

Introduction

The Ethernet 1000BASE-X PCS/PMA or SGMII LogiCORE™ provides a flexible core for connection to an Ethernet Media Access Controller (MAC) or other custom logic and supports two standards of operation which can be dynamically selected:

- 1000BASE-X Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) operation, as defined in the *IEEE802.3* standard
- GMII to Serial-GMII (SGMII) bridge, as defined in the Serial-GMII specification (ENG-46158)

Features

- 1000BASE-X Physical Coding Sublayer (PCS) designed to *IEEE802.3* specification for the following:
 - 1000BASE-X Physical Medium Attachment (PMA) using an integrated RocketIO™
 - 1000BASE-X parallel Ten-Bit-Interface (TBI) for connection to external SERDES
- Configured and monitored through the serial MDIO Interface (MII Management), which can optionally be omitted from the core
- Supports 1000BASE-X Auto-Negotiation for information exchange with a link partner, which can optionally be omitted from the core
- Internal or external GMII to MAC or custom logic
- Alternative Serial-GMII (SGMII) functionality for connection to external PHYs
- Available under terms of the [SignOnce IP Site License](#)

LogiCORE Facts	
Core Specifics	
Supported Family ¹	Virtex™-4 ² , Virtex-II Pro Virtex-II, Spartan™-3 Spartan-3E
Speed Grade	<ul style="list-style-type: none"> • -4 in Virtex-II, Spartan-3, Spartan-3E • -5 in Virtex-II Pro • -10 in Virtex-4
Performance	1.25 Gbps
Core Resources	
Slices	166-597 ³
LUTs	204-659 ³
FFs	168-632 ³
DCM	0-3 ³
BUFG	2-4 ³
Rocket IO	0-1 ³
Core Highlights	
• Designed to IEEE802.3	Simulation Only Evaluation •
• Hardware Verified	Hardware Evaluation •
Provided with Core	
Documentation	Product Specification User Guide Getting Started Guide
Design File Formats	NGC Netlist HDL Example Design Demonstration Test Bench Scripts
Constraints File	User Constraints File (.ucf)
Example Designs	<ul style="list-style-type: none"> • 1000BASE-X PCS/PMA using RocketIO™ • 1000BASE-X PCS with Ten-Bit Interface • GMII to SGMII Bridge
	Demo Test Environment
Design Tool Requirements	
Supported HDL	VHDL and/or Verilog
Synthesis	XST 8.1i
Xilinx Tools	ISE 8.1i
Simulation tools	Mentor Modelsim® Cadence IUS ⁴

1. For supported family configurations. See [Table 9](#)
2. Virtex-4 FX solutions require the latest silicon stepping and are pending hardware verification.
3. The exact number depends on user configuration. See [Table 10](#).
4. Scripts provided for Mentor ModelSim and Cadence IUS only.

© 2006 Xilinx, Inc. All rights reserved. XILINX, the Xilinx logo, and other designated brands included herein are trademarks of Xilinx, Inc. All other trademarks are the property of their respective owners. Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

Applications

Typical applications for the Ethernet 1000BASE-X PCS/PMA or SGMII core include:

- Ethernet 1000BASE-X
- Serial-GMII (SGMII)

Ethernet 1000BASE-X

Figure 1 illustrates a typical application for the Ethernet 1000BASE-X PCS/PMA or SGMII core with the core operating to the 1000BASE-X standard using a RocketIO to provide the Physical Coding Sub-layer (PCS) and Physical Medium Attachment (PMA) sublayers for 1-Gigabit Ethernet.

- The PMA is connected to an external off the shelf GBIC or SFP optical transceiver to complete the Ethernet port.
- The GMII of the Ethernet 1000BASE-X PCS/PMA is connected to an embedded Ethernet Media Access Controller (MAC), for example, the Xilinx 1-Gigabit Ethernet MAC core.

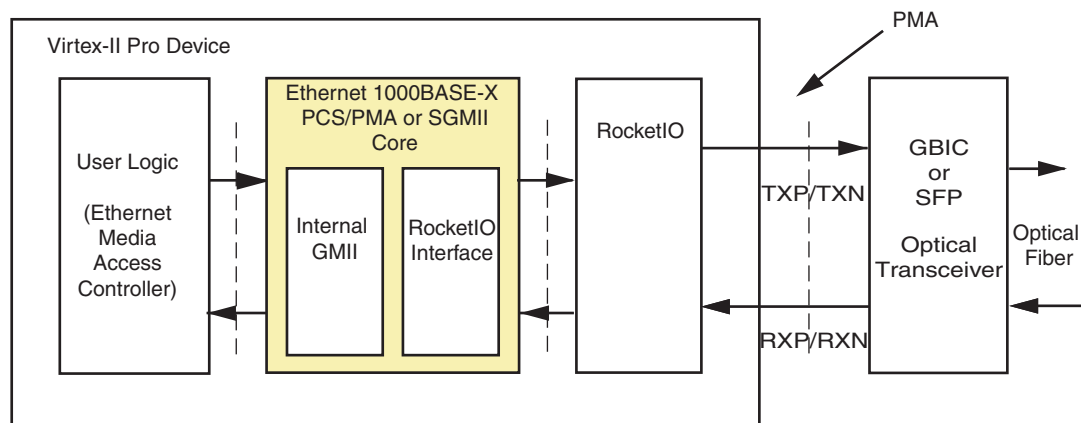


Figure 1: Typical 1000BASE-X Application

Serial-GMII (SGMII)

Figure 2 illustrates a typical application for the Ethernet 1000BASE-X PCS/PMA or SGMII core, which shows the core providing a GMII to SGMII bridge using a RocketIO to provide the serial interface.

