QLPG3116 POS-PHY Level 3 Gigabit Ethernet Controller

Data Sheet

QUICKLOGIC®
Beyond Programmable Logic

Corporate Information

Telephone: 408 990 4000 (US)
416 497 8884 (Canada)
44 1932 57 9011 (Europe)
49 89 930 86 170 (Germany)
852 8106 9091 (Asia)
81 45 470 5525 (Japan)
E-mail: info@quicklogic.com
Support: support@quicklogic.com
Web site: http://www.quicklogic.com/

Revision History

Table 1: Revision History

Revision	Date	Originator and Comments
Rev. A - Preliminary	Apr. 2002	First Release - Paul Micallef and John Kim
Rev. B - Preliminary	Apr. 2002	Changed one pin and one register. - Paul Micallef and John Kim

Copyright Information

Section 1. Features

- Cost effective, integrated full duplex Gigabit Ethernet Controller with POS-PHY Level 3 PHY interface
- Implements POS-PHY Level 3 PHY interface providing connectivity to industry standard or custom L3 devices such as network processors
- Provides up to 1.6 Gbps packet rate transfers with a 104 MHz, 16-bit POS-PHY Level 3 PHY interface that exceeds the requirements for Gigabit Ethernet
- Implements full IEEE 802.3 specification with preamble/SFD generation, frame padding generation and CRC generation that checks on transmit and receive respectively
- Provides seamless interface to standard Gigabit Ethernet PHY device via Gigabit Medium Independent interface (GMII) operation at 125 MHz
- Generic 16-bit processor interface for host management and configuration
- Packet rate decoupling in both directions with 2048-byte receive FIFO and 512-byte transmit FIFO
- POS-PHY Level 3 PHY interface with direct status indication
- Gigabit Ethernet MAC with:
  - Fully automated Pause frame (IEEE 802.3 Annex 31A) generation and termination providing fully automated flow control independent from the user's application
  - Optional Pause frame generation by user application offering flexible traffic flow control
  - Programmable Pause quanta for use within transmitted Pause frames
  - Programmable receive FIFO threshold for Pause frame generation
  - Programmable maximum frame length providing support for any frame length (e.g. Jumbo)
  - Support for VLAN tagged frames according to IEEE 802.1Q
  - Programmable MAC address
  - Programmable Promiscuous Mode support to omit MAC destination address checking on receive
  - Multicast address filtering on receive based on 64 entries hash table reducing higher layer processing load
  - Statistics indicators for frame traffic as well as errors and Pause frames.
- 2.5 V/3.3 V compatible I/O
- Commercial temperature range (0^0-70^0 C)
- 280-pin FPBGA (Fine Pitched Ball Grid Array) package
- IEEE 1149.1 JTAG support

Section 2. Application Examples

With a POS-PHY interface, the QLPG3116 controller can be used in a large variety of applications such as routers, switches, test equipment, or Ethernet over SONET (EOS). The controller's POS-PHY interface provides a seamless connection to Network Processor Units (NPUs) which implement high layer processing services such as packet classification, scheduling, and forwarding. Figure 1 illustrates a typical application.

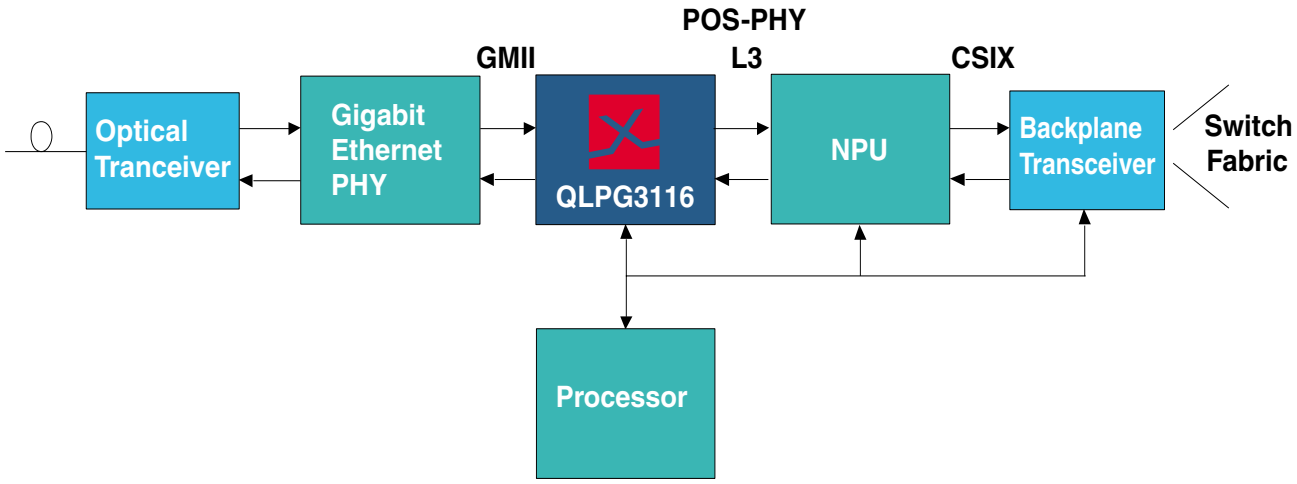


Figure 1: Application Example – Gigabit Ethernet Line Card

Section 3. Block Diagram

Diagram Figure 2 shows the component blocks which make up the QLPG3116.

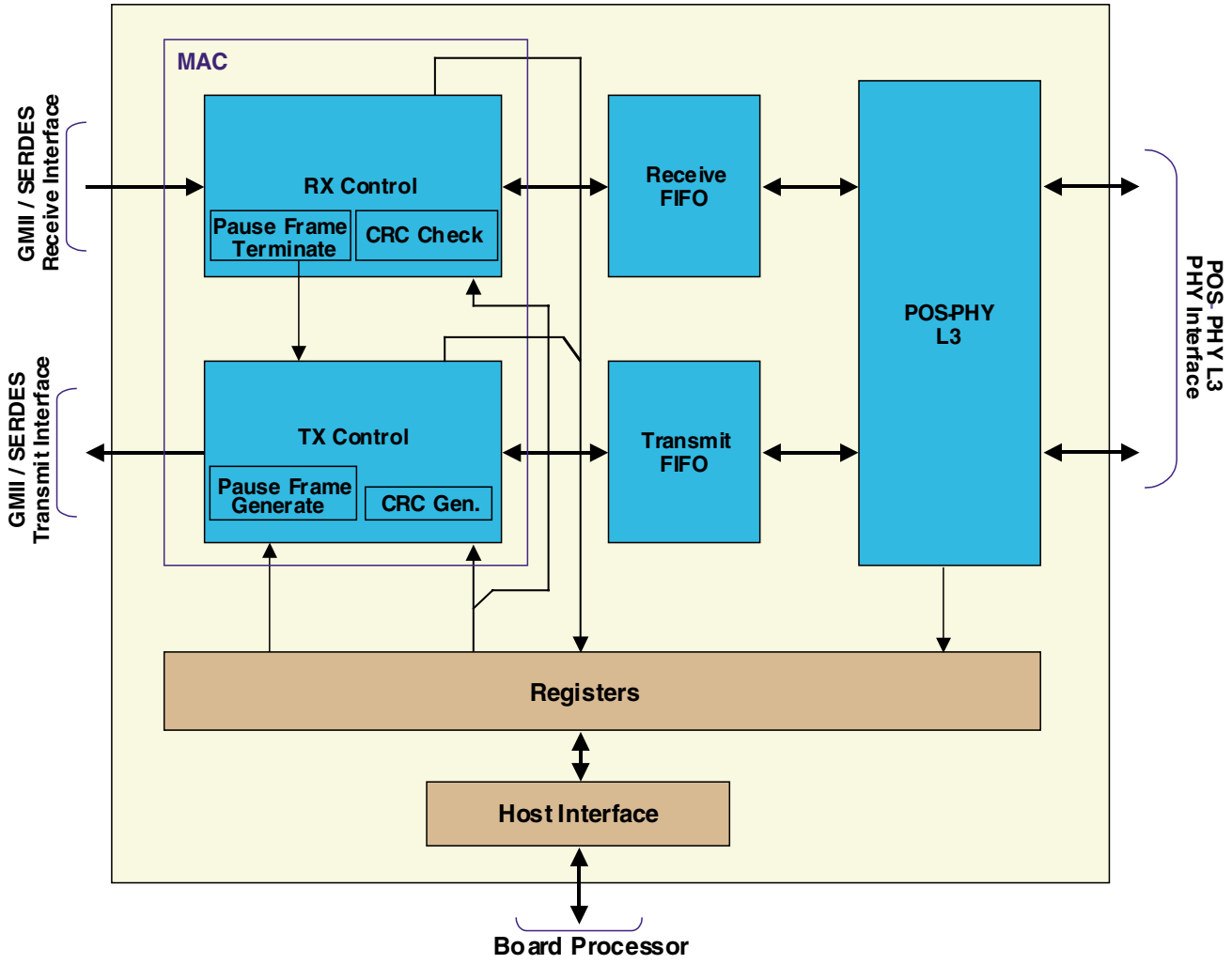


Figure 2: Block Diagram

Section 4. Controller Pinout

The block diagram in Figure 3 shows the pinout for the QLPG3116 controller.

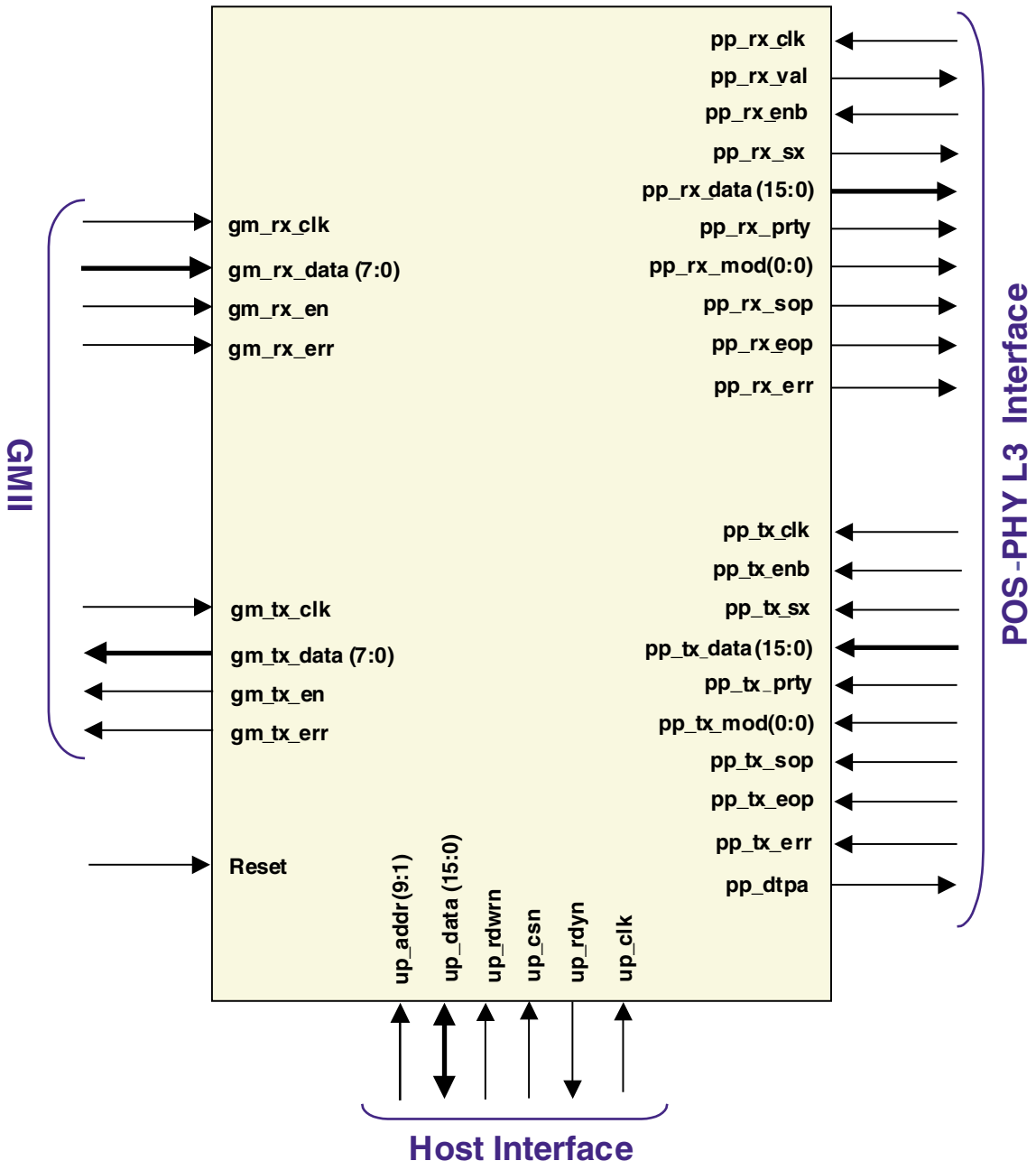


Figure 3: Pinout Diagram

4.1 Signal Descriptions

Table 2: System Global Signals

Global		
reset	In	Active high global reset.

