# HB0627
# Handbook
# CoreSGMII v3.3

**Microsemi 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
Email: sales.support@microsemi.com
www.microsemi.com





## About Microsemi

Microsemi, a wholly owned subsidiary of Microchip Technology Inc. (Nasdaq: MCHP), offers a comprehensive portfolio of semiconductor and system solutions for aerospace & defense, communications, data center 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; enterprise storage and communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Learn more at www.microsemi.com.

# Contents

# Figures

# Tables

# 1 Revision History

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

## 1.1 Revision 5.0

Added PolarFire® SoC support.

## 1.2 Revision 4.0

Updated changes related to CoreSGMII v3.2.

## 1.3 Revision 3.0

Updated changes related to CoreSGMII v3.1.

## 1.4 Revision 2.0

Updated changes related to CoreSGMII v3.0.

## 1.5 Revision 1.0

The first publication of this document.

# 2 Preface

## 2.1 About this Document

This handbook provides details about the CoreSGMII DirectCore module, and how to use it.

## 2.2 Intended Audience

FPGA designers using Libero® System-on-Chip (SoC).

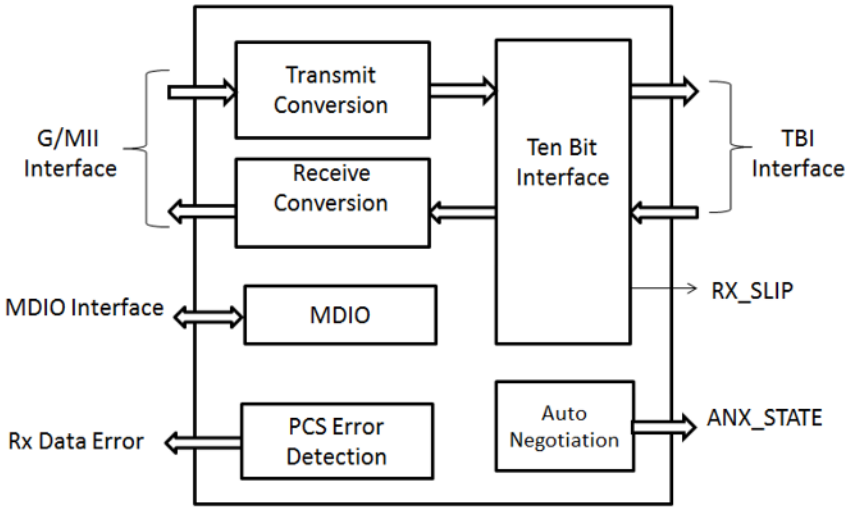
# 3 Introduction

## 3.1 Overview

The CoreSGMII provides a solution for ten bit interface (TBI) on gigabit media independent interface (G/MII) based designs. The CoreSGMII takes the G/MII data stream and encodes it into 10-bit symbols. In the Receive direction, the 10-bit symbols are decoded and converted into the receive G/MII signal set. The CoreSGMII is managed and monitored through the management data input/output (MDIO) interface.

The CoreSGMII supports Auto-Negotiation, which allows two link partners to exchange details of capabilities and determine the appropriate link operation.

***Figure 1 •*** **Top-Functional Block Diagram**




## 3.2 Features

CoreSGMII supports the following features:

- Full-duplex support for 1000 Mbps operation
- Full and half-duplex support for 10/100 Mbps operation
- G/MII for interfacing to a MAC
- MDIO interface to configure and monitor
- Implements 8b/10b encoding and decoding
- Clause 37 Auto-Negotiation
- Ten Bit Interface
- Comma alignment

## 3.3 Core Version

This handbook is for CoreSGMII version 3.3.

## 3.4 Supported Families

- PolarFire® SoC
- PolarFire®
- SmartFusion®2
- IGLOO®2
- RTG4™

## 3.5 Device Utilization and Performance

A summary of the utilization data of CoreSGMII is listed in Table 1.

Speed Grade - STD, Core Voltage - 1.2V and Operating Condition- IND.

***Table 1 •*** **CoreSGMII Device Utilization and Performance**

| | Device | FPGA Resources | | | | |
|---|---|---|---|---|---|---|
| **FPGA Family** | | **Combinatorial** | **Sequential** | **Total** | **Utilization** | **Clock Rate (MHz)** |
| PolarFire | PA5M300 | 2,112 | 1,291 | 3,403 | ~1.2% | TBI_TX_CLK > 125<br>RXCLK > 125<br>TXCLK > 125 |
| IGLOO2 | M2GL050T | 2,236 | 1,294 | 3,530 | ~ 6% | TBI_TX_CLK > 125<br>RXCLK > 125<br>TXCLK > 125 |
| SmartFusion2 | M2S050T | 2,171 | 1,294 | 3,465 | ~ 6% | TBI_TX_CLK > 125<br>RXCLK > 125<br>TXCLK > 125 |

**Note:** FPGA resources and performance data for the PolarFire SoC family is similar to PolarFire family.

**Note:** Data in this table are achieved using synthesis and layout settings optimized for speed along with interfacing to Transceiver.

# 4 Functional Description

The CoreSGMII IP has the following interfaces:

## 4.1 G/MII Interface

Gigabit media-independent Interface (G/MII) is an interface between the media access control (MAC) device and physical layer (PHY). It defines speeds up to 1000 Mbps, implemented using an 8-bit data interface clocked at 125 MHz, and is backwards compatible with the media-independent interface (MII) specification. It can also operate at fallback speeds of 10 or 100 Mbps as per the MII specification.