- The RocketIO is connected to an external off the shelf Ethernet PHY device which also supports SGMII. This could be a tri-speed PHY providing 10BASE-T, 100BASE-T, and 1000BASE-T operation.
- The GMII of the Ethernet 1000BASE-X PCS/PMA or SGMII core is connected to an embedded Ethernet MAC (for example, the Tri-Mode Ethernet MAC core from Xilinx).

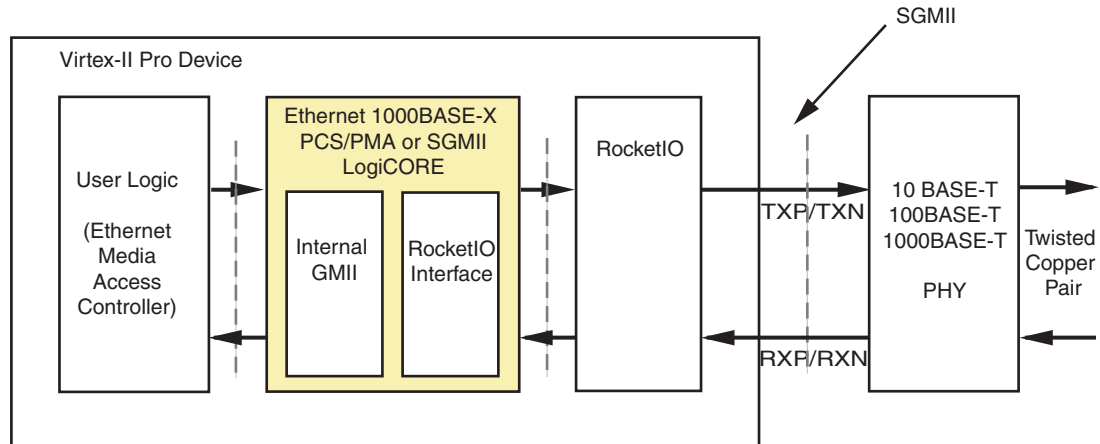


Figure 2: Typical SGMII Mode Application

Overview of Ethernet Architecture

The 1-Gigabit Ethernet PCS and PMA sublayers provided by this core are part of the Ethernet architecture illustrated in Figure 3. The part of this architecture, from the MAC to the right, is defined in the IEEE 802.3 specification. This figure also shows where the supported interfaces fit into the architecture.

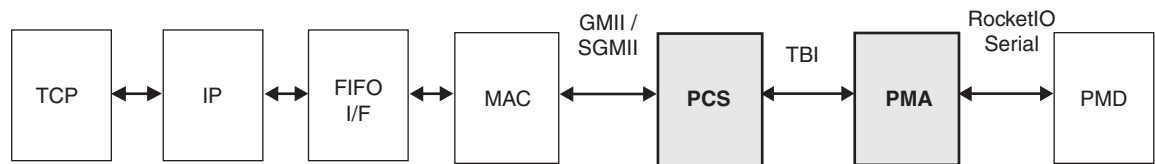


Figure 3: Overview of Ethernet Architecture

MAC

The Ethernet Media Access Controller (MAC) is defined in IEEE802.3, clauses 2, 3, and 4. A MAC is responsible for the Ethernet framing protocols and error detection of these frames. The MAC is independent of, and can connect to, any type of physical layer device.

GMII / SGMII

The Gigabit Media Independent Interface (GMII) is defined in IEEE802.3, clause 35. This is a parallel interface connecting a MAC to the physical sublayers (PCS, PMA, and PMD). For a MAC operating at a speed of 1 Gbps, the full GMII is used; for a MAC operating at a speed of 10 Mbps or 100 Mbps, the GMII is replaced with a Media Independent Interface (MII) that uses a subset of the GMII signals.

The Serial-GMII (SGMII) is an alternative interface to the GMII/MII that converts the parallel interface of the GMII/MII into a serial format capable of carrying traffic at speeds of 10 Mbps, 100 Mbps, and 1 Gbps. This radically reduces the I/O count and is therefore often favored by PCB designers. Note that the SGMII specification is closely related to the 1000BASE-X PCS and PMA sublayers, which enables it to be offered in this core.

