HB0788

# Handbook

# CoreSmartBERT v2.6

Microsemi

a Microchip company

**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

Microsemi makes no warranty, representation, or guarantee regarding the information contained herein or the suitability of its products and services for any particular purpose, nor does Microsemi assume any liability whatsoever arising out of the application or use of any product or circuit. The products sold hereunder and any other products sold by Microsemi have been subject to limited testing and should not be used in conjunction with mission-critical equipment or applications. Any performance specifications are believed to be reliable but are not verified, and Buyer must conduct and complete all performance and other testing of the products, alone and together with, or installed in, any end-products. Buyer shall not rely on any data and performance specifications or parameters provided by Microsemi. It is the Buyer's responsibility to independently determine suitability of any products and to test and verify the same. The information provided by Microsemi hereunder is provided "as is, where is" and with all faults, and the entire risk associated with such information is entirely with the Buyer. Microsemi does not grant, explicitly or implicitly, to any party any patent rights, licenses, or any other IP rights, whether with regard to such information itself or anything described by such information. Information provided in this document is proprietary to Microsemi, and Microsemi reserves the right to make any changes to the information in this document or to any products and services at any time without notice.

## 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 7.0

Updated for CoreSmartBERT v2.6, March 2020.

## 1.2 Revision 6.0

Updated for CoreSmartBERT v2.5, August 2019.

## 1.3 Revision 5.0

Updated for CoreSmartBERT v2.4, March 2019.

## 1.4 Revision 4.0

Updated for CoreSmartBERT v2.3, December 2018.

## 1.5 Revision 3.0

Updated for CoreSmartBERT v2.2, August 2018.

## 1.6 Revision 2.0

Added PF_XCVR information. Updated for CoreSmartBERT v2.1, May 2018.

## 1.7 Revision 1.0

Revision 1.0 was the first publication of this document. Created for CoreSmartBERT v2.0, August 2017.

# 2 Introduction

CoreSmartBERT core provides a broad-based evaluation and demonstration platform for PolarFire Transceiver (PF_XCVR). Parameterizable to use different transceivers and clocking topologies, the SmartBERT core can also be customized to use different line rates and reference clock rates. Data pattern generators and checkers are included for each PF_XCVR, giving several different Pseudo-random binary sequences PRBS ($2^7$, $2^9$, $2^{23}$, and $2^{31}$). The pattern generator sends data out through the transmitter. The pattern checker accepts data through the receiver and checks it against an internally generated pattern. These patterns are optimized for the logic width that was selected at run time. SmartDebug will provide the user interface to the this core.

This handbook provides information on the CoreSmartBERT IP core and the features it supports. The purpose of this IP core is to add more test features for the PolarFire Transceiver (PF_XCVR) so that, the end user can evaluate the PMA functionality of PF_XCVR on a board. The SmartDebug tool will interface with this core, which allows the user to have an interactive GUI control.

## 2.1 Features

- Ability to support PF_XCVR built-in PRBS generator or checker
- Ability to generate a list of patterns
- Ability to insert an error into the transmit pattern
- Ability to support checking of errors in the receive pattern
- Ability to instantiate the pattern functionality several times to support multiple lanes simultaneously

## 2.2 Core Version

This handbook supports CoreSmartBERT version 2.6.

## 2.3 Supported Families

- PolarFire®

## 2.4 Device Utilization and Performance

CoreSmartBERT has been implemented in the following Microsemi device families. A summary of the implementation data for CoreSmartBERT is listed in Table 1, page 2

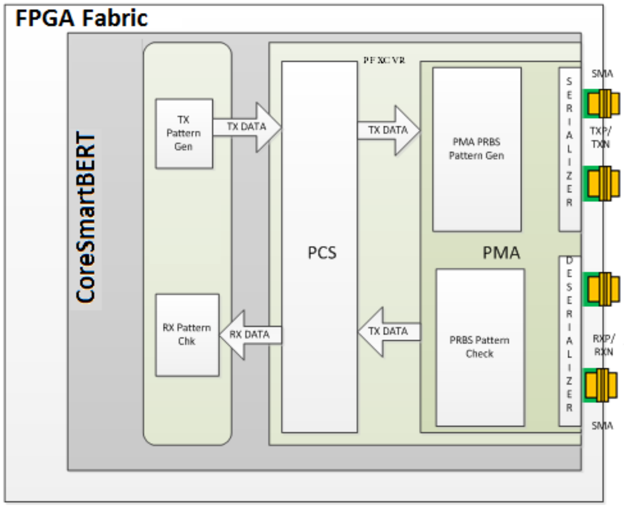
***Table 1 •*** **CoreSmartBERT Utilization**