Data on the interface is framed using the IEEE Ethernet standard. It consists of the following:

- Preamble
- Start frame delimiter
- Ethernet headers
- Protocol specific data
- Cyclic redundancy check (CRC)

In case of G/MII transmission, there are two clocks, depending on whether the PHY is operating at 1000 Mbps or 10/100 Mbps speeds. TBI_TX_CLK is supplied to the PHY for 1000 Mbps speed, and the transmit data and control signals are synchronized to this. Otherwise, for 10/100 Mbps, the TXCLK supplied by the PHY is used for synchronizing those signals. This operates at either 25 MHz for 100 Mbps, or 2.5 MHz for 10 Mbps connections. The RXCLK is 2.5/25/125 MHz for 10/100/1000 respectively, is supplied from PHY/XCVR.

## 4.2 MDIO Interface

The CoreSGMII registers are accessed through the MDIO interface. The MDIO controller in the MAC can read and write the control and status registers of the CoreSGMII.

## 4.3 Serial G/MII Ten Bit Interface

The TBI provides independent Ten bit interface functionality contained in 802.3z Clauses 36 and 37. TBI can interface directly to the encoded physical coding sub layer (EPCS) interface of the Serializer/Deserializer (XCVR).

## 4.4 Clocking and Reset

- Master Reset (RESET) is external reset used for the complete core.
- PHY Reset (0x00 Control) is a bit from the MII Management Control register. This reset is synchronized to the native clocks of the TEX, REX, and ANX sub modules and reset these modules.
- TXCLK is used for converting G/MII data into TBI_TX_CLK domain.
- RXCLK is used for driving 8b data on to G/MII interface (8b data is received from TBI's TBI_RX_CLK domain).
- TBI_TX_CLK is used for Ten bit interface module.
- TBI_RX_CLK is used for RX conversion.

## 4.5 Transmit Conversion

This module synchronizes G/MII transmit path data to the TBI_TX_CLK clock domain. This module repeats each data byte 10/100 times for 10/100 Mbps respectively.

## 4.6 Receive Conversion

This module stores the G/MII transmit data from the Ten bit interface module to drive out to the G/MII RXCLK clock domain. This module only stores data every 10/100 clocks for 10/100 Mbps respectively, and artificially creates preamble and SFD.

## 4.7 Ten Bit Interface

### 4.7.1 TEX (Transmit Exchange functionality)

- This module performs clause 36 transmit related functionality of 802.3z. TEX operation is governed by auto negotiation, which provides CFG/IDL/DAT information.
- In CFG mode, TEX sends /C/ ordered sets with data from ANX.
- In IDL mode, TEX sends /I/ order sets.
- In DAT mode, TEX send 8b10b encoded packets.

### 4.7.2 REX (Receive Exchange functionality)

- Performs comma alignment and passes aligned two-code-group wide data to the PCS.
- Performs clause 36 receive related functionality of 802.3z.
- The code-groups from PCS are decoded for 10b8b and inspected by the receive logic.
- The PCS module looks for Configuration ordered sets and passes the Receive Configuration Register contents to the ANX module.
- After Auto-Negotiation completes, removes the encapsulation codes and passes the received packet.

### 4.7.3 TBM (Ten-bit interface management)

- This module provides the control and status path to an external processor.
- All the outputs from TBM module to ANX module are synchronous to MDC, which runs at a lower frequency and then, they are double registered synchronous to the TBI_TX_CLK, which runs at 125 MHz.

### 4.7.4 RX_SLIP

- The RX_SLIP from CoreSGMII and the RX_VALID from XCVR acts as a request/acknowledge handshake during slip functionality during SGMII comma alignment.
- The handshake works as follows. The RX_SLIP request will not be asserted until RX_VALID from XCVR is high. The XCVR will respond by lowering the RX_VALID. The SGMII shall then lower the RX_SLIP and wait until RX_VALID rises. RX_SLIP request pulses will be asserted until SGMII attains comma.

## 4.8 Auto Negotiation (ANX)

- The ANX module is responsible for the Auto-Negotiation function specified in Clause 37 of IEEE 802.3z. It provides an implementation for link partners to exchange ability information and link operation characteristics. These characteristics include duplex mode and remote fault conditions.
- The auto - negotiation function is managed via the MDIO interface. Software programs the ability information via the AN Advertisement register and status information about the Auto-Negotiation process can be gathered by reading several registers.
- For instance, the software can detect that ability information has been received from bit 1 of the AN Expansion register - Page Received.
- The Auto negotiation has four key functions, ability_match, acknowledge_match, consistency_match, and Idle_match.
- The Auto-Negotiation function has a mechanism to acknowledge reception of ability information and to send additional information via Next Pages. After both link partners properly exchange ability information and acknowledge reception, normal packet transmission may occur. Either end of the link may choose to re-negotiate at will and do so by sending a 'Break Link', which comprises of Configuration ordered_sets with a zeroed ability field.
- Currently AN Status information about the Auto-Negotiation process can be gathered by reading several registers. For instance, the software can detect that ability information has been received from bit 1 of the AN Expansion register - Page Received. Status register (0x01) has one bit filed, which indicates the successful completion auto negotiation.

## 4.9 PCS Error Detection

The CoreSGMII reports the running disparity error (RDERR) and code error (BCERR) of the RCG data stream. The RCG consists of a 6-bit block and 4-bit block. The running disparity rules are applicable to the sub-block boundaries. The running disparity at the end of the sub-block must be -1 or +1. Failure to meet these criteria is also flagged as a disparity error. The error code indicates that the RCG is not a valid member of the code group.

## 4.10 Register Map

The following registers are accessed through the MDIO interface clause 22 of the IEEE 802.3 specification. The PHY address for the MDIO registers can be configured.

The CoreSGMII contains the management registers specified in IEEE 802.3, Clause 37 - Control, Status, Auto Negotiation Advertisement, Link Partner Ability, Auto Negotiation Expansion and Extended Status. The register set is read/write through MDIO interface.