PCS

The Physical Coding Sublayer (PCS) for 1000BASE-X operation is defined in IEEE802.3 clause 36 and 37 and performs the following:

- Encoding (and decoding) of GMII data octets to form a sequence of ordered sets
- 8B10B encoding (and decoding) of the sequence ordered sets
- 1000BASE-X Auto-Negotiation for information exchange with the link partner

Ten Bit Interface

The Ten-Bit-Interface (TBI) is defined in *IEEE802.3* clause 36. This is a parallel interface connecting the PCS to the PMA, and transfers the 8B10B encoded sequence-ordered sets.

Physical Medium Attachment

The Physical Medium Attachment (PMA) for 1000BASE-X operation is defined in IEEE802.3 clause 36 and performs the following:

- Serialization (and deserialization) of code-groups for transmission (and reception) on the underlying serial PMD
- Recovery of clock from the 8B/10B-coded data supplied by the PMD

The Virtex-II Pro or Virtex-4 RocketIO provides the serial interface required to connect the PMA to the PMD.

Physical Medium Dependent

The Physical Medium Dependent (PMD) sublayer is defined in *IEEE802.3* clause 38 for 1000BASE-LX and 1000BASE-SX (long and short wavelength laser). This type of PMD is provided by the external GBIC or SFP optical transceivers. An alternative PMD for 1000BASE-CX (short haul copper) is defined in IEEE802.3, clause 39.

Overview of the Ethernet 1000BASE-X PCS/PMA or SGMII Core

Using the Ethernet 1000BASE-X PCS/PMA or SGMII core with Virtex-II Pro or Virtex-4 RocketIO provides the functionality to implement the 1000BASE-X PCS and PMA sublayers. Alternatively, it may be used to provide a GMII to SGMII bridge.

The core interfaces directly to a RocketIO Multi-Gigabit Transceiver, providing some of the PCS layer functionality such as 8B/10B encoding/decoding, the PMA SERDES, and clock recovery. Figure 4 illustrates the remaining PCS sublayer functionality and the major functional blocks of the core. A description of the functional blocks and signals is provided in the sections that follow.

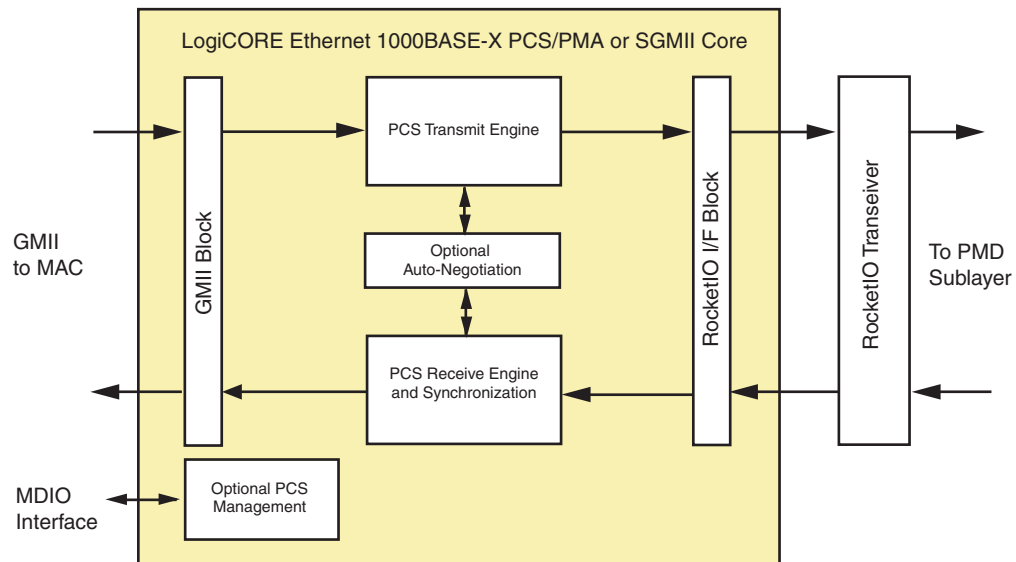


Figure 4: Functional Block Diagram of the Ethernet 1000BASE-X PCS/PMA or SGMII Core using RocketIO

GMII Block

The core provides a client-side GMII. This may be used as an internal interface for connection to an embedded MAC or other custom logic. Alternatively, the core GMII may be routed to device IOBs to provide an off-chip GMII.

PCS Transmit Engine

The PCS transmit engine converts the GMII data octets into a sequence of ordered sets by implementing the state diagrams of *IEEE802.3* (Figures 36-5 and 36-6).

PCS Receive Engine and Synchronization

The synchronization process implements the state diagram of *IEEE802.3* (Figure 36-9). The PCS receive engine converts the sequence of ordered sets to GMII data octets by implementing the state diagrams of *IEEE802.3* (Figures 36-7a and 36-7b).

Optional Auto-Negotiation Block

IEEE802.3 clause 37 describes the 1000BASE-X Auto-Negotiation function that allows a device to advertise the supported modes of operation to a device at the remote end of a link segment (link partner), and to detect corresponding operational modes that the link partner may be advertising.

Auto-Negotiation is controlled and monitored through the PCS Management Registers.