Table 3: POS-PHY Receive Interface

Signal Name	Mode	Description
pp_rx_clk	In	**POS-PHY Receive Clock.** This signal is used to synchronize data transfer transactions between the Link Layer device and the PHY layer device. RFCLK may cycle at a rate up to104 MHz.
pp_rx_val	Out	**Receive Data Valid.** This signal indicates the validity of the receive data signals. RVAL is low between transfers and when RSX is asserted; RVAL is also low when the PHY pauses a transfer due to an empty receive FIFO. When a transfer is paused by holding pp_rx_enb high, pp_rx_val will hold its value unchanged, although no new data will be present on pp_rx_data(15:0) until the transfer resumes. When pp_rx_val is high, the pp_rx_data(15:0), pp_rx_mod(0:0), pp_rx_sop, pp_rx_eop and pp_rx_err signals are valid. When pp_rx_val is low, the pp_rx_data(15:0), pp_rx_mod(0:0), pp_rx_sop, pp_rx_eop and pp_rerr signals are invalid and must be disregarded. The pp_rx_sx signal is valid when pp_rx_val is low.
pp_rx_enb	In	**Receive Read Enable.** The RENB signal is used to control the flow of data from the receive FIFOs. During data transfer, pp_rx_val must be monitored as it will indicate if the pp_rx_data(15:0), pp_rx_prty, pp_rx_mod(0:0), pp_rx_sop, pp_rx_eop, pp_rx_err and pp_rx_sx are valid. The system may deassert pp_rx_enb anytime if it is unable to accept data from the PHY device. When pp_rx_enb is sampled low by the controller, a read is performed from the receive FIFO and the pp_rx_data(15:0), pp_rx_prty, pp_rx_mod(0:0), pp_rx_sop, pp_rx_eop, pp_rx_err, pp_rx_sx and pp_rx_val signals are updated on the following rising edge of pp_rx_clk. When pp_rx_enb is sampled high by the controller, a read is not performed and the pp_rdat(15:0), pp_rx_prty, pp_rx_mod(0:0), pp_rx_sop, pp_rx_eop, pp_rx_err, pp_rx_sx and pp_rx_val signals will remain unchanged on the following rising edge of pp_rx_clk.
pp_rx_data(15:0)	Out	**Receive Packet Data Bus.** This signal carries the packet octets that are read from the receive FIFO and the in-band port address of the selected receive FIFO. RDAT(15:0) is considered valid only when pp_rx_val is asserted.
pp_rx_prty	Out	**Receive Parity Signal.** This signal indicates the odd parity calculated over the pp_rx_data bus.
pp_rx_mod(0:0)	Out	**Receive Word Modulo.** This signal indicates the number of valid bytes of data in pp_rx_data(15:0). The pp_rx_mod(0:0) bus should always be all zero, except during the last double-word transfer of a packet on pp_rx_data(15:0). When pp_rx_eop is asserted, the number of valid packet data bytes on pp_rx_data(15:0) is as follows: pp_rx_mod(0:0) = "0" pp_rx_data(15:0) valid pp_rx_mod(0:0) = "1" pp_rx_data(15:8) valid

Table 3: POS-PHY Receive Interface

pp_rx_sx	Out	**Receive Start Of Transfer.** This signal indicates when the in-band port address is present on the pp_rx_data bus. When pp_rx_sx is high, the value of pp_rx_data(7:0) is the address of the receive FIFO to be selected. Subsequent data transfers on the pp_rx_data bus will be from the FIFO specified by this in-band address.
pp_rx_sop	Out	**Receive Start Of Packet.** This signal delineates the packet boundaries on the pp_rx_data bus. When pp_rx_sop is high, the start of the packet is present on the pp_rx_data bus. This signal is required to be present at the start of every packet and is considered valid when pp_rx_val is asserted.
pp_rx_eop	Out	**Receive End Of Packet.** This signal is used to delineate the packet boundaries on the pp_rx_data bus. When pp_rx_eop is high, the end of the packet is present on the RDAT bus. This signal is required to be present at the end of every packet and is considered valid only when pp_rx_val is asserted.
pp_rx_err	Out	**Receive Error Indicator.** This signal is used to indicate that the current packet is in error. This signal is only asserted when pp_rx_eop is asserted. This signal is considered valid only when pp_rx_val is asserted.

Table 4: POS-PHY Transmit Interface

Signal Name	Mode	Description
pp_tx_clk	In	**POS-PHY Transmit Clock.** This signal is used to synchronize data transfer transactions between the Link Layer device and controller device. May cycle at a rate up to104 MHz.
pp_tx_enb	In	**Transmit Write Enable.** This signal is used to control the flow of data to the transmit FIFOs. When high, the pp_tx_data, pp_tx_mod, pp_ tx_sop, pp_tx_eop and pp_tx_err signals are invalid and are ignored by the controller. The pp_ tx_sx signal is valid and is processed by the controller when pp_ tx_enb is high. When pp_tx_enb is low, the pp_tx_data, pp_tx_mod, pp tx_sop, pp_tx_eop and pp_ tx_err signals are valid and are processed by the controller. Also, the pp_tx_sx signal is ignored by the controller when pp_tx_enb is low.
pp_tx_data(15:0)	In	**Transmit Packet Data Bus.** This bus carries the packet octets that are written to the selected transmit FIFO and the in-band port address to select the desired transmit FIFO. The pp_tx_data bus is considered valid only when pp_tx_enb is simultaneously asserted.
pp_tx_prty	In	**Transmit Bus Parity.** This signal indicates the odd parity calculated over the TDAT bus.
pp_tx_mod(0:0)	In	**Transmit Word Modulo**. This signal indicates the number of valid bytes of data in pp_tx_data(15:0). This signal should always be zero, except during the last double-word transfer of a packet on pp_tx_data(15:0). When pp_tx_eop is asserted, the number of valid packet data bytes on pp_tx_data(15:0) is specified by pp_tx_mod(0:0) as follows: pp_tx_mod(0:0) = “0” pp_tx_data(15:0) valid pp_tx_mod (0:0) = “1”pp_tx_data (15:8) valid

Table 4: POS-PHY Transmit Interface

pp_tx_sx	In	**Transmit Start Of Transfer.** This signal indicates when the in-band port address is present on the pp_ tx_dat bus. When pp_tx_sx is high and pp_ tx_enb is high, the value of pp_ tx_dat(7:0) is the address of the transmit FIFO to be selected. Subsequent data transfers on the pp_ tx_dat bus will fill the FIFO specified by this in-band address.
pp_tx_sop	In	**Transmit Start Of Packet.** This signal is used to delineate the packet boundaries on the pp_tx_data bus. When TSOP is high, the start of the packet is present on the pp_tx_data bus. This signal is required to be present at the beginning of every packet and is considered valid only when pp_tx_enb is asserted.
pp_tx_eop	In	**Transmit End Of Packet.** This signal is used to delineate the packet boundaries on the pp_tx_data bus. When high, the end of the packet is present on the pp_tx_data bus. TEOP is required to be present at the end of every packet and is considered valid only when pp_tx_enb is asserted.
pp_dtpa	Out	**Direct Transmit Packet Available.** This signal provides direct status indication to the Link Layer device of the transmit FIFO. Transitions high when 256 bytes are available in the FIFO. Once high, indicates that the transmit FIFO is not full.
pp_tx_err	In	**Transmit Error Indicator Signal.** This signal is used to indicate that the current packet is in error. This signal is only asserted when pp_tx_eop is asserted. This signal is considered valid only when pp_tx_enb is asserted.

Table 5: Receive GMII Interface

Signal Name	Mode	Receive GMII Interface Description
gm_rx_clk	In	**Receive 125MHz GMII Reference Clock.** This signal is received from the PHY device.
gm_rx_data(7:0)	In	**Receive GMII Data.** This signal is synchronous to gm_rx_clk and valid when gm_rx_en active is received from the Switch device.
gm_rx_en	In	**Receive Data Valid.** When gm_rx_en is asserted (set to "1"), gm_rx_data(7:0) contains valid data. gm_rx_en is asserted from the first preamble byte until the last CRC byte.
gm_rx_err	In	**Transmit Error.** This signal is an indication from the PHY that a physical medium error was detected.

Table 6: Transmit GMII Interface

Signal Name	Mode	Transmit GMII Interface Description
gm_tx_clk	In	**Transmit 125MHz GMII Reference Clock.** This signal is received from an external reference and is also provided to the PHY device.
gm_tx_data(7:0)	Out	**Transmit GMII Data.** This signal is synchronous to gm_tx_clk and valid when gm_tx_en is active.
gm_tx_en	Out	**Transmit Data Valid.** When this signal is asserted (set to "1"), gm_tx_data(7:0) contains valid data. gm_tx_en is asserted from the first preamble byte until the last CRC byte.
gm_tx_err	Out	**Transmit Error.** This signal is an indication from the POS-PHY Interface that an error is detected during frame reception from the NPU.

Table 7: Host Interface

Signal Name	Mode	MPC860 Interface Description
up_clk	In	CPU clock 40MHz (max)
up_addr(9:1)	In	Address Bus.
up_data(15:0)	Bidir	Data Bus.
up_rdwrn	In	Read / Write Strobe
up_rdyn	Out	Data Acknowledge
up_csn	In	Chip Select

Section 5. Application Clock Distribution

An application clock distribution scheme is shown in Figure 4. In this example, the GMII receive clock is generated (recovered from the Gigabit Ethernet line) by the Gigabit Ethernet PHY. A 125 MHz oscillator is implemented to drive the GMII transmit clock. On the POS-PHY interface, a single oscillator drives both the NPU and the QLPG3116 transmit and receive clocks.

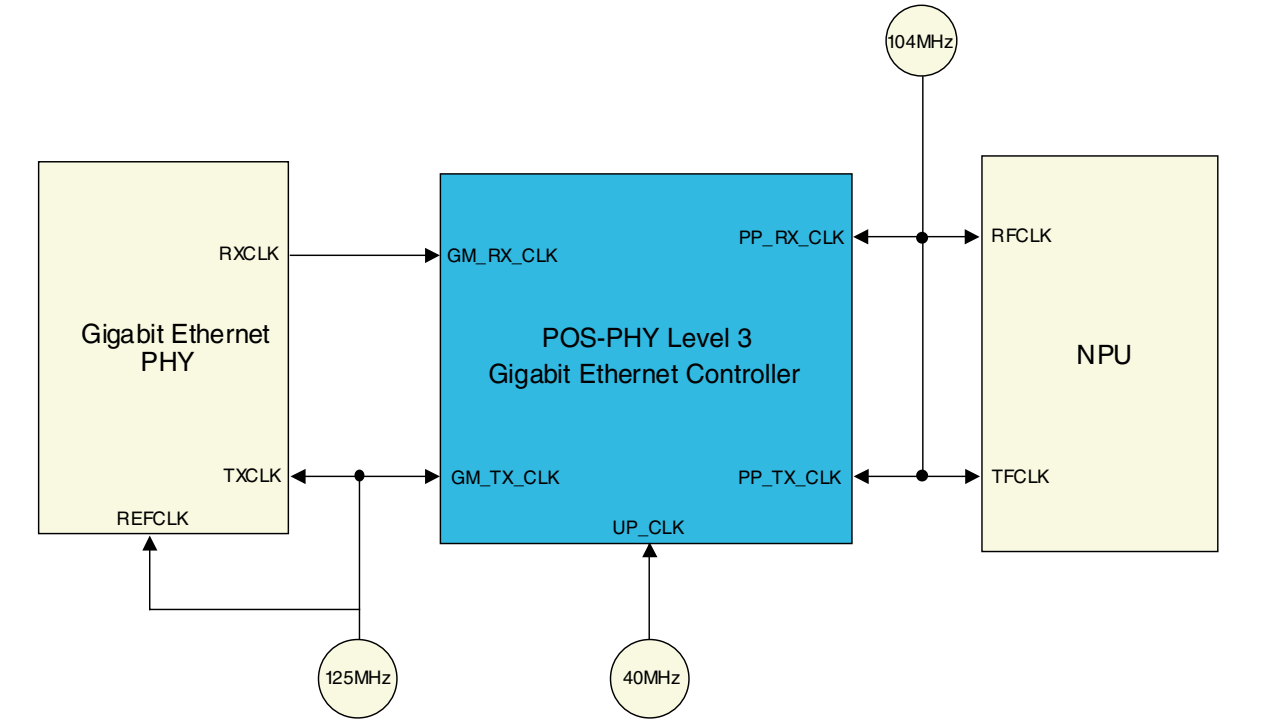


Figure 4: Application Clock Distribution

Section 6. Ethernet MAC Frame Formats

6.1 Overview

The IEEE 802.3 Standard defines the Ethernet Frame format as follows:

An Ethernet frame has a minimum length of 64 bytes and a maximum length of 1518 bytes, excluding the preamble and the SFD bytes. An Ethernet frame format consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Two address fields
- Length or type field
- Data field
- Frame check sequence (CRC value)
- An EXTENSION field is defined only for half-duplex implementations and is not supported by the QLPG3116 controller.

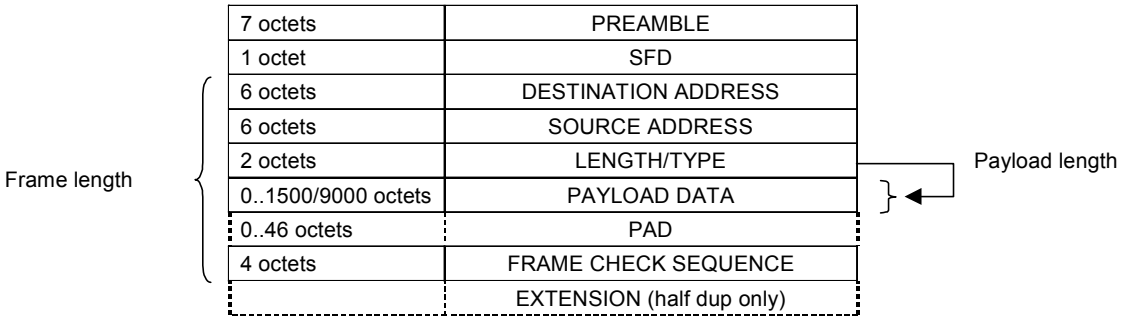


Figure 5: MAC Frame Format Overview

Optionally, MAC frames can be VLAN-tagged with an additional 4-Byte field (VLAN Tag and VLAN Info) inserted between the MAC Source Address and the Type/Length Field. VLAN tagging is defined by the IEEE P802.Q specification. VLAN-tagged frames have a maximum length of 1522 bytes, excluding the preamble and the SFD bytes.

