ATLAS Level-1 Calorimeter Trigger Update

HUB Firmware Specification

Draft for PRR

Version: 1.3

09 December 2018

Table of Contents

1. HUB overview

The Hub module is an integral part of the L1Calo system. Its primary functions are to support FEX system readout, provide switching functionality for module control and DCS IPbus networks and to distribute timing and control signals to the FEX modules. There are to be two Hub modules per shelf. Both Hub modules will receive multi-gigabit FEX data over the ATCA Fabric Interface, which will be fanned out to a ROD mezzanine on the Hub and to the Hub's own FPGA. This high-speed data path will include two data channels from the other Hub module. The Hub module in logical slot 1 will provide switching capability for a network that routes module control signals on the base interface, while the Hub in logical slot 2 will provide switching for a network that routes DCS information. The Hub module in slot 1 will further receive TTC information from the FELIX system, and these signals will be decoded and fanned out to the FEX modules, ROD modules and also to the Hub in slot 2. The fanned-out TTC control data stream will be interleaved with ROD-to-FEX communications including, for example, back-pressure signals. The Hub module has connections to the other slots in the ATCA shelf over three distinct electrical interfaces. ATCA backplane Zone-2 consists of the Fabric Interface and the Base Interface. The Fabric Interface provides 8 differential pairs (channels) from each node slot to each Hub slot (8 to Hub-1 and 8 to Hub-2). There are a total of 8 Fabric Interface channels between Hub-1 and Hub-2 (not 16 total). The Fabric Interface pairs have a nominal bandwidth specification of 10 Gbps / channel. The Base Interface provides 4 differential pairs between each node slot and each Hub slot. There are a total of 4 Base Interface channels between Hub-1 and Hub-2. The Base Interface lines have a nominal bandwidth specification of 500 Mbps / channel, suitable for Gbps Ethernet protocol. Finally, ATCA backplane Zone-1 provides each node and Hub slot with a connection to the Intelligent Platform Management Bus (IPMB) with a total bandwidth of 100 kbps. The Hub module will provide MPO connectors in the ATCA Zone-3 region, which will allow for the routing of fiber-optic cables to/from the MiniPODs on the Hub and ROD modules. The L1Calo FEX-Hub system will consist of eight modules. There will be two eFEX shelves, one jFEX shelf and one L1Topo shelf, each hosting two Hub modules.

1.1 HUB Functionality

1.1.1 Support of the ROD Mezzanine Card

The FEX Hub physically holds the ROD Mezzanine Card and provides electrical connections to it through two 400 pin Meg-Array connectors.

1.1.2 FEX and Hub Readout Data Distribution

The Hub receives over the Fabric Interface 6 serial streams of Readout Data from each FEX Module. Each FEX-Hub also receives over the Fabric Interface 2 serial streams of Readout Data from the other FEX-Hub in the crate. These 74 high speed serial streams are fanned out on the FEX-Hub. One copy of each stream is sent to the ROD and one copy is sent to the Hub's own UltraScale FPGA. The Hub FPGA also sends 2 serial streams with its own Readout Data to its own ROD. Each ROD thus receives a total of 76 high speed Readout Data streams: 6 streams from each FEX, 2 streams from the local Hub FPGA and 2 streams from the other Hub's Hub FPGA. The data rate per readout stream will be 10 Gbps or less.

1.1.3 TTC Clock and Data Stream Distribution

The Hub in Slot 1 uses a 12-channel MiniPOD optical receiver to receive TTC signals from the upstream FELIX system. The FEX-Hub receives two types of TTC signals: a copy of the LHC clock and TTC control data. These signals need to be fanned out to each FEX module, to the local ROD, to the local Hub FPGA and to the FEX-Hub in Slot 2 (including its ROD). The LHC clock is directly forwarded without any processing on the FEX-Hub. The TTC control data will be merged with additional control information coming from the ROD module from each FEX-Hub before being fanned out. The FEX-Hub uses two ports from the Fabric Interface Channel to each Node Slot to fanout these two signals to each FEX. These two TTC and control signals sent to the FEX plus the 6 Readout Data streams received from each FEX use all 8 signals pairs of each Fabric Channel connecting one FEX to the FEX-Hub, albeit with an unconventional port direction usage. The FEX-Hub in Slot 2 does receive the TTC information from FELIX directly, but receives the TTC Clock and the TTC/ROD readout control stream from the FEX-Hub in Slot 1. The FEX-Hub in Slot 2 sends any required ROD readout control data generated by its own ROD to the FEX-Hub in Slot 1 for inclusion in the combined TCC/ROD readout control data stream.

1.1.4 Ethernet Network Switch

The FEX-Hub hosts an un-managed 10/100/1000 Base-T switch to provide the following 19 Gigabit Ethernet connections [see Figure 1]:

- 1 connection on the front panel for the "up-link"
- 12 connections to the "FEX Node" modules in this crate
  via the Base Channel Fabric
- 1 connection to the ROD on this Hub (or IPMC on the other Hub) via the front panel
- 1 connection to the ROD on the other Hub (or IPMC on this Hub) via the front panel
- 1 connection to the other Hub's UltraScale FPGA via the Update Channel Interface
- 1 spare front panel connection

1.1.5 Slow Control (IPbus to HUB and ROD)

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. Figure below shows the Hub's Base Interface Ethernet Switch in the context of the other cards in the ATCA shelf. ROD: An Ethernet link is provided from the main ROD FPGA to the Ethernet switch on the Hub. This will allow a computer using IPbus to:

- Access registers within the ROD FPGA, setting parameters and controlling modes of operation.
- Store FPGA configurations into the SPI-Flash Configuration Memory.
- Initiate the loading of configurations from the SPI-Flash.

This can be used to load a configuration from one of a number of other SPI-Flash sectors. These sectors can be written via IPbus.

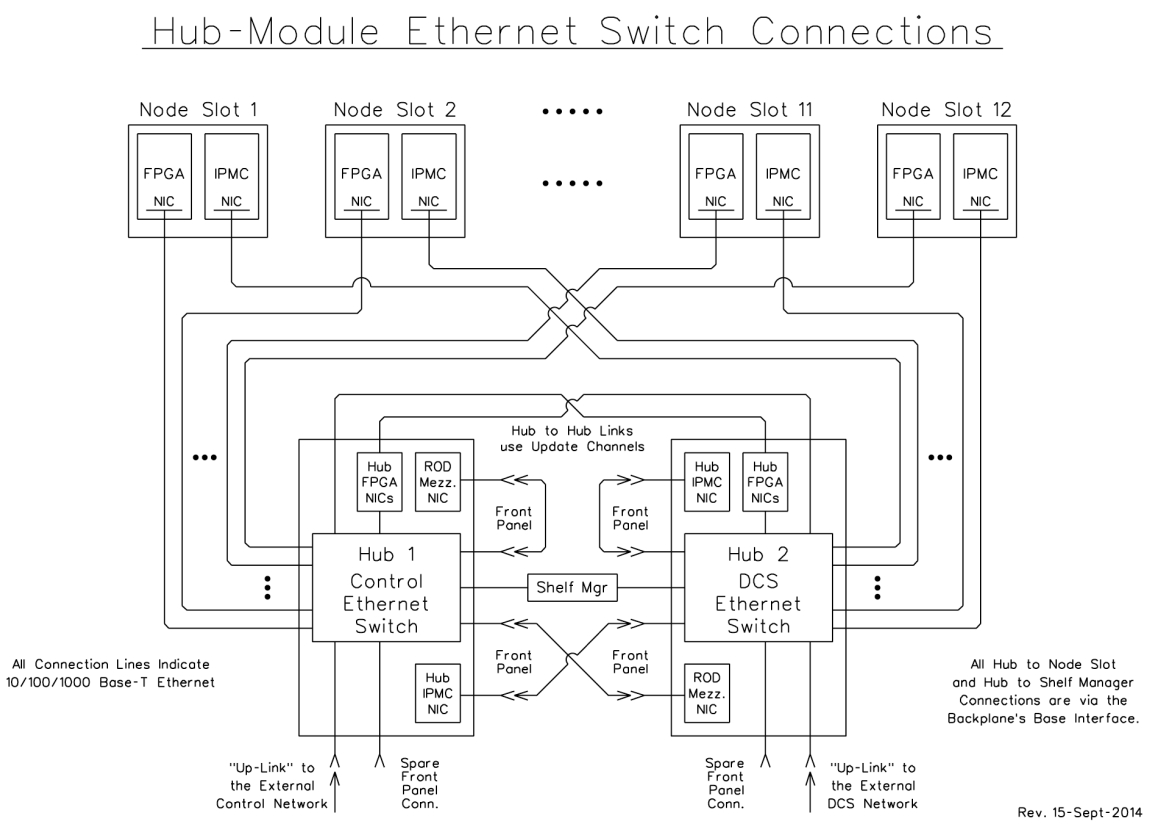


Figure 1: Illustration of FEX-Hub Ethernet network connections.

The Hub Module requires two physical chips for the Ethernet Base Interface connections to its FPGA (Micrel KSZ9031RNX http://ww1.microchip.com/downloads/en/DeviceDoc/00002117B.pdf – it can operate with a 1.8V RGMII port and thus directly connect to the Virtex7 HP I/O pins). Two FPGA MACs are connected to the physical chips via RGMII ports. This chip has both the RGMII signal connection to the FPGA that is used to move the actual Ethernet data and provides access to internal registers and also has a 2 wire serial "Management Data" port.