Optional PCS Management Registers

Configuration and status of the core, including access to and from the optional Auto-Negotiation function, is performed with the 1000BASE-X PCS Management Registers as defined in IEEE802.3 clause 37. These registers are accessed through the serial Management Data Input/Output Interface (MDIO), defined in IEEE802.3 clause 22, as if it were an externally connected PHY.

The PCS Management Registers may be omitted from the core when the core is performing the 1000BASE-X standard. In this situation, configuration and status is made possible by using an alternative configuration vector and a status signal.

When the core is performing the SGMII standard, PCS Management Registers become mandatory and information in the registers takes on a different interpretation. See the *LogiCORE Ethernet 1000BASE-X PCS/PMA or SGMII User Guide*.

RocketIO Interface Block

This enables the core to connect directly to a Virtex-II Pro RocketIO (using the GT_ETHERNET_1 primitive) or to a Virtex-4 RocketIO using a small logic shim (provided).

Ethernet 1000BASE-X PCS/PMA or SGMII core with Ten-Bit-Interface

When used with the TBI, the Ethernet 1000BASE-X PCS/PMA or SGMII core provides the functionality to implement the 1000BASE-X PCS sublayer. There is no support for the SGMII standard when using the core with TBI.

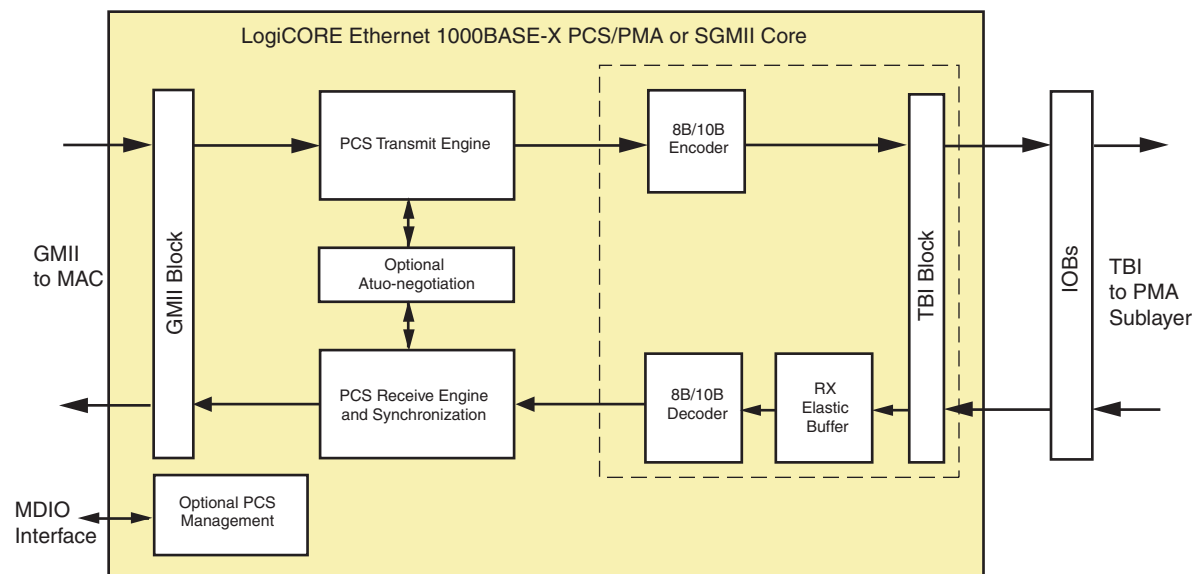


Figure 5: Functional Block Diagram of the Ethernet 1000BASE-X PCS/PMA or SGMII Core with TBI

The optional TBI is used in place of the RocketIO to provide a parallel interface for connection to an external PMA SERDES device, providing an alternative implementation for families without RocketIO. In this implementation, additional logic blocks are required in the core to replace some of the RocketIO functionality. These blocks are surrounded by a dashed line (see Figure 5). Other blocks are identical to those previously defined.

8B/10B Encoder

8B10B encoding, as defined in *IEEE802.3* (Tables 36-1a to 36-1e and Table 36-2), is implemented in a Block SelectRAM™, configured as ROM, and used as a large look-up table.

8B/10B Decoder

8B10B decoding, as defined in *IEEE802.3* (Tables 36-1a to 36-1e and Table 36-2), is implemented in a Block SelectRAM, configured as ROM, and used as a large look-up table.

Receiver Elastic Buffer

The Receiver Elastic Buffer enables the 10-bit parallel TBI data, received from the PMA sublayer synchronously to the TBI receiver clocks, to be transferred onto the cores internal 125MHz clock domain.

The Receiver Elastic Buffer is an asynchronous FIFO implemented in Distributed RAM. The operation of the Receiver Elastic Buffer is to attempt to maintain a constant occupancy by inserting or removing Idle sequences as necessary. This causes no corruption to the frames of data.

TBI Block

The core provides a TBI interface, which should be routed to device IOBs to provide an off-chip TBI. See the *Ethernet 1000BASE-X PCS/PMA or SGMII User Guide* for more information.