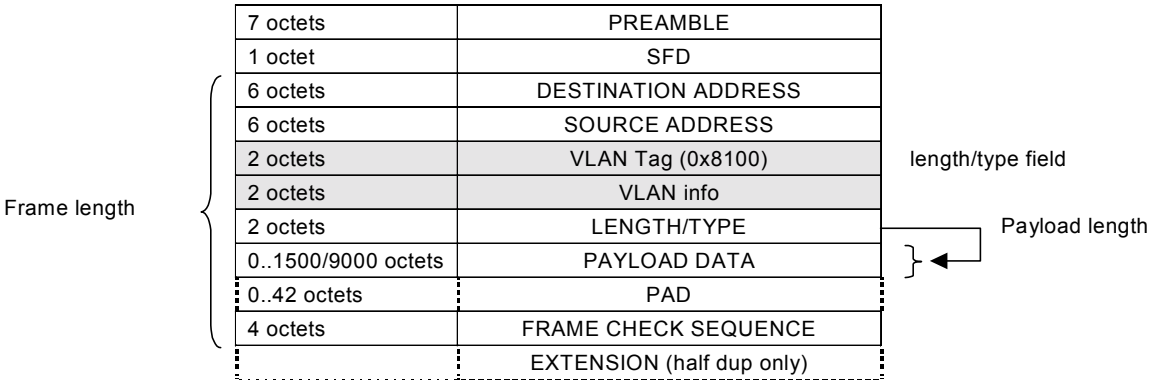


Figure 6: VLAN Tagged MAC Frame Format Overview

Table 8: Mac Frame Definition

Term	Description
Frame Length	The length, in octets, defines the length of the complete frame without preamble and SFD. Legal sizes are: • Standard Ethernet Frame: 64 ... 1518 (payload 0..1500) • Jumbo Frame: 64...9022 (payload 0..9000) • VLAN Frame: 64 ... 1522 (payload 0..1500) • Pause frame: 64 always
Payload Length	If the length / type field indicates a normal frame (not VLAN and not Pause), Payload Length defines the length of the frame's payload section. The most significant byte is sent/received first. If the Length field is set to a value lower than 46, the payload is padded so that the Ethernet minimum frame length requirement (64 Bytes) is met. A non-Jumbo Frame can have a payload Length set to any value between 0 and 1500 Bytes. A Jumbo Frame can have a payload Length set to any value between 0 and 9000 bytes.
Destination and Source Address	48 Bit MAC addresses. The least significant byte is sent/received first and the two first bits (Two Least Significant bits) of the MAC address are used to distinguish MAC frames.

NOTE: Although the IEEE specification defines a maximum frame length, the QLPG3116 controller provides the flexibility to program any value for the frame maximum length.

6.2 Pause Frames

A Pause frame is generated by the receiving device to indicate to the emmitting device that a congestion has occurred. After receiving this indication, the emitting device should stop sending data.

Pause frames are indicated by the Length/Type set to 0x8808. The two first bytes of a Pause frame following the type defines a 16-bit Opcode field always set to 0x0001. A 16-Bit Pause Quanta is defined in the frame payload bytes 2 (byte P1) and 3 (byte P2) as presented in Table 9. The Pause Quanta byte P1 is the most significant.

Table 9: Pause Frame Format (values in hex.)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					

15	16	17	18	19	20	21	22	23	24	25	26	27 - 68
00	00	00	00	00	00	88	08	00	01	hi	lo	00
Source Address						Type		Opcode		P1	P2	pad (42)

69	70	71	72
26	6B	AE	0A
CRC-32			

There is no Payload Length field found within a Pause frame and a Pause frame is always padded with 42 bytes (0x00).

If a Pause frame with a pause value greater than 0 (XOFF Condition) is received, the MAC stops transmitting data as soon the current frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One Pause Quanta fraction refers to 512 bit times.

If a Pause frame with a pause value of 0 (XON Condition) is received, the transmitter is allowed to send data immediately (see "Local Device / FIFO Congestion" on page 21 for details).

Section 7. Gigabit MAC Receive Description

7.1 Overview

The MAC receive engine performs the following tasks:

- Checks frame framing
- Removes frame preamble and frame SFD field
- Discards frames based on Frame Destination address field
- Terminates Pause frames
- Checks frame length
- Removes payload padding if it exists
- Calculates and verifies CRC-32

- Writes received frames in the QLPG3116 controller receive FIFO

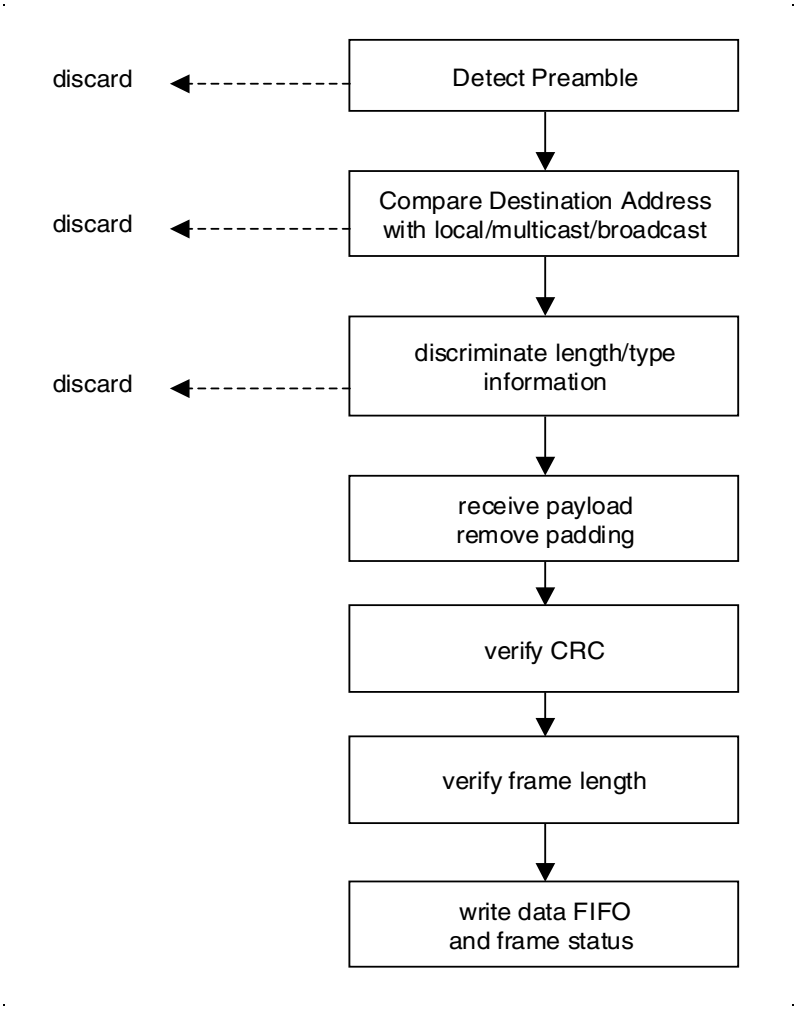


Figure 7: MAC Receive Operation

7.2 Preamble Processing

The IEEE 802.3 standard allows a maximum size of 56 bits (7 bytes) for the preamble, while the QLPG3116 controller allows any arbitrary preamble length. The QLPG3116 controller checks for the Start Frame Delimiter (SFD) byte. If the next byte of the preamble is different from 0x55, and is not 0x5D, the frame is discarded. In this case, the controller waits until the network activity stops before monitoring the network activity for a new preamble.

Although the IEEE standard specifies that frames should be separated at least by 96 clock cycles (Inter Packet Gap or IPG), the QLPG3116 controller is designed to accept frames only separated by 48 GMII clock cycles.

The controller removes all the preamble bytes and the SFD byte.

7.3 MAC Address Check

7.3.1 Overview

The destination address bit 0 (first bit of octet 0) is used to differentiate Multicast and Unicast Addresses in the following ways:

- If bit 0 is set to 0, the MAC address is an individual (Unicast) address.
- If bit 0 is set to 1, the MAC address defines a group address (Multicast Address).
- If all 48 bits of the MAC address are set to 1, it indicates a broadcast address.

7.3.2 Unicast Address Check

If a Unicast address is received, the destination MAC address is compared to the MAC address programmed by the host on the QLPG3116 controller register **MAC(47:0)** (registers **MAC_0**, **MAC_1** and **MAC_2**). If the destination address matches the MAC address, the frame is accepted.

If Promiscuous mode is enabled (command bit promis_en set to "1" in register **COMMAND_CONFIG**), no address checking is performed and all Unicast frames are accepted.

7.3.3 Multicast Address Resolution

Multicast addresses are, in typical implementations, resolved using a software task running on the system host processor. If the multicast address resolution generates acceptable processor load for 10 Mbps or 100 Mbps Ethernet connections with Gigabit Ethernet connections, this task can significantly load the host processor. To remove load from the host processor, the QLPG3116 controller implements a hardware Multicast Address Resolution Engine.

A 64-entry Hash table is calculated and written by the host processor in a 64x1 look-up table (DPRAM). When a Multicast frame is received, the MAC address decoding function sends the frame's destination MAC address to a hashing code generation function which compresses the 46-Bit MAC address in a 6-bit code. The look-up table, programmed by the host processor, is used as a fast matching engine. In one clock cycle, the fast matching engine indicates if the hash code is or is not programmed in the table. If the hash code matching fails, the frame is rejected.

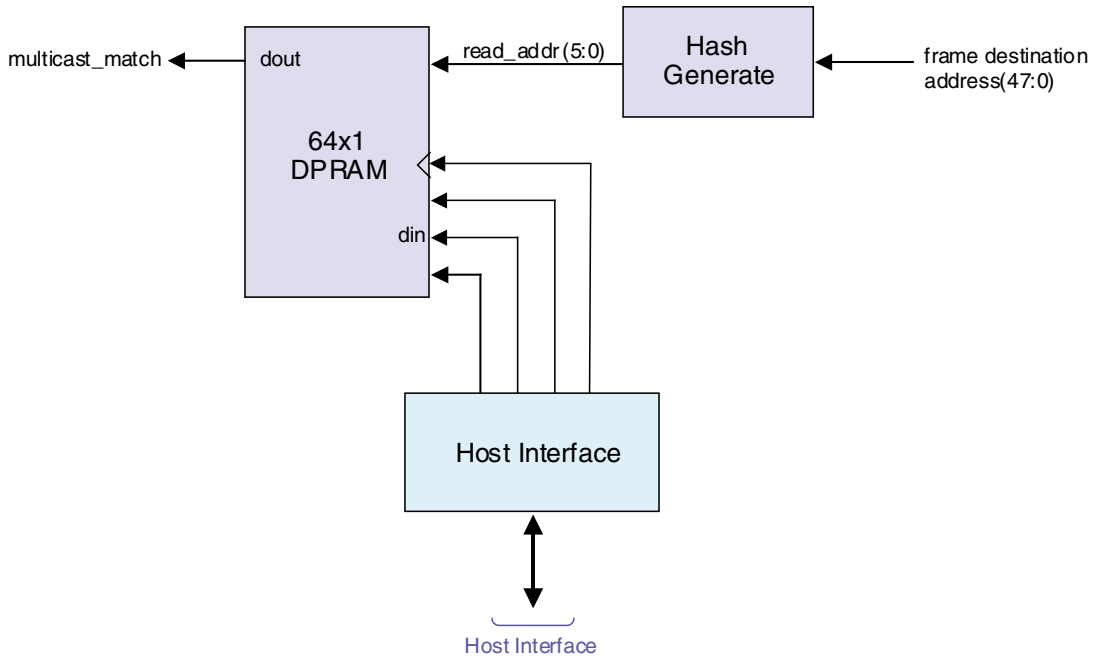


Figure 8: Multicast Address Resolution Overview

To build the Hash table, the host processor generates a 6-bit code for each Multicast address by XOR'ing the MAC address bits as detailed in Table 10. The code is used to address the Look-Up table.

For each code (Look-Up Address) writing a "1" indicates that all the multicast MAC addresses represented by the code should be accepted. Writing a "0" indicates that all the multicast MAC addresses represented by the code should be rejected.

Table 10: Hash Code Generation

Hash Code Bit	Value
0	xor multicast MAC address bits 7:0
1	xor multicast MAC address bits 15:8
2	xor multicast MAC address bits 23:16
3	xor multicast MAC address bits 31:24
4	xor multicast MAC address bits 39:32
5	xor multicast MAC address bits 47:40

The Look-Up table is mapped in the device address space and is programmed via the QLPG3116 controller host interface (see section Host Interface).

7.3.4 Broadcast Frames

Broadcast frames with the Destination address set to the Broadcast address are always accepted by the QLPG3116 controller.

7.4 Frame Length / Type Verification

The frame type/length field is evaluated and the received frames are classified as shown in Table 11.

Table 11: Length/Type Verification

Type Field Value	Description
0 to programmed max value-1	Valid data frame
0x8808	Pause frame
0x8100	VLAN tagged Frame

VLAN-tagged frames are received as normal frames and are completely (including the VLAN tag) pushed in the FIFO. In addition, the frame status word will indicate that a VLAN tagged frame has been received.

If an invalid type field is decoded, the received frame is discarded and the event counter rx_type_err is incremented.

7.5 Pause Frame Termination

Pause frames are terminated within the receive engine and not transferred to the receive FIFO. The Quanta is extracted and sent to the MAC Transmit path via a small internal clock rate decoupling asynchronous FIFO. The Quanta is written only if a correct CRC and a correct frame length are detected by the control state machine. If not, the Quanta is discarded.

When a Pause frame is received, the signal pause_frm_rcv is asserted for one gm_rx_clk cycle.

7.6 Frame Length Check

The complete frame length is checked for frame-too-short and frame-too-long conditions. A frame has a wrong length if it violates any of the following conditions:

- Frame length is smaller than 64 bytes
- Frame length is larger than maximum configured by the register frm_length

If an invalid frame length (greater than the programmed maximum length) is detected, the frame is truncated and is marked as Invalid.

7.7 CRC Check

The CRC-32 field is fully checked, discarded and not forwarded to the FIFO interface. The CRC polynomial, as specified in the 802.3 Standard, is:

```
FCS(X) = X31 +X26 +X23 +X22 +X16 +X12 +X11 +X10 +X8 +X7 +X5 +X4 +X2 +
X1 +1
```

The 32 bits of the CRC value are placed in the FCS field so that the X31 term is the right-most bit of the first octet. The CRC bits are thus received in the following order:
X31, X30,..., X1, X0.

If a CRC-32 error is detected, then event counter **RX_CRC_ERR** is incremented.

7.8 Cell Padding Remove

When a frame is received with a Payload length lower than 46 bytes (42 bytes for VLAN tagged frames), the "0" padding is removed before the frame is written into the data FIFO.

