6 pin real time status. 0 = Pin low 1 = Pin high	RO	0
4	GPIO5	GPIO5 pin real time status. 0 = Pin low 1 = Pin high	RO	0
3	GPIO4	GPIO4 pin real time status. 0 = Pin low 1 = Pin high	RO	0
2	GPIO3	GPIO3 pin real time status. 0 = Pin low 1 = Pin high	RO	0
1	GPIO2	GPIO2 pin real time status. 0 = Pin low 1 = Pin high	RO	0
0	GPIO1	GPIO1 pin real time status. 0 = Pin low 1 = Pin high	RO	0

7.2.18 PTP_IR

When the latched status bit is set and the associated interrupt enable bit in PTP_IE is set an interrupt signal can be driven onto one of the GPIO pins by configuring the GPIOCR1 register.

PTP Interrupt Register (MDIO Page 1.20, SPI 4)

Bit(s)	Name	Description	R/W	Reset
15	IR	Interrupt from IR register. Set if any latched status and its associated enable bit are both set. See section 6.3.3. 0 = interrupt source not active 1 = interrupt source is active	RO	0
14	PTP_IR	Interrupt from this register. Set if any latched status and its associated enable bit are both set. See section 6.3.3. 0 = interrupt source not active 1 = interrupt source is active	RO	0
13	TS3_OF	Timestamp 3 FIFO Overflow. Set when FIFO overflows. See sections 6.13.4 and 6.13.6.3. 0 = No overflow occurred since last read 1 = Overflow occurred	RO, LH-E	0
12	TS3_NE	Timestamp 3 FIFO Not Empty. Set when FIFO goes from empty to not empty. See sections 6.13.4 and 6.13.6.3. 0 = No transition occurred since last read 1 = Empty-to-not-empty transition occurred	RO, LH-E	0
11	TS2_OF	Timestamp 2 FIFO Overflow. Set when FIFO overflows. See sections 6.13.4 and 6.13.6.3. 0 = No overflow occurred since last read 1 = Overflow occurred	RO, LH-E	0
10	TS2_NE	Timestamp 2 FIFO Not Empty. Set when FIFO goes from empty to not empty. See sections 6.13.4 and 6.13.6.3. 0 = No transition occurred since last read 1 = Empty-to-not-empty transition occurred	RO, LH-E	0
9	TS1_OF	Timestamp 1 FIFO Overflow. Set when FIFO overflows. See sections 6.13.4 and 6.13.6.3. 0 = No overflow occurred since last read 1 = Overflow occurred	RO, LH-E	0
8	TS1_NE	Timestamp 1 FIFO Not Empty. Set when FIFO goes from empty to not empty. See sections 6.13.4 and 6.13.6.3. 0 = No transition occurred since last read 1 = Empty-to-not-empty transition occurred	RO, LH-E	0
7	Reserved	Ignore on Read	RO	0
6	P2FNF	PEG2 FIFO not full. See section 6.13.3. 0 = No transition occurred since last read 1 = Full-to-not-full transition occurred	RO, LH-E	0
5	P2SC	PEG2 Sequence Complete. Set when a repeat sequence in the PEG2 controller is complete. See section 6.13.3. 0 = No repeat sequence completed since last read 1 = Repeat sequence completed since last read	RO, LH-E	0
4	P2EC	PEG2 Event Complete. Set when a PEG2 event that is marked for status latching (PEG command FIFO bit 21=1) completes. See section 6.13.3. 0 = No marked event completed since last read 1 = Marked event completed since last read	RO, LH-E	0
3	P1FNF	PEG1 FIFO not full. See section 6.13.3. 0 = No transition occurred since last read 1 = Full-to-not-full transition occurred	RO, LH-E	0

Bit(s)	Name	Description	R/W	Reset
2	P1SC	PEG1 Sequence Complete. Set when a repeat sequence in the PEG1 controller is complete. See section 6.13.3. 0 = No repeat sequence completed since last read 1 = Repeat sequence completed since last read	RO, LH-E	0
1	P1EC	PEG2 Event Complete. Set when a PEG2 event that is marked for status latching (PEG command FIFO bit 21=1) completes. See section 6.13.3. 0 = No marked event completed since last read 1 = Marked event completed since last read	RO, LH-E	0
0	TAC	Time Adjustment complete. See section 6.13.1. 0 = No time adjustment completed since last read 1 = Time adjustment completed since last read	RO, LH-E	0

7.2.19 PTP_IE

The bits in this register are used to enable an interrupt to occur when the associated latched status bit in register PTP_IR is set. For each bit, 0=interrupt disabled; 1=interrupt enabled.

PTP Interrupt Enable Register (MDIO Page 1.21, SPI 5)

Bit(s)	Name	Description	R/W	Reset
15:14	Reserved	Ignore on Read	RO	0
13	TS3_OF	Interrupt Enable for TS3_OF	RW	0
12	TS3_NE	Interrupt Enable for TS3_NE	RW	0
11	TS2_OF	Interrupt Enable for TS2_OF	RW	0
10	TS2_NE	Interrupt Enable for TS2_NE	RW	0
9	TS1_OF	Interrupt Enable for TS1_OF	RW	0
8	TS1_NE	Interrupt Enable for TS1_NE	RW	0
7	Reserved	Ignore on Read	RW	0
6	P2FNF	Interrupt Enable for P2FNF	RW	0
5	P2SC	Interrupt Enable for P2SC	RW	0
4	P2EC	Interrupt Enable for P2EC	RW	0
3	P1FNF	Interrupt Enable for P1FNF	RW	0
2	P1SC	Interrupt Enable for P1SC	RW	0
1	P1EC	Interrupt Enable for P1EC	RW	0
0	TAC	Interrupt Enable for TAC	RW	0

7.2.20 PTP_SR

PTP Status Register (MDIO P1.22, SPI 6)

Bit(s)	Name	Description	R/W	Reset
15:10	Reserved	Ignore on Read	RO	0
9	TS3_NE	Timestamp 3 FIFO not empty real time status. See sections 6.13.4 and 6.13.6.3. 0 = Empty 1 = Not empty	RO	0
8	TS2_NE	Timestamp 2 FIFO not empty real time status. See sections 6.13.4 and 6.13.6.3. 0 = Empty 1 = Not empty	RO	0
7	TS1_NE	Timestamp 1 FIFO not empty real time status. See sections 6.13.4 and 6.13.6.3. 0 = Empty 1 = Not empty	RO	0
6	P2FF	PEG2 control FIFO full/not-full real time status. See section 6.13.3. 0 = Not full 1 = Full	RO	0
5:4	Reserved	Ignore on Read	RO	0
3	P1FF	PEG1 control FIFO full/not-full real time status. See section 6.13.3. 0 = Not full 1 = Full	RO	0
2:1	Reserved	Ignore on Read	RO	0
0	OVW_ERR	Indicates Ethernet or IP multicast address overwrite error. See section 6.13.7.7. 0 = No overwrite error since last read 1 = Overwrite error since last read	RO, LH-E	0

7.2.21 HDR_DAT1

Header Data 1 Register (MDIO P1.23, SPI 7)

Bit(s)	Name	Description	R/W	Reset
15:12	MSG	When a packet timestamp is read from one of the timestamp FIFOs into the TEIO registers, the message type is loaded into this field from the FIFO. The message type is the lower 4 bits of the control field from a 1588v1 message, or the 4-bit messageType field from a 1588v2 message. See section 6.13.5. When an input signal timestamp is read from one of the timestamp FIFOs, this field should be ignored by system software.	RO	0
11:0	UID	When a packet timestamp is read from one of the timestamp FIFOs into the TEIO registers, the 12-bit identity code is loaded into this field from the FIFO. See section 6.13.5. When an input signal timestamp is read from one of the timestamp FIFOs, this field should be ignored by system software.	RO	0

7.2.22 HDR_DAT2

Header Data 2 Register (MDIO P1.24, SPI 8)

Bit(s)	Name	Description	R/W	Reset
15:0	SEQ	When a packet timestamp is read from one of the timestamp FIFOs into the TEIO registers, the 16-bit PTP (or TDMoIP) sequenceID is loaded into this register from the FIFO. See section 6.13.5. When an input signal timestamp is read from one of the timestamp FIFOs, this field should be ignored by system software.	RO	0

7.2.23 TEIO1 – TEIO5

The time engine I/O registers TEIO1 – TEIO5 are used to (1) write time, period, and time adjustment information to the time engine (2) write commands to PEG1 and PEG2, (3) set values for mean path delay and correction field adjustment, (4) read the current time, and (5) read the timestamper FIFOs.

The values in these registers are copied to the indirect register(s) specified by TERW.WRSEL when write control bit TERW.WR is set. The last values written to the TEIO registers can be read back until the registers are overwritten by system software or by a read of indirect registers. Values are copied to the TEIO registers from the indirect register(s) specified by TERW.RDSEL when TERW.RD is set. The TEIO registers are mapped to indirect registers as detailed in Table 7-3 and Table 7-4 below.

Note that when TS3 is configured to timestamp ingress packets (TSCR.TS3SRC_SEL=0), when the TS3 timestamp FIFO is read into the TEIO registers, the HDR_DAT1 and HDR_DAT2 registers are simultaneously updated from the TS3 FIFO. Similarly, when TS2 is configured to timestamp egress packets (TSCR.TS2SRC_SEL=0), when the TS2 timestamp FIFO is read into the TEIO registers, the HDR_DAT1 and HDR_DAT2 registers are simultaneously updated from the TS2 FIFO. Also, when a timestamper is configured to timestamp input signal edges (TSCR.TS3SRC_SEL≠0), when the timestamp FIFO is read into the TEIO registers, bit 15 of TEIO5 indicates the polarity of the input signal edge: 0=falling, 1=rising. See section 7.3.

Table 7-3. TEIO Register Mapping to RDSEL Sources

RDSEL	Source	TEIO1	TEIO2	TEIO3	TEIO4	TEIO5
000	TIME	SEC[15:0]	SEC[31:16]	SEC[47:32]	NS[15:0]	00,NS[29:16]
001	TS1_FIFO	SEC[15:0]	SEC[31:16]	SEC[47:32]	NS[15:0]	EDGE, 0, NS[29:16]
010	TS2_FIFO	SEC[15:0]	SEC[31:16]	SEC[47:32]	NS[15:0]	EDGE, 0, NS[29:16]
011	TS3_FIFO	SEC[15:0]	SEC[31:16]	SEC[47:32]	NS[15:0]	EDGE, 0, NS[29:16]

Table 7-4. TEIO Register Mapping to WRSEL Destinations

WRSEL	Destination	TEIO1	TEIO2	TEIO3	TEIO4	TEIO5
0000	TIME	SEC[15:0]	SEC[31:16]	SEC[47:32]	NS[15:0]	NS[29:16]
0001	PERIOD	--	--	FRACNS[7:0], 8'h00	FRAC[23:8]	NS[7:0], FRAC[31:24]
0010	Time Adjust (PER_ADJ and ADJ_CNT)	--	COUNT[15:0]	FRAC[7:0], COUNT[23:16]	FRAC[23:8]	NS[7:0], FRAC[31:24]
0011	PEG1_FIFO	--	--	--	FIFO[15:0]	8'h00, 00, FIFO[21:16]
0100	PEG2_FIFO	--	--	--	FIFO[15:0]	8'h00, 00, FIFO[21:16]
1000	MEAN_PATH_DELAY Adjust	--	--	--	NS[13:0], FRAC[1:0]	0, NS[28:14]
1001	Correction Field Adjust 1 CF_COR1	--	--	--	NS[13:0], FRAC[1:0]	NS[29:14]
1010	Correction Field Adjust 2 CF_COR2	--	--	--	NS[13:0], FRAC[1:0]	NS[29:14]
1011	Correction Field Adjust 3 CF_COR3	--	--	--	NS[13:0], FRAC[1:0]	NS[29:14]

SEC=Seconds. NS=Nanoseconds. FRAC=Fractional nanoseconds.

Time Engine I/O Register 1 (MDIO P1.25, SPI 9)

Bit(s)	Name	Description	R/W	Reset
15:0	TEIO1	See Table 7-3 and Table 7-4.	RW	0

Time Engine I/O Register 2 (MDIO P1.26, SPI 10)

Bit(s)	Name	Description	R/W	Reset
15:0	TEIO2	See Table 7-3 and Table 7-4.	RW	0

Time Engine I/O Register 3 (MDIO P1.27, SPI 11)

Bit(s)	Name	Description	R/W	Reset
15:0	TEIO3	See Table 7-3 and Table 7-4.	RW	0

Time Engine I/O Register 4 (MDIO P1.28, SPI 12)

Bit(s)	Name	Description	R/W	Reset
15:0	TEIO4	See Table 7-3 and Table 7-4.	RW	0

Time Engine I/O Register 5 (MDIO P1.29, SPI 13)

Bit(s)	Name	Description	R/W	Reset
15:0	TEIO5	See Table 7-3 and Table 7-4.	RW	0

7.2.24 TERW

The TERW register is used to write the values of the TEIO registers into indirect registers as well as to read the values of indirect registers into the TEIO registers.

Read and write operations can be requested at the same time. When this is done the current TEIO register values are written to the register specified by WRSEL, and then the values of the registers specified by RDSEL are read into TEIO registers.

Time Engine Read/Write Register (MDIO P1.30, SPI 14)

Bit(s)	Name	Description	R/W	Reset
15	RD	Set this bit to read the values of the indirect registers specified by RDSEL into the TEIO registers.	RW, SC	0
14:11	RSVD	Ignore on Read	RO	0
10:8	RDSEL[2:0]	This field specifies the registers to be read into the TEIO registers when RD=1. See Table 7-3 for RDSEL decodes.	RW	0
7	WR	Set this bit to write the values of the TEIO registers into the registers specified by WRSEL.	RW, SC	0
6:4	Reserved	Ignore on Read	RO	0
3:0	WRSEL[2:0]	This field specifies the registers to be written from the TEIO registers when WR=1. See Table 7-4 for WRSEL decodes.	RW	0

Note: SC = self-clearing.