***Table 2 •*** **CoreSGMII Register Map**

| ADR | Register |
|---|---|
| 00h | Control |
| 01h | Status |
| 04h | AN Advertisement |
| 05h | AN Link Partner Base Page Ability |
| 06h | AN Expansion |
| 07h | AN Next Page Transmit |
| 08h | AN Link Partner Ability Next Page |
| 0Fh | Extended Status |
| 10h | Jitter Diagnostics |
| 11h | TBI Control |

T

***Table 3 •*** **Register Descriptions**

| Address | Function |
|---|---|
| 0x00 | **Control**<br>**[15] (R/W, SC) PHY RESET: Default 0**<br>Setting this bit causes the TEX, REX, and ANX sub-modules in the CoreSGMII core to be reset.<br>This bit is self-clearing.<br>**[14] (R/W) LOOP BACK: Default 0**<br>Setting this bit causes the transmit output of the CoreSGMII to be connected to the receive inputs. Clearing this bit result in normal operation.<br>**[13] Reserved**.<br>**[12] (R/W) AUTO-NEGOTIATION ENABLE Default 0**<br>Setting this bit enables the Auto-negotiation process.<br>**[11:10] Reserved**.<br>**[9] (R/W, SC) RESTART AUTO-NEGOTIATION: Default 0**<br>Setting this bit causes the Auto-negotiation process to restart. This action is only available when Auto-Negotiation has been enabled.<br>**[8:0] Reserved**. |

*Table 3 •* **Register Descriptions *(continued)***

| Address | Function |
|---|---|
| 0x01 | **Status - Default Value 0x149**<br>**[15:9] Reserved.**<br>**[8] (RO) EXTENDED STATUS: Default 1**<br>This bit indicates that PHY status information.<br>**[7] Reserved.**<br>**[6] (RO) MF PREMABLE SUPPRESSION ENABLE: Default 1**<br>This bit indicates whether the PHY is capable of handling MII Management Frames without the 32-bit preamble field.<br>Returns 1 to indicate support for suppressing preamble MII Management Frames.<br>**[5] (RO) AUTO-NEGOTIATION COMPLETE:**<br>When 1, this bit indicates that the Auto-negotiation process has completed.<br>This bit returns '0' when either the Auto-negotiation process is underway or the Auto-negotiation function is disabled.<br>**[4] (RO) REMOTE FAULT: Default 0**<br>When 1, a remote fault condition has been detected between the CoreSGMII and the PHY.<br>**[3] (RO) AUTO-NEGOTIATION ABILITY: Default 1**<br>When 1, this bit indicates that the CoreSGMII has the ability to perform Auto-negotiation.<br>**[2] (RO) LINK STATUS: Default 0**<br>When 1, this bit indicates that a valid link has been established between the CoreSGMII and PHY.<br>When 0, no valid link has been established.<br>**[1] Reserved.**<br>**[0] (RO) EXTENDED CAPABILITY: Default 1**<br>This bit indicates that the CoreSGMII contains the extended set of registers. |
| 0x02 | **Reserved** |
| 0x03 | **Reserved** |

***Table 3 •*** **Register Descriptions *(continued)***

| Address | Function |
|---|---|
| 0x04 | **AN Advertisement (SGMII)**:<br>• Bit- 15, Tx_config from PHY to MAC- Link: 1=link up, 0=link down, Tx_config from MAC to PHY- 0:Reserved<br>• Bit- 14, Tx_config from PHY to MAC- Reserved for AN ACK, Tx_config from MAC to PHY- 1<br>• Bit- 13, Tx_config from PHY to MAC- 0:Reserved, Tx_config from MAC to PHY- 0:Reserved<br>• Bit- 12, Tx_config from PHY to MAC- Duplex mode: 1=full, 0=half, Tx_config from MAC to PHY- 0:Reserved<br>• Bit- 11:10, Tx_config from PHY to MAC- Speed:<br>11 = Reserved 10 = 1000 Mbps<br>01 =100 Mbps 00 = 10 Mbps<br>• Bit- 9:1, Tx_config from PHY to MAC- 0:Reserved, Tx_config from MAC to PHY- 0:Reserved<br>• Bit- 0, Tx_config from PHY to MAC- 1, Tx_config from MAC to PHY- 1<br>**[15] (R/W) LINK UP**: Assertion of this bit indicates that the link is up. When the CoreSGMII integrated within MAC this bit must be written '0'.<br>**[14] (RO) ACK (Reserved). Ignore on read.**<br>**[13] (R/W) RESERVED**:<br>This bit must be written '0' for correct CoreSGMII operation.<br>**[12] (R/W) FULL-DUPLEX**:<br>The assertion of this bit indicates that the link is transferring data in full duplex mode. When the CoreSGMII is integrated within MAC this bit must be written '0'.<br>**[11:10] (R/W) LINK SPEED**:<br>Link speed set by the application for the Auto negotiation. These bits have no relation with the SPEEDO port signal.<br>**[9:0] (R/W)**: These bits must always be written '0000000001' for correct CoreSGMII operation.<br>**AN Advertisement (1000BASE-X)**<br>**[15] (R/W) NEXT PAGE: Default 0**<br>The local device asserts this bit to either request Next Page transmission or advertise Next Page exchange capability. This bit can thus be set when the local has no Next Pages but wishes to allow reception of Next Pages. If the local device has no Next Pages, and the Link Partner wishes to send Next Pages, the local device must send Null Message Codes and have the MESSAGE PAGE set to 0b000_0000_0001. This bit must be cleared where the local device wishes not to engage in Next Page exchange.<br>**[14] Reserved.**<br>**[13:12] (R/W) REMOTE FAULT: Default 0**<br>Encodes the local device's remote fault condition. A fault may be indicated by setting a non-zero Remote Fault encoding and re-negotiating.<br>• RF1 (4.12)- 0, RF2 (4.13)- 0, No error, link Ok.<br>• RF1 (4.12)- 0, RF2 (4.13)- 1, Offline.<br>• RF1 (4.12)- 0, RF2 (4.13)- 0, Link_Failure.<br>• RF1 (4.12)- 0, RF2 (4.13)- 1, Auto-Negotiation_Error. |