Interface Descriptions

All ports of the core are internal connections in FPGA fabric. An HDL example design, provided in both VHDL and Verilog, is delivered with the core. Where appropriate, the example design connects the core to a RocketIO, and/or add IBUFs, OBUFs, and IOB flip-flops to the external signals of the GMII and TBI. IOBs are added to the remaining unconnected ports to take the example design through the Xilinx implementation software. All clock management logic is placed in this example design allowing for more flexibility in implementation; for example, in designs using multiple cores. For more information about the example designs, see the *Ethernet 1000BASE-X PCS/PMA or SGMII Getting Started Guide*.

GMII Signal Definition

Table 1 defines the GMII-side interface signals common to all parameterizations of the core. These are typically attached to an Ethernet MAC, either off-chip or internally integrated. The HDL example design delivered with the core connects these signals to IOBs to provide a place-and-routable example.

Table 1: GMII Interface Signal Pinout

Signal	Direction	Clock Domain	Description
gmii_txd[7:0]	Input	see note 1	GMII Transmit data from MAC.
gmii_tx_en	Input	see note 1	GMII Transmit control signal from MAC.
gmii_tx_er	Input	see note 1	GMII Transmit control signal from MAC.
gmii_rxd[7:0]	Output	see note 1	GMII Received data to MAC.
gmii_rx_dv	Output	see note 1	GMII Received control signal to MAC.

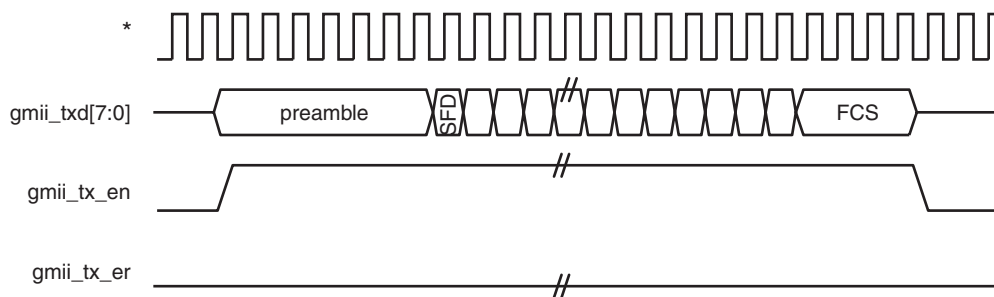
Table 1: GMII Interface Signal Pinout (Continued)

Signal	Direction	Clock Domain	Description
gmii_rx_er	Output	see note 1	GMII Received control signal to MAC.
gmii_isolate	Output	see note 1	IOB Tri-state control for GMII Isolation. Only of use when implementing an External GMII as illustrated by the example design HDL.
1. Signals are synchronous to the cores internal 125MHz reference clock; userclk2 when used with the RocketIO, and gtx_clk when used with TBI.			

GMII Usage Example

Standard Frame Transmission

Figure 6 illustrates the timing of normal outbound frame transfer. This shows that an Ethernet frame is preceded by an 8-byte preamble field and completed with a 4-byte Frame Check Sequence (FCS) field (*IEEE802.3*, clause 3). This is driven by the core transmitter client logic (usually a MAC connected to the other end of the GMII). The PCS treats any value placed on `gmii_txd[7:0]` within the `gmii_tx_en` assertion window as data.

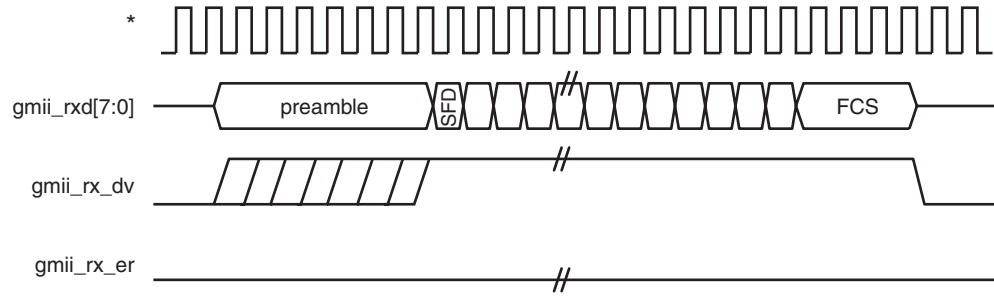


* See note 1 in Table 1

Figure 6: GMII Standard Frame Transmission

Standard Frame Reception

Figure 7 illustrates the timing of normal inbound frame transfer. This shows that Ethernet frame reception is preceded by a preamble field; the IEEE802.3 specification allows for up to all of the seven preamble bytes that proceed the Start of Frame Delimiter (SFD) to be lost (IEEE802.3 clause 35). The SFD will always be present in well-formed frames. This frame is presented by the core to the receiver client logic (usually a MAC connected to the other end of the GMII).



* See note 1 in Table 1

Figure 7: GMII Standard Frame Reception

Common Signal Definition

Table 2 defines the signals common to all parameterizations of the core.

