WWW.ESISTREAM.COM

ESIstream

The **E**fficient **S**erial **I**nterface

UG-01 ESIstream SFP example design

Version 1.0

ESIstream
The Efficient Serial Interface

ESIstream is an Open high efficiency serial interface protocol based on 14b/16b encoding.

Its main benefits are low overhead and ease of hardware implementation.

TABLE OF CONTENT

FIGURE TABLE

Issue	Date	Comments
1.0	November 2016	Creation

You can contact contact@ESIstream.com if you have any question.

Scope

1.1 Objective

This user guide describes how to start working with the example design of the ESIstream IPs using an SFP interface on VC709 evaluation boards.

1.2 Related documents

ESIstream protocol: www.ESIstream.com
VC709 user guide: https://www.xilinx.com/support/documentation/boards_and_kits/vc709/ug887-vc709-eval-board-v7-fpga.pdf

1.3 Overview of the example design

This example design contains the sources for 2 projects: one for a serial interface transmitter (TX) and one for a serial interface receiver (RX); both use the ESIstream protocol. They were developed and tested on VC709 evaluation boards implementing a Virtex 7 FPGA from Xilinx (xc7vx690tffg1761-2).

In terms of software, this example design requires Vivado 2015.4; this allows simulating and modifying the projects. For compilation it requires a license for the xc7vx690tffg1761-2 which is included with VC709 board.
In terms of hardware (board testing), it requires 2 VC709; one for TX and one for RX. These boards would be connected through optic fiber included with the VC709 board. A frequency generator capable of generating a frequency of 187.5MHz along with SMA cables and baluns to provide a differential clock to each board is also necessary. Finally another SMA cable is needed and connects the 2 boards.

The interface is made by 4 serial links transmitting between boards through optical fiber (using the on-board SFP connectors) running at 6Gbps. Each board needs to have a differential reference clock at 187.5MHz supplied through SMA connectors. And a signal is transmitted between boards through SMA connector as well for synchronization. For more information on the set-up of the boards, refer to sections TX and RX.

Each folder contains all the sources necessary to simulate and test on board the example design (VHDL sources, simulation testbench, constraints, Xilinx ip xci files) and a tcl script to generate the project.

2. GENERATING THE PROJECTS

A tcl script (“script.tcl” in each project folder) is provided with the sources and can be used to generate the project on Vivado. It is recommended to use Vivado 2015.4 as updating the IP (in particular the GTH) to a more recent version may cause issue.

You can load a tcl script on Vivado through “tools > Run Tcl Script…”

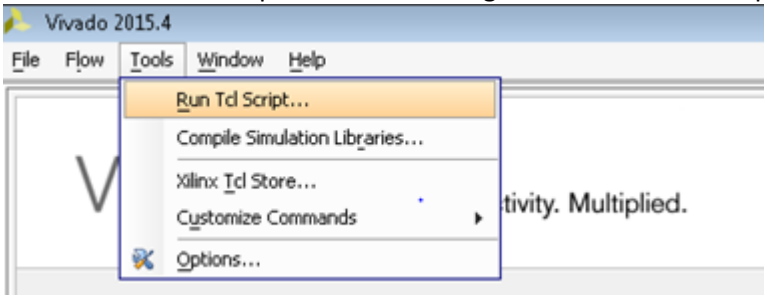

Figure 1: Running a tcl script with Vivado

When the project has been created and the IP generated, you can start compiling, simulating or modifying the example design like any Vivado project.
Note: If the TCL script is moved from its original folder it will not work as it uses relative path.

3. TX

3.1 TX Overview

The figure below shows the VC709 and highlights the parts used by the RX:

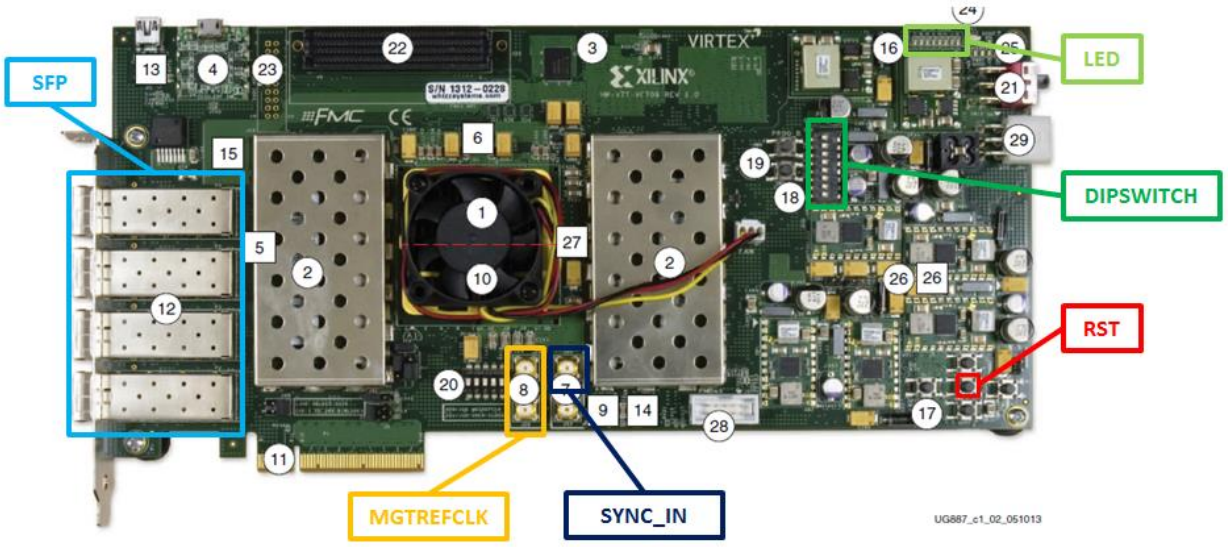

Figure 2: VC709 highlights for TX

MGTREFCLK: It corresponds to the reference clock of the transceiver. It is a differential signal that should be at 187.5MHz.

SFP: These are the 4 optical transmitters/receivers that transmit the serial data through optic fiber provided with the board. They should connect to the receiver board SFP.

DIPSWITCH: They are used for configuration. See section TX Configuration for the details.

RST: This pushbutton resets the FPGA.

SYNC_IN: This signal is the synchronization request from the RX used to synchronize the serial interface. It should be connected to the RX SYNC_OUT signal.

LED: They are the VC709 led. See section TX LED for the details.

3.2 TX Configuration

Dipswitch	Control	Comment
0-1	Data encoded	“00”: data encoded is “00000000000000” “01”: data encoded is 14bits positive ramp “10”: data encoded is 14bits negative ramp “11”: data encoded is “11111111111111”
2	Scrambling enabled	Should be set to ‘ON’. Can be useful for debug
3	Disparity enabled	Should be set to ‘ON’. Can be useful for debug

3.3 TX LED

LED	Output	Comment
0	Reset status	'0': reset pushbutton on '1': reset pushbutton off
1	SYNC status	'0': SYNC_IN is '1' '1': SYNC_IN is '0'
2	GTH PLL status	'0': unlocked '1': locked
3	GTH reset status	'0': reset in progress '1': reset done
4	SFP TX status	'0': Error from SFP connector '1': Optical transmission working
5	None	Always '0'
6	Dipswitch(0)	
7	Dipswitch(1)	

4. RX

4.1 RX Overview

The figure below shows the VC709 and highlights the parts used by the RX:

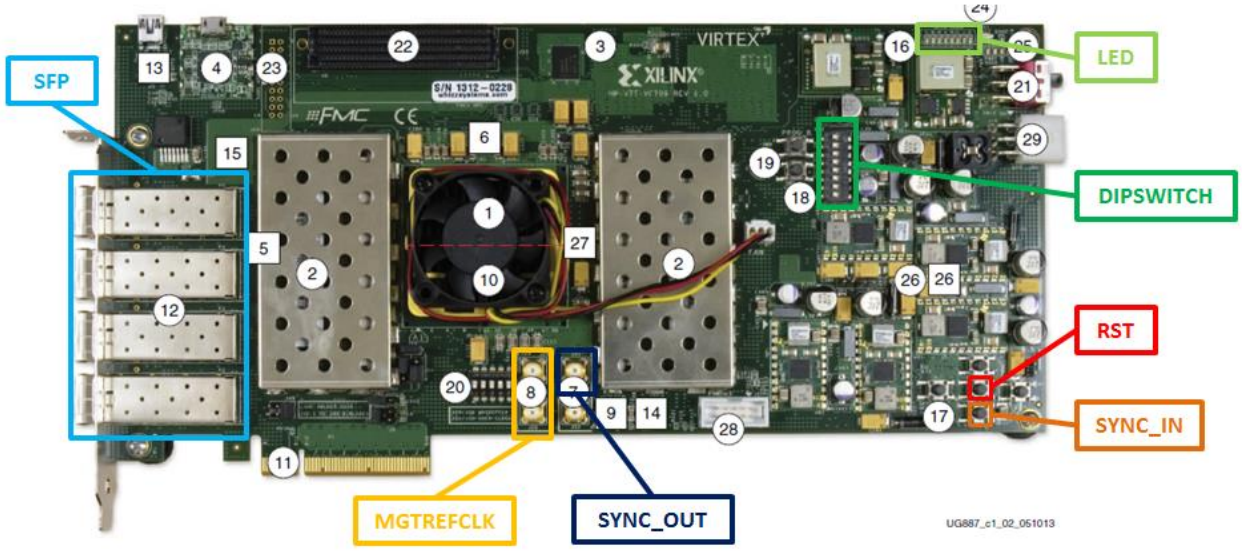

Figure 3: VC709 highlights for RX

MGTREFCLK: It corresponds to the reference clock of the transceiver. It is a differential signal that should be at 187.5MHz.

SFP: These are the 4 optical transmitters/receivers that receive the serial data through optic fiber provided with the board. They should connect to the transmitter board SFP.

DIPSWITCH: They are used for configuration. See section RX Configuration for the details.

RST: This pushbutton resets the FPGA.

SYNC_IN: This pushbutton request a synchronization between TX and RX.

SYNC_OUT: This signal is the synchronization request that is transmitted to the TX. It should be connected to the TX SYNC_IN signal.

LED: They are the VC709 led. See section RX LED for the details.

4.2 RX Configuration

Dipswitch	Control	Comment
0	Descrambling enabled	Should be set to 'ON'. Can be useful for debug
1-3	LED 4-7 output	See section RX LED
4-7	Lane enabled	Should be set to 'ON' to enable lane (i – 4)

4.3 RX LED

LED	Dipswitch(3 downto 1)	Output	Comment
0	X	Reset status	'0': reset pushbutton on '1': reset pushbutton off
1	X	SYNC status	'0': SYNC_IN pushbutton on '1': SYNC_IN pushbutton off
2	X	GTH status	'0': not ready '1': ready
3	X	Synchronization status	'0': synchronization not done or unsuccessful '1': synchronization done
4	"000"	None	Always '0'
	"001"	Lane 0 ramp+ status	'0': Decoded data on lane 0 is not a positive ramp '1': Decoded data on lane 0 is a positive ramp
	"010"	Lane 0 ramp- status	'0': Decoded data on lane 0 is not a negative ramp '1': Decoded data on lane 0 is a negative ramp
	"011"	None	Always '0'
5	"000"	None	Always '0'
	"001"	Lane 1 ramp+ status	'0': Decoded data on lane 1 is not a positive ramp '1': Decoded data on lane 1 is a positive ramp
	"010"	Lane 1 ramp- status	'0': Decoded data on lane 1 is not a negative ramp '1': Decoded data on lane 1 is a negative ramp
	"011"	None	Always '0'
6	"000"	All lane '0' status	'0': At least one lane has a decoded data different than "00000000000000" '1': Decoded data on all lane is "00000000000000"
	"001"	Lane 2 ramp+ status	'0': Decoded data on lane 2 is not a positive ramp '1': Decoded data on lane 2 is a positive ramp
	"010"	Lane 2 ramp- status	'0': Decoded data on lane 2 is not a negative ramp '1': Decoded data on lane 2 is a negative ramp
	"011"	Clk bit status	'0': Overhead clk bit is not as expected on at least one lane '1': Overhead clk bit is correctly seen on all lanes
7	"000"	All lane '1' status	'0': At least one lane has a decoded data different than "11111111111111" '1': Decoded data on all lane is "11111111111111"
	"001"	Lane 3 ramp+ status	'0': Decoded data on lane 3 is not a positive ramp '1': Decoded data on lane 3 is a positive ramp
	"010"	Lane 3 ramp- status	'0': Decoded data on lane 3 is not a negative ramp '1': Decoded data on lane 3 is a negative ramp
	"011"	Alignment status	'0': Overhead clk bit is not equal on all lanes '1': Overhead clk bit is equal on all lanes

5. INITIALIZATION OF THE TX-RX INTERFACE

The TX should be reset first, and then the RX should be reset. And then the RX SYNC_IN should be pressed. The configuration can be changed before or after reset and sync at the exception of the lanes standby on the RX after which the SYNC_IN should be pressed again.