HB0792
Handbook
CoreLSM v2.0

Microsemi
Power Matters.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA
Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Fax: +1 (949) 215-4996
Email: sales.support@microsemi.com
www.microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) 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. Microsemi is headquartered in Aliso Viejo, California, and has approximately 4,800 employees globally. Learn more at www.microsemi.com.

50200792. 1.0 9/17

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.

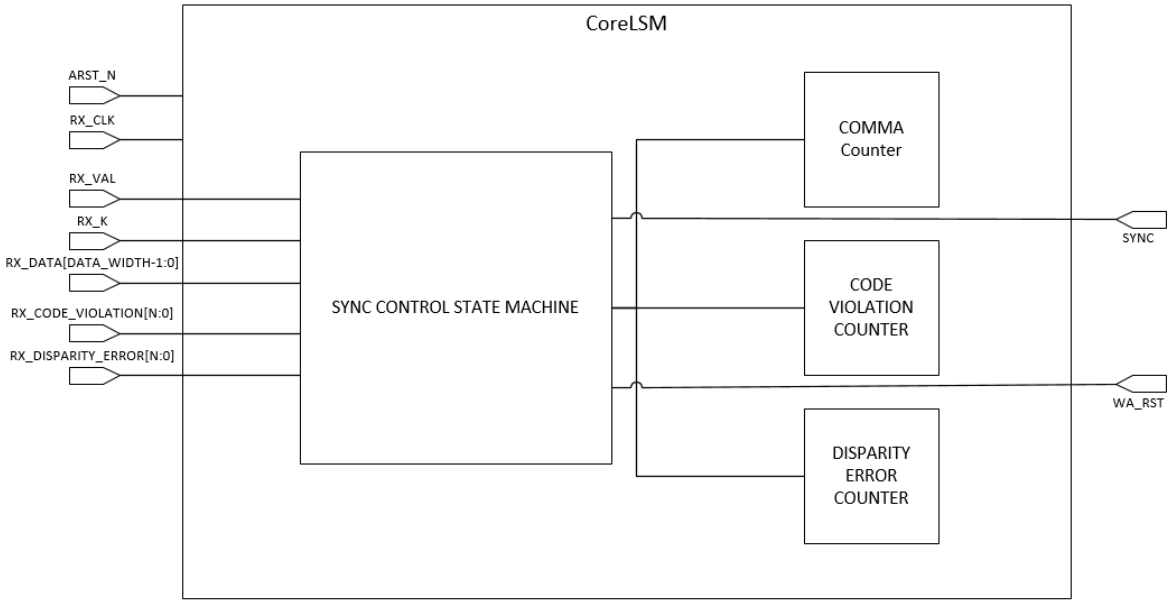
1.1 Revision 1.0

Revision 1.0 is the first publication of this document. Created for CoreLSM v2.0.

2 Introduction

CoreLSM identifies when a link is correctly word aligned with valid COMMAs and free from code violations and disparity errors on the PolarFire XCVR 8b10b. The CoreLSM block diagram is as shown in Figure 1, page 2.

Figure 1 • **CoreLSM Block Diagram**



Note: N = (DATA_WIDTH / 8) – 1

2.1 CoreLSM Blocks

CoreLSM consists of four major blocks, as described below. All signals are clocked using RX_CLK.

2.1.1 Sync Control State Machine

Determines when a link is correctly word aligned with valid COMMAs and free from code violations and disparity errors.

2.1.2 COMMA Counter

Keeps track of how many comma have been seen. When RX_VAL asserts any 8b10b error restarts the counter.

2.1.3 Code Violation Counter

Keeps track of how many code violation has been seen between COMMAs, when it reaches the number declared by CODE_VIOLATION_OUT_OF_SYNC it is out of sync.

2.1.4 Disparity Error Counter

Keeps track of how many disparity error has been seen between COMMAs. When it reaches the number declared by DISPARITY_ERROR_OUT_OF_SYNC, it is out of sync.

