Targeted for ACDS 21.4

Intel® Agilex® I-Series F-Tile Superlite IV Demo designs Supporting Documentation V1.1

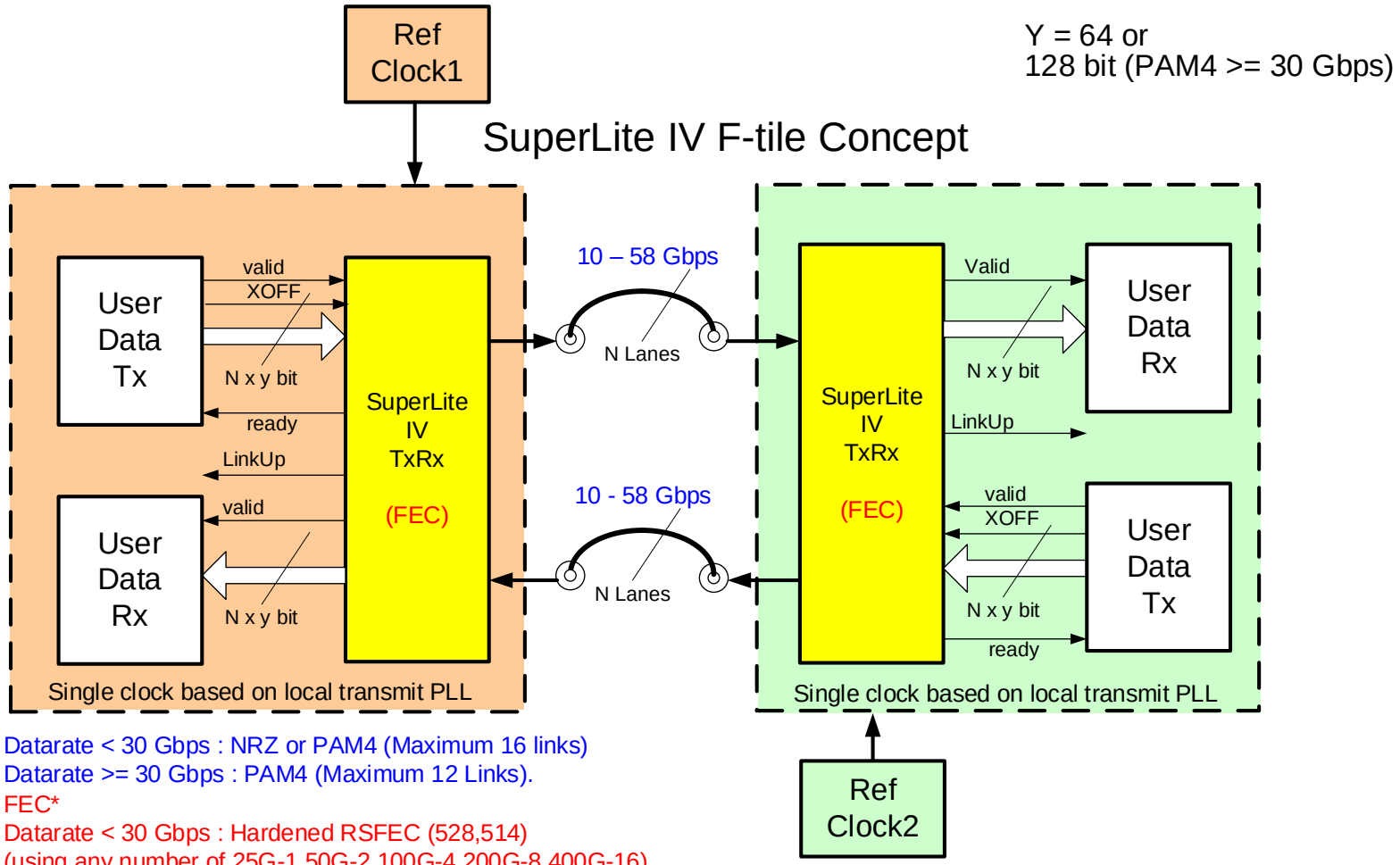
Introduction & Motivation

- This document serves as supporting documentation for the various F-tile Superlite IV demo designs.
- All F-tile demo designs follow a similar configuration and can have different number of channels, different clocking, using FEC or not, FGT or FHT etc. but the look and feel is always the same, both from RTL implementation as well from software implementation.
- The software is written in such a way that there is a single main.c file and supporting files that is being re-used by all designs (whether FEC is used or not) and the parameterization is done using the parameters.h file. Those software files are also maintained on the Intel Forum Agilex I-Series Demo page https://community.intel.com/t5/FPGA-Wiki/High-Speed-Transceiver-Demo-Designs-Agilex-I-Series-F-Tile/ta-p/1315123
- This allows to maintain a single set of source files to be used across all designs and only very little parameterization is required. E.g. to move from a design that is using 2 PHY's with 4 channels each with PAM4 without FEC to a design that is using 3 PHY's with FEC only very few lines need to be modified in the parameters.h file. Examples of those are also maintained in the software section on the Agilex I-Series Demo page.

Introduction & Motivation (continued)

- This document focusses on one variant of Superlite IV using F-tile.
- The version described here is using PHY direct IP with RSFEC configured in RS(544,514) aggregate mode (200GbE). The amount of 200GbE modules to be “combined” can be easily achieved and there is both a 2x200GbE (400Gbps aggregate) as well as a 3x200GbE version (600Gbps Aggregate).

‘Superlite IV’ Concept



Datarate < 30 Gbps : NRZ or PAM4 (Maximum 16 links)
Datarate >= 30 Gbps : PAM4 (Maximum 12 Links).
FEC*
Datarate < 30 Gbps : Hardened RSFEC (528,514)
(using any number of 25G-1,50G-2,100G-4,200G-8,400G-16)
Datarate >= 30 Gbps : Hardened KP-FEC (544,514)
(using any number of 50G-1,100G-2,200G-4 or 400G-8)

‘SuperLite IV’ Variants

- As of January 26th 2022 the following Superlite IV Variants exist
- Intel Agilex I-Series PCIe devkit :

Released	Xcvr	Variant	Aggr. BW.	Line Rate	Encoding	#Lanes	Clocking	Bonding	Internal Noise	Fec Type	QII Ve	Link to Design (when released)	Date Rel.
Pending	FGT	**Superlite IV**	400Gbps	53Gbps	PAM4	1x8	System PLL clocking	Bonded (2x200G aggregate)		(544,514)	21.4	agilex-pcie-devkit-superliteiv-ftile-fgt-2x200g-1x8-lanes.zip	26/01/2022
Pending	FGT	**Superlite IV**	600Gbps	53Gbps	PAM4	1x12	System PLL clocking	Bonded (3x200G aggregate)		(544,514)	21.4	agilex-pcie-devkit-superliteiv-ftile-fgt-3x200g-1x12-lanes.zip	18/01/2022
Pending	FGT	**Superlite IV**	400Gbps	53Gbps	PAM4	4x2	System PLL clocking	Bonded (100G aggregate)		(544,514)	21.4	agilex-pcie-devkit-superliteiv-ftile-fgt-1x100g-4x2-lanes.zip	26/01/2022

- Intel Agilex I-Series SI/SOC devkit :

Released	Xcvr	Variant	Aggr. BW.	Line Rate	Encoding	#Lanes	Clocking	Bonding	Internal Nois	Fec Type	QII Ve	Link to Design (when released)/Filename (when not released yet)
Pending	FGT	**Superlite IV**	1600Gbps	53Gbps	PAM4	4x8	System PLL clocking	Bonded (2x200G aggregate)		(544,514)	21.4	agilex-si-board-superliteiv-ftile-fgt-2x200g-4x8-lanes.zip
Pending	FGT	**Superlite IV**	2400Gbps	53Gbps	PAM4	4x12	System PLL clocking	Bonded (3x200G aggregate)		(544,514)	21.4	agilex-si-board-superliteiv-ftile-fgt-3x200g-4x12-lanes.zip

‘SuperLite IV’ Concept

- Transport data from point A to point B as simple as possible multiple bidirectional serial links using FEC.
- Superlite IV on F-tile makes use of the PHY direct IP which can be used with NRZ or PAM4 and which includes FEC. In case NRZ is used without FEC, Superlite II can be used instead (also available)
- This particular demo is using PAM4 and KPFEC.
- Superlite IV exchanges information between local and return side (handshaking) as well as allows to sent XON/XOFF control (backpressure)
- On the transmit side data will only be written in the FIFO when the valid signal is asserted, the fifo also can issue a backpressure signal (indicated by the ‘ready’ signal to indicate if data can be transmitted. This is equivalent to the Avalon Streaming interface.
- When XOFF is received from the remote side, all traffic will be halted on the local side (link remains running though).
- All the logic is clocked on one single clock (i.e. both transmit and receive core logic), this clock is the clock derived from the transmit PLL and is basically the line rate/128 for the PAM4 high datarate case.
- This implies that the combination of data valid with the locally generated clock from the local transmit PLL is frequency locked to the clock used at the remote side (which will typically vary in ppm). PPM difference is also measured inside the design.
- The Superlite IV implementation can accommodate various implementations of “copies” of the PHY direct IP. Any combination (up to the capability of the F-tile) of 50G-1,100G-2,200G-4 or 400G-8 can be used for the PAM4 implementations while for the NRZ version it can be any combination of 25G-1,50G-2,100G-4,200G-8 or 400G-16.
- This presentation focuses on the variant that is using multiple instances of the PHY direct configured in 200GbE Aggregate Mode.

‘Superlite IV’ Concept (continued)

- For aligning the lanes across the multiple PHY Direct Instances, idle characters and alignment characters will be sent at regular intervals (at the same pace as the generation of the mandatory Alignment marker insertion). The period to insert Alignment markers depends on the selected PHY Direct IP configuration (for 200Gbps Aggregate this is every 40K parallel clock cycles).
- The inserted alignment characters will be used to handle the deskew between the different FEC modules, the control words are being used to exchange link information from one side to the other.
- The Superlite IV TX module will automatically pause Tx user data if needed to insert idle and alignment characters and will also pause Tx user data based on the “ready” generated by the logic. So the userdata needs to be able to immediately stop sending traffic. If this is not possible one can add an additional FIFO with a slower running write clock.
- User data depends on configuration and is 128 bits per physical lane/64 bit per virtual lane.
- The amount of physical lanes used in this demo is 8 lanes per Superlite IV Instance.

F-Tile Superlite IV demo design Goals

- Use Intel Agilex I-Series PCIe Development kit as platform for demonstration.
  - Demonstrates the SuperliteIV concept where the 8 lanes of the 400Gb aggregate is connected to QSFPDD1
  - Uses dedicated txrx_pcs_64b66b_fgt_aggr module which instantiates all relevant additional logic required for RSFEC operation as the lower physical layer used by the Superlite IV protocol.
- User data width is 128-bits per physical lane. So for a 400Gbps implementation this means a user data width of 1024 bit. For the 600Gbps implementation (maximum amount achievable in one F-tile), the user datawidth is 1536 bits
- Actual throughput is measured in the design by measuring the data_valid received in combination with the clock. The net datarate is 399.8730 Gbps for the 2x200Gbps implementation (so slightly less than 400 Gbps, because of AM overhead and idle/alignment insertion) : 7 idle words (configurable) + 2 AM words + 4 idle offset words (configurable) every 40K gives a net rate of [(40960-(7+2+4))/40960] * 2* 200 Gbps = 399.8730 Gbps.
- Since RSFEC is used in the PHY Direct IP, system PLL clocking is used. And both Tx and Rx user interface are clocked by the System PLL/2 clock which is running at 53.125Gbps/128 = 415.0390625 Mhz. As it receives backpressures from the Superlite IV Tx module the data will be halted at specific times.
- To create the 1024 bit user data per link a combination of a counterpattern and prbspatterns are used. One 60-bit PRBS-23 generator for every virtual lane of 64-bit and a 16-bit counterpattern which is spread across the 16 virtual lanes (to detect the deskew is correctly handled).
- Depending on which FEC implementation is chosen, scrambling of the user data is either enabled or bypassed. For 200GbE (and 400GbE) scrambling is part of the FEC so in this case no scrambling is being used.
- Design uses a Qsys system with Nios® II Processor to control the PHY Direct IP's used in the Superlite IV design and control all other logic (like temperature measurement etc.)