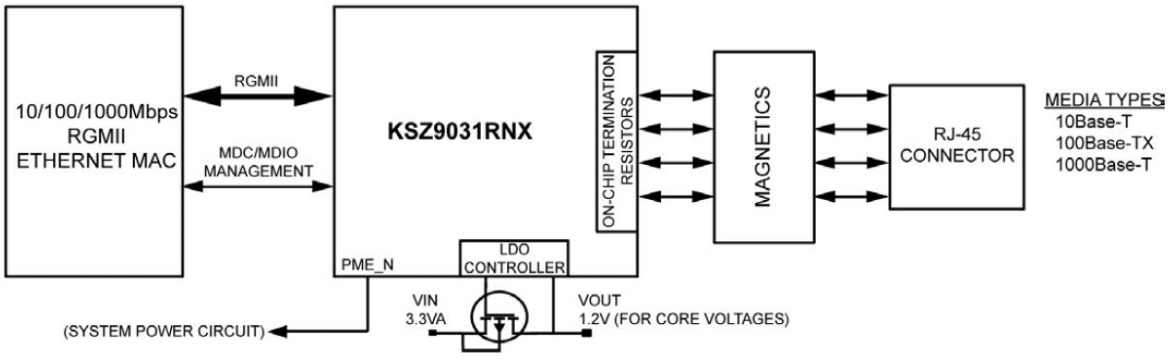


Figure 2: Illustration of Xilinx ETHERNET MAC IP and PHYS chip (KSZ9031RNX).

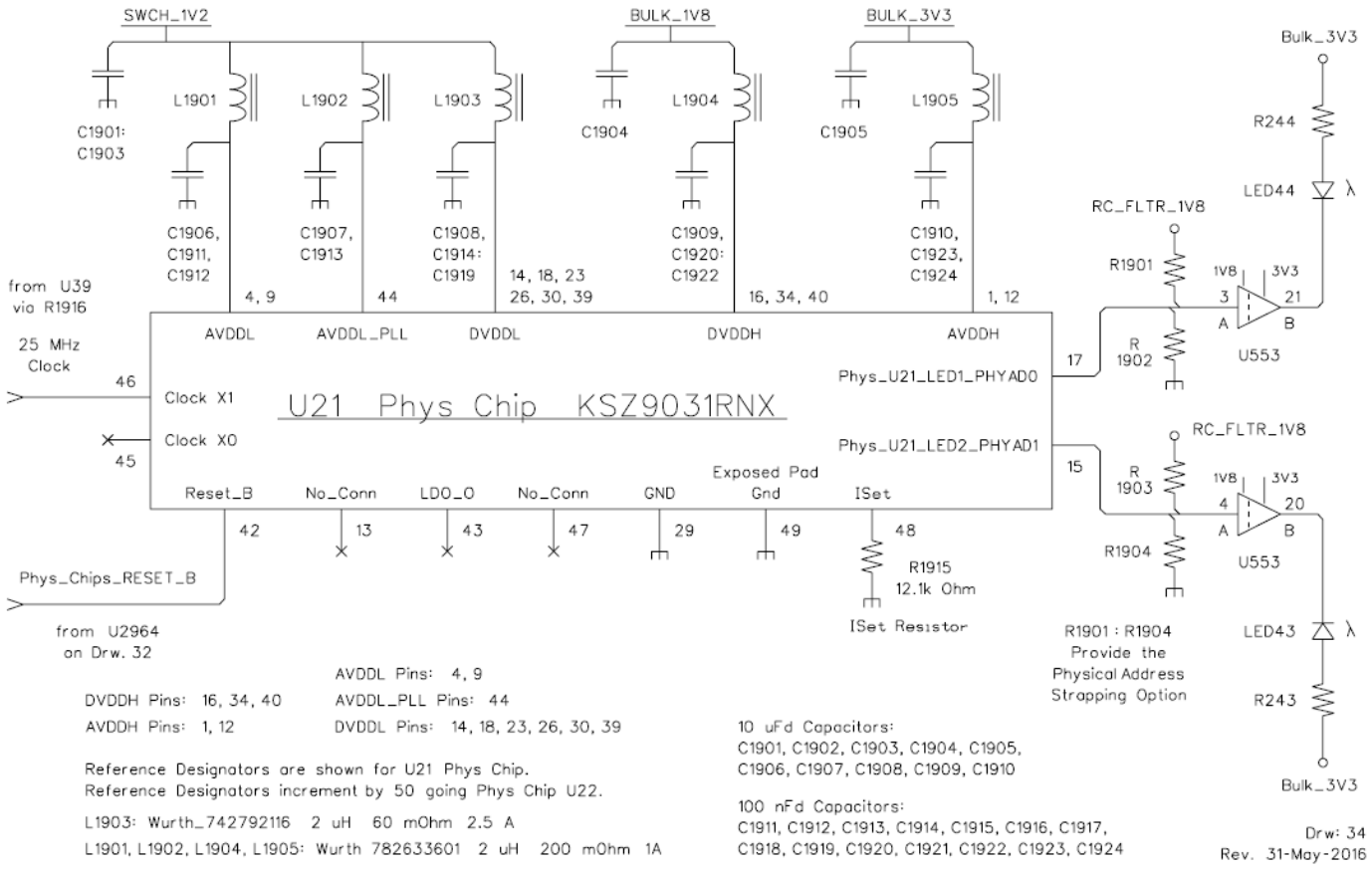


Figure 3: Schematic view of U21 Phys chip: Power, Clock, Reset.

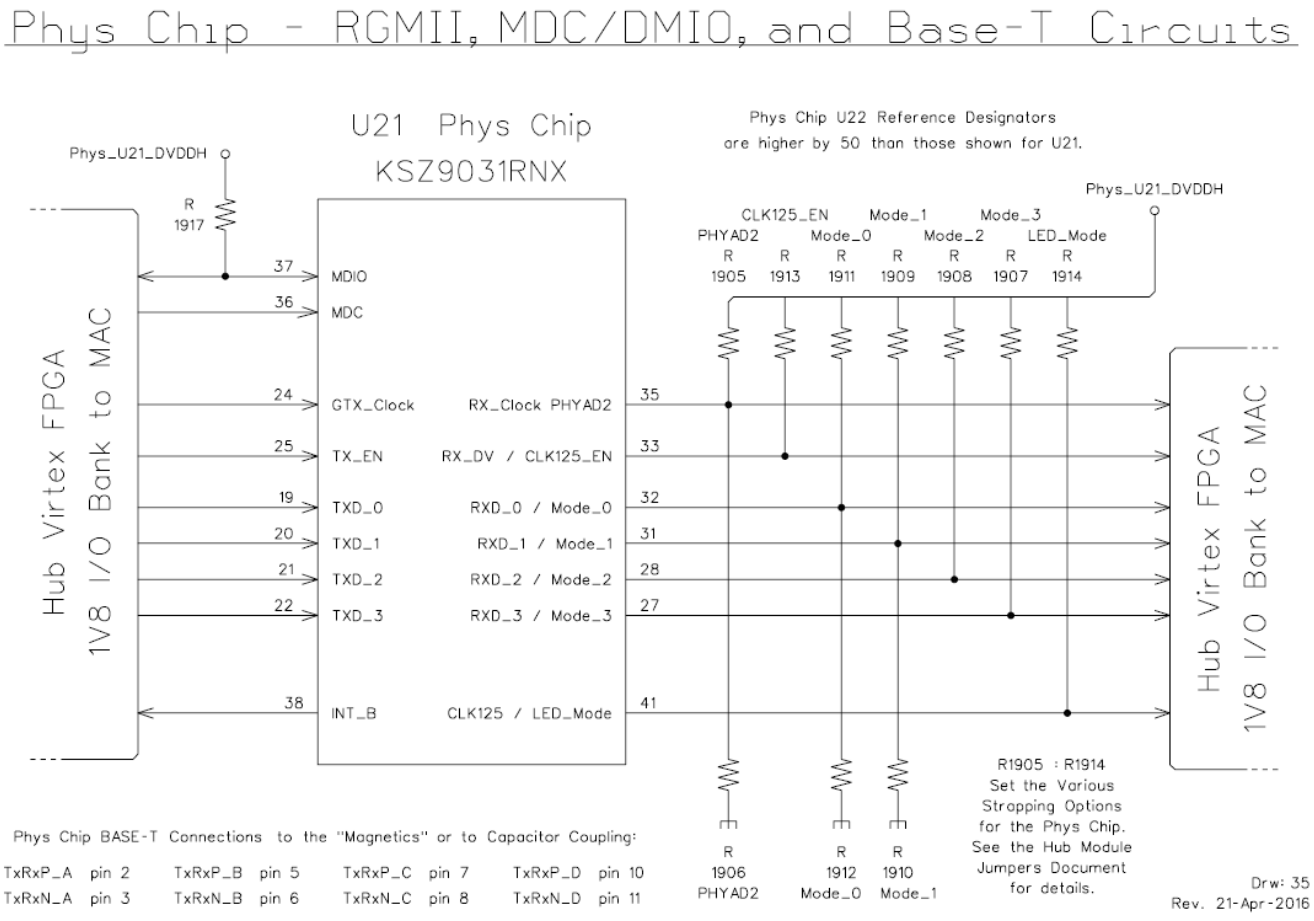


Figure 4: Scheme of Phys Chip: RGMII, MDC/DMIO, and Base-T Circuits.

