SMPTE-292 Scrambler and Descrambler/Framer

For SpartanII, Virtex and Virtex-E families

October 2, 2001 Product Specification

Andraka Consulting Group, Inc.

16 Arcadia Drive
North Kingstown, RI USA
Phone: +1 401-884-7930
Fax: +1 401-884-7950
URL: www.andraka.com
E-mail: info@andraka.com

Features

- Fully compatible with SMPTE specification for 292M Bit-Serial Digital Interface for High Definition Television Systems
- Designed for use with AMCC S8401/S8501 serializer and deserializer chipset
- Separate Macro blocks for transmit scrambler and receiver descrambler/framer
- Fully synchronous operation
- >75 MHz performance in all compatible devices
- Transmit macro accepts 20 bit parallel data (10 bit E_Y and 10 bit E_{Cb} E_{Cr}), then codes it using the scramble polynomial ($X^9 + X^4 + 1$) and NRZI (x+1) encoding. Outputs 20 bits parallel to AMCC S8401serializer on each cycle of transmit word clock.
- Receive macro accepts 20 bit de-serialized data from AMCC S8501 deserializer, reverses the NRZI coding and descrambles the data. Framing logic aligns the bits with the 20 bit parallel output (aligns 10 bit E_Y and 10 bit E_{Cb} E_{Cr})
- Both Macros are relatively placed to ensure a successful route and timing

Applications

The SMPTE 292 Scrambler and Descrambler/Framing Cores are used with the AMCC S8401/S8501 High Definition Serial Interface (HD-SCI) chipset. The chipset and these cores form the basic SMPTE 292M interface for HDTV applications.

General Description

The SMPTE292 core set, coupled with the AMCC S8401/S8501 serializer/deserializer chipset is fully compliant to the SMPTE 292M specification for Bit Serial Interfaces for High Definition Television Systems. The core set includes separate Xilinx Virtex/Spartan-II cores for transmitter coding (scrambling and NRZI) and receiver decoding (NRZ, descrambling, sync detect and word framing).

Functional Description

The CORE set is supplied as two cores, one for transmit and one for receive. The block diagrams are shown in Figure 1 in a system context. The cores may be purchased separately or as a set.

Core Facts

Core Specifics		
Device Families	Spartan II, Virtex, Virtex-E	
Slices Used	Transmit:	45 Slices in a 3W x 10 H CLB area
	Receive:	137 Slices in a 6W x 12 H CLB area
IOBs Used	Transmit:	none in macro (21 inputs, 20 outputs plus clock)
	Receive:	none in macro (20 inputs, 21 outputs plus clock)
System clock	System clock: 74.25 MHz (specified by SMPTE 292M)	
Device Features used	Carry logic, SRL16, relative placement	
Supported Devices		
Family	Minimum speed grade (receive)	Maximum clock for listed speed grade
Virtex	-4	101 MHz
SpartanII	-5	112 MHz
Virtex-E	-6	130 MHz
Minimum device size	Transmit only:	XCV50/XC2S30
	Receive only:	XCV50/XC2S30
	Both:	XCV50/XC2S30
Provided with Core		
Documentation	Core interface document	
Design file format	EDIF netlist VHDL source available extra	
Constraints	.UCF file with timing constraint. Placement information is embedded in design	
Schematic symbols	Foundation ISE	
Evaluation model	VHDL behavioral model	
Reference Designs and application notes	none	
Additional Items	none	
Design tools		
Xilinx Core tools	3.3i sp8	
Design verification		
Support		
Provided by Andraka Consulting Group		