Goals (continued)

- Supports serial loopback (note that if Serial loopback has been set it will remain set even after a reset or after restarting the software).
- Supports reverse parallel loopback
- Additional control signals required for the RSFEC operation are also generated in the design in module txrx_pcs_64b66b_fgt_aggr
  - tx_pcs_ready : low every 17th clock cycle (32/34 ratio)
  - am_pulse : 2 clock periods every 40960 parallel clock cycles (note that am pulse width and am pulse period is different for different configurations).
- Used 2 different Avalon Memory Mapped interfaces to each PHY direct IP : the transceiver reconfiguration interface and the transceiver datapath interface (so in this demo 4 Avalon Memory Mapped interfaces are used in total)
- Reports errorcount and calculates BER based on the received data.
- Extensive FEC statistics are provided for each virtual lane and physical lane (see screenshots further for information shown).
- Displays measured core temperatures.

Goals (continued)

- Uses LockAlarm signal : the LockAlarm signal will be asserted when the Locked will go down (due to external causes e.g. cable pull/plug in) and remain asserted until a reset errorcount or reset of the phy is being done. This allows to capture any events that could occur that could bring down a link for a period of time to keep track of this.
- Through NDME Toolkit support is automatically available
- Reads out and print out PMA settings for all channels
- Allows to find the optimum Transmit PMA settings (VOD, Pre-emphasis) using PMA sweep function.
- Option ‘c’ allows to provide updates on the selected channels in terms of BER and FEC statistics, this can be used over time to trace any events going on and can be further post-processed.
- Option ‘#’ builds a “FEC error tree” live on the screen.
- Allows to mute the FGT transmit (to mimic e.g. a cable pull)
- Contains several stress test functions including a dedicated function to measure the latency
- I2C support for the QSFPDD modules for the 4 F-tile Demo board. (not supported yet for the PCIe Devkit as the I2C control is handled by dedicated I2C controller)
- Includes Testbench for a single PHY configuration.

Future Enhancements

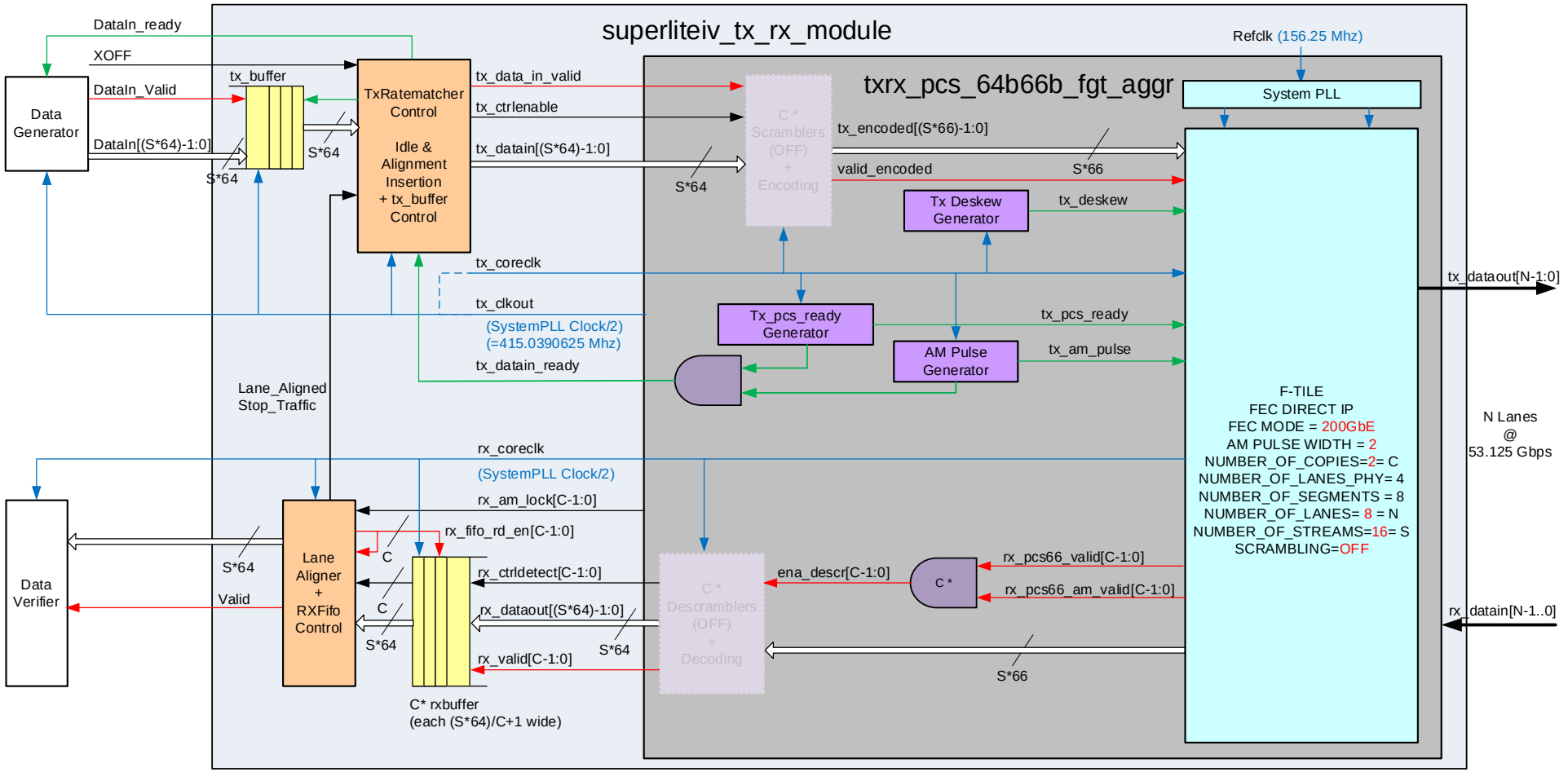
- Add F-tile temperature readout (requires production silicon)

‘Superlite IV’ Resources (1 Link of 2x200G)

- Superlite IV Ftile TxRx Module with PHY direct IP using 8 lanes at 53.125 Gbps and 2 200Gbps with KPFEC modules.
- Note : The Tx Ratematcher FIFO is implemented as M20K but could also be implemented using MLABs if preferred.
  - ~ 2400 ALUT’s
  - ~ 2600 ALM’s
  - ~ 5500 Dedicated logic registers
  - 52 M20K (for the Tx Ratematcher FIFO and Rx Buffers for deskew)

Instance	Entity	ALMs needed [=A-B+C]	ALMs used for memory	Combinational ALUTs	Dedicated Logic Registers	Block Memory Bits	M20Ks
superliteiv_txrx_module_inst	superliteiv_tx...	2613.7 (21.5)	640.0 (0.0)	2385 (38)	5523 (1022)	41024	52
Generate_RxBuffer[0].retimer_rx_valid2	hyper_pipe	0.3 (0.3)	0.0 (0.0)	0 (0)	1 (1)	0	0
Generate_RxBuffer[0].rxbuffer_inst	rxbuffer	23.3 (0.0)	0.0 (0.0)	47 (0)	34 (0)	16416	13
Generate_RxBuffer[1].retimer_rx_valid2	hyper_pipe						
Generate_RxBuffer[1].rxbuffer_inst	rxbuffer	23.5 (0.0)	0.0 (0.0)	46 (0)	34 (0)	16416	13
Reset_Synchro_inst1	reset_synchro	1.8 (1.8)	0.0 (0.0)	1 (1)	4 (4)	0	0
Reset_Synchro_inst2	reset_synchro	2.0 (2.0)	0.0 (0.0)	0 (0)	4 (4)	0	0
Rx_Path_inst	Rx_Path_Des...	306.1 (76.9)	0.0 (0.0)	108 (108)	1028 (51)	0	0
Tx_Ratematcher_Control_inst	Tx_Ratematc...	716.1 (697.3)	0.0 (0.0)	1187 (1160)	1348 (1279)	8192	26
txrx_pcs_64b66b_fgt_aggr_inst	txrx_pcs_64b...	1519.0 (36.5)	640.0 (0.0)	958 (65)	2048 (156)	0	0

Detailed Block Diagram of the Implementation (2x200G)



Tx Path (for one 2x200G Module)

- A datastream is generated at a clock of 415.039 Mhz @ 1024 bits when DataIn_ready is high. The datapattern in the demo is a combination of 16 60-bit Prbs-23 and one 16-bit counterpattern.
  - The 16-bit counterpattern is spread across the 16 virtual lanes, 1 bit per lane. This is to make sure any deskew on the lanes would be immediately spotted. The remaining 3 bits left of the 64 bit are tied to ground.
- Since idle +alignment characters and AM markers have to be inserted at regular intervals and because of the overclocking the pcs is generating periodic tx_pcs_ready pulses the data stream will be halted at regular intervals. This is indicated by the Superlite IV TX module by the DataIn_ready signal.
- Single biterrors can be inserted using inserterror input. As there is one PRBSGenerator per virtual lane, the biterror insert will produce 16 bit-errors at a time.
- This data is presented to the Superlite IV Tx Module
- The data is written into the TxFifo (single clocked fifo) using the 415.039 Mhz as clock and using the DataIn_Valid as write-enable.
- The read_enable is pulsed at the same pace as the write-enable.
- Once every 40K (for 200G/400G implementations) valid cycles idle characters + alignment characters + alignment markers will be inserted in the datastream. If no valid data is present idle characters will be sent.

Tx Path (continued)

- For 200G mode (and 400G as well) the data may not be scrambled as this is part of the FEC itself (so this reduces again the amount of required logic).
- The special control characters being used for idle and alignment are the following :
- In terms of idle character a value of “00….00BCFF” is being used (valid 802.3 control character). The alignment character is sent as the last control character. The lower 16 bits of this control word (at the alignment position) encodes the status of the link in addition to the alignment indication.
- If no local alignment has been found : the last control word sent is “..7CFF”
- If local wordalignment has been found : the last control word sent is “..FDFF”
- If XOFF is being sent to the remote side : the last control word sent is “..5CFF”