Section 8. Gigabit Ethernet MAC Transmit Description

8.1 Overview

Ethernet frame transmission starts when the Transmit FIFO holds enough data. The FIFO threshold is fixed to 256 bytes (FIFO depth 512 bytes). Once a transfer has started, the transmit engine performs the following tasks:

- Generates Preamble and SFD field before frame transmission
- Generates Pause frames if the Receive FIFO reports a congestion or if the command bit pause_gen (register **COMMAND_CONFIG**) is asserted by the host processor
- Suspends Ethernet frame transfer (XOFF) if a non-zero Pause Quanta is received from the MAC receive path
- Adds padding to the frame if required
- Calculates and appends CRC-32 to the transmitted frame.
- Sends frame with correct Inter Packet Gap (IPG)

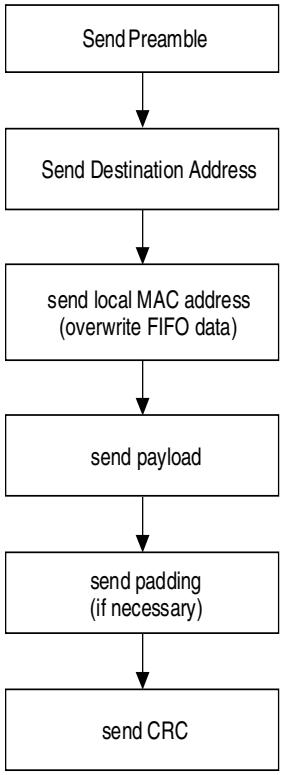


Figure 9: Transmit Operation Overview

8.2 Frame Payload Padding

The IEEE 802.3 Standard defines a minimum frame length at 64 bytes. If the frame sent to the MAC from the user application has a payload size smaller than 46 bytes or 42 bytes (VLAN frame), the MAC automatically inserted padding bytes (0x00) so that frames transmitted to the Ethernet link do not violate the Ethernet minimum frame length specification.

8.3 MAC Address Insertion

On transmit, each MAC address of the frame source is replaced by the address programmed in the register **MAC** (Registers **MAC_0, MAC_1 AND MAC_2**). The programming is done by overwriting any setting on the transmitted frame.

8.4 CRC-32 generation

The CRC-32 field is generated and inserted at the end of each frame transmitted to the GMII interface. The CRC-32 calculation includes pad bytes if padding is required to meet the Ethernet Frame's minimum length requirement.

The CRC polynomial, as specified in the 802.3 Standard, is:

```
FCS(X) = X31 +X26 +X23 +X22 +X16 +X12 +X11 +X10 +X8 +X7 +X5 +X4 +X2
+X1 +1
```

The 32 bits of the CRC value are placed in the FCS field so that the X31 term is the right-most bit of the first octet. The CRC bits are thus transmitted in the following order: X31, X30,..., X1, X0.

8.5 Inter-Packet Gap

An Inter-Packet Gap (IPG) of at least 96 bit times is maintained both before and after transmission of a frame.

Section 9. Ethernet Flow Control

9.1 Overview

The QLPG3116 controller's Flow Control engine handles the following conditions:

- **Remote Device Congestion:** The remote device connected to the same Ethernet segment as the controller reports a congestion requesting the controller to stop sending data.
- **Controller FIFO Congestion:** When the controller's receive FIFO reaches the almost full threshold (user programmable via the register **PAUSE_THRESHOLD**), the controller sends a Pause frame back to the remote device requesting data transfer to be stopped.
- **Local Device Congestion:** Any device connected to the controller can request (via the register **COMMAND_CONFIG**) the remote device to stop transmitting data.

9.2 Remote Device Congestion

When the MAC Transmit control gets a valid Pause Quanta from the receive path, the MAC completes the transfer of the current frame and stops sending data for the amount of time specified by the Pause Quanta in 512 bit time increments.

Frame transfer resumes when the time specified by the Quanta has expired and if no new Quanta value has been received or if a new Pause frame with a Quanta value set to 0x0000 is received.

9.3 Local Device / FIFO Congestion

Pause frames are generated by the MAC transmit engine when:

- the local receive FIFO is not able to receive more than a pre-defined number of words (user programmable via the register **PAUSE_THRESHOLD**). The Receive FIFO asserts its Almost Full flag (internal). A Pause frame is generated automatically when the current frame transfer is completed and as long as the Receive FIFO maintains its Almost Full Flag asserted.
- a Pause frame generation is requested when the local host processor asserts the PAUSE_GEN command bit in the register **COMMAND_CONFIG**. A Pause frame is generated when the current frame transfer is completed and as long as the command bit PAUSE_GEN is asserted.

When a Pause frame is generated, the frame's Pause Quanta (Payload byte P1 and P2) is filled with the setting given at the QLPG3116 controller configuration register **PAUSE_QUANT**. The Source Address is set to the MAC address set in the register **MAC(47:0)** and the destination address is set to the fixed multicast address 0x0180C2000001.

Generated Pause frames are compliant to the IEEE 802.3 annex 31A&B.

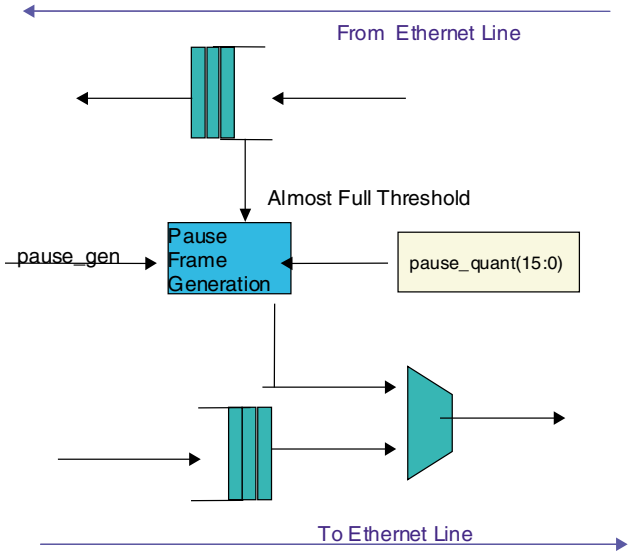


Figure 10: Pause Frame Generation Overview

NOTE: Although the flow control mechanism should prevent any FIFO overflow on the MAC Controller receive path, the QLPG3116 receive FIFO is protected. When an overflow is detected on the receive FIFO, the current frame is truncated with an error indication set in the frame status word. The frame should subsequently be discarded by the user application.

Section 10. GMII Interface Operation

10.1 Transmit

On Transmit, all data transfers are synchronous to the gm_tx_clk rising edge.

The GMII data enable signal gm_tx_en is asserted to indicate the start of a new frame and remains asserted until the last byte of the frame is present on gm_tx_data bus. Between frames, gm_tx_en remains de-asserted. This is illustrated in the Figure 11.

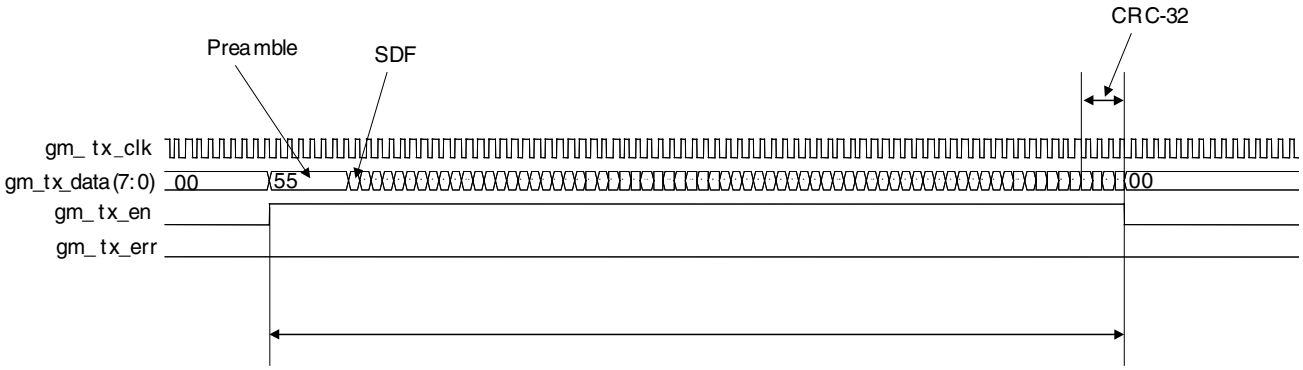


Figure 11: GMII Transmit Operation

If the application or MAC detects an error on the frame transmitted, it asserts the GMII error signal, gm_tx_err, for one clock cycle at any time during the packet transfer to inform the PHY of this event. This is shown in Figure 14.

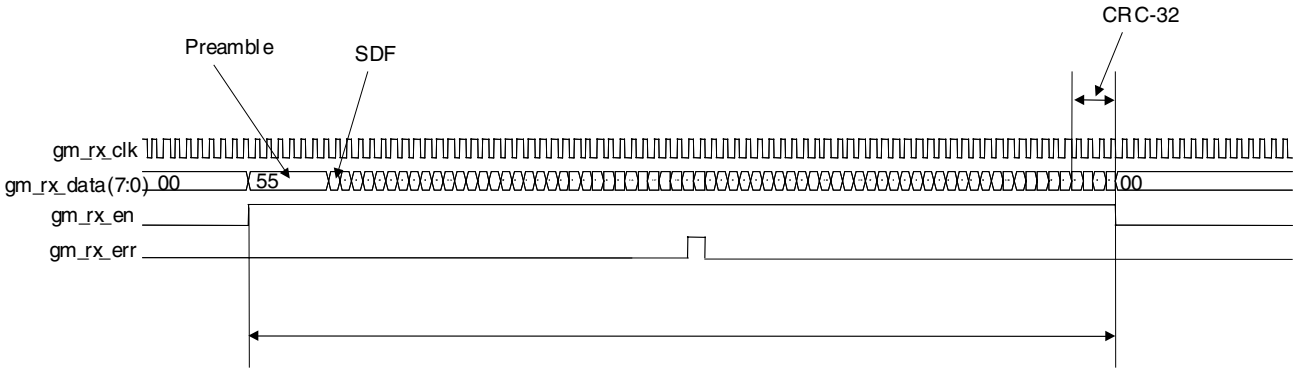


Figure 12: GMII Transmit Operation - Erroneous Frame

10.2 Receive

On Receive, all signals are sampled on the **gm_rx_clk** rising edge. The GMII data enable signal **gm_rx_en** is asserted by the Gigabit Ethernet PHY to indicate the start of a new frame and remains asserted until the last byte of the frame is present on **gm_rx_data** bus. Between frames, **gm_rx_en** remains de-asserted, as shown in Figure 13.

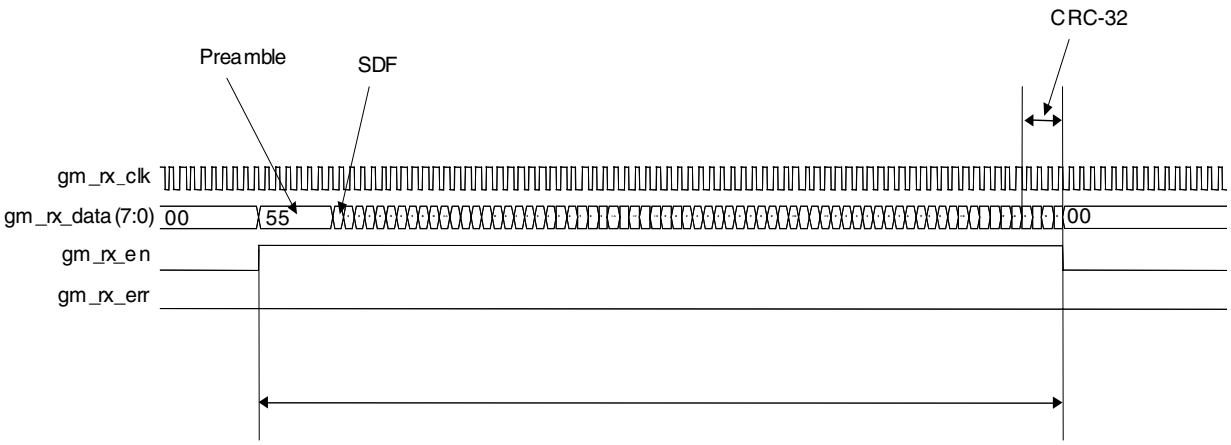


Figure 13: GMII Receive Operation

If the Gigabit Ethernet PHY detects an error on the frame received from the line, the PHY asserts the GMII error signal, **gm_rx_err**, for one clock cycle at any time during the packet transfer. This is illustrated in Figure 14.

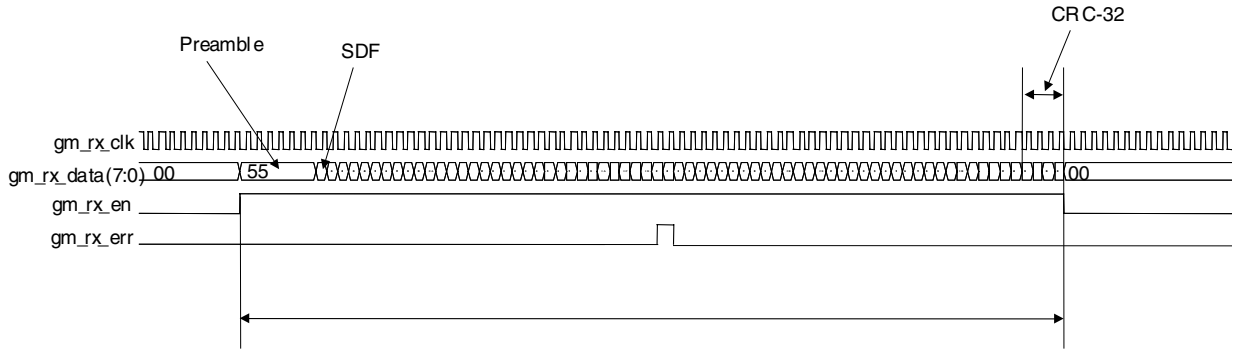


Figure 14: GMII Receive Operation - Erroneous Frame

Section 11. POS-PHY L3 Interface Operation

11.1 Transmit Operation (Egress) – Direct Status Indication

If the controller Transmit FIFO has space to receive a 256-byte data section, the controller informs the Link Layer device by asserting the Transmit Packet Available signal (**pp_dtpa**), indicated by number **1** on Figure 15.

The Link Layer device then writes the PHY address (Should be 0x"0000") on the data bus (pp_tx_data(15:0)) with the Transmit Start of Transfer signal (**pp_tx_sx**) asserted (set to "1"), indicated by number **2**.

To start data transfer, the Link Layer device asserts (Set to "0") the active low transmit enable signal (pp_tx_enb), indicated by **3**.

The first word of a packet is driven on pp_tx_data(15:0) together with the start of packet signal (pp_tx_sop) asserted (Set to "1"), indicated by **4**.

Data on the POS-PHY interface are valid when the POS-PHY transmit enable signal (pp_tx_enb) is asserted (Set to '0') by the Link Layer device, indicated by **5**.

Parity is provided by the Link Layer device on pp_tx_prty, indicated by **6,** and is checked by the Controller if enabled.

To complete the packet transmission, the Link Layer asserts the End of Packet signal (pp_tx_eop) and indicates the valid portion of the data bus on pp_tx_mod(0:0), indicated by **7**.

If no packet follows, the Link Layer de-asserts (sets to "1") the transmit enable signal (pp_tx_enb), indicated by **8**.

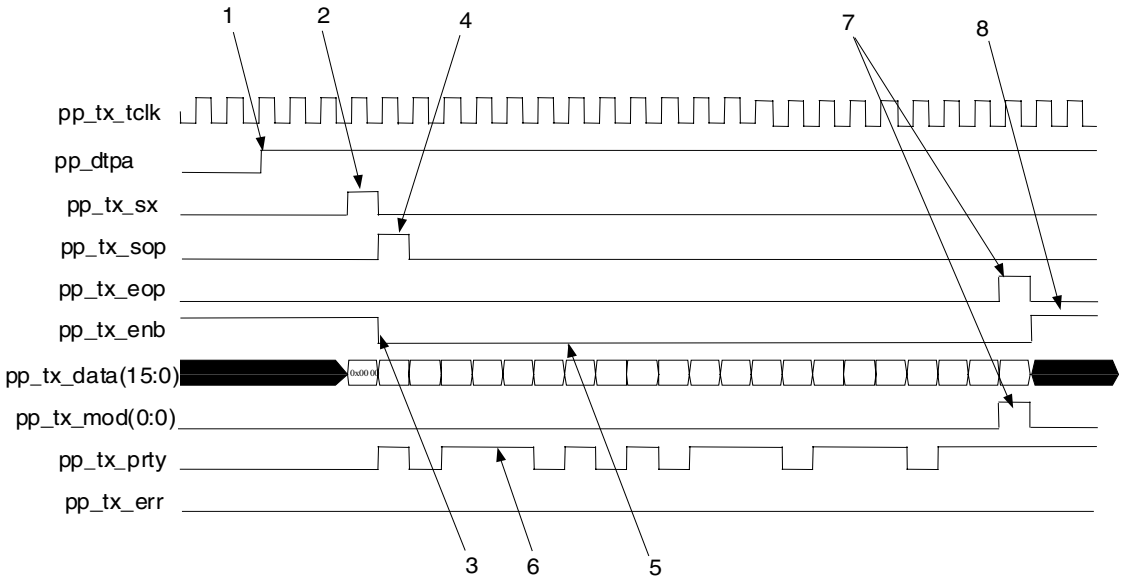


Figure 15: POS-PHY Transmit

11.1.1 Transmit with Link Layer Data Transfer Pause

The Link Layer device pauses the data transfer by de-asserting the POS-PHY Write Enable signal (pp_tx_enb), indicated by number **1** on Figure 16.

The device samples low pp_tx_enb (on pp_tx_clk rising edge) and stops writing data to the FIFO. When the Link Layer device re-asserts pp_tx_enb, the transfer re-starts, indicated by number **2**.

The data driven by the Link Layer device and pp_tx_enb are sampled in by the device on pp_tx_clk rising edge.

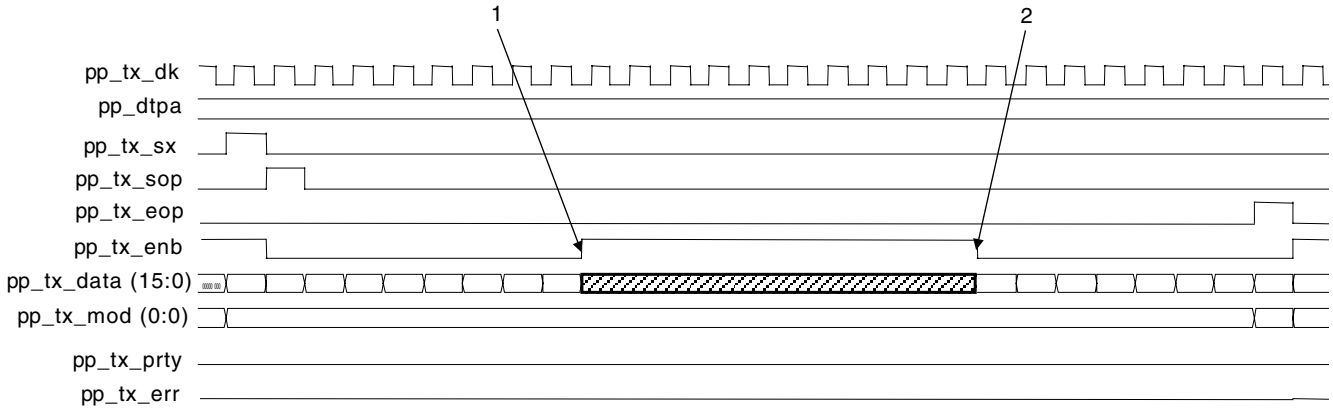


Figure 16: POS-PHY Transmit with Link Layer Wait State

11.1.2 Transmit with POS-PHY Error

If an error is detected, the Link Layer device can mark a packet with the POS-PHY Error signal (pp_tx_err). To be valid, pp_tx_err is asserted (set to "1") together with the end-of-packet signal, indicated by number **1** on Figure 17.

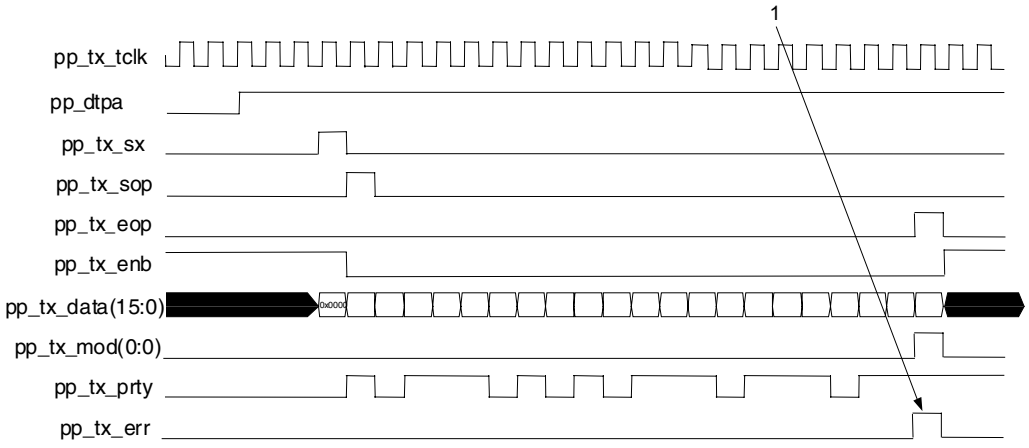


Figure 17: POS-PHY Transmit with Error

11.1.3 Back to Back Packet Transmit

The Link Layer can transmit consecutive packets if the QLPG3116 controller FIFO can receive subsequent data. The QLPG3116 POS-PHY interface fully supports back-to-back packet reception without limitations.

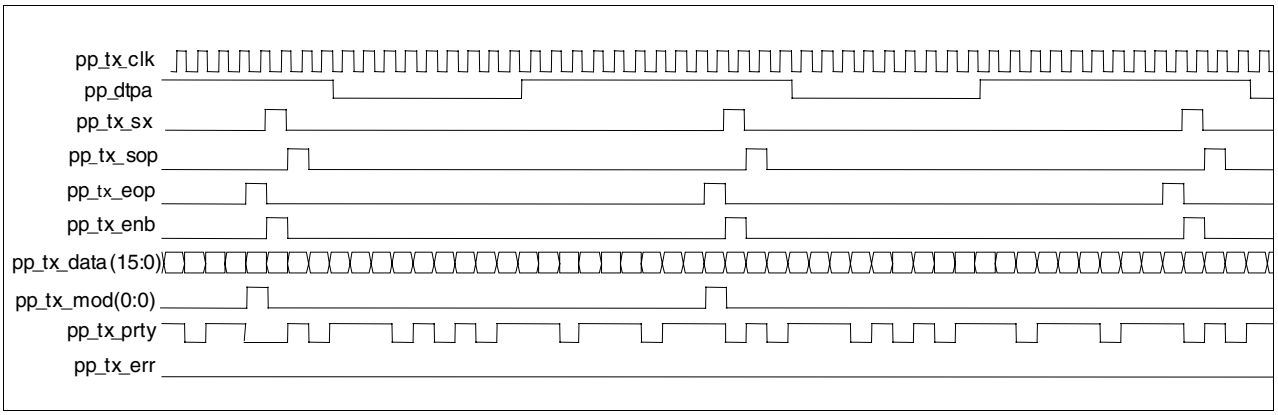


Figure 18: POS-PHY Back-to-Back Transmit

11.2 Receive POS-PHY Interface Operation

When either 256 bytes of data or a data with an end-of-packet indication is available in the device FIFO, and if the Link Layer device is able to receive data, the active low enable signal is asserted (pp_rx_enb set to "0"), indicated by number 1 on Figure 19. The controller then sends the in-band address (which is always 0x00), by asserting pp_rx_sx, indicated by number **2**.

The device asserts pp_rx_val to indicate that valid data is transferred, indicated by number **3**.

The first word of a packet is driven on pp_rx_data(15:0) together with the start of packet signal (pp_rx_sop) asserted (set to "1") , indicated by number **4**.

The packet transfer is completed when the end of packet signal (pp_rx_eop) is asserted, indicated by number **5**.

The word modulo bit (pp_rx_mod(0:0)) defines, for the last word of a packet, the significant bytes, , indicated by number **6**.

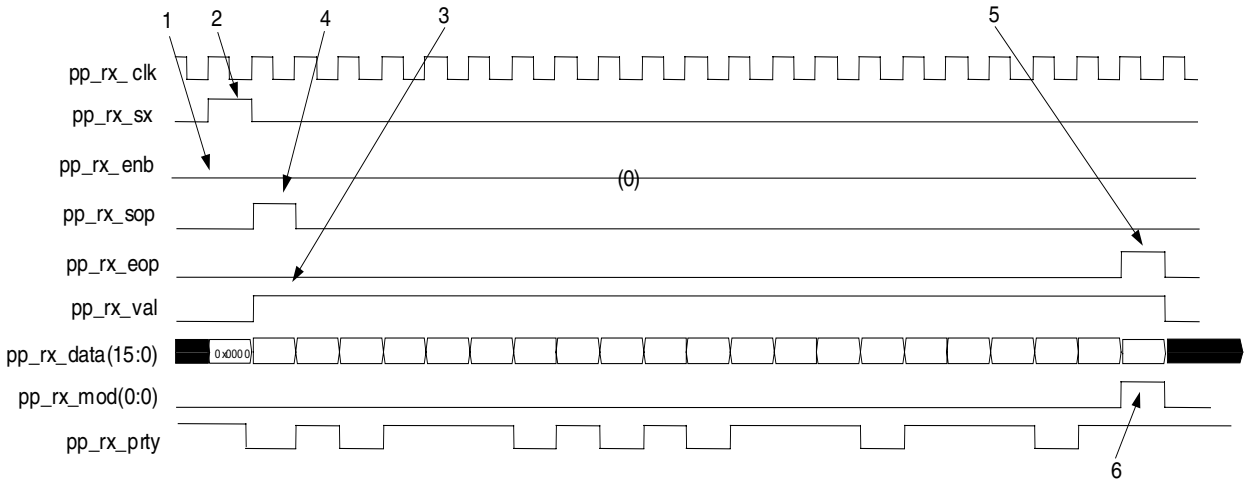


Figure 19: POS-PHY Receive

Between two consecutive packets sent to the link layer device, the controller inserts a gap of two clock cycles. This is implemented so that the controller can be connected to any Link Layer device.

11.2.1 Receive with PHY Pause

The controller pauses a transfer by de-asserting the Data Valid signal (pp_rx_val), indicated by number **1** on Figure 20. The controller de-asserts pp_rx_val when the receive FIFO reaches the almost empty threshold (4 bytes without end of packet or less remaining in the FIFO). The transfer re-starts when
256 bytes in the device FIFO or when a data with an end of packet are available in device FIFO.

To resume the transfer on the POS-PHY interface, the controller asserts pp_rx_val, indicated by number **2**.

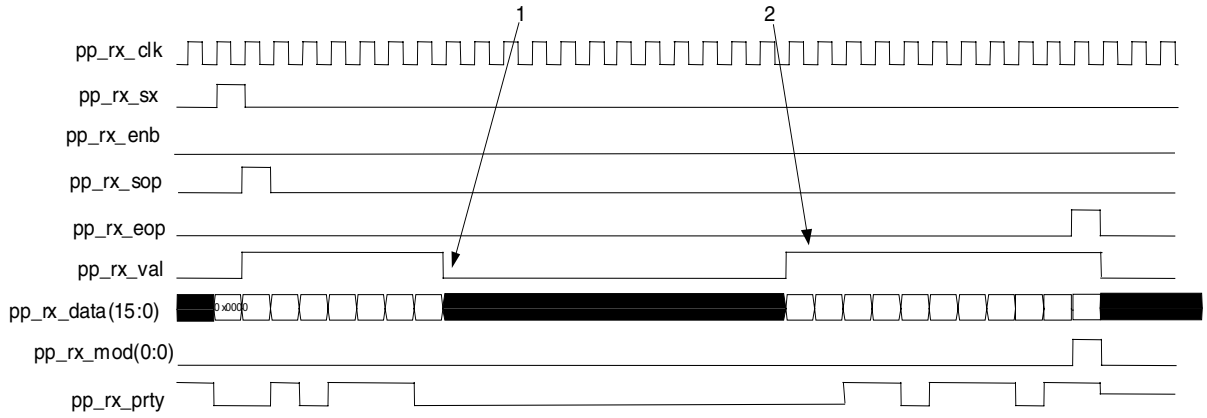


Figure 20: POS-PHY Receive with PHY Pause

11.2.2 Receive with Link Layer Pause

The link layer device pauses a transfer by de-asserting the active low receive enable signal (pp_rx_enb), indicated by number **1** on Figure 21.