Table 2: Other Common Signals

Signal	Direction	Clock Domain	Description
reset	Input	n/a	Asynchronous reset for the entire core. Active High.
signal_detect	Input	n/a	Signal direct from PMD sublayer indicating the presence of light detected at the optical receiver. If set to 1, this indicates that the optical receiver has detected light. If set to 0 this indicates the absence of light. If unused this signal should be set to 1 to enable correct operation the core.

Optional Management I/F Signal Definition

Table 3 describes the optional MDIO interface signals of the core used to access the PCS Management Registers. These signals are typically connected to the MDIO port of a MAC device, either off-chip or to an internally integrated MAC core.

Table 3: Optional MDIO Interface Signal Pinout

Signal	Direction	Clock Domain	Description
mdc	Input	n/a	Management clock (<= 2.5MHz)
mdio_in	Input	mdc	Input data signal

Table 3: Optional MDIO Interface Signal Pinout (Continued)

Signal	Direction	Clock Domain	Description
mdio_out	Output	mdc	Output data signal
mdio_tri	Output	mdc	Output tri-state driver for mdio_out. Active Low
phyad[4:0]	Input	n/a	Physical Address of the PCS Management register set. It is expected that this signal will be tied off to a logical value.

Alternative to the Management I/F: Configuration Vector and Status Definition

Table 4 describes the alternative to the optional MDIO—the optional configuration and status vectors.

Table 4: Optional Configuration and Status Vectors

Signal	Direction	Clock Domain	Description
configuration_vector[3:0]	Input	see note 1	<p>Bit[0]: Reserved (currently unused)</p> <p>Bit[1]: Loopback Control</p> <p>When the core with Virtex-II Pro RocketIO is used, this places the core into internal loopback mode. With the TBI version, Bit 1 is connected to <i>ewrap</i>. When set to 1 this indicates to the external PMA module to enter loopback mode.</p> <p>Bit[2]: Power Down</p> <p>When the Virtex-II Pro RocketIO is used, when set to 1 the RocketIO is placed in a low power state. A reset must be applied to clear. With the TBI version this bit is unused.</p> <p>Bit[3]: Isolate</p> <p>When set to 1 the GMII should be electrically isolated. When set to 0 normal operation is enabled.</p>
sync_acquired_status	Output	see note 1	<p>The output from the receiver's synchronization state machine of IEEE802.3, clause 36. When driven high, synchronization on the received bitstream has been obtained (The state machine is in one of the SYNC_AQUIRED states of IEEE802.3 figure 36-9). When driven low, synchronisation has not yet been obtained.</p>
<p>1. Signals are synchronous to the core's internal 125 MHz reference clock; userclk2 when used with RocketIO; gtx_clk when used with TBI.</p>			

Optional 1000BASE-X PCS/PMA (or SGMII) using RocketIO Signal Definition

Table 5 defines the optional interface to the RocketIO. The core is connected to a RocketIO in the HDL example design delivered with the core. For a complete description of the RocketIO interface, see the *Xilinx RocketIO User Guide*.

Table 5: Optional RocketIO Interface Pinout

Signal	Direction	Clock Domain	Description
mgt_rx_reset	Output	userclk2	Reset signal issued by the core to the RocketIO receiver path. Connect to RXRESET signal of RocketIO.
mgt_tx_reset	Output	userclk2	Reset signal issued by the core to the RocketIO transmitter path. Connect to TXRESET signal of RocketIO.
userclk	Input	n/a	Also connected to TXUSRCLK and RXUSRCLK of the RocketIO.
userclk2	Input	n/a	Also connected to TXUSRCLK2 and RXUSRCLK2 of the RocketIO ¹
dcm_locked	Input	n/a	A DCM may be used to derive userclk and userclk2. This is implemented in the HDL design example delivered with the core. The core will use this input to hold the RocketIO in reset until the DCM obtains lock.
rxbufstatus[1:0]	Input	userclk2	Connect to RocketIO signal of the same name
rxchariscomma	Input	userclk2	Connect to RocketIO signal of the same name
rxcharisk	Input	userclk2	Connect to RocketIO signal of the same name
rxclkcorcnt[2:0]	Input	userclk2	Connect to RocketIO signal of the same name
rxdata[7:0]	Input	userclk2	Connect to RocketIO signal of the same name
rxdisperr	Input	userclk2	Connect to RocketIO signal of the same name
rxnotintable	Input	userclk2	Connect to RocketIO signal of the same name
rxrundisp	Input	userclk2	Connect to RocketIO signal of the same name
txbuferr	Input	userclk2	Connect to RocketIO signal of the same name
powerdown	Output	userclk2	Connect to RocketIO signal of the same name
txchardispmode	Output	userclk2	Connect to RocketIO signal of the same name
txchardispval	Output	userclk2	Connect to RocketIO signal of the same name
txcharisk	Output	userclk2	Connect to RocketIO signal of the same name
txdata[7:0]	Output	userclk2	Connect to RocketIO signal of the same name
enablealign	Output	userclk2	Allow the transceivers to serially realign to a comma character. Connect to ENMCOMMAALIGN and ENPCOMMAALIGN of the RocketIO.