7.2.25 PTPCR1

PTP Control Register 1 (MDIO P2.16, SPI 16)

Bit(s)	Name	Description	R/W	Reset
15	TE_RST	Time Engine Reset. This bit resets the logic of the 1588 time engine and related blocks. Self clearing. See section 6.3.1. 0 = normal operation 1 = reset	RW, SC	0
14	PC_PM_DIS	Packet Classifier, Packet Modifier Disable. This bit disables all packet classification (section 6.13.6) and packet modification (section 6.13.7) logic. To minimize packet classification and modification errors, this bit should be set before modifying PC or PM configuration registers and then cleared after all registers changes have been made. If PC_PM_DIS goes high in the middle of a packet, the PC and PM continue with existing configuration until end-of-packet and then are disabled. After setting PC_PM_DIS, system software should wait the duration of a maximum size packet before modifying PC/PM registers. When PC_PM_DIS goes low, the PC and PM ignore any packet in progress and start using the new configuration at the next start-of-packet. 0 = Enabled 1 = Disabled	RW	1
13	TOP_MODE	Set to enable TOP mode. See section 6.13.8. 0 = PTP mode. PTP packets are processed. 1 = TOP mode. Circuit emulation packets are processed.	RW	0
12	TS3_FIFO	Set to enable gating of TS3 FIFO entries. Used for TOP mode. See section 6.13.8. 0 = Write all timestamps to the TS3 FIFO 1 = Write an entry to the TS3 FIFO only when a new timestamp has been written to the TS1 FIFO since the last TS3 FIFO entry was written.	RW	0
11:7	Reserved	Ignored on read	RO	0
6	VID2FIFO	VLAN ID to FIFO. See section 6.13.5. 0 = Write 12-bit hash code of packet's sourcePortIdentity to FIFO 1 = Write packet's outer VLAN ID to FIFO	RW	0
5	PLL_PWDN	TX PLL Power Down. Setting this bit powers down and bypasses the TX PLL. See section 6.3.2. 0 = Power-down disabled 1 = Power-down enabled	RW	0
4	DP_PWDN	Datapath Power Down. Setting this bit disables clocks to the parallel MII, PCS encoder and decoder, the Ethernet/IP address replacement logic and all other parallel datapath logic, both receive and transmit. See section 6.3.2. 0 = Power-down disabled 1 = Power-down enabled	RW	0
3	TX_PWDN	Serial Interface Transmit Power Down. Setting this bit powers down the transmit serializer and the TDP/TDN and TCLKP/TCLN output drivers. See section 6.3.2 and section 6.6. 0 = Power-down disabled 1 = Power-down enabled	RW	0
2	RX_PWDN	Serial Interface Receive Power Down. Setting this bit powers down the RDP/RDN inputs, the clock and data recovery PLL, and the de-serializer. See section 6.3.2 and section 6.6. 0 = Power Down Disabled 1 = Power Down Enabled	RW	0

Bit(s)	Name	Description	R/W	Reset
1	PKT_PWDN	Packet Classifier and Packet Modifier Power Down. Setting this bit disables clocks to the packet classifier and packet modifier logic, both ingress and egress. See section 6.3.2. 0 = Power-down disabled 1 = Power-down enabled	RW	0
0	TE_PWDN	Time Engine Power Down. Setting this bit disables clocks to the time engine, output clock generator, PEGs and timestampers. See section 6.3.2. 0 = Power-down disabled 1 = Power-down enabled	RW	0

7.2.26 PTPCR2

PTP Control Register 2 (MDIO P2.17, SPI 17)

Bit(s)	Name	Description	R/W	Reset
15	PTPOE	PTP over Ethernet - Multicast MAC DA. See section 6.13.7.7. 0 = Disable checking and overwriting 1 = Enable checking and overwriting	RW	0
14	P2POE	PTP Peer-to-peer over Ethernet Multicast MAC DA. See section 6.13.7.7. 0 = Disable checking and overwriting 1 = Enable checking and overwriting	RW	0
13	PTPOIP	PTP primary over IP - Multicast MAC DA and IP DA. See section 6.13.7.7. 0 = Disable checking and overwriting 1 = Enable checking and overwriting	RW	0
12	PTP1OIP	PTP alternate-1 over IP - Multicast MAC DA and IP DA. See section 6.13.7.7. 0 = Disable checking and overwriting 1 = Enable checking and overwriting	RW	0
11	PTP2OIP	PTP alternate-2 over IP - Multicast MAC DA and IP DA. See section 6.13.7.7. 0 = Disable checking and overwriting 1 = Enable checking and overwriting	RW	0
10	PTP3OIP	PTP alternate-3 over IP - Multicast MAC DA and IP DA. See section 6.13.7.7. 0 = Disable checking and overwriting 1 = Enable checking and overwriting	RW	0
9	P2POIP	PTP Peer-to-Peer over IP - Multicast MAC DA and IP DA. See section 6.13.7.7. 0 = Disable checking and overwriting 1 = Enable checking and overwriting	RW	0
8	CLKO_INV	PTP_CLKO Invert Control. See section 6.13.2. 0 = Non-inverted 1 = Inverted	RO	0
7:0	CLKO_DIV[7:0]	Sets the PTP_CLKO output frequency by dividing 125MHz by CLKO_DIV which can be 1 to 255. When CLKO_DIV = 0 the output clock is disabled. See section 6.13.2.	RW	0

7.2.27 TSCR

Timestamp Control Register (MDIO P2.18, SPI 18)

Bit(s)	Name	Description	R/W	Reset
15:14	TS3_EDGE[1:0]	Specifies which signal edge(s) should be timestamped by timestamper 3. When T3SRC_SEL=0 this field is ignored. See section 6.13.4. 00 = None 01 = Positive edge 10 = Negative edge 11 = Both edges	RW	0
13:12	TS3SRC_SEL[1:0]	Specifies the source of the signal going to the TS3 timestamper. See sections 6.13.4 and 6.13.5. 00 = Ingress packet start of packet 01 = GPIO1 10 = GPIO2 11 = GPIO3	RW	0
11:10	TS2_EDGE[1:0]	Specifies which signal edge(s) should be timestamped by timestamper 2. When T2SRC_SEL=0 this field is ignored. See section 6.13.4. 00 = None 01 = Positive edge 10 = Negative edge 11 = Both edges	RW	0
9:8	TS2SRC_SEL[1:0]	Specifies the source of the signal going to the TS2 timestamper. See section 6.13.4 and 6.13.5. 00 = Egress packet start of packet 01 = GPIO1 10 = GPIO2 11 = GPIO3	RW	0
7:6	TS1_EDGE[1:0]	Specifies which signal edge(s) should be timestamped by timestamper 1. See section 6.13.4. 00 = None 01 = Positive edge 10 = Negative edge 11 = Both edges	RW	0
5:4	TS1SRC_SEL[1:0]	Specifies the source of the signal going to the TS1 timestamper. The signal frequency is divided by the values specified by TS1_DIV1 and TSCR.TS1_DIV2 before going to the TS1 timestamper. See section 6.13.4. The PEG1 feedback signal is used in TOP mode (see section 6.13.8). 00 = PEG1 feedback signal 01 = GPIO1 10 = GPIO2 11 = GPIO3	RW	0
3	Reserved	Ignore on read	RO	0
2:0	TS1_DIV2[2:0]	Configures the second of two input signal dividers associated with timestamper 1.The second divider is configured by TS1_DIV1. See section 6.13.4. 000 = Do not divide 001 = Divide by 10 010 = Divide by 100 011 = Divide by 1000 100 = Divide by 2 101 = Divide by 80	RW	0

Bit(s)	Name	Description	R/W	Reset
		110 = Divide by 800 111 = Divide by 8000		

7.2.28 PEGCR

PEG Control Register (MDIO P2.19, SPI 19)

Bit(s)	Name	Description	R/W	Reset
15:7	Reserved	Ignore on read	RO	0
6	P2RST	A low-to-high transition of this bit causes programmable event generator 2 (PEG2) to reset. This empties the PEG2 command FIFO, resets the PEG2 control logic, and causes PEG2's output signal to be driven low. See section 6.13.3.	RW	0
5	P2DIS	PEG2 disable. This bit can be used to delay the processing of the commands written to the PEG2 FIFO. See section 6.13.3. 0 = enable processing new commands 1 = disable processing new commands	RW	0
4	P2RES	PEG2 resolution. See section 6.13.3. 0 = 1 nanosecond resolution 1 = 1/256 nanosecond resolution	RW	0
3	Reserved	Ignore on read	RO	0
2	P1RST	A low-to-high transition of this bit causes programmable event generator 1 (PEG1) to reset. This empties the PEG1 command FIFO, resets the PEG1 control logic, and causes PEG1's output signal to be driven low. See section 6.13.3.	RW	0
1	P1DIS	PEG1 disable. This bit can be used to delay the processing of the commands written to the PEG1 FIFO. See section 6.13.3. 0 = enable processing new commands 1 = disable processing new commands	RW	0
0	P1RES	PEG1 resolution. See section 6.13.3. 0 = 1 nanosecond resolution 1 = 1/256 nanosecond resolution	RW	0

7.2.29 TS1_DIV1

Timestamper 1 Divider 1 Register (MDIO P2.20, SPI 20)

Bit(s)	Name	Description	R/W	Reset
15:0	TS1_DIV1[15:0]	Configures the first of two input signal dividers associated with timestamper 1. TS1_DIV1=0 or 1 bypasses this divider (i.e. divide by 1). The second divider is configured by TSCR.TS1_DIV2. See section 6.13.4.	RW	0

7.2.30 TS_FIFO_EN

For these bits to have any effect, the timestamp FIFOs must be configured to accept ingress or egress packet timestamps using register TSCR bit fields TS2SRC_SEL (egress) and TS3SRC_SEL (ingress). See section 6.13.6.3.

Timestamp FIFO Enable Register (MDIO 2.21, SPI 21)

Bit(s)	Name	Description	R/W	Reset
15	EM7_EN	Enable egress message type 0x7 (Reserved) to TS2 FIFO	RW	0
14	EM6_EN	Enable egress message type 0x6 (Reserved) to TS2 FIFO	RW	0
13	EM5_EN	Enable egress message type 0x5 (Reserved) to TS2 FIFO	RW	0
12	EM4_EN	Enable egress message type 0x4 (Reserved) to TS2 FIFO	RW	0
11	EM3_EN	Enable egress message type 0x3 (Pdelay_Resp) to TS2 FIFO	RW	0
10	EM2_EN	Enable egress message type 0x2 (Pdelay_Req) to TS2 FIFO	RW	0
9	EM1_EN	Enable egress message type 0x1 (Delay_Req) to TS2 FIFO	RW	0
8	EM0_EN	Enable egress message type 0x0 (Sync) to TS2 FIFO	RW	0
7	IM7_EN	Enable ingress message type 0x7 (Reserved) to TS3 FIFO	RW	0
6	IM6_EN	Enable ingress message type 0x6 (Reserved) to TS3 FIFO	RW	0
5	IM5_EN	Enable ingress message type 0x5 (Reserved) to TS3 FIFO	RW	0
4	IM4_EN	Enable ingress message type 0x4 (Reserved) to TS3 FIFO	RW	0
3	IM3_EN	Enable ingress message type 0x3 (Pdelay_Resp) to TS3 FIFO	RW	0
2	IM2_EN	Enable ingress message type 0x2 (Pdelay_Req) to TS3 FIFO	RW	0
1	IM1_EN	Enable ingress message type 0x1 (Delay_Req) to TS3 FIFO	RW	0
0	IM0_EN	Enable ingress message type 0x0 (Sync) to TS3 FIFO	RW	0

7.2.31 TS_INSERT_EN

Timestamp Insertion Enable Register (MDIO 2.22, SPI 22)

Bit(s)	Name	Description	R/W	Reset
15	TC_CF_PD (Note 1)	Enables one-step transparent clock residence time correction for Pdelay_Req and Pdelay_Resp event messages. Ignored when TC_CF_EN=0. See section 6.13.7.4.	RW	0
14	CLR_TS_INS	When this bit is set the MAX24288 clears the bytes specified by the fields of the TS_INSERT register in egress messages. See section 6.13.7.2.	RW	0
13	TOP_EN	Enables insertion of ingress packet timestamp and PEG1 feedback timestamp into ingress circuit emulation messages. Ignored when PTPCR1.TOP_MODE = 0. See section 6.13.8.	RW	0
12	TC_CF_EN (Note 1)	Enables one-step transparent clock residence time correction. See section 6.13.7.4.	RW	0
11	EM3_EN (Note 1)	Enables one-step operation for egress type 0x3 (Pdelay_Resp) messages. See section 6.13.7.3.	RW	0
10	EM2_EN (Note 1)	Enables one-step operation for egress type 0x2 (Pdelay_Req) messages. See section 6.13.7.3.	RW	0
9	EM1_EN (Note 1)	Enables one-step operation for egress type 0x1 (Delay_Req) messages. See section 6.13.7.3.	RW	0
8	EM0_EN (Note 1)	Enables one-step operation for egress type 0x0 (Sync) messages. See section 6.13.7.3.	RW	0
7	IM7_EN	Enables writing ingress timestamps into ingress type 0x7 (Reserved) messages. See section 6.13.7.1.	RW	0
6	IM6_EN	Enables writing ingress timestamps into ingress type 0x6 (Reserved) messages. See section 6.13.7.1.	RW	0
5	IM5_EN	Enables writing ingress timestamps into ingress type 0x5 (Reserved) messages. See section 6.13.7.1.	RW	0
4	IM4_EN	Enables writing ingress timestamps into ingress type 0x4 (Reserved) messages. See section 6.13.7.1.	RW	0
3	IM3_EN	Enables writing ingress timestamps into ingress type 0x3 (Pdelay_Resp) messages. See section 6.13.7.1.	RW	0
2	IM2_EN	Enables writing ingress timestamps into ingress type 0x2 (Pdelay_Req) messages. See section 6.13.7.1.	RW	0
1	IM1_EN	Enables writing ingress timestamps into ingress type 0x1 (Delay_Req) messages. See section 6.13.7.1.	RW	0
0	IM0_EN	Enables writing ingress timestamps into ingress type 0x0 (Sync) messages. See section 6.13.7.1.	RW	0

Note 1: If EM0_EN=1, the state of TC_CF_EN is ignored for Sync messages.
If EM1_EN=1, the state of TC_CF_EN is ignored for Delay_Req messages.
If EM2_EN=1, the state of TC_CF_PD is ignored for Pdelay_Req messages.
If EM3_EN=1, the state of TC_CF_PD is ignored for Pdelay_Resp messages.