*Table 3 •* **Register Descriptions *(continued)***

| Address | Function |
|---|---|
| | **[11:9] Reserved**<br>**[8:7] (R/W) PAUSE: Encodes the local device's PAUSE capability.**<br>**Pause Encoding:**<br>• PAUSE1 (4.7)- 0, ASM_DIR (4.8)- 0, Capability- No PAUSE.<br>• PAUSE1 (4.7)- 0, ASM_DIR (4.8)- 1, Capability- Asymmetric PAUSE toward link partner.<br>• PAUSE1 (4.7)- 1, ASM_DIR (4.8)- 0, Capability- Symmetric PAUSE.<br>• PAUSE1 (4.7)- 1, ASM_DIR (4.8)- 1, Capability- Asymmetric PAUSE toward local device.<br>**[6] (R/W) HALF-DUPLEX:**<br>Setting this bit means the local device is capable of half-duplex operation.<br>**[5] (R/W) FULL-DUPLEX:**<br>Setting this bit means the local device is capable of full-duplex operation.<br>**[4:0] Reserved** |
| 0x05 | **AN Link Partner Base Page Ability (SGMII)**<br>**[15] (RO) LINK UP**: This bit indicates that the link is up.<br>**[14:13] Reserved**<br>**[12] (RO) FULL DUPLEX**: Link partner full duplex ability received.<br>**[11:10] (RO) LINK SPEED**: Link partner speed ability received.<br>**[9:0] (R/W)**: These bits must always be written 0000000001 for correct CoreSGMII operation.<br>**AN Link Partner Base Page Ability (1000BASE-X)**<br>**[15] (RO) NEXT PAGE**:<br>The Link Partner asserts this bit either to request Next Page transmission or to indicate the capability to receive Next Pages. When 0, the Link Partner has no subsequent Next Pages or is not capable of receiving Next Pages.<br>**[14] (RO) ACK (Reserved): Ignore on read.**<br>**[13:12] (RO) REMOTE FAULT**: Encodes the Link Partner's remote fault condition.<br>• RF1 (4.12)- 0, RF2 (4.13)- 0, No error, link OK.<br>• RF1 (4.12)- 0, RF2 (4.13)- 1, Offline<br>• RF1 (4.12)- 1, RF2 (4.13)- 0, Link_Failure<br>• RF1 (4.12)- 1, RF2 (4.13)- 1, Auto-Negotiation_Error.<br>**[11:9] Reserved**<br>**[8:7] (RO) PAUSE**: Encodes of the Link Partner's PAUSE capability.<br>Pause Encoding:<br>• PAUSE1 (4.7)- 0, ASM_DIR (4.8)- 0, Capability- No PAUSE<br>• PAUSE1 (4.7)- 0, ASM_DIR (4.8)- 1, Capability- Asymmetric PAUSE toward link partner.<br>• PAUSE1 (4.7)- 1, ASM_DIR (4.8)- 0, Capability- Symmetric PAUSE.<br>• PAUSE1 (4.7)- 1, ASM_DIR (4.8)- 1, Capability- Asymmetric PAUSE toward local device.<br>**[6] (RO) HALF-DUPLEX**: When 1, Link Partner is capable of half-duplex operation. When 0, Link Partner is incapable of half-duplex mode.<br>**[5] (RO) FULL-DUPLEX**: When 1, Link Partner is capable of full-duplex operation. When 0, Link Partner is incapable of full-duplex mode.<br>**[4:0] Reserved** |

***Table 3 •*** **Register Descriptions *(continued)***

| Address | Function |
|---|---|
| 0x06 | **AN Expansion (SGMII)**<br>**[15:3] Reserved**<br>**[2] (RO) NEXT PAGE ABLE: Default 1**<br>When 1, indicates that the local device supports the Next Page function.<br>**[1] (RO) PAGE RECEIVED**:<br>When 1, indicates that a new page has been received and stored in the applicable AN LINK PARTNER ABILITY or AN NEXT PAGE.<br>**[0] Reserved**<br>**AN Expansion (1000BASE-X)**<br>**[15:3] Reserved**<br>**[2] (RO) NEXT PAGE ABL**E: 1 indicates the local device supports Next Page function. Returns 1 on read.<br>**[1] (RO,LH) PAGE RECEIVED**:<br>1 indicates that a new page has been received and stored in the applicable AN LINK PARTNER ABILITY or AN NEXT PAGE register. This bit latches high in order for the software to detect when polling. The bit is cleared on a read to the register. |
| 0x07 | **AN Next Page Transmit (SGMII)**<br>Use of this register is user dependent. User can define the functionality of bits of this register as per system required.<br>**[15:0] User defined Register**<br>**AN Next Page Transmit (1000BASE-X)**<br>**[15] (R/W) NEXT PAGE**:<br>Assert this bit to indicate additional Next Pages to follow. The bit is cleared to indicate last page.<br>**[14] (RO) ACK (Reserved)**: Write '0', ignore on read.<br>**[13] (R/W) MESSAGE PAGE**:<br>Assert bit to indicate Message Page. Clear bit to indicate Unformatted Page.<br>**[12] (R/W) ACKNOWLEDGE 2**:<br>Used by Next Page function to indicate device has ability to comply with the message. Assert bit if local device complies with message. Clear bit if the local device cannot comply with the message.<br>**[11] (RO) TOGGL**E:<br>Used to ensure synchronization with the Link Partner during Next Page exchange. This bit always takes the opposite value of the Toggle bit of the previously exchanged Link Code Word. The initial value in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word.<br>**[10:0] (R/W) MESSAGE / UNFORMATTED CODE FIELD**:<br>Message pages are formatted pages that carry a predefined Message Code, which is enumerated in IEEE 802.3u/Annex 28C. Unformatted Code Fields take on an arbitrary value. |