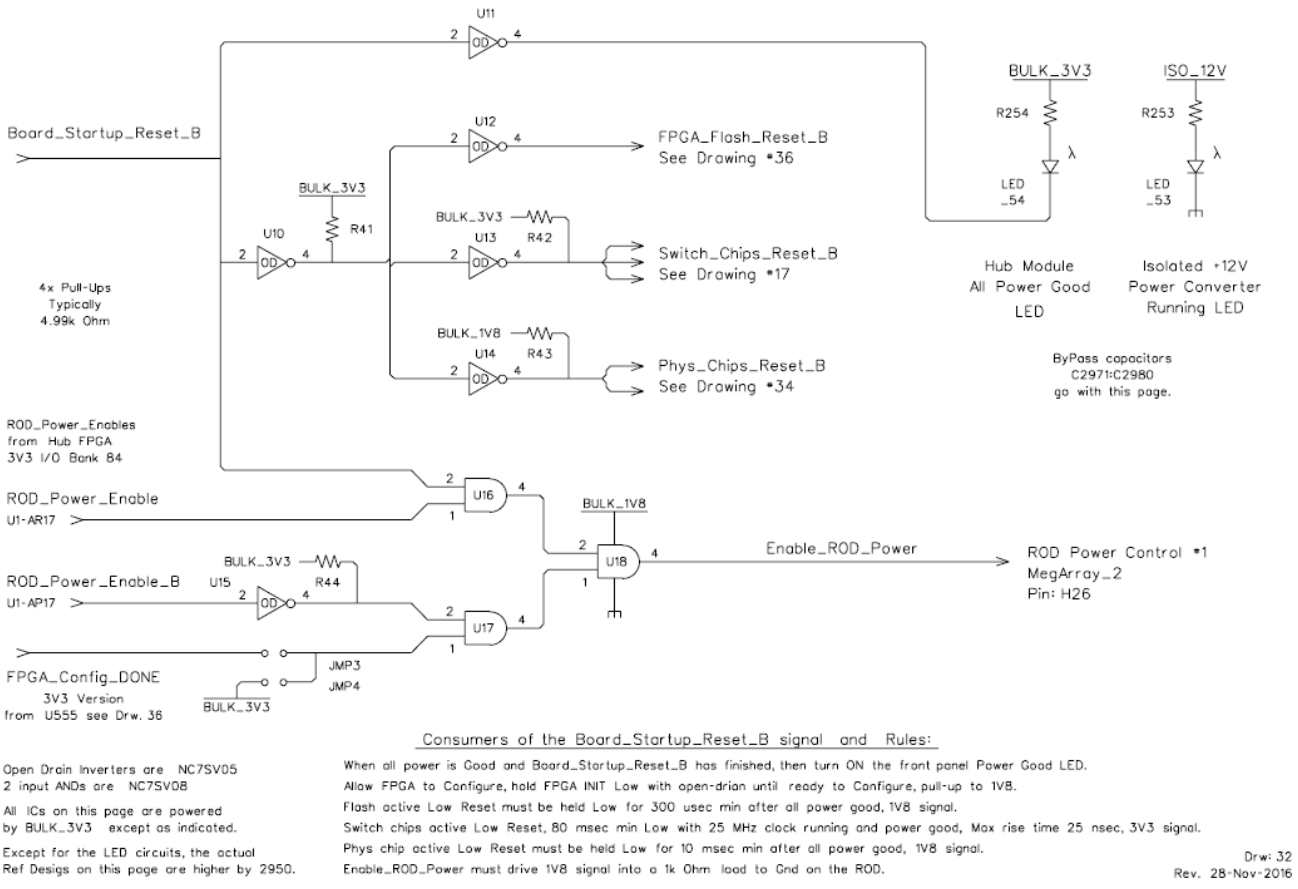


Figure 5: Board Reset Distribution, ROD Power Control.

For each of its two Ethernet Phys Chips (PHY) the Hub's FPGA will need to instance a MAC that supports an RGMII connection (along with MDIO/MDC lines) to the PHY. All of these signals to/from the PHY are currently routed through the 1V8 HP Select I/O Bank 68.
After power-up the KSZ9031RNX is configured to RGMII mode if the MODE [3:0] strap-in pins are set to one of the RGMII mode capability options.
There is no reset signal to the KSZ9031RNX from FPGA. An ad hoc manual push button was attached to the PHY chip on the HUB for debugging purposes.
The KSZ9031RNX RGMII port connects to HP I/O pins on the FPGA. The RGMII port consists of 12 signals:

- Transmit Clock to the KSZ9031RNX
- Transmit Control (enable) to the KSZ9031RNX
- Transmit Data 0:3 to the KSZ9031RNX
- Receive Clock from the KSZ9031RNX
- Receive Control (enable) from the KSZ9031RNX
- Receive Data 0:3 from the KSZ9031RNX

The KSZ9031RNX includes a MII Management port. This type of port is also called MDIO Management Data Input/Output. This port allows higher-level devices to monitor and control the KSZ9031RNX. This port allows direct access to the IEEE defined MIIM registers, and the vendor specific registers. This port also allows indirect access to the MMD address space and registers. This port consists of signals: MDC - the clock and MDIO - the data line.
The Hub Module has two KSZ9031RNX PHY chips. There are 14 jumpers associated with each of these PHY chips. These jumpers are resistors that bias a pin in one direction or the other and this value is read when the PHY chip first powers up or is reset.
The KSZ9031RNX has 9 pins (called "Strapping Options") that are read in this way at power up. Because of space limitations and because there is an obvious why that the Hub Module wants some of these Strapping Options set, 4 of them have only one jumper to pull that pin in the direction that is obviously needed for rational operation of the Hub Module.
The PHYADx jumpers set the address of the Management Interface Port on the KSZ9031RNX. The Management Port PHYAD bits 3 and 4 are internally always set to 0,0. Bits 2, 1 and 0 set to Low. Therefore, the PHYADx set to 0.
The Hub Module provides easy control of only the Mode_0 and Mode_1 lines. This provides the following 4 options for the Phys chip (Mode bits listed Mode_3, ..., Mode_0).

1100 RGMII 1000 Base-T full duplex only
1101 RGMII 1000 Base-T full or half duplex
1110 RGMII 10/100/100 all but 1000 half duplex
1111 RGMII 10/100/1000 full or half duplex

Mode: SET MODE {3..0} = 1100 - RGMII 1000 Base-T full duplex only

The Xilinx Tri-Mode Ethernet MAC core is a parameterizable core, in 1000 Mbps mode, the TEMAC core can also connect with industry standard PHY devices. Optional MDIO interface to managed objects in PHY layers (MII Management).

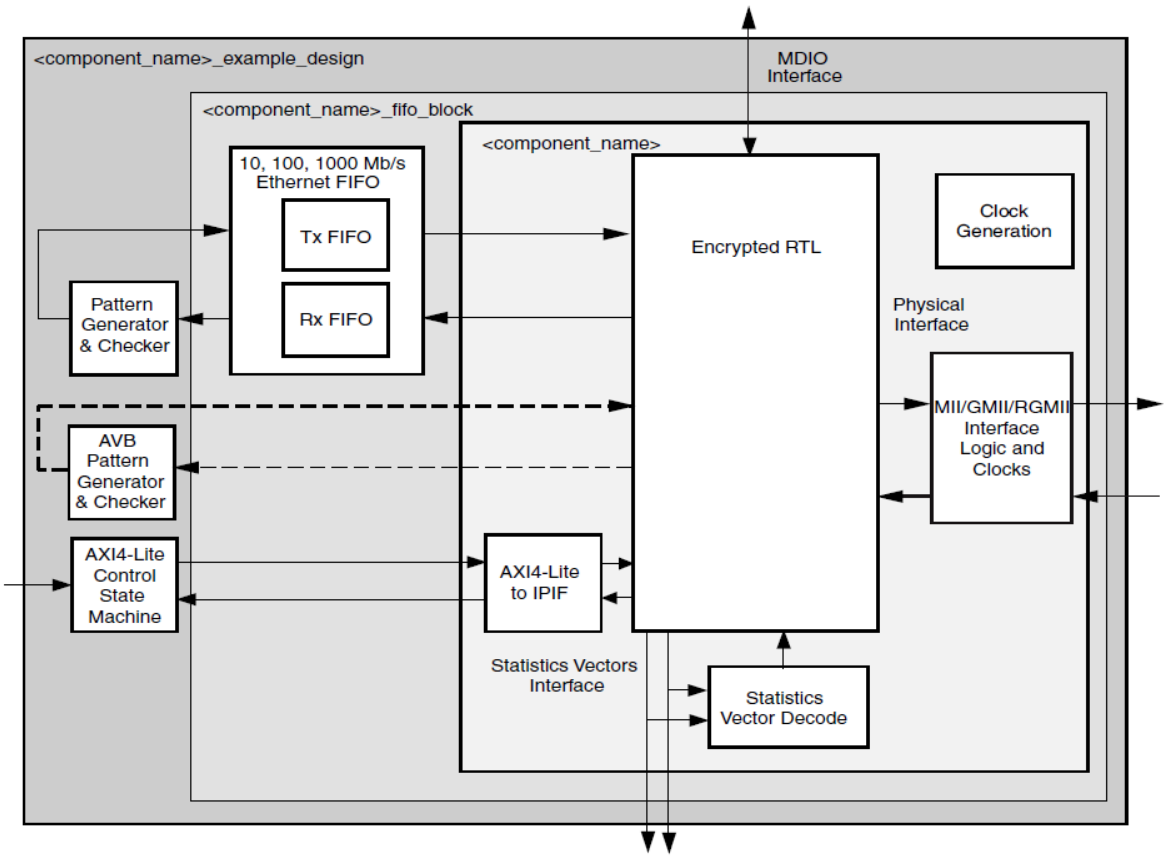


Figure 6: Scheme of Xilinx MAC Example HDL Design (UltraScale).