B7	B6	B5	B4	B3	B2	B1	B0
00	Latency Count Tx	00	00	7C	7C	FD/5C/7C	FF

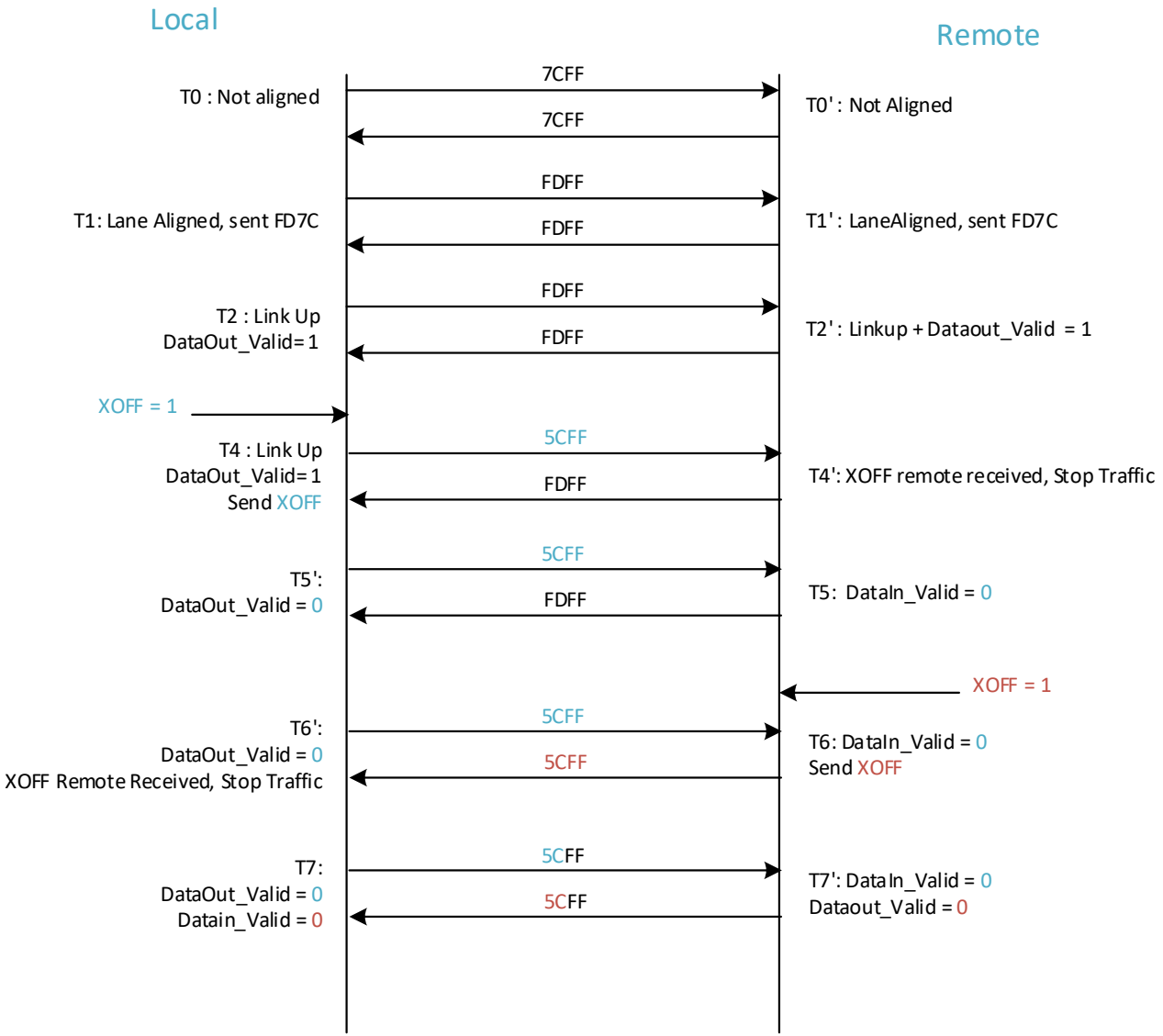
Rx Path (one 2x200G module)

- Each 200G PHY direct IP will perform all necessary wordalignment, decoding, deskew of the virtual lanes, RSFEC dedoding + correction and each PHY will present 512 bits of aligned data and control to the Rx path deskew module (in combination with a valid signal and received alignment markers (which can be discarded).
- Based on the position of the alignment word, the delay for each 200G block is measured, the statemachine inside the Rx_path_deskew module will use the information of that measured delay to control a special Rx FIFO signal “rx_fifo_rd_en” on all the RxBuffers’s instantiated externally to deskew the 200G modules.
- Once lane alignment is achieved, the received data in addition with a data valid signal will be sent to the data verifier.
- Here PRBS and counterverification will take place and single biterrors will be counted.
- Prbslocked is asserted when 128 consecutive valid Prbs-23 60-bit words have been received on each of the virtual lanes and de-asserted when 128 consecutive non-valid Prbs-23 60-bit words have been received.
- Countlocked is asserted when 128 consecutive valid 16-bit counter words have been received on the parts of the virtual lanes they were received.
- There are 16 60 bit PRBS Verifiers (one for every virtual lane) and 1 16-bit CountVerifier to validate the 1024 bits of data.

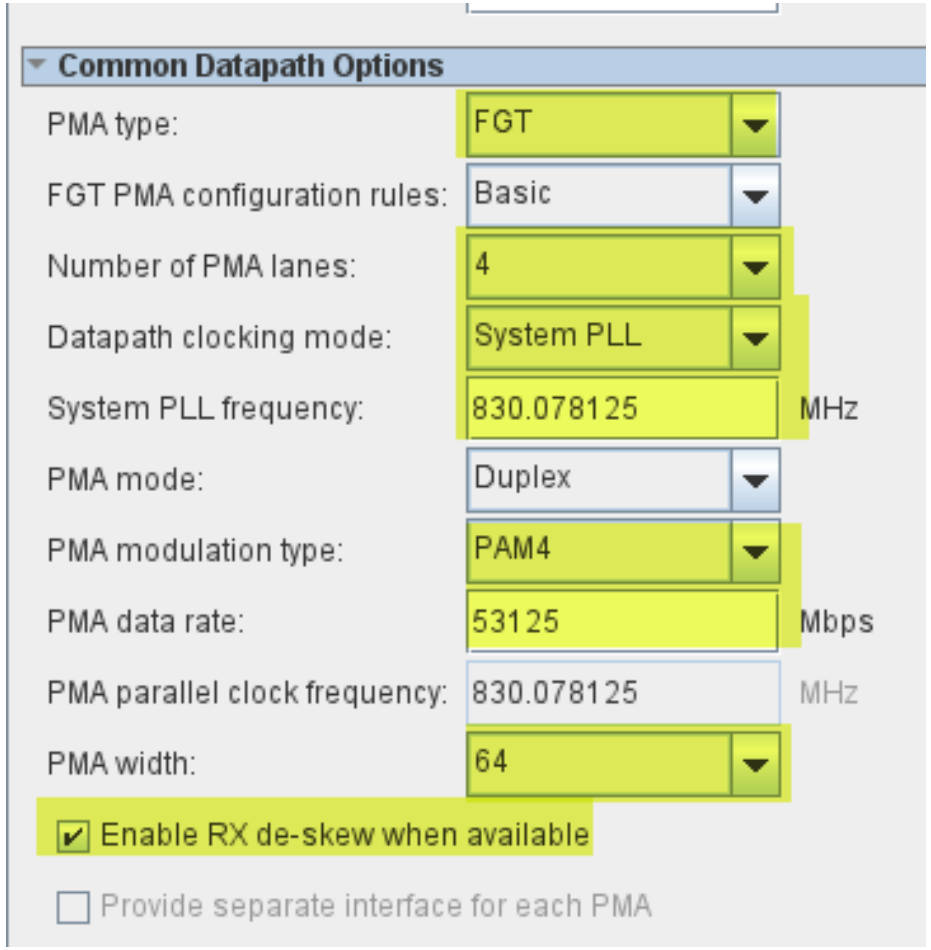
Rx Path (continued)

- The Rx_path_deskew machine also has the logic to determine the local Rx state which depends on the incoming “alignment” control word
- If last control word received is “…FDFF” it means the remote side has reached Wordalignment, if local side also has reached wordalignment this results in Linkup to be asserted, indicating that both sides are up and running. This can be used as the trigger to sent data across e.g.
- If last control word received is “…5CFF” it means an XOFF has been received from the remote side, this will be forwarded to the local transmitter to stop generating traffic on the local side.
- (See Flow chart on next slide for more details)

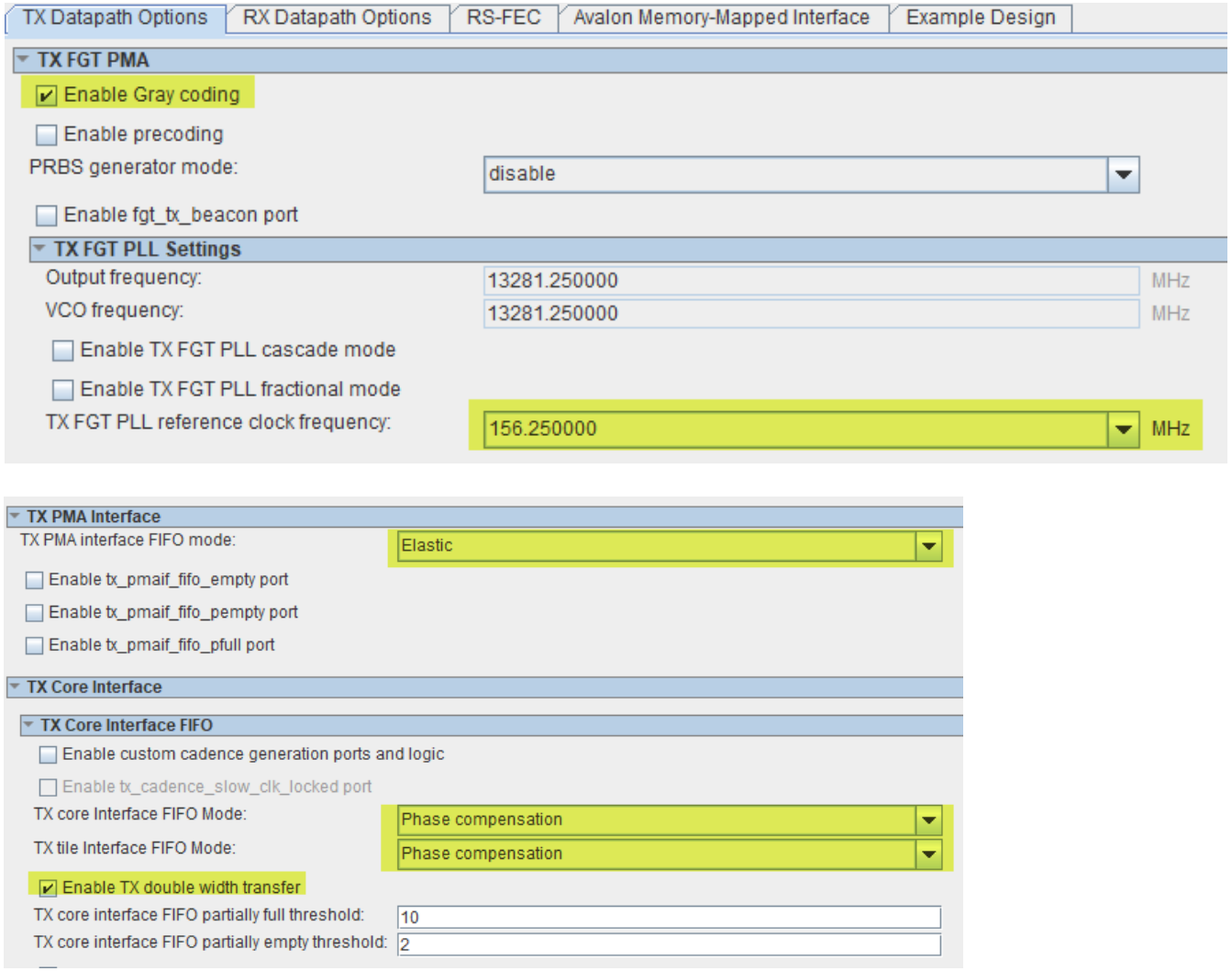
Control Characters used in combination with XOFF



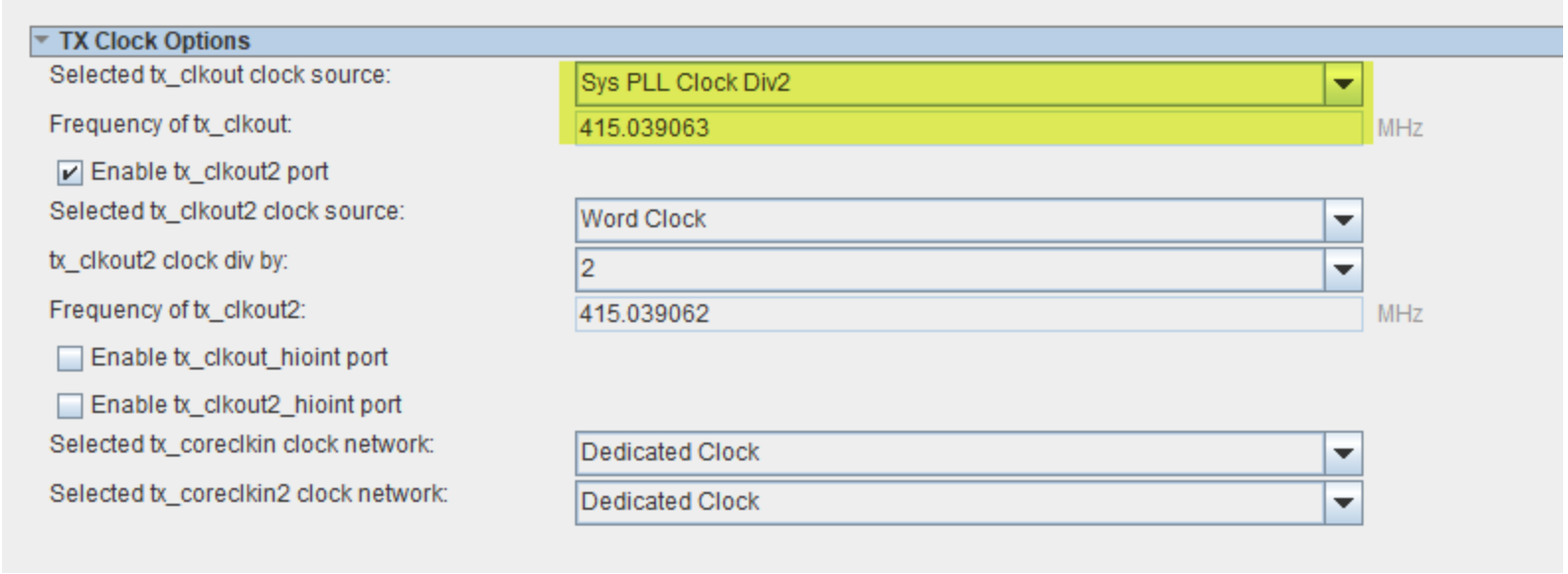
F-Tile Phy Direct IP (Common Datapath)



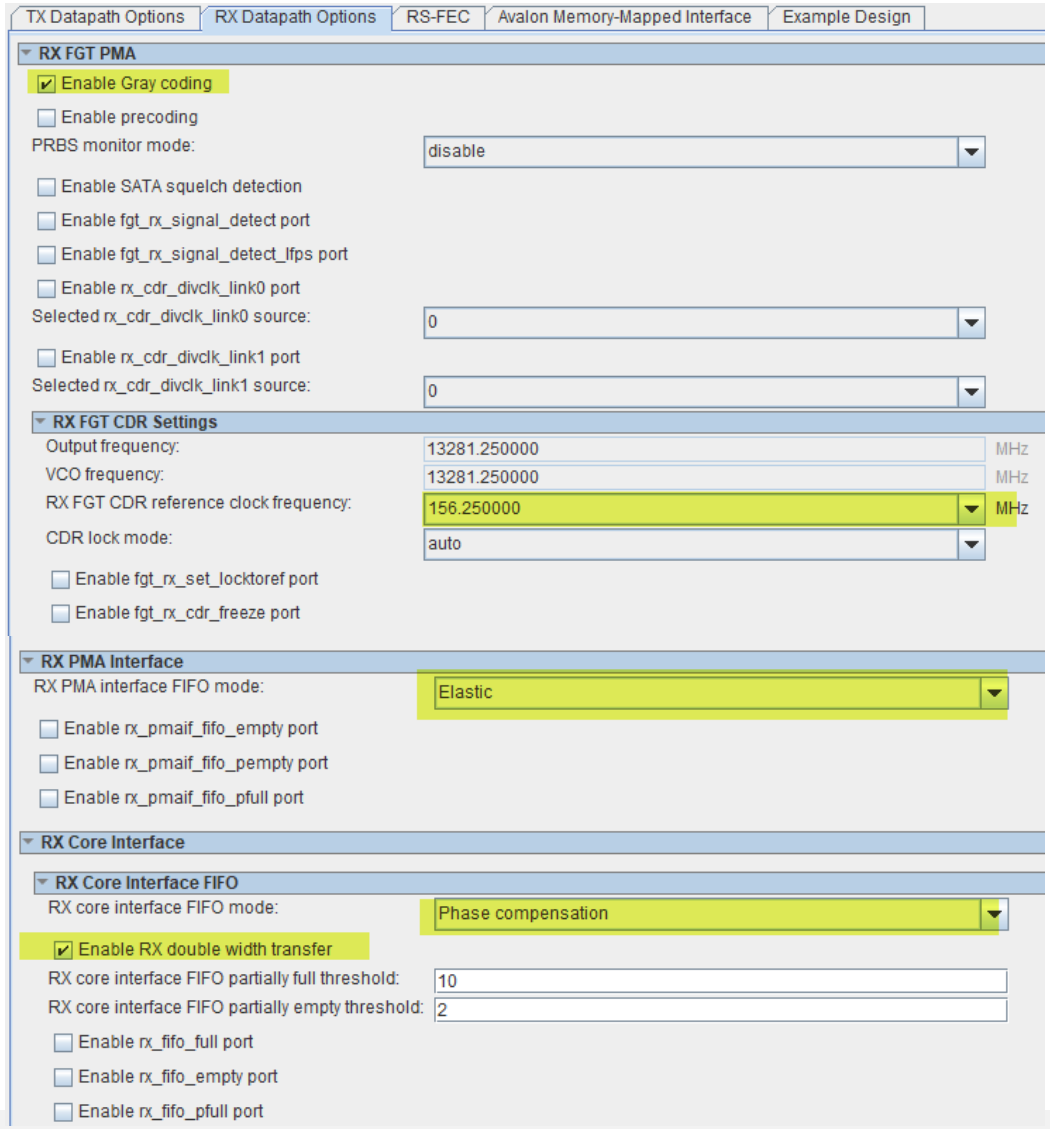
F-Tile Phy Direct IP (Tx Datapath Options)



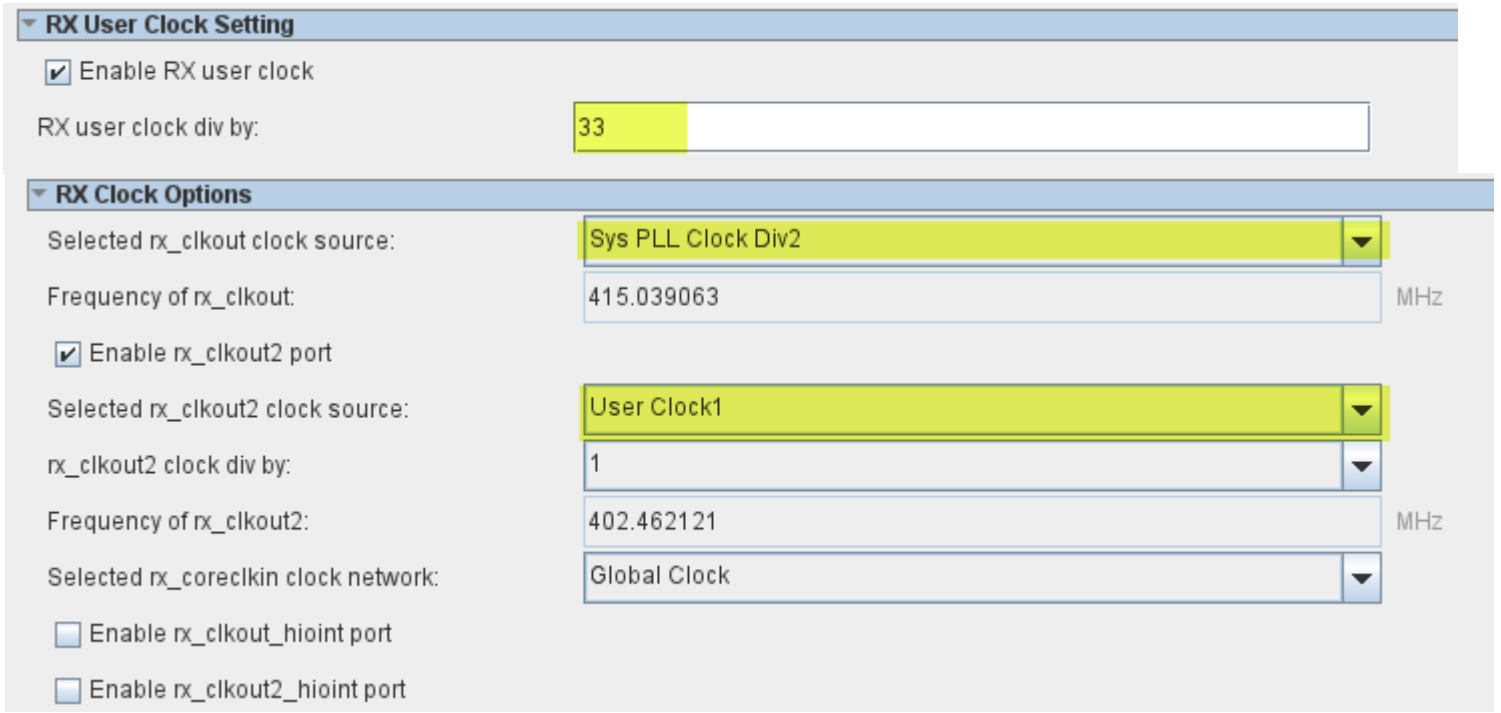
F-Tile Phy Direct IP (Tx Datapath Options cont.)



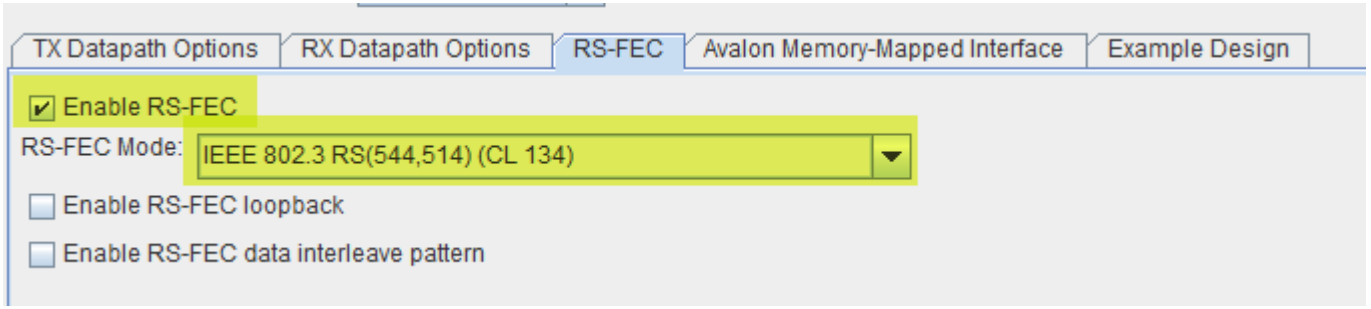
F-Tile Phy Direct IP (Rx Datapath Options)



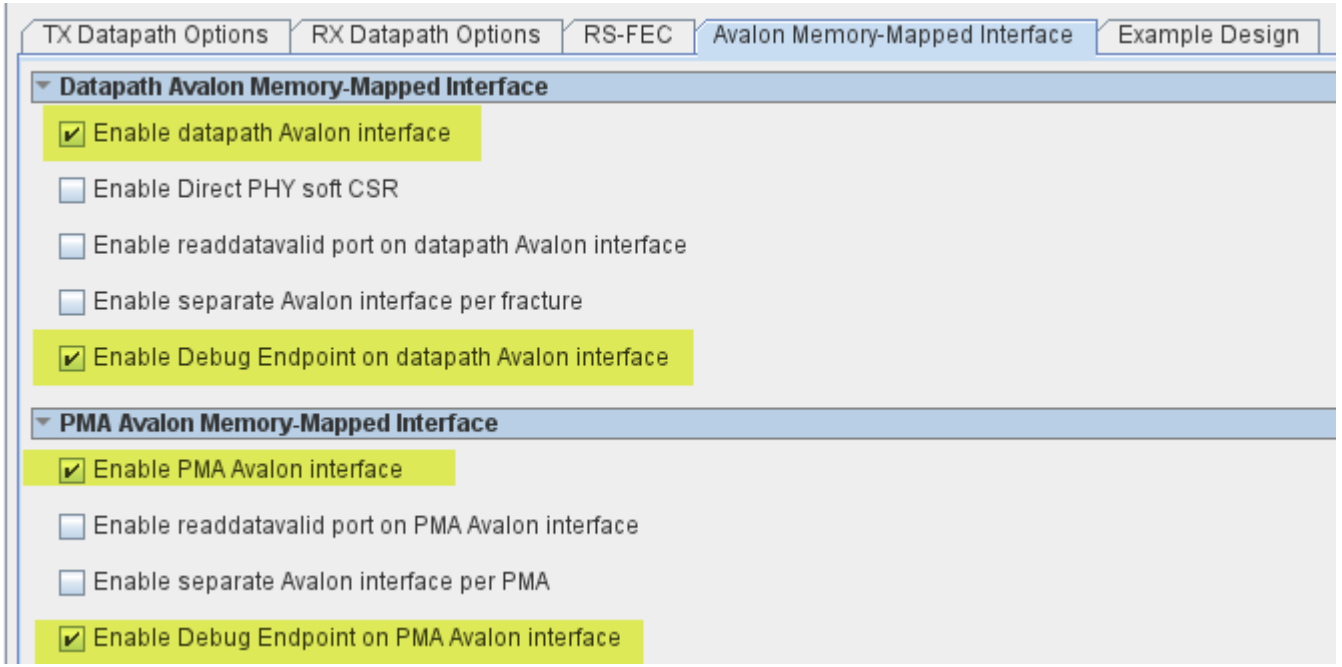
F-Tile Phy Direct IP (Rx Datapath Options cont.)



F-Tile Phy Direct IP (RS-FEC)

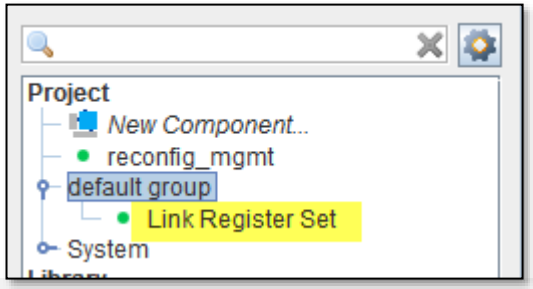


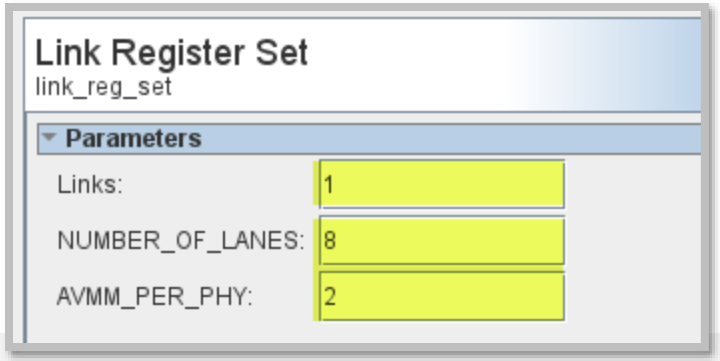
F-Tile Phy Direct IP (Avalon® Memory-Mapped Interface)



Link Register Set

- The design is using a custom .qsys component that can be parameterized. The component is called “Link Register Set” (using link_reg_set_hw.tcl) and is available in the default group section of the Qsys library.
- This component can be parameterized to the number of links, the number of LANES per Link and the number of PHY instances per link you want to instantiate in your design and makes the generation of the Qsys system much more streamlined.
- As in this design 1 Link with 2 phy’s are being and each link has 8 lanes you end up with the configuration below (see screenshot)
- For every Link that you use in your design it will generate a number of registers (to control/readout the measured clocks) as well as all the required Avalon Memory Mapped reconfiguration interfaces for the Native PHY (which is for every PHY instance : 1 Avalon Memory Mapped for the transceiver configuration interface and 1 Avalon Memory Mapped for the RSFEC configuration interface).
- For the Avalon Memory Mapped interfaces the Link Register set is calling another custom component : “reconfig_mgmt” (reconfig_mgmt_hw.tcl)





Timing Closure (Setup)

- The design is fully constraint without any timing violations. The entire logic is basically clocked from one clock (the highlighted one below). The rx_clkout2 is for measuring the recovered clock but is not used for clocking the logic.
- Compiled for AGIB027R29A1E2VR0

Setup Summary

Show: Visible | Hide | <<Filter>>

	Clock	Slack	End Point TNS	Failing End Points	Worst-Case Operating Conditions
1	Generate_Superlite_IV_Instances[0].ins...rect_inst|directphy_f_0|tx_clkout|ch23	0.146	0.000	0	Slow vid2 100C Model
2	Generate_Superlite_IV_Instances[0].ins...ect_inst|directphy_f_0|rx_clkout2|ch23	1.530	0.000	0	Slow vid2 100C Model
3	clock_divider_inst|intelclkctrl_0|clkdiv_inst|clock_div2	1.943	0.000	0	Slow vid2 100C Model
4	altera_int_osc_clk	1.986	0.000	0	Slow vid2 100C Model
5	src_divided_osc_clk	4.523	0.000	0	Slow vid2b 100C Model
6	altera_reserved_tck	12.396	0.000	0	Slow vid2 100C Model

Timing Closure (Hold)

- The design is fully constraint without any timing violations
- Compiled for AGIB027R29A1E2VR0

Hold Summary

Show: Visible | Hide | <<Filter>>

	Clock	Slack	End Point TNS	Failing End Points	Worst-Case Operating Conditions
1	clock_divider_inst|intelclkctrl_0|clkdiv_inst|clock_div2	0.000	0.000	0	Fast vid2 100C Model
2	Generate_Superlite_IV_Instances[0].ins...rect_inst|directphy_f_0|tx_clkout|ch23	0.006	0.000	0	Fast vid2 100C Model
3	altera_reserved_tck	0.064	0.000	0	Fast vid2 100C Model
4	altera_int_osc_clk	0.066	0.000	0	Fast vid2a 0C Model
5	src_divided_osc_clk	0.071	0.000	0	Fast vid2a 0C Model
6	Generate_Superlite_IV_Instances[0].ins...ect_inst|directphy_f_0|rx_clkout2|ch23	0.100	0.000	0	Fast vid2a 0C Model

Native Phy Debug Master Endpoint Interface

- In addition to the Avalon Memory Mapped interface the Native PHY also enable the Native Phy Debug Master Endpoint interface which can also be accessed through the Qsys system due to the addition of the jtag_debug_module.
- Access to the Native Phy Debug Master Endpoint interface can be done through System Console totally in parallel while the Nios controller(s) are being used.
- The Native Phy Debug Master Endpoint interface can be used for advanced debugging and reporting (see further) .
- The major advantage of having the Native Phy Debug Master Endpoint interface is that it allows to run directly the Universal Toolkit (see further).

Nios II Control & Monitoring

- The Nios II controller is used to control and monitor the design using the Nios II terminal.
  - Display status signals for each of the channels
    - Tile/Quad information
    - Encoding (PAM4/NRZ and gray and/or 1/1+D encoding)
    - Link Up
    - DataLocked (combination of the PrbsLocked and Countlocked)
    - LaneAligned
    - Effective Datarate
    - Latency (only when looped back on its own)
    - RSFEC Statistics
    - RX AM Lock Alarm Count
    - Lock Alarm
    - Freq_Locked
    - Tx_ready/Rx_ready
    - Serial Loop + Reverse Parallel Loop
    - Channel Type
    - Connection Type
    - Rx/Tx Polarity Inversion
    - Number of biterrors.
  - Displays the bitrate based on measured reference clock frequency
  - Displays also the recovered clock frequency (of lane 0 only). Note that if lane 0 has no connection or is bit errors not in serial loopback the measured recovered clock frequency is a random number as the CDR is not locked to anything.

Nios II Control & Monitoring (continued)

- Control Serial Loopback on all lanes and per phy
- Control Reverse Parallel Loopback on all lanes and per phy
- Error insertion : Soft Biterror (Pre-FEC) (which results in one error per virtual link) or error insertion through RS-FEC directly.
- Displays the number of biterrors and the BER.
- Reads out and print out PMA settings for all channels
- Control Invert Tx and Rx Polarity on all lanes and per phy
- PMA tuning : Allows to find the optimum Transmit PMA settings (VOD, Pre-emphasis) using PMA sweep function.
- Stress Tests : 9 different stress tests are available (see dedicated slide) including latency measurement
- Displays the time the test has been running on the Transceiver Block.

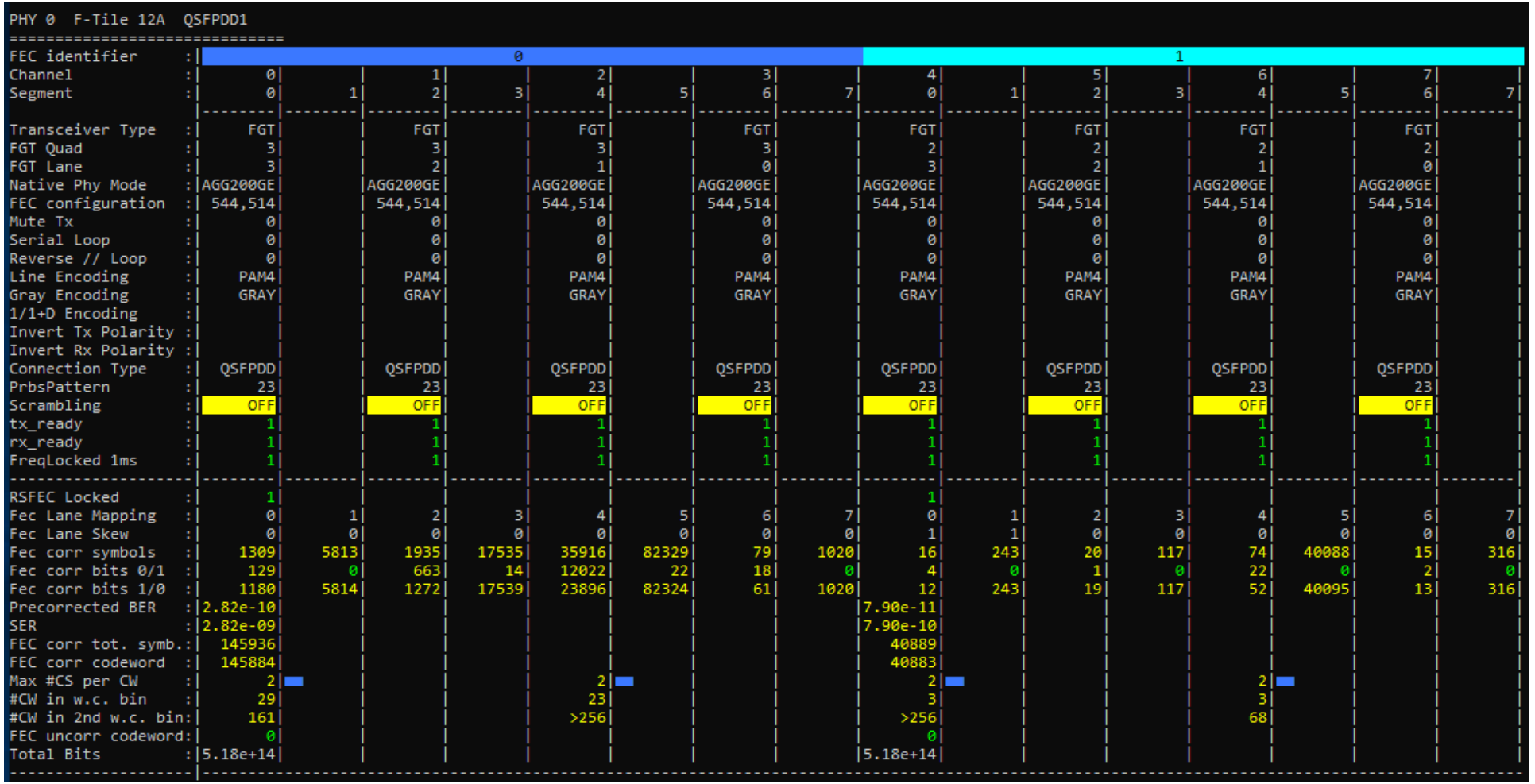
Nios II Control & Monitoring (to be added)

- Measures the temperatures of the core (supported) and the transceiver tile (to be added)
- Dumps I2C information of connected modules and cables like Vendor, part number, cable length, temperature of the module. (due to a hardware issue on the PCIe devkit (ES Version) the I2C interface does not work for some modules, so the software will ask whether you want to enable the I2C access when you download the .elf file) This assumes the hardware modification to the board is done (i.e replace R122 and R127 with 0R resistors or short them).
- Allows to dump the I2C page Lower page and Page 00h registers of the selected module/cable

Nios II Output in Nios II SDK Shell

```
Agilex I-series PCIe Devkit Superlite IV Demo
-------------------------------------------------------------------------------
|Board Revision           : Agilex I-Series PCIe Devkit F-tile (ES Version)
|Hardware Revision        : 01/05/2022 variant 01
|Clocking of PHY's        : System PLL clocking
|FGT Firmware version     : 0x3cb40195
|Software Build Date      : Jan  6 2022  18:03:37
-------------------------------------------------------------------------------
|PHY                                |       0|
|RefClock Measured (Mhz)            |156.2500|
|Line rate (Gbps)                   | 53.1250|
|Aggregate rate (Gbps)              |425.0000|
|Selected Channel                   |       0|
|Number of Phy's                    |       1
|Selected Phy                       |       0
-------------------------------------------------------------------------------
|Superlite Related statistics:
|Read Length                        |   40960|
|Idle Length                        |       7|
|Ratio                              |0.999829|
|Throttle Datarate                  |       0|
|Net Received Bandwidth (Gbps)      |399.8730|
|Efficiency (%)                     |94.08776|
|Measured Max Latency (ns)          |     291| (only valid with loopback (int or ext)
|Measured Max Latency (clocks)      |     121| (only valid with loopback (int or ext)
-------------------------------------------------------------------------------
```

Test with electrical loopback module on QSFP-DD 1



PHY 0 F-Tile 12A QSFPDD1

	0								1							
Channel	0		1		2		3		4		5		6		7	
Segment	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Transceiver Type	FGT		FGT		FGT		FGT		FGT		FGT		FGT		FGT	
FGT Quad	3		3		3		3		2		2		2		2	
FGT Lane	3		2		1		0		3		2		1		0	
Native Phy Mode	AGG200GE		AGG200GE		AGG200GE		AGG200GE		AGG200GE		AGG200GE		AGG200GE		AGG200GE	
FEC configuration	544,514		544,514		544,514		544,514		544,514		544,514		544,514		544,514	
Mute Tx	0		0		0		0		0		0		0		0	
Serial Loop	0		0		0		0		0		0		0		0	
Reverse // Loop	0		0		0		0		0		0		0		0	
Line Encoding	PAM4		PAM4		PAM4		PAM4		PAM4		PAM4		PAM4		PAM4	
Gray Encoding	GRAY		GRAY		GRAY		GRAY		GRAY		GRAY		GRAY		GRAY	
1/1+D Encoding																
Invert Tx Polarity																
Invert Rx Polarity																
Connection Type	QSFPDD		QSFPDD		QSFPDD		QSFPDD		QSFPDD		QSFPDD		QSFPDD		QSFPDD	
PrbsPattern	23		23		23		23		23		23		23		23	
Scrambling	OFF		OFF		OFF		OFF		OFF		OFF		OFF		OFF	
tx_ready	1		1		1		1		1		1		1		1	
rx_ready	1		1		1		1		1		1		1		1	
FreqLocked 1ms	1		1		1		1		1		1		1		1	
RSFEC Locked	1								1							
Fec Lane Mapping	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
Fec Lane Skew	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
Fec corr symbols	1309	5813	1935	17535	35916	82329	79	1020	16	243	20	117	74	40088	15	316
Fec corr bits 0/1	129	0	663	14	12022	22	18	0	4	0	1	0	22	0	2	0
Fec corr bits 1/0	1180	5814	1272	17539	23896	82324	61	1020	12	243	19	117	52	40095	13	316
Precorrected BER	2.82e-10								7.90e-11							
SER	2.82e-09								7.90e-10							
FEC corr tot. symb.	145936								40889							
FEC corr codeword	145884								40883							
Max #CS per CW	2				2				2				2			
#CW in w.c. bin	29				23				3				3			
#CW in 2nd w.c. bin	161				>256				>256				68			
FEC uncorr codeword	0								0							
Total Bits	5.18e+14								5.18e+14							

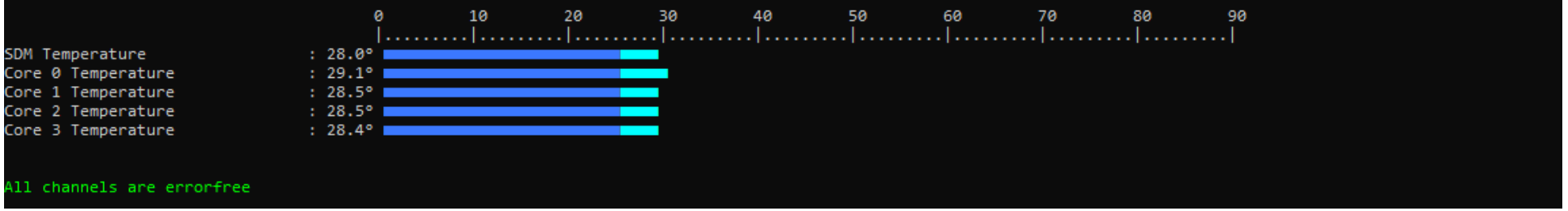
Test with electrical loopback module on QSFP-DD 1 (continued)

```
Pre corr BER Aggr  :|1.805249e-10  (Pre-Corrected Aggregrate BER)
SER Aggregate      :|1.804815e-09  (Symbol Error Rate Aggregate)
Total Bits Aggr    :|1.035148e+15

XOFF Sent          :        0
XOFF Received      :        0
Throttle Data      :        0
Error Deskew       :        0
Fifo Error         :        0
WordAligned        :        1
LaneAligned        :        1
LinkUp             :        1
DataLocked         :        1
DataLocked Alarm   :
Errorcount         :        0


BER (CL=0.95) Link         : 2.898137e-15

Test Time PHY 0                    :  0h 40m 35s
Tx Clkout Frequency                : 415.03906 MHz
DataClock_out Frequency            : 390.50101 Mhz
Recovered Clock Frequency (Lane 0) : 402.46210 MHz
Measured ppm difference            : 0
```




```
All channels are errorfree
```

Available commands

```
Select Action :
==============
P. Toggle Display to show info for All Phy's or only Selected Phy
[. Mute all FGT transmitters
]. Unmute all FGT transmitters
1. Perform a reset on the Selected Phy
3. Select Channel To Control
5. Insert Biterrors on Link (16 at a time)
*. Inject RSFEC errors during 1ms on Selected Channel/Segment in Selected PHY (pat=0x01,rate=0x01)
#. Build FEC Errortree live for Selected Phy
6. Reset ErrorCounter
7. Show Status
8. Control Serial loopback in Selected Phy
{. Control Reverse Parallel loopback in Selected Phy
C. continuously update BER and errorcount every 2 seconds
D. Input new BER Time Interval
E. Show Transceiver PMA Settings on all channels
G. Change TX PMA settings on the Selected Phy
M. Control Tx Polarity in Selected Phy
N. Control Rx Polarity in Selected Phy
Q. Clear Errorcounters on all Phys (including RSFEC counters)
R. Rerun adaptation on all channels in Selected Phy
S. Store register space of Selected Channel
T. Readout register space of Selected Channel and compare with previously stored register space and list differences
U. Sweep PMA settings and program best setting on selected channel
K. Toggle Throttle Datarate (set to 50% throughput) on Selected Phy
X. Toggle XOFF to partner to stop/start sending traffic at remote side
}. Dump I2C registers of selected module (Lower Page and Page 00h)
!. Read Register of PHY Direct in Selected Phy using rd_channel
:. Read Register of PHY Direct in Selected Phy using rd_channel_ftile
L. Stress test functions including latency measurement
Z. Dump address space
To stop the test press CTRL-C (this will automatically create output.log file
```

Stress Tests

- Option ‘L’ provides a submenu to perform the following “stress tests”:

```
Press 1 if you want to reset only the Selected PHY and check it status
Press 2 if you want to reset the Selected PHY and check if both PHY's come up properly (requires connection between the PHY's)
Press 3 if you want to do rx_reset of the Selected PHY and check it status
Press 4 if you want to stress test AVMM accesses to the reconfiguration interface and the PDP interface
Press 5 to reset the rx of all lanes of the Selected Phy until the link is up on all lanes of the Selected Phy
Press 6 to reset the Selected Rx channel until the link is up on that Channel
Press 7 to run a test mimicing cable pull/plug-in using Tx mute option on local phy and verify if remote PHY comes up properly
Press 8 to run the same test as test 1 but specifically when used in combination with SiPh which take seconds to come up
Press 9 to measure latency across 1000 resets
```

Measuring Latency

- Stress test ‘9’ is measuring the maximum latency across 1000 resets and provides a histogram of this maximum latency being measured.
- As can be seen maximum latency is on average 121 parallel clock cycles which is ~ 292 ns at a line rate of 53.125 Gbps. (This is with external loopback on 8 of the physical lanes) using QSFP-DD 1x1 loopback.

```
Number of cases Maximum Latency is equal to 118 parallel clocks is :    4
Number of cases Maximum Latency is equal to 119 parallel clocks is :   39
Number of cases Maximum Latency is equal to 120 parallel clocks is :  282
Number of cases Maximum Latency is equal to 121 parallel clocks is :  570
Number of cases Maximum Latency is equal to 122 parallel clocks is :  105
```

Show PMA settings (option ‘e’) using QSFP-DD loopback

Channel	0	1	2	3	4	5	6	7	
FGT Quad	3	3	3	3	2	2	2	2	
FGT Lane	3	2	1	0	3	2	1	0	
Transceiver Type	FGT	FGT	FGT	FGT	FGT	FGT	FGT	FGT	
Line Encoding	PAM4	PAM4	PAM4	PAM4	PAM4	PAM4	PAM4	PAM4	
tx_ready	1	1	1	1	1	1	1	1	
rx_ready	1	1	1	1	1	1	1	1	
rx_freqlocked_1ms	1	1	1	1	1	1	1	1	
Serial Loop	0	0	0	0	0	0	0	0	
Reverse Parallel	0	0	0	0	0	0	0	0	
Main Tap	35	35	35	35	35	35	35	35	
Pre Tap 2 Level	0	0	0	0	0	0	0	0	
Pre Tap 1 Level	5	5	5	5	5	5	5	5	
Post Tap 1 Level	0	0	0	0	0	0	0	0	
RXEQ_HF_BOOST	0	0	0	0	0	0	0	0	
RXEQ_VGA_GAIN	0	0	0	0	0	0	0	0	
RXEQ_DFE_TAP 1	27	27	12	30	25	24	25	25	
RXEQ_DFE_TAP 2	27	11	29	11	5	4	13	14	
RXEQ_DFE_TAP 3	14	10	13	25	11	26	10	25	
RXEQ_DFE_TAP 4	10	5	10	7	13	5	13	4	
RXEQ_DFE_TAP 5	7	13	1	6	7	15	10	3	
RXEQ_DFE_TAP 6	14	1	5	11	2	4	-1	13	
RXEQ_DFE_TAP 7	2	12	15	7	5	3	12	7	
RXEQ_DFE_TAP 8	6	3	2	6	5	5	7	3	
RXEQ_DFE_TAP 9	2	7	3	2	2	7	6	5	
RXEQ_DFE_TAP10	1	-1	-3	0	3	-3	3	3	
RXEQ_DFE_TAP11	6	1	3	3	0	7	3	2	
RXEQ_DFE_TAP12	3	1	1	1	1	0	-1	-3	
RXEQ_DFE_TAP13	3	2	7	6	3	3	6	3	
RXEQ_DFE_TAP14	0	0	1	-1	1	1	-1	6	
RXEQ_DFE_TAP15	-2	-3	3	0	0	-6	-1	-2	
RXEQ_DFE_TAP16	1	-1	-12	-1	-1	3	1	0	
FOM	49	47	46	50	53	55	52	54	
VGA	41	44	46	45	43	45	42	41	
CTLE	5	3	0	1	4	0	5	4	
Convergence time	91	90	124	131	84	82	87	113	(ms) (only LSB portion)
Rx Flow State	6	8	6	11	9	6	11	9	

Sweep PMA function (electrical loopback) (option ‘u’)

```
setting   0 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  0 , rx_ready : 0, no_lock, Vertical Eye : -1
setting   1 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  1 , rx_ready : 0, no_lock, Vertical Eye : 45
setting   2 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  2 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting   3 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  3 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 50
setting   4 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  4 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 50
setting   5 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  5 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 49
setting   6 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  6 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 48
setting   7 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  7 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting   8 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  8 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 50
setting   9 Tx_phy 0 Rx_phy 0 Ch 0 : VOD :  9 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting  10 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 10 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 49
setting  11 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 11 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 48
setting  12 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 12 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 48
setting  13 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 13 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting  14 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 14 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 46
setting  15 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 15 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 48
setting  16 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 16 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 48
setting  17 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 17 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 49
setting  18 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 18 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 46
setting  19 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 19 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting  20 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 20 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 38
setting  21 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 21 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 41
setting  22 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 22 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 43
setting  23 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 23 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 42
setting  24 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 24 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 43
setting  25 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 25 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 46
setting  26 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 26 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 46
setting  27 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 27 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting  28 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 28 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 45
setting  29 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 29 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 48
setting  30 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 30 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 48
setting  31 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 31 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 46
setting  32 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 32 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 48
setting  33 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 33 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 49
setting  34 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 34 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 49
setting  35 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 35 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 48
setting  36 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 36 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting  37 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 37 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting  38 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 38 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting  39 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 39 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 49
setting  40 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 40 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting  41 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 41 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 47
setting  42 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 42 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 46
setting  43 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 43 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 46
setting  44 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 44 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 45
setting  45 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 45 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 45
setting  46 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 46 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 45
setting  47 Tx_phy 0 Rx_phy 0 Ch 0 : VOD : 47 , BER : 0.000000e+00, ErrorCount : 0.000000e+00 Vertical Eye : 44
Range              :  47
Optimum range      :  46
Ber_Minimum        : 0.000000e+00
Start_value        :   2
End_value          :  47
sweep_low          :   0
best_setting BER   :  25
```

Continuous Update of BER and FEC statistics (option ‘c’)

```
Phy 0 Ch 0 BER related statistics every 2 seconds
Press Enter to stop the loop

==========================================
>>>> HH:MM:SS,Core_Temp[0],Core_Temp[1],Core_Temp[2],Core_Temp[3],PrbsLockAlarm,BER,Errorcount(63:32),ErrorCount(31:0),Errorcount(float),Uncorrected CW,Corrected CW,Incremental Corrected CW,Precorrected BER,Totalbits
>>>>  0h: 0m: 1s, 28.5, 27.6, 27.1, 26.9,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 2.160000e+02,      216, 2.046242e-09, 1.055594e+11
>>>>  0h: 0m: 3s, 28.5, 27.0, 27.5, 27.2,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 4.420000e+02,      226, 2.101495e-09, 2.112781e+11
>>>>  0h: 0m: 5s, 28.5, 27.5, 27.5, 26.9,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 6.490000e+02,      207, 2.054337e-09, 3.168906e+11
>>>>  0h: 0m: 7s, 28.1, 27.5, 27.9, 27.0,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 8.690000e+02,      220, 2.063631e-09, 4.225562e+11
>>>>  0h: 0m: 9s, 28.1, 27.8, 27.1, 27.0,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 1.088000e+03,      219, 2.065420e-09, 5.282219e+11
>>>>  0h: 0m:11s, 28.5, 27.5, 27.5, 26.9,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 1.304000e+03,      216, 2.061880e-09, 6.338875e+11
>>>>  0h: 0m:13s, 28.1, 27.6, 27.9, 26.9,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 1.518000e+03,      214, 2.058000e-09, 7.395531e+11
>>>>  0h: 0m:15s, 28.1, 27.5, 27.1, 27.2,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 1.716000e+03,      198, 2.036287e-09, 8.451656e+11
>>>>  0h: 0m:17s, 28.1, 27.5, 27.1, 26.9,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 1.941000e+03,      225, 2.046630e-09, 9.508312e+11
>>>>  0h: 0m:19s, 28.1, 27.5, 27.5, 26.9,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 2.183000e+03,      242, 2.070995e-09, 1.056497e+12
>>>>  0h: 0m:21s, 27.9, 27.5, 27.1, 26.9,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 2.399000e+03,      216, 2.069418e-09, 1.162162e+12
>>>>  0h: 0m:23s, 28.1, 27.5, 27.1, 26.9,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 2.658000e+03,      259, 2.101231e-09, 1.267828e+12
>>>>  0h: 0m:25s, 28.1, 27.5, 27.5, 26.8,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 2.884000e+03,      226, 2.104205e-09, 1.373441e+12
>>>>  0h: 0m:27s, 28.1, 27.6, 27.5, 27.0,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 3.121000e+03,      237, 2.114115e-09, 1.479106e+12
>>>>  0h: 0m:29s, 28.1, 27.6, 27.5, 26.9,  0, 0.000000e+00,         0,         0, 0.000000e+00,       0, 3.359000e+03,      238, 2.123965e-09, 1.584772e+12
```

FEC Error Tree (option ‘#’) (electrical loopback)

- This will build “live”, note that if a number is in red it means at least 1 overrun has occurred for that particular bin (each bin is only 8 bits (so 256), so for the lower bins the software can’t read fast enough and print at the same time to prevent an overrun. If only 1 overrun has occurred the value is correct though (so the higher bins values are correct)

```
Fec Error Tree Phy 0
Press 's' + enter to stop
==========================
Time                :| 0h:14m:10s

FEC identifier      :|                                 0                                 |                                 1                                 |
Channel             :|       0|        |       1|        |       2|        |       3|        |       4|        |       5|        |       6|        |       7|        |
Segment             :|       0|       1|       2|       3|       4|       5|       6|       7|       0|       1|       2|       3|       4|       5|       6|       7|
                     |--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
Transceiver Type    :|     FGT|        |     FGT|        |     FGT|        |     FGT|        |     FGT|        |     FGT|        |     FGT|        |     FGT|        |
FGT Quad            :|       3|        |       3|        |       3|        |       3|        |       2|        |       2|        |       2|        |       2|        |
FGT Lane            :|       3|        |       2|        |       1|        |       0|        |       3|        |       2|        |       1|        |       0|        |
Native Phy Mode     :|AGG200GE|        |AGG200GE|        |AGG200GE|        |AGG200GE|        |AGG200GE|        |AGG200GE|        |AGG200GE|        |AGG200GE|        |
FEC configuration   :| 544,514|        | 544,514|        | 544,514|        | 544,514|        | 544,514|        | 544,514|        | 544,514|        | 544,514|        |
RSFEC Locked        :|       1|        |        |        |        |        |        |        |       1|        |        |        |        |        |        |        |
FEC uncorr codeword:|       0|        |        |        |        |        |        |        |       0|        |        |        |        |        |        |        |
                     |--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
#CW without errors  :|  440752|        |        |        |  437979|        |        |        |  437181|        |        |        |  436330|        |        |        |
#CW with  1 CS      :|   50201|        |        |        |   49348|        |        |        |    6392|        |        |        |    6276|        |        |        |
#CW with  2 CS      :|      38|        |        |        |      27|        |        |        |       8|        |        |        |       9|        |        |        |
#CW with  3 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with  4 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with  5 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with  6 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with  7 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with  8 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with  9 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with 10 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with 11 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with 12 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with 13 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with 14 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |
#CW with 15 CS      :|       0|        |        |        |       0|        |        |        |       0|        |        |        |       0|        |        |        |

Note that if a count is indicated in red it means that counter has experienced at least one overrun
If only one overrun has occurred the value is correct
CW = CodeWord, CS = Corrected Symbols
```

Readout of the I2C information (not available on the PCIe devkit ES-version)

- Every time a status update is done all QSFP-DD modules will be checked if anything is plugged in and if this is the case the I2C information will be readout.
- See screenshot below showing various module and cables connected in the system.
- Information provided is Vendor, Part Number, temperature for modules and cable length for QSFP-DD cables or QSFP cables.
- One can dump the entire I2C Lower Page and Page 00h using option '{'

```
QSFPDD0 plugged in Vendor : Molex          Part  : 2015911010      QSFP-DD Passive Cable with length 1.00 m
QSFPDD1 plugged in Vendor : Molex          Part  : 2015911010      QSFP-DD Passive Cable with length 1.00 m
```

Board setup

- Make connections to the QSFPDD1 (Using loopback module, optical module or cable (to another board)
- Connect micro-USB Cable to J8 in order to use onboard USB-Blaster II.
- Power on the board
- All designs so far use the default clock frequency of 156.25 Mhz. Not required to use clock control GUI to change clocks.
- Download ‘Devkit_Demo.sof’ to the AGIB027R29A1E2VR0 device using QII 21.4 programmer

Running the software on Windows10

This is when you want to run the software provided as .elf file (part of the deliverable) Do the following :

1. Start a Nios II command Shell (can be found in the Quartus installation (e.g. C:\quartus_21_4\nios2eds) and go to the directory where the .elf and run1.bsh files are located.
2. Since 21.4 Nios II terminal is using WSL, a quick way to go to the selected directory is to copy the complete path using windows explorer and in the Nios 21.4 terminal you type cd "$(wslpath " => then you paste the path you copied (using right mouse click) => and end with typing ")"

   e.g. cd "$(wslpath " C:\ALTERA\Reference_Designs\SUPERLITEIV\Agilex_F-tile\Agilex_PCie_Kit_SuperliteIV_2x200G_1x8\software\app\Agilex_xCh")"
3. Since the Intel Quartus Prime software archive is restored in windows the files need to be converted from Dos to Linux. To do so either use "dos2unix –k *" or run the provided .bsh file "./convert.bsh" (note option –k keeps the original date of the file intact).
4. Type In The Following Command : "./run1.bsh" This will set the JtagSpeed to 16M ,download the .elf file and start the downloaded Nios II code inside the main program memory.

Rebuilding the software on Windows 10

- In order to rebuild the software do the following (using QII 21.4 or later):
- Start a Nios II 21.4 Command Shell
  1. go to the ../software/app/Agilex_xCh directory (which is part of the archive) using the method descried on the previous slide
  2. Run the bash file ./convert.bsh (this will also convert the files in the bsp folder)
  3. Run ./create-this-app .This will compile the system libraries (in the bsp folder), compile the software code (main.c and a number of other .c files in this case) and produce a Nios II executable.
- If you make any changes to the software you need to rebuild by typing “make” in the Nios II 21.4 shell (after having created the software a first time)
- To run the software : type In The Following Command : “./run1.bsh” This will set the JtagSpeed to 16M and download the Nios II code to the FPGA and open the terminal.

Running the software on Linux

Follow the below steps when you want to run the software provided as .elf file (part of the deliverable):

1. Start a konsole with enough memory (e.g. “arc submit -i node/"memory>=128000" -- konsole”)
2. Make sure all restored files are unix files and are executable (use “./convert.bsh”)
3. Open an acds shell (e.g. “arc shell acds/21.4”)
4. Start a Nios Command Shell (e.g. “/p/psg/swip/releases/acds/21.4/67/linux64/nios2eds/nios2_command_shell.*”) One can figure out the path using “which Intel Quartus Prime software” while being in the acds shell
5. Download the .elf file using command “nios2-download -r -c 1 -d 1 -i "0" -g Agilex_xCh.elf” Note that this assumes the device is connected to USB-Cable with index 1. If another index is used, change accordingly.
6. Next use “nios2-terminal -c 1 -d 1 -i "0" |tee output.log” to run the software from program memory.

Rebuilding the software on Linux

- In order to rebuild the software do the following (using QII 21.4 or later):
- Start a Nios II 21.4 Command Shell using the steps on the previous slide
  1. go to the ../software/app/Agilex_xCh directory (which is part of the archive)
  2. Run the bash file **./convert.bsh** (should already have been done)
  3. Run **./create-this-app** .This will compile the system libraries (in the bsp folder), compile the software code (main.c and a number of other .c files in this case) and produce a Nios II executable.
- If you make any changes to the software you need to rebuild by typing “make” in the Nios II 21.4 shell (after having created the software a first time)

  To run the software :

  1. Download the .elf file using command “nios2-download -r -c 1 -d 1 -i "0" -g Agilex_xCh.elf”. Note that this assumes the device is connected to USB-Cable with index 1. If another index is used, change accordingly.
  2. Next use “nios2-terminal -c 1 -d 1 -i "0" |tee output.log” to run the software from program memory.

Using Transceiver Toolkit (need patch on top of 21.4)

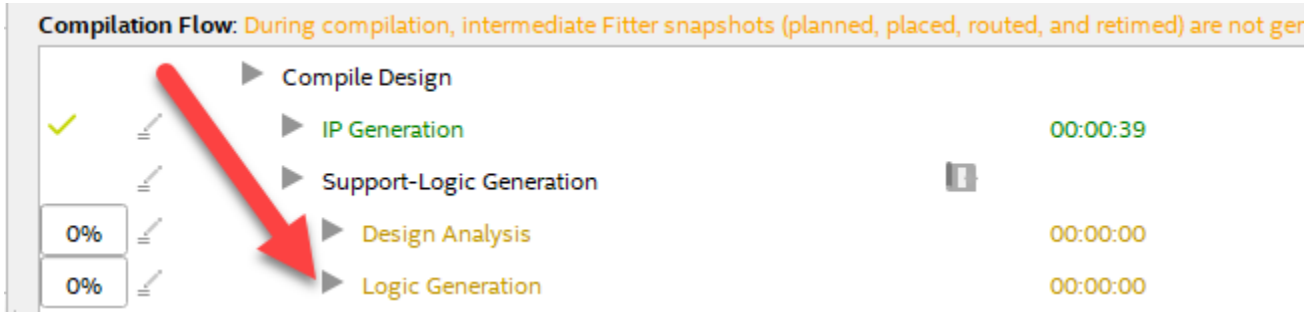
- To start the Transceiver Toolkit simply start it from the Tools Menu in Quartus (with the project loaded). This will automatically detect the design and start up the Transceiver Toolkit which allows you to run further tests in optimizing the PMA settings, do autosweeps etc.
- Make sure to stop all traffic running in TTK and close the TTK before going back to the Nios terminal to continue with the design (If the hard PRBS generator and checker have been enabled during testing those channels will show errors (which is normal). Using option ‘q’ these errors can be cleared

Signaltap II Debug Example

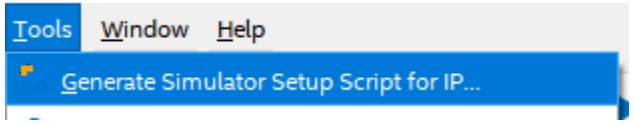
- The design contains a number of signaltap instances
- Signaltap II acquisition can be done simultaneously with Nios II control.

How to run a simulation using Questasim/Modelsim (1/4)

1. Open the project using Quartus in directory <restored_project>/simulation/mentor (DO NOT USE THE ORIGINAL PROJECT AFTER RESTORING THE ARCHIVE), the simulation has its own project
2. Make sure the IP settings are set correctly (see slide (3/4). IP settings are not fully governed by .qsf only, hence the additional check to be done to make sure they are correctly set
3. Press “Logic Generation” (see screenshot below) (this generates the __tiles files which are mandatory for the simulation) and also regenerates all .ip files and .spd files required for the simulation.

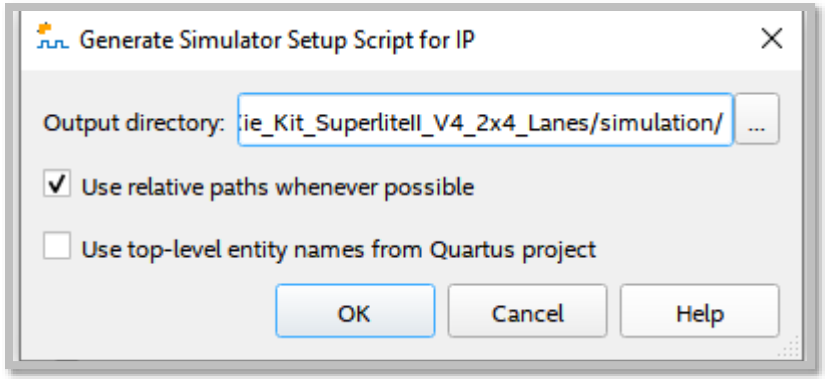


4. From the Quartus Menu Start Tools => Generate Simulator Setup Script for IP…

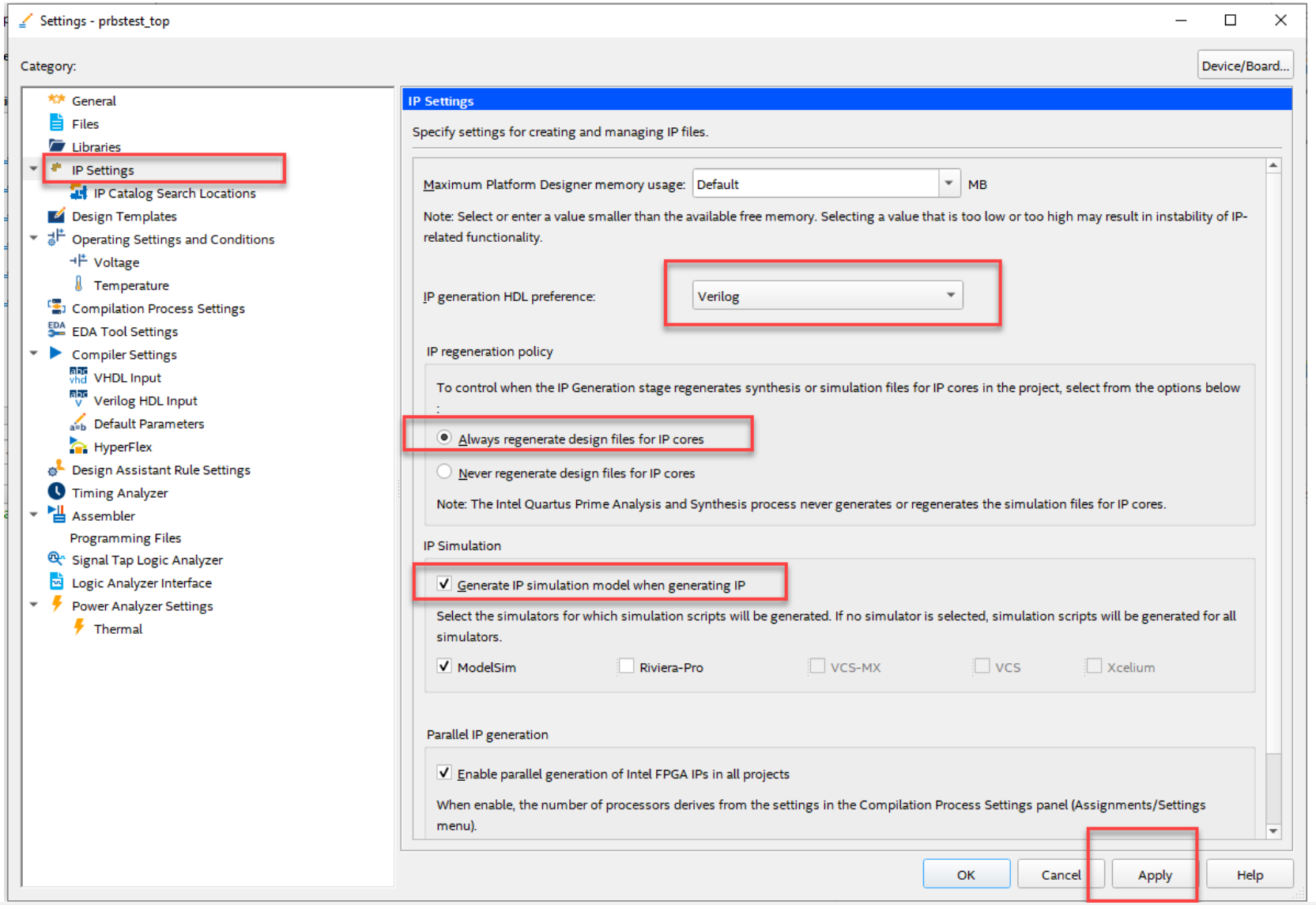


How to run a simulation using Questasim/Modelsim (2/4)

5. In the window that pops-up, remove the mentor in the path name so that you get <restored_project>/simulation/
   - Make sure “use top-level entity names from Quartus project” is not selected (in 21.3 this has already been removed as an option)



How to run a simulation using Questasim/Modelsim (3/4)



How to run a simulation using Questasim/Modelsim (4/4)

5. Go to the simulation/mentor folder and launch questasim e.g. using the following command *arc submit -i questasim/2020.4 questasim_sver-lic priority=100 -- "vsim -i -64"*
6. Type “sim.do” => this will do the entire simulation.
7. In case modifications are made to the source files you can rerun the simulation using “resim.do”

Deliverables

- Quartus 21.4 B67 archive which contains the demo design files, the software source files, files required to run the simulation
- The testbenches and project files required for simulation can be found after restoring the archive: <restored_project>/simulation/mentor
- The software source files can be found after restoring the archive in the <restored_project>/software/app and <restored_project>/software/bsp folders/
- SOF File
- Devkit_demo.elf File (Software)
- Readme.txt file

Status & Revision

- V21.4 : January 26th , 2022
  - Update to show the different Superlite IV variants that currently exist

- V21.4 : January 14th , 2022
  - Added skew measurement statistics to the design (shows skew between the FECs)
  - Successfully tested with Electrical Loopback.

- V21.4 : January 10th , 2022
  - Initial release of the design
  - Successfully tested with Electrical Loopback.

Disclaimer

- Intel technologies may require enabled hardware, software or service activation.
- No product or component can be absolutely secure.
- Your costs and results may vary.
- Performance varies by use, configuration and other factors.
- See our complete legal Notices and Disclaimers.
- Intel is committed to respecting human rights and avoiding complicity in human rights abuses.
- See Intel's Global Human Rights Principles. Intel's products and software are intended only to be used in applications that do not cause or contribute to a violation of an internationally recognized human right.