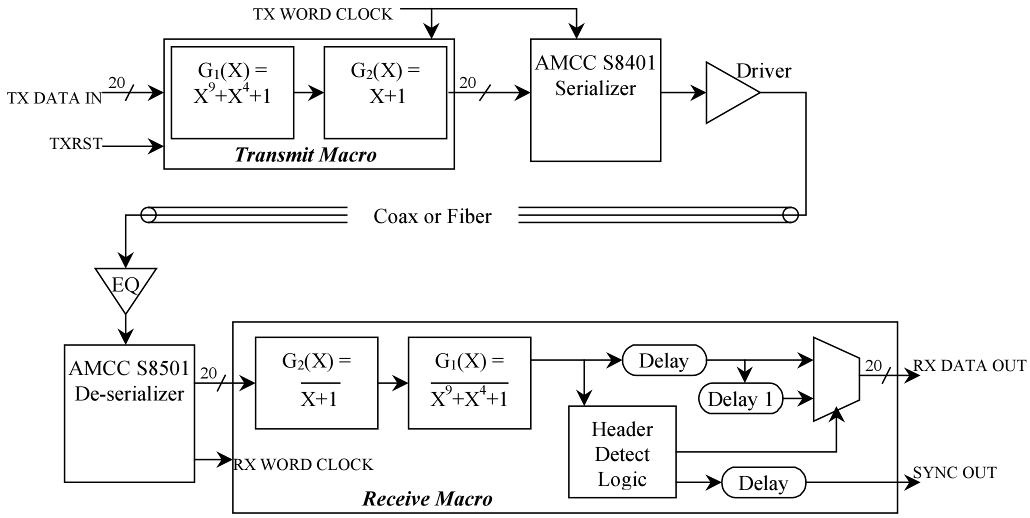

Figure 1. Transmit and Receiver block diagrams shown in system context

Scrambler (Transmit) Block

The transmitter block performs the X^9+X^4+1 scrambled channel coding, followed by the X+1 NRZI scrambling. Clock latency through the block is one clock cycle. The block performs scrambling on the 20 bit parallel data word. Figure 3 shows the equivalent bit serial scrambler circuit. This is the scrambler circuit that would be used if the unscrambled stream was first serialized then scrambled.

Inputs to the scrambler block are the 20 bit data, synchronous reset and the word rate clock. The 20 bit data input consists of the 10 bit E_Y channel and the separate10 bit E_{Cb} E_{Cr} channel. This bit assignment interleaves the data per the SMPTE 292 specification. The word rate clock should be 74.25 MHz, and is the same clock applied to the AMCC S8401 serializer's REFCLK input. The AMCC serializer has an internal phase lock loop that synthesizes the serial bit clock from this word clock. The synchronous reset forces the scrambler outputs to zero when the macro is clocked.

The macro output is 20 bit parallel scrambled data. The output should be connected to the AMCC serializer via registered outputs on the FPGA (OFD I/O macros), also clocked by the transmit word clock. Transmit data is clocked through the core by the rising edge of the transmit word clock.

Descrambler/Framer (Receive) Block

The receiver block descrambles the 20 bit parallel receive data using the complement of the scramble polynomial. Header sync detect logic parses the descrambled data for a frame header (EAV or SAV block) sync pattern consisting of string of 20 '1' bits followed immediately by 40 '0' bits (serial data is presented to the AMCC S8501 least significant bit first). The sync word is at an arbitrary alignment relative to the 20 bit parallel data word. When a header sync pattern is found, a rotator shifts the data to align the bits with the parallel output word. That alignment is maintained until the next header is detected. The header detect signal is also output for use as a frame sync signal. That frame sync is coincident with the first word of the output header sync pattern (the 1's word).

The example in Figure 2 shows the unscrambled data misaligned by 6 bits. The header detect logic recognizes the misaligned header sync sequence then selects the amount of shift required to align the data with the parallel output. That alignment is maintained until another header is detected. The output listing also shows the timing of the sync signal.

Unscrambled Din (bits 19 … 0)

11	1111	1111	11	11xx	xxxx
00	0000	0000	00	0011	1111
00	0000	0000	00	0000	0000
aa	aaaa	aaaa	aa	aa00	0000
bb	bbbb	bbbb	bb	bbbb	bbbb
cc	cccc	cccc	cc	cccc	cccc

Dout (bits 19 … 0)

						Sync
xx	xxxx	xxxx	xx	xxxx	xxxx	0
11	1111	1111	11	1111	1111	1
00	0000	0000	00	0000	0000	0
00	0000	0000	00	0000	0000	0
bb	bbbb	aaaa	aa	aaaa	aaaa	0
cc	cccc	bbbb	bb	bbbb	bbbb	0

Figure 2. Framing example

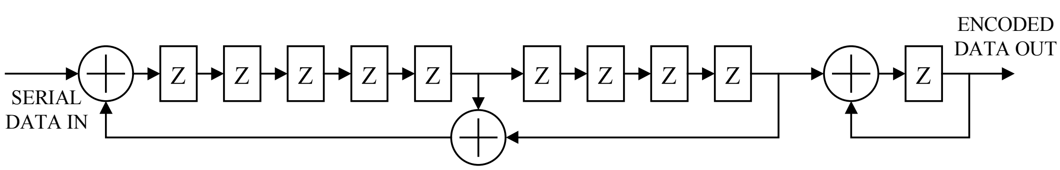

Figure 3. Equivalent serial scrambler circuit

The input to the scrambler should come directly from the AMCC S8501 via registered inputs to the FPGA (IFD macros). The descrambler macro does not include these IOBs to provide flexibility to the designer. Output from the receiver block is 20 bit data aligned to the 20 bit word. The user's logic should use the header sync output as a marker to aid in recovering the EAV/SAV frame timing. Latency through the receiver block is 10 cycles of the word clock (from input word containing MSB of aligned data). Data is always left shifted to obtain alignment. The core and the registered inputs should be clocked by the receive word clock generated by the AMCC S8501 deserializer. The core uses the rising edge of the clock input throughout.

Core Modifications

The core is provided as a black box relatively placed macro (RPM). Andraka Consulting Group, Inc can customize or retarget the core for additional cost. This includes adding or removing blocks, changing the scrambler polynomial, or integrating the macro into your design. The design source document (VHDL for Synplicity) is available at additional cost.

Pinout

Signal names for interfacing the cores are shown in figure 1 and described in tables 1 and 2. The core must be wired to FPGA I/O pads by the user.

Verification Methods

The FPGA core was verified through functional simulation and static timing analysis. Functional simulation included comparison of the scrambler and descrambler function to the bit serial model presented in this document, and thorough simulation of the framing logic. A system simulation was performed with the scrambler output serialized, subjected to a variable bit delay, de-serialized and passed trough the receiver. The results were compared to the input to the scrambler to check for data integrity. A functionally accurate VHDL behavioral model is supplied with the core, and is available at no charge upon request.

Table 1. Transmit macro interface signals

Signal	Signal Direction	Description
TCLK	Input	Transmit word clock: 74.25 MHz. Same as AMCC S8401 reference clock
TXRST	Input	Synchronous Reset: '1' at clock rising edge clears scrambler
Din[19:0]	Input	20 bit parallel data input. The 10 bit E_Y word is input to Din[19:10], The 10 bit E_{Cb} E_{Cr} word is input to Din[9:0]. Data is clocked in on rising edge of TCLK
Dout[19:0]	Output	Parallel output data to S8401. Bit indexes match indexes on S8401. Data changes on rising edge of TCLK

Ordering Information

This product is available directly from Andraka Consulting Group, Inc. Please contact them for further information, pricing, and functional models.

Recommended Design Experience

Users should be familiar with SMPTE 292 and related standards, as well as with standard Xilinx tool flows. The user should also be familiar with incorporating black box designs into his tool flow.

Related Information

ANSI/SMPTE 292M-1996 Standard
SMPTE
595 West Hartsdale Avenue
White Plains, NY 10607 USA
Phone: +1 914 761 1100
Fax: +1 914 761 3115
email: smpte@smpte.org
URL: www.smpte.org

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:
Xilinx Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com

For general Xilinx literature, contact:
Phone: +1 800-231-3386 (inside the US)
+1 408-879-5017 (outside the US)
E-mail: literature@xilinx.com

Table 2. Receive macro interface signals

Signal	Signal Direction	Description
RCLKN	Input	Receive word clock from S8501 RCLKN pin. Nominally 74.25 MHz.
Din[19:0]	Input	Parallel output data from S8501. Bit indexes match indexes on S8501. Data is clocked in on rising edge of RCLKN
Dout[19:0]	Output	20 bit parallel data output. The 10 bit E_Y word is output from Dout[19:10], The 10 bit E_{Cb} E_{Cr} word is output from Dout[9:0]. Data changes on rising edge of RCLKN
Sync	Output	Header Sync pulse. Logic '1' when 1's word of header sync is at Dout[19:0], Logic '0' otherwise. This signal should be used for frame synchronization by the user's logic.