The following design approach is based on the suggestion by Ed Flaherty (University of Cambridge):
Step 1: Generate Xilinx MAC Example Design (UltraScale RGMII).
Step 2: Modify the Example Design to HUB board hardware.
Step 3: Generate Tx packets with built-in Simple Frame Generator and capture them in Wireshark.
Step 4: Packet generation logic replaced with simple read fifo (Rx looped back to Tx).
Step 5: Packets sent from server (Ostinato) returned via the loopback proving Rx and Tx paths.
Step 6: Use Wireshark to check returned packets

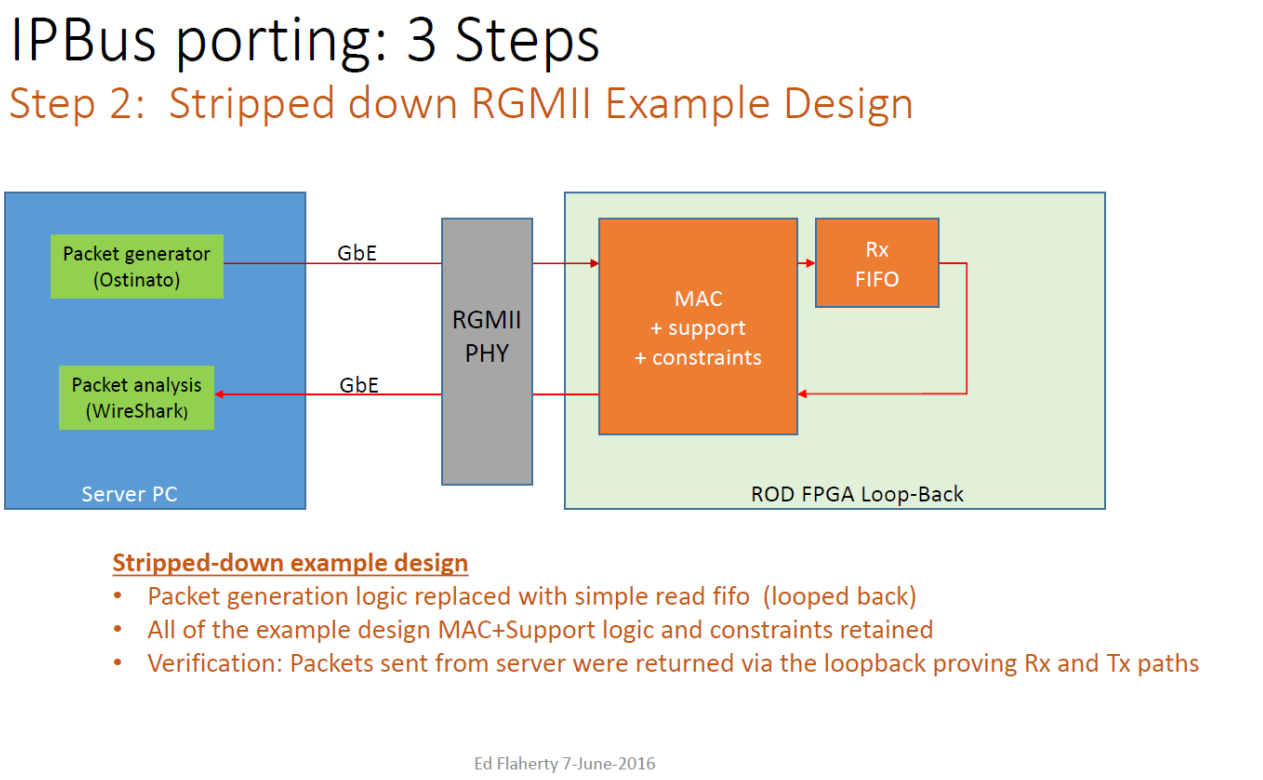


Figure 7: IPBus porting.

As soon as the MAC part works, the IPbus control part and the IPbus slaves can be added to the design, as described in the note: Notes on Firmware Implementation of an IPbus SoC Bus, V1.0 23/5/2012, DMN".

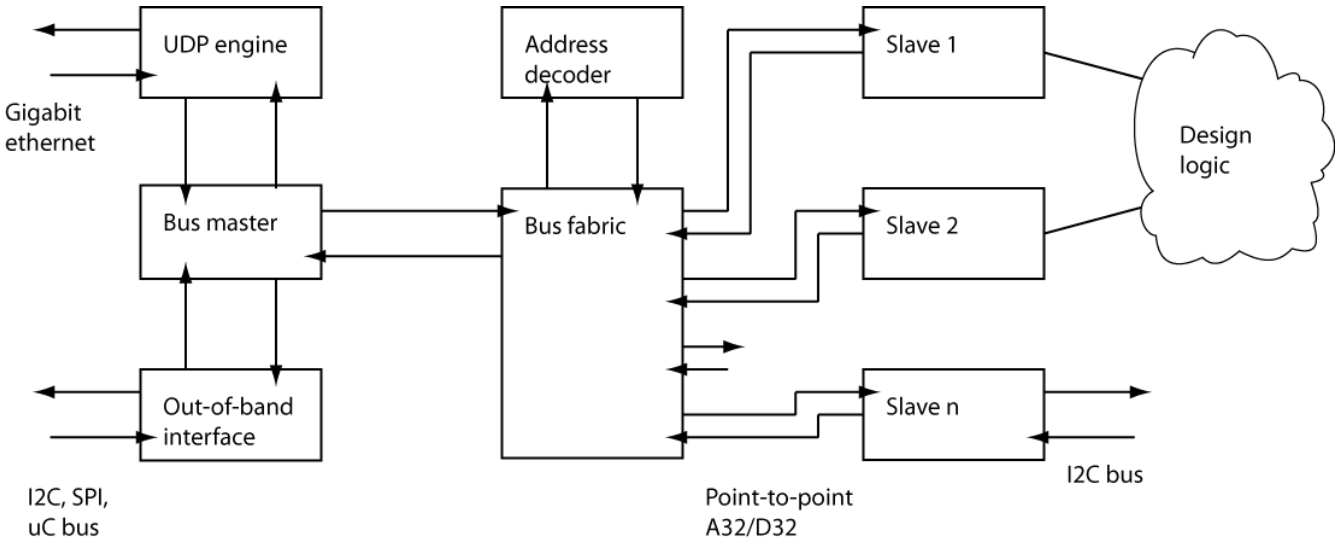


Figure 8: IPBus topology.

The SoC bus is fully synchronous, and operates from a single system clock. There is no constraint onthe relationship of the bus clock to the 125MHz GbE clock, as the firmware contains handshaking logic. For slaves, which do not require wait states, the 32b data path on the SoC bus allows full utilization of the Ethernet interface as long as it runs at >32MHz; this allows straightforward timing closure for complex multi-slave designs. Designs therefore typically drive the Ipbus clock at ¼ of the GbE clock (i.e. 31.25MHz).

2. IPbus Register Map

```
                                                                    Current Rev.  05-Nov-2018
All IPbus registers are 32-bit. Address is HEX
Bit allocation inside the register can be changed if needed

Overview: (*) - not all bits are implemented
-------------------------------------------

1. Set of registers for the external FPGA control/status pins (IO)
00000000  RW  control          HUB + HTM(*)
00000001  RW  lock             HUB + HTM
00000002  RW  leds             HUB + HTM
00000003  RW  test             HUB + HTM
00000004  RO  module           HUB + HTM
00000005  RO  address          HUB + HTM(*)
00000006  RO  status           HUB + HTM(*)
00000007  RO  tmp              HUB + HTM

2. Test/Monitoring registers
00000008-0000000F testreg      HUB + HTM  8 registers
00000010  RW  w_count          HUB + HTM
00000011  RW  r_count          HUB + HTM
00001000-000013FF ram          HUB + HTM  1K registers
00002000  RW  pram_addr        HUB + HTM  10-bits (1K RAM)
00002001  RW  pram_data        HUB + HTM  32-bit data

3. Block of 8 GPIO RW registers in the FPGA Block Design for tests
40000000-40070000 gpio_regs    HUB + HTM  8 registers

4. Transmitter and Receiver MiniPOD Control registers
40800000  Trans_MiniPOD        HUB + HTM
40810000  Recvr_MiniPOD        HUB + HTM

5. Registers for the interface to the DCDC Converters
40820000  DCDC Converters      HUB    -

6. Registers for the interface to the Si Lab 5338A
40830000  HTM's SiLab 5338A     -    HTM

7. Registers for the interface to the switches
40E00000  Chip A               HUB    -
40E10000  Chip B               HUB    -
40E20000  Chip C               HUB    -

8. System Monitor registers
44A00000  System Monitor       HUB + HTM

9. Test registers
00000012  RW  test2            HUB + HTM
00001400  RW  test3            HUB + HTM
00002002  RW  test4            HUB + HTM
40080000  RW  test5_1          HUB + HTM
40801000  RW  test5_2          HUB + HTM
40811000  RW  test5_3          HUB + HTM
40821000  RW  test5_4          HUB + HTM
40831000  RW  test5_5          HUB + HTM
```

```
40E01000  RW  test5_6          HUB + HTM
40E11000  RW  test5_7          HUB + HTM
40E21000  RW  test5_8          HUB + HTM
44A01000  RW  test5_9          HUB + HTM
```

10. Control/status registers for the internal FPGA logic - TBD

Additional information about HUB/HTM FPGA signal Types can be found in:
HUB: https://web.pa.msu.edu/hep/atlas/l1calo/hub/hardware/details/hub_0_ab_fpga_signal_types.txt
HTM: https://web.pa.msu.edu/hep/atlas/l1calo/htm/hardware/details/htm_fpga_mezz_connections.txt

1. Set of registers for the external FPGA control/status pins (IO)

Addr	Type	Name	Content		Bit	Comment
00000000	RW	control	other_hub_Clk		[31]	FPGA internal, selects clock from other Hub
			fex_clk_en		[30]	Enable clock to FEXs
			mgt_equ_en		[29:17]	Enable equalization in MGT Fanout chips
			i2c_buf_en		[16]	I2C_Buf_1501_ENABLE
					[15]	I2C_Buf_1502_ENABLE
					[14]	I2C_Buf_1503_ENABLE
			spare		[13-11]	spare bits
			rod_pwr_en	?	[10]	ROD may turn ON its power supplies
			rod_pwr_en_b	?	[9]	ROD may turn ON its power supplies
			spare		[8:6]	spare bits
			mpod_rst_b	*	[5]	Trans_MiniPOD_RESET_B
				*	[4]	Recvr_MiniPOD_RESET_B
			spare		[3:0]	spare bits

This is control register for the individual pins (exeption - other_hub_Clk, Drw: 40A)
All bits set to '0' after power-on.