| | Logic Elements | | | Utilization | | |
|---|---|---|---|---|---|---|
| **Family** | **Sequential** | **Combinatorial** | **Total** | **Device** | **Total %** | **Performance MHz** |
| PolarFire | 1078 | 5850 | 6928 | MPF300T | 2.31 | 156.25 |

# 3 Functional Description

CoreSmartBERT includes the PolarFire Transceiver, which will interface with the SmartDEBUG tool through a user control GUI to run the hardened PRBS generator and checkers. It also has fabric pattern generators and checks with more features (for example, error injection) than the ones included in PolarFire Transceiver.

***Figure 1 •*** **CoreSmartBERT**



## 3.1 CoreSmartBERT Blocks

CoreSmartBERT blocks consists of the following:

### 3.1.1 TX Pattern Gen

This Transmitter block supports the generation of the following pattern:

- PRBS 7, 9, 23, 31 with error insertion logic

### 3.1.2 RX Pattern CHK

The is Receiver block supports the checking of the following patterns:

- PRBS 7, 9, 23, 31

### 3.1.3 PF XCVR

The PF XCVR is PolarFire's Transceiver macro in PMA mode.

# 4 Operation

## 4.1 Test Pattern Overview

CoreSmartBERT supports the following test patterns:

### 4.1.1 PRBS

Pseudo Random Binary Sequence (PRBS) Test Patterns are intended to generate deterministic sequences with the properties of highly random signals, for example: white noise.

CoreSmartBERT supports the built in PBRS pattern generators and checkers in the PolarFire Transceiver and will add the support for fabric PRBS pattern generators and checkers with the ability to inject errors into the transmitter path.

Theses include support for the following:

- PRBS 7
- PRBS 9
- PRBS 23
- PRBS 31

## 4.2 Smart Debug Tool

SmartDebug provides the User Interface to control the CoreSmartBERT to use its features.

The following are the capabilities provided by SmartDebug:

1. Ability to control **CoreSmartBERT** and have the signal integrity controls on the screen at the same time.
2. Automatic detection of the presence of the **CoreSmartBERT** in the design.
3. Ability to select the particular transceiver lane associated with the CoreSmartBERT.
4. A drop-down list to select the specific pattern
5. An ability to enable to start the pattern transmitter
6. An ability to enable to start the pattern receiver
7. A button to insert a single error
8. An error counter with a clear button

### 4.2.1 SLE_DEBUG Marco

SLE_DEBUG Marco will be used to communicate with SmartDebug. The **SLE_DEBUG mechanism** gives ability to run synthesis while preserving a set of registers. It provides ability to identify, rename, and classify registers for SmartDebug.

CoreSmartBERT will have SLE_DEBUG write and read registers that are used to inform the Smart Debug tool of parameters settings chosen, IP core versions number, and control various functions (for example, error injection, read errors, and so on.). Table 2, page 5 describes the SLE_DEBUG registers that will be used in CoreSmartBERT.

***Table 2 •*** **SLE_DEBUG Registers**

| Bits | Function | Type | Description |
|---|---|---|---|
| 14 | SLE_DATA_RATE | R | Reads data rate selected from the GUI. |
| 4 | SLE_TX_CLK_DIV_FACTOR | R | Reads Tx Clock Divide Factor selected from the GUI. |
| 1 | SLE_CDR_REFERENCE_CLK_SOURCE | R | Reads the CDR Reference Clock Source selected from the GUI:<br>0: Dedicated<br>1: Fabric |
| 4 | SLE_CDR_REFERENCE_CLK_FREQ | R | Reads the CDR Reference Clock Frequency selected from the GUI:<br>0: 25.00<br>1: 31.25<br>2: 50.00<br>3: 62.50<br>4: 75.00<br>5: 100.00<br>6: 125.00<br>7: 150.00<br>8: 156.25<br>9: 312.50 |
| 2 | SLE_NUMBER_OF_LANES | R | Reads the number of lanes this IP core has enabled from the GUI. |
| 1 | SLE_PATTERN_PRBS7 | R | Reads the PRBS7 pattern enable from the GUI. |
| 1 | SLE_PATTERN_PRBS9 | R | Reads the PRBS9 pattern enable from the GUI. |
| 1 | SLE_PATTERN_PRBS23 | R | Reads the PRBS23 pattern enable from the GUI. |
| 1 | SLE_PATTERN_PRBS31 | R | Reads the PRBS31 pattern enable from the GUI. |
| 16 | SLE_CPZ_VERSION | R | Reads the CPZ Version number.<br>This registers represents 8bit major and 8bit minor version number.<br>For example, v2.1 = {8'd2, 8'd1} |