1. When the core is used with the RocketIO, userclk2 is used as the 125 MHz reference clock for the entire core.

Optional 1000BASE-X PCS with TBI Signal Definition

Table 6 defines the optional TBI signals that can be used as an alternative to the RocketIO interface. The appropriate HDL example design delivered with the core connects these signals to IOBs to provide an external TBI suitable for connection to an off-chip PMA SERDES device.

Table 6: Optional TBI Interface Signal Pinout

Signal	Direction	Clock Domain	Description
gtx_clk	Input	n/a	Clock signal at 125 MHz. Tolerance must be within IEEE802.3 specification. ¹
tx_code_group[9:0]	Output	gtx_clk	10-bit parallel transmit data to PMA Sublayer (SERDES).
loc_ref	Output	n/a	Causes the PMA sublayer clock recovery unit to lock to pma_tx_clk. This signal is currently tied to Ground.
ewrap	Output	gtx_clk	When 1 this indicates to the external PMA SERDES device to enter loopback mode. When 0, this indicates normal operation.
rx_code_group0[9:0]	Input	pma_rx_clk0	10-bit parallel received data from PMA Sublayer (SERDES). This is synchronous to pma_rx_clk0.
rx_code_group1[9:0]	Input	pma_rx_clk1	10-bit parallel received data from PMA Sublayer (SERDES). This is synchronous to pma_rx_clk1.
pma_rx_clk0	Input	n/a	Received clock signal from PMA Sublayer (SERDES) at 62.5MHz.
pma_rx_clk1	Input	n/a	Received clock signal from PMA Sublayer (SERDES) at 62.5MHz. This is 180 degrees out of phase with pma_rx_clk0.
en_cdet	Output	gtx_clk	Enables the PMA Sublayer to perform comma realignment. This is driven from the PCS Receive Engine during the <i>Loss-Of-Sync</i> state.

1. When the core is used with the TBI, gtx_clk is used as the 125 MHz reference clock for the entire core.

Optional Auto-Negotiation Signal Definition

Table 7 defines the signals whenever the optional Auto-Negotiation is present.

Table 7: Optional Auto-Negotiation Interface Signal Pinout

Signal	Direction	Clock Domain	Description
link_timer_value[8:0]	Input	see note 1	Used to configure the duration of the Auto-Negotiation function's Link Timer. The duration of this timer is set to the binary number input into this port multiplied by 4096 clock periods of the 125MHz reference clock (8 ns). It is expected that this signal will be tied off to a logical value. This port is replaced when using the dynamic switching mode.
an_interrupt	Output	see note 1	Active high interrupt to signal the completion of an Auto-Negotiation cycle. This interrupt can be enabled/disabled and cleared by writing to the appropriate PCS Management Register. For more information, see the <i>Ethernet 1000BASE-X PCS/PMA or SGMII User Guide</i> .
1. Signals are synchronous to the core's internal 125 MHz reference clock; userclk2 when the core is used with the RocketIO, and gtx_clk when the core is used with TBI.			

Optional Dynamic Switching Signal Pinout

Table 8 describes the additional signals present whenever the core is generated with the optional Dynamic Switching capability between 1000BASE-X and SGMII standards.

Table 8: Optional Dynamic Standard Switching Signals

Signal	Direction	Clock Domain	Description
link_timer_base[8:0]	Input	userclk2	Used to configure the duration of the Auto-Negotiation Link Timer period when performing the 1000BASE-X standard. The duration of this timer is set to the binary number input into this port multiplied by 4096 clock periods of the 125 MHz reference clock (8 ns). It is expected that this signal will be tied off to a logical value.

Table 8: Optional Dynamic Standard Switching Signals (Continued)

Signal	Direction	Clock Domain	Description
link_timer_sgmiic[8:0]	Input	userclk2	Used to configure the duration of the Auto-Negotiation Link Timer period when performing the SGMII standard. The duration of this timer is set to the binary number input into this port multiplied by 4096 clock periods of the 125 MHz reference clock (8 ns). It is expected that this signal will be tied off to a logical value.
basex_or_sgmiic	Input	userclk2	Used as the reset default to select the standard. It is expected that this signal will be tied off to a logical value: '0' signals that the core will come out of reset operating as 1000BASE-X; '1' signals that the core will come out of reset operating as SGMII. Note: The standard can be set following reset using the MDIO Management.

Core Latency

The standalone core does not meet all of the latency requirements specified in IEEE 802.3 due to the latency of the Elastic Buffers in both TBI and RocketIO versions. However, the core may be used for backplane and other applications where strict adherence to the IEEE latency specification is not a requirement.

Where strict adherence to the IEEE 802.3 specification is required, the core may be used with an Ethernet MAC core which is within the IEEE specified latency for a MAC sublayer. For example, when the core is connected to the Xilinx 1 Gigabit Ethernet MAC core, the system as a whole is compliant with the overall IEEE802.3 latency specifications.