The controller samples pp_rx_enb on the following clock edge and stops driving data, indicated by number **2**.

The link layer device resumes the transfer by asserting the active low receive enable signal (pp_rx_enb), indicated by number **3**.

The controller samples pp_rx_enb on the following clock edge and start driving new data, indicated by number **4**.

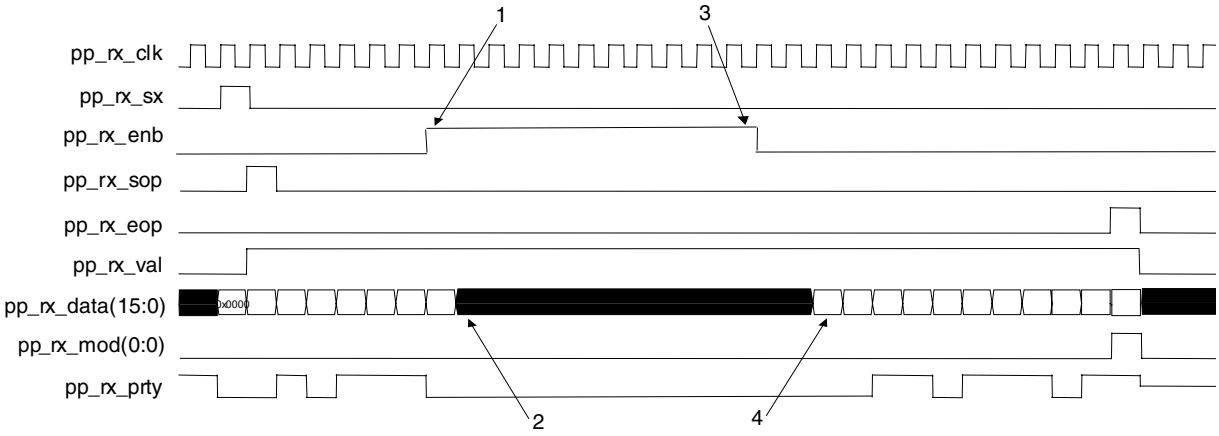


Figure 21: POS-PHY Receive with Link Layer Pause

11.2.3 POS-PHY Receive with Error

When an error is detected, the PHY application can mark a packet invalid. This packet is then transmitted on the POS-PHY interface with the Packet Error signal asserted (pp_rx_err) together with the last word of the packet, as indicated by number **1** on Figure 22.

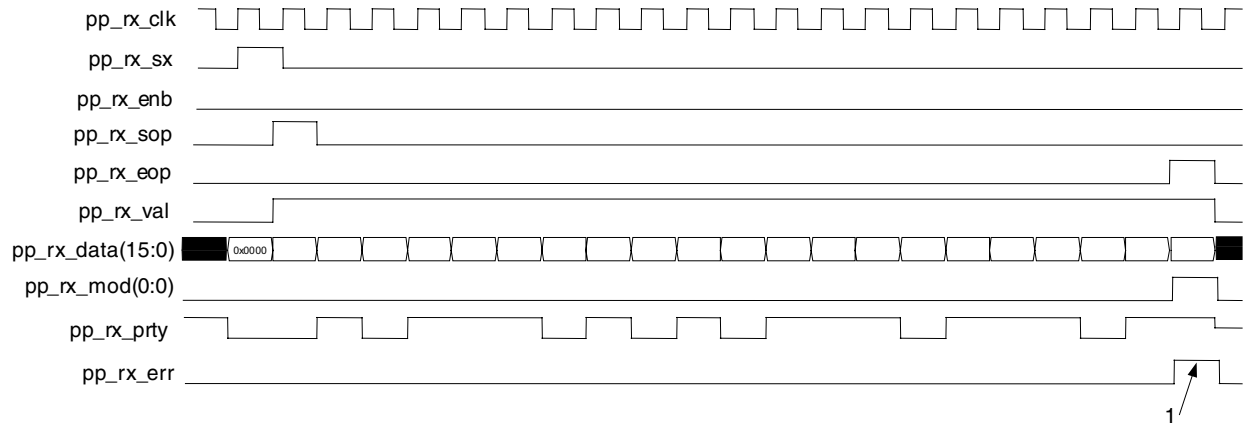


Figure 22: POS-PHY receive with Error

Section 12. Host Interface

12.1 Overview

The controller implements a simple 16-bit processor interface which can be connected to various industry standard processors such as the Motorola MPC8260.

12.2 Register Read

To read a controller register, the host processor selects the controller by asserting the active low Chip Select pin (controller up_csn), drives up_rdwrn to "1", and puts the register address on up_addr(9:1). The controller indicates that valid data is on the bus (up_data(15:0) by asserting the active low ready signal up_rdyn. To complete the read cycle, the host de-asserts up_csn. Figure 23 illustrates this read process.

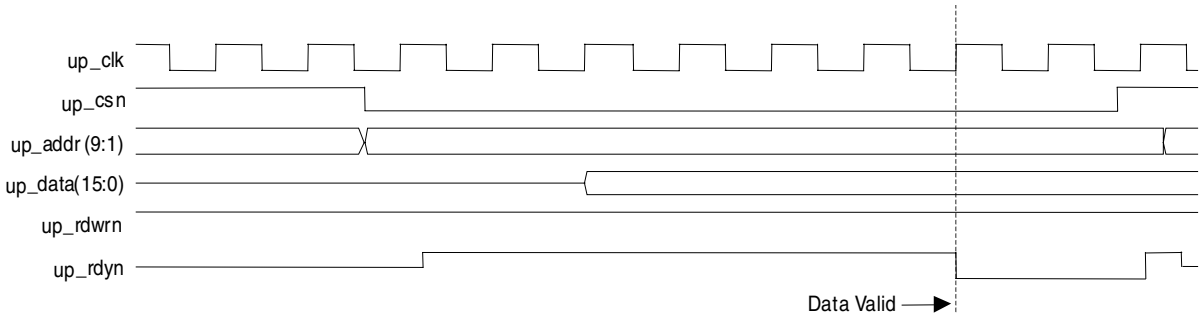


Figure 23: Host Interface - Read Access

12.3 Register Write

To write data in a controller register, the host processor selects the controller by asserting the active low Chip Select pin (controller up_csn), drives up_rdwrn to '0', puts the register address on up_addr(9:1) and the data on up_data(15:0). The controller indicates that the data on up_data(15:0) has been registered by asserting the active low ready signal up_rdyn. The host can complete read cycle by de-asserting up_csn. Figure 24 illustrates this process.

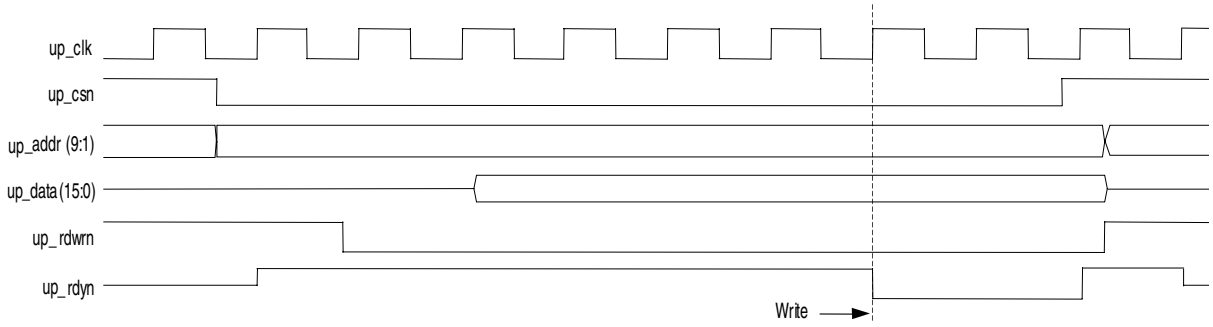


Figure 24: Host Interface - Write Access

Section 13. JTAG Support

The QLPG3116 controller supports IEEE Standard 1149.1a. The following public instructions are supported:

- **Extest Instruction:** This instruction performs a PCB interconnect test. This test places a device into an External Boundary Test mode, selecting the **BOUNDARY SCAN** register to be connected between the TAP's Test Data In (TDI) and Test Data Out (TDO) pins. Boundary Scan cells are preloaded with test patterns (via the Sample/Preload Instruction) and Input Boundary cells capture the input data for analysis.
- **Sample/Preload Instruction:** This instruction allows a device to remain in its Functional mode while selecting the **BOUNDARY SCAN** register to be connected between the TDI and TDO pins. For this test, the **BOUNDARY SCAN** register can be accessed via a data scan operation, allowing users to sample the functional data entering and leaving the device.
- **Bypass Instruction:** This instruction allows data to skip a device's boundary scan entirely, so the data passes through the **BYPASS** register. The Bypass instruction allows users to test a device without passing through other devices. The **BYPASS** register is connected between the TDI and TDO pins, allowing serial data to be transferred through a device without affecting the operation of the device.

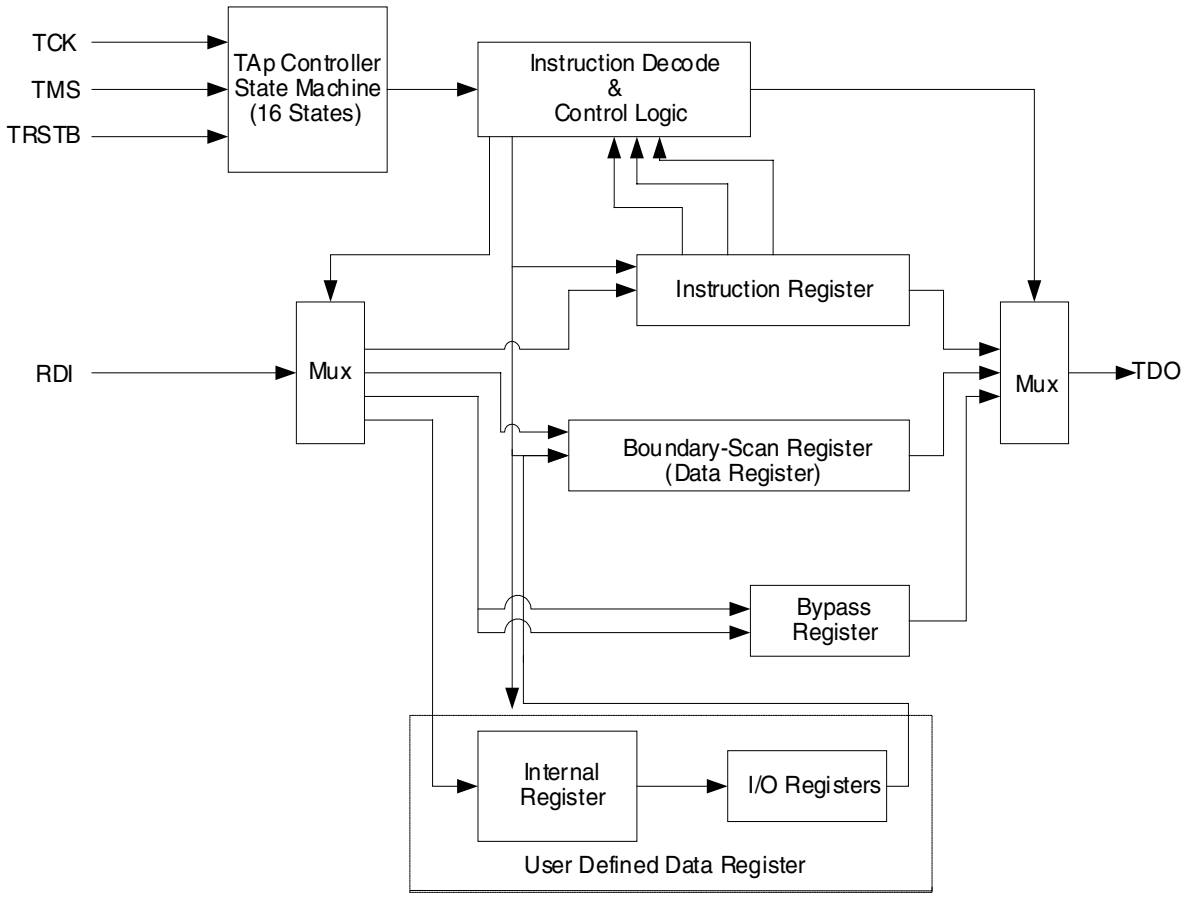


Figure 25: JTAG Block Diagram

Section 14. Register Map

This section provides information on the QLPG3116 registers in Table 12. Each register is 16 bits wide. Details of the **COMMAND_CONFIG** register are shown in Figure 26 and in Table 13.

14.1 Overview

Table 12: Device Register Map

Address	Register	Description
0x00	REV	**Design Revision** (RO). Read only register which contains the device design revision.
0x02	SCRATCH	**Scratch Register** (RW). The scratch register provides a memory location available for the host processor to test device memory operation.
0x04	COMMAND_CONFIG	**Command Register** (RW). Used by the host processor to control and configure the POS Gigabit Ethernet Controller.
0x06	MAC_0	**MAC Address 16-Bit Word 0** (RW). Program MAC address bit 15 to 0 (Octet 1,0).
0x08	MAC_1	**MAC Address 16-Bit Word 1** (RW). Program MAC address bit 31 to 16 (Octet 3,2).
0x0A	MAC_2	**MAC Address 16-Bit Word 2** (RW). Program MAC address bit 47 to 32 (Octet 5,4).
0x0C	FRM_LENGTH	**Maximum Frame Length** (RW). Defines a 14-Bit maximum frame length used by the MAC receive logic to check frames. Bit 13 and 15 of the register are not significant.
0x0E	PAUSE_QUANT	**Receive Pause Quanta** (RW). In increments of 512 Ethernet bit time, the pause quanta is set in each Pause frame generated to the remote Ethernet device.
0x10	PAUSE_THRESHOLD	Defines, in sections of 32 bytes, the receive FIFO threshold from which Pause frames are generated. When set to 0, the Pause frame generation is disabled.
0x12	RX_CRC_ERR	Number of frames received with a CRC-32 Error (RRST).
0x14	RX_GMII_ERR	Number of frames received with the GMII error signal gm_tx_err asserted on the GMII receive interface (RRST).
0x16	RX_LENGTH_ERR	Number of frames received with a length error on the GMII receive interface (RRST).
0x18	TX_PAUSE_FRM_CNT	Number of Pause frames transmitted by the MAC.
0x1A	RX_FRM_ERR	Number of frames received with an invalid SFD field on the GMII receive interface (RRST).
0x1C	RX_TYPE_ERR	Number of frames received with an invalid Type field on the GMII receive interface (RRST).
0x1E	RX_FRM_CNT	Number of frames (Unicast, Multicast or Broadcast) received from the Ethernet line (RRST).
0x20	RX_PAUSE_FRM_CNT	Number of Pause frames received from the Ethernet line (RRST).

0x22	TX_FRM_CNT	Number of frames (Unicast, Multicast or Broadcast) transmitted to the Ethernet line (RRST).
0x24	TX_POS-PHY_ERR	Number of frames received with the POS-PHY error signal pp_tx_err asserted on the GMII transmit interface (RRST).
0x26	TX_POS-PHY_PERR	Number of frames received on the POS-PHY interface with an odd parity error (RRST).
0x28 to 0x7E	Reserved	Always return 0x00 when read by the host processor (RO).
0x80 to 0xFE	HASH_TABLE_LOAD	Multicast address resolution table mapped in the controller address space (WO). When programming the table, only bit “0” is significant of the host interface 16-Bit data bus. If a “1” is written, all multicast addresses represented by the hash code (Address bits 0 to 3) are accepted by the controller. If a “0” is written, matching multicast addresses are rejected.

RO: Read Only register.
WO: Write Only register.
RW: Read/Write register.
RRST: Read only register with reset on read.

14.2 Register Description

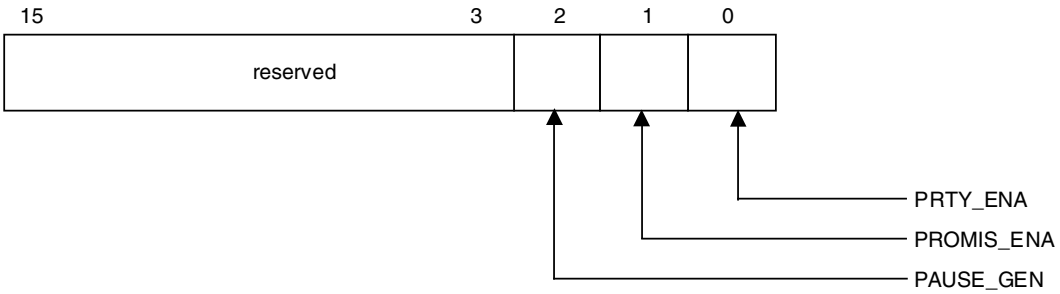


Figure 26: COMMAND_CONFIG Register

Table 13: COMMAND_CONFIG Register Description

Bit Name	Description
PRTY_ENA	**POS-PHY Parity Enable** (RW). When set to “1”, enables parity generation/check on the device interface. When set to “0” (Reset value), disables parity on the device POS-PHY interface.
PROMIS_ENA	**Promiscuous Mode Enable** (RW). When set to “0” (Reset value), the MAC address programmed in registers MAC_0, MAC_1 and MAC_2 is used to discriminate Unicast frames received from the line. When set to “1”, all Unicast frames are accepted without MAC address check.
PAUSE_GEN	**Pause Frame Generate** (RW). When set to “1” by the host processor, the MAC generates, independently of the Receive FIFO status, a Pause frame with the Pause quanta programmed in register PAUSE_QUANT. When set to “0” (Reset value), Pause frames are only generated when the controller Receive FIFO reports a congestion.

RW: Read / Write register.

Section 15. Device Pinout

This section provides device pinout data, pinout symbol descriptions, and details of JTAG port pins.

Table 14: QLPG3116: 280-Pin FPBGA (PT280) Pinout Table

PIN	Function	PIN	Function	PIN	Function	PIN	Function	PIN	Function
A1	gnd	D1	pp_tx_data[14]	G19	gm_rx_data[3]	N16	gnd	U6	gnd
A2	gnd	D2	pp_tx_data[13]	H1	gnd	N17	gm_tx_err	U7	pp_rx_data[5]
A3	gnd	D3	pp_tx_data[12]	H2	gnd	N18	gnd	U8	pp_rx_data[6]
A4	gnd	D4	pp_tx_data[11]	H3	gnd	N19	gnd	U9	vccio
A5	gnd	D5	pp_tx_data[10]	H4	gnd	P1	pp_rx_sx	U10	pp_rx_clk
A6	gnd	D6	pp_tx_data[9]	H5	vcc	P2	pp_rx_val	U11	vccio
A7	gnd	D7	pp_tx_data[8]	H15	vcc	P3	gnd	U12	up_data[0]
A8	gnd	D8	pp_tx_data[7]	H16	vcc	P4	gnd	U13	up_data[1]
A9	gnd	D9	gnd	H17	gm_rx_data[6]	P5	vcc	U14	gnd
A10	gnd	D10	gnd	H18	gm_rx_data[5]	P15	gnd	U15	vccio
A11	reset	D11	gnd	H19	gm_rx_data[4]	P16	up_data[14]	U16	up_data[2]
A12	gnd	D12	gnd	J1	gnd	P17	up_data[15]	U17	tdo
A13	gnd	D13	gnd	J2	gnd	P18	gnd	U18	gnd
A14	gnd	D14	gnd	J3	vccio	P19	gnd	U19	up_data[3]
A15	gnd	D15	gnd	J4	gnd	R1	pp_rx_mod[0:0]	V1	gnd
A16	gnd	D16	gnd	J5	gnd	R2	pp_rx_err	V2	gnd
A17	gnd	D17	gnd	J15	vcc	R3	vccio	V3	gnd
A18	vcc	D18	gnd	J16	gm_rx_data[7]	R4	pp_rx_prty	V4	pp_rx_data[1]
A19	gnd	D19	gnd	J17	vccio	R5	gnd	V5	pp_rx_data[0]
B1	vcc	E1	pp_tx_data[15]	J18	gm_rx_en	R6	gnd	V6	gnd
B2	gnd	E2	pp_tx_mod[0:0]	J19	gm_rx_err	R7	vcc	V7	pp_rx_eop
B3	pp_tx_data[1]	E3	vccio	K1	vcc	R8	vcc	V8	pp_rx_sop
B4	pp_tx_data[0]	E4	pp_tx_err	K2	tck	R9	gnd	V9	pp_rx_enb
B5	pp_tx_eop	E5	gnd	K3	gnd	R10	gnd	V10	gnd
B6	gnd	E6	vcc	K4	gnd	R11	vcc	V11	gm_rx_clk
B7	pp_tx_sop	E7	vcc	K5	gnd	R12	vcc	V12	up_addr[8]
B8	pp_tx_enb	E8	vcc	K15	gnd	R13	vcc	V13	up_addr[7]
B9	tms	E9	vcc	K16	gm_tx_data[0]	R14	vcc	V14	gnd
B10	gnd	E10	gnd	K17	gm_tx_data[1]	R15	gnd	V15	up_addr[6]
B11	gnd	E11	gnd	K18	gm_tx_data[2]	R16	up_data[11]	V16	up_addr[5]
B12	gnd	E12	vcc	K19	trstb	R17	vccio	V17	up_addr[4]
B13	gnd	E13	vcc	L1	gnd	R18	up_data[12]	V18	gnd
B14	gnd	E14	gnd	L2	gnd	R19	up_data[13]	V19	gnd

Table 14: QLPG3116: 280-Pin FPBGA (PT280) Pinout Table

B15	gnd	**E15**	gnd	**L3**	vccio	**T1**	pp_rx_data[15]	**W1**	gnd
B16	gnd	**E16**	gnd	**L4**	gnd	**T2**	pp_rx_data[14]	**W2**	vcc
B17	vcc	**E17**	vccio	**L5**	vcc	**T3**	pp_rx_data[13]	**W3**	gnd
B18	gnd	**E18**	gnd	**L15**	gnd	**T4**	pp_rx_data[12]	**W4**	gnd
B19	gnd	**E19**	gnd	**L16**	gm_tx_data[3]	**T5**	pp_rx_data[11]	**W5**	gnd
C1	pp_tx_data[2]	**F1**	gnd	**L17**	vccio	**T6**	gnd	**W6**	gnd
C2	vcc	**F2**	gnd	**L18**	gm_tx_data[4]	**T7**	pp_rx_data[10]	**W7**	gnd
C3	pp_tx_data[3]	**F3**	pp_dtpa	**L19**	gm_tx_data[5]	**T8**	pp_rx_data[9]	**W8**	gnd
C4	pp_tx_data[4]	**F4**	pp_tx_prty	**M1**	gnd	**T9**	pp_rx_data[8]	**W9**	tdi
C5	vccio	**F5**	gnd	**M2**	gnd	**T10**	pp_rx_data[7]	**W10**	up_clk
C6	gnd	**F15**	vcc	**M3**	gnd	**T11**	gm_tx_clk	**W11**	up_csn
C7	pp_tx_data[5]	**F16**	gnd	**M4**	gnd	**T12**	up_data[10]	**W12**	up_rdwrn
C8	pp_tx_data[6]	**F17**	gnd	**M5**	vcc	**T13**	up_data[9]	**W13**	up_rdyn
C9	vccio	**F18**	gnd	**M15**	vcc	**T14**	up_data[8]	**W14**	gnd
C10	pp_tx_clk	**F19**	gnd	**M16**	gnd	**T15**	up_data[7]	**W15**	up_addr[9]
C11	vccio	**G1**	gnd	**M17**	gm_tx_data[6]	**T16**	up_data[6]	**W16**	up_addr[1]
C12	gnd	**G2**	gnd	**M18**	gm_tx_data[7]	**T17**	vcc	**W17**	up_addr[2]
C13	gnd	**G3**	gnd	**M19**	gm_tx_en	**T18**	up_data[5]	**W18**	up_addr[3]
C14	gnd	**G4**	pp_tx_sx	**N1**	gnd	**T19**	up_data[4]	**W19**	gnd
C15	vccio	**G5**	vcc	**N2**	gnd	**U1**	pp_rx_data[2]		
C16	gnd	**G15**	vcc	**N3**	gnd	**U2**	pp_rx_data[3]		
C17	gnd	**G16**	gm_rx_data[0]	**N4**	gnd	**U3**	vcc		
C18	gnd	**G17**	gm_rx_data[1]	**N5**	vcc	**U4**	pp_rx_data[4]		
C19	gnd	**G18**	gm_rx_data[2]	**N15**	vcc	**U5**	vccio		

Table 15: Pinout Symbol Description

Symbol	Description
gnd	Ground
vcc	2.5V Core Power Supply
vccio	2.5 or 3.3V I/O Power Supply

Table 16: JTAG Port

Pin	Description
tdi	JTAG Data In. Connected to VCC if not used
tdo	JTAG Data Out. Unconnected in not used
tck	JTAG Clock. Connected to GND if not used
tms	JTAG Test Mode Select. Connected to VCC if not used
trstb	JTAG Reset. Connected to GND if not used

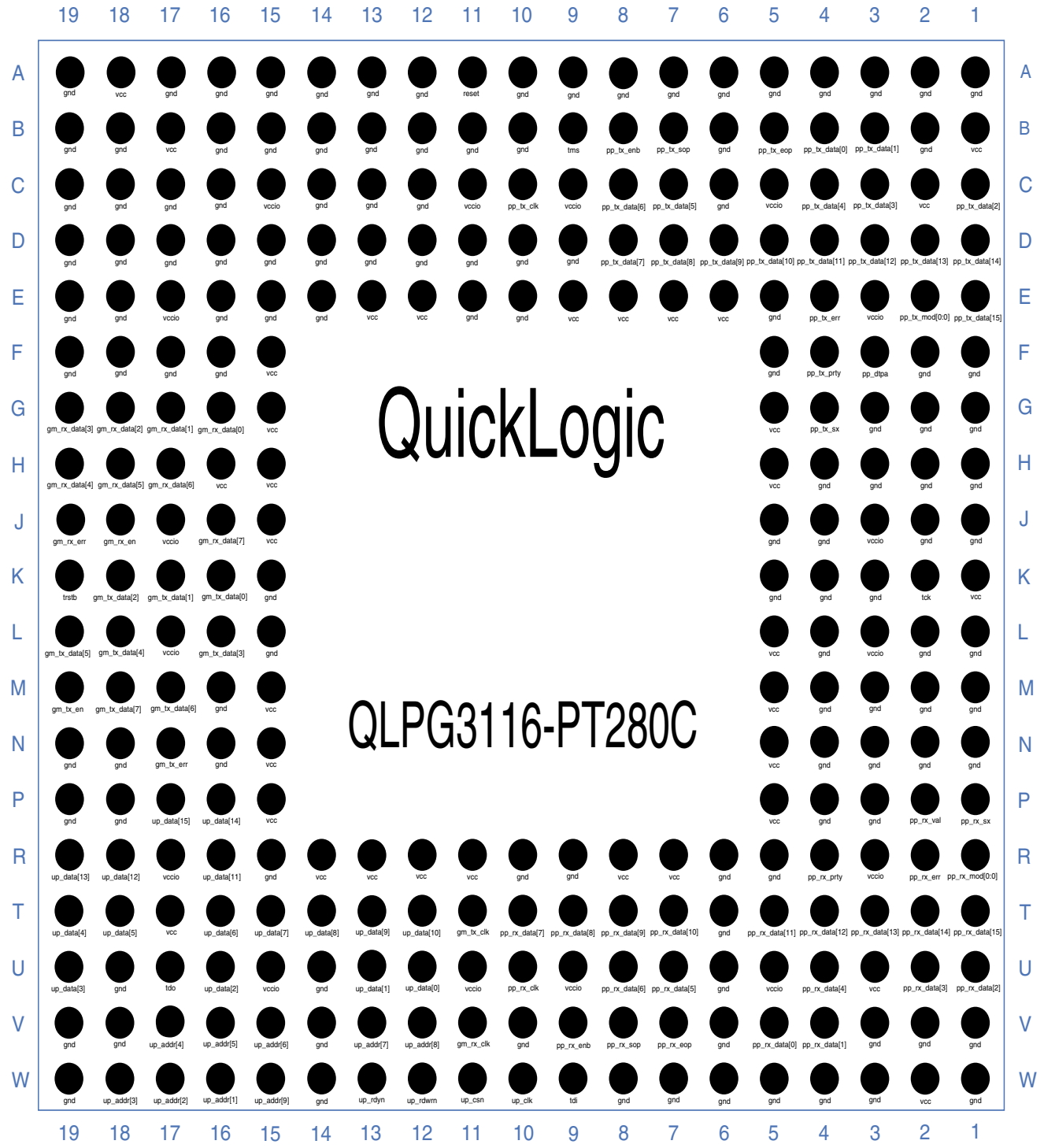


Figure 27: PT280 bottom view (0.8mm FPBGA)