*Table 2 •* **SLE_DEBUG Registers**

| Bits | Function | Type | Description |
|---|---|---|---|
| 4 | SLE_TX_LANE[n]_PATTEN_GEN | RW | Transmitter Pattern Generator:<br>0: PRBS7<br>1: PRBS 9<br>1: PRBS23<br>2: PRBS31<br>**Note:** Default value will be set to 0. |
| 1 | SLE_TX_LANE[n]_GEN_EN | RW | Transmitter Pattern Generator Enable:<br>0: Disabled<br>1: Enabled<br>**Note:** Default will be set to 0. |
| 4 | SLE_RX_LANE[n]_PATTEN_CHK | RW | Receiver Pattern Checker:<br>0: PRBS7<br>1: PRBS9<br>2: PRBS23<br>3: PRBS31<br>**Note:** Default value will be set to 0. |
| 1 | SLE_RX_LANE[n]_CHR_EN | RW | Receiver Pattern Checker Enable:<br>0: Disabled<br>1: Enabled<br>**Note:** Default will be set to 0. |
| 32 | SLE_RX_LANE[n]_ERR_CNT | R | Receiver error counter. |
| 1 | SLE_ RX_LANE[n]_ERR_CNT_CLR | RW | Receiver error counter clear button. |
| 1 | SLE_ RX_LANE[n]_ALIGN | R | Receiver channel aligned to pattern. |

# 5 Interface

## 5.1 Configuration GUI Parameters

CoreSmartBERT has GUI parameters for configuring the core as described in Table 3, page 7.

*Table 3 •* **CoreSmartBERT Parameters Descriptions**

| Name | Range | Default | Description |
|---|---|---|---|
| UI_PATTERN_PRBS7 | 0 or 1 | 1 | PRBS7 Pattern Enable. |
| UI_PATTERN_PRBS9 | 0 or 1 | 1 | PRBS9 Pattern Enable. |
| UI_PATTERN_PRBS23 | 0 or 1 | 1 | PRBS23 Pattern Enable. |
| UI_PATTERN_PRBS31 | 0 or 1 | 1 | PRBS31 Pattern Enable. |
| UI_NUMBER_OF_LANES | 1-4 | 4 | Number of lanes this IP core has enabled. |
| UI_DATA_RATE | 250 - 12700 | 5000 | Transceiver data rate.<br>Supported rates:<br>• 250 Mbps<br>• 1000 Mbps<br>• 1250 Mbps<br>• 2500 Mbps<br>• 3125 Mbps<br>• 5000 Mbps<br>• 6250 Mbps<br>• 8000 Mbps<br>• 10000 Mbps<br>• 12500 Mbps<br>• 12700 Mbps |
| UI_TX_CLK_DIV_FACTOR | 1, 2, 4, 8 & 11 | 1 | TX clock division factor. |
| UI_CDR_REFERENCE_CLK_SOURCE | Dedicated or Fabric | Fabric | CDR reference clock source. |
| UI_CDR_REFERENCE_CLK_FREQ | 0-312.5 | 125 | CDR reference clock frequency.<br>Supported frequencies:<br>• 25.00 Mhz<br>• 31.25 Mhz<br>• 50.00 Mhz<br>• 62.50 Mhz<br>• 75.00 Mhz<br>• 100.00 Mhz<br>• 125.00 Mhz<br>• 150.00 Mhz<br>• 156.25 Mhz<br>• 312.50 Mhz<br>**Note:** Must match PF_XCVR valid values. |

## 5.2 I/O Signals

The initial version of CoreSmartBERT will have the following ports available as shown in Table 4, page 8:

*Table 4 •* **CoreSmartBERT I/O Signal Descriptions**

| Name | Width | Direction | Description |
|---|---|---|---|
| SYS_RESET_N | 1 | IN | Active low system reset. |
| LANE[n]_CDR_REF_CLK_FAB | 1 | IN | CDR reference clock from fabric, only exposed when **Fabric** is selected as CDR reference clock source. |
| LANE[n]_CDR_REF_CLK_0 | 1 | IN | CDR reference clock from dedicated pin, only exposed when **Dedicated** is selected as CDR reference clock source. |
| LANE[n]_TX_BIT_CLK_0 | 1 | IN | Tx Bit Clock. |
| LANE[n]_TX_PLL_REF_CLK_0 | 1 | IN | PLL Reference Clock. |
| LANE[n]_TX_PLL_LOCK_0 | 1 | IN | PLL Lock. |

*Table 5 •* **CoreSmartBERT PAD Signal Descriptions**

| Name | Direction | Description |
|---|---|---|
| LANE[n]_RXD_P | IN | Receiver Serial Data.<br>Where n can be 0, 1, 2, or 3 depending on the number of configured lanes. |
| LANE[n]_RXD_N | IN | |
| LANE[n]_RXD_P | IN | Receiver Serial Data<br>Where n can be 0, 1, 2, or 3 depending on the number of configured lanes. |
| LANE[n]_RXD_N | IN | |

# 6 Tool Flows

## 6.1 Licensing

The CoreSmartBERT does not require any license.

## 6.2 RTL

Complete RTL source code is provided for the core.

## 6.3 SmartDesign

CoreSmartBERT is preinstalled in the SmartDesign IP Deployment design environment.

The core should be configured using the configuration GUI within SmartDesign, as shown in Figure 2, page 9. Information on using SmartDesign to instantiate and generate cores, refer to Libero SoC online help.

***Figure 2 •*** **CoreSmartBERT Full I/O View**

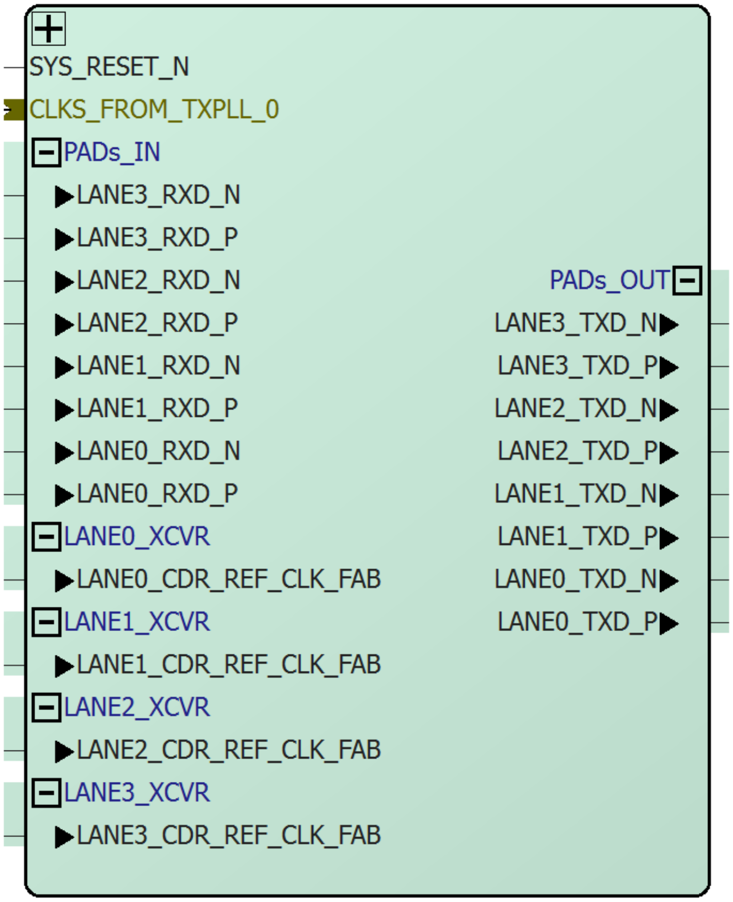
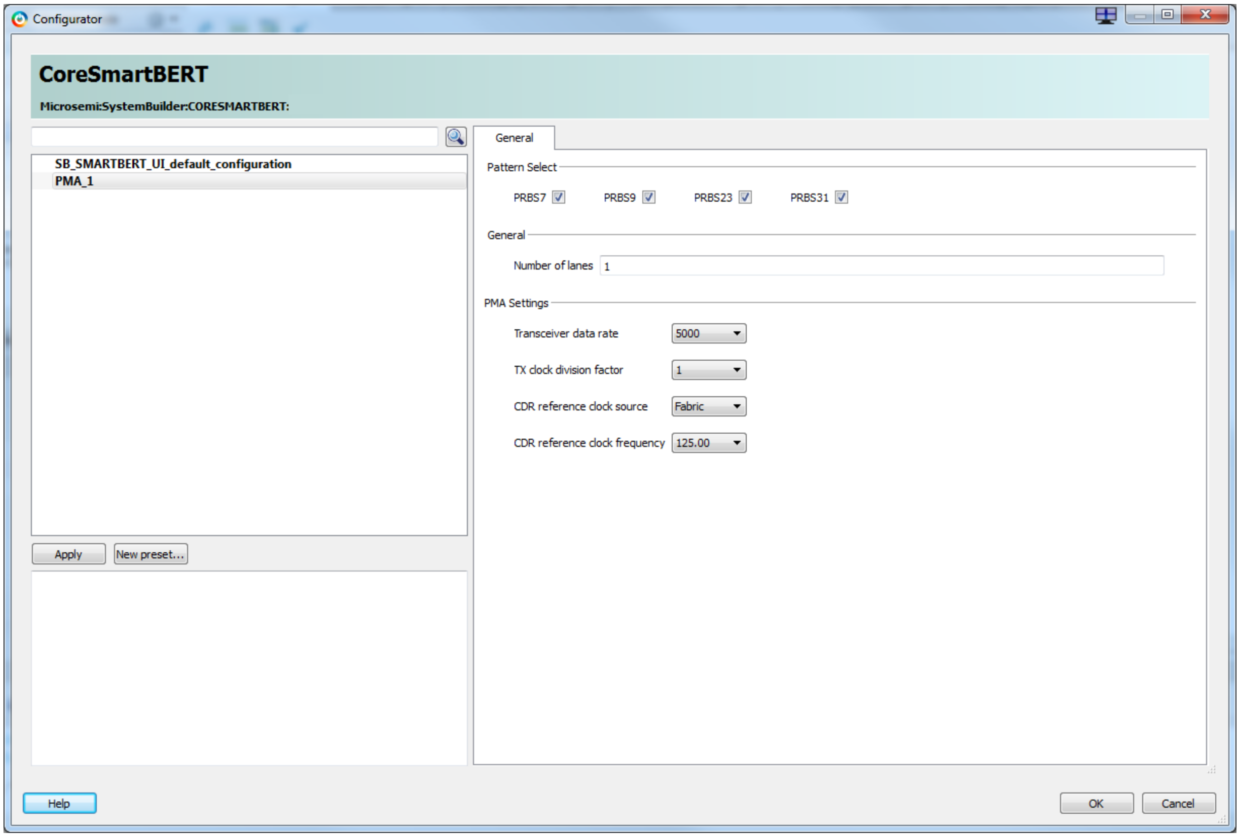


*Figure 3 •* **CoreSmartBERT SmartDesign Configuration GUI**



CoreSmartBERT

Microsemi:SystemBuilder:CORESMARTBERT:

SB_SMARTBERT_UI_default_configuration

PMA_1

General

Pattern Select

PRBS7 ☑ PRBS9 ☑ PRBS23 ☑ PRBS31 ☑

General

Number of lanes 1

PMA Settings

Transceiver data rate 5000

TX clock division factor 1

CDR reference clock source Fabric

CDR reference clock frequency 125.00

## 6.4 Synthesis in Libero SoC

After setting the design root appropriately for the design, use the following steps to run the Synthesis:

1. Click **Synthesis** in the Libero SoC software. The Synthesis window appears displaying the Synplicity project.
2. Set Synplicity to use the Verilog 2001 standard if Verilog is used.
3. Click **Run** to run the Synthesis.

## 6.5 Place-and-Route in Libero SoC

After setting the design route appropriately for the design, and running Synthesis, click **Layout** in the Libero SoC software to invoke Designer. CoreSmartBERT does not require special place-and-route settings.