*Table 3 •* **Register Descriptions *(continued)***

| Address | Function |
|---|---|
| 0x08 | **AN Link Partner Ability Next Page (SGMII)**<br>Use of this register is user dependent. User can define the functionality of bits of this register as per system required.<br>**[15:0] User defined Register**<br>**AN Link Partner Ability Next Page (1000BASE-X)**<br>**[15] (RO) NEXT PAGE**:<br>The Link Partner asserts this bit to indicate additional Next Pages to follow. When 0, indicates last Next Page from link partner.<br>**[14] (RO) ACK (Reserved): Ignore on read.**<br>**[13] (RO) MESSAGE PAGE**:<br>When 1, indicates Message Page. When '0', indicates Unformatted Page.<br>**[12] (RO) ACKNOWLEDGE 2**:<br>Indicates Link Partner's ability to comply with the message. When 1, Link Partner complies with message. When 0, Link Partner cannot comply with the message.<br>**[11] (RO) TOGGLE**:<br>Used to ensure synchronization with the Link Partner during Next Page exchange. This bit always takes the opposite value of the Toggle bit of the previously exchanged Link Code Word. The initial value in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word.<br>**[10:0] (RO) MESSAGE / UNFORMATTED CODE FIEL**D:<br>Message pages are formatted pages that carry a predefined Message Code, which is enumerated in IEEE 802.3u/Annex 28C. Unformatted Code Fields take on an arbitrary value. |
| 0x0F | **Extended Status**<br>**[15] (RO) 1000BASE-X FULL-DUPLEX: Default 1**<br>When 1, indicates PHY can operate in 1000BASE-X full-duplex mode. When '0', indicates PHY cannot operate in this mode.<br>**[14] (RO) 1000BASE-X HALF-DUPLEX: Default 0**<br>When 1, indicates PHY can operate in 1000BASE-X half-duplex mode. When 0, indicates PHY cannot operate in this mode.<br>**[13] (RO) 1000BASE-T FULL-DUPLEX: Default 1**<br>When 1, indicates PHY can operate in 1000BASE-T full-duplex mode. When 0, indicates PHY cannot operate in this mode.<br>**[12] (RO) 1000BASE-T HALF-DUPLEX: Default 0**<br>When 1, indicates PHY can operate in 1000BASE-T half-duplex mode. When 0, indicates PHY cannot operate in this mode.<br>**[11:0] Reserved** |

***Table 3 •*** **Register Descriptions *(continued)***

| Address | Function |
|---|---|
| 0x10 | **Jitter Diagnostics**<br>**[15] (R/W) JITTER DIAGNOSTIC ENABLE: Default 0**<br>Set this bit to enable the CoreSGMII to transmit the jitter test patterns defined in IEEE 802.3z 36A. Clear this bit to enable normal transmit-operation.<br>**[14:12] (R/W) JITTER PATTERN SELECT: Default is 0**<br>Selects the jitter pattern to be transmitted in diagnostics mode.<br>Jitter Pattern Select Encodings<br>• Bit14-0, Bit13-0, Bit12-0, Jitter Pattern Select- User Defined Custom Pattern<br>• Bit14-0, Bit13-0, Bit12-1, Jitter Pattern Select- 802.3z 36A Defined High Frequency 10101010101010101010…<br>• Bit14-0, Bit13-1, Bit12-0, Jitter Pattern Select- 802.3z 36A Defined Mixed Frequency 11111010110000010100…<br>• Bit14-0, Bit13-1, Bit12-1, Jitter Pattern Select- Custom Defined Low Frequency 11111000001111100000...<br>• Bit14-1, Bit13-0, Bit12-0, Jitter Pattern Select- Random Jitter Pattern<br>• Bit14-1, Bit13-0, Bit12-1, Jitter Pattern Select- 802.3z 36A Defined Low Frequency 11111000001111100000…<br>• Bit14-1, Bit13-1, Bit12-0, Jitter Pattern Select- Reserved<br>• Bit14-1, Bit13-1, Bit12-1, Jitter Pattern Select- Reserved<br>**[11:10] Reserved**<br>**[9:0] (R/W) CUSTOM JITTER PATTERN: Default 0**<br>Used in conjunction with JITTER PATTERN SELECT and JITTER DIAGNOSTIC ENABLE. Set this field to the desired custom pattern. |

***Table 3 •*** **Register Descriptions *(continued)***

| Address | Function |
|---|---|
| 0x11 | **Ten Bit Interface Control**<br>**[15] (R/W) SOFT RESET: Default 0**<br>This bit resets the functional modules in the CoreSGMII. Clear it for normal operation.<br>**[14] (R/W) SHORTCUT LINK TIMER: Default 0**<br>Set this bit 1 to reduce the value of Go Link Timer and Sync. Status Fail Timer to 64 clock pulse. This reduces the simulation time needed to time the 1.6ms Link Timer. Clear it for normal operation. In normal operation, the value of Go Link Timer is 200000 clock pulses and the value of the Sync. Status Fail Timer is 1250000 clock pulses.<br>**[13] (R/W) DISABLE RECEIVE RUNNING DISPARITY**: Default 0<br>Set this bit to disable the running disparity calculation and checking in the receive direction. This bit must be 0 for correct CoreSGMII operation.<br>**[12] (R/W) DISABLE TRANSMIT RUNNING DISPARITY**: Default 0<br>Set this bit to disable the running disparity calculation and checking in the transmit direction. This bit must be 0 for correct CoreSGMII operation.<br>**[11] (R/W) GO LINK TIMER VALUE CONTRO**L: Default 0<br>When 0 the Go Link Timer Value=1.6ms<br>When set to 1 the Go Link Timer Value=10ms<br>**[10:9] Reserved**<br>**[8] (R/W) AUTO-NEGOTIATION SENSE**: Default 0<br>Set this bit to allow the Auto-Negotiation function to sense either a MAC in Auto-Negotiation bypass mode or an older MAC without Auto-Negotiation capability. When sensed, Auto-Negotiation Complete becomes true; however, Page Received will be low, indicating no page is exchanged. Management can then act accordingly. Clear this bit when IEEE 802.3z Clause 37 behavior is desired, which results in the link not coming up.<br>**[7:4] Reserved**<br>**[3:2] (R/W) SPEEDO**: Default '10'<br>• Bit 3- 0, Bit2- 1, Reserved<br>• Bit 3- 1, Bit2- 0, 1000 Mbps<br>• Bit 3- 0, Bit2- 1, 100 Mbps<br>• Bit 3- 0, Bit2- 0, 10 Mbps<br>**[1:0] Reserved** |

# 5 Interface

## 5.1 Ports

The port signals for CoreSGMII are described in Table 4 and as shown in Figure 2.

***Figure 2 •*** **CoreSGMII I/O Signals**

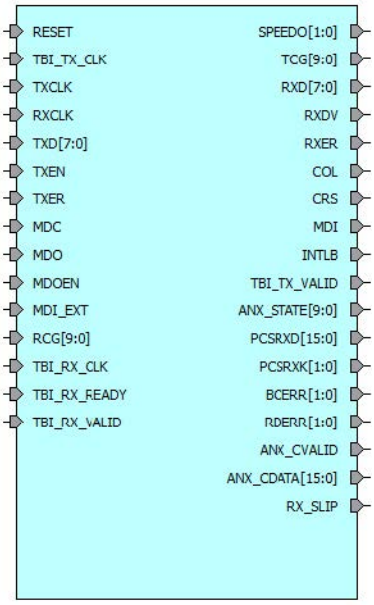



***Table 4 •*** **I/O Signal Description**

| Signal | Direction | Description |
|---|---|---|
| | | **Reset** |
| RESET | Input | Active high reset. |
| | | **Clocks** |
| TBI_TX_CLK | Input | 125 MHz TBI transmit clock from XCVR. |
| TXCLK | Input | 2.5/25/125 MHz transmit clock generated from XCVR TX clock according to 10/100/1000 Mbps support. |
| RXCLK | Input | 2.5/25/125 MHz receive clock generated from XCVR's RX clock according to 10/100/1000 Mbps. |
| TBI_RX_CLK | Input | 125 MHz TBI receive clock from XCVR. |
| | | **G/MII Interface** |
| TXD[7:0] | Input | G/MII transmit data |
| TXEN | Input | G/MII transmit enable |
| TXER | Input | G/MII transmit error |
| RXD[7:0] | Output | G/MII Receive data |
| RXDV | Output | G/MII Receive data valid |
| RXER | Output | G/MII Receive error |

***Table 4 •*** **I/O Signal Description *(continued)***

| Signal | Direction | Description |
|---|---|---|
| COL | Output | MII collision |
| CRS | Output | MII carrier sense |
| | | **TBI Interface** |
| RCG[9:0] | Input | TBI Receive code group |
| TCG[9:0] | Output | TBI Transmit code group |
| TBI_RX_READY | Input | RCG valid, recommended to connect with XCVR Ready. |
| TBI_TX_VALID | Output | TCG valid, recommended to connect with XCVR transmit valid<br>**Note:** This signal is not available for PolarFire and PolarFire SoC. |
| TBI_RX_VALID | Input | Available for PolarFire and PolarFire SoC., recommended to connect with receive valid of XCVR |
| RX_SLIP | Output | Available for PolarFire and PolarFire SoC., recommended to connect with receive slip signal of XCVR |
| | | **MDIO Interface** |
| MDC | Input | Management data clock, recommended to drive 2.5 MHz |
| MDO | Input | Management data output |
| MDOEN | Input | Management data output enable |
| MDI_EXT | Input | Management data input from external PCS/PHY |
| MDI | Output | MII Management data Input |
| | | **Other Interface Signals** |
| INTLB | Output | Loop back enabled status |
| SPEEDO[1:0] | Output | SPEEDO refers to '00'/'01'/'10' for 10/100/1000 Mbps respectively. SPEEDO signal is used for 2.5/25/125 MHz TXCLK and RXCLK clocks generation. PHY speed read by the driver/application must be updated to Ten Bit Interface Control (0x11) |
| ANX_STATE[9:0] | Output | Auto negotiation status information<br>$0^{th}$ bit - DISABLE_LINK_OK state<br>$1^{st}$ bit - AN_ENABLE state<br>$2^{nd}$ bit - AN_RESTART state<br>$3^{rd}$ bit - ABILITY_DETECT state<br>$4^{th}$ bit - ACKNOWLEDGE_DETECT state<br>$5^{th}$ bit - NEXT_PAGE_WAIT state<br>$6^{th}$ bit - COMPLETE_ACKNOWLEDGE state<br>$7^{th}$ bit - IDLE_DETECT state<br>$8^{th}$ bit - LINK_OK state<br>$9^{th}$ bit - Received configuration frame data |
| BCERR[1:0] | Output | BCERR [1] indicates RCG code error in PCSRXD[15:8] and BCERR [0] indicates RCG code error in PCSRXD[7:0], BCERR[1:0] is an active high signal |
| RDERR[1:0] | Output | RDERR[1] indicates RCG disparity error in PCSRXD[15:8] and RDERR[0] indicates RCG disparity error in PCSRXD[7:0], RDERR[1:0] is an active high signal |
| PCSRXD[15:0] | Output | This port is intended to be used to monitor the decoded receive data |
| PCSRXK[1:0] | Output | PCSRXK[1] indicates K character indicator associated with the PCSRXD[15:8] and PCSRXK[0] indicates K character indicator associated with the PCSRXD[7:0], PCSRXK[1:0] is an active high signal |

*Table 4 •* **I/O Signal Description *(continued)***

| Signal | Direction | Description |
|---|---|---|
| ANX_CDATA[15:0] | Output | This port provides the received C1 and C2 data from the auto-negotiation process. The auto-negotiation state machine inside the CoreSGMII terminates the C1 and C2 data. This port is intended to be used to monitor the received data |
| ANX_CVALID | Output | C1 and C2 ordered set valid indicator |

## 5.2 Configuration Parameters

The register transfer level (RTL) code for CoreSGMII has parameters for configuring the core. While working with the core in the SmartDesign tool, a configuration GUI is used to set the values of these parameters.

