HUB Firmware overview

HUB team
Michigan State University

HUB Final Design Review
March 20, 2018

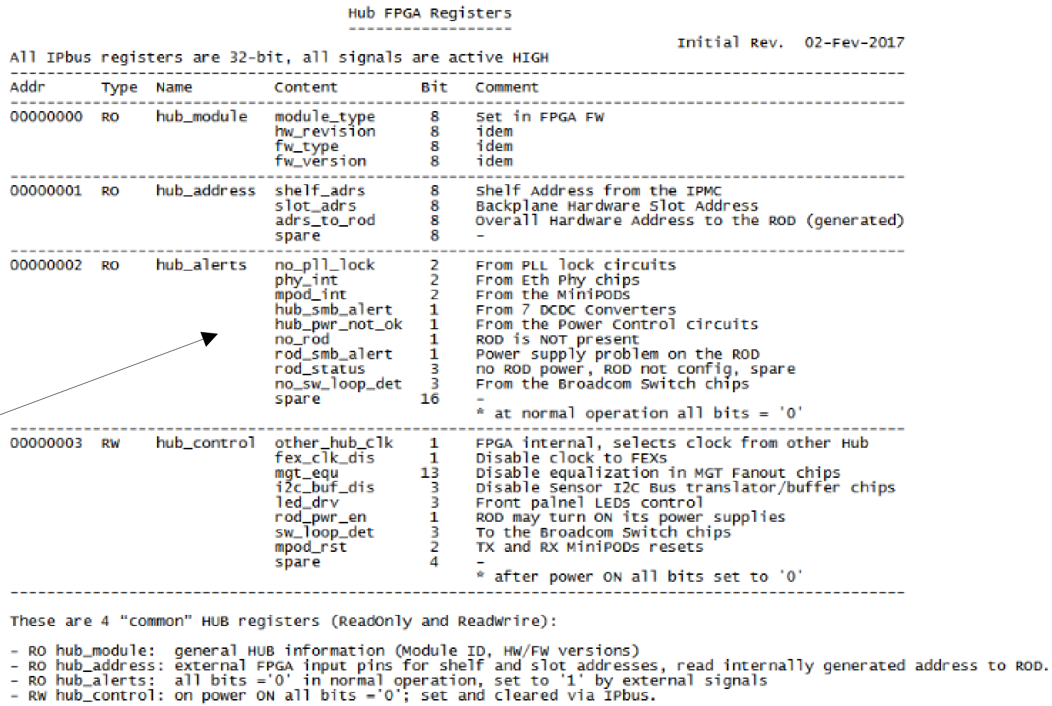
Introduction

- MAC + IPBus overview
- Combined_TTC/Data stream
- Readout_Control stream
- HUB Safe Configuration
- Readout (Aurora8b10b)
- HUB FW development scheme
- IBERT test (MGT channels)
- Conclusions

IPbus overview

- HUB: An IPbus interface provides high-level, functional control of the FEX-Hub module. This allows, for example, setting any firmware parameters, controlling modes of operation and reading monitoring data. This will allow a computer using IPbus to:
    - Access registers within the ROD FPGA, setting parameters and controlling modes of operation
    - Access external FPGA interfaces (I2C)

The Hub Module requires two physical chips for the Ethernet Base Interface connections. Two FPGA MACs are connected to the physical chips via RGMII ports.The single IPbus controller is connected to two MACs via MUX. IPbus was demonstrated to work with 2 MACs and PHY chips and demo slave registers.



```
                                   Hub FPGA Registers
                                   ------------------
                                                                 Initial Rev.   02-Fev-2017
All IPbus registers are 32-bit, all signals are active HIGH
-------------------------------------------------------------------------------------------
Addr      Type  Name         Content          Bit   Comment
-------------------------------------------------------------------------------------------
00000000  RO    hub_module   module_type        8   Set in FPGA FW
                             hw_revision        8   idem
                             fw_type            8   idem
                             fw_version         8   idem
-------------------------------------------------------------------------------------------
00000001  RO    hub_address  shelf_adrs         8   Shelf Address from the IPMC
                             slot_adrs          8   Backplane Hardware Slot Address
                             adrs_to_rod        8   Overall Hardware Address to the ROD (generated)
                             spare              8   -
-------------------------------------------------------------------------------------------
00000002  RO    hub_alerts   no_pll_lock        2   From PLL lock circuits
                             phy_int            2   From Eth Phy chips
                             mpod_int           2   From the MiniPODs
                             hub_smb_alert      1   From 7 DCDC Converters
                             hub_pwr_not_ok     1   From the Power Control circuits
                             no_rod             1   ROD is NOT present
                             rod_smb_alert      1   Power supply problem on the ROD
                             rod_status         3   no ROD power, ROD not config, spare
                             no_sw_loop_det     3   From the Broadcom Switch chips
                             spare             16   -
                                                    * at normal operation all bits = '0'
-------------------------------------------------------------------------------------------
00000003  RW    hub_control  other_hub_clk      1   FPGA internal, selects clock from other Hub
                             fex_clk_dis        1   Disable clock to FEXs
                             mgt_equ           13   Disable equalization in MGT Fanout chips
                             i2c_buf_dis        3   Disable Sensor I2C Bus translator/buffer chips
                             led_drv            3   Front palnel LEDs control
                             rod_pwr_en         1   ROD may turn ON its power supplies
                             sw_loop_det        3   To the Broadcom Switch chips
                             mpod_rst           2   TX and RX MiniPODs resets
                             spare              4   -
                                                    * after power ON all bits set to '0'
-------------------------------------------------------------------------------------------

These are 4 "common" HUB registers (ReadOnly and ReadWrire):

- RO hub_module:  general HUB information (Module ID, HW/FW versions)
- RO hub_address: external FPGA input pins for shelf and slot addresses, read internally generated address to ROD.
- RO hub_alerts:  all bits ='0' in normal operation, set to '1' by external signals
- RW hub_control: on power ON all bits ='0'; set and cleared via IPbus.
```

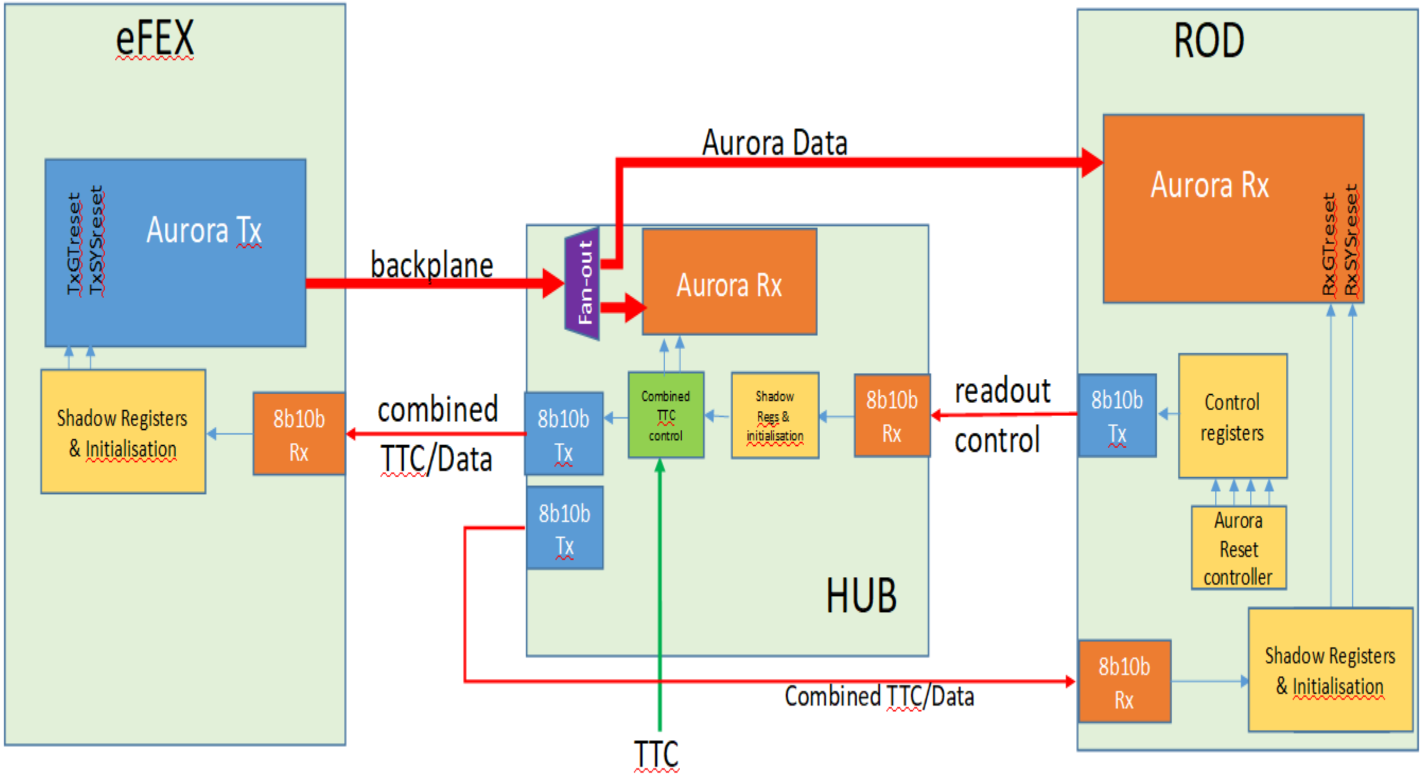
HUB FPGA Register – Initial map

Combined_TTC/DATA overview

- HUB module is obliged to distribute TTC information throughout the shelf. The combined_TTC/DATA bits are defined to provide TTC information as well as initialization functions (Aurora).The link is set to run at 6.4Gbps raw rate
- The HUB uses MiniPOD optical receiver to receive TTC signals from the FELIX system
- Receive the Reset signal (Aurora Initialization) from the Readout_CTRL stream and distribute it to the appropriate shelf slot
- Several Combined_TTC links within the shelf; each FEX slot (3-14); one between each HUB and ROD; links between two HUBs
- Combined_TTC/DATA links on the HUB FPGA is implemented with use of several components, including the MGT transceivers (GTH and GTY), control and diagnostic logic

Full spec at: Specification for Readout Control & Combined TTC Serial links in L1Calo

Combined_TTC/DATA overview



Full spec at: Specification for Readout Control & Combined TTC Serial links in L1Calo

Combined_TTC/DATA Physical Implementation

- The Combined_TTC/Data link on the HUB FPGA is implemented with the use of several components, including the MGT transceivers (GTH and GTY), control and the diagnostic logic.

- The Combined_TTC/DATA link is designed to operate at 6.4 Gbps.

- The physical implementation of the Combined_TTC/DATA stream assumes that there are 4 Control Registers on the Hub TX side, and Shadow Registers on the Rx side (Receiver: FEX, ROD and other HUB).

- The transmitter side generates the 128 bit message from 4 Control registers: Word_0, Word_1, Word_2, and Word_3. The transmitter side logic is in charge to write control information into these Control Registers. The contents of these registers are continuously transmitted to the modules
- (within the shelf) which receives the data into a duplicate set of 4 registers referred to as shadow registers. Anything written into a Control Register at the transmitter side will appear in the corresponding Shadow register at the receiving side within the following LHC clock.

- The least significant byte of Word_0 is reserved for the 8b10b Comma character K28.5.

Combined_TTC/DATA Bit definition

- The bits within the Combined_TTC/DATA control words are primarily defined to provide TTC information as well as initialization functions for all of the Fex data links (Aurora in the case of eFex). The TTC information is sourced from a dedicated TTC interface on the Hub. The Reset information is received from the ROD via the Readout_Ctrl link.

- **Comma Character**
- #Bits 7 to 0 of Word_0 contain the Comma character that maintains alignment of the 4 shadow registers with their corresponding Control registers. The chosen character is K28.5 = 0xBC.

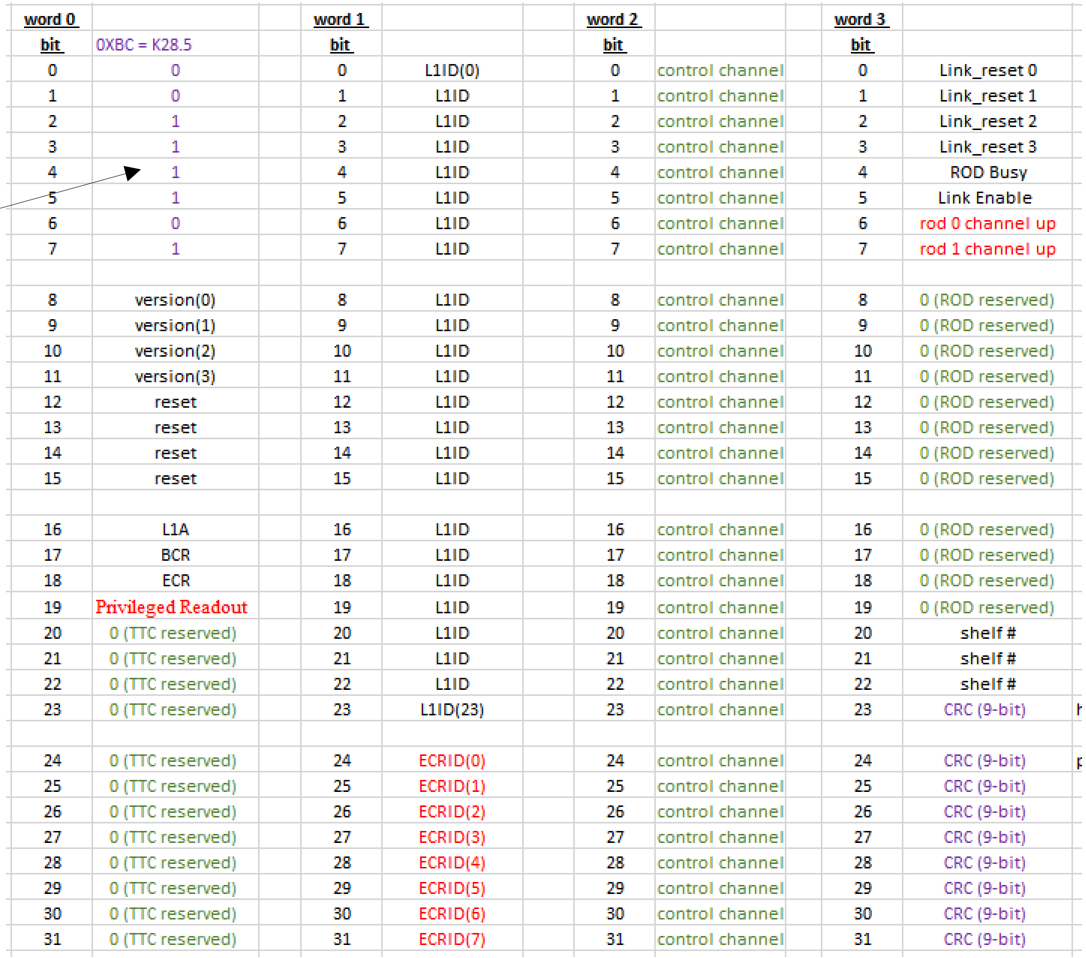
- **Version**
  #The 4-bit value contains the version number of this overall bit assignment. It will be held at "0000" through the initial debug phases, where many changes may occur.

- **Reset 3:0**
- #These bits provide a system level reset/enable function. There are four per slot, and the functionality of these bits will be specified by the targets (eFex, jFex, etc).

- **Level-1 Accept (L1A)**
  #L1 Accept is used to indicate when an event has been accepted by the Central Trigger Processor.

Combined_TTC/DATA Bit definition

Combined_TTC/DATA bit definitions: The least significant byte of Word_0 is reserved for the 8b10b Comma Character K28.5

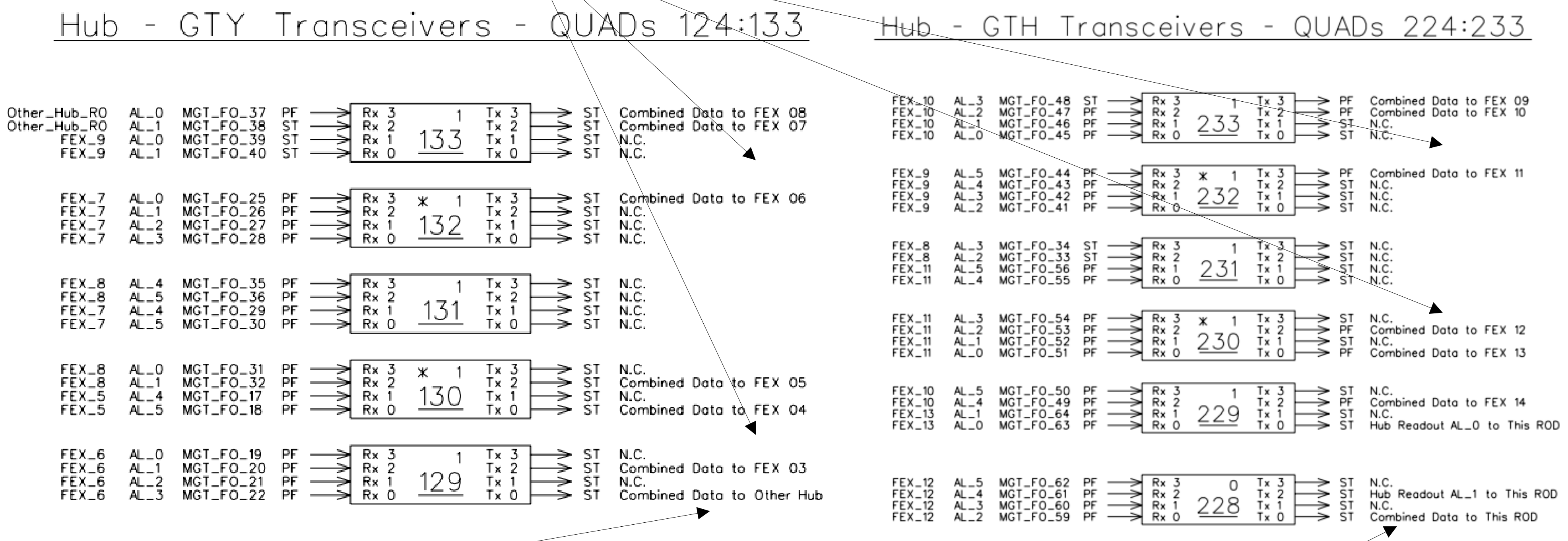


word 0		word 1		word 2		word 3	
bit	0XBC = K28.5	bit		bit		bit	
0	0	0	L1ID(0)	0	control channel	0	Link_reset 0
1	0	1	L1ID	1	control channel	1	Link_reset 1
2	1	2	L1ID	2	control channel	2	Link_reset 2
3	1	3	L1ID	3	control channel	3	Link_reset 3
4	1	4	L1ID	4	control channel	4	ROD Busy
5	1	5	L1ID	5	control channel	5	Link Enable
6	0	6	L1ID	6	control channel	6	rod 0 channel up
7	1	7	L1ID	7	control channel	7	rod 1 channel up
8	version(0)	8	L1ID	8	control channel	8	0 (ROD reserved)
9	version(1)	9	L1ID	9	control channel	9	0 (ROD reserved)
10	version(2)	10	L1ID	10	control channel	10	0 (ROD reserved)
11	version(3)	11	L1ID	11	control channel	11	0 (ROD reserved)
12	reset	12	L1ID	12	control channel	12	0 (ROD reserved)
13	reset	13	L1ID	13	control channel	13	0 (ROD reserved)
14	reset	14	L1ID	14	control channel	14	0 (ROD reserved)
15	reset	15	L1ID	15	control channel	15	0 (ROD reserved)
16	L1A	16	L1ID	16	control channel	16	0 (ROD reserved)
17	BCR	17	L1ID	17	control channel	17	0 (ROD reserved)
18	ECR	18	L1ID	18	control channel	18	0 (ROD reserved)
19	Privileged Readout	19	L1ID	19	control channel	19	0 (ROD reserved)
20	0 (TTC reserved)	20	L1ID	20	control channel	20	shelf #
21	0 (TTC reserved)	21	L1ID	21	control channel	21	shelf #
22	0 (TTC reserved)	22	L1ID	22	control channel	22	shelf #
23	0 (TTC reserved)	23	L1ID(23)	23	control channel	23	CRC (9-bit)
24	0 (TTC reserved)	24	ECRID(0)	24	control channel	24	CRC (9-bit)
25	0 (TTC reserved)	25	ECRID(1)	25	control channel	25	CRC (9-bit)
26	0 (TTC reserved)	26	ECRID(2)	26	control channel	26	CRC (9-bit)
27	0 (TTC reserved)	27	ECRID(3)	27	control channel	27	CRC (9-bit)
28	0 (TTC reserved)	28	ECRID(4)	28	control channel	28	CRC (9-bit)
29	0 (TTC reserved)	29	ECRID(5)	29	control channel	29	CRC (9-bit)
30	0 (TTC reserved)	30	ECRID(6)	30	control channel	30	CRC (9-bit)
31	0 (TTC reserved)	31	ECRID(7)	31	control channel	31	CRC (9-bit)

Full spec at: Specification for Readout Control & Combined TTC Serial links in L1Calo

Combined_TTC/DATA
Link placement

Links between each FEX slot and HUB



Hub - GTY Transceivers - QUADs 124:133

1
133

Other_Hub_RO	AL_0	MGT_FO_37	PF	Rx 3	Tx 3	ST	Combined Data to FEX 08
Other_Hub_RO	AL_1	MGT_FO_38	ST	Rx 2	Tx 2	ST	Combined Data to FEX 07
FEX_9	AL_0	MGT_FO_39	ST	Rx 1	Tx 1	ST	N.C.
FEX_9	AL_1	MGT_FO_40	ST	Rx 0	Tx 0	ST	N.C.

* 1
132

FEX_7	AL_0	MGT_FO_25	PF	Rx 3	Tx 3	ST	Combined Data to FEX 06
FEX_7	AL_1	MGT_FO_26	PF	Rx 2	Tx 2	ST	N.C.
FEX_7	AL_2	MGT_FO_27	PF	Rx 1	Tx 1	ST	N.C.
FEX_7	AL_3	MGT_FO_28	PF	Rx 0	Tx 0	ST	N.C.

1
131

FEX_8	AL_4	MGT_FO_35	PF	Rx 3	Tx 3	ST	N.C.
FEX_8	AL_5	MGT_FO_36	PF	Rx 2	Tx 2	ST	N.C.
FEX_7	AL_4	MGT_FO_29	PF	Rx 1	Tx 1	ST	N.C.
FEX_7	AL_5	MGT_FO_30	PF	Rx 0	Tx 0	ST	N.C.

* 1
130

FEX_8	AL_0	MGT_FO_31	PF	Rx 3	Tx 3	ST	N.C.
FEX_8	AL_1	MGT_FO_32	PF	Rx 2	Tx 2	ST	Combined Data to FEX 05
FEX_5	AL_4	MGT_FO_17	PF	Rx 1	Tx 1	ST	N.C.
FEX_5	AL_5	MGT_FO_18	PF	Rx 0	Tx 0	ST	Combined Data to FEX 04

1
129

FEX_6	AL_0	MGT_FO_19	PF	Rx 3	Tx 3	ST	N.C.
FEX_6	AL_1	MGT_FO_20	PF	Rx 2	Tx 2	ST	Combined Data to FEX 03
FEX_6	AL_2	MGT_FO_21	PF	Rx 1	Tx 1	ST	N.C.
FEX_6	AL_3	MGT_FO_22	PF	Rx 0	Tx 0	ST	Combined Data to Other Hub

Hub - GTH Transceivers - QUADs 224:233

1
233

FEX_10	AL_3	MGT_FO_48	ST	Rx 3	Tx 3	PF	Combined Data to FEX 09
FEX_10	AL_2	MGT_FO_47	PF	Rx 2	Tx 2	PF	Combined Data to FEX 10
FEX_10	AL_1	MGT_FO_46	PF	Rx 1	Tx 1	ST	N.C.
FEX_10	AL_0	MGT_FO_45	PF	Rx 0	Tx 0	ST	N.C.

* 1
232

FEX_9	AL_5	MGT_FO_44	PF	Rx 3	Tx 3	PF	Combined Data to FEX 11
FEX_9	AL_4	MGT_FO_43	PF	Rx 2	Tx 2	ST	N.C.
FEX_9	AL_3	MGT_FO_42	PF	Rx 1	Tx 1	ST	N.C.
FEX_9	AL_2	MGT_FO_41	PF	Rx 0	Tx 0	ST	N.C.

1
231

FEX_8	AL_3	MGT_FO_34	ST	Rx 3	Tx 3	ST	N.C.
FEX_8	AL_2	MGT_FO_33	ST	Rx 2	Tx 2	ST	N.C.
FEX_11	AL_5	MGT_FO_56	PF	Rx 1	Tx 1	ST	N.C.
FEX_11	AL_4	MGT_FO_55	PF	Rx 0	Tx 0	ST	N.C.

* 1
230

FEX_11	AL_3	MGT_FO_54	PF	Rx 3	Tx 3	ST	N.C.
FEX_11	AL_2	MGT_FO_53	PF	Rx 2	Tx 2	PF	Combined Data to FEX 12
FEX_11	AL_1	MGT_FO_52	PF	Rx 1	Tx 1	ST	N.C.
FEX_11	AL_0	MGT_FO_51	PF	Rx 0	Tx 0	PF	Combined Data to FEX 13

1
229

FEX_10	AL_5	MGT_FO_50	PF	Rx 3	Tx 3	ST	N.C.
FEX_10	AL_4	MGT_FO_49	PF	Rx 2	Tx 2	PF	Combined Data to FEX 14
FEX_13	AL_1	MGT_FO_64	PF	Rx 1	Tx 1	ST	N.C.
FEX_13	AL_0	MGT_FO_63	PF	Rx 0	Tx 0	ST	Hub Readout AL_0 to This ROD

0
228

FEX_12	AL_5	MGT_FO_62	PF	Rx 3	Tx 3	ST	N.C.
FEX_12	AL_4	MGT_FO_61	PF	Rx 2	Tx 2	ST	Hub Readout AL_1 to This ROD
FEX_12	AL_3	MGT_FO_60	PF	Rx 1	Tx 1	ST	N.C.
FEX_12	AL_2	MGT_FO_59	PF	Rx 0	Tx 0	ST	Combined Data to This ROD

Link between two HUBs

Link between the ROD and HUB

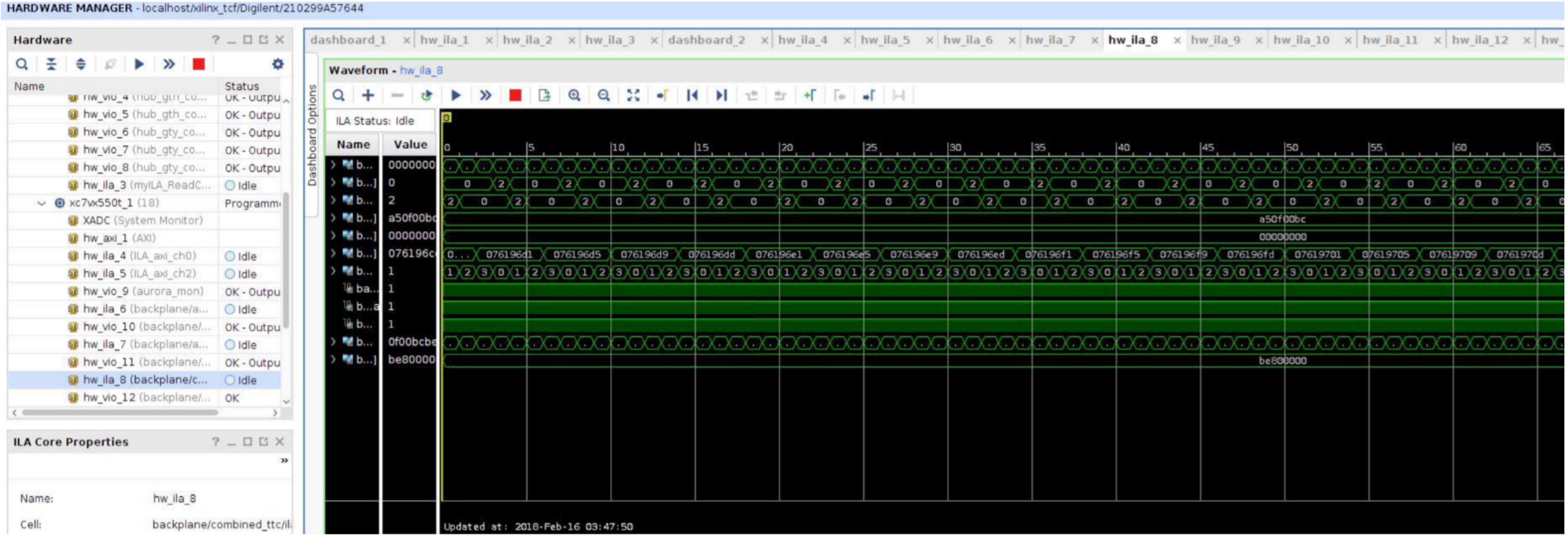
Combined_TTC/DATA FW development strategy

- Combined_TTC/DATA Firmware development comprises several stages
- First stage assumes to transmit the Combined_TTC/Data stream to the ROD module, also to the FEX slot 3
- In order to debug the design, the standard Xilinx diagnostic components are being used to monitor the data flow (as for example like the ILA and VIO).
- The physical layer is configured with the use of GT wizard
- For the purpose of the initial test, the HUB transmitter side generates the 128 bits from 4 Control registers but only some static patterns are written into these registers
- In the next step, the Readout_Ctrl data received on the HUB from ROD module are retransmitted as the Combined_TTC/Data stream to the ROD and FTM module in slot 3
- Once the communication between the HUB and other modules within shelf is established, a test pattern generator will be replaced by real TTC component
- Next development steps assumes to add (gradually) the remaining receivers in the shelf

Combined_TTC/DATA FW development strategy

- Combined_TTC Data link test:

- **A correct Combined_TTC stream on the ROD**



HARDWARE MANAGER - localhost/xilinx_tcf/Digilent/210299A57644

Hardware

Name	Status
hw_vio_4 (hub_gth_co...	OK - Outpu
hw_vio_5 (hub_gth_co...	OK - Outpu
hw_vio_6 (hub_gty_co...	OK - Outpu
hw_vio_7 (hub_gty_co...	OK - Outpu
hw_vio_8 (hub_gty_co...	OK - Outpu
hw_ila_3 (myILA_ReadC...	○ Idle
xc7vx550t_1 (18)	Programm
XADC (System Monitor)	
hw_axi_1 (AXI)	
hw_ila_4 (ILA_axi_ch0)	○ Idle
hw_ila_5 (ILA_axi_ch2)	○ Idle
hw_vio_9 (aurora_mon)	OK - Outpu
hw_ila_6 (backplane/a...	○ Idle
hw_vio_10 (backplane/...	OK - Outpu
hw_ila_7 (backplane/a...	○ Idle
hw_vio_11 (backplane/...	OK - Outpu
hw_ila_8 (backplane/c...	○ Idle
hw_vio_12 (backplane/...	OK

ILA Core Properties

Name: hw_ila_8

Cell: backplane/combined_ttc/il

Waveform - hw_ila_8

ILA Status: Idle

Name	Value
b...	0000000
b...]	0
b...	2
b...]	a50f00bc
b...]	0000000
b...]	076196c
b...	1
ba...	1
b...a	1
b...	1
b...	0f00bcbe
b...]	be80000

Updated at: 2018-Feb-16 03:47:50

Readout_Control overview

- The HUB module is obliged to receive the Readout Control (Readout_CTRL) information from the ROD via serial link named Readout_CTRL. The HUB module is the only one module within the shelf which gets the Readout Control data from the ROD

- That information is used by the Hub, also fanned out to the rest of the system. The main purpose is to provide resets to all of the data links (Aurora) between the Fex's and the ROD plus HUB module

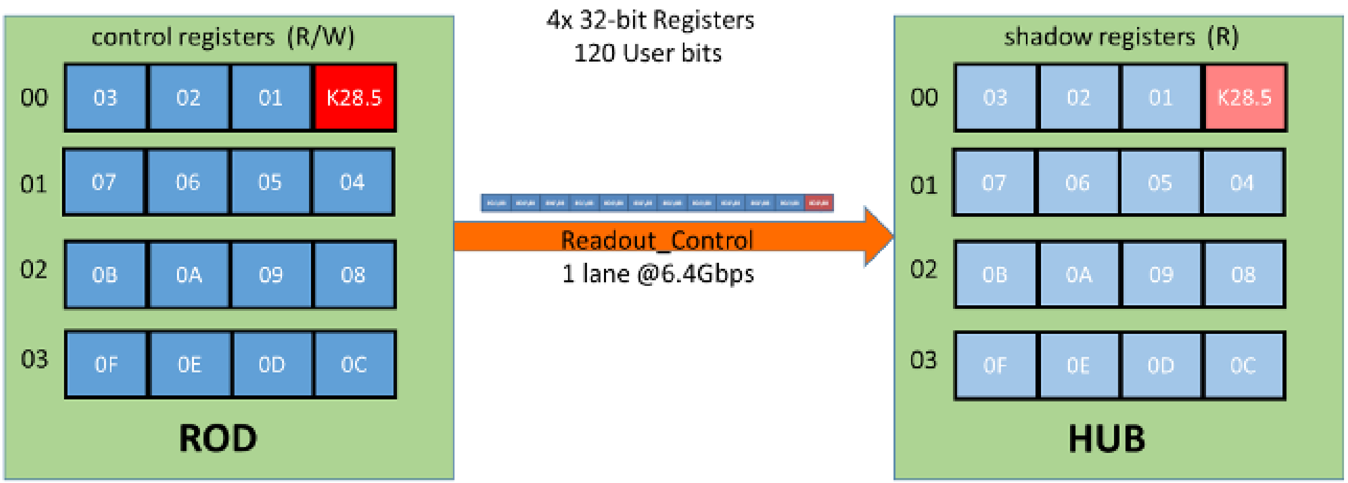
Full spec at: Specification for Readout Control & Combined TTC Serial links in L1Calo

Readout_Control Physical Implementation

- The Readout_Ctrl link on the HUB module is implemented with the use of MGT Transceiver GTH, control and diagnostic logic
- The Readout_Ctrl link is designed to operate at 6.4 Gbps.
- In order to to control message transmission within a single LHC clock period, the length of the message is limited to 128 bits
- The HUB Readout Control FW features one receiver (RX). In order to debug the design, the standard Xilinx diagnostic components are being used to monitor the data flow (as for example like the ILA and VIO).
- The physical layer is configured with the use of GT wizard
- The physical implementation of the Readout_CTRL links assumes that there are Control Registers on the Tx ROD side, and Shadow Registers on the Rx HUB side. The transmitter side generates the 128 bit message from 4 Control registers: Word_0, Word_1, Word_2, and Word_3. The transmitter side logic is in charge to write control information into these registers for transmission to the modules within the shelf. These registers are continuously transmitted to the HUB which receives the data into a duplicate set of 4 registers referred to as Shadow Registers. Anything written into a Control Register at the transmitter side will appear in the corresponding Shadow register at the receiving side within the following LHC clock.

Readout_Control Physical Implementation

There are Control Registers on the TX side and Shadow Register on the RX side



control registers (R/W)

00	03	02	01	K28.5
01	07	06	05	04
02	0B	0A	09	08
03	0F	0E	0D	0C

ROD

shadow registers (R)

00	03	02	01	K28.5
01	07	06	05	04
02	0B	0A	09	08
03	0F	0E	0D	0C

HUB

Figure 1: Control and Shadow Registers

Readout_Control Bit definition

- The bits within the Readout Control words are primarily defined to provide initialization functions for all of the FEX data links

- **Comma Character**
  #Bits 7 to 0 of Word_0 contain the Comma character that maintains alignment of the 4 shadow registers with their corresponding Control registers. The chosen character is K28.5 = 0xBC

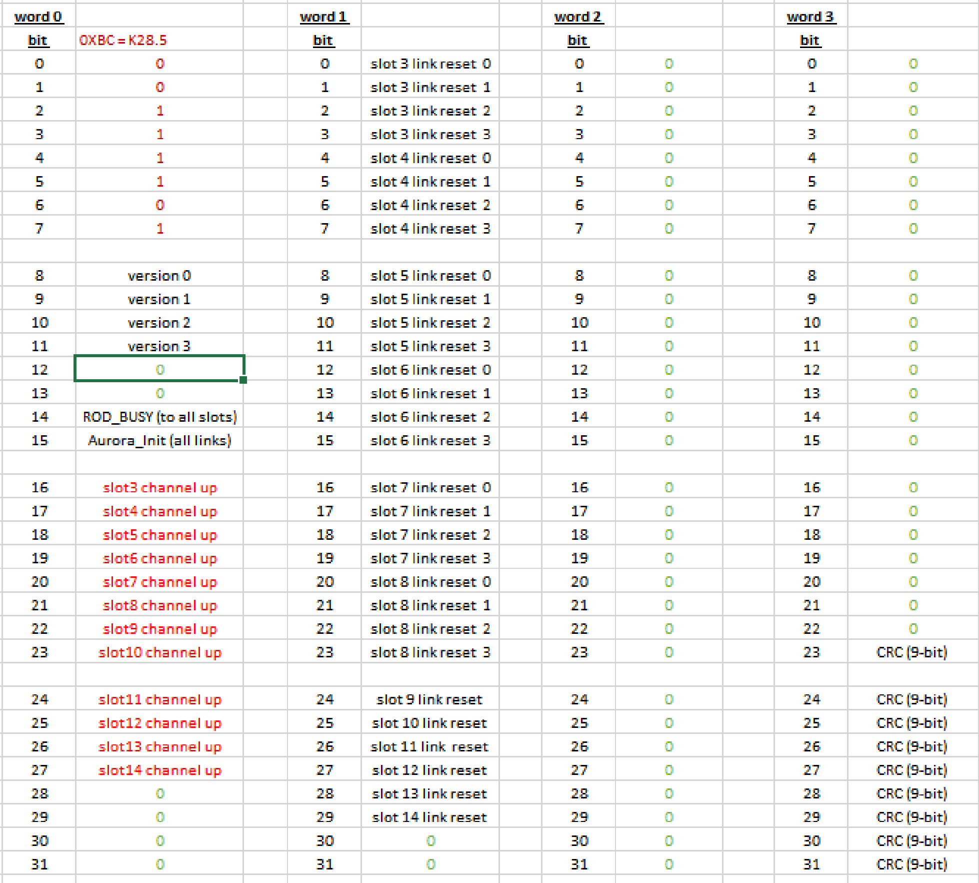
- **Version**
  #The 4-bit value contains the version number of this overall bit assignment

- **ROD_BUSY**
  #When active, this signal indicates that the ROD cannot currently accept further data from Fex sources. This signal is fanned-out to the shelf FEX's by the Hub via the Combined_TTC link

- **Global_Link_Reset**
  #This single bit is used to reset all of the data (Aurora) links within the shelf. The primary use is in the first initialisation after power-up. The ROD can hold this reset active for an indefinite amount of time. On the trailing edge (deactivation), the eFex's should provide additional timing control for the GTReset and Reset signals on the Aurora interface. This signal is fanned-out to the shelf FEX's by the Hub via the Combined_TTC link

Readout_Control
Bit definition

- The bits within the Readout Control words are primarily defined to provide initialization functions for all of the FEX data links



word 0		word 1		word 2		word 3	
bit	0XBC = K28.5	bit		bit		bit	
0	0	0	slot 3 link reset 0	0	0	0	0
1	0	1	slot 3 link reset 1	1	0	1	0
2	1	2	slot 3 link reset 2	2	0	2	0
3	1	3	slot 3 link reset 3	3	0	3	0
4	1	4	slot 4 link reset 0	4	0	4	0
5	1	5	slot 4 link reset 1	5	0	5	0
6	0	6	slot 4 link reset 2	6	0	6	0
7	1	7	slot 4 link reset 3	7	0	7	0
8	version 0	8	slot 5 link reset 0	8	0	8	0
9	version 1	9	slot 5 link reset 1	9	0	9	0
10	version 2	10	slot 5 link reset 2	10	0	10	0
11	version 3	11	slot 5 link reset 3	11	0	11	0
12	0	12	slot 6 link reset 0	12	0	12	0
13	0	13	slot 6 link reset 1	13	0	13	0
14	ROD_BUSY (to all slots)	14	slot 6 link reset 2	14	0	14	0
15	Aurora_Init (all links)	15	slot 6 link reset 3	15	0	15	0
16	slot3 channel up	16	slot 7 link reset 0	16	0	16	0
17	slot4 channel up	17	slot 7 link reset 1	17	0	17	0
18	slot5 channel up	18	slot 7 link reset 2	18	0	18	0
19	slot6 channel up	19	slot 7 link reset 3	19	0	19	0
20	slot7 channel up	20	slot 8 link reset 0	20	0	20	0
21	slot8 channel up	21	slot 8 link reset 1	21	0	21	0
22	slot9 channel up	22	slot 8 link reset 2	22	0	22	0
23	slot10 channel up	23	slot 8 link reset 3	23	0	23	CRC (9-bit)
24	slot11 channel up	24	slot 9 link reset	24	0	24	CRC (9-bit)
25	slot12 channel up	25	slot 10 link reset	25	0	25	CRC (9-bit)
26	slot13 channel up	26	slot 11 link reset	26	0	26	CRC (9-bit)
27	slot14 channel up	27	slot 12 link reset	27	0	27	CRC (9-bit)
28	0	28	slot 13 link reset	28	0	28	CRC (9-bit)
29	0	29	slot 14 link reset	29	0	29	CRC (9-bit)
30	0	30	0	30	0	30	CRC (9-bit)
31	0	31	0	31	0	31	CRC (9-bit)

Full spec at: Specification for Readout Control & Combined TTC Serial links in L1Calo

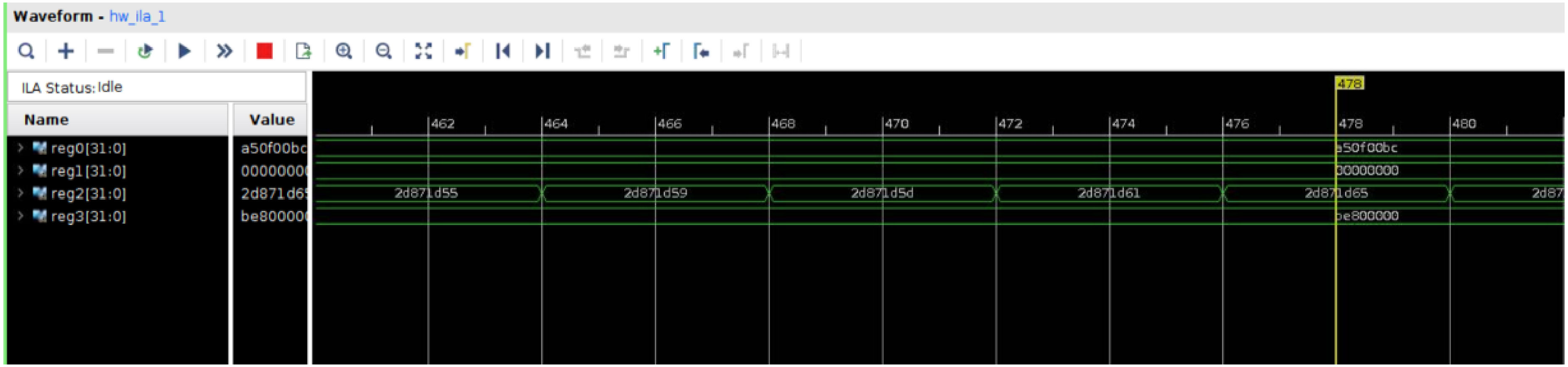
Readout_Control
FW development strategy

- The HUB Readout Control FW features one receiver (RX). In order to debug the design, the standard Xilinx diagnostic components are being used to monitor the data flow (as for example like the ILA and VIO). The physical layer is configured with the use of GT wizard.

Readout_Control
FW development strategy

Probing the aligned data from the GT

Expected Results
Word_0: constant a50f00bc
Word_1: constant 00000000
Word_2: counter value
Word_3: constant be800000



Waveform - hw_ila_1

ILA Status: Idle

Name	Value
reg0[31:0]	a50f00bc
reg1[31:0]	0000000
reg2[31:0]	2d871d6
reg3[31:0]	be80000

Physical implementation of the Readout_CTRL link is similar to the Combined_TTC links. There are Control Registers on the TX side and Shadow Register on the RX side

HUB Safe Configuration (1)

- During the regular operation, the HUB firmware is obliged to control the group of signals which are wired to the FPGA. These signals are handled on the HUB FPGA by the Safe Configuration component. This piece of firmware is in charge to properly receive the signals and control them by the ILA and VIO component. The HUB Safe Configuration needs to present in any type of HUB configuration

- Signal types →See Table 3 (HUB Firmware Specification)

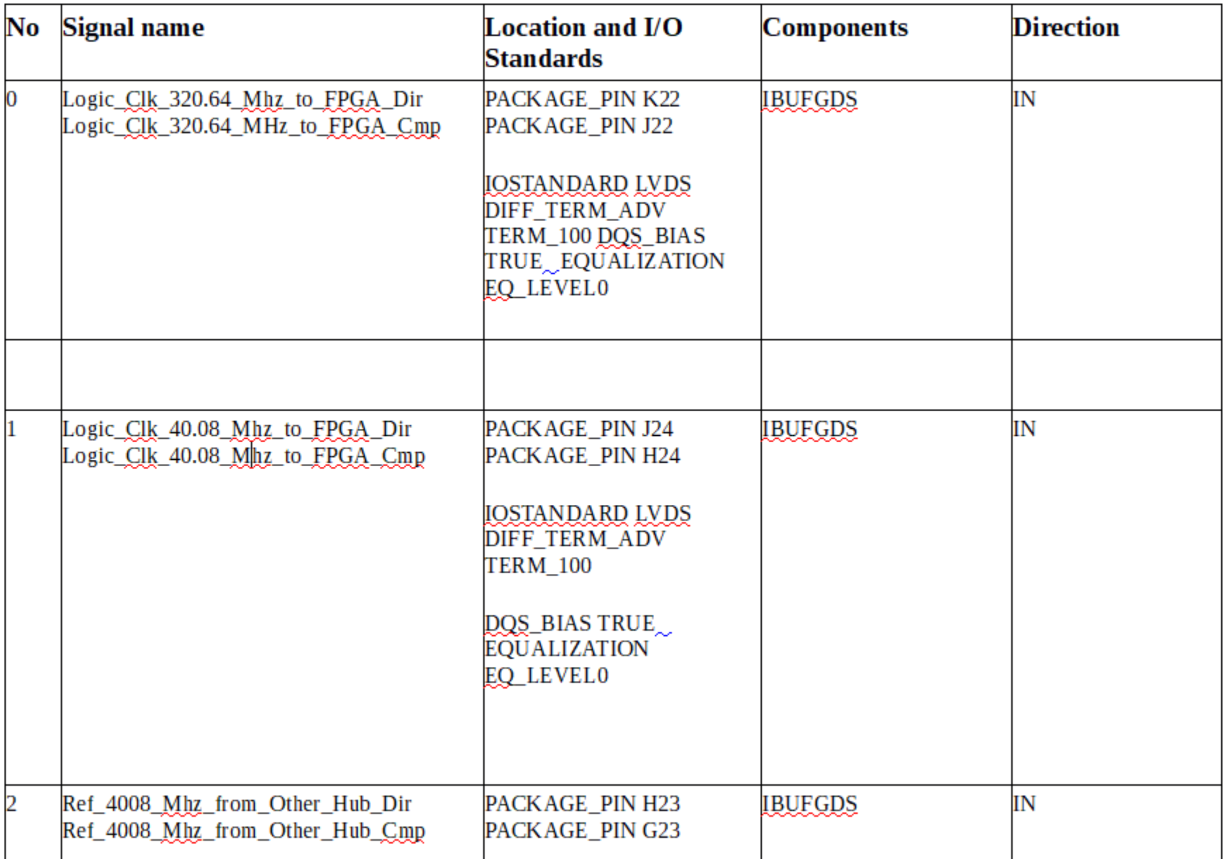
  Logic Analyzer (ILA). The customizable Integrated Logic Analyzer (ILA) IP core is a logic analyzer core that can be used to monitor the internal signals of a design.

- Virtual Input/Output (VIO). The LogiCORETM IP Virtual Input/Output (VIO) core is a customizable core that can both monitor and drive internal FPGA signals in real time

- SYSMON. Optionally, the HUB configuration can include the SYSMON (SLR0 and SLR1): Each super logic region: SLR0 and SLR1 has one system monitor to provide for monitoring supply voltages within the SLR. The I2C DRP and JTAG DRP access is limited to the master SLR only (SYSMONE1_X0Y0 for devices with two SLRs).

HUB Safe Configuration (2)

HUB Signal types →See Table 3 (HUB Firmware Specification)

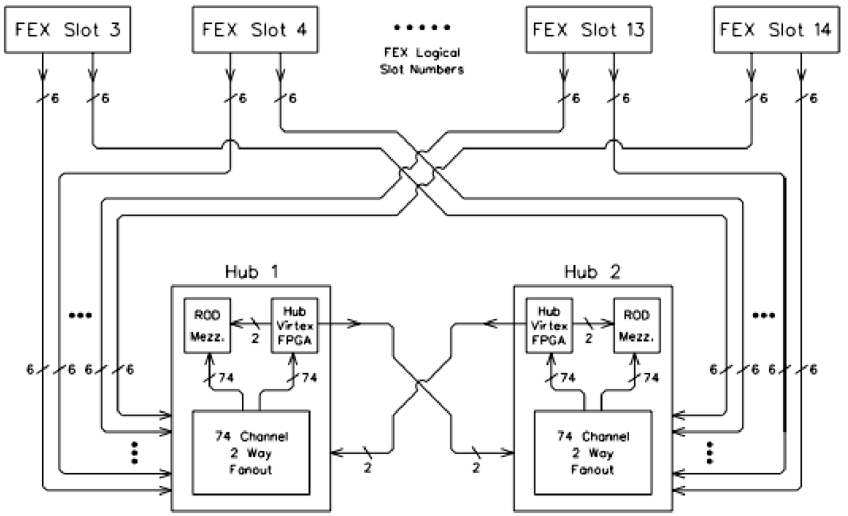


No	Signal name	Location and I/O Standards	Components	Direction
0	Logic_Clk_320.64_Mhz_to_FPGA_Dir Logic_Clk_320.64_MHz_to_FPGA_Cmp	PACKAGE_PIN K22 PACKAGE_PIN J22 IOSTANDARD LVDS DIFF_TERM_ADV TERM_100 DQS_BIAS TRUE EQUALIZATION EQ_LEVEL0	IBUFGDS	IN
1	Logic_Clk_40.08_Mhz_to_FPGA_Dir Logic_Clk_40.08_Mhz_to_FPGA_Cmp	PACKAGE_PIN J24 PACKAGE_PIN H24 IOSTANDARD LVDS DIFF_TERM_ADV TERM_100 DQS_BIAS TRUE EQUALIZATION EQ_LEVEL0	IBUFGDS	IN
2	Ref_4008_Mhz_from_Other_Hub_Dir Ref_4008_Mhz_from_Other_Hub_Cmp	PACKAGE_PIN H23 PACKAGE_PIN G23	IBUFGDS	IN

Readout (Aurora8b10b) - overview

- This figure shows the Hub's distribution of readout data in the context of the cards in the ATCA shelf. All of this data flows to both the ROD and to the FPGA on each Hub. It supports 2 independent streams of readout data. That is, the readout stream processed by the ROD and Hub FPGA on Hub-1 can be Independent of the readout stream flowing into Hub-2.

Hub-Module Readout Data Distribution



Each line indicates six Fabric Interface Channels carrying FEX Slot Readout Data to the ROD's and to the Hub-Module's Virtex FPGAs

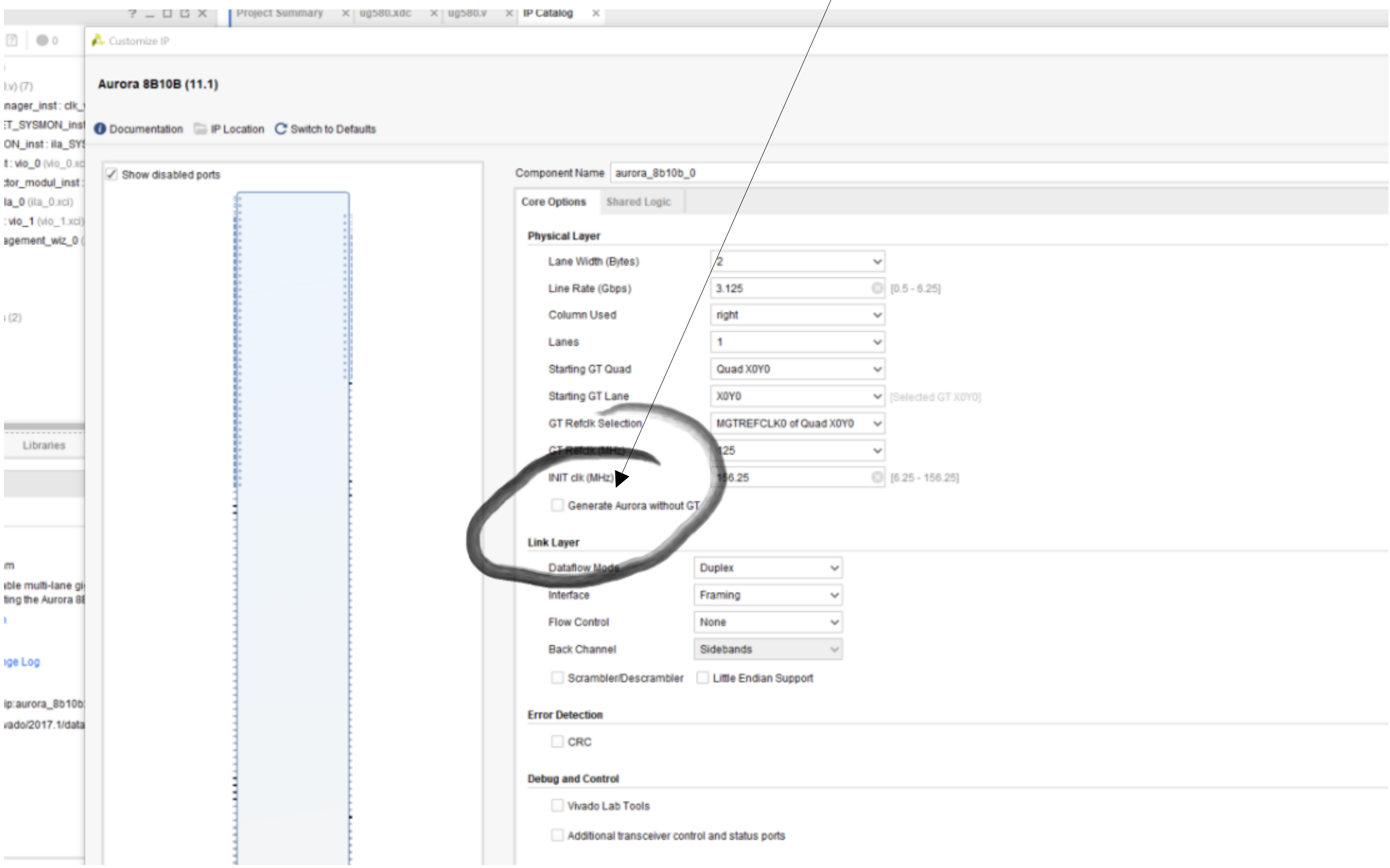
Hub to Hub Links use the Fabric Interface.

The FEX Slot Readout Data on each Fabric Interface Channel is received by an MGT Transceiver on both the ROD's and the Hub-Module's Virtex FPGAs.

Drw: 2 Rev. 15-Feb-2016

Readout (Aurora8b10b) Implementation method

- Aurora IP core (without GT) to generate the Aurora protocol files
- GT Wizard to configure the physical layer
- User merging process



Aurora 8B10B (11.1)

Documentation | IP Location | Switch to Defaults

☑ Show disabled ports

Component Name: aurora_8b10b_0

Core Options | Shared Logic

Physical Layer

Lane Width (Bytes)	2	
Line Rate (Gbps)	3.125	[0.5 - 6.25]
Column Used	right	
Lanes	1	
Starting GT Quad	Quad X0Y0	
Starting GT Lane	X0Y0	[Selected GT X0Y0]
GT Refclk Selection	MGTREFCLK0 of Quad X0Y0	
GT Refclk (MHz)	125	
INIT clk (MHz)	156.25	[6.25 - 156.25]

□ Generate Aurora without GT

Link Layer

Dataflow Mode	Duplex
Interface	Framing
Flow Control	None
Back Channel	Sidebands

□ Scrambler/Descrambler □ Little Endian Support

Error Detection

□ CRC

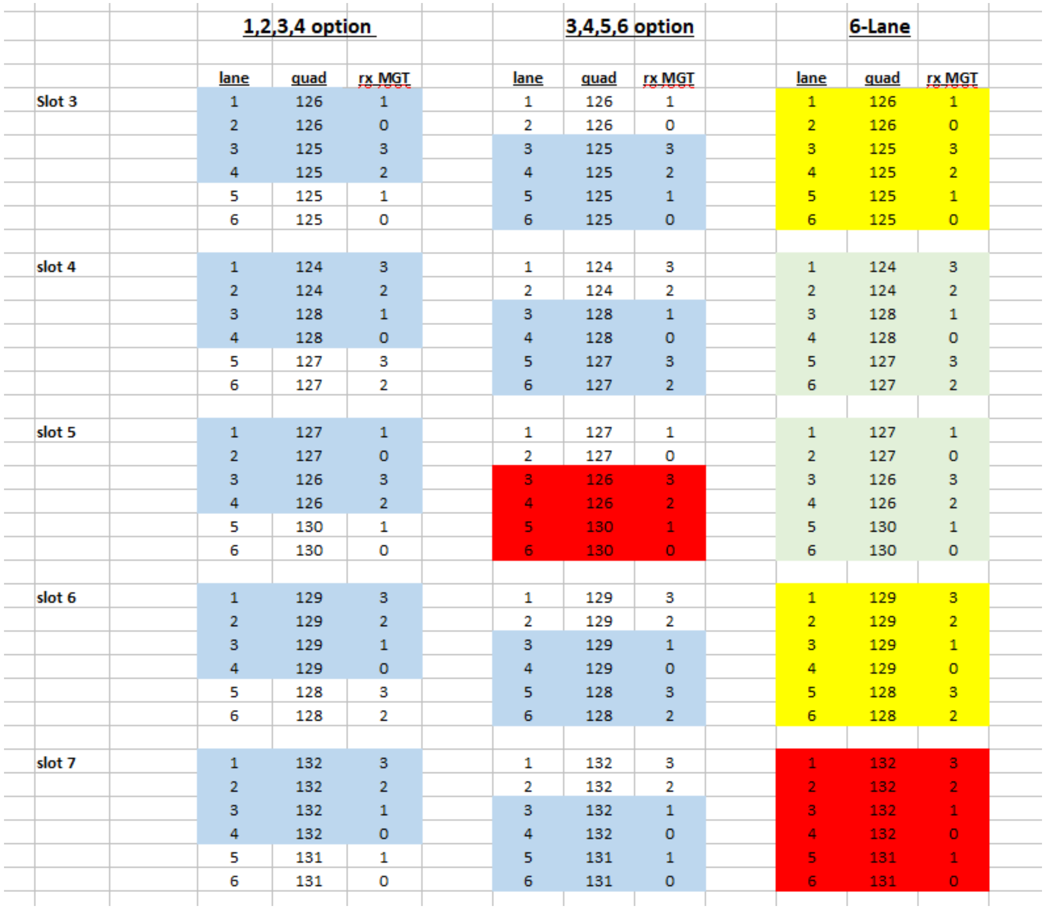
Debug and Control

□ Vivado Lab Tools

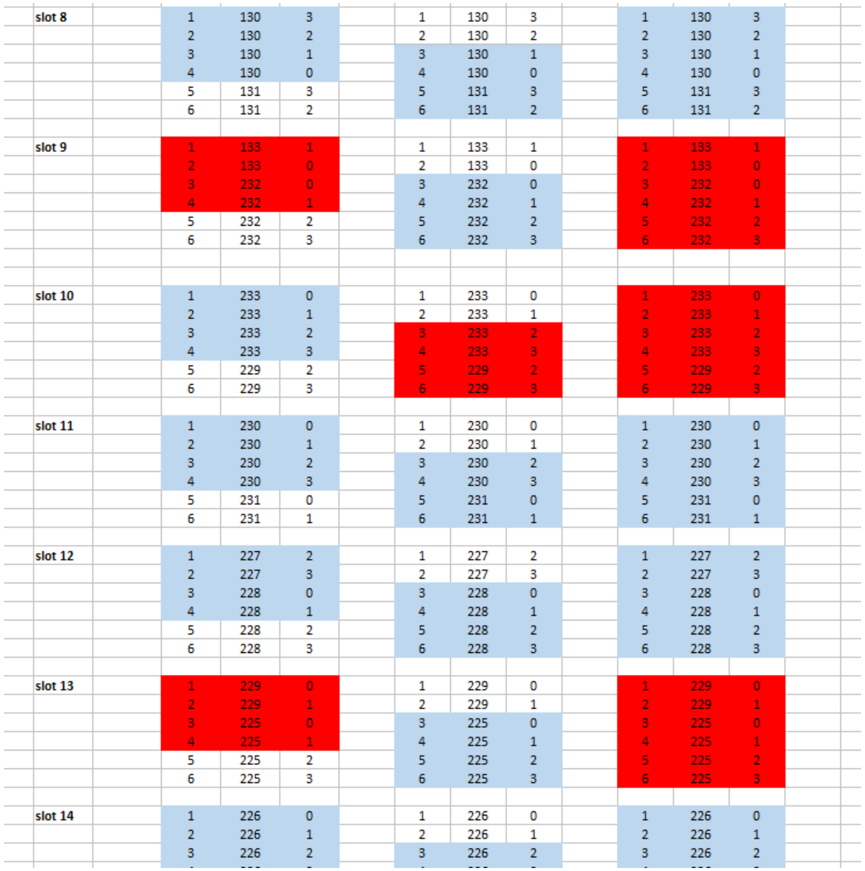
□ Additional transceiver control and status ports

Readout (Aurora8b10b)
Aurora Line Mapping

Option A: 1,2,3,4
Option B: 3,4,5,6 (mix of GTY and GTH)
Option C: 6-lane



	1,2,3,4 option			3,4,5,6 option			6-Lane		
	lane	quad	rx MGT	lane	quad	rx MGT	lane	quad	rx MGT
Slot 3	1	126	1	1	126	1	1	126	1
	2	126	0	2	126	0	2	126	0
	3	125	3	3	125	3	3	125	3
	4	125	2	4	125	2	4	125	2
	5	125	1	5	125	1	5	125	1
	6	125	0	6	125	0	6	125	0
slot 4	1	124	3	1	124	3	1	124	3
	2	124	2	2	124	2	2	124	2
	3	128	1	3	128	1	3	128	1
	4	128	0	4	128	0	4	128	0
	5	127	3	5	127	3	5	127	3
	6	127	2	6	127	2	6	127	2
slot 5	1	127	1	1	127	1	1	127	1
	2	127	0	2	127	0	2	127	0
	3	126	3	3	126	3	3	126	3
	4	126	2	4	126	2	4	126	2
	5	130	1	5	130	1	5	130	1
	6	130	0	6	130	0	6	130	0
slot 6	1	129	3	1	129	3	1	129	3
	2	129	2	2	129	2	2	129	2
	3	129	1	3	129	1	3	129	1
	4	129	0	4	129	0	4	129	0
	5	128	3	5	128	3	5	128	3
	6	128	2	6	128	2	6	128	2
slot 7	1	132	3	1	132	3	1	132	3
	2	132	2	2	132	2	2	132	2
	3	132	1	3	132	1	3	132	1
	4	132	0	4	132	0	4	132	0
	5	131	1	5	131	1	5	131	1
	6	131	0	6	131	0	6	131	0



slot 8	1	130	3	1	130	3	1	130	3
	2	130	2	2	130	2	2	130	2
	3	130	1	3	130	1	3	130	1
	4	130	0	4	130	0	4	130	0
	5	131	3	5	131	3	5	131	3
	6	131	2	6	131	2	6	131	2
slot 9	1	133	1	1	133	1	1	133	1
	2	133	0	2	133	0	2	133	0
	3	232	0	3	232	0	3	232	0
	4	232	1	4	232	1	4	232	1
	5	232	2	5	232	2	5	232	2
	6	232	3	6	232	3	6	232	3
slot 10	1	233	0	1	233	0	1	233	0
	2	233	1	2	233	1	2	233	1
	3	233	2	3	233	2	3	233	2
	4	233	3	4	233	3	4	233	3
	5	229	2	5	229	2	5	229	2
	6	229	3	6	229	3	6	229	3
slot 11	1	230	0	1	230	0	1	230	0
	2	230	1	2	230	1	2	230	1
	3	230	2	3	230	2	3	230	2
	4	230	3	4	230	3	4	230	3
	5	231	0	5	231	0	5	231	0
	6	231	1	6	231	1	6	231	1
slot 12	1	227	2	1	227	2	1	227	2
	2	227	3	2	227	3	2	227	3
	3	228	0	3	228	0	3	228	0
	4	228	1	4	228	1	4	228	1
	5	228	2	5	228	2	5	228	2
	6	228	3	6	228	3	6	228	3
slot 13	1	229	0	1	229	0	1	229	0
	2	229	1	2	229	1	2	229	1
	3	225	0	3	225	0	3	225	0
	4	225	1	4	225	1	4	225	1
	5	225	2	5	225	2	5	225	2
	6	225	3	6	225	3	6	225	3
slot 14	1	226	0	1	226	0	1	226	0
	2	226	1	2	226	1	2	226	1
	3	226	2	3	226	2	3	226	2

Readout (Aurora8b10b) Background

- Aurora8b10b IP core supports: 7 series GTX/GTH, UltraScale™ GTH, UltraScale+™ GTH, GTP transceivers
- Officially the Aurora8b10b IP core does not support the GTY transceivers
- Service Request (SR, Xilinx) to resolve the issue

Readout (Aurora8b10b) Feedback from Xilinx (1)

- *Me: Is there any plan to upgrade the Aurora8b10b core to support the GTY transceivers?*
- Xilinx: I checked in for the internal resources to see if there is any plan in near future to support the Aurora8b10b protocol in UltraScale GTY transceivers but right now it is not yet planned.
- *Me: I can see that GT wizard offers the Aurora8b10b protocol for the GTY?*
- Xilinx: Even if the GT wizard offers Aurora8b10b protocol for the GTY transceivers, it will just customize the GT part of the protocol (not the complete protocol).

Readout (Aurora8b10b) Feedback from Xilinx (2)

- *Me: Is there any technical reason behind Aurora8b10b protocol not supporting GTY transceivers?*

- **Xilinx: There is not really technical reason behind Aurora8b10b protocol not supporting GTY transceivers**

  But since GTY transceivers are meant to be targeted for higher line rate, we recommend to use Aurora64b66b protocol for better throughput

Readout (Aurora8b10b) Feedback from Xilinx (3)

- Finally, we figured out the method to implement the Aurora8b10b protocol in GTY transceivers. The test design is based on the Xilinx example
- The design was sent to Xilinx for validation (including video and text documentation)
- Xilinx seemed to be pleased with our design. The design is stored in Xilinx internal database and it will be offered for anyone who wants to implement similar application in future

Readout (Aurora8b10b) GTY (test scope)

- Successfully tested, VCU108 (GTY), Xilinx dev board:

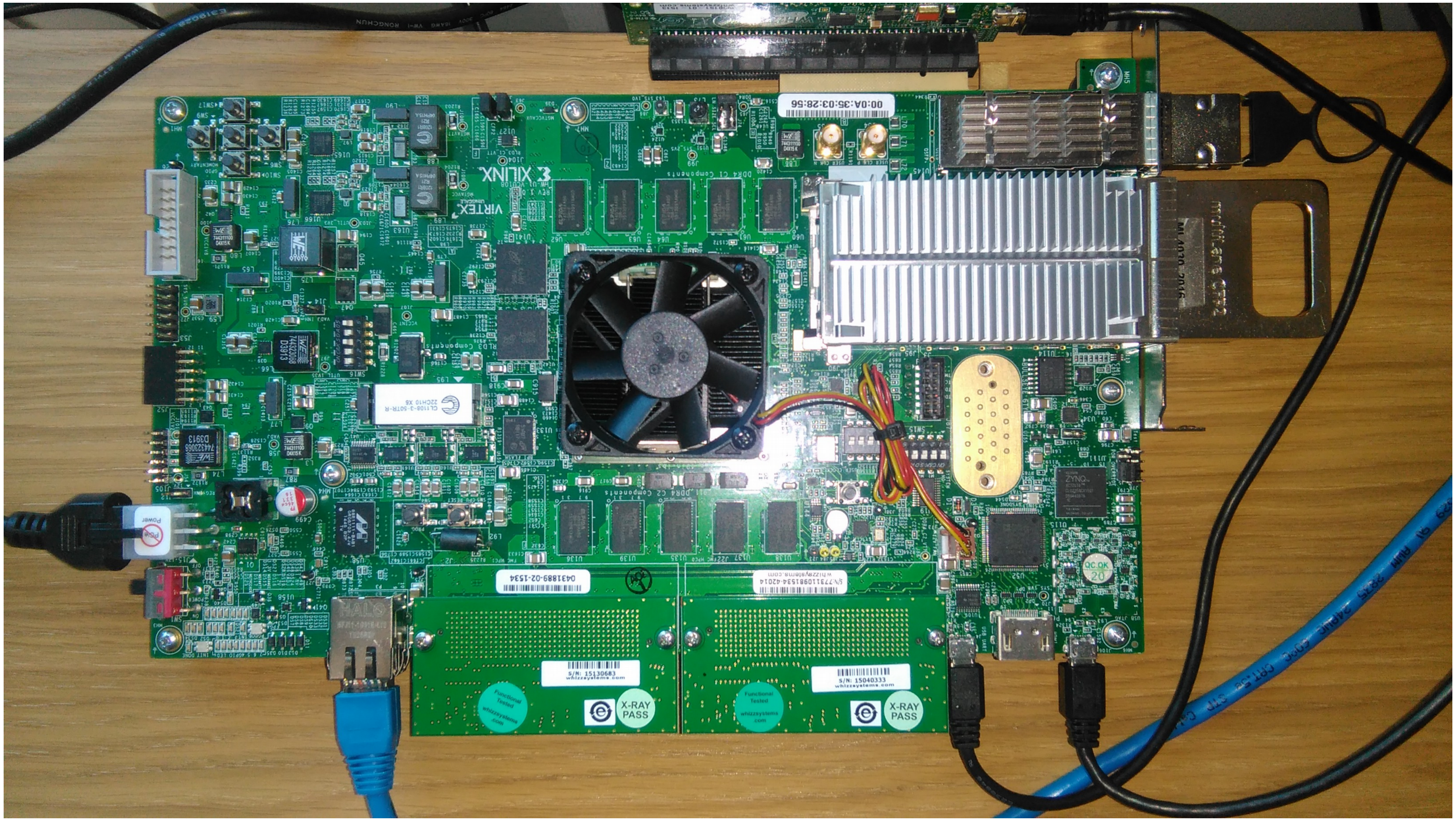
- GTY ↔ GTY (@5 Gbps, QSFP loopback module)
- GTY ↔ GTY (@6.4 Gbps, QSFP loopback module)

- Successfully tested, VCU108 (GTY) ↔ ZYNQ (GTX):

- GTY → GTX (@5 Gbps, QSFP-SFP passive splitter cable)
- GTX → GTY (@5 Gbps, QSFP-SFP passive splitter cable)

Readout (Aurora8b10b) Implementation in GTY, setup (1)

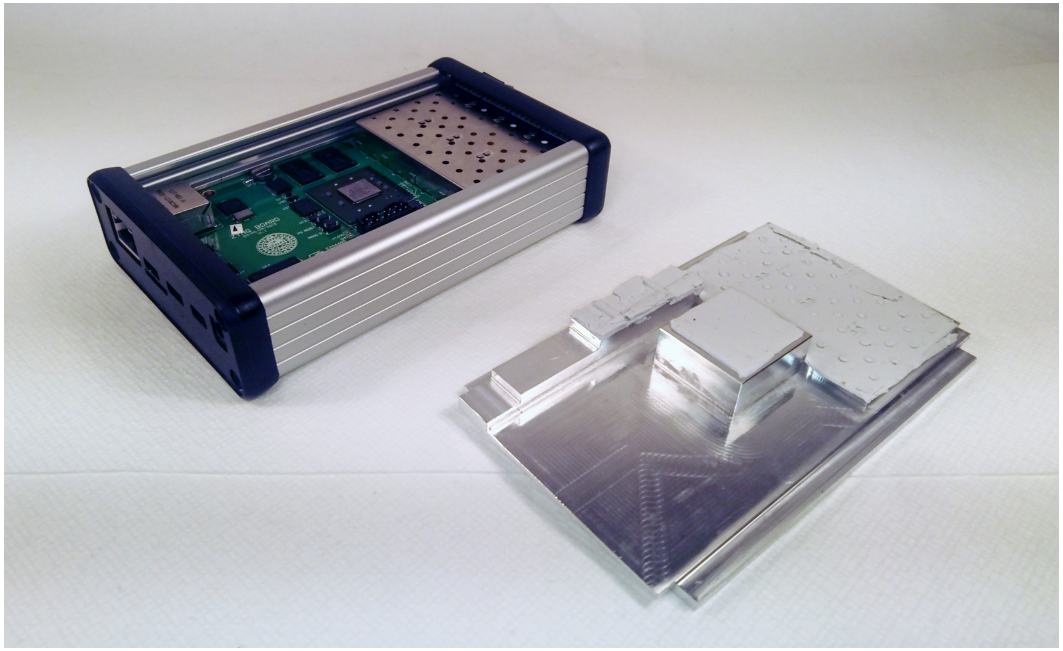
VCU108, Xilinx Evaluation Board:
GTH/GTY, QSFP → SFP passive splitter cable

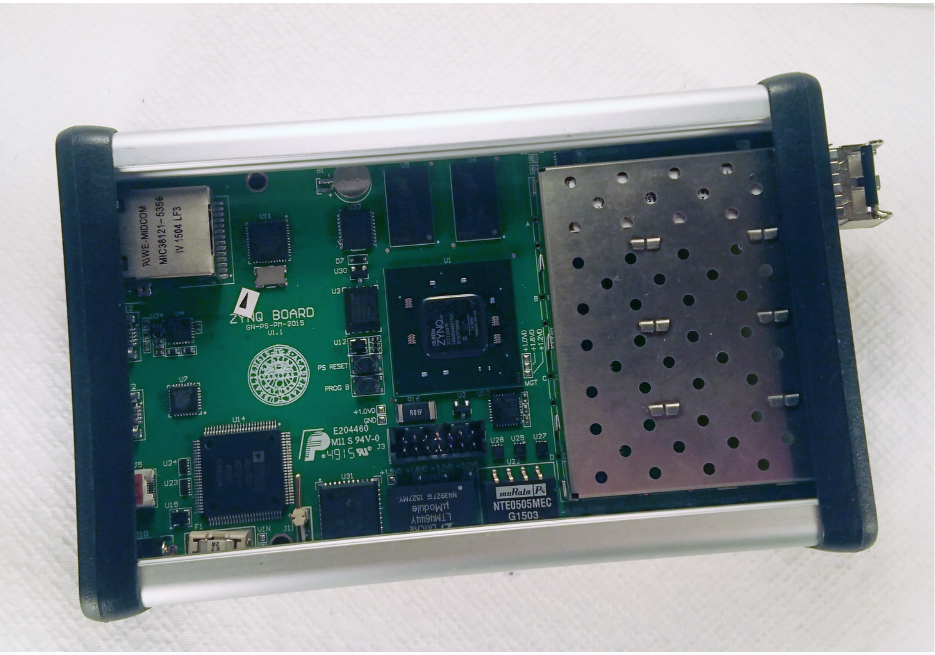


Readout (Aurora8b10b) Implementation in GTY, setup (2)

The ZYNQ board for Panda Experiment. Designed by Pawel Marciniewski (Uppsala University)

This board is a DAQ device featuring 4 optical interfaces with up to 6.6 Gbps bandwidth each.



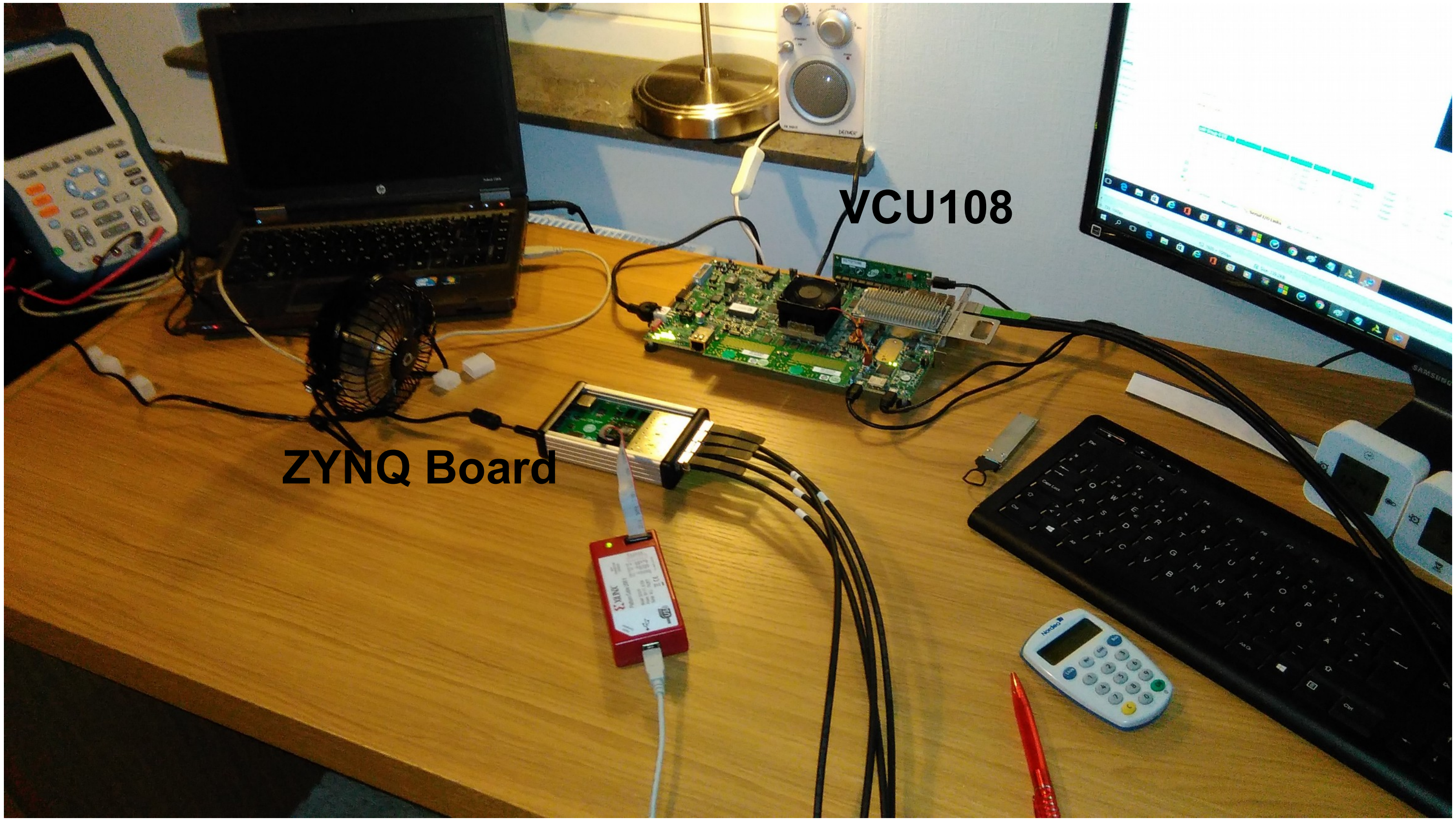


The device was tested with Ubuntu 14.4

The unit is powered by a ZYNQ XC7Z030 SoC
- 512 MB DDR3 RAM
- USB 2.0
- Uart port
- MicroSD
- GbE port
- HDMI
- 4 NIM I/O

Readout (Aurora8b10b) Implementation in GTY, setup (3)

VCU108 (GTY) ↔ ZYNQ (GTX), @5Gbps (1 line, RX/TX simplex, timer)
Diagnostic: ILA/VIO. Xilinx Data generator/checker.



Readout (Aurora8b10b) Implementation in GTY, setup (4)

Aurora8b10b Core configures two parts: link layer (protocol) and physical layer (GT part). In real, the Aurora8b10b core uses the GT wizard to configure the physical layer. Basically, two IP cores are being used to provide full setup. Thus, in order to Implement the Aurora8b10b in GTY, the GT wizard needs to be run “manually” to configure the GTY transceivers

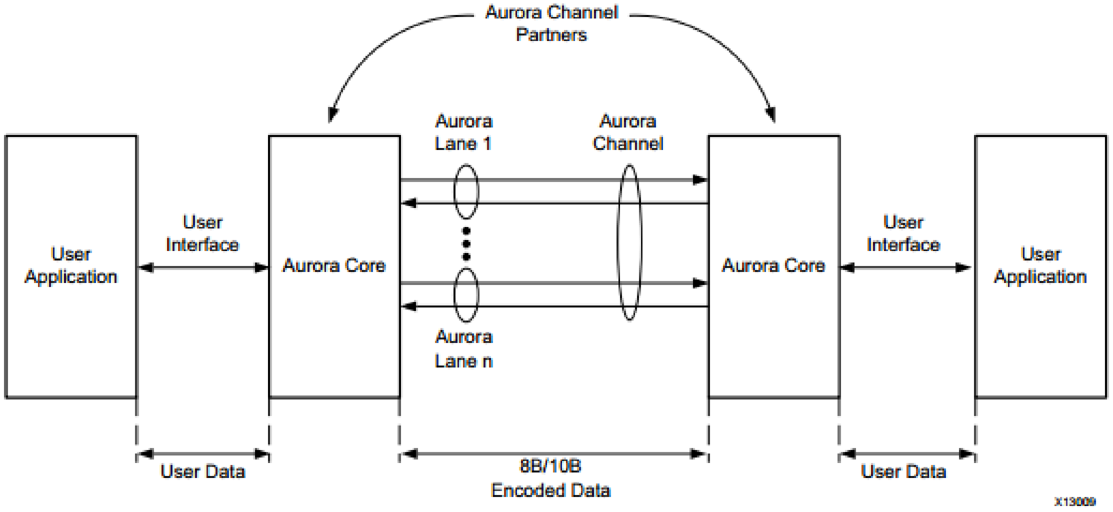
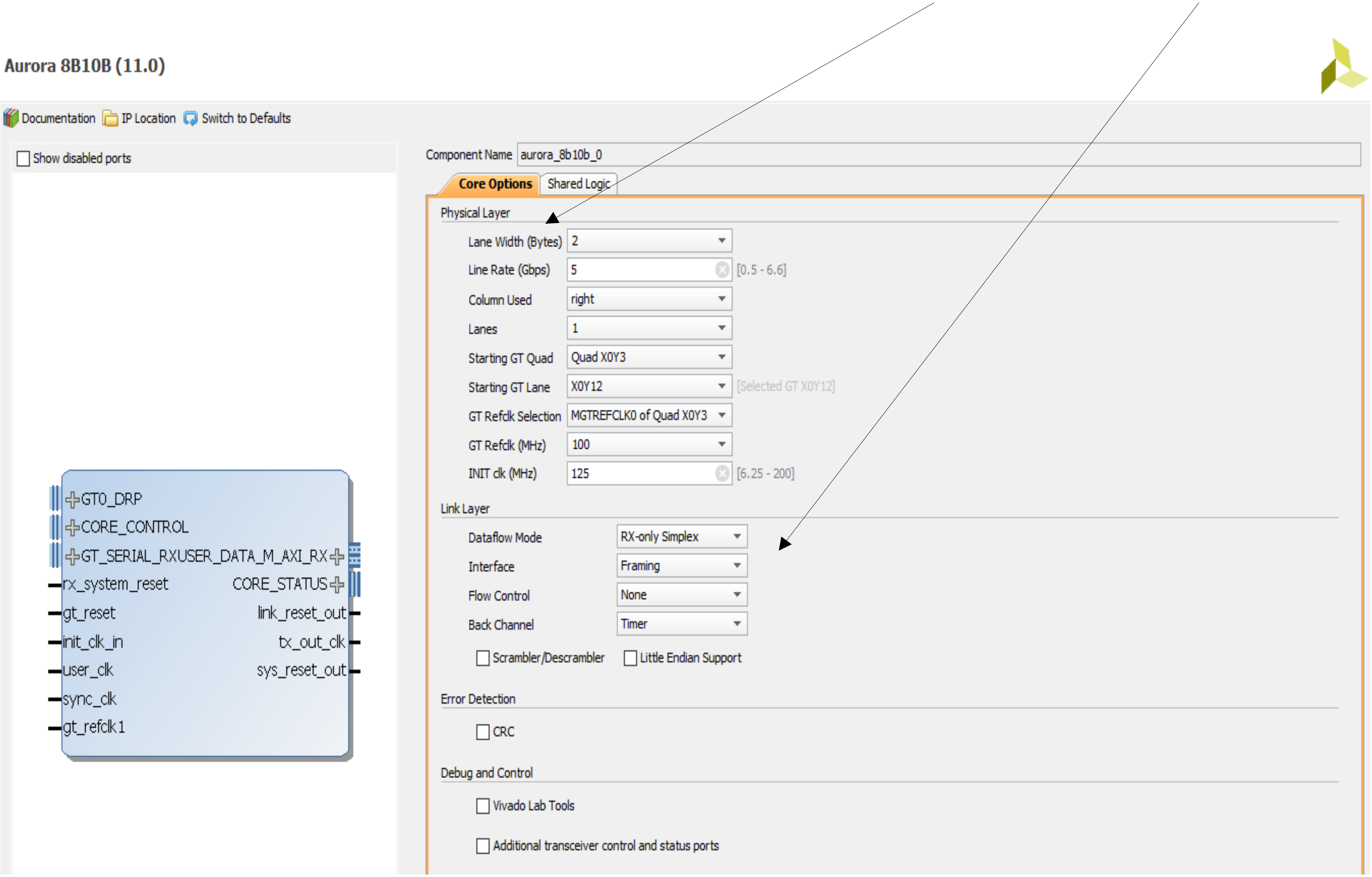


Figure: Aurora8b10b overview (source: Xilinx documentation)

Readout (Aurora8b10b) Implementation in GTY, recipe (1)

Standard Aurora8b10b IP core (GTH), configure the Physical and Link Layer.



Aurora 8B10B (11.0)

□ Show disabled ports

Component Name: aurora_8b10b_0

Physical Layer

Lane Width (Bytes)	2	
Line Rate (Gbps)	5	[0.5 - 6.6]
Column Used	right	
Lanes	1	
Starting GT Quad	Quad X0Y3	
Starting GT Lane	X0Y12	[Selected GT X0Y12]
GT Refclk Selection	MGTREFCLK0 of Quad X0Y3	
GT Refclk (MHz)	100	
INIT clk (MHz)	125	[6.25 - 200]

Link Layer

Dataflow Mode	RX-only Simplex
Interface	Framing
Flow Control	None
Back Channel	Timer

□ Scrambler/Descrambler □ Little Endian Support

Error Detection

□ CRC

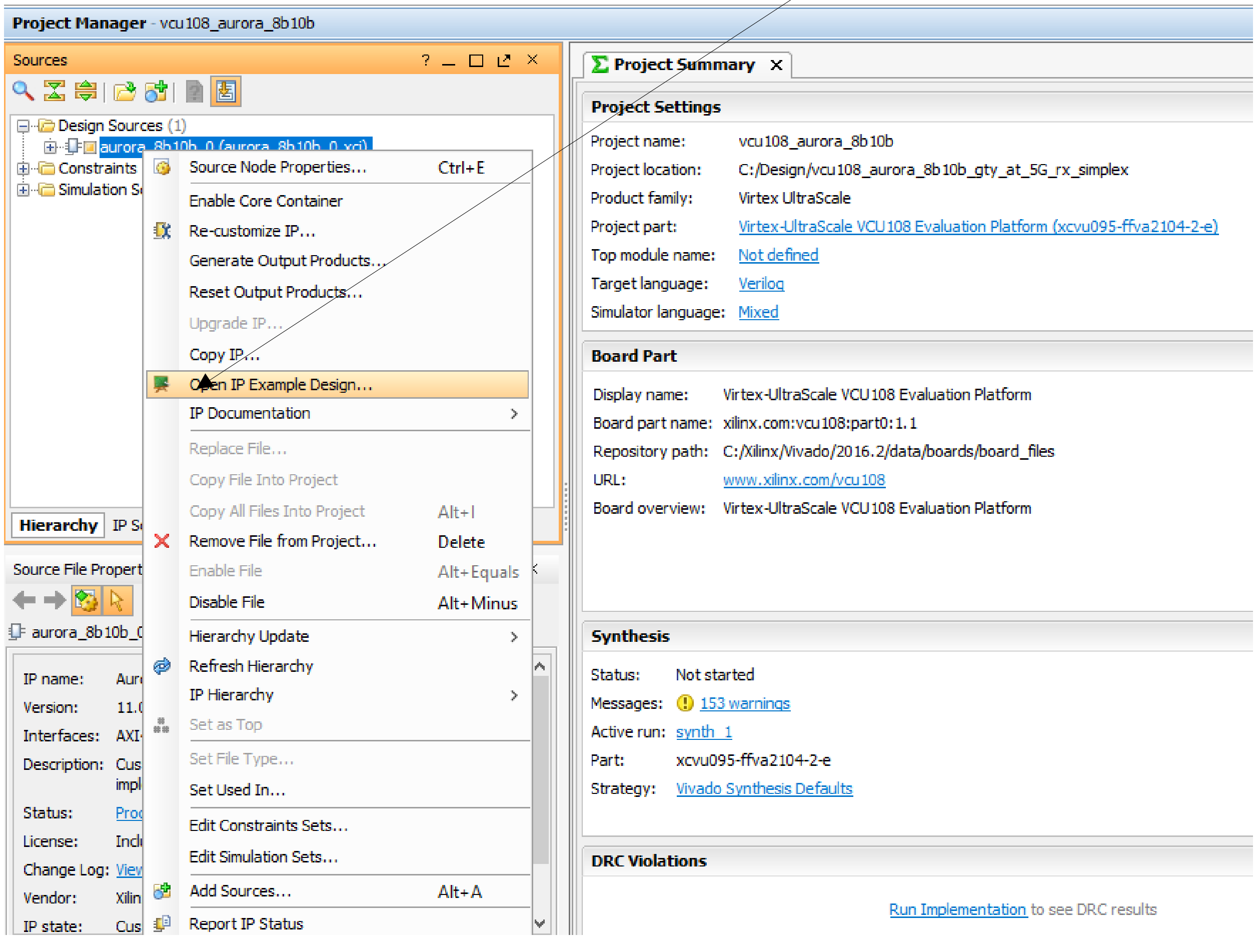
Debug and Control

□ Vivado Lab Tools

□ Additional transceiver control and status ports

Readout (Aurora8b10b) Implementation in GTY, recipe (1)

Once the Aurora8b10b core is generated, open IP example Design



IP name: Aur
Version: 11.
Interfaces: AXI
Description: Cus impl
Status: Pro
License: Incl
Change Log: View
Vendor: Xilin
IP state: Cus

Project Settings

Project name: vcu108_aurora_8b10b
Project location: C:/Design/vcu108_aurora_8b10b_gty_at_5G_rx_simplex
Product family: Virtex UltraScale
Project part: Virtex-UltraScale VCU108 Evaluation Platform (xcvu095-ffva2104-2-e)
Top module name: Not defined
Target language: Verilog
Simulator language: Mixed

Board Part

Display name: Virtex-UltraScale VCU108 Evaluation Platform
Board part name: xilinx.com:vcu108:part0:1.1
Repository path: C:/Xilinx/Vivado/2016.2/data/boards/board_files
URL: www.xilinx.com/vcu108
Board overview: Virtex-UltraScale VCU108 Evaluation Platform

Synthesis

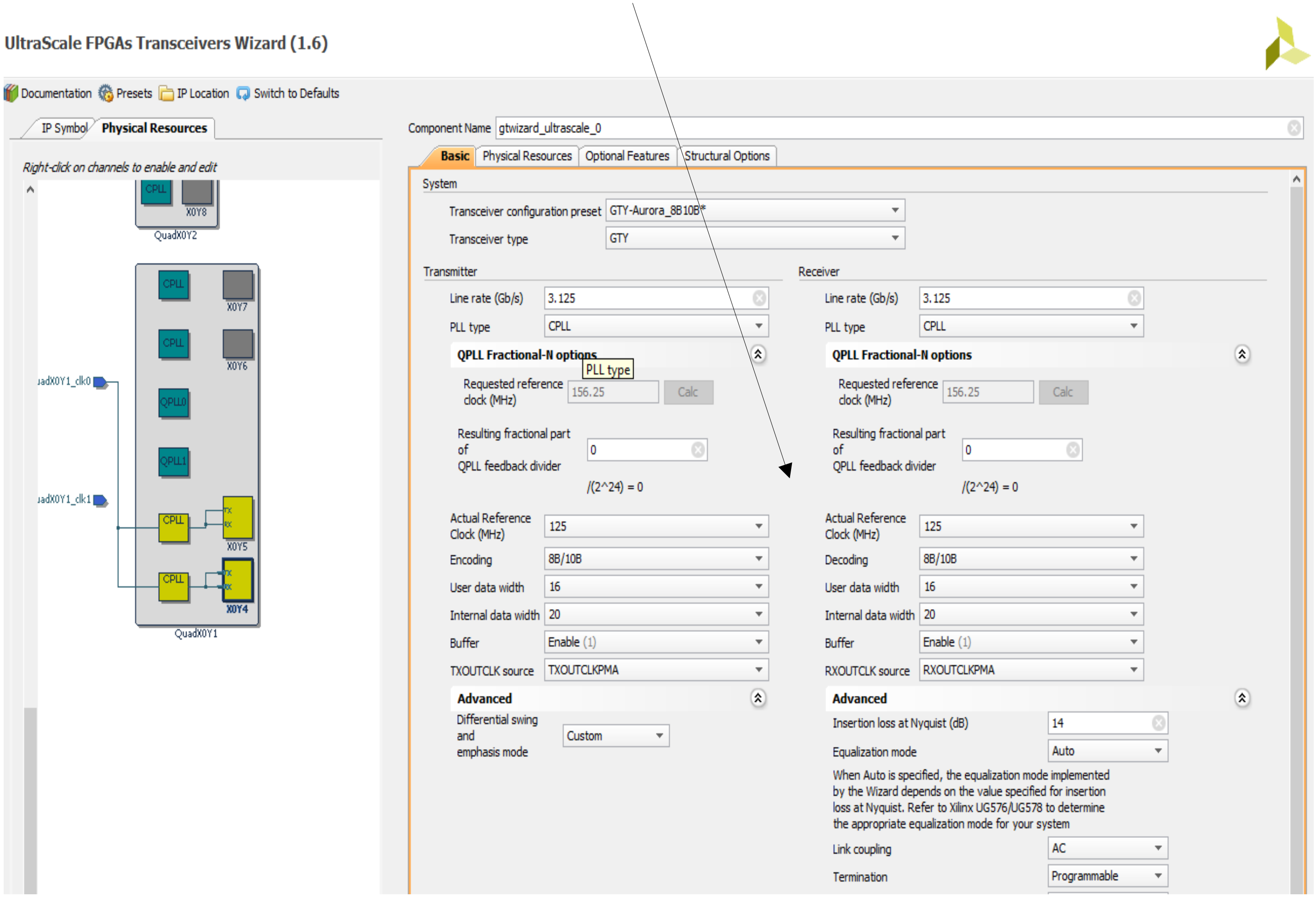
Status: Not started
Messages: ! 153 warnings
Active run: synth_1
Part: xcvu095-ffva2104-2-e
Strategy: Vivado Synthesis Defaults

DRC Violations

Run Implementation to see DRC results

Readout (Aurora8b10b) Implementation in GTY, recipe (3)

Open GT wizard to configure the GT part of the protocol



UltraScale FPGAs Transceivers Wizard (1.6)

Component Name: gtwizard_ultrascale_0

System

- Transceiver configuration preset: GTY-Aurora_8B10B*
- Transceiver type: GTY

Transmitter

- Line rate (Gb/s): 3.125
- PLL type: CPLL
- **QPLL Fractional-N options**
  - Requested reference clock (MHz): 156.25
  - Resulting fractional part of QPLL feedback divider: 0 /(2^24) = 0
- Actual Reference Clock (MHz): 125
- Encoding: 8B/10B
- User data width: 16
- Internal data width: 20
- Buffer: Enable (1)
- TXOUTCLK source: TXOUTCLKPMA
- **Advanced**
  - Differential swing and emphasis mode: Custom

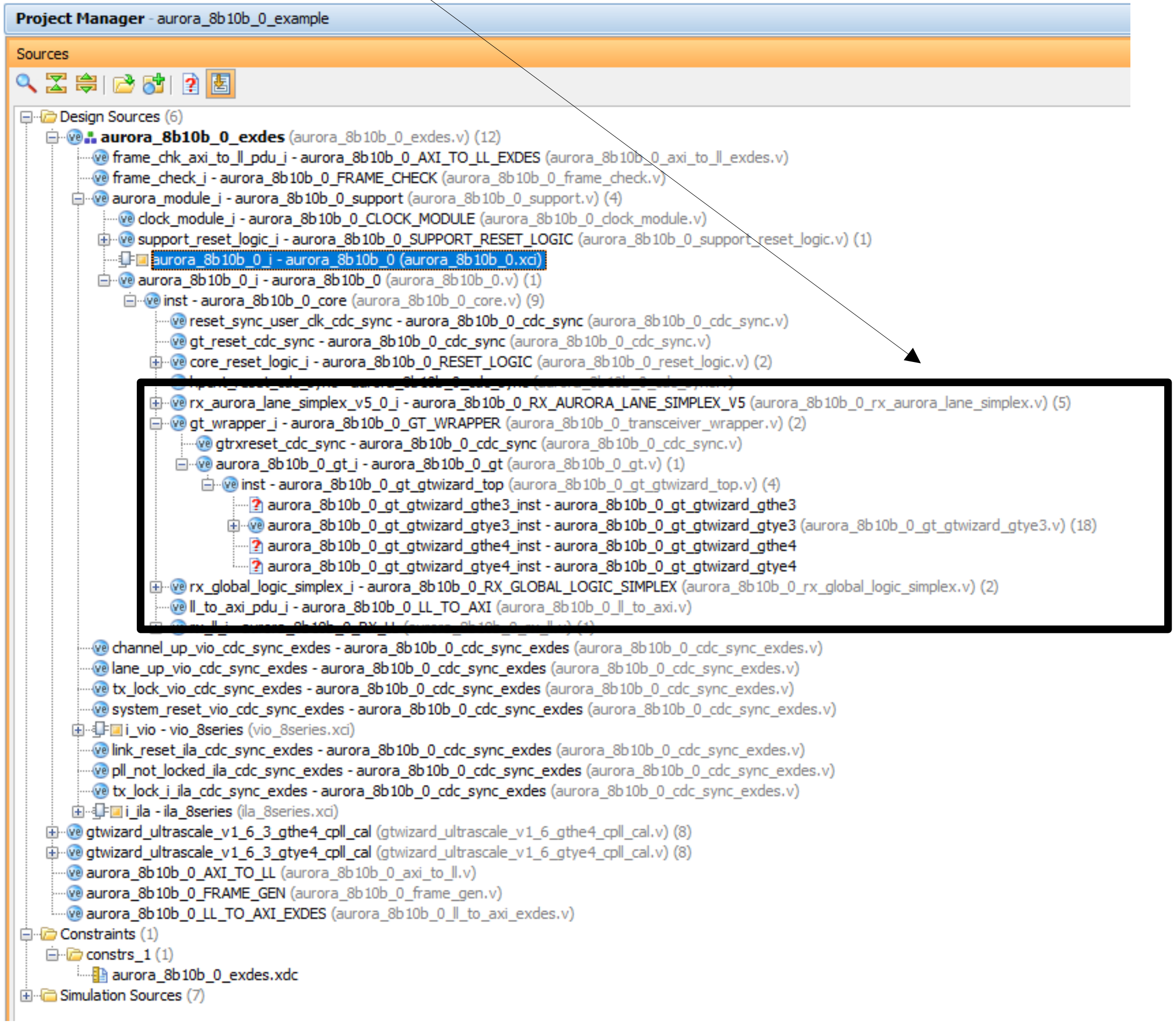
Receiver

- Line rate (Gb/s): 3.125
- PLL type: CPLL
- **QPLL Fractional-N options**
  - Requested reference clock (MHz): 156.25
  - Resulting fractional part of QPLL feedback divider: 0 /(2^24) = 0
- Actual Reference Clock (MHz): 125
- Decoding: 8B/10B
- User data width: 16
- Internal data width: 20
- Buffer: Enable (1)
- RXOUTCLK source: RXOUTCLKPMA
- **Advanced**
  - Insertion loss at Nyquist (dB): 14
  - Equalization mode: Auto

    When Auto is specified, the equalization mode implemented by the Wizard depends on the value specified for insertion loss at Nyquist. Refer to Xilinx UG576/UG578 to determine the appropriate equalization mode for your system
  - Link coupling: AC
  - Termination: Programmable

Readout (Aurora8b10b) Implementation in GTY, recipe (4)

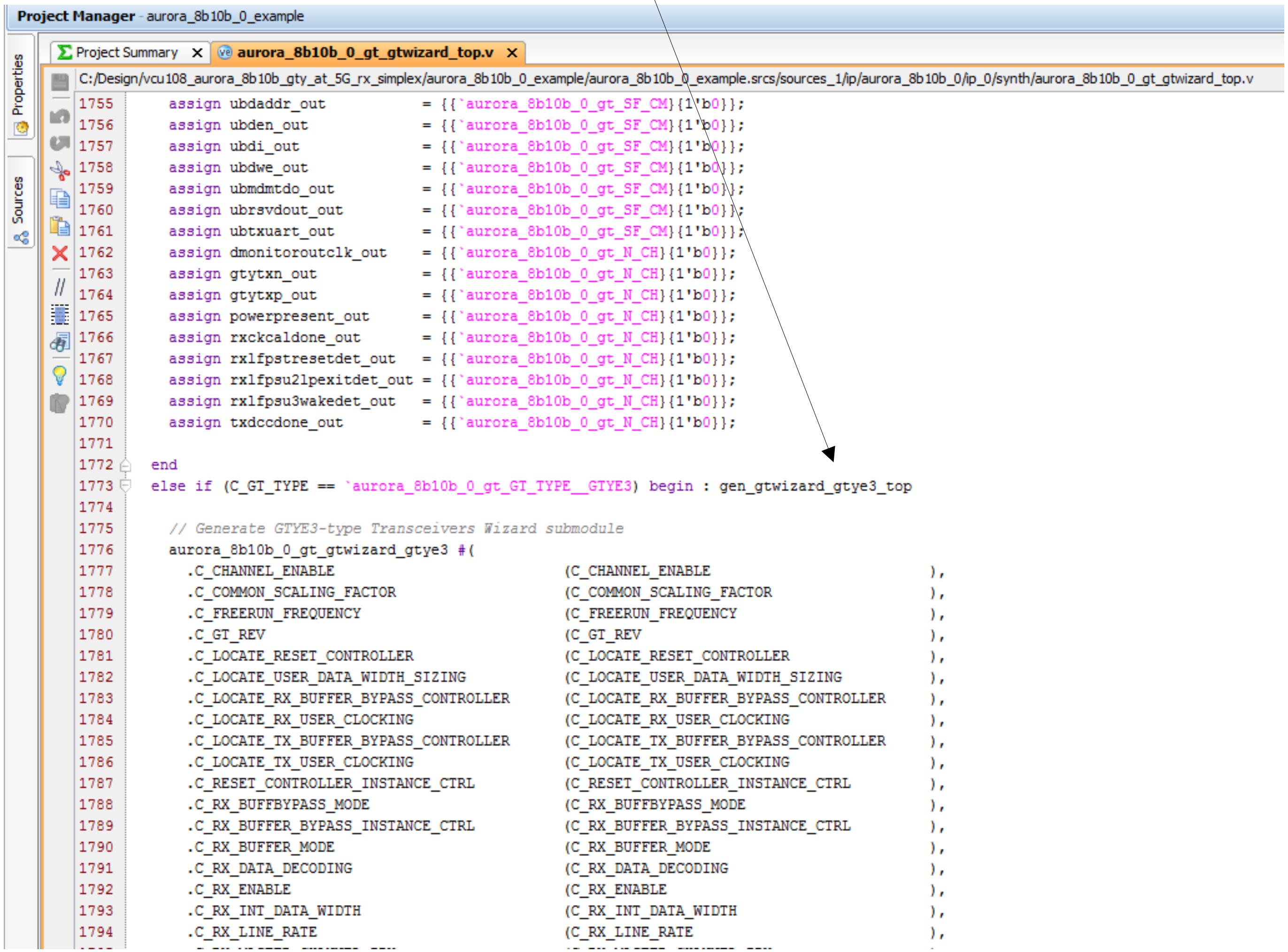
Once the GT files are generated, replaced them with the files generated by Aurora8b10b



```
Design Sources (6)
  aurora_8b10b_0_exdes (aurora_8b10b_0_exdes.v) (12)
    frame_chk_axi_to_ll_pdu_i - aurora_8b10b_0_AXI_TO_LL_EXDES (aurora_8b10b_0_axi_to_ll_exdes.v)
    frame_check_i - aurora_8b10b_0_FRAME_CHECK (aurora_8b10b_0_frame_check.v)
    aurora_module_i - aurora_8b10b_0_support (aurora_8b10b_0_support.v) (4)
      clock_module_i - aurora_8b10b_0_CLOCK_MODULE (aurora_8b10b_0_clock_module.v)
      support_reset_logic_i - aurora_8b10b_0_SUPPORT_RESET_LOGIC (aurora_8b10b_0_support_reset_logic.v) (1)
      aurora_8b10b_0_i - aurora_8b10b_0 (aurora_8b10b_0.xci)
      aurora_8b10b_0_i - aurora_8b10b_0 (aurora_8b10b_0.v) (1)
        inst - aurora_8b10b_0_core (aurora_8b10b_0_core.v) (9)
          reset_sync_user_clk_cdc_sync - aurora_8b10b_0_cdc_sync (aurora_8b10b_0_cdc_sync.v)
          gt_reset_cdc_sync - aurora_8b10b_0_cdc_sync (aurora_8b10b_0_cdc_sync.v)
          core_reset_logic_i - aurora_8b10b_0_RESET_LOGIC (aurora_8b10b_0_reset_logic.v) (2)
          hpcnt_reset_cdc_sync - aurora_8b10b_0_cdc_sync (aurora_8b10b_0_cdc_sync.v)
          rx_aurora_lane_simplex_v5_0_i - aurora_8b10b_0_RX_AURORA_LANE_SIMPLEX_V5 (aurora_8b10b_0_rx_aurora_lane_simplex.v) (5)
          gt_wrapper_i - aurora_8b10b_0_GT_WRAPPER (aurora_8b10b_0_transceiver_wrapper.v) (2)
            gtrxreset_cdc_sync - aurora_8b10b_0_cdc_sync (aurora_8b10b_0_cdc_sync.v)
            aurora_8b10b_0_gt_i - aurora_8b10b_0_gt (aurora_8b10b_0_gt.v) (1)
              inst - aurora_8b10b_0_gt_gtwizard_top (aurora_8b10b_0_gt_gtwizard_top.v) (4)
                aurora_8b10b_0_gt_gtwizard_gthe3_inst - aurora_8b10b_0_gt_gtwizard_gthe3
                aurora_8b10b_0_gt_gtwizard_gtye3_inst - aurora_8b10b_0_gt_gtwizard_gtye3 (aurora_8b10b_0_gt_gtwizard_gtye3.v) (18)
                aurora_8b10b_0_gt_gtwizard_gthe4_inst - aurora_8b10b_0_gt_gtwizard_gthe4
                aurora_8b10b_0_gt_gtwizard_gtye4_inst - aurora_8b10b_0_gt_gtwizard_gtye4
          rx_global_logic_simplex_i - aurora_8b10b_0_RX_GLOBAL_LOGIC_SIMPLEX (aurora_8b10b_0_rx_global_logic_simplex.v) (2)
          ll_to_axi_pdu_i - aurora_8b10b_0_LL_TO_AXI (aurora_8b10b_0_ll_to_axi.v)
          rx_ll_i - aurora_8b10b_0_RX_LL (aurora_8b10b_0_rx_ll.v) (1)
    channel_up_vio_cdc_sync_exdes - aurora_8b10b_0_cdc_sync_exdes (aurora_8b10b_0_cdc_sync_exdes.v)
    lane_up_vio_cdc_sync_exdes - aurora_8b10b_0_cdc_sync_exdes (aurora_8b10b_0_cdc_sync_exdes.v)
    tx_lock_vio_cdc_sync_exdes - aurora_8b10b_0_cdc_sync_exdes (aurora_8b10b_0_cdc_sync_exdes.v)
    system_reset_vio_cdc_sync_exdes - aurora_8b10b_0_cdc_sync_exdes (aurora_8b10b_0_cdc_sync_exdes.v)
    i_vio - vio_8series (vio_8series.xci)
    link_reset_ila_cdc_sync_exdes - aurora_8b10b_0_cdc_sync_exdes (aurora_8b10b_0_cdc_sync_exdes.v)
    pll_not_locked_ila_cdc_sync_exdes - aurora_8b10b_0_cdc_sync_exdes (aurora_8b10b_0_cdc_sync_exdes.v)
    tx_lock_i_ila_cdc_sync_exdes - aurora_8b10b_0_cdc_sync_exdes (aurora_8b10b_0_cdc_sync_exdes.v)
    i_ila - ila_8series (ila_8series.xci)
  gtwizard_ultrascale_v1_6_3_gthe4_cpll_cal (gtwizard_ultrascale_v1_6_gthe4_cpll_cal.v) (8)
  gtwizard_ultrascale_v1_6_3_gtye4_cpll_cal (gtwizard_ultrascale_v1_6_gtye4_cpll_cal.v) (8)
  aurora_8b10b_0_AXI_TO_LL (aurora_8b10b_0_axi_to_ll.v)
  aurora_8b10b_0_FRAME_GEN (aurora_8b10b_0_frame_gen.v)
  aurora_8b10b_0_LL_TO_AXI_EXDES (aurora_8b10b_0_ll_to_axi_exdes.v)
Constraints (1)
  constrs_1 (1)
    aurora_8b10b_0_exdes.xdc
Simulation Sources (7)
```

Readout (Aurora8b10b) Implementation in GTY, recipe (5)

This is the original Aurora_8b10b_0_gt_gtwizard_top file. It looks that the Xilinx experts were pretty close to add the GTY to the Aurora8b10b IP core.



Project Manager - aurora_8b10b_0_example
Properties
Sources
Project Summary
aurora_8b10b_0_gt_gtwizard_top.v

C:/Design/vcu108_aurora_8b10b_gty_at_5G_rx_simplex/aurora_8b10b_0_example/aurora_8b10b_0_example.srcs/sources_1/ip/aurora_8b10b_0/ip_0/synth/aurora_8b10b_0_gt_gtwizard_top.v

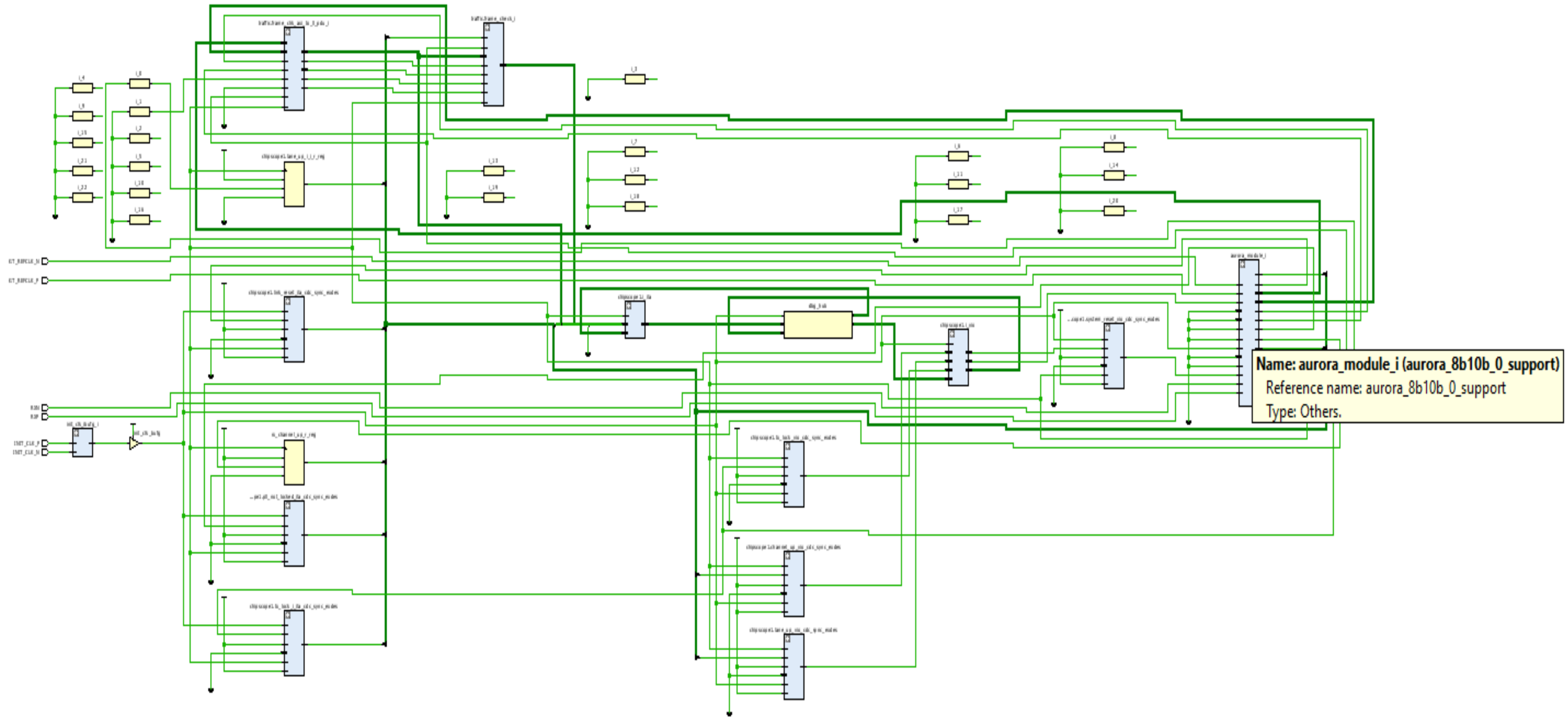
```
      assign ubdaddr_out           = {{`aurora_8b10b_0_gt_SF_CM}{1'b0}};
      assign ubden_out             = {{`aurora_8b10b_0_gt_SF_CM}{1'b0}};
      assign ubdi_out              = {{`aurora_8b10b_0_gt_SF_CM}{1'b0}};
      assign ubdwe_out             = {{`aurora_8b10b_0_gt_SF_CM}{1'b0}};
      assign ubmdmtdo_out          = {{`aurora_8b10b_0_gt_SF_CM}{1'b0}};
      assign ubrsvdout_out         = {{`aurora_8b10b_0_gt_SF_CM}{1'b0}};
      assign ubtxuart_out          = {{`aurora_8b10b_0_gt_SF_CM}{1'b0}};
      assign dmonitoroutclk_out    = {{`aurora_8b10b_0_gt_N_CH}{1'b0}};
      assign gtytxn_out            = {{`aurora_8b10b_0_gt_N_CH}{1'b0}};
      assign gtytxp_out            = {{`aurora_8b10b_0_gt_N_CH}{1'b0}};
      assign powerpresent_out      = {{`aurora_8b10b_0_gt_N_CH}{1'b0}};
      assign rxckcaldone_out       = {{`aurora_8b10b_0_gt_N_CH}{1'b0}};
      assign rxlfpstresetdet_out   = {{`aurora_8b10b_0_gt_N_CH}{1'b0}};
      assign rxlfpsu2lpexitdet_out = {{`aurora_8b10b_0_gt_N_CH}{1'b0}};
      assign rxlfpsu3wakedet_out   = {{`aurora_8b10b_0_gt_N_CH}{1'b0}};
      assign txdccdone_out         = {{`aurora_8b10b_0_gt_N_CH}{1'b0}};

    end
    else if (C_GT_TYPE == `aurora_8b10b_0_gt_GT_TYPE__GTYE3) begin : gen_gtwizard_gtye3_top

      // Generate GTYE3-type Transceivers Wizard submodule
      aurora_8b10b_0_gt_gtwizard_gtye3 #(
        .C_CHANNEL_ENABLE                     (C_CHANNEL_ENABLE                     ),
        .C_COMMON_SCALING_FACTOR              (C_COMMON_SCALING_FACTOR              ),
        .C_FREERUN_FREQUENCY                  (C_FREERUN_FREQUENCY                  ),
        .C_GT_REV                             (C_GT_REV                             ),
        .C_LOCATE_RESET_CONTROLLER            (C_LOCATE_RESET_CONTROLLER            ),
        .C_LOCATE_USER_DATA_WIDTH_SIZING      (C_LOCATE_USER_DATA_WIDTH_SIZING      ),
        .C_LOCATE_RX_BUFFER_BYPASS_CONTROLLER (C_LOCATE_RX_BUFFER_BYPASS_CONTROLLER ),
        .C_LOCATE_RX_USER_CLOCKING            (C_LOCATE_RX_USER_CLOCKING            ),
        .C_LOCATE_TX_BUFFER_BYPASS_CONTROLLER (C_LOCATE_TX_BUFFER_BYPASS_CONTROLLER ),
        .C_LOCATE_TX_USER_CLOCKING            (C_LOCATE_TX_USER_CLOCKING            ),
        .C_RESET_CONTROLLER_INSTANCE_CTRL     (C_RESET_CONTROLLER_INSTANCE_CTRL     ),
        .C_RX_BUFFBYPASS_MODE                 (C_RX_BUFFBYPASS_MODE                 ),
        .C_RX_BUFFER_BYPASS_INSTANCE_CTRL     (C_RX_BUFFER_BYPASS_INSTANCE_CTRL     ),
        .C_RX_BUFFER_MODE                     (C_RX_BUFFER_MODE                     ),
        .C_RX_DATA_DECODING                   (C_RX_DATA_DECODING                   ),
        .C_RX_ENABLE                          (C_RX_ENABLE                          ),
        .C_RX_INT_DATA_WIDTH                  (C_RX_INT_DATA_WIDTH                  ),
        .C_RX_LINE_RATE                       (C_RX_LINE_RATE                       ),
```

Readout (Aurora8b10b) Implementation in GTY, recipe (5)

Aurora8b10b protocol in GTY
(based on the Xilinx Example Design – non standard implementation)
Visual inspection needed (design, xdc file, system clock, mgt reference clock)

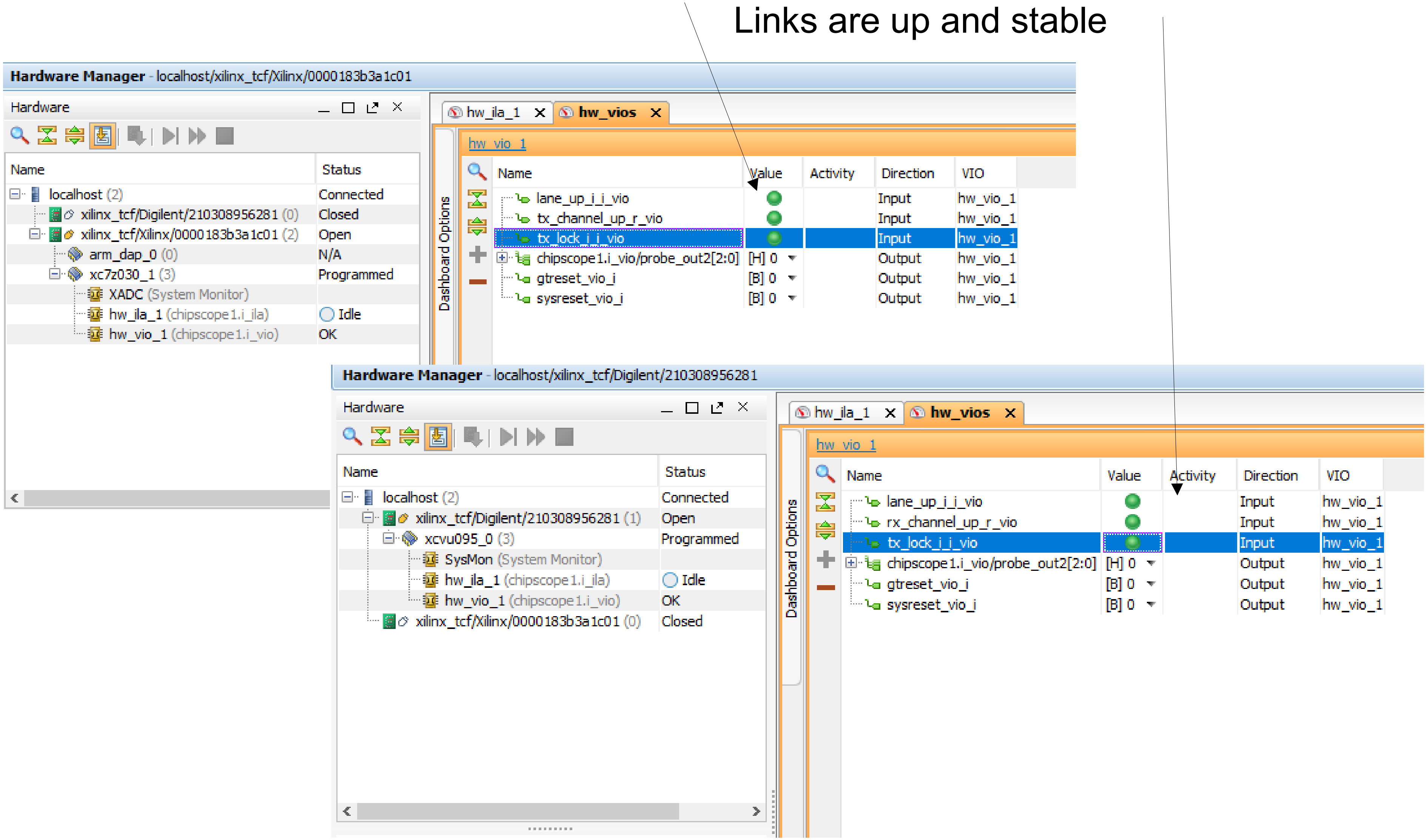


Name: aurora_module_i (aurora_8b10b_0_support)
Reference name: aurora_8b10b_0_support
Type: Others.

Readout (Aurora8b10b) Implementation in GTY, test (1)

ZYNQ Board (GTX)   VCU108 Xilinx Dev Board

Links are up and stable



Hardware Manager - localhost/xilinx_tcf/Xilinx/0000183b3a1c01

Hardware

Name	Status
localhost (2)	Connected
xilinx_tcf/Digilent/210308956281 (0)	Closed
xilinx_tcf/Xilinx/0000183b3a1c01 (2)	Open
arm_dap_0 (0)	N/A
xc7z030_1 (3)	Programmed
XADC (System Monitor)	
hw_ila_1 (chipscope1.i_ila)	○ Idle
hw_vio_1 (chipscope1.i_vio)	OK

Name	Value	Activity	Direction	VIO
lane_up_i_i_vio	●		Input	hw_vio_1
tx_channel_up_r_vio	●		Input	hw_vio_1
tx_lock_i_i_vio	●		Input	hw_vio_1
chipscope1.i_vio/probe_out2[2:0]	[H] 0		Output	hw_vio_1
gtreset_vio_i	[B] 0		Output	hw_vio_1
sysreset_vio_i	[B] 0		Output	hw_vio_1

Hardware Manager - localhost/xilinx_tcf/Digilent/210308956281

Hardware

Name	Status
localhost (2)	Connected
xilinx_tcf/Digilent/210308956281 (1)	Open
xcvu095_0 (3)	Programmed
SysMon (System Monitor)	
hw_ila_1 (chipscope1.i_ila)	○ Idle
hw_vio_1 (chipscope1.i_vio)	OK
xilinx_tcf/Xilinx/0000183b3a1c01 (0)	Closed

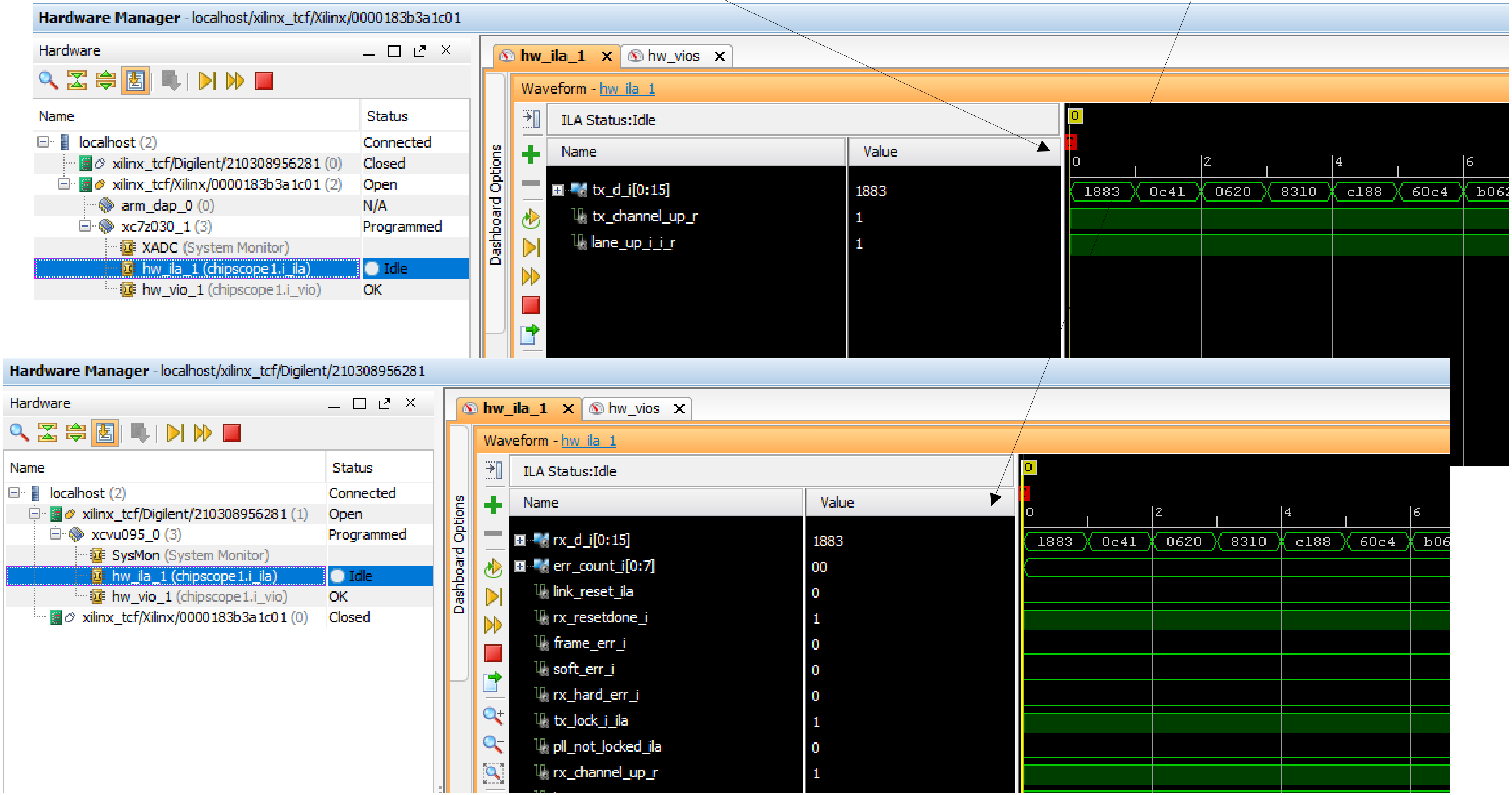
Name	Value	Activity	Direction	VIO
lane_up_i_i_vio	●		Input	hw_vio_1
rx_channel_up_r_vio	●		Input	hw_vio_1
tx_lock_i_i_vio	●		Input	hw_vio_1
chipscope1.i_vio/probe_out2[2:0]	[H] 0		Output	hw_vio_1
gtreset_vio_i	[B] 0		Output	hw_vio_1
sysreset_vio_i	[B] 0		Output	hw_vio_1

Readout (Aurora8b10b) Implementation in GTY, test (2)

ZYNQ Board (GTX)

VCU108 Xilinx Dev Board

TX and RX data check



Hardware Manager - localhost/xilinx_tcf/Xilinx/0000183b3a1c01

Hardware

Name	Status
localhost (2)	Connected
xilinx_tcf/Digilent/210308956281 (0)	Closed
xilinx_tcf/Xilinx/0000183b3a1c01 (2)	Open
arm_dap_0 (0)	N/A
xc7z030_1 (3)	Programmed
XADC (System Monitor)	
hw_ila_1 (chipscope1.i_ila)	Idle
hw_vio_1 (chipscope1.i_vio)	OK

ILA Status:Idle

Name	Value
tx_d_i[0:15]	1883
tx_channel_up_r	1
lane_up_i_i_r	1

Hardware Manager - localhost/xilinx_tcf/Digilent/210308956281

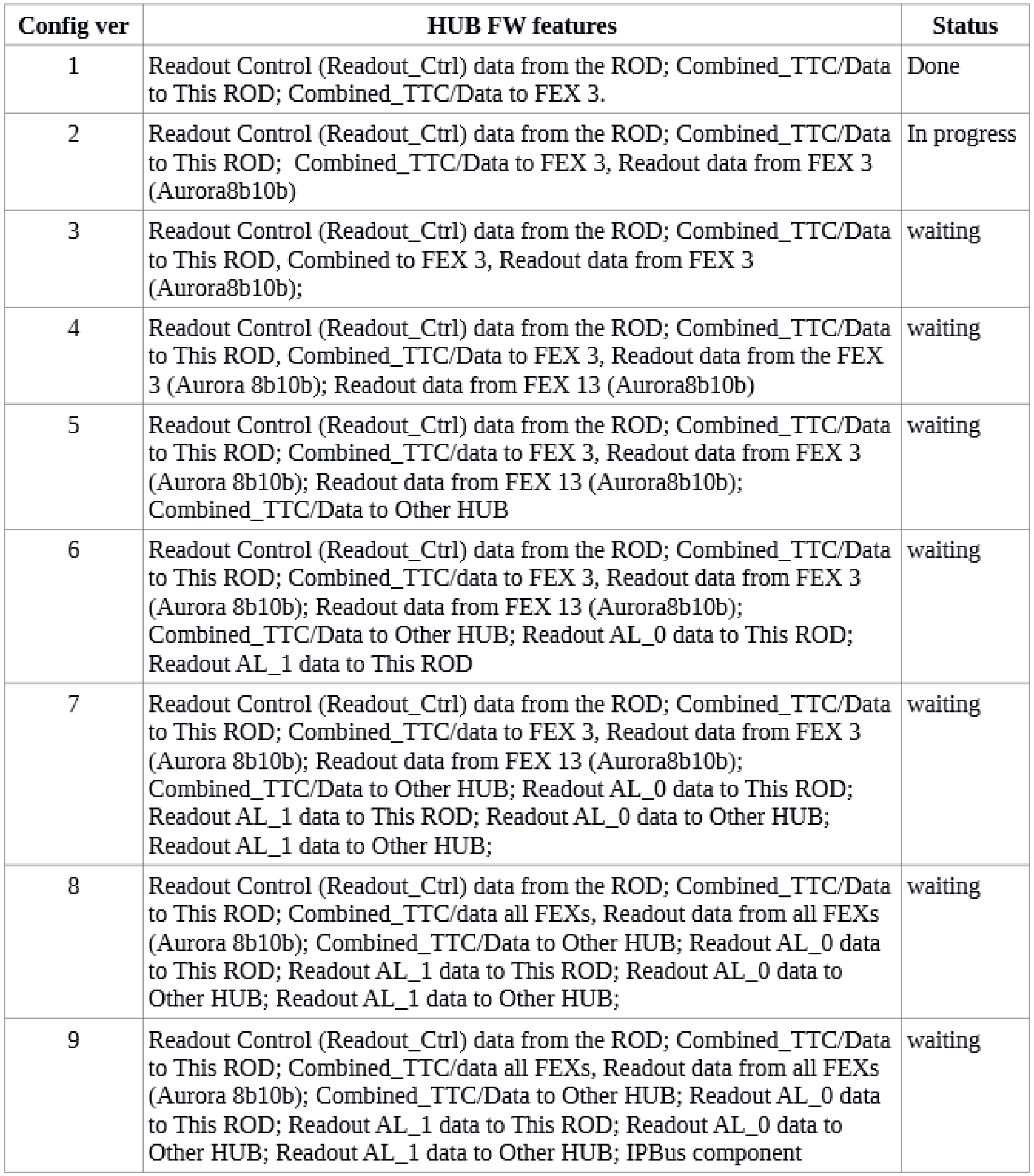
Hardware

Name	Status
localhost (2)	Connected
xilinx_tcf/Digilent/210308956281 (1)	Open
xcvu095_0 (3)	Programmed
SysMon (System Monitor)	
hw_ila_1 (chipscope1.i_ila)	Idle
hw_vio_1 (chipscope1.i_vio)	OK
xilinx_tcf/Xilinx/0000183b3a1c01 (0)	Closed

ILA Status:Idle

Name	Value
rx_d_i[0:15]	1883
err_count_i[0:7]	00
link_reset_ila	0
rx_resetdone_i	1
frame_err_i	0
soft_err_i	0
rx_hard_err_i	0
tx_lock_i_ila	1
pll_not_locked_ila	0
rx_channel_up_r	1

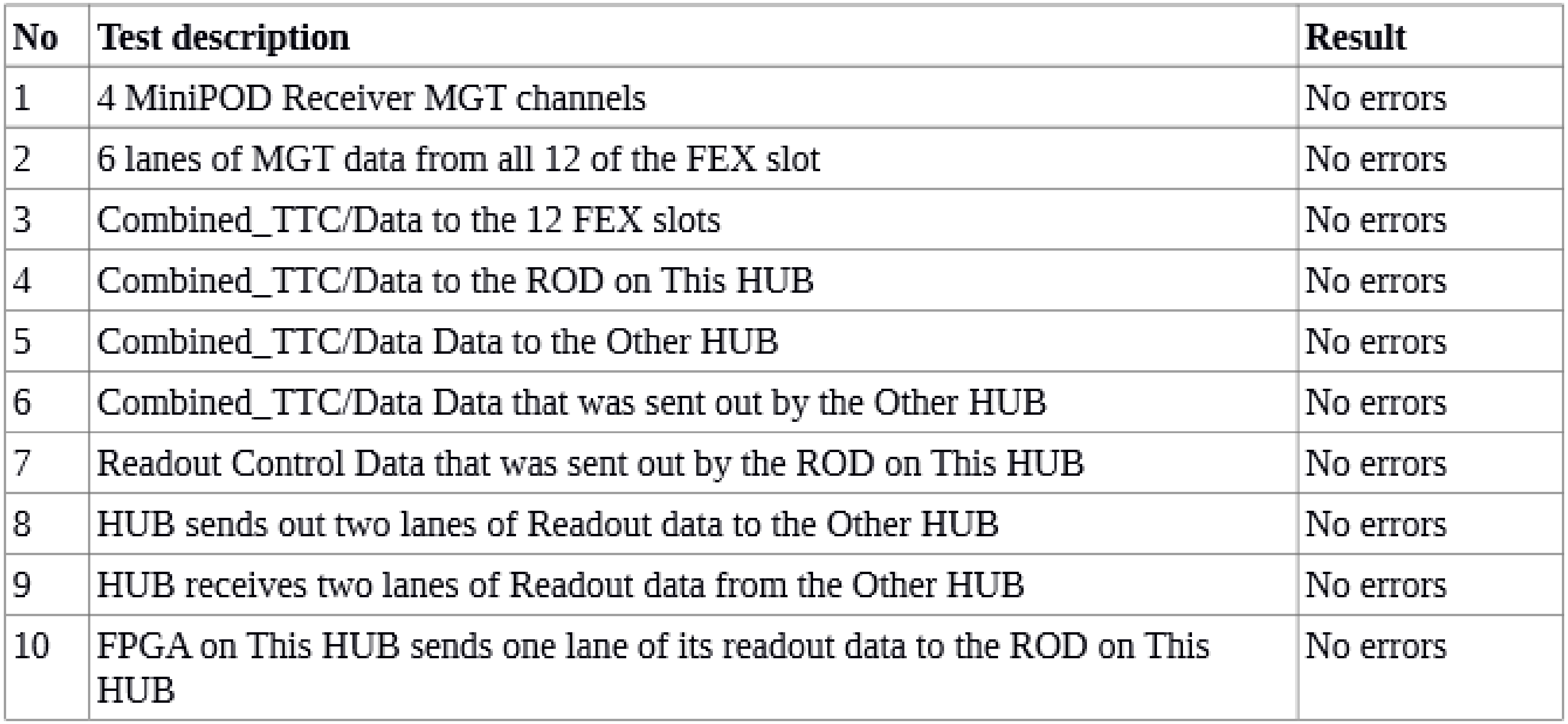
HUB FW development scheme



Config ver	HUB FW features	Status
1	Readout Control (Readout_Ctrl) data from the ROD; Combined_TTC/Data to This ROD; Combined_TTC/Data to FEX 3.	Done
2	Readout Control (Readout_Ctrl) data from the ROD; Combined_TTC/Data to This ROD; Combined_TTC/Data to FEX 3, Readout data from FEX 3 (Aurora8b10b)	In progress
3	Readout Control (Readout_Ctrl) data from the ROD; Combined_TTC/Data to This ROD, Combined to FEX 3, Readout data from FEX 3 (Aurora8b10b);	waiting
4	Readout Control (Readout_Ctrl) data from the ROD; Combined_TTC/Data to This ROD, Combined_TTC/Data to FEX 3, Readout data from the FEX 3 (Aurora 8b10b); Readout data from FEX 13 (Aurora8b10b)	waiting
5	Readout Control (Readout_Ctrl) data from the ROD; Combined_TTC/Data to This ROD; Combined_TTC/data to FEX 3, Readout data from FEX 3 (Aurora 8b10b); Readout data from FEX 13 (Aurora8b10b); Combined_TTC/Data to Other HUB	waiting
6	Readout Control (Readout_Ctrl) data from the ROD; Combined_TTC/Data to This ROD; Combined_TTC/data to FEX 3, Readout data from FEX 3 (Aurora 8b10b); Readout data from FEX 13 (Aurora8b10b); Combined_TTC/Data to Other HUB; Readout AL_0 data to This ROD; Readout AL_1 data to This ROD	waiting
7	Readout Control (Readout_Ctrl) data from the ROD; Combined_TTC/Data to This ROD; Combined_TTC/data to FEX 3, Readout data from FEX 3 (Aurora 8b10b); Readout data from FEX 13 (Aurora8b10b); Combined_TTC/Data to Other HUB; Readout AL_0 data to This ROD; Readout AL_1 data to This ROD; Readout AL_0 data to Other HUB; Readout AL_1 data to Other HUB;	waiting
8	Readout Control (Readout_Ctrl) data from the ROD; Combined_TTC/Data to This ROD; Combined_TTC/data all FEXs, Readout data from all FEXs (Aurora 8b10b); Combined_TTC/Data to Other HUB; Readout AL_0 data to This ROD; Readout AL_1 data to This ROD; Readout AL_0 data to Other HUB; Readout AL_1 data to Other HUB;	waiting
9	Readout Control (Readout_Ctrl) data from the ROD; Combined_TTC/Data to This ROD; Combined_TTC/data all FEXs, Readout data from all FEXs (Aurora 8b10b); Combined_TTC/Data to Other HUB; Readout AL_0 data to This ROD; Readout AL_1 data to This ROD; Readout AL_0 data to Other HUB; Readout AL_1 data to Other HUB; IPBus component	waiting

IBERT test (MGT channels)

- In order to test the specific MGT channels the IBERT FW is provided for the FTM, ROD and HUB modules. The table below describes the list of tests and results.



No	Test description	Result
1	4 MiniPOD Receiver MGT channels	No errors
2	6 lanes of MGT data from all 12 of the FEX slot	No errors
3	Combined_TTC/Data to the 12 FEX slots	No errors
4	Combined_TTC/Data to the ROD on This HUB	No errors
5	Combined_TTC/Data Data to the Other HUB	No errors
6	Combined_TTC/Data Data that was sent out by the Other HUB	No errors
7	Readout Control Data that was sent out by the ROD on This HUB	No errors
8	HUB sends out two lanes of Readout data to the Other HUB	No errors
9	HUB receives two lanes of Readout data from the Other HUB	No errors
10	FPGA on This HUB sends one lane of its readout data to the ROD on This HUB	No errors

Summary

- HUB FW development in good shape
- HUB FW comprises several components [MAC + IPBus, Readout_Control, Combined_TTC/Data, Readout (Aurora 8b10b),...]
- These components were successfully tested on the HUB module and/or on the Xilinx dev board
- Firmware development split into several stages
- Aurora, Readout_Control and Combined_TTC/Data initialization scheme needs to be discussed (dedicated meeting is foreseen)
- Repository structure will be defined soon