For more information about latency, see the *Ethernet 1000BASE-X PCS/PMA or SGMII User Guide*.

Verification

The Ethernet 1000BASE-X PCS/PMA or SGMII core has been verified with extensive simulation and hardware verification.

Simulation

A highly parameterizable transaction-based test bench was used to test the core. The tests included the following.

- Register Access
- Loss of Synchronization
- Auto-Negotiation and error handling
- Frame Transmission and error handling
- Frame Reception and error handling
- Clock Compensation in the Elastic Buffers

Hardware Verification

The core has been tested in a variety of hardware test platforms at Xilinx to represent a variety of parameterizations, including the following:

- The core used with RocketIO and performing the 1000BASE-X standard has been tested with the Xilinx 1 Gigabit Ethernet MAC core, which follows the architecture shown in [Figure 1 on page 2](#). A test platform was built around these cores, including a back-end FIFO capable of performing a simple ping function, and a test pattern generator. Software running on the embedded PowerPC™ provided access to all configuration and status registers. Version 3.0 of this core was taken to the University of New Hampshire Interoperability Lab (UNH IOL) where conformance and interoperability testing was performed.
- The core used with RocketIO and performing the SGMII standard has been tested with the Tri-speed Ethernet MAC core from Xilinx. This was connected to an external PHY capable of performing 10BASE-T, 100BASE-T and 1000BASE-T, and the system was tested at all three speeds. This follows the architecture shown in [Figure 2 on page 3](#) and also included the PowerPC-based test platform described above.

Family Support

Table 9: Family Support for the 1000BASE-X PCS/PMA or SGMII Core

Family	LogiCORE Functionality			
	1000BASE-X		GMII to SGMII Bridge	1000BASE-X and SGMII Standards with Dynamic Switching
	With TBI	Using RocketIO		
Virtex-4	Supported	Supported	Supported	Supported
Virtex-II Pro	Supported	Supported	Supported	Supported
Virtex-II	Supported	Not supported	Not supported	Not supported
Spartan-3	Supported	Not supported	Not supported	Not supported
Spartan-3E	Supported	Not supported	Not supported	Not supported

Device Utilization

Table 10 provides approximate utilization figures for various core options when instantiated in a Virtex-II Pro device. Other families have similar utilization figures, except where stated. The shaded row indicates the option available for the core when performing the SGMII standard. All options are available for the core when performing the 1000BASE-X standard.

Table 10: Device Utilization

Parameter Values				Device Resources					
Physical Interface		MDIO Interface	Auto-Negotiation	Slices	LUTs	FFs	BRAMs	BUFGs	DCMs
Rocket IO	TBI								
Yes	No	Yes	Yes	451	527	467	0	2	1
Yes	No	Yes	No	255	314	251	0	2	1
Yes	No	No	N/A ¹	166	204	168	0	2	1
No	Yes	Yes	Yes	545	609	578	2	3	0 ²
No	Yes	Yes	No	350	394	369	2	3	0 ²
No	Yes	No	N/A ¹	264	285	282	2	3	0 ²

1. Auto-negotiation is only available when the MDIO Interface is selected.
2. Spartan-3 devices require two DCMs to meet TBI setup and hold times.

References

- [1] Virtex-4, Virtex-II, Virtex-II Pro User Guides
- [2] Spartan-3, Spartan-3E Data Sheets
- [2] RocketIO User Guides
- [3] IEEE802.3-2002
- [4] Serial-GMII Specification Revision 1.7

Support

For technical support, visit <http://www.xilinx.com/support>. Xilinx provides technical support for this LogiCORE product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not listed in the documentation, if customized beyond that allowed in the product documentation, or if any changes are made in sections of design marked *DO NOT MODIFY*.

Ordering Information

This Xilinx LogiCORE module is provided under the [SignOnce IP Site License](#). Two license options are available, both free of charge. The Evaluation license lets you assess the core functionality alongside your own design and demonstrates the various interfaces to the core in simulation, and the Full license provides full access to all core functionality both in simulation and in hardware. The core may be downloaded from the Xilinx [IP Center](#) for use with the Xilinx CORE Generator™. The Xilinx CORE Generator is bundled with ISE Foundation software v8.1i at no additional charge.

Please contact your local Xilinx [sales representative](#) or visit the Xilinx [Silicon Online Store](#) for pricing and availability on Xilinx LogiCORE modules and software. Information about additional Xilinx LogiCORE modules is available at the [Xilinx IP Center](#).

Revision History

The table below shows the version number of the document and includes the history of changes made to this document since its initial release.

Date	Version	Revision
9/24/04	5.0	Initial Xilinx release.
10/11/04	5.1	Document updated with corrections to Table 4 (configuration_vector[3:0] definition).
4/28/05	5.2	Updated core to v6.0, Xilinx tools v7.1i SP2, and ISE Foundation software v7.1i.
1/11/06	5.3	Updated core to v7.0, Xilinx tools v8.1i