*Table 5 •* **CoreSGMII Configuration Parameters**

| Name | Valid Range | Default | Description |
|---|---|---|---|
| MDIO_PHYID | 0 or 31 | 12 | MDIO PHY address.<br>**Note:** MSS Hard MAC user to avoid the address 30 |
| SLIP_ENABLE | 0 or 1 | 0 | Include receive slip logic<br>1: Include<br>0: Do not include<br>**Note:** Applicable for PolarFire and PolarFire SoC. The SLIP_ENABLE in CoreSGMII has to be configured according to configuration in XCVR. |

# 6 Timing Diagrams

## 6.1 G/MII Timing Diagrams

Refer to http://standards.ieee.org/getieee802/download/802.3-2012_section2.pdf

- Figure 22-4: Transmission with no collision
- Figure 22-5: Propagating an error
- Figure 22-6: LPI transition
- Figure 22-7: Reception with no errors
- Figure 22-9: Reception with errors
- Figure 22-10: False Carrier indication
- Figure 22-11: Transmission with collision

## 6.2 TBI Interface Timing Diagrams

Refer to http://standards.ieee.org/getieee802/download/802.3-2012_section3.pdf

- Figure 36-14: TBI transmit interface timing diagram
- Figure 36-15: TBI receive interface timing diagram

## 6.3 MDIO Timing Diagrams

Refer to http://standards.ieee.org/getieee802/download/802.3-2012_section2.pdf

- Figure 22-15: Behavior of MDIO during TA field of a read transaction
- Figure 22-18: MDIO sourced by STA
- Figure 22-19: MDIO sourced by PHY

# 7 Tool Flow

## 7.1 License

CoreSGMII is licensed core and is available as obfuscated and Encrypted RTL.

### 7.1.1 Obfuscated

Complete RTL code is provided for the core, enabling the core to be instantiated with SmartDesign. Simulation, Synthesis, and Layout can be performed with Libero® System-on-Chip (SoC) or Integrated Design Environment (IDE). The RTL code for the core is obfuscated and some of the testbench source files are not provided. Instead, they are precompiled into the compiled simulation library.

### 7.1.2 RTL

Complete RTL source code is provided for the core and testbenches.

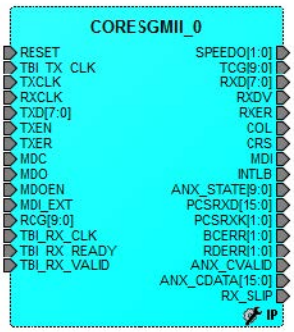
## 7.2 SmartDesign

CoreSGMII is available for download in the Libero IP catalog through the web repository. Once it is listed in the catalog, the core can be instantiated using the SmartDesign flow. To know how to create SmartDesign project using the IP cores, refer to *Libero SoC documents page* and use the latest SmartDesign user guide. An example instantiated view is shown in Figure 3.

After configuring and generating the core instance, the basic functionality can be simulated using the test-bench supplied with the CoreSGMII. The testbench parameters automatically adjust to the CoreSGMII configuration. The CoreSGMII can be instantiated as a component of a larger design.

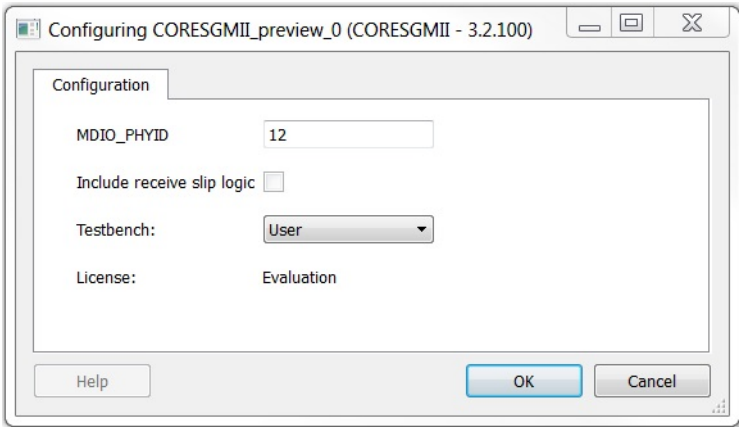
CoreSGMII is compatible with Libero SoC

***Figure 3 •*** **SmartDesign CoreSGMII Instance View**



## 7.3 Configuring CoreSGMII in SmartDesign

***Figure 4 •*** **Configuring CoreSGMII in SmartDesign**




## 7.4 Simulation Flows

To run simulations, select the user testbench in the core configuration window. After generating the CoreSGMII, the pre-synthesis test-bench hardware description language (HDL) files are installed in Liberia.

## 7.5 Synthesis in Libero

To run synthesis on the CoreSGMII, set the design root to the IP component instance and run the synthesis tool from the Libero design flow pane.

## 7.6 Place-and-Route in Libero

After the design is synthesized, run the compilation and then place-and-route the tools. CoreSGMII requires no special place-and-route settings.