HUB: all bits are implemented
HTM: only bits with (*) are implemented - mpod_rst_b

NB: rod_pwr_en and rod_pwr_en_b needs clarification

Addr	Type	Name	Content	Bit	Comment
00000001	RW	lock	arbitration	[31:2]	access key
				[1]	lock bit
				[0]	watchdog bit

A single R/W register to arbitrate access between multiple IPbus clients and implement timeout
Bits 31 downto 2 provide an access key, bit 1 is the lock bit, bit 0 is the watchdog bit
Generic IPbus arbitration, David Sankey, Monday, 12 June 2017: https://indico.cern.ch/event/641268/

Addr	Type	Name	Content	Bit	Comment		
00000002	RW	leds	LEDs	[31]	leds_control		
					HUB:	HTM:	
						[7]	LED_1_TOP_CONTROL
						[6]	LED_2_CONTROL
						[5]	LED_3_CONTROL
						[4]	LED_4_CONTROL
						[3]	LED_5_CONTROL
				[2]	Front palnel led50	[2]	LED_6_CONTROL
				[1]	Front palnel led51	[1]	LED_7_CONTROL
				[0]	Front palnel led52	[0]	LED_8_BOT_CONTROL

This register is implemented in the HUB and the HTM

Lower two LEDs display the IPbus Rx and Tx transactions when leds_control is set to '0'
(also after power-on),

```
HUB:                                        HTM:
Front palnel led51 <= IPbus pkt_rx_led;     LED_7_CONTROL     <= IPbus pkt_rx_led;
Front palnel led52 <= IPbus pkt_tx_led;     LED_8_BOT_CONTROL <= IPbus pkt_tx_led;
```

When leds_control is set to '1', they are controlled by this register (see above).

All other LEDs are controlled by this register independently of leds_control.

Addr	Type	Name	Content		Bit	Comment
00000003	RW	test	xxxxxxxx		[31:0]	test register

Addr	Type	Name	Content		Bit	Comment
00000004	RO	module	module_type		[31:24]	HUB: X"01" HTM: X"02"
			hw_revision		[23:16]	HUB: X"01" HTM: X"01"
			fw_type		[15:8]	HUB: X"01" HTM: X"01"
			fw_version		[7:0]	HUB: X"01" HTM: X"01"

This is status information, set internally in the FW, will be properly set upou decision.

Addr	Type	Name	Content		Bit	Comment
00000005	RO	address	shelf_addr		[31:24]	Shelf Address from the IPMC
			slot_addr	*	[23:16]	Backplane Hardware Slot Address
			addr_to_rod		[15:8]	Hardware Address to the ROD (generated)
			spare		[7:0]	spare bits

This is external information (shelf_adrs and slot_adrs) and internally generated (addr_to_rod).

HUB: all bits are implemented
HTM: only bits with (*) are implemented - Backplane Hardware Slot Address

Addr	Type	Name	Content		Bit	Comment
00000006	RO	status	pll_lock	*	[31]	From PLL lock circuits - pll40_lock
					[30]	From PLL lock circuits - pll320_lock
			phy_int	*	[29]	From Eth Phy chips - phyU21_int_b
					[28]	From Eth Phy chips - phyU22_int_b
			mpod_int	*	[27]	From the MiniPODs - trans_mpod_intr_b
				*	[26]	From the MiniPODs - recvr_mpod_intr_b
			hub_smb_alert		[25]	From 7 DCDC Converters
			hub_pwr_good		[24]	From the Power Control circuits
			rod_present_b		[23]	ROD is present
			rod_smb_alert_b		[22]	Power supply problem on the ROD
			rod_status		[21]	ROD power
					[20]	ROD config
			rod_status		[19]	spare rod_power_ctrl4
			sw_loop_det		[18]	Broadcom Switch chips switch_a_loop_det
					[18]	Broadcom Switch chips switch_b_loop_det
					[18]	Broadcom Switch chips switch_c_loop_det
			spare		[15:0]	spare bits, set to all zeros

This is a status information from individual pins.
HUB: all bits are implemented
HTM: only bits with (*) are implemented - [31] PLL_40_MHZ_LOCKED_MON
[29] ETH2_INT
[27] MP_TRANS_INTR
[26] MP_REC_INTR

Addr	Type	Name	Content	Bit	Comment
00000007	RO	tmp	test_bits	[31:0]	for tests, now set to all zeros

2. Test/Monitoring registers

Addr	Type	Name	Content	Bit	Comment
00000008-0000000F	RW	testreg	xxxxxxxx	[31:0]	8 test RW registers

Block of 8 RW registers in the FPGA logic for tests

Addr	Type	Name	Content	Bit	Comment
00000010	RW	w_count	xxxxxxxx	[31:0]	Write packet counter
00000011	RW	r_count	xxxxxxxx	[31:0]	Read packet counter

These two registers are packet counters for monitoring packet loss
Counters are reset by any write

Addr	Type	Name	Content	Bit	Comment
00001000-000013FF	RW	ram	block ram	[31:0]	1Kword array of registers

Addr	Type	Name	Content	Bit	Comment
00002000	RW	pram_addr	address	[9:0]	1Kword test RAM pointer register
00002001	RW	pram_data	data	[31:0]	1Kword test RAM data register

3. Block of 8 GPIO RW registers in the FPGA Block Design for tests

Addr	Type	Name	Content	Bit	Comment
40000000	RW	gpio_0	test register	[31:0]	Base address to GPIO test register
40010000	RW	gpio_0	test register	[31:0]	Base address to GPIO test register
40020000	RW	gpio_0	test register	[31:0]	Base address to GPIO test register
40030000	RW	gpio_0	test register	[31:0]	Base address to GPIO test register
40040000	RW	gpio_0	test register	[31:0]	Base address to GPIO test register
40050000	RW	gpio_0	test register	[31:0]	Base address to GPIO test register
40060000	RW	gpio_0	test register	[31:0]	Base address to GPIO test register
40070000	RW	gpio_0	test register	[31:0]	Base address to GPIO test register

Addr	Type	Name	Content	Bit	Comment
40080000	RW	test5_1	xxxxxxxx	[31:0]	test RW register

4. Transmitter and Receiver MiniPOD Control registers

Addr	Type	Name	Content	Bit	Comment
40800000		trans_mpod			Base address to access Trans_MiniPOD
40810000		trans_mpod			Base address to access Recvr_MiniPOD

These two registers are base addresses to access to the Transmitter and Receiver MiniPODs.
The I2C master interfaces are implemented as XILINX IP in the AXI sybsystem.
Each interface provides a set of 22 RW registers to generate I2C bus transactions.

5. Registers for the interface to the DCDC Converters

Addr	Type	Name	Content	Bit	Comment
40820000		hub_dcdc			Base address to access Hub's DCDC Converters

This Master I2C bus makes connection to the Hub's DCDC Converter power supplies.
The I2C master interface is implemented as XILINX IP in the AXI sybsystem.
It provides a set of 22 RW registers to generate I2C bus transactions.

6. Registers for the interface to the Si Lab 5338A

Addr	Type	Name	Content	Bit	Comment
40830000		htm_silab			Base address to access HTM's SiLab 5338A

This Master I2C bus makes connection to the HTM's SiLab 5338A.
The I2C master interface is implemented as XILINX IP in the AXI sybsystem.
It provides a set of 22 RW registers to generate I2C bus transactions.

7. Registers for the interface to the switches

Addr	Type	Name	Content	Bit	Comment
40E00000		switch_a			Base address to access Ethernet Switch Chip A
40E10000		switch_b			Base address to access Ethernet Switch Chip B
40E20000		switch_c			Base address to access Ethernet Switch Chip C

These three registers are base addresses to access to the Ethernet Switch Chips.
The MDIO interfaces are implemented as XILINX IP in the AXI system.
Each interface provides a set of 4 RW registers to generate MDIO bus transactions.

8. System Monitor registers

Addr	Type	Name	Content	Bit	Comment
44A00000		hub_sysmon	volt&temp		Base address to access FPGA System Monitor

This is a base register to access the FPGA SysMon from IPbus.

Interface to the System Monitor can be implemented with AXI interface and read via IPbus,
I2C interface of the System Monitor will be connected to the Slave I2C interface.

9. Test registers

Addr	Type	Name	Content	Bit	Comment
00000012	RW	test2	xxxxxxxx	[31:0]	test RW register
00001400	RW	test3	xxxxxxxx	[31:0]	test RW register
00002002	RW	test4	xxxxxxxx	[31:0]	test RW register
40080000	RW	test5_1	xxxxxxxx	[31:0]	test RW register
40801000	RW	test5_2	xxxxxxxx	[31:0]	test RW register
40811000	RW	test5_3	xxxxxxxx	[31:0]	test RW register
40821000	RW	test5_4	xxxxxxxx	[31:0]	test RW register
40831000	RW	test5_5	xxxxxxxx	[31:0]	test RW register
40E01000	RW	test5_6	xxxxxxxx	[31:0]	test RW register
40E11000	RW	test5_7	xxxxxxxx	[31:0]	test RW register
40E21000	RW	test5_8	xxxxxxxx	[31:0]	test RW register

```
44A01000  RW    test5_9      xxxxxxxx        [31:0]   test RW register


10. Control/status registers for the internal FPGA logic
-------------------------------------------------------------------------------------------------
Addr      Type  Name         Content         Bit   Comment
-------------------------------------------------------------------------------------------------

These registers will provide control/status information for the internal FPGA logic:

- HUB-ROD interface
- HUB-GBT/TTC interface
- HUB-IPMC interface

-------------------------------------------------------------------------------------------------
```

Table 1: The initial HUB FPGA register map.

3. TTC Clock recovery/GBT

The Hub in Slot 1 uses a 12-channel MiniPOD optical receiver to receive TTC signals from the upstream FELIX system. The Hub receives two types of TTC signals: a copy of the LHC clock and TTC control data.

3.1 Implementation

Trigger system require a fixed and deterministic latency in the transmission of the clock and data. In order to recover the LHC clock with the fixed latency, the HUB FW uses the GBT-FPGA component [8,9] which is being developed and supported by EP-ESE-BE section at CERN. The GBT-FPGA firmware supports the "Latency-fixed" implementation. In addition, the GBT-FPGA supports all three available encoding/decoding scheme: "GBT-Frame" (Reed-Solomon), "Wide-Bus" and "8b10b". The HUB FW uses a "GBT-Frame" scheme [Table 2]. The GBT FW is implemented with MGT transceiver and complexcontrol logic [Figure 9]. It is set to run at 4.8 Gbps (MGT REF CLK – 320.64 MHz, User/Internal Data Width – 40 bits, CPLL or QPLL).

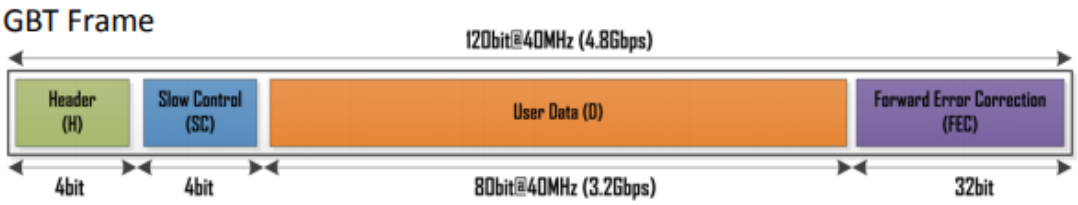


Table 2. GBT-Frame (based on Reed-Solomon) structure.

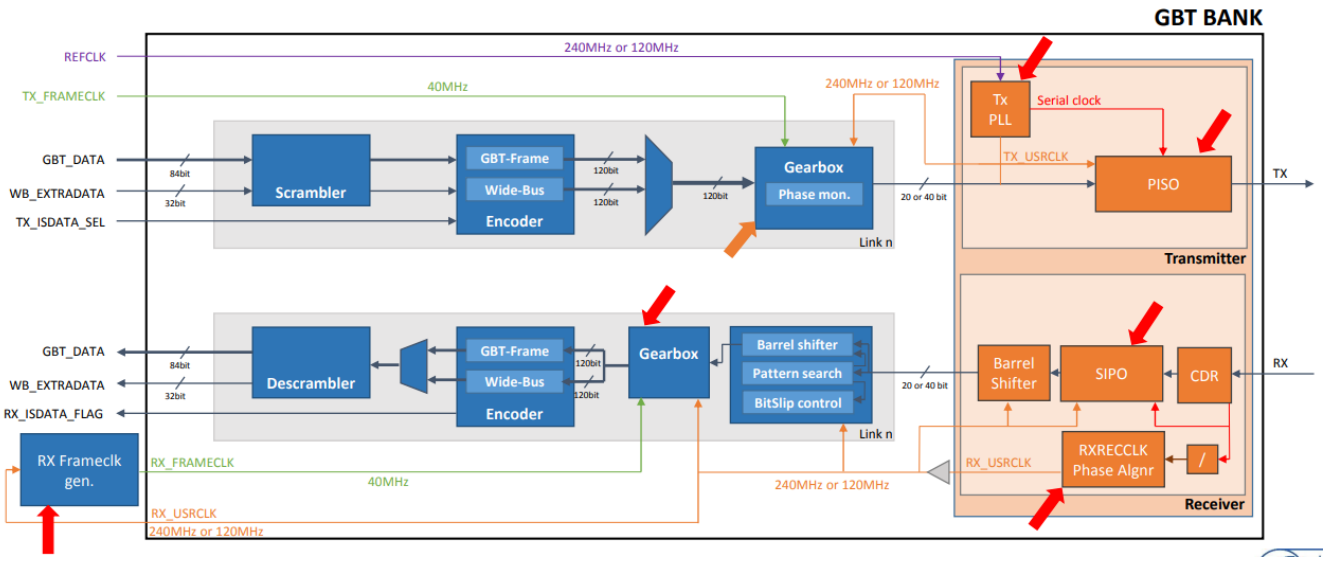
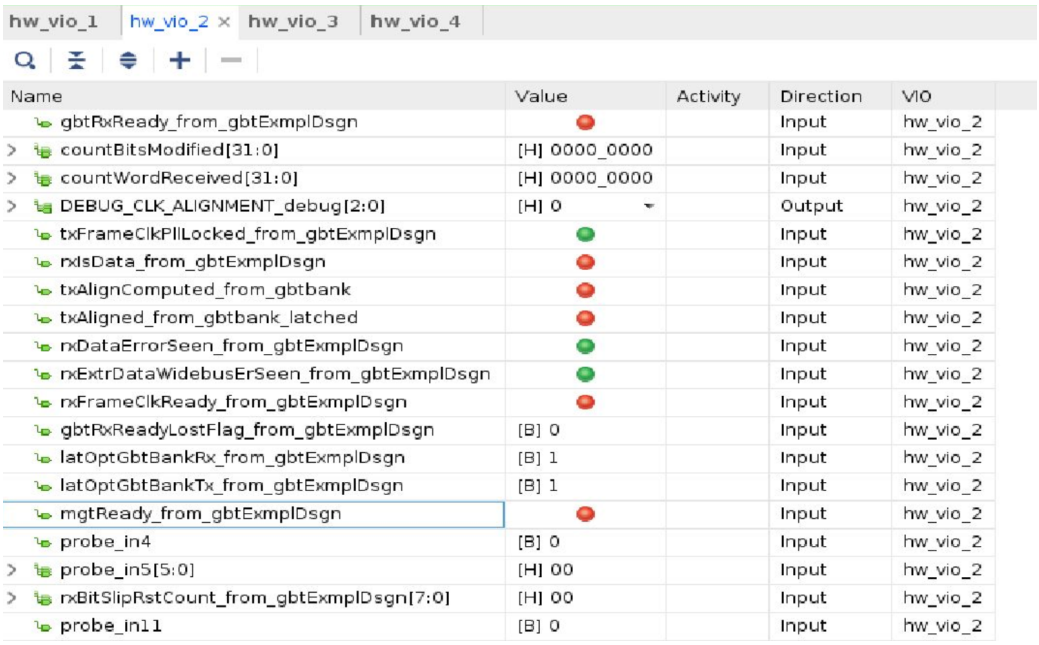


Figure 9. The GBT-FPGA implementation method.

3.2 Latency Measurement

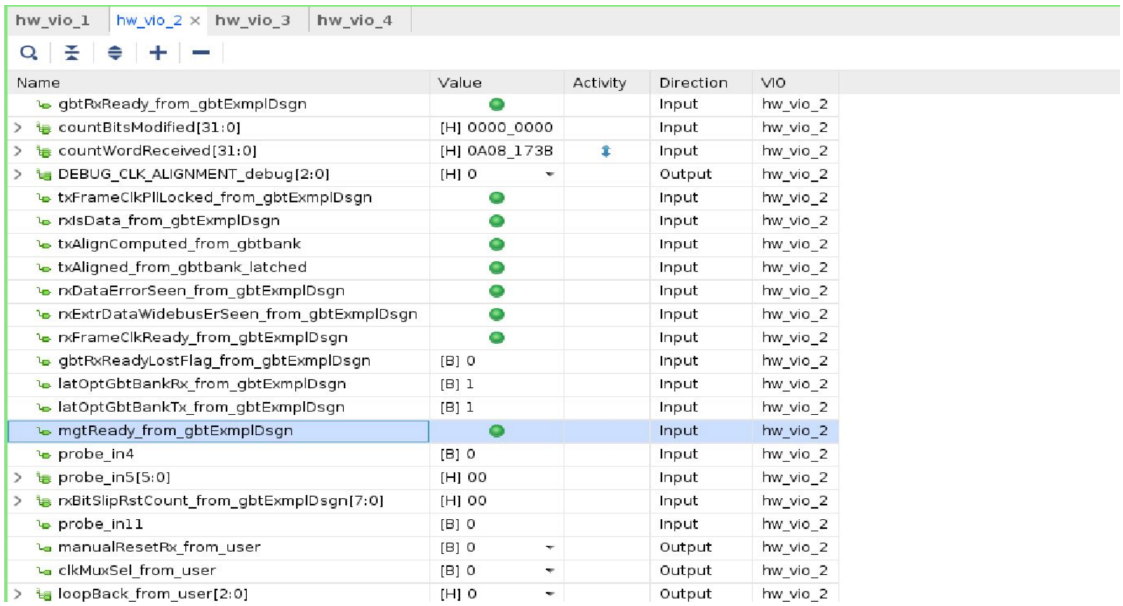
The Hub module successfully recovers the 40.08 MHz LHC [Figure 10] clock using the GBT-FPGA core in "latency-optimized" mode. The MGT eventually tracks the 4.8 Gbps serial data stream, no relevant phase noise due to rate conversion. The "Frame Aligner" status and "Bit Slip Control" status are monitored to ensure proper GBT functionality. The clock phase observed to be constant (to better than 100 ps) over a range of repeated tests.

GBT/MGT status if the reset is hold:



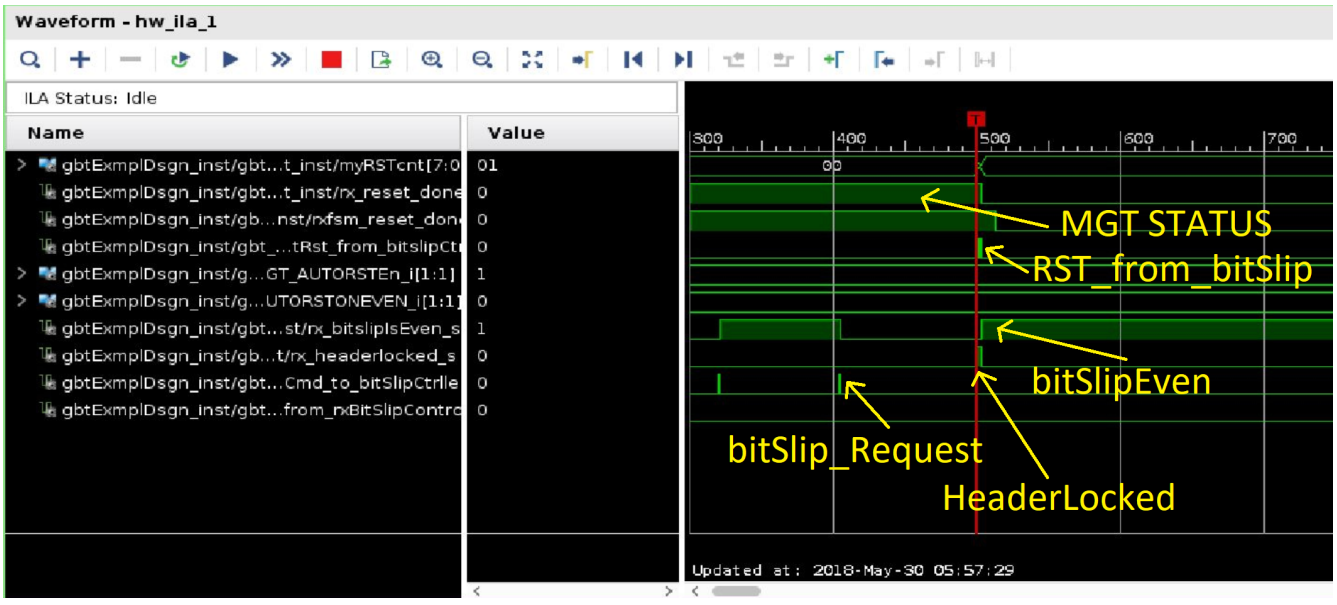
Name	Value	Activity	Direction	VIO
gbtRxReady_from_gbtExmplDsgn			Input	hw_vio_2
countBitsModified[31:0]	[H] 0000_0000		Input	hw_vio_2
countWordReceived[31:0]	[H] 0000_0000		Input	hw_vio_2
DEBUG_CLK_ALIGNMENT_debug[2:0]	[H] 0		Output	hw_vio_2
txFrameClkPllLocked_from_gbtExmplDsgn			Input	hw_vio_2
rxIsData_from_gbtExmplDsgn			Input	hw_vio_2
txAlignComputed_from_gbtbank			Input	hw_vio_2
txAligned_from_gbtbank_latched			Input	hw_vio_2
rxDataErrorSeen_from_gbtExmplDsgn			Input	hw_vio_2
rxExtrDataWidebusErSeen_from_gbtExmplDsgn			Input	hw_vio_2
rxFrameClkReady_from_gbtExmplDsgn			Input	hw_vio_2
gbtRxReadyLostFlag_from_gbtExmplDsgn	[B] 0		Input	hw_vio_2
latOptGbtBankRx_from_gbtExmplDsgn	[B] 1		Input	hw_vio_2
latOptGbtBankTx_from_gbtExmplDsgn	[B] 1		Input	hw_vio_2
mgtReady_from_gbtExmplDsgn			Input	hw_vio_2
probe_in4	[B] 0		Input	hw_vio_2
probe_in5[5:0]	[H] 00		Input	hw_vio_2
rxBitSlipRstCount_from_gbtExmplDsgn[7:0]	[H] 00		Input	hw_vio_2
probe_in11	[B] 0		Input	hw_vio_2

GBT/MGT status if the reset is released:



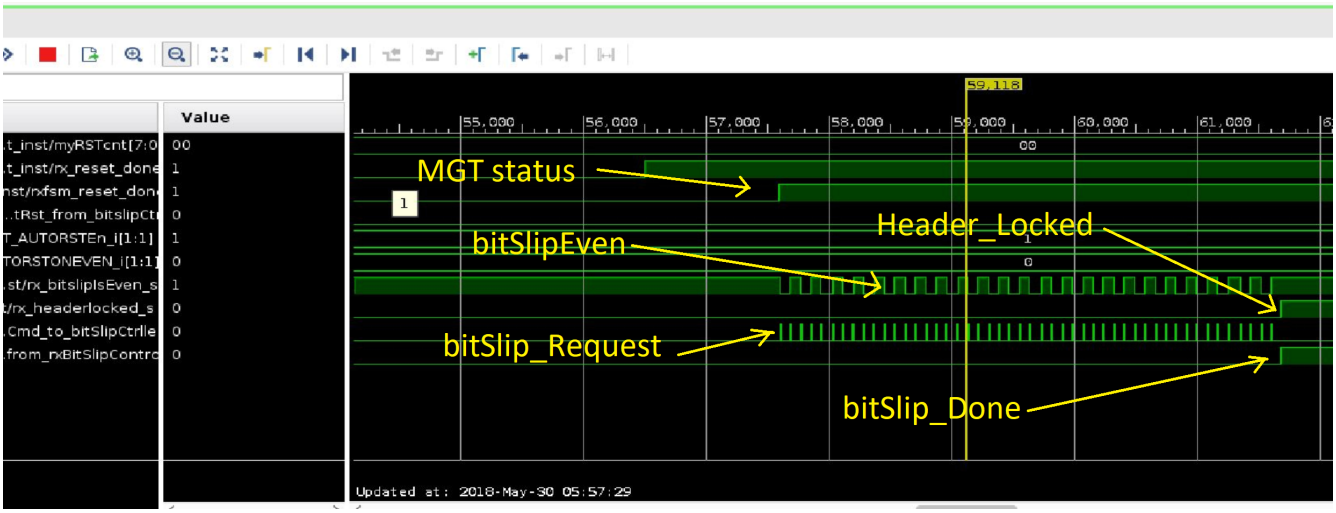
GBT startup, part 1:

The MGT status OK. Th bitSlipEven, *must* be *odd*. But, in this case is *even*, that's why the MGT RST is issued. The GBT startup procedure will be repeated.



GBT startup, part 2:

The MGT status OK. The bitSlipEven, must be *odd*.
Yes, in this case is *odd*. The GBT startup is completed.



The 40.08 MHz LHC recovered clock (light blue)

- Manual intervention, force GBT realignment
- Reset receiver MGT on Hub
- Power cycle Hub
- Reconfigure Hub FPGA
- Unplug/reinsert TTC fiber @ TTCfx/VC109
- Reconfigure TTCfx/VC709
- Re-initialize the TTCfx/VC709
- Reset of the TTCfx/VC709

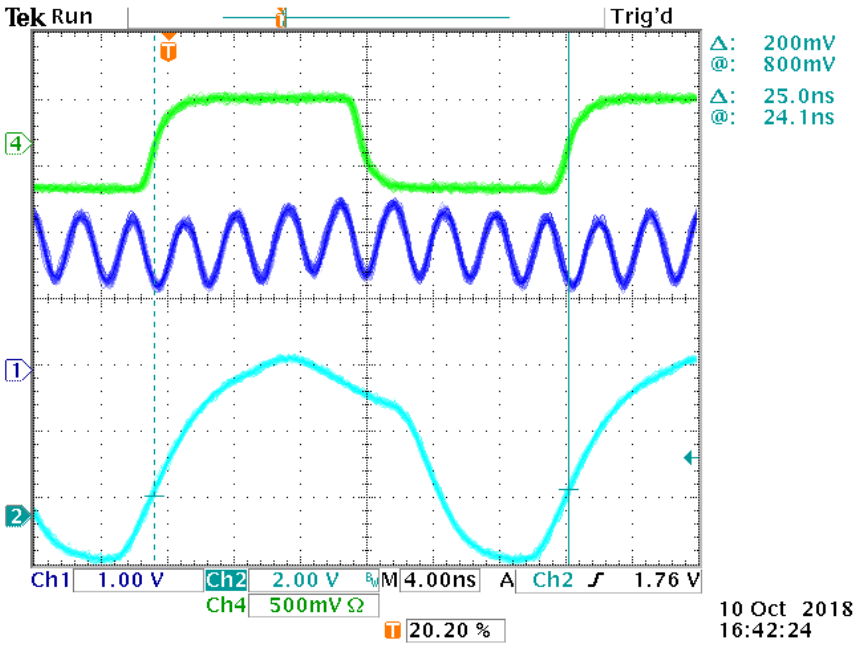


Figure 10. The Hub module successfully recovers the 40.08 MHz LHC.

4.0 Readout Control (Readout_Ctrl)

The HUB module is obliged to receive the Readout Control (Readout_CTRL) information from the ROD via serial link named Readout_CTRL [2]. 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

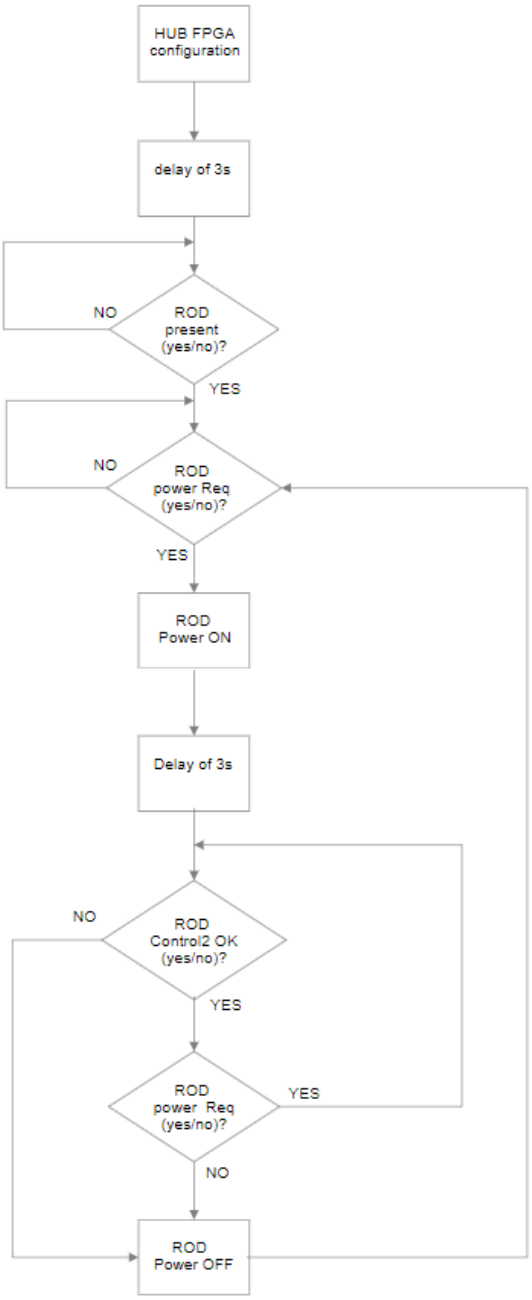
fanned out to the rest of the system. The main purpose is to provide resets to all of the data links (Aurora) between the Fex's and the ROD plus HUB module.

4.1 ROD Power Up scheme

When the HUB FPGA is configured, ROD is power-up automatically. This procedure is implemented with the use of the state machine. Once the HUB FPGA is configured, the logic is obliged to check if the ROD is present. In order to do so, the ROD_PRESENT_B_TO_FPGA signal is check. This signal is a pull-down resistor on the ROD when ROD is not present this signal goes 'HI' on the HUB. In addition, the IPbus register (ROD_Power_Req) provides the information if there is a request to power-up the ROD. If the ROD power Req goes 'HI', the ROD is powered-up. This is done with the use of two signals: ROD_Power_Enable and ROD_Power_Enable_B. These two signals are output from the HUB FPGA that through some hardwired logic on the HUB inform the ROD when it may turn ON its power supplies. After the delay of 3 seconds, the logic starts to monitor the ROD_Power_Control_2_FPGA signal. When 'HI', this signal indicates that all power supplies on the ROD are operating correctly. If there are no problems with the power supplies, the logic checks if the ROD_Power_Req is 'HI'. If so, the ROD_Power_Control_2_FPGA signal is also checked. In cases where, there is a problem with the power supplies or there is a request to power-down the ROD, the ROD is powered-down and the engine returns to its initial state.

The ROD_Power_Req signal is treated as an extra signal for the test purposes and it might be removed from the power up engine in the production firmware.

In addition, there are two signals that are not being used by the power up engine: ROD_Power_Control_3_FPGA, this is an input to the HUB's FPGA. When 'HI' this signal indicates that the ROD is configured and fully ready for normal L1Calo operation. The FPGA_RODs_SMBALERT_B, this is an input to the HUB's FPGA that when 'LOW' indicates a power supply problem on the ROD. During normal operation this signal should be 'HI'.



1. Signals that are being used by the power up engine

1.a ROD_PRESENT_B_TO_FPGA

This signal is a pull-down resistor on the ROD. When the ROD is NOT present this signal goes HI on the Hub.

1.b ROD_Power_Control_2_FPGA

This is an input to the Hub's FPGA. When HI this signal indicates that all power supplies on the ROD are operating correctly.

ROD_Power_Enable
ROD_Power_Enable_B

These two signal are outputs from the Hub FPGA that through some hardwired logic on the Hub tell the ROD when it may turn ON its power supplies.

1.c ROD power request is controlled by the IPbus register

2. Signals that are not being used by the power up engine

2.a ROD_Power_Control_3_FPGA

This is an input to the Hub's FPGA. When HI this signal indicates that the ROD is Configured and fully ready for normal L1Calo operation.

2.b FPGA_RODs_SMBALERT_B

This is an input to the Hub's FPGA that when LOW indicates a power supply problem on the ROD. During normal operation this signal should always be HI.

Figure 11. The ROD power up scheme

4.2 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. It will be held at "0000" through the initial debug phases, where many changes may occur.

ROD_XOFF

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. This control halts data from ALL FEX' s as a group. There is no mechanism for halting individual FEX's.
This bit remains active for the duration of the time that data should be paused.

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.

Slot_N_link_reset_M

These signals allow the ROD to reset individual links providing data to it. It is asserted when the ROD is attempting to bring up a link which has gone down for some reason.

The appropriate bits (as defined by logical slot) are placed into the Combined_TTC link to each node by the Hub. Within the Combined_TTC link, they are referred to as Link_Reset[3:0].

The node module (FEX) should use the signal to reset its Aurora interface plus any additional logic required to resume normal data transmission.

The active time of these bits is indeterminate, and may be only a few broadcast cycles. On the trailing edge (deactivation), the nodes should provide additional timing control for the GTReset and Reset signals on the Aurora interface.

The jFEX modules will electrically connect to logical slots 04, 05, 08, 09, 12, and 13. Each jFEX module has four independent Aurora data links which can be controlled separately. Four independent Link Resets are therefore provided for these logical slots.

In slots where multiple resets are provided, eFEX modules, having only one Aurora interface, should use only the M=0 reset bits.

Slot N Channel Up M

These bits report on whether each slot's Aurora Channel is 'UP' or 'Down'. The information is rebroadcast to each FEX slot in the so that the Aurora transmitter knows if the receiver side of the channel (the ROD) is up. If a channel goes down, the ROD shall attempt to reset both Rx and Tx ends of the link.

Slots 3-8 have four channel up status bits each. This is to cover the case of jFex where each of four processors has its own data link to the ROD. It is expected that eFex boards in these slots will only use the M=0 Channel Up bit.

Slot N Link Enable

The ROD may disable the links from some boards within the shelf. This would happen in the case where a link reset had been applied, however the link still does not come up. It can also be used during commissioning to eliminate processing of selected slots. Note that in the case of jFEX, all four aurora links should be disabled if the slot enable is not present (='1').

Shelf Number

These bits are currently reserved. There is a possibility to derive the Shelf Number from a rotary switch on the ROD. Use of this option is still TBD.

CRC (9-bit)

CRC is included to provide additional robustness on this link. Erroneous resets could cause loss of data. Choice of polynomial is from https://users.ece.cmu.edu/~koopman/crc/index.html

The chosen polynomial is 0x17d in Koopman format of 0x2fb in explicit+1 format.

The CRC is calculated over all four words of each transmission. When a CRC error occurs, all four of the associated words must therefore be ignored.

All bits within the four-word broadcast will be held active for at least two four-word broadcast periods in order to reduce the probability of missing an active signal due to CRC error.

4.3 Readout Control/Implementation

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

Table 3: Readout_Ctrl bit assignments.

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. When 8b/10b overhead is accounted for, the effective rate is reduced to 6.4 x 0.8 = 5.12 Gbps. In order to to control message transmission within a single LHC clock period, the lengh of the message is limited tp 128 bts. 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. The least significant byte of Word_0 is reserved for the 8b10b Comma character K28.5.

$$(1/(6.4\text{Gbps} * 0 .8)) * 128 \text{ bits} = 2.5\text{ns} = 1/40\text{MHz}$$

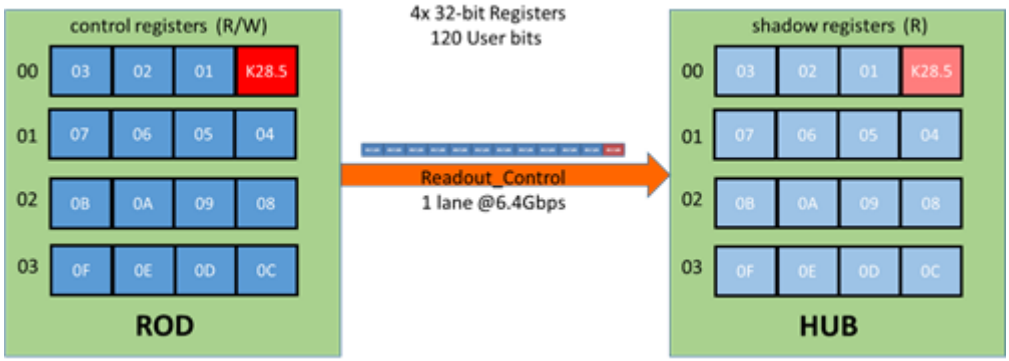


Figure 12. Control and Shadow Register

5. Combined_TTC/DATA

The HUB FW is obliged to distribute TTC information throughout the shelf [2]. