HB0831

CoreSDITX v2.2

Handbook

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 3.0

Updated for CoreSDITX v2.2.

1.2 Revision 2.0

Updated for CoreSDITX v2.1.

1.3 Revision 1.0

Revision 1.0 was the first publication of this document. Created for CoreSDITX v2.0.

2 References

SMPTE ST 259 -SDTV Digital Signal/Data - Serial Digital Interface

SMPTE ST 292-1 - 1.5 Gb/s Signal/Data Serial Interface

SMPTE ST 424 - 3 Gb/s Signal/Data Serial Interface

AMBA 3 APB Protocol Specification

UG0667 - Microsemi PolarFire FPGA User Guide

Contents

1 Revision History 2
1.1 Revision 3.0 2
1.2 Revision 2.0 2
1.3 Revision 1.0 2
2 References 3
3 Introduction 8
3.1 Overview 8
3.2 Features 8
3.3 Core Version 8
3.4 Supported Families 8
3.5 Device Utilization and Performance 9
4 Functional Description 10
4.1 SDI Frame Builder 10
4.1.1 EAV and SAV Packet Generation 11
4.1.2 Line number Packet Generation 11
4.1.3 SDI CRC Generation 11
4.1.4 SDI Frame Concatenation 11
4.2 SDI Scrambler and NRZI Encoder 11
4.3 Configuration and Status Module 11
5 Register Map 12
5.1 CFG_SDI 12
5.2 STAT_SDI 13
6 Interface 14
6.1 Configuration Parameters 14
6.2 Ports 14
7 Timing Diagrams 17
7.1 APB Slave Interface 17
7.2 Uncompressed Video Interface 17
8 Tool Flow 19
8.1 License 19
8.2 SmartDesign 19
8.3 Simulation in Libero 20
8.4 Synthesis in Libero 20
8.5 Place-and-Route in Libero 20

9 Testbench 21
9.1 User Test-bench 21
10 System Integration 23
10.1 CoreSDITX System Integration for 3G/HD-SDI mode 23
10.2 CoreSDITX System Integration for SD-SDI mode 24

List of Figures

Figure 1 CoreSDITX Functional Block Diagram 10
Figure 2 Uncompressed Video Interface Timing Diagram during Active Video Region in HD/3G-SDI mode 18
Figure 3 Uncompressed Video Interface Timing Diagram during Vertical Blanking Region in HD/3G-SDI mode 18
Figure 4 Uncompressed Video Interface Timing Diagram during Active Video Region in SD-SDI mode 18
Figure 5 Uncompressed Video Interface Timing Diagram during Vertical Blanking Region in SD-SDI mode 18
Figure 6 CoreSDITX Instance View 19
Figure 7 Configuring CoreSDITX in SmartDesign 20
Figure 8 CoreSDITX User Test-bench 21

List of Tables

Table 1 Device Utilization 9
Table 2 Register Map 12
Table 3 SDI Configuration Register 12
Table 4 SDI Configuration Register Bit Field Description 12
Table 5 SDI Status Register 13
Table 6 SDI Status Register Bit Field Description 13

3 Introduction

3.1 Overview

CoreSDITX DirectCore IP is a Serial Digital Interface (SDI) Framer. CoreSDITX supports Standard Definition SDI (SD-SDI) Level C, High Definition SDI (HD-SDI) and 3 Gigabits per second SDI (3G-SDI) Level A SDI standards defined by the Society of Motion Picture and Television Engineers (SMPTE).

3.2 Features

CoreSDITX supports the following features:

- Compliant with SMPTE 259 (SD-SDI) standard.
- Compliant with SMPTE 292 (HD-SDI) standard.
- Compliant with SMPTE 424 (3G-SDI) standard.
- Supports data rates 270 Mb/s and 270/1.001 Mb/s for SD-SDI Level C mode.
- Supports data rates 1.485 Gb/s and 1.485/1.001 Gb/s for HD-SDI mode.
- Supports data rates 2.97 Gb/s and 2.97/1.001 Gb/s for 3G-SDI Level A mode.
- Generates and inserts Start of Active Video (SAV) and End of Active Video (EAV) packets from timing reference information.
- Generates and inserts line number (LN) packets (in HD-SDI or 3G-SDI mode only).
- Generates and inserts Cyclic Redundancy Check (CRC) packets (in HD-SDI or 3G-SDI mode only).
- Performs Scrambling and Non-Return-to-Zero Inverted (NRZI) encoding.

3.3 Core Version

This handbook is for CoreSDITX version 2.2.

3.4 Supported Families

- PolarFire®

3.5 Device Utilization and Performance

Device utilization and performance data is provided in Table 1 for the supported device families. The data listed in this table is indicative only. The overall device utilization and performance of the core is system dependent.

Table 1 Device Utilization

Family (Device)	Logic Elements				Performance (in MHz)	
	Sequential (DFF)	Combinatorial (4LUT)	Total	Percentage	TX_CLK Frequency	PCLK Frequency
PolarFire (MPF300T_ES)	952	1030	1982	0.33	188	305

Note: The data in the Table 1 is achieved using typical synthesis and layout settings. Frequency (in MHz) was set to 100 and speed grade was Standard.

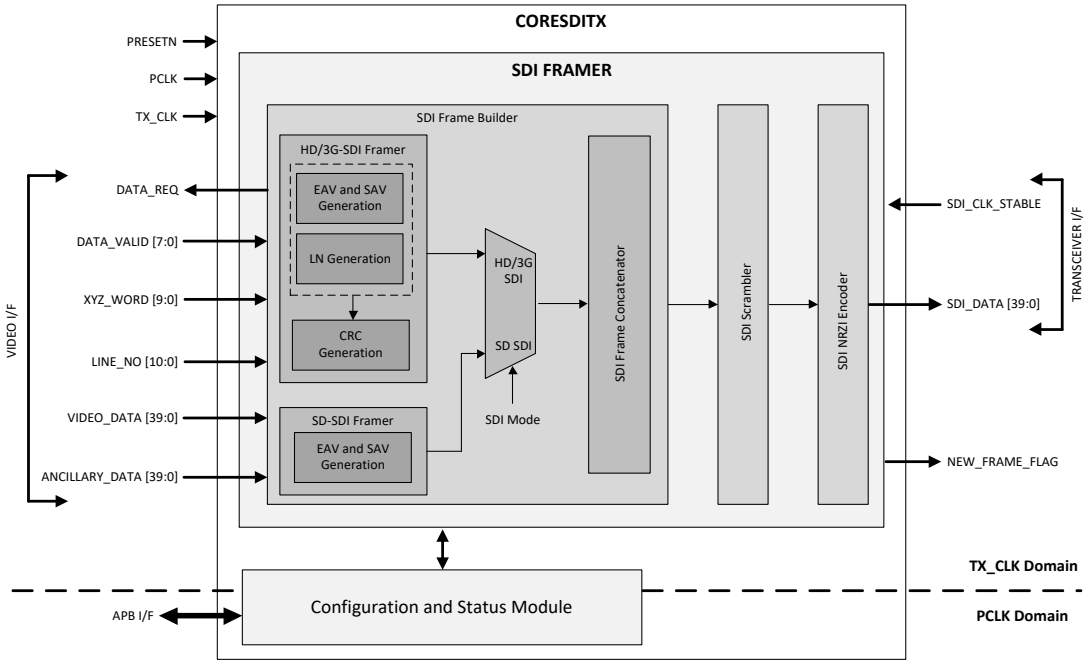
4 Functional Description

CoreSDITX is a SDI Framer. CoreSDITX performs the framing of the uncompressed or raw video data available on the uncompressed video interface into SDI data stream. The framing of the data is performed as per the SDI protocol and for the mode being configured.

Figure 1 shows the Functional block diagram of CoreSDITX. The major blocks are SDI Frame Builder, SDI CRC Generator, SDI Frame Concatenator, SDI Scrambler and SDI NRZI Encoder. The core has three interfaces, which are: Uncompressed Video Interface, Transceiver Interface, and APB Slave Interface.

For more information on the interfaces, refer to Interface section.

Figure 1 CoreSDITX Functional Block Diagram



4.1 SDI Frame Builder

SDI Frame Builder block accepts the video data (active video data or ancillary data) and control signals (XYZ word data and line number data) from the uncompressed video interface and frames the video data into a SDI data stream as per the SMPTE SD/HD/3G-SDI protocol specification. This block performs

- Generation of End of Active Video (EAV) and Start of Active Video (SAV) packets
- Generation of line number packets (in HD-SDI or 3G-SDI mode only)
- Generation of CRC packets (in HD-SDI or 3G-SDI mode only)
- Insertion of the EAV / SAV packets, Line number packets and CRC packets at appropriate places of the SDI data stream based on the selected SDI mode. The Line number packet and CRC packets are not inserted in the SDI data stream when core is configured for SD-SDI mode.
- Indicates whenever a new frame flag is transmitted by asserting NEW_FRAME_FLAG for one cycle of transmit clock

4.1.1 EAV and SAV Packet Generation

SDI Frame Builder block generates the EAV and SAV packets to be inserted into the SDI data stream from the XYZ word data input as defined in SMPTE SD/HD/3G-SDI specification. The core inserts the header ($3FF_h$, 000_h, 000_h) words along with the 10-bit XYZ word input to form EAV or SAV packet. EAV and SAV packets are generated for both Color-difference channel and Luma channel data.

4.1.2 Line number Packet Generation

The SDI Frame Builder block encodes the line number data input into line number packets LN to be inserted into the SDI data stream as defined in SMPTE HD/3G-SDI specification. The line number packets are generated for Color-difference channel and Luma channel data.

4.1.3 SDI CRC Generation

SDI CRC generator block computes the 18-bit CRC. The CRC computation is performed for every line of the video data for active video region and along with the EAV and LN packet data. The 18-bit CRC generated is encoded to form CRC packets to be inserted into the SDI data stream as defined in SMPTE HD/3G-SDI specification. Core computes CRC for Color-difference channel and Luma channel data. CRC computation is as per CRC polynomial specified in the SMPTE HD/3G-SDI specification.

$$CRC(X) = X^{18} + X^5 + X^4 + 1$$

4.1.4 SDI Frame Concatenation

SDI Frame Concatenator block inserts the EAV or SAV packets, LN packets and CRC packets in the appropriate place to form a SDI data stream as specified in SMPTE SD/HD/3G-SDI specification.

4.2 SDI Scrambler and NRZI Encoder

SDI Scrambler block performs the scrambling on the SDI data stream provided by the SDI Frame Builder block. The scrambling is performed as per the scrambling polynomial specified in the SMPTE SD/HD/3G-SDI specification.

$$G_1(X) = X^9 + X^4 + 1$$

The NRZI encoding is performed by SDI NRZI Encoder on the scrambled data. NRZI encoding is performed as per the polynomial specified in the SMPTE SD/HD/3G-SDI specification.

$$G_2(X) = X + 1$$

The output of this block is mapped on to the SDI_DATA output port.

4.3 Configuration and Status Module

This block configures the core for selected configuration based on the data written into configuration register. This block allows reading the status of the core. The configuration register and status registers are accessible through the APB interface. For more information on the registers refer Register Map section.

5 Register Map

The table lists registers available in CoreSDITX. These registers can be accessed through APB Interface.

Table 2 Register Map

Address	Register Name	Type	Width	Reset value	Description
0x00	CFG_SDI	R/W	32	0x00000082	SDI Configuration Register.
0x10	STAT_SDI	R	32	0x00000000	SDI Status Register. Reports back the actual SDI configuration for which the core is currently working.
0x14	SDI_FRM_CNT	R	32	0x00000000	SDI frame count register. Counts the number of times a SDI frames was put on the Transceiver data line for transmission. This is a roll over count.

5.1 CFG_SDI

Register to configure the core for required mode color and Chroma subsampling.

Table 3 SDI Configuration Register

Address	Register Name	Type	Width	Reset value	Description
0x00	CFG_SDI	R/W	32	0x00000082	SDI Configuration Register

Table 4 SDI Configuration Register Bit Field Description

Bit	Name	Type	Reset Value	Description
31:25	Reserved	-	-	-
24	VALID	R/W	1	Valid bit. Active High. Indicates the configuration data is valid
23:10	Reserved	-	-	-
9:6	MODE	R/W	0010	**SDI Mode Type** 0010: HD-SDI 0000: SD-SDI - Level C 0101: 3G-SDI - Level A Other values are reserved.
5:3	Reserved	-	-	-
2:0	COLOR	R/W	010	**SDI Color Type** Value / Color Coding / Chroma Subsampling 000 / RGB / 4:4:4 001 / YCbCr / 4:4:4 010 / YCbCr / 4:2:2 Other values are reserved.

5.2 STAT_SDI

Register to get status of the core configuration in which it is currently working.

Table 5 SDI Status Register

Address	Register Name	Type	Width	Reset Value	Description
0x10	STAT_SDI	R	32	0x00000000	SDI Status Register. Provides the status of the core configuration in which it is currently working.

Table 6 SDI Status Register Bit Field Description

Bit	Name	Type	Reset Value	Description
31:10	Reserved	R	-	-
9:6	MODE	R	0000	**SDI Mode Type** 0000: SD-SDI - Level C 0010: HD-SDI 0101: 3G-SDI - Level A Other values are reserved.
5:3	Reserved	-	-	-
2:0	COLOR	R	000	**SDI Color Type** Value / Color Coding / Chroma Subsampling 000 / RGB / 4:4:4 001 / YCbCr / 4:4:4 010 / YCbCr / 4:2:2 Other values are reserved.

6 Interface

CoreSDITX has following three interfaces:

Uncompressed Video Interface

A simple custom video data interface that accepts the uncompressed video data. The uncompressed video input to the CoreSDITX must be provided as described in the Timing Diagram section.

Transceiver Interface

The 40-bit SDI framed data from CoreSDITX is driven to the PolarFire Transceiver through this interface. The SDI framed data is provided to the Transceiver on a continuous basis at the rate Transceiver is configured to operate with a specific selected SDI mode SD/HD/3G-SDI.

APB Slave Interface

The APB interface provides access to the Configuration and Status registers of CoreSDITX.

6.1 Configuration Parameters

There are no configurable parameters / generics.

6.2 Ports

The port signals for CoreSDITX are described in the following table.

<table>
<tr><th>Port Name</th><th>Width</th><th>Type</th><th>Description</th></tr>
<tr><td colspan="4">Reset</td></tr>
<tr><td>PRESETN</td><td>1</td><td>Input</td><td>Active LOW asynchronous reset. Asynchronous assertion and synchronous de-assertion. This reset is synchronized and used inside the core with respect to each of the clock inputs.</td></tr>
<tr><td colspan="4">Clocks</td></tr>
<tr><td>PCLK</td><td>1</td><td>Input</td><td>APB system clock. All APB interface signals are clocked on the rising edge of this signal.</td></tr>
<tr><td>TX_CLK</td><td>1</td><td>Input</td><td>Transmit clock from Transceiver. All the Transceiver interface signals and uncompressed video interface signals are clocked on rising edge of this clock.
Recommended to connect LANEx_TX_CLK of Transceiver.
<table>
<tr><th>Mode</th><th>Data rate</th><th>TX_CLK frequency</th></tr>
<tr><td rowspan="2">3G-SDI Level A</td><td>2.97 Gb/s</td><td>74.25 MHz</td></tr>
<tr><td>2.97/1.001 Gb/s</td><td>74.25/1.001 MHz</td></tr>
<tr><td rowspan="2">HD-SDI</td><td>1.485 Gb/s</td><td>37.125 MHz</td></tr>
<tr><td>1.485/1.001 Gb/s</td><td>37.125/1.001 MHz</td></tr>
<tr><td rowspan="2">SD-SDI</td><td>270 Mb/s</td><td>6.75 MHz</td></tr>
<tr><td>270/1.001 Mb/s</td><td>6.75/1.001 MHz</td></tr>
</table></td></tr>
<tr><td colspan="4">TRANSCEIVER Interface</td></tr>
<tr><td>SDI_CLK_STABLE</td><td>1</td><td>Input</td><td>Transmit clock stable from Transceiver.
Recommended to connect LANEx_TX_CLK_STABLE of Transceiver.</td></tr>
</table>

Port Name	Width	Type	Description
SDI_DATA	40	Output	Transmit data to Transceiver. Recommended to connect LANEx_TX_DATA of Transceiver.
Uncompressed Video Interface			
DATA_REQ	1	Output	Data request signal. Core asserts this signal high whenever core is ready to accept data. Data must be available at all time when DATA_REQ is high.
DATA_VALID	8	Input	Data Valid signal. This signal indicates which of the uncompressed video interface signal is valid. In the given clock cycle, only one of the eight bits of this signal can be high in a given clock cycle. The bits are defined as follows and indicates: [7] VIDEO_DATA [39:0] is valid. Accompanies: VIDEO_DATA [6] Reserved [5] ANCILLARY_DATA is valid. Accompanies: ANCILLARY_DATA [4] Reserved [3] Reserved [2] Reserved [1] TRS (EAV / SAV) packet data. Accompanies: XYZ_WORD [0] Line number data is valid. Accompanies: LINE_NO **Note:** Tie DATA_VALID [0] bit to zero when core is configured in SD-SDI mode.
XYZ_WORD	10	Input	The XYZ_WORD input provides the framing information. DATA_VALID [1] must be asserted high to indicate the data provided on this port is valid. XYZ_WORD is used to generate Timing Reference Signals (TRS) - Start of Active Video (SAV) and End of Active Video (EAV) packets.
LINE_NO	11	Input	Current line number of the video frame. DATA_VALID [0] must be asserted high to indicate the line number data is valid. LINE_NO is used to generate LN0 and LN1 line number packets for HD-SDI and 3G-SDI (Level A). **Note:** Tie this input to zero when core is configured in SD-SDI mode.
ANCILLARY_DATA	40	Input	Ancillary data input. This is the ancillary data input which is to be transmitted. DATA_VALID [5] must be asserted high to indicate the ancillary data is valid. ANCILLARY_DATA [39:30] is the first 10 bit data word to be transmitted on the SDI data stream. ANCILLARY_DATA [29:20] is the second 10 bit data word to be transmitted on the SDI data stream. ANCILLARY_DATA [19:10] is the third 10 bit data word to be transmitted on the SDI data stream. ANCILLARY_DATA [9:0] is the last 10 bit data word to be transmitted on the SDI data stream.
VIDEO_DATA	40	Input	Active Video Input.

Port Name	Width	Type	Description
			This is the active video data input which is to be transmitted. DATA_VALID [7] must be asserted high to indicate the active video data is valid. Each 10 bit of the video data corresponds to C or Y channel data. The video data must be provided as indicated below: VIDEO_DATA [39:30] - **Cb** data (Blue difference chroma sample) VIDEO_DATA [29:20] - **Y0** data (Luma sample 0) VIDEO_DATA [19:10] - **Cr** data (Red difference chroma sample) VIDEO_DATA [9: 0] - **Y1** data (Luma sample 1)
FLAGS			
NEW_FRAME_FLAG	1	Output	SDI new frame flag. This flag goes high for one clock cycle of transmit clock indicating a new SDI frame is output from core.
APB Slave Interface			
PADDR	8	Input	APB address bus.
PSEL	1	Input	APB select signal. A HIGH indicates the slave device is selected and data transfer is required.
PENABLE	1	Input	APB enable signal. A HIGH indicates second and subsequent cycles of APB transfer.
PWRITE	1	Input	APB direction signal. Indicates APB write access when HIGH and APB read access when LOW.
PWDATA	32	Input	APB write data bus.
PREADY	1	Output	APB ready signal. A HIGH indicates slave is ready for APB data transfers.
PRDATA	32	Output	APB read data bus.
PSLVERR	1	Output	APB slave error signal. A HIGH indicates APB data transfer failure.

Note: x in LANEx can be 0,1,2, or 3

7 Timing Diagrams

7.1 APB Slave Interface

Refer AMBA APB document for APB I/F timing.

7.2 Uncompressed Video Interface

The uncompressed video data input to the CoreSDITX is input as per this protocol:

- Data to be transmitted must be available at all time when DATA_REQ signal is asserted high.
- Framing information is to be provided to the CoreSDITX through XYZ_WORD port indicating SAV or EAV and asserting the DATA_VALID [1] bit.
- The line number data must be provided to the core on LINE_NO port in the next clock cycle after TRS indicating EAV is provided on the XYZ_WORD port. DATA_VALID [0] bit must be asserted indicating valid line number data. Tie the LINE_NO and DATA_VALID [0] ports to zero in SD-SDI mode.
- The first 40-bit ancillary data of the horizontal ancillary data space (HANC) must be provided on ANCILLARY_DATA port in the next cycle after the line number data is provided on LINE_NO port. DATA_VALID [5] bit must be asserted indicating ancillary data is valid. After all the HANC is provided to the core on ANCILLARY_DATA port, TRS indicating SAV must be provided in the next cycle on XYZ_WORD port.

  Note: In case of SD-SDI mode, the first 40-bit ancillary data of the HANC must be provided on ANCILLARY_DATA port in the next cycle after TRS indicating EAV is provided on the XYZ_WORD port.
- The first 40-bit ancillary data of the vertical ancillary data space (VANC) must be provided on ANCILLARY_DATA port in the next clock cycle after the TRS indicating SAV is provided on XYZ_WORD port. DATA_VALID [5] bit must be asserted indicating ancillary data is valid. After all the VANC is provided to the core on ANCILLARY_DATA port, TRS indicating EAV must be provided in the next cycle on XYZ_WORD port.
- The first 40-bit active video data must be provided on VIDEO_DATA port in the next clock cycle after the TRS indicating SAV is provided on XYZ_WORD port. DATA_VALID [7] bit must be asserted indicating active video data is valid. After all active video is provided to the core on VIDEO_DATA port, TRS indicating EAV must be provided in next clock cycle on XYZ_WORD port.

Figure 2 and Figure 3 provides the uncompressed video interface timing diagrams during the active video region and vertical blanking region in HD/3G-SDI mode. These timing diagrams are for a 4:2:2 YCbCr video with the Raster resolution: 2200 X 1125 and active video resolution: 1920 X 1080.

Figure 4 and Figure 5 provides the uncompressed video interface timing diagrams during the active video region and vertical blanking region in SD-SDI mode. These timing diagrams are for a 4:2:2 YCbCr video with the Raster resolution: 1728 X 625 and active video resolution: 1440 X 576.

Note: A0, A2 …represent 40-bit ancillary data in VANC. H0, H2… represent 40-bit ancillary data in HANC. V0, V2… represent 40-bit active video data.

Figure 2 Uncompressed Video Interface Timing Diagram during Active Video Region in HD/3G-SDI mode

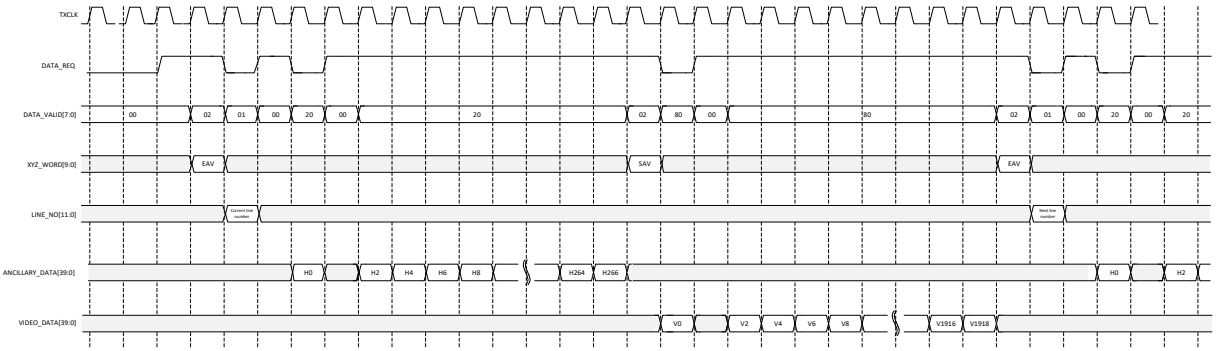


Figure 3 Uncompressed Video Interface Timing Diagram during Vertical Blanking Region in HD/3G-SDI mode

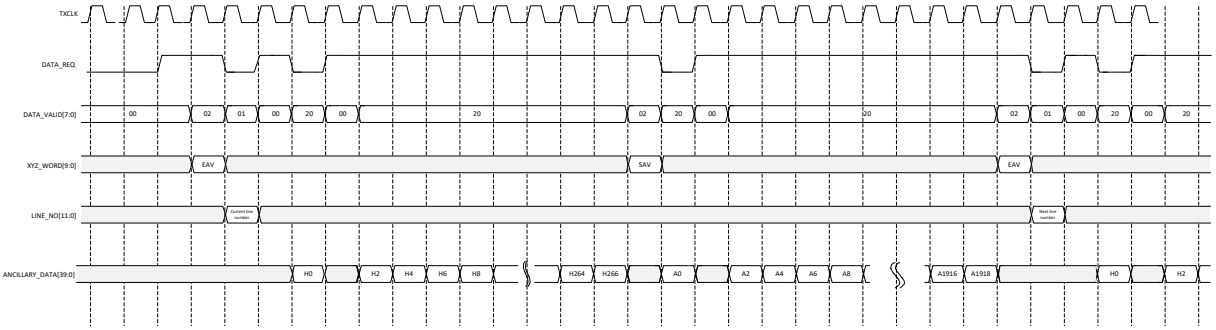


Figure 4 Uncompressed Video Interface Timing Diagram during Active Video Region in SD-SDI mode

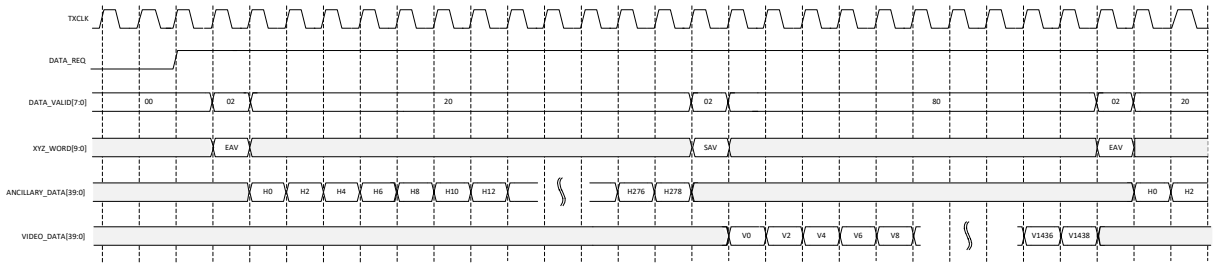
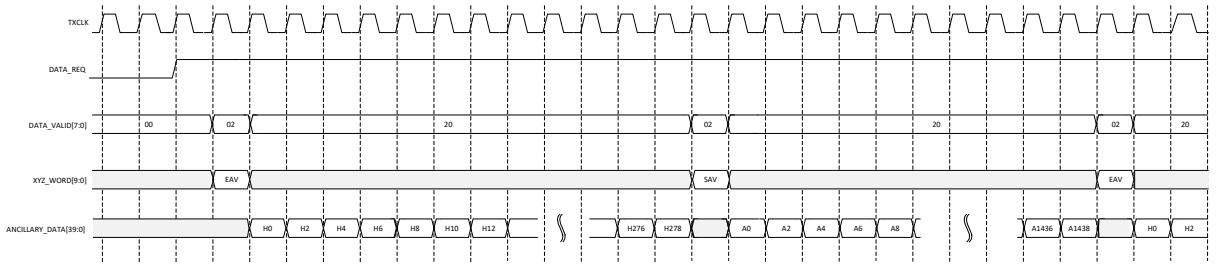


Figure 5 Uncompressed Video Interface Timing Diagram during Vertical Blanking Region in SD-SDI mode



8 Tool Flow

8.1 License

CoreSDITX is available in two versions:

- Obfuscated
- Evaluation

The Evaluation version is freely available and supports four hours of functionality on silicon.

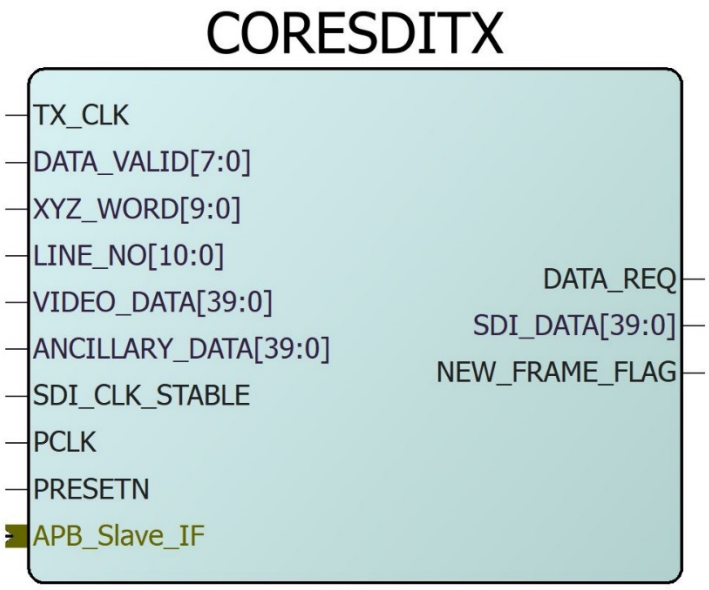
The Obfuscated version is license locked and will be available only with Libero Gold and Platinum Licenses.

8.2 SmartDesign

CoreSDITX is pre-installed in the Libero SmartDesign IP deployment design environment or downloaded from the online repository. Figure 6 shows an example instantiated.

Note: Unless specified otherwise, this document uses the name Libero to identify Libero SoC PolarFire.

Figure 6 CoreSDITX Instance View



The core can be configured using the configuration GUI within SmartDesign. An example of the GUI is shown in Figure 7.

Figure 7 Configuring CoreSDITX in SmartDesign

Note: For information on using SmartDesign to instantiate and generate cores, refer to Libero User Guide.

8.3 Simulation in Libero

User testbench is provide along with CoreSDITX.

To run user testbench simulations, select User Testbench flow in core configuration window. When SmartDesign generates the design files, it also generates the user testbench files. Set the design root to the CoreSDITX instantiation in the Libero design hierarchy pane and click Simulation in the Libero Design Flow window. This invokes ModelSim and automatically runs the user testbench simulation.

8.4 Synthesis in Libero

To run synthesis on the CoreSDITX, set the design root to the IP component instance and run the Synthesis tool from the Libero Design Flow pane.

8.5 Place-and-Route in Libero

After the design is synthesized, run the Place-And-Route tool from the Libero design flow pane.

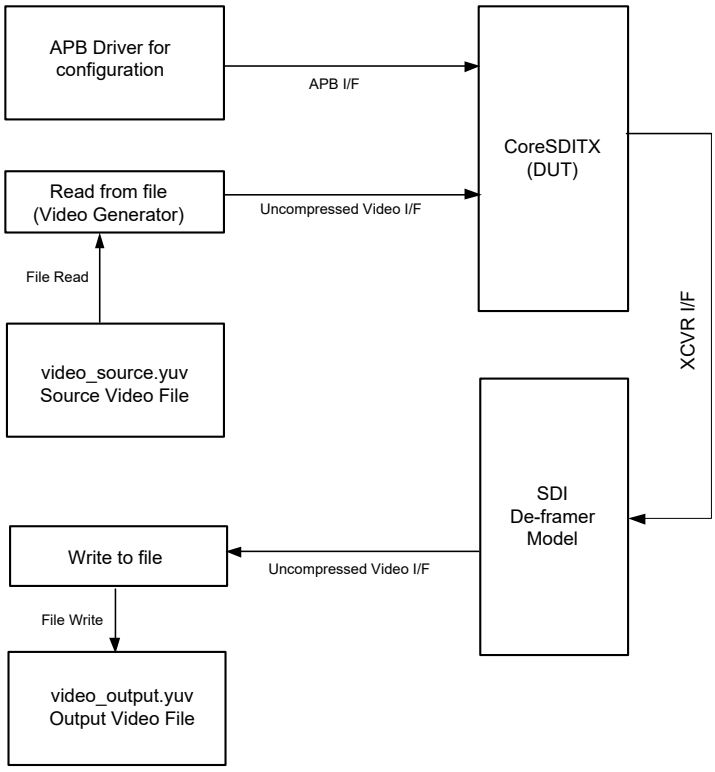
9 Testbench

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

9.1 User Test-bench

A simplified block diagram of the user testbench is as shown in Figure 8.

Figure 8 CoreSDITX User Test-bench



The source video file video_source.yuv is packaged in CoreSDITX CPZ and will be copied to simulation folder of Libero project when the core is instantiated and generated in Libero SmartDesign. The testbench has CoreSDITX (DUT), SDI Deframer model and video generator block.

The APB master model of the testbench writes to the Configuration registers of CoreSDITX (DUT) and configures the core for required SDI mode. The SDI Deframer model and the video generator are configured with the same configuration for which the DUT is configured. The video generator of the testbench reads the active video data from source video file and drives it on uncompressed video I/F of the DUT along with other required timing reference and data valid signals. The DUT generates the SDI frame from the uncompressed video. The DUT drives the framed video on to the Transceiver interface.

The user test bench has a SDI De-framer Model. The Transceiver I/F of DUT is connected to the Transceiver I/F of SDI De-framer model. SDI De-framer model generates uncompressed video from the framed video. The uncompressed video is written to video_output.yuv file.

The source and the output video files will be available in the simulation folder of the Libero project for comparison. The source and output video files with .yuv extension are in YUV422 8-bit colour format. These videos can be played on any commercial yuv video players.

In SD-SDI mode, the testbench generates color bar pattern and drives it on to the uncompressed video interface of the DUT. The SDI output of DUT is loop backed to SDI input of SDI Deframer Model. The output from SDI Deframer model is written to video_output.yuv file. The testbench generated pattern is written to video_output_vg_sd.yuv file.

In SD-SDI mode, the video_output_vg_sd.yuv and the video_output.yuv files will be available in the simulation folder of the Libero project and are in UYVY 8-bit color format. These videos can be played on any commercial yuv video players.

10 System Integration

This section provides hints to ease the integration of CoreSDITX.

10.1 CoreSDITX System Integration for 3G/HD-SDI mode

Refer **Figure 9** for 3G/HD-SDI mode System Integration.

In this example design:

- Contains CoreSDITX (SDI_Framer_0) and CoreSDIRX (SDI_Deframer_0) which is interfaced with MiV (MiV_Processor_0) soft processor and PF_XCVR (XCVR_SDI_0).
- Output pin “FABRIC_RESET_N” of CoreRESET_PF (RESET_PF_0) is used to drive MiV_Processor_0 reset pin “RESETN”. “FABRIC_RESET_N” and LANE0_TX_CLK_STABLE of XCVR_SDI_0 are used to drive PRESETN pin of both SDI_Framer_0 and SDI_Deframer_0.
- The SDI_Framer_0 has PCLK and TX_CLK clocks. SDI_Deframer_0 has PCLK and RX_CLK clocks.
- PCLK of both SDI_Framer_0 and SDI_Deframer_0 is an 80 MHz clock, driven from the output port “OUT0_FABCLK_0” of CCC_0.
- TX_CLK of SDI_Framer_0 is driven from LANE0_TX_CLK_R of XCVR_SDI_0 and RX_CLK of SDI_Deframer_0 is driven from LANE0_RX_CLK_R of XCVR_SDI_0.
- The LAN0_TX_CLK_R and LANE0_RX_CLK_R clock frequency depends on PF_XCVR configuration. 74.25 MHz for 3G mode and 37.125 MHz for HD mode. PF_XCVR can be configured through PFDRI_0.
- For 3G mode, PF_XCVR is configured with 2970 Mbps transceiver data rate, PMA mode with 40 bit @74.25 MHz. For HD mode, PF_XCVR is configured with 1485 Mbps transceiver data rate, PMA mode with 40 bit @37.125 MHz. In this example design, CDR reference clock frequency is 148.5 MHz.
- PF_XCVR_TXPLL_0 is configured with jitter cleaning mode.
- The SDI_Deframer_0 raw video data is looped back onto SDI_Framer_0 through CoreFIFO (FIFO_SDITX_RX_0).

10.2 CoreSDITX System Integration for SD-SDI mode

Refer **Figure 10** for SD-SDI mode System Integration.

In this example design:

- Contains CoreSDITX (SDI_Framer_0) and CoreSDIRX (SDI_Deframer_0) which is interfaced with MiV (MiV_Processor_0) soft processor and PF_XCVR (XCVR_top_0).
- Output pin “FABRIC_RESET_N” of CoreRESET_PF (RESET_PF_0) is used to drive MiV_Processor_0 reset pin “RESETN”.
- LANE0_RX_READY of XCVR_top_0, XCVR_INIT_DONE of DEV_INIT_0, PLL_LOCK_0 of CCC_Framer_0 and CCC_Deframer_0 are used to drive PRESETN pin of both SDI_Framer_0 and SDI_Deframer_0.
- The SDI_Framer_0 has PCLK and TX_CLK clocks. SDI_Deframer_0 has PCLK and RX_CLK clocks.
- PCLK of both SDI_Framer_0 and SDI_Deframer_0 is a 50 MHz clock, driven from the output port “OUT0_FABCLK_0” of CCC_Proc_0.
- RX_CLK of SDI_Deframer_0 is driven from “OUT0_FABCLK_0” of CCC_Deframer_0 which generates 6.75MHz and reference clock for the CCC_Deframer_0 is connected to LANE0_RX_CLK_R of XCVR_top_0.
- TX_CLK of SDI_Framer_0 is driven from “OUT0_FABCLK_0” of CCC_Framer_0 which generates 6.75MHz and reference clock for the CCC_Framer_0 is connected to LANE0_TX_CLK_R of XCVR_top_0.
- The LAN0_TX_CLK_R and LANE0_RX_CLK_R is working at 67.5 MHz for SD mode. Here the XCVR_top_0 is configured with 2700Mbps transceiver data rate, PMA mode with 40bit@ 67.5 MHz and the CDR mode is lock to reference. CDR reference clock frequency is 135 MHz. XCVR_top_0 is operated at 10 times oversampling rate.
- The TRANSITION_DETECTOR_0 and DOWNSAMPLER_0 modules are used to interface between XCVR_top_0 and SDI_Deframer_0. TRANSITION_DETECTOR_0 aligns the oversampled data from XCVR_top_0 to the bit boundary. DOWNSAMPLER_0 performs the 10x down sampling on the aligned data from TRANSITION_DETECTOR_0. The down-sampled data is provided to sdi input data of SDI_DEFRAMER_0.
- The UPSAMPLER_0 module is used to interface between SDI_Framer_0 and XCVR_top_0. UPSAMPLER_0 performs the 10x up-sampling on the sdi data output from SDI_Framer_0.
- The SDI_Deframer_0 raw video data is looped back onto SDI_Framer_0 through CoreFIFO (FIFO_SDI_RX_TO_TX_0).

Run the Libero flow with enabling the Timing Driven, High Effort Layout and Repair Minimum Delay Violations. The example design can be obtained from the Microsemi technical support team.

Figure 9 CoreSDITX System Integration for 3G/HD-SDI mode

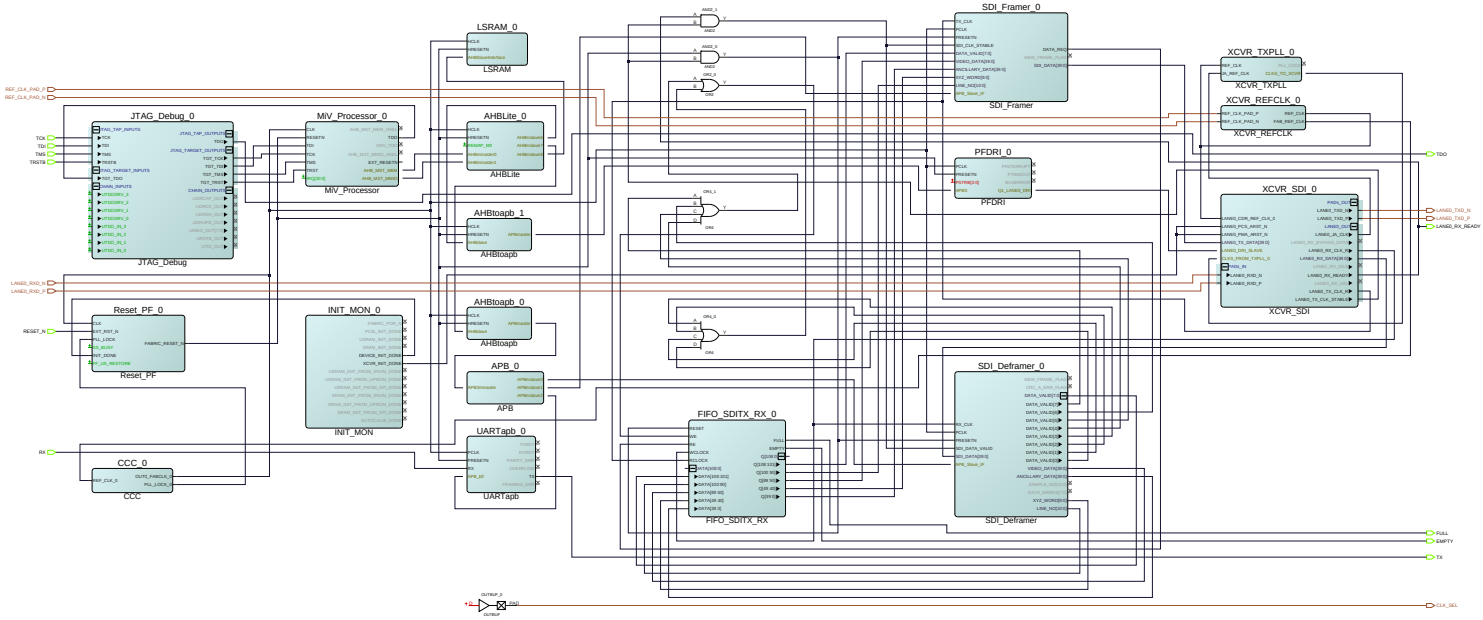
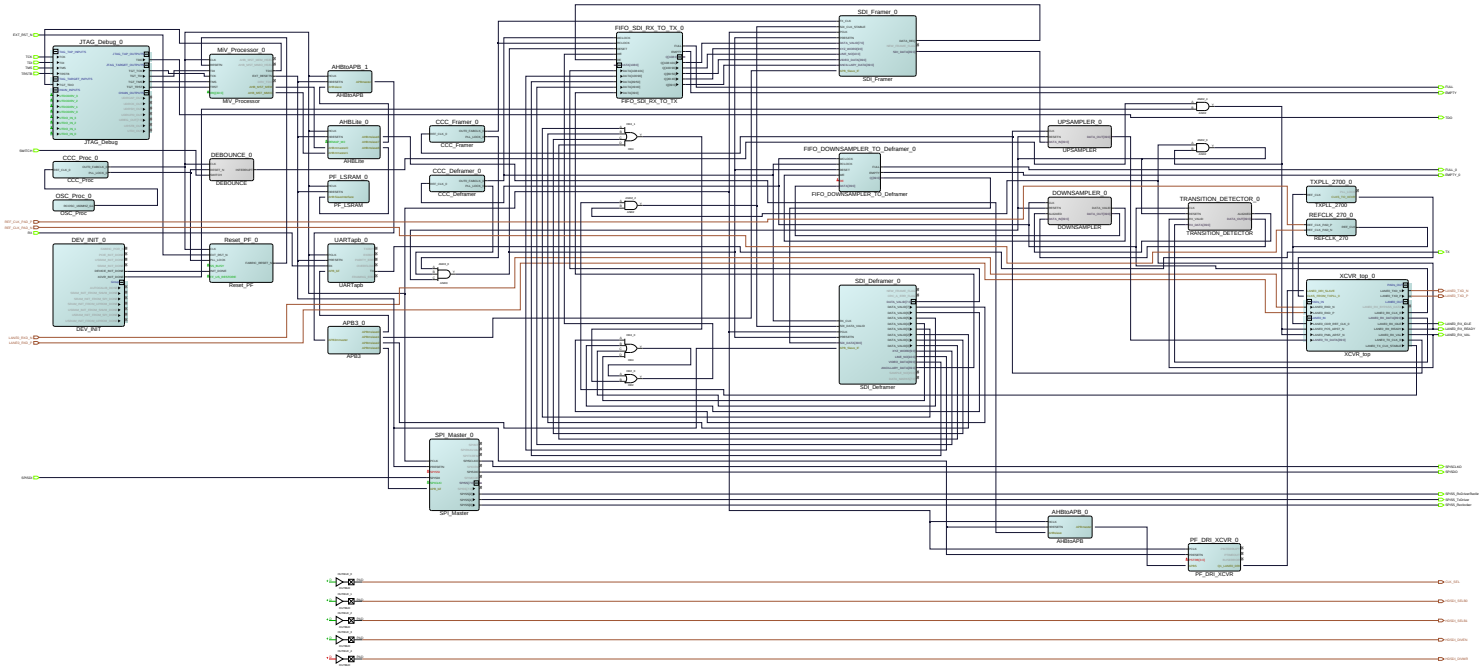


Figure 10 CoreSDITX System Integration for SD-SDI mode



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

©2018 Microsemi, a wholly owned subsidiary of Microchip Technology Inc. All rights reserved. Microsemi and the Microsemi logo are registered trademarks of Microsemi Corporation. All other trademarks and service marks are the property of their respective owners.

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.

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.