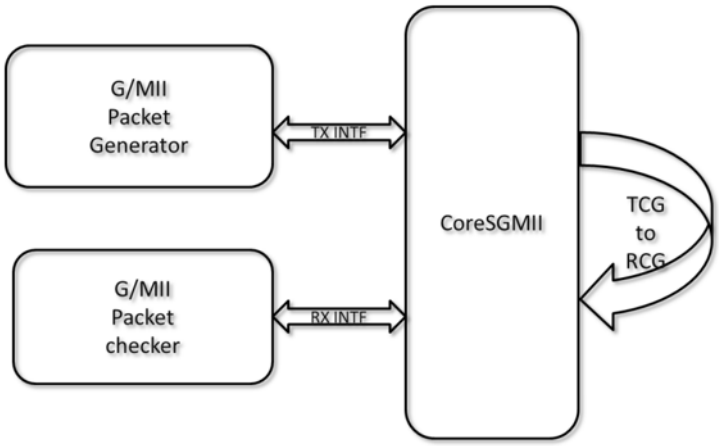
# 8 Testbench

A unified test-bench is used to verify and test CoreSGMII called as user test-bench.

## 8.1 User Test-bench

The user test-bench is included with the releases of CoreSGMII that verifies few features of the CoreSGMII.

***Figure 5 •*** **CoreSGMII User Test-bench**




A simplified block diagram of the user test-bench is, as shown in Figure 5.

The user test-bench generates necessary clocks, resets.

Test bench is equipped with below tasks to check the AN capability and transaction activity.

- Write, read access to registers using MDIO interface.
- Generate packets in G/MII and TBI interfaces.
- Monitor and data integrity check on G/MII.

### 8.1.1 Test Cases

#### 8.1.1.1 RX_SLIP Test Case

This test case checks the SLIP functionality by sending un-aligning the RCG code group to CoreSGMII and checks for correct number of RX_SLIP pulses. This test will be simulated only when slip functionality is enabled.

- Configure MDIO registers for PHY reset.
- Wait for few clocks for PHY reset completion and shift the RCG value.
- Wait for RX_SLIP before correcting the RCG shift.
- Check for RX_SLIP is applied to align RCG code group is sent.
- TBI_RX_VALID will be driven low 3 clocks after RX_SLIP until for 17 clock cycles.

#### 8.1.1.2 Auto negotiation Test Case

- Configure MDIO registers for restarting and enable auto negotiation.
- Waits for auto negotiation completion.
- Verifies the Autonegation status in MDIO registers and ANX_STATE port status.

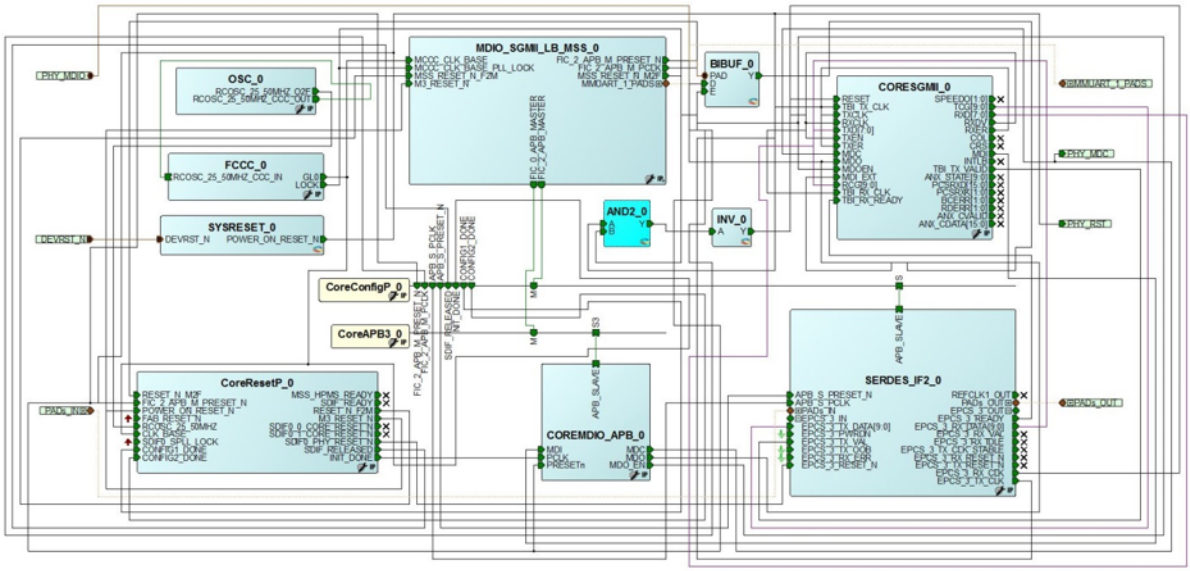
#### 8.1.1.3 Loop Back Test Case in 1G Mode

- Configure MDIO registers for full duplex and 1000mbps mode of operation.
- Inputs G/MII test vector with data byte incrementing from 'h00 to 'h7.
- Verifies the data received on a G/MII interface with expected data.

# 9 System Integration

The example design explains the CoreSGMII features and implements the loopback application for 1000Mbps mode on the SmartFusion2 Security Evaluation Kit.

***Figure 6 •*** **CoreSGMII Example Design**




- FABRIC RESET (SYSRESET_0) is used for all resets.
- CORESGMII_0 has TXCLK, RXCLK, TBI_TX_CLK, TBI_RX_CLK, and MDC clocks.
- TBI_TX_CLK and TXCLK are connected to 125MHz EPCS_3_TX_CLK of SERDES_IF2_0.
- TBI_RX_CLK and RXCLK are connected to 125MHz EPCS_3_RX_CLK of SERDES_IF2_0.
- 2.5 MHz MDC is sourced from MDC of COREMDIO_APB_0.
- Microcontroller Subsystem (MSS) configures the CORESGMII_0 and SERDES_IF2_0 through the application.

Run the Libero flow with enabling the Timing Driven and High Effort Place and Route options enabled. The example design can be obtained from the Microsemi technical support team.