2.2 Key Features

- Identifies when a link is word aligned.
- Identifies when Code Violations and Disparity Errors occur and increments there respective counters.
- If the maximum number of Code Violations or Disparity Errors are not met within the 2^32 clock cycles then the counters are reset.

2.3 Core Version

This handbook is for CoreLSM version 2.0.

2.4 Supported Families

- PolarFire™

2.5 Utilization and Performance

CoreLSM has been implemented for the PolarFire device family. A summary of the implementation data for CoreLSM is listed in Table 1, page 3.

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

	Tiles			Utilization		
Family	**Sequential**	**Combinational**	**Total**	**Device**	**Total %**	**Performance MHz**
PolarFire	78	179	257	MPF300T	0.09	312.50 MHz

3 Design Description

3.1 Verilog / VHDL Parameters

CoreLSM has parameters (Verilog) or generics (VHDL) for configuring the register transfer level (RTL) code, described in Table 2, page 4. All parameters and generics are integer types.

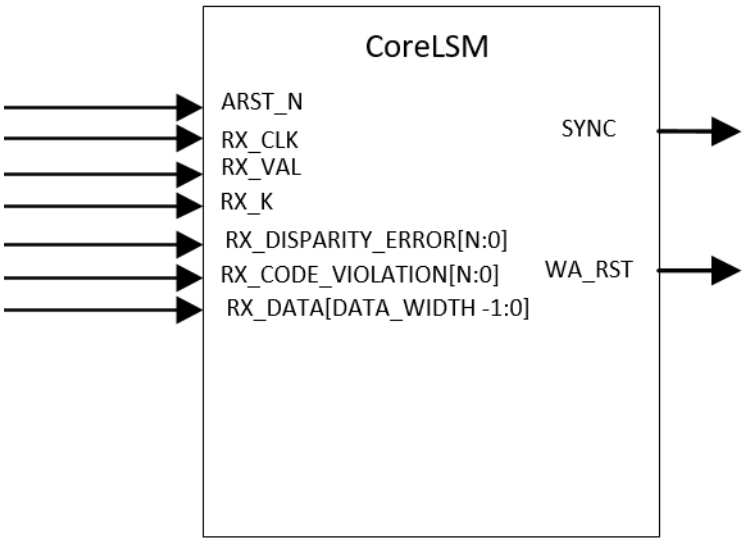
Table 2 • **CoreLSM Parameters/Generics Descriptions**

Parameter Name	Valid Range	Default	Description
FAMILY	26	26	Must be set to the required FPGA family: 26: PolarFire
DATA_DWIDTH	8, 16, 32, or 64	8	Receiver data width.
COMMA_IN_SYNC	1024, 1	3	Number of COMMAs to declare in SYNC (not back to back).
CODE_OUT_SYNC	1 to 1024	3	Number of Code Violations between COMMAs to declare out of SYNC state.
DIS_OUT_SYNC	1 to 1024	3	Number of Disparity Errors between COMMAs to declare out of SYNC state.
ENABLE_COMMA_K28_1	1 or 0	1	Enables or Disables the state machine looking for K28.1 as a valid COMMA character.
ENABLE_COMMA_K28_5	1 or 0	1	Enables or Disables the state machine looking for K28.5 as a valid COMMA character.
ENABLE_COMMA_K28_7	1 or 0	1	Enables or Disables the state machine looking for K28.7 as a valid COMMA character.

3.2 I/O Signals

The port signals for the CoreLSM macro are as shown in Figure 2, page 4 and defined in Table 3, page 5.

Figure 2 • **CoreLSM I/O Signal Diagram**



Note: N = (DATA_WIDTH / 8) – 1

Table 3 • **CoreLSM I/O Signal Descriptions**

Port Name	Type	Description
A_RST_N	Input	Active low asynchronous reset. This is used to reset the core to its initial state.
RX_CLK	Input	Recovered clock from XCVR.
RX_VAL	Input	Indicates XCVR receiver word aligner has found a COMMA.
RX_K	Input	Receive K indicator from XCVR.
RX_DISPARITY_ERROR[N:0]	Input	Receive disparity error.
RX_CODE_VIOLATION[N:0]	Input	Receive code violation.
RX_DATA[DATA_WIDTH -1:0]	Input	Receive data form XCVR.
SYNC	Output	1 – When in SYNC state 0 – When not in SYNC states
WA_RST	Output	Resets the word aligner reset to XCVR when maximum Disparity or Code Violations have been reached.

Note: All signals are active High (logic 1) unless otherwise noted.

Note: N = (DATA_WIDTH/8) -1

4 Tool Flow

4.1 Licensing

CoreLSM is license free.

4.1.1 RTL

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

4.2 SmartDesign

CoreLSM is preinstalled in the SmartDesign IP Deployment design environment.

The core should be configured using the configuration GUI within SmartDesign, as shown in Figure 3, page 6. For information on using the SmartDesign to instantiate and generate cores, see Libero SoC online help.

Figure 3 • **CoreLSM Full I/O View**

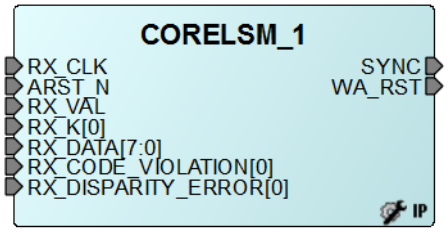
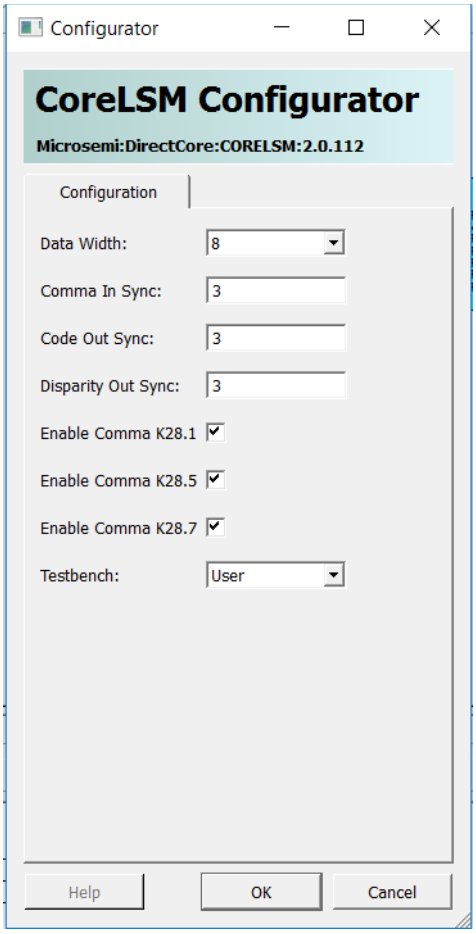


Figure 4, page 7 shows the options available in Configuration tab.

Figure 4 • **CoreLSM SmartDesign Configuration Window**



4.3 Simulation Flows

The user testbench for CoreLSM is included in all releases.

To run simulations, select the **User Testbench** flow within the **SmartDesign CoreLSM** configuration GUI, right-click the canvas, and select **Generate Design.**

When SmartDesign generates the design files, it will install the user testbench files.

To run the user testbench, Set the design root to the CoreLSM instantiation in the LiberoSoC design hierarchy pane and click the Simulation icon in the Libero SoC Design Flow window. This will invoke ModelSim® and automatically run the simulation.

4.4 Synthesis in Libero SoC

After setting the design root appropriately for your design, click the **Synthesis** icon in the Libero SoC. The Synthesis window appears, displaying the Synplicity® project. Set Synplicity to use the Verilog 2001 standard if Verilog is being used. To run Synthesis, click the **Run** icon.

4.5 Place-and-Route in Libero SoC

After setting the design root appropriately for the design, and after running Synthesis, click the **Layout** icon in the Libero SoC software to invoke Designer. CoreLSM requires no special place-and-route settings.