7.2.32 TS_INSERT

Timestamp Insertion Control Register (MDIO 2.23, SPI 23)

Bit(s)	Name	Description	R/W	Reset
15:14	SEC_ENA[1:0]	Enables writing the seconds field of the ingress timestamp into PTP ingress packets at the offset specified by the SEC_OFF field. See section 6.13.7.1. 00 = Disable writing seconds 01 = Enable, write SEC[7:0] (1 byte) 10 = Enable, write SEC[31:0] (4 byte) 11 = Enable, write SEC[47:0] (6 byte)	RW	0
13:8	SEC_OFF[5:0]	When SEC_ENA≠00, this field specifies the byte offset from the start of the PTP message header to the point at which the seconds information should be written. SEC_OFF=0 specifies the first byte of the PTP message header. See section 6.13.7.1.	RW	0
7	Reserved	Ignore on Read	RO	0
6	NSEC_ENA	PTPCR1.TOP_MODE=0: Enables writing the nanoseconds field of the ingress timestamp into PTP ingress packets at the offset specified by the NSEC_OFF field. See section 6.13.7.1. 0 = Disable writing nanoseconds 1 = Enable, write NSEC[29:0] (4 bytes, bits 31:30 set to 0). PTPCR1.TOP_MODE=1: When TS_INSERT_EN.TOP_EN=1, this bit enables writing timestamp information into ingress circuit emulation packets at the offset specified by the NSEC_OFF field. See section 6.13.8. 0 = Disable writing timestamp information 1 = Enable, write 10 bytes of timestamp information and correction information as described in the *Writing Timestamps to the Ingress Packet* paragraph in section 6.13.8.	RW	0
5:0	NSEC_OFF[5:0]	PTPCR1.TOP_MODE=0: When NSEC_ENA≠0, this field specifies the byte offset from the start of the PTP message header to the point at which the nanoseconds information should be written. See section 6.13.7.1. PTPCR1.TOP_MODE=1: When NSEC_ENA≠0, this field specifies the byte offset from the start of the sequence number (in circuit emulation control word) to the point at which the ingress packet timestamp and the PEG1 feedback timestamp should be written. See section 6.13.8. NSEC_OFF=0 specifies the first byte of the PTP message header or the first byte of the circuit emulation control word.	RW	0

7.2.33 CF_INGRESS

Each 2-bit field below selects one of 3 registers to add to the ingress event message correctionField of a specific PTP event message type.

00 = Do not add anything to correctionField
01 = Add CF_COR1 to correctionField
10 = Add CF_COR2 to correctionField
11 = Add CF_COR3 to correctionField

Ingress Correction Field Register (MDIO 2.24, SPI 24)

Bit(s)	Name	Description	R/W	Reset
15:14	IM7_CF[1:0]	Add selected value to correctionField of ingress type 0x7 (Reserved) messages. See section 6.13.7.6.	RW	0
13:12	IM6_CF[1:0]	Add selected value to correctionField of ingress type 0x6 (Reserved) messages. See section 6.13.7.6.	RW	0
11:10	IM5_CF[1:0]	Add selected value to correctionField of ingress type 0x5 (Reserved) messages. See section 6.13.7.6.	RW	0
9:8	IM4_CF[1:0]	Add selected value to correctionField of ingress type 0x4 (Reserved) messages. See section 6.13.7.6.	RW	0
7:6	IM3_CF[1:0]	Add selected value to correctionField of ingress type 0x3 (Pdelay_Resp) messages. See section 6.13.7.6.	RW	0
5:4	IM2_CF[1:0]	Add selected value to correctionField of ingress type 0x2 (Pdelay_Req) messages. See section 6.13.7.6.	RW	0
3:2	IM1_CF[1:0]	Add selected value to correctionField of ingress type 0x1 (Delay_Req) messages. See section 6.13.7.6.	RW	0
1:0	IM0_CF[1:0]	Add selected value to correctionField of ingress type 0x0 (Sync) messages. See section 6.13.7.6.	RW	0

7.2.34 CF_EGRESS

Each 2-bit field below selects one of 3 registers to subtract from the egress event message correctionField of a specific PTP event message type.

00 = Do not subtract anything from correctionField
01 = Subtract CF_COR1 from correctionField
10 = Subtract CF_COR2 from correctionField
11 = Subtract CF_COR3 from correctionField

Egress Correction Field Register (MDIO 2.25, SPI 25)

Bit(s)	Name	Description	R/W	Reset
15:14	EM7_CF[1:0]	Subtract selected value from correctionField of egress type 0x7 (Reserved) messages. See section 6.13.7.6.	RW	0
13:12	EM6_CF[1:0]	Subtract selected value from correctionField of egress type 0x6 (Reserved) messages. See section 6.13.7.6.	RW	0
11:10	EM5_CF[1:0]	Subtract selected value from correctionField of egress type 0x5 (Reserved) messages. See section 6.13.7.6.	RW	0
9:8	EM4_CF[1:0]	Subtract selected value from correctionField of egress type 0x4 (Reserved) messages. See section 6.13.7.6.	RW	0
7:6	EM3_CF[1:0]	Subtract selected value from correctionField of egress type 0x3 (Pdelay_Resp) messages. See section 6.13.7.6.	RW	0
5:4	EM2_CF[1:0]	Subtract selected value from correctionField of egress type 0x2 (Pdelay_Req) messages. See section 6.13.7.6.	RW	0
3:2	EM1_CF[1:0]	Subtract selected value from correctionField of egress type 0x1 (Delay_Req) messages. See section 6.13.7.6.	RW	0
1:0	EM0_CF[1:0]	Subtract selected value from correctionField of egress type 0x0 (Sync) messages. See section 6.13.7.6.	RW	0

7.2.35 PTPCR3

PTP Control Register 3 (MDIO P2.26, SPI 26)

Bit(s)	Name	Description	R/W	Reset
15:13	EXT_SRC[2:0]	External clock source. When EXT_CLK_EN=1, this field specifies the source of EXT_CLK signal tracked by the 1588 time engine. See section 6.13.1.4. 000 = REFCLK pin 001 = GPIO1 pin 010 = GPIO2 pin 011 = GPIO3 pin 100 = GPIO4 pin 101 = GPIO5 pin 110 = GPIO6 pin 111 = GPIO7 pin	RW	0
12	EXT_CLK_EN	External clock enable. See section 6.13.1.4. 0 = The rate at which time advances in the 1588 time engine is controlled by the PERIOD register and the REFCLK signal 1 = The rate at which time advances in the 1588 time engine is controlled by the fractional frequency offset of the EXT_CLK signal.	RW	0
11:10	EXT_DIV[1:0]	Set this bit to internally divide the EXT_CLK frequency. Must be non-zero for EXT_CLK frequencies >= 125MHz. See section 6.13.1.4. 00 = don't divide 01 = divide by 2 10 = divide by 4 11 = divide by 8	RW	0
9:8	EXT_LIM[1:0]	Specifies the maximum number of nanoseconds to adjust the time engine accumulator period from the nominal value set by the PERIOD register. See section 6.13.1.4. 00 = 0.5ns 01 = 1ns 10 = 2ns 11 = 3ns	RW	0
7:0	EXT_PER[7:0]	Specifies the nominal period, in integer nanoseconds, of the EXT_CLK signal AFTER the divider controlled by EXT_DIV. Examples 8 = 125MHz, 100 = 10MHz. See section 6.13.1.4.	RW	0

7.2.36 UID_CHK

UID Check Register (MDIO 2.28, SPI 28)

Bit(s)	Name	Description	R/W	Reset
15	CHK_EN	For each ingress PTP message a 12-bit identity code is computed from the 10-byte sourcePortIdentity field (1588v2) or the messageType, sourceCommunicationTechnology and sourceUuid fields (1588v1). When this CHK_EN bit is set, the computed identity code must match the value stored in UID_CHK.UID for the message timestamp to be stored in the TS3 FIFO (if enabled as described in section 6.13.6.3). This does not affect on-the-fly packet modification as described in section 6.13.7.	RW	0
14:12	Reserved	Ignore when Read	RO	0
11:0	UID[11:0]	The expected 12-bit identity code described in UID_CHK.CHK_EN.	RW	0

7.2.37 PKT_CLASS

Packet Classifier Configuration Register (MDIO 3.16, SPI 32)

Bit(s)	Name	Description	R/W	Reset
15	MEF_CFG	HPC should examine PTP over MEF pseudowire header packets with MEF ECID specified by MEF_ECID. See section 6.13.6.1. 0 = disable 1 = enable	RW	0
14	MPLS_UCAST	HPC should examine PTP over MPLS unicast over Ethernet packets (Ethertype=0x8847, MPLS inner label specified by MPLS_LABEL) See section 6.13.6.1. 0 = disable 1 = enable	RW	0
13	MPLS_MCAST	HPC should examine PTP over MPLS multicast over Ethernet packets (Ethertype=0x8848, MPLS inner label specified by MPLS_LABEL). See section 6.13.6.1. 0 = disable 1 = enable	RW	0
12	ENET_CFG	HPC should examine PTP over 802.3/Ethernet packets with Ethertype set by ETYPE_ALT. See section 6.13.6.1. 0 = disable 1 = enable	RW	0
11	ENET	HPC should examine PTP over 802.3/Ethernet packets (Ethertype 0x88F7). See section 6.13.6.1. 0 = disable 1 = enable	RW	0
10	UDP_IPv6	HPC should examine PTP over UDP/IPv6/Ethernet packets. (Ethertype=0x86DD, UDP destination and/or source port specified by UDP_SRC and UDP_DST). See section 6.13.6.1. 0 = disable 1 = enable	RW	0
9	UDP_IPv4	HPC should examine PTP over UDP/IPv4/Ethernet packets. (Ethertype=0x0800, UDP destination and/or source port specified by UDP_SRC and UDP_DST). See section 6.13.6.1. 0 = disable 1 = enable	RW	0
8:5	CFG_START	CFG_START[3:0] Specifies where to start the CPC comparisons. See Table 6-22 for encodings.	RW	0
4	CFG_OR	This bit specifies whether the CPC criteria are in addition to the HPC criteria ("AND") or are instead of some or all of the HPC criteria ("OR"). See section 6.13.6.2. 0 = AND (both CPC AND HPC criteria must be met) 1 = OR (either CPC OR HPC criteria must be met)	RW	0
3	VER_EXACT	Specifies how the versionPTP field of incoming packets is compared to the PTP_VERSION register field. 0 = versionPTP must be <= PTP_VERSION 1 = versionPTP must be equal to PTP_VERSION	RW	0
2:0	PTP_VERSION[2:0]	Packets must have the lower three bits of the PTP versionPTP field "match" this register field to be qualified. The match comparison can be "equal to" or "less than or equal to" as specified by the VER_EXACT field. See section 6.13.6.1.	RW	0

7.2.38 VLAN2_ID

VLAN ID Register (MDIO 3.17, SPI 33)

Bit(s)	Name	Description	R/W	Reset
15:0	VLAN2_ID[15:0]	An alternate (optional) 16-bit Ethertype code for VLAN headers. See section 6.13.6.	RW	0x8100

7.2.39 MEF_ECID_HI

MEF ECID Upper Word Register (MDIO 3.20, SPI 36)

Bit(s)	Name	Description	R/W	Reset
15:0	MEF_ECID[19:4]	The 20-bit MEF_ECID field spans this register and MEF_ECID_LO. When PKT_CLASS. MEF_CFG=1, MEF packets must have an ECID field that matches this field to be qualified. See section 6.13.6.1.	RW	0

7.2.40 MEF_ECID_LO

MEF ECID Lower Word Register (MDIO 3.21, SPI 37)

Bit(s)	Name	Description	R/W	Reset
15:12	MEF_ECID[3:0]	See the MEF_ECID_HI register description above.	RW	0
11:0	Reserved		RO	0

7.2.41 MPLS_LABEL_HI

MPLS LABEL Upper Word Register (MDIO 3.22, SPI 38)

Bit(s)	Name	Description	R/W	Reset
15:0	MPLS_LABEL[19:4]	The 20-bit MPLS_LABEL field spans this register and MPLS_LABEL_LO. When PKT_CLASS. MPLS_UCAST =1 or PKT_CLASS. MPLS_MCAST =1, MPLS packets must have an inner MPLS label that matches this field to be qualified. See section 6.13.6.1.	RW	0

7.2.42 MPLS_LABEL_LO

MPLS LABEL Lower Word Register (MDIO 3.23, SPI 39)

Bit(s)	Name	Description	R/W	Reset
15:12	MPLS_LABEL[3:0]	See the MPLS_LABEL_HI register description above.	RW	0
11:0	Reserved		RO	0

7.2.43 ETYPE_ALT

User Configurable Ethertype Register (MDIO 3.24, SPI 40)

Bit(s)	Name	Description	R/W	Reset
15:0	ETYPE_ALT[15:0]	When PKT_CLASS. ENET_CFG=1, packets with an Ethertype that matches this field can be qualified if all other configured criteria are met. See section 6.13.6.1.	RW	0

7.2.44 UDP_SRC

UDP Source Port Register (MDIO 3.25, SPI 41)

Bit(s)	Name	Description	R/W	Reset
15:0	UDP_SRC[15:0]	When UDP_SRC≠0xFFFF, if PKT_CLASS.UDP_IPv4=1, IPv4 packets must have a UDP source port number that matches this field to be qualified. When UDP_SRC≠0xFFFF, if PKT_CLASS.UDP_IPv6=1, IPv6 packets must have a UDP source port number that matches this field to be qualified. When UDP_SRC=0xFFFF, UDP source port number matching is disabled. See section 6.13.6.1.	RW	0xFFFF

7.2.45 UDP_DST

UDP Destination Port Register (MDIO 3.26, SPI 42)

Bit(s)	Name	Description	R/W	Reset
15:0	UDP_DST[15:0]	When UDP_DST≠0xFFFF, if PKT_CLASS.UDP_IPv4=1, IPv4 packets must have a UDP destination port number that matches this field to be qualified. When UDP_SRC≠0xFFFF, if PKT_CLASS.UDP_IPv6=1, IPv6 packets must have a UDP destination port number that matches this field to be qualified. When UDP_SRC=0xFFFF, UDP destination port number matching is disabled. See section 6.13.6.1.	RW	0x013F (319 decimal)

7.2.46 CFG_MASK

Configurable Packet Classifier Mask Register (MDIO 3.27, SPI 43)

Bit(s)	Name	Description	R/W	Reset
15:0	CFG_MASK[15:0]	Together, CFG_MASK, CFG_MATCH, CFG_OFFSET and CFG_WR are used to write and enable/disable any of eight configurable packet classifier criteria. See section 6.13.6.2. CFG_MASK specifies which bits are to be compared at the position in the packet specified by CFG_OFFSET. 0 = Don't compare 1 = Compare Bit 15 is the most significant bit of the 16-bit word and is the first bit transmitted or received.	RW	0

7.2.47 CFG_MATCH

Configurable Packet Classifier Match Register (MDIO 3.28, SPI 44)

Bit(s)	Name	Description	R/W	Reset
15:0	CFG_MATCH[15:0]	CFG_MATCH specifies the values of bits to be compared in a configurable packet classifier criterion. See section 6.13.6.2. When a bit of CFG_MASK=0, the corresponding bit of the packet is not checked and the corresponding bit of CFG_MATCH is ignored. When a bit of CFG_MASK=1, the corresponding bit of the packet must match the corresponding bit of CFG_MATCH to meet the configurable criterion. Bit 15 is the most significant bit of the 16-bit word and is the first bit transmitted or received.	RW	0

7.2.48 CFG_OFFSET

Configurable Packet Classifier Offset Register (MDIO 3.29, SPI 45)

Bit(s)	Name	Description	R/W	Reset
15	ENABLE	Enables the configurable packet classifier criterion. Ignored when CFG_WR.CFG_SEL=0xF. See section 6.13.6.2. 0 = Disable the packet classifier criterion 1 = Enable the packet classifier criterion	RW	0
14:8	Reserved		RO	0
7:0	OFFSET[7:0]	Specifies the offset in bytes from the CPC start position to the compare point of the configurable criterion. See section 6.13.6.2. Note: the value of this field must be less than PTP_OFFSET + 32.	RW	0

7.2.49 CFG_WR

Configurable Packet Classifier Write Register (MDIO 3.30, SPI 46)

Bit(s)	Name	Description	R/W	Reset
15:8	Reserved		RO	0
7	WR	Set this bit to write CFG_MASK, CFG_MATCH and CFG_OFFSET to the packet classifier criterion specified by the CFG_SEL field or to write CFG_OFFSET to the PTP_OFFSET indirect register. See section 6.13.6.2.	RW SC	0
6:4	Reserved		RO	0
3:0	CFG_SEL[3:0]	This field specifies the indirect register(s) that are written when the WR bit is set to 1. See section 6.13.6.2. 0000 = Write packet classifier criterion 0 0001 = Write packet classifier criterion 1 0010 = Write packet classifier criterion 2 0011 = Write packet classifier criterion 3 0100 = Write packet classifier criterion 4 0101 = Write packet classifier criterion 5 0110 = Write packet classifier criterion 6 0111 = Write packet classifier criterion 7 1000 – 1110 = unused values 1111 = Write CFG_OFFSET to the PTP_OFFSET indirect register	RW	0

7.2.50 PHY_MATCH

PHY Address Match Register (not accessible through MDIO, SPI 47)

Bit(s)	Name	Description	R/W	Reset
15	TEST	Factory Test. Always write 0.	RW	0
14:6	Reserved			
5	ENABLE	Enables SPI device address matching for common chip-select applications. When ENABLE=1, only the device whose PHY_MATCH.ADDR field matches the MDIO PHY address latched from pins at reset responds to SPI read and write commands. Devices where the addresses don't match ignore all SPI read and write commands except writes to this PHY_MATCH register. See section 6.4.	R/W	0
4:0	ADDR[4:0]	The PHY_MATCH address. See the ENABLE bit description above. See section 6.4.	R/W	0

7.3 IEEE1588 Indirect Registers

7.3.1 TIME

Format	48 bits of seconds, 30 bits of ns.
Read Access	Write TERW with RD=1 and RDSEL=000 then read time from TEIO1 –TEIO5 as shown in Table 7-3.
Write Access	Write time to TEIO1 – TEIO5 as shown in Table 7-4 and then write TERW with WR=1 and WRSEL=0000.
Description	The time in the accumulator of the time engine can be written or read directly through this register. See section 6.13.1.

7.3.2 PERIOD

Format	8 bits of ns, 32 bits of fractional ns (i.e. 40 bits with lsb of 2^{-32}ns)
Read Access	None
Write Access	Write period to TEIO3 – TEIO5 as shown in Table 7-4 and then write TERW with WR=1 and WRSEL=0001.
Description	The value in this register is added to the accumulator in the time engine every cycle of the 125MHz master clock. See Figure 6-13 and Figure 6-14. In free-run, period is set to exactly 8.0ns (0x08 0000 0000). When tracking a timing master, the period register is continually adjusted by system software as part of a hardware-software PLL. See section 6.13.1.

7.3.3 PER_ADJ

Format	8 bits of ns, 32 bits of fractional ns (i.e. 40 bits with lsb of 2^{-32}ns). 2s-complement.
Read Access	None
Write Access	Write period adjustment to TEIO3 – TEIO5 and write count to TEIO2 – TEIO3 as shown in Table 7-4 and then write TERW with WR=1 and WRSEL=0010.
Description	To perform a precise time adjustment, both the PER_ADJ value and the PERIOD value are added to the accumulator in the time engine for the number of 125MHz master clock cycles specified in the ADJ_CNT register. This results in an exact time change equal to PER_ADJ * ADJ_CNT. See Figure 6-13 and Figure 6-14 in section 6.13.1.

7.3.4 ADJ_CNT

Format	24-bit unsigned integer
Read Access	None
Write Access	Write period adjustment to TEIO3 – TEIO5 and write count to TEIO2 – TEIO3 as shown in Table 7-4 and then write TERW with WR=1 and WRSEL=0010.
Description	This register specifies the number of 125MHz master clock cycles that the PER_ADJ field should be added to the accumulator in the time engine. See the PER_ADJ register description for more details.

7.3.5 PEG1_FIFO, PEG2_FIFO

Format	22-bit FIFO command. See Table 6-18.
Read Access	None
Write Access	Write command to TEIO4 – TEIO5 as shown in Table 7-4 and then write TERW with WR=1 and WRSEL=0011 for PEG1 or WRSEL=0100 for PEG2.
Description	Commands for the programmable event generators, PEG1 and PEG2, are written to the command FIFOs through these registers. See section 6.13.3.

7.3.6 TS1_FIFO, TS2_FIFO, TS3_FIFO

Format	48 bits of seconds, 30 bits of ns. For packet timestamps, 4 bit message type, 12-bit identity code and 16-bit sequence number. For input signal timestamps, 1 bit edge polarity.
Read Access	Write TERW with RD=1 and RDSEL=001 for TS1_FIFO, RDSEL=010 for TS2_FIFO, or RDSEL=011 for TS3_FIFO then read timestamp from TEIO1 – TEIO5 as shown in Table 7-3. For packet timestamps, also read message type, identity code and sequence number from HDR_DAT1 and HDR_DAT2.
Write Access	None
Description	Timestamps, of packets or input signal edges, are read from the timestamp FIFOs through these registers. See sections 6.13.4 and 6.13.5.

7.3.7 MEAN_PATH_DELAY

Format	29 bits of ns, 2 bits of fractional ns (i.e. 31 bits with lsb of 2^{-2}=0.25 ns). Unsigned.
Read Access	None
Write Access	Write value to TEIO4 – TEIO5 as shown in Table 7-4 and then write TERW with WR=1 and WRSEL=1000.
Description	The value in this register is automatically added to the correctionField of ingress Sync packets on-the-fly. See section 6.13.7.5.

7.3.8 CF_COR1, CF_COR2, CF_COR3

Format	30 bits of ns, 2 bits of fractional ns (i.e. 32 bits with lsb of 2^{-2}=0.25 ns). 2s-complement.
Read Access	None
Write Access	Write value to TEIO4 – TEIO5 as shown in Table 7-4 and then write TERW with WR=1 and WRSEL=1001 for CF_COR1, WRSEL=1010 for CF_COR2, or WRSEL=1011 for CF_COR3.
Description	The three registers hold values that can be added to the correctionField of event messages on ingress and/or egress to correct for path asymmetries. See section 6.13.7.6.

7.3.9 Configurable Packet Classifier Criteria

Format	16-bit match register, 16-bit mask register, 8-bit offset (unsigned), 1-bit enable
Read Access	None
Write Access	Write the match value to CFG_MATCH, the mask to CFG_MASK, the offset to CFG_OFFSET (and set CFG_OFFSET.ENABLE=0 or 1) then write the CFG_WR register with the number of the CPC matching criterion in the CFG_SEL field and WR=1.
Description	This match/mask/offset set of registers allow up to eight custom criteria to be set up for the configurable packet classifier (CPC). See section 6.13.6.2.

7.3.10 PTP_OFFSET

Format	8-bit offset value (unsigned)
Read Access	None
Write Access	Write the value to CFG_OFFSET then write CFG_WR register with CFG_SEL=1111, WR=1.
Description	This register specifies the offset from the CPC start position to the start of the PTP packet header for CPC OR mode. See section 6.13.6.2.2.

8. JTAG Test Access Port and Boundary Scan

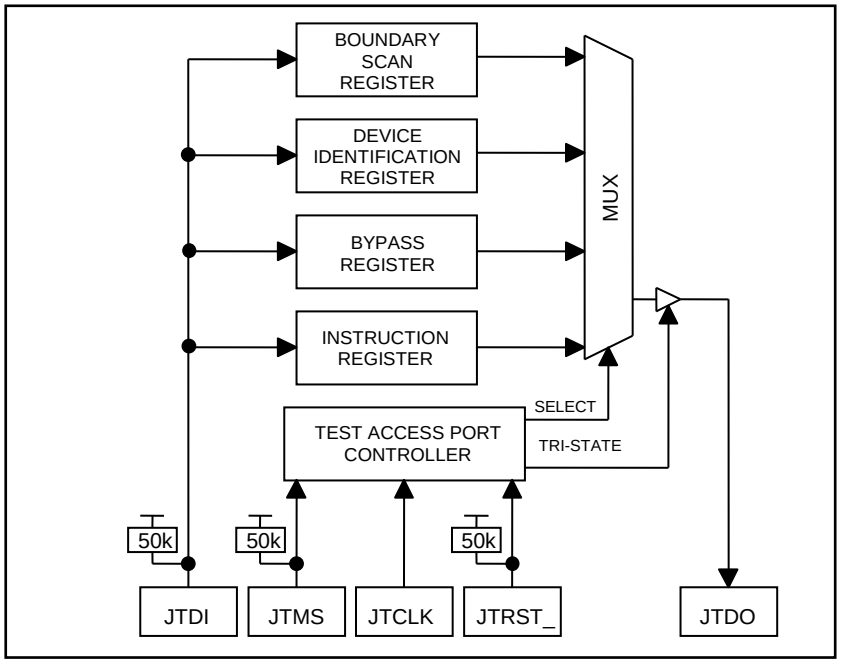
8.1 JTAG Description

The MAX24288 supports the standard instruction codes SAMPLE/PRELOAD, BYPASS, and EXTEST. Optional public instructions included are HIGHZ, CLAMP, and IDCODE. Figure 8-1 shows a block diagram. The MAX24288 contains the following items, which meet the requirements set by the IEEE 1149.1 Standard Test Access Port and Boundary Scan Architecture:

Test Access Port (TAP)	Bypass Register
TAP Controller	Boundary Scan Register
Instruction Register	Device Identification Register

The TAP has the necessary interface pins, namely JTCLK, JTRST_N, JTDI, JTDO, and JTMS. Details on these pins can be found in Table 5-5 Details about the boundary scan architecture and the TAP can be found in IEEE 1149.1-1990, IEEE 1149.1a-1993, and IEEE 1149.1b-1994.

Figure 8-1. JTAG Block Diagram



8.2 JTAG TAP Controller State Machine Description

This section discusses the operation of the TAP controller state machine. The TAP controller is a finite state machine that responds to the logic level at JTMS on the rising edge of JTCLK. Each of the states denoted in Figure 8-2 are described in the following paragraphs.

Test-Logic-Reset. Upon device power-up, the TAP controller starts in the Test-Logic-Reset state. The instruction register contains the IDCODE instruction. All system logic on the device operates normally.

Run-Test-Idle. Run-Test-Idle is used between scan operations or during specific tests. The instruction register and all test registers remain idle.

Select-DR-Scan. All test registers retain their previous state. With JTMS low, a rising edge of JTCLK moves the controller into the Capture-DR state and initiates a scan sequence. JTMS high moves the controller to the Select-IR-SCAN state.

Capture-DR. Data can be parallel-loaded into the test register selected by the current instruction. If the instruction does not call for a parallel load or the selected test register does not allow parallel loads, the register remains at its current value. On the rising edge of JTCLK, the controller goes to the Shift-DR state if JTMS is low or to the Exit1-DR state if JTMS is high.

Shift-DR. The test register selected by the current instruction is connected between JTDI and JTDO and data is shifted one stage toward the serial output on each rising edge of JTCLK. If a test register selected by the current instruction is not placed in the serial path, it maintains its previous state.

Exit1-DR. While in this state, a rising edge on JTCLK with JTMS high puts the controller in the Update-DR state, which terminates the scanning process. A rising edge on JTCLK with JTMS low puts the controller in the Pause-DR state.

Pause-DR. Shifting of the test registers is halted while in this state. All test registers selected by the current instruction retain their previous state. The controller remains in this state while JTMS is low. A rising edge on JTCLK with JTMS high puts the controller in the Exit2-DR state.

Exit2-DR. While in this state, a rising edge on JTCLK with JTMS high puts the controller in the Update-DR state and terminates the scanning process. A rising edge on JTCLK with JTMS low puts the controller in the Shift-DR state.

Update-DR. A falling edge on JTCLK while in the Update-DR state latches the data from the shift register path of the test registers into the data output latches. This prevents changes at the parallel output because of changes in the shift register. A rising edge on JTCLK with JTMS low puts the controller in the Run-Test-Idle state. With JTMS high, the controller enters the Select-DR-Scan state.

Select-IR-Scan. All test registers retain their previous state. The instruction register remains unchanged during this state. With JTMS low, a rising edge on JTCLK moves the controller into the Capture-IR state and initiates a scan sequence for the instruction register. JTMS high during a rising edge on JTCLK puts the controller back into the Test-Logic-Reset state.

Capture-IR. The Capture-IR state is used to load the shift register in the instruction register with a fixed value. This value is loaded on the rising edge of JTCLK. If JTMS is high on the rising edge of JTCLK, the controller enters the Exit1-IR state. If JTMS is low on the rising edge of JTCLK, the controller enters the Shift-IR state.

Shift-IR. In this state, the instruction register's shift register is connected between JTDI and JTDO and shifts data one stage for every rising edge of JTCLK toward the serial output. The parallel register and the test registers remain at their previous states. A rising edge on JTCLK with JTMS high moves the controller to the Exit1-IR state. A rising edge on JTCLK with JTMS low keeps the controller in the Shift-IR state, while moving data one stage through the instruction shift register.

Exit1-IR. A rising edge on JTCLK with JTMS low puts the controller in the Pause-IR state. If JTMS is high on the rising edge of JTCLK, the controller enters the Update-IR state and terminates the scanning process.

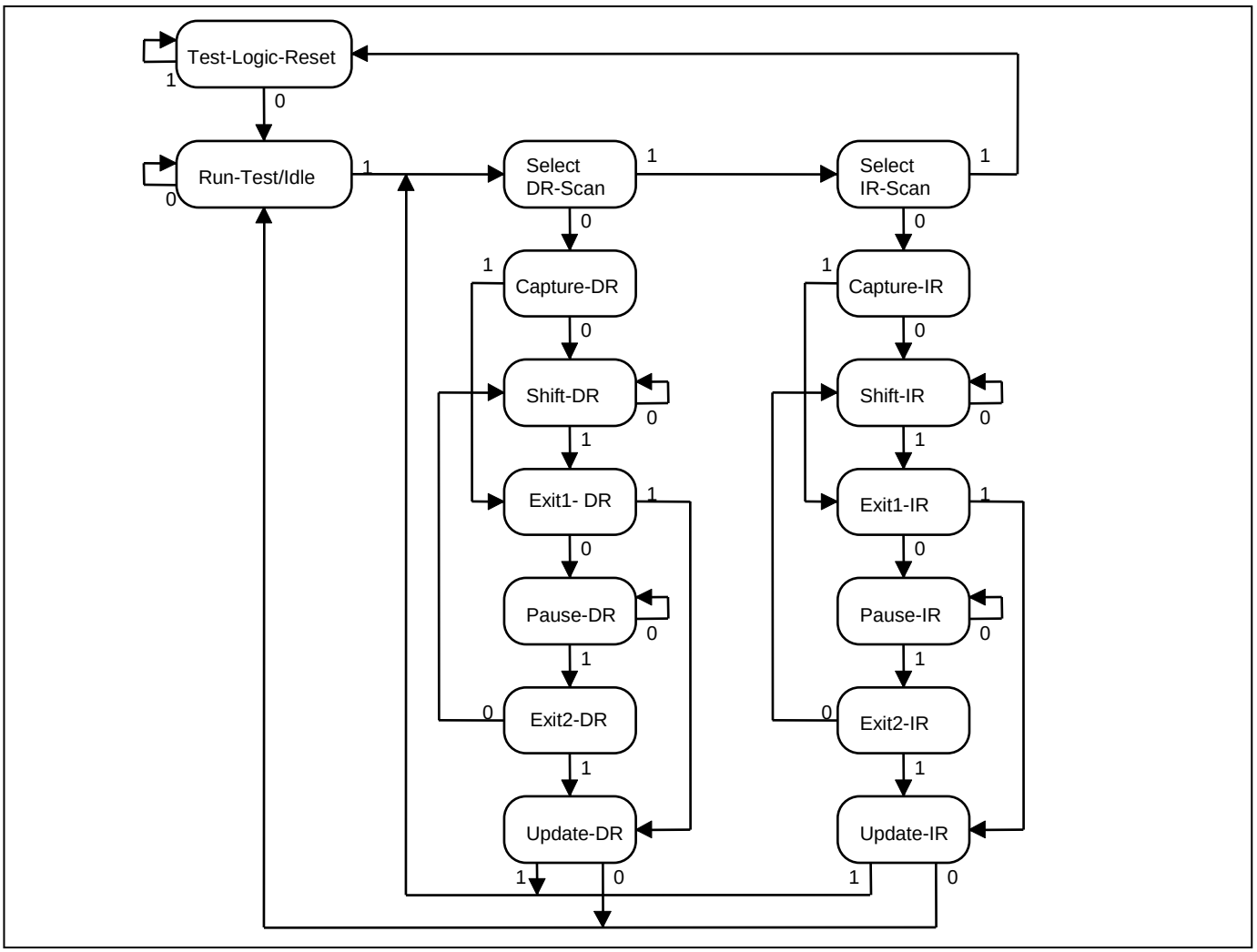
Pause-IR. Shifting of the instruction register is halted temporarily. With JTMS high, a rising edge on JTCLK puts the controller in the Exit2-IR state. The controller remains in the Pause-IR state if JTMS is low during a rising edge on JTCLK.

Exit2-IR. A rising edge on JTCLK with JTMS high puts the controller in the Update-IR state. The controller loops back to the Shift-IR state if JTMS is low during a rising edge of JTCLK in this state.

Update-IR. The instruction shifted into the instruction shift register is latched into the parallel output on the falling edge of JTCLK as the controller enters this state. Once latched, this instruction becomes the current instruction. A

rising edge on JTCLK with JTMS low puts the controller in the Run-Test-Idle state. With JTMS high, the controller enters the Select-DR-Scan state.

Figure 8-2. JTAG TAP Controller State Machine



8.3 JTAG Instruction Register and Instructions

The instruction register contains a shift register as well as a latched parallel output and is 3 bits in length. When the TAP controller enters the Shift-IR state, the instruction shift register is connected between JTDI and JTDO. While in the Shift-IR state, a rising edge on JTCLK with JTMS low shifts data one stage toward the serial output at JTDO. A rising edge on JTCLK in the Exit1-IR state or the Exit2-IR state with JTMS high moves the controller to the Update-IR state. The falling edge of that same JTCLK latches the data in the instruction shift register to the instruction parallel output. Table 8-1 shows the instructions supported by the MAX24288 and their respective operational binary codes.

Table 8-1. JTAG Instruction Codes

INSTRUCTIONS	SELECTED REGISTER	INSTRUCTION CODES
SAMPLE/PRELOAD	Boundary Scan	010
BYPASS	Bypass	111
EXTEST	Boundary Scan	000
CLAMP	Bypass	011
HIGHZ	Bypass	100
IDCODE	Device Identification	001

SAMPLE/PRELOAD. SAMPLE/RELOAD is a mandatory instruction for the IEEE 1149.1 specification. This instruction supports two functions. First, the digital I/Os of the device can be sampled at the boundary scan register, using the Capture-DR state, without interfering with the device's normal operation. Second, data can be shifted into the boundary scan register through JTDI using the Shift-DR state.

EXTEST. EXTEST allows testing of the interconnections to the device. When the EXTEST instruction is latched in the instruction register, the following actions occur: (1) Once the EXTEST instruction is enabled through the Update-IR state, the parallel outputs of the digital output pins are driven. (2) The boundary scan register is connected between JTDI and JTDO. (3) The Capture-DR state samples all digital inputs into the boundary scan register.

BYPASS. When the BYPASS instruction is latched into the parallel instruction register, JTDI is connected to JTDO through the 1-bit bypass register. This allows data to pass from JTDI to JTDO without affecting the device's normal operation.

IDCODE. When the IDCODE instruction is latched into the parallel instruction register, the device identification register is selected. The device ID code is loaded into the device identification register on the rising edge of JTCLK, following entry into the Capture-DR state. Shift-DR can be used to shift the ID code out serially through JTDO. During Test-Logic-Reset, the ID code is forced into the instruction register's parallel output.

HIGHZ. All digital outputs are placed into a high-impedance state. The bypass register is connected between JTDI and JTDO.

CLAMP. All digital output pins output data from the boundary scan parallel output while connecting the bypass register between JTDI and JTDO. The outputs do not change during the CLAMP instruction.

8.4 JTAG Test Registers

IEEE 1149.1 requires a minimum of two test registers—the bypass register and the boundary scan register. An optional test register, the identification register, has been included in the device design. It is used with the IDCODE instruction and the Test-Logic-Reset state of the TAP controller.

Bypass Register. This is a single 1-bit shift register used with the BYPASS, CLAMP, and HIGHZ instructions to provide a short path between JTDI and JTDO.

Boundary Scan Register. This register contains a shift register path and a latched parallel output for control cells and digital I/O cells. The BSDL file is available on the MAX24288 page of Microsemi's website.

Identification Register. This register contains a 32-bit shift register and a 32-bit latched parallel output. It is selected during the IDCODE instruction and when the TAP controller is in the Test-Logic-Reset state. The device identification code for the MAX24288 is shown in Table 8-2.

Table 8-2. JTAG ID Code

DEVICE	REVISION	DEVICE CODE	MANUFACTURER CODE	REQUIRED
MAX24288	Note 1	0101 1110 1110 0000	00010100001	1

Note 1: 0000 = rev A1. 0001 = rev B1. Other values: contact factory.

9. Electrical Characteristics

ABSOLUTE MAXIMUM RATINGS

Voltage Range on Any Signal IO Lead with Respect to V_{SS} ..... -0.3V to +3.63V
Supply Voltage (VDD12) Range with Respect to VSS ..... -0.3V to +1.32V
Supply Voltage (VDD33) Range with Respect to VSS ..... -0.3V to +3.63V
Operating Temperature Range: Industrial ..... -40°C to +85°C
Storage Temperature Range ..... -55°C to +125°C
Lead Temperature (soldering, 10s) ..... +300°C
Soldering Temperature (reflow) ..... +260°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device. Ambient operating temperature range when device is mounted on a four-layer JEDEC test board with no airflow.

Note 1: The typical values listed in the tables of section 9 are not production tested.
Note 2: Specifications to T_A = -40°C are guaranteed by design and not production tested.

9.1 Recommended Operating Conditions

Table 9-1. Recommended DC Operating Conditions

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage, Nominal 1.2V	VDD12		1.14	1.2	1.26	V
Supply Voltage, Nominal 3.3V	VDD33		3.135	3.3	3.465	V
Ambient Temperature Range	T_A		-40		+85	°C
Junction Temperature Range	T_J		-40		+125	°C

9.2 DC Electrical Characteristics

Unless otherwise stated, all specifications in this section are valid for VDD12 = 1.2V ±5%, VDD33 = 3.3V ±5% and T_A = -40°C to +85°C.

Table 9-2. DC Characteristics

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Current, VDD12 Pins	I_{DD12}	10, 25 or 125MHz REFCLK		180		mA
Supply Current, VDD33 Pin	I_{DD33}	10, 25 or 125MHz REFCLK		100		mA
Supply Current, VDD12 Pins	I_{DD12}	12.8MHz REFCLK (Note 1)		205	260	mA
Supply Current, VDD33 Pin	I_{DD33}	12.8MHz REFCLK		140	175	mA
Supply Current Reduction, Rx Power-down, VDD12	ΔI_{DDRX12}	PTPCR1.RX_PWDN=1		35		mA
Supply Current Reduction, Rx Power-down, VDD33	ΔI_{DDRX33}	PTPCR1.RX_PWDN=1		20		mA
Supply Current Reduction, Tx Power-down, VDD12	ΔI_{DDTX12}	PTPCR1.TX_PWDN=1		55		mA
Supply Current Reduction, Tx Power-down, VDD33	ΔI_{DDTX33}	PTPCR1.TX_PWDN=1		10		mA
Supply Current Reduction, Time Engine Power-down, VDD12	ΔI_{DDTE12}	PTPCR1.TE_PWDN=1		20		mA
Supply Current, Full Power-down, VDD12	I_{DDPD12}	All PWDN=1 in PTPCR1		2		mA
Supply Current, Full Power-down, VDD33	I_{DDPD33}	All PWDN=1 in PTPCR1		20		mA
Input Capacitance	C_{IN}			4		pF
Output Capacitance	C_{OUT}			7		pF

Note 1: When a 12.8MHz oscillator is used the TX PLL uses a two-stage process to perform the frequency conversion and therefore consumes additional power.

9.2.1 CMOS/TTL DC Characteristics

Table 9-3. DC Characteristics for Parallel, MDIO and SPI Interfaces

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output High Voltage	V_{OH}	I_{OH} = -1mA, V_{DD33}=3.135V	2.4		V_{DD33}	V
Output Low Voltage	V_{OL}	I_{OL} = 1mA, V_{DD33}=3.135V	0		0.4	V
Output High Voltage	V_{OH}	MDC, MDIO. Notes 1, 3, 4	2.4		V_{DD33}	V
Output Low Voltage	V_{OL}	MDC, MDIO. Notes 2, 3, 4	0		0.4	V
Input High Voltage	V_{IH}		2.0		V_{DD33}+0.2	V
Input Low Voltage	V_{IL}		-0.2		0.8	V
Input High Current	I_{IH}	VIN=3.3V			10	µA
Input Low Current, all other Input Pins	I_{IL}	VIN=0V	-10			µA
Output and I/O Leakage (when High Impedance)	I_{LO}		-10		+10	µA

Note 1: I_{OH}=-4mA
Note 2: I_{OL}=4mA
Note 3: MDC load: 340pF max.
Note 4: MDIO load: 340pF max plus 2kΩ±5% pulldown resistor.

9.2.2 SGMII/1000BASE-X DC Characteristics

Table 9-4. SGMII/1000BASE-X Transmit DC Characteristics

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Voltage High, TDP or TDN (Single-Ended)	$V_{OH,DC}$	DC-coupled. Load: 50Ω pullup resistors to TVDD33 on TDP pin and on TDN pin. (Note 1)		TVDD33		V
Output Voltage Low, TDP or TDN (Single-Ended)	$V_{OL,DC}$			TVDD33 – 0.4		V
Output Common Mode Voltage	$V_{OCM,DC}$			TVDD33 –0.2V		V
Differential Output Voltage $\|V_{TDP} – V_{TDN}\|$	$V_{OD,DC}$		320	400	500	mV
Differential Output Voltage $\|V_{TDP} – V_{TDN}\|$ Peak-to-Peak	$V_{OD,DC,PP}$		640	800	1000	mV_{P-P}
Output Common Mode Voltage	$V_{OCM,AC}$	AC-coupled to100Ω load. See Figure 6-7.		TVDD33 –0.4V		V
Differential Output Voltage, $\|V_{TDP} – V_{TDN}\|$	$V_{OD,AC}$			400		mV
Differential Output Voltage, $\|V_{TDP} – V_{TDN}\|$ Peak-to-Peak	$V_{OD,AC,PP}$			800		mV_{P-P}
Output Impedance	R_{OUT}	Single Ended, to TVDD33	35	50	65	Ω
Mismatch in a pair	ΔR_{OUT}				10	%

Table 9-5. SGMII/1000BASE-X Receive DC Characteristics

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Voltage, RDP pin	V_{RDP}		1.4		RVDD33	V
Input Voltage, RDN pin	V_{RDN}		1.4		RVDD33	V
Input Common Mode Voltage	V_{ICM}	External components as shown in Figure 6-7.		2.64		V
Input Differential Voltage	V_{ID}	$\|V_{RDP} – V_{RDN}\|$	200		1600	mV

9.3 AC Electrical Characteristics

Unless otherwise stated, all specifications in this section are valid for VDD12 = 1.2V ±5%, VDD33 = 3.3V ±5% and T_A = -40°C to +85°C.

9.3.1 REFCLK AC Characteristics

Table 9-6. REFCLK AC Characteristics

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Frequency Accuracy	Δf/f		-100		+100	ppm
Duty Cycle		Note 1	40		60	%
Rise Time (20–80%)	t_R				1	ns
Fall Time (20–80%)	t_F				1	ns

REFCLK Jitter: See Table 9-10 below.

Note 1: If the TX PLL is bypassed by setting PTPCR1.PLL_PWDN=1 then REFCLK duty cycle should be as close to 50% as possible because both edges of REFCLK are used by the 1588 circuitry.

9.3.2 SGMII/1000BASE-X Interface Receive AC Characteristics

Table 9-7. 1000BASE-X and SGMII Receive AC Characteristics

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Data Rate, Nominal	f_{IN}			1250		Mbps
Input Frequency Accuracy	Δf/f		-100		+100	ppm
Skew, RDP vs. RDN	|tskew|	Note 1			20	ps

Note 1: Measured at 50% of the transition

Table 9-8. 1000BASE-X and SGMII Receive Jitter Tolerance

PARAMETER	SYMBOL	CONDITIONS	UI p-p	ps p-p
Rx Jitter Tolerance, Deterministic Jitter, max	DJ_{RD}	Note 1, 2	0.46	370
Rx Jitter Tolerance, Total Jitter, max	TJ_{RD}	Note 1, 2	0.75	600

Note 1: Jitter requirements represent high-frequency jitter (above 637kHz) and do not represent low-frequency jitter or wander. Random jitter = Total Jitter minus Deterministic Jitter.

Note 2: The bandwidth of the CDR PLL is approximately 4MHz.

9.3.3 SGMII/1000BASE-X Interface Transmit AC Characteristics

Table 9-9. SGMII and 1000BASE-X Transmit AC Characteristics

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
TCLKP/TCLKN Duty Cycle		at 625MHz	48		52	%
Rise Time (20–80%)	t_R		100		200	ps
Fall Time (20–80%)	t_F		100		200	ps
TCLK edge to TD valid data	$t_{clock2q}$	Note 1	250		550	ps

Note 1: Measured at 0V differential. Does not include effects of jitter. Guaranteed by design and not production tested.

Table 9-10. 1000BASE-X Transmit Jitter Characteristics

PARAMETER	SYMBOL	CONDITIONS	Typical		Max	
			UI p-p	ps p-p	UI p-p	ps p-p
Tx Output Jitter, Deterministic	DJ_{TD}	Note 1, 2	0.025	20	0.10	80
Tx Output Jitter, Total	TJ_{TD}	Note 1, 2	0.0875	70	0.24	192

Note 1: Jitter requirements represent high-frequency jitter (above 637kHz) and do not represent low-frequency jitter or wander. Random jitter = Total Jitter minus Deterministic Jitter.

Note 2: Typical values are room-temperature measurements with a Connor-Winfield MX010 crystal oscillator connected to the REFCLK pin. Note that the bandwidth of the TX PLL is in the 300-400kHz range.

9.3.4 Parallel Interface Receive AC Characteristics

Figure 9-1. MII/GMII/RGMII Receive Timing Waveforms

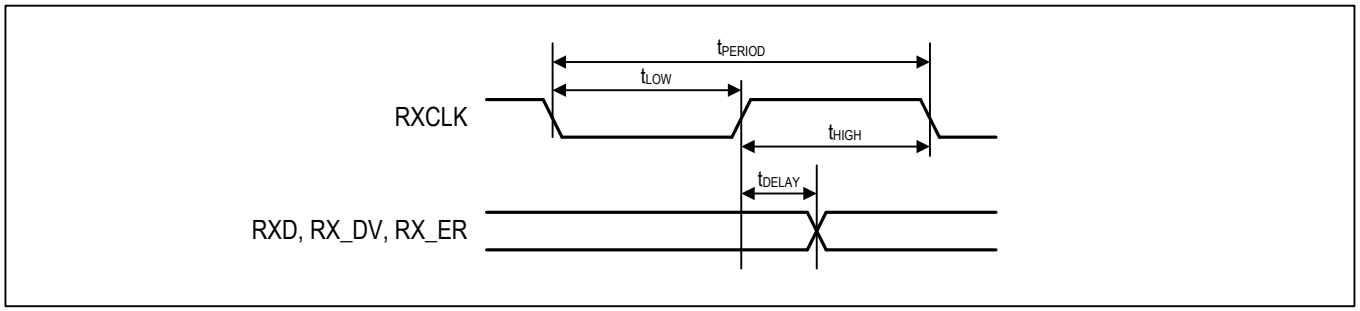


Table 9-11. GMII Receive AC Characteristics

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
RXCLK Period	t_{PERIOD}		7.5	8	8.5	ns
Rx Clock Duty Cycle			40		60	%
RXCLK Rising Edge to RXD, RX_DV, RX_ER Valid	t_{DELAY}	Note 1	1		5.5	ns
RXCLK Rise Time	t_R	GMII mode, 0.7V to 1.9V			1	ns
RXCLK Fall Time	t_F	GMII mode, 1.9V to 0.7V			1	ns
RXCLK Slew Rate Rising		0.7V to 1.9V, Note 2	0.6			V/ns
RXCLK Slew Rate Falling		1.9V to 0.7V, Note 2	0.6			V/ns

Note 1: 802.3 specifies setup and hold times for the receiver of the signals. This output delay specification has values that ensure 802.3 setup and hold specifications are met.
Note 2: Clock Slew rate is the instantaneous rate of change of the clock potential with respect to time (dV/dt), not an average value over the entire rise or fall time interval. Conformance with this specification guarantees that the clock signals will rise and fall monotonically through the switching region.
Note 3: All specifications in this table are guaranteed by design with output load of 5pF.

Table 9-12. RGMII-1000 Receive AC Characteristics

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
RXCLK Period	t_{PERIOD}		7.5	8	8.5	ns
RXCLK Duty Cycle		t_{LOW} % of t_{PERIOD} , Note 1	45		55	%
RXCLK to RXD, RX_CTL Delay	t_{DELAY}	Notes 2, 3	-0.2		0.8	ns
Rise Time, All RX Signals	t_R	20% to 80%			0.75	ns
Fall Time, All RX Signals	t_F	20% to 80%			0.75	ns

Note 1: Per the RGMII spec, duty cycle may be stretched/shrunk during speed changes or while transitioning to a received packet's clock domain as long as minimum duty cycle is not violated and stretching occurs for no more than three clock cycles of the lowest speed transitioned between.
Note 2: RXCLK timing is from both edges in RGMII 1000 Mbps mode.
Note 3: The RGMII specification requires clocks to be routed such that a trace delay is added to the RXCLK signal to provide sufficient setup time for RXD and RX_CTL vs. RXCLK at the receiving component.
Note 4: All specifications in this table are guaranteed by design with output load of 5pF.

Table 9-13. RGMII-10/100 Receive AC Characteristics

PARAMETER	SYMBOL	10 Mbps			100 Mbps			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
RXCLK Period	t_{PERIOD}		400			40		ns
RXCLK Duty Cycle (Note 4)		45		55	45		55	%
RXCLK to RXD, RX_CTL Delay (Notes 1, 2)	t_{DELAY}	-0.2		0.8	-0.2		0.8	ns
Rise Time, All RX Signals, 20% to 80%	t_R			0.75			0.75	ns
Fall Time, All RX Signals, 20% to 80%	t_F			0.75			0.75	ns

Note 1: RXCLK to RXD is timed from rising edge.
Note 2: RXCLK to RX_CTL is timed from both RXCLK edges.
Note 3: Per the RGMII spec, duty cycle may be stretched/shrunk during speed changes or while transitioning to a received packet's clock domain as long as minimum duty cycle is not violated and stretching occurs for no more than three clock cycles of the lowest speed transitioned between.
Note 4: All specifications in this table are guaranteed by design with output load of 5pF.

Table 9-14. MII–DCE Receive AC Characteristics

PARAMETER	SYMBOL	10 Mbps			100 Mbps			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
RXCLK Period	t_{PERIOD}		400			40		ns
RXCLK Duty Cycle		45		55	45		55	%
RXCLK to RXD, RX_DV, RX_ER Delay	t_{DELAY}	180		230	18		30	ns

Note 1: RXCLK is an output in this mode.
Note 2: 802.3 specifies setup and hold times, but setup and hold specifications are typically for inputs to an IC rather than outputs. This output delay specification has values that ensure 802.3 setup and hold specifications are met.
Note 3: All specifications in this table are guaranteed by design with output load of 5pF.

Table 9-15. MII–DTE Receive AC Characteristics

PARAMETER	SYMBOL	10 Mbps			100 Mbps			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
RXCLK Period	t_{PERIOD}		400			40		ns
RXCLK Duty Cycle		45		55	45		55	%
RXCLK to RXD, RX_DV, RX_ER Delay	t_{DELAY}	0		10	0		10	ns

Note 1: RXCLK is an input in this mode.
Note 2: 802.3 specifies setup and hold times, but setup and hold specifications are typically for inputs to an IC rather than outputs. This output delay specification has values that ensure 802.3 setup and hold specifications are met.
Note 3: All specifications in this table are guaranteed by design with output load of 5pF.

9.3.5 Parallel Interface Transmit AC Characteristics

Figure 9-2. MII/GMII/RGMII Transmit Timing Waveforms

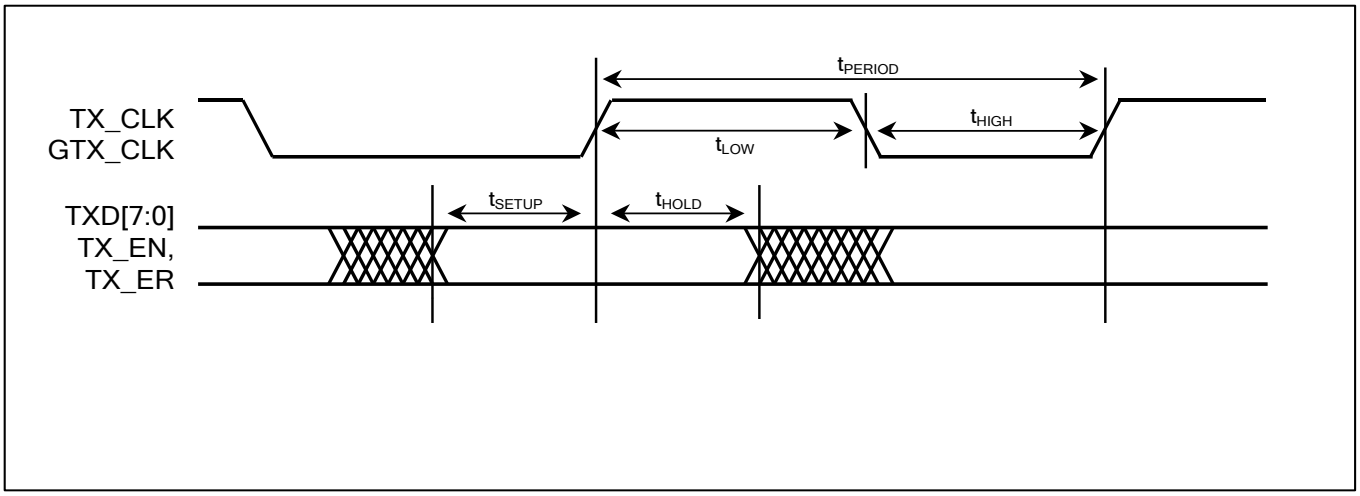


Table 9-16. GMII and RGMII-1000 Transmit AC Characteristics

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Input Voltage Low, AC	V_{IL_AC}				0.9	V
Input Voltage High, AC	V_{IH_AC}		1.7			V
GTXCLK Period	t_{PERIOD}		7.2	8	8.8	ns
GTXCLK Low Time	t_{LOW}		2.5			ns
GTXCLK High Time	t_{HIGH}		2.5			ns
GTXCLK Duty Cycle		GMII mode, t_{LOW} % of t_{PERIOD}	40		60	%
GTXCLK Duty Cycle		RGMII-1000 mode, t_{LOW} % of t_{PERIOD}	45		55	%
GTXCLK, TXD, TX_DV, TX_ER Rise Time	t_R	30% to 70% of VDD33, Note 1	0.5		2	ns
GTXCLK, TXD, TX_DV, TX_ER Fall Time	t_F	70% to 30% of VDD33, Note 1	0.5		2	ns
TXD, TX_DV, TX_ER to GTXCLK Setup Time	t_{SETUP}	Note 1	1			ns
GTXCLK to TXD, TX_DV, TX_ER Hold Time	t_{HOLD}	Note 1	0			ns

Note 1: GTXCLK timing is from both edges in RGMII 1000 Mbps mode.
Note 2: All specifications in this table are guaranteed by design.

Table 9-17. RGMII-10/100 Transmit AC Characteristics

PARAMETER	SYMBOL	10 Mbps			100 Mbps			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
GTXCLK Period	t_{PERIOD}		400			40		ns
GTXCLK Duty Cycle (Note 3)		40		60	40		60	%
TXD, TX_CTL to GTXCLK Setup Time	t_{SETUP}	1			1			ns
GTXCLK to TXD, TX_CTL Hold Time	t_{HOLD}	0			0			ns
Rise Time, All TX Signals, 0.5V to 2.0V	t_R			0.75			0.75	ns
Fall Time, All TX Signals, 0.5V to 2.0V	t_F			0.75			0.75	ns

Note 1: TXCLK to TXD is timed from rising edge.
Note 2: TXCLK to TX_CTL is timed from both TXCLK edges.
Note 3: Per the RGMII spec, duty cycle may be stretched/shrunk during speed changes or while transitioning to a received packet's clock domain as long as minimum duty cycle is not violated and stretching occurs for no more than three clock cycles of the lowest speed transitioned between.
Note 4: All specifications in this table are guaranteed by design.

Table 9-18. MII–DCE Transmit AC Characteristics

PARAMETER	SYMBOL	10 Mbps			100 Mbps			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
TXCLK Period	t_{PERIOD}		400			40		ns
TXCLK Duty Cycle		45		55	45		55	%
TXD, TX_DV, TX_ER to TXCLK Setup Time	t_{SETUP}	6.5			6.5			ns
TXCLK to TXD, TX_DV, TX_ER Hold Time	t_{HOLD}	0			0			ns

Note 1: TXCLK is an output in this mode.
Note 2: All specifications in this table are guaranteed by design.

Table 9-19. MII–DTE Transmit AC Characteristics

PARAMETER	SYMBOL	10 Mbps			100 Mbps			UNITS
		MIN	TYP	MAX	MIN	TYP	MAX	
TXCLK Period	t_{PERIOD}		400			40		ns
TXCLK Duty Cycle		40		60	40		60	%
TXD, TX_DV, TX_ER to TXCLK Setup Time	t_{SETUP}	6.5			6.5			ns
TXCLK to TXD, TX_DV, TX_ER Hold Time	t_{HOLD}	0			0			ns

Note 1: TXCLK is an input in this mode.
Note 2: All specifications in this table are guaranteed by design.

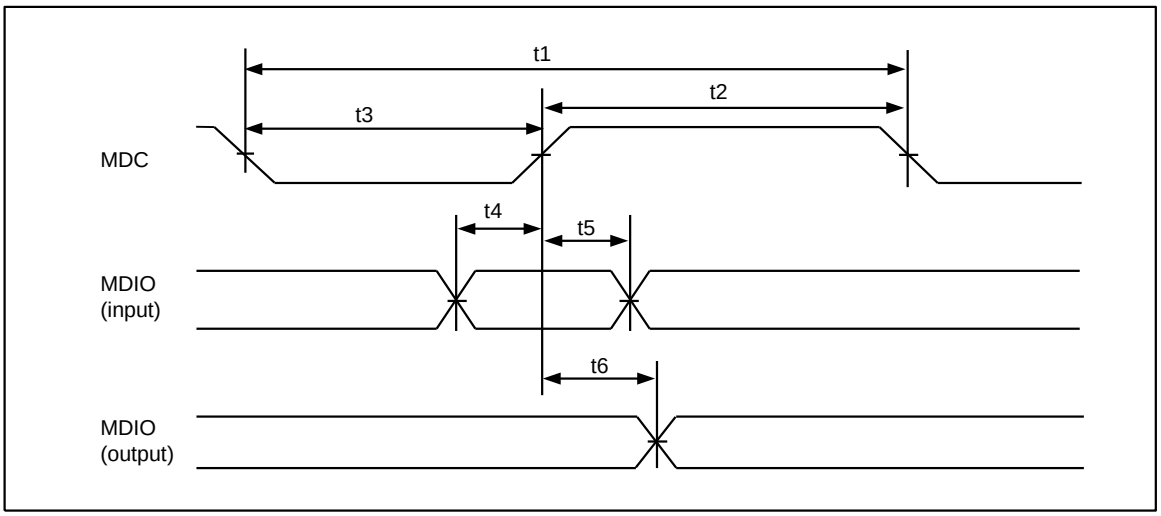
9.3.6 MDIO Interface AC Characteristics

Table 9-20. MDIO Interface AC Characteristics

Parameter	Symbol	Conditions	Min	Typ	Max	Units
MDC Input Period (12.5MHz)	t1	Note 1	80			ns
MDC Input High	t2	Notes 1, 2	30			ns
MDC Input Low	t3	Notes 1, 2	30			ns
MDIO Input Setup Time to MDC	t4	Note 1	10			ns
MDIO Input Hold Time from MDC	t5	Note 1	10			ns
MDC to MDIO Output Delay	t6	Note 1,3	0		40	ns
MDC to MDIO High Impedance	t6	Note 1,4	0		40	ns

Note 1: The input/output timing reference level for all signals is VDD33 / 2. All parameters are with 340pF load on MDC and 340pF load and 2kΩ pulldown on MDIO.
Note 2: All specifications in this table are guaranteed by design.
Note 3: Data is valid on MDIO until min delay time.
Note 4: When going to high impedance, data is valid until min and signal is high impedance after max.

Figure 9-3. MDIO Interface Timing



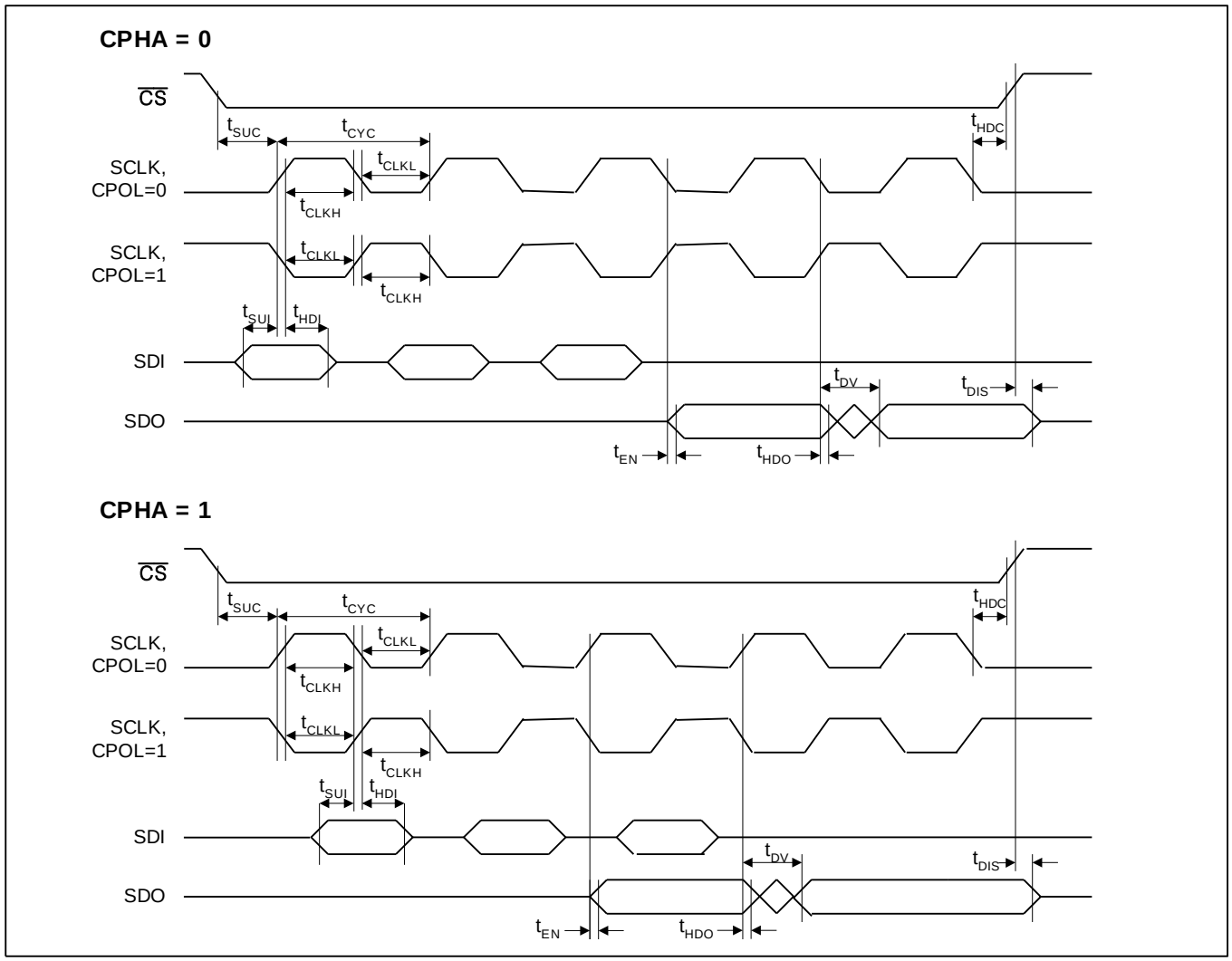
9.3.7 SPI Interface AC Characteristics

Table 9-21. SPI Interface Timing

PARAMETER (Note 1)	SYMBOL	MIN	TYP	MAX	UNITS
SCLK Frequency	f_{BUS}			25	MHz
SCLK Cycle Time	t_{CYC}	40			ns
CS_N Setup to First SCLK Edge	t_{SUC}	5			ns
CS_N Hold Time After Last SCLK Edge	t_{HDC}	5			ns
SCLK High Time	t_{CLKH}	15			ns
SCLK Low Time	t_{CLKL}	15			ns
SDI Data Setup Time	t_{SUI}	5			ns
SDI Data Hold Time	t_{HDI}	5			ns
SDO Enable Time (High-Impedance to Output Active) (Note 4)	t_{EN}	5		15	ns
SDO Disable Time (Output Active to High-Impedance)	t_{DIS}			15	ns
SDO Data Valid Time (Note 3)	t_{DV}	5		16	ns

Note 1: All timing is specified with 100pF load on all SPI pins.
Note 2: All specifications in this table are guaranteed by design.
Note 3: Data is valid on SDO until min delay time.
Note 4: SDO is high impedance for at least min enable time.

Figure 9-4. SPI Interface Timing Diagram



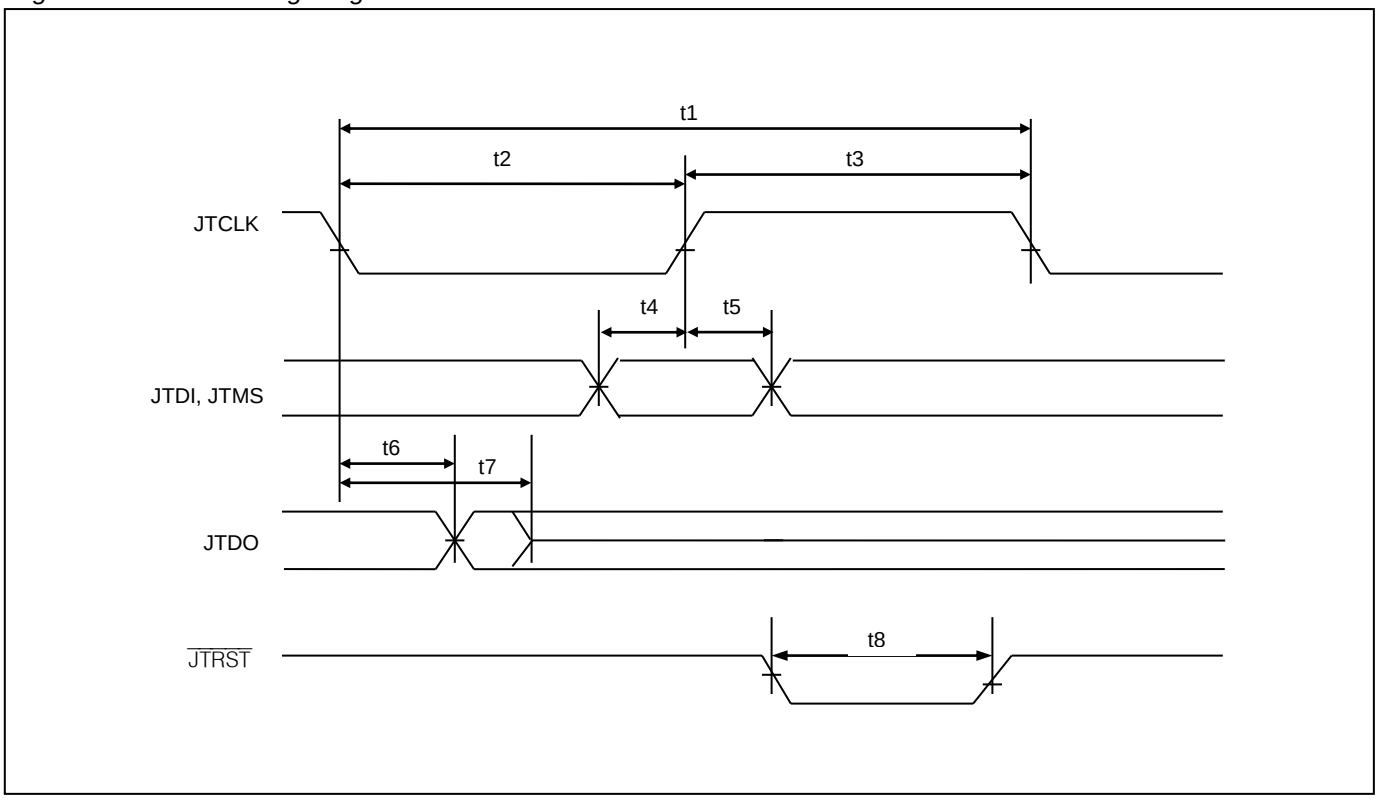
9.3.8 JTAG Interface AC Characteristics

Table 9-22. JTAG Interface Timing

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
JTCLK Clock Period	t1		1000		ns
JTCLK Clock High/Low Time (Note 1)	t2/t3	50	500		ns
JTCLK to JTDI, JTMS Setup Time	t4	50			ns
JTCLK to JTDI, JTMS Hold Time	t5	50			ns
JTCLK to JTDO Delay	t6	2		50	ns
JTCLK to JTDO High-Impedance Delay (Note 2)	t7			50	ns
$\overline{\text{JTRST}}$ Width Low Time	t8	100			ns

Note 1: Clock can be stopped high or low.
Note 2: All specifications in this table are guaranteed by design.

Figure 9-5. JTAG Timing Diagram



9.3.9 1588 GPIO Propagation Delays

Table 9-23. 1588 GPIO Propagation Delays

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS
GPIO1 Input Edge to Input Timestamper 1	$t_{IO1-TS1}$		2.46		ns
GPIO2 Input Edge to Input Timestamper 1	$t_{IO2-TS1}$		2.46		ns
GPIO3 Input Edge to Input Timestamper 1	$t_{IO3-TS1}$		2.49		ns
GPIO1 Input Edge to Input Timestamper 2	$t_{IO1-TS2}$		1.82		ns
GPIO2 Input Edge to Input Timestamper 2	$t_{IO2-TS2}$		1.85		ns
GPIO3 Input Edge to Input Timestamper 2	$t_{IO3-TS2}$		1.84		ns
GPIO1 Input Edge to Input Timestamper 3	$t_{IO1-TS3}$		1.83		ns
GPIO2 Input Edge to Input Timestamper 3	$t_{IO2-TS3}$		1.86		ns
GPIO3 Input Edge to Input Timestamper 3	$t_{IO3-TS3}$		1.84		ns
PEG1 to GPO1 Output Edge	t_{P1-O1}		2.65		ns
PEG1 to GPIO1 Output Edge	t_{P1-IO1}		2.53		ns
PEG1 to GPIO3 Output Edge	t_{P1-IO3}		2.44		ns
PEG1 to GPIO4 Output Edge	t_{P1-IO4}		2.42		ns
PEG1 to GPIO5 Output Edge	t_{P1-IO5}		2.48		ns
PEG1 to GPIO6 Output Edge	t_{P1-IO6}		2.54		ns
PEG1 to GPIO7 Output Edge	t_{P1-IO7}		2.50		ns
PEG2 to GPO2 Output Edge	t_{P2-O2}		2.33		ns
PEG2 to GPIO2 Output Edge	t_{P2-IO2}		2.45		ns
PEG2 to GPIO4 Output Edge	t_{P2-IO4}		2.45		ns
PEG2 to GPIO5 Output Edge	t_{P2-IO5}		2.51		ns
PEG2 to GPIO6 Output Edge	t_{P2-IO6}		2.33		ns
PEG2 to GPIO7 Output Edge	t_{P2-IO7}		2.50		ns

Note 1: The values in the table above are valid when the TX PLL is not bypassed (see section 6.10.2)

9.3.10 Packet Timestamp Latencies

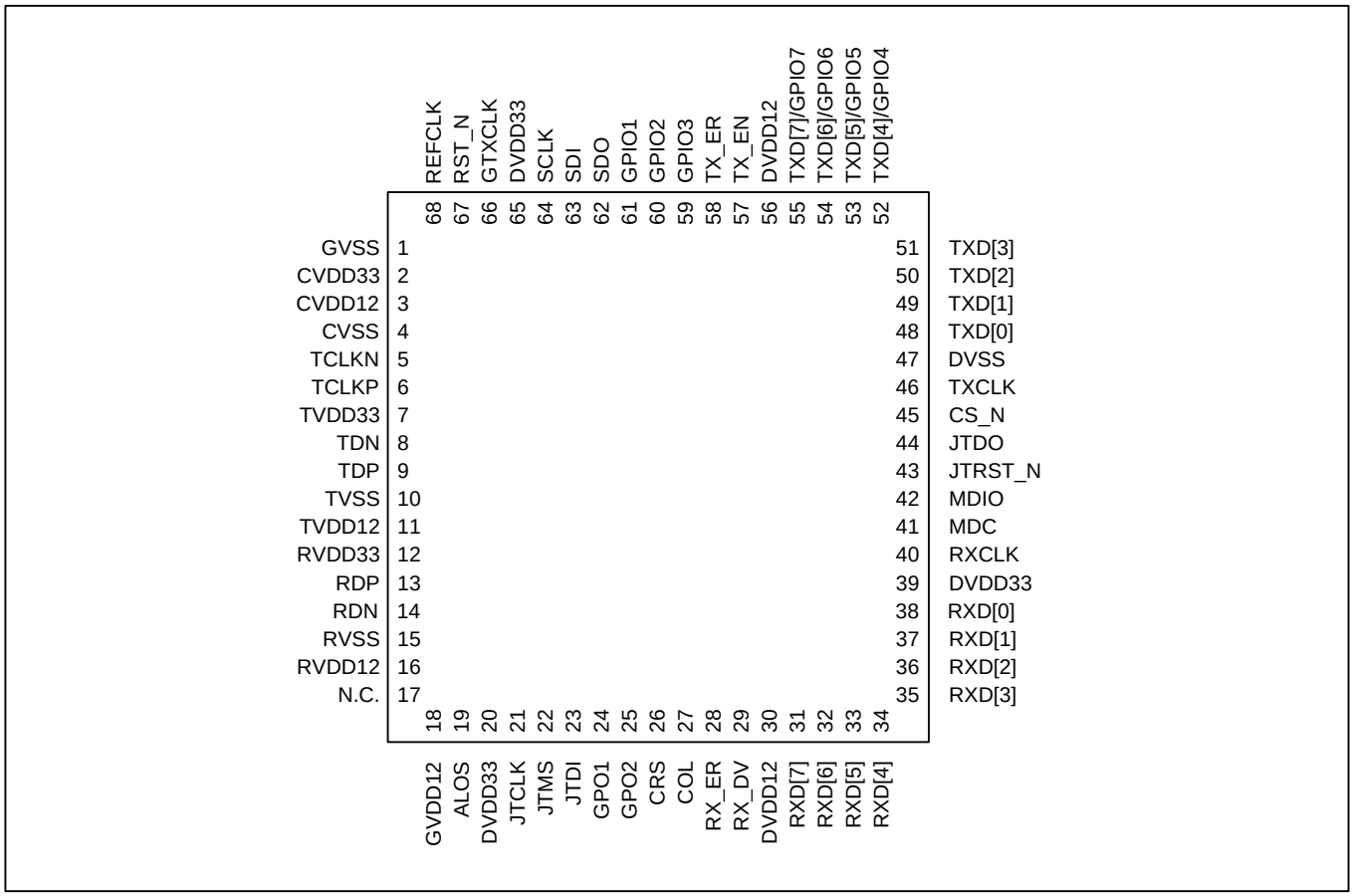
Table 9-24. Transmit/Egress Packet Timestamp to First Bit After SFD on TDP/TDN

MODE OF OPERATION	MIN	TYP	MAX	UNITS
10/100/1000Mbps parallel interface, no egress on-the-fly packet modifications enabled (see section 6.13.7)		46		ns
1000Mbps parallel interface, any egress on-the-fly packet modifications enabled		366		ns
100Mbps parallel interface, any egress on-the-fly packet modifications enabled		3.174		µs
10Mbps parallel interface, any egress on-the-fly packet modifications enabled		31.254		µs

Table 9-25. Receive/Ingress First Bit After SFD on RDP/RDN to Packet Timestamp

MODE OF OPERATION	MIN	TYP	MAX	UNITS
All parallel interface data rates, all modes		9		ns

10. Pin Assignments



N.C. = Not connected internally.

11. Package and Thermal Information

Note: The exposed pad (EP) on the bottom of this package must be connected to the ground plane. EP also functions as a heatsink. Solder to the circuit-board ground plane to achieve the thermal specifications listed below.

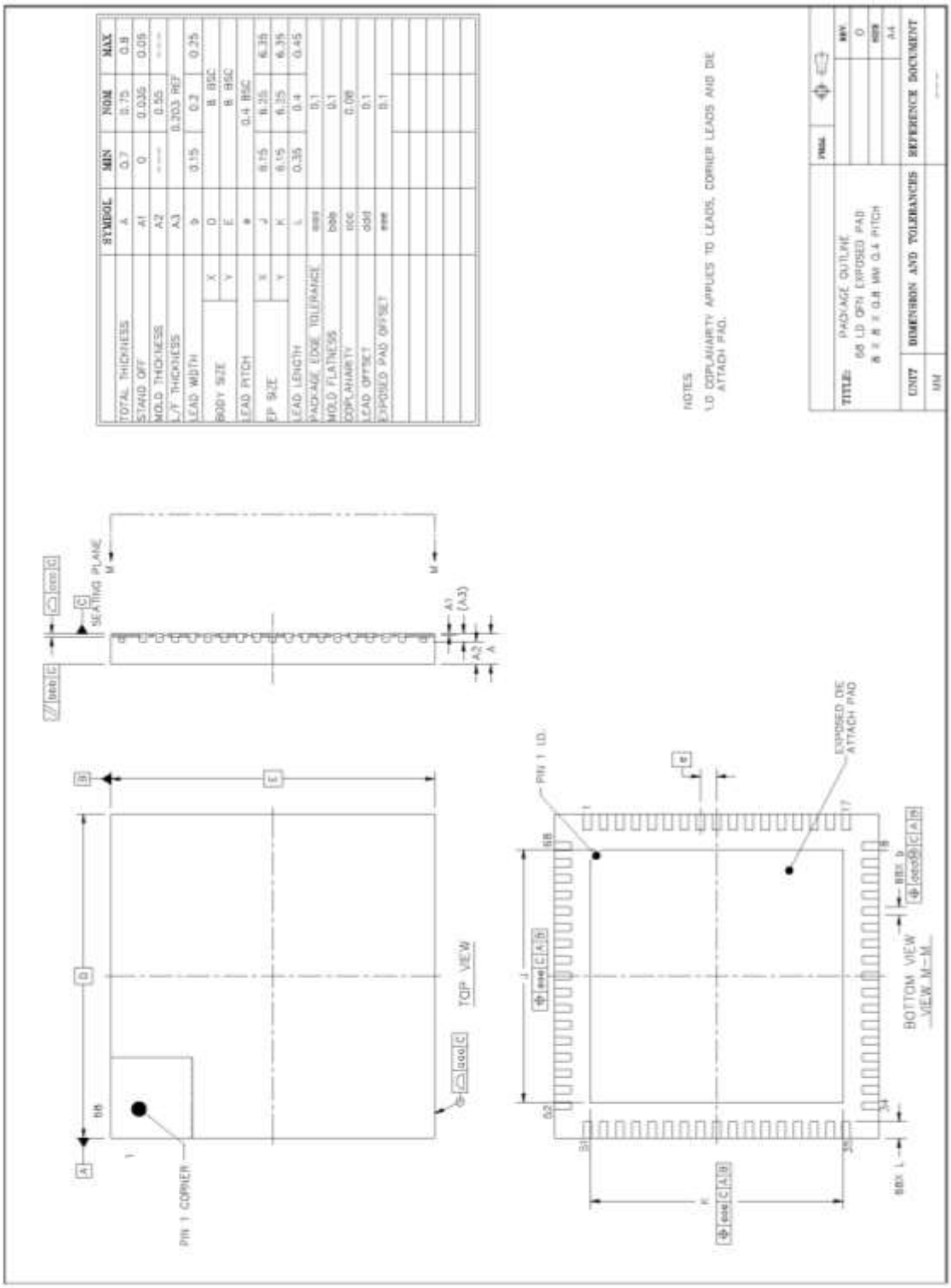


Table 11-1. Package Thermal Properties, Natural Convection

PARAMETER	CONDITIONS	MIN	TYP	MAX
Ambient Temperature	Note 1	-40°C		+85°C
Junction Temperature		-40°C		+125°C
Theta-JA (θ_{JA})	Note 2		20.2 °C/W	
Theta-JC (θ_{JC})			1 °C/W	

Note 1: The package is mounted on a four-layer JEDEC standard test board with no airflow and dissipating maximum power.

Note 2: Theta-JA (θ_{JA}) is the junction to ambient thermal resistance, when the package is mounted on a four-layer JEDEC standard test board with no airflow and dissipating maximum power.

12. Data Sheet Revision History

REVISION DATE	DESCRIPTION
2011-06	Initial release
2012-04	Reformatted for Microsemi. No content change.
2012-06	In the paragraph before Table 6-3, added a note to indicate that when RX_ER and RXD[7:4] are all high at reset the device enters factory test mode. In Table 9-2 changed I_{DDPD33} from 5mA typ to 20mA to correct a typo.
2012-07	Added section 6.15, Power Supply Considerations.
2012-08	Added section 6.16, Startup Procedure.
2013-01	Deleted section 6.15 because the recommendation there is no longer needed when the startup procedure in section 6.16 is followed. In section 9.3.1, loosened input REFCLK duty cycle spec from 48% min, 52% max to 40% min, 60% max and added Note 1. In Table 11-1 added Theta-JC spec. In Table 5-2 added a sentence to the REFCLK pin description to say that it is internally biased with a 10kΩ resistor to 1.2V.
2016-02	Redid section 11 to include the package outline drawing instead of referring to a separate document.
2016-04	In the JIT_DIAG.JIT_PAT description, corrected the 001 decode text from Annex 36A.4 to Annex 36A.1.
2016-11	In Table 5-5 clarified that JTRST_N should be held low during power-up.
2019-04	Added tape-and-reel ordering part number. Change "+" to "2" in ordering part numbers.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; security technologies and scalable anti-tamper products; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif., and has approximately 3,400 employees globally. Learn more at **www.microsemi.com**.