Section 16. DC Characteristics

Table 17: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
V_{CC}	Power Supply	-0.5	+ 3.6	V
V_{CCIO}	I/O Voltage Setting	-0.5	4.6	V
T_S	Storage Temperature	-65	+ 150	°C
V_{IN}	Input Pin Voltage	-0.5	V_{CCIO} + 0.5	V
I_{CC}	DC Input Current	-20	+ 20	mA
-	ESD Protection	-2000	2000	V

Table 18: Operating Ranges

Symbol	Parameter	Min	Typical	Max	Unit
V_{CC}	Supply Voltage	2.3	2.5	2.7	V
V_{CCIO}	Input/Output Voltage Setting	2.3	-	3.6	V
T_A	Ambient Temperature (Commercial Grade)	0	25	70	°C

Table 19: DC Characteristics

Symbol	Parameter	Conditions	Min	Max	Unit
I_I	Input Leakage Current	$V_{CCIO} = V_{CC}$ or Gnd	-0.5	+ 2.7	ìA
I_{CC}	DC Supply Current		0.50	2	mA
C_I	Input Capacitance		-	8	pF
V_{IH}	Input High Voltage	$V_{CCIO} = 2.5$	2.0	2.8	V
	Input High Voltage	$V_{CCIO} = 3.3$	2.0	3.6	V
V_{IL}	Input Low Voltage	$V_{CCIO} = 2.5$ or 3.3	-0.3	0.8	V
V_{OH}	Output High Voltage	$I_{OH} = 2$ mA	2.4	V_{CCIO}	V
V_{OL}	Output Low Voltage	$I_{OL} = -2$ mA	0	0.4	V

Table 20: Thermal and Power Dissipation Characteristics

0_{ja} (*C/W vs. Airflow @ m/sec.)				0_{jc} (*C/W)	Estimated Maximum Power Dissipation (W)
0.0	0.5	1.0	2.0		
18.5	17.0	15.5	14.0	7.0	TBD

Section 17. AC Characteristics

This section provides timing characteristics for the QLPG3116.

17.1 GMII Interface

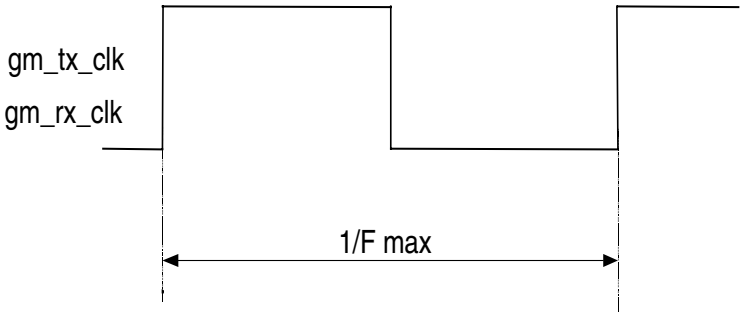


Figure 28: GMII Clock Frequency Definition

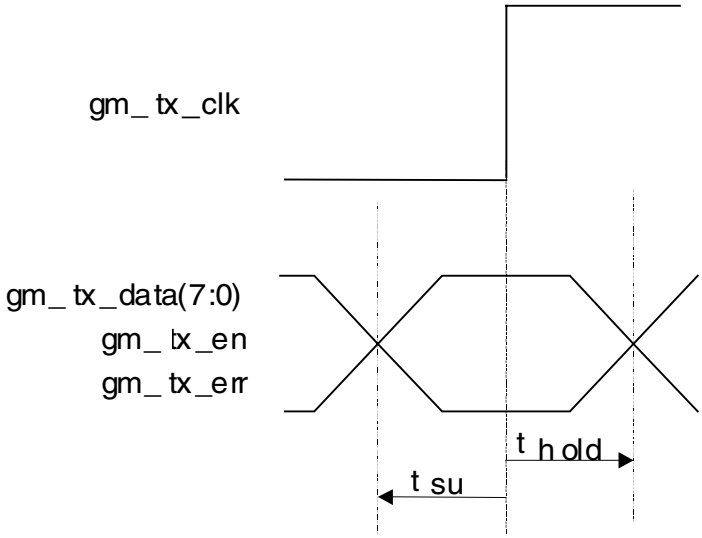


Figure 29: GMII Input Setup and Hold Definition

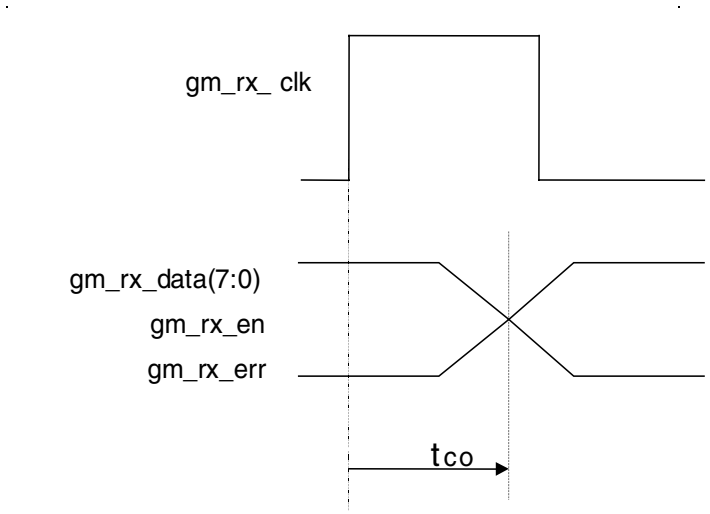


Figure 30: GMII Output Clock to Out Definition

Table 21: GMII I/O Timing Characteristics - PRELIMINARY

Symbol	Description	Value	
		Min	Max
Fmax	GMII Maximum Frequency	-	125MHz
tsu	GMII Inputs Setup Time	2.4ns	-
thold	GMII Inputs Hold Time	0ns	-
tco	GMII Outputs Clock to Out	-	4ns

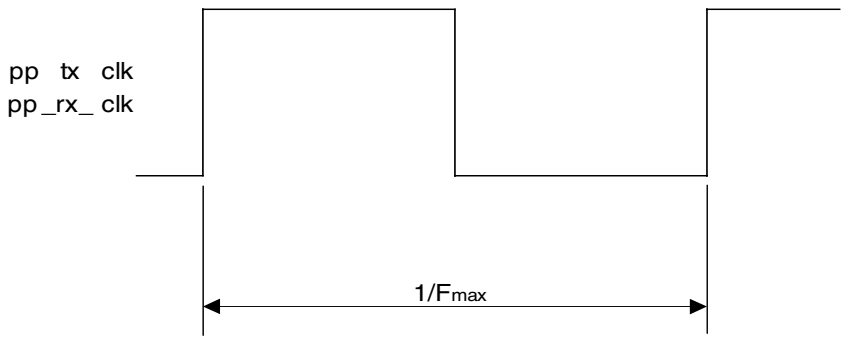


Figure 31: POS-PHY L3 Clock Frequency Definition

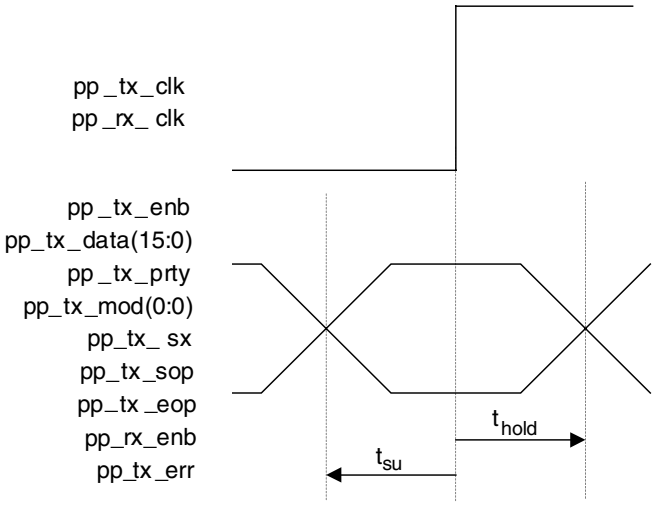


Figure 32: POS-PHY L3 Input Setup and Hold Definition

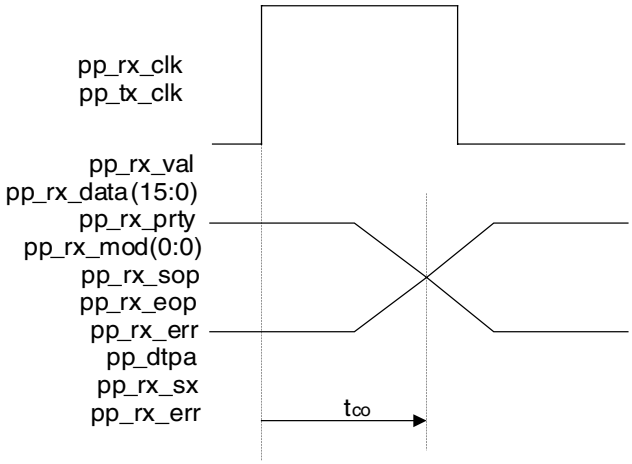
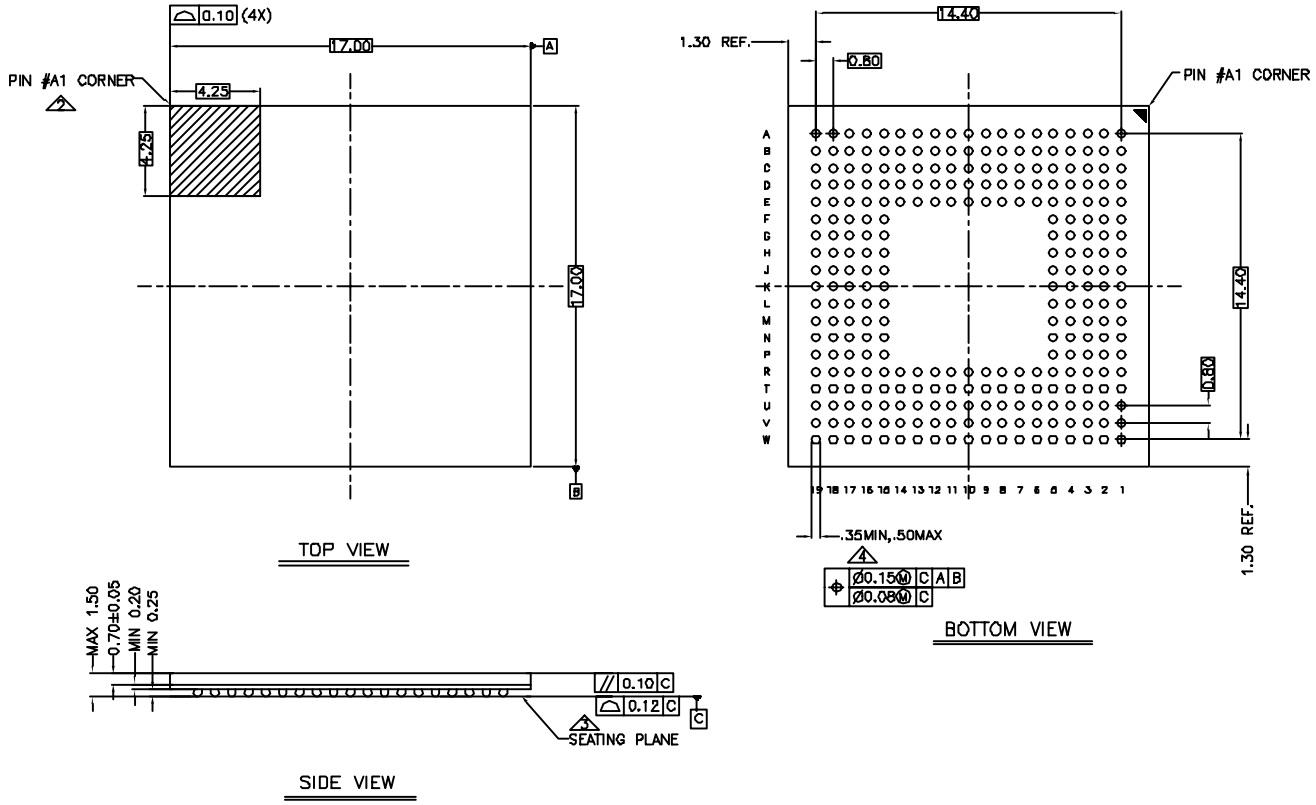


Figure 33: POS-PHY L3 Output Clock to Out Definition

Table 22: POS-PHY L3 I/O Timing Characteristics - PRELIMINARY

Symbol	Description	Value	
		Min	Max
Fmax	POS-PHY Maximum Frequency	-	104MHz
tsu	POS-PHY Inputs Setup Time	2.1ns	-
thold	POS-PHY Inputs Hold Time	0ns	-
tco	POS-PHY Output Clock to Out	-	5.9ns

Section 18. Mechanical Specifications



NOTES:

1. ALL DIMENSIONS AND TOLERANCE CONFORM TO ASME Y 14.5M–1994.
2. DETAILS OF PIN 1 IDENTIFIER ARE OPTIONAL, MAY CONSIST OF INK DOT, LASER MARK OR METALLIZED MARKINGS, MUST BE LOCATED WITHIN THE INDICATED ZONE.
3. PRIMARY DATUM C AND SEATING PLANE ARE DEFINED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
4. DIMENSION IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER, PARALLEL TO PRIMARY DATUM C.
5. REFER TO JEDEC STANDARD MO–205.
6. ALL DIMENSNIONS ARE IN MILLIMETERS.

Figure 34: Mechanical Specifications

Section 19. Ordering Information

QLPG3116-PT280C

Figure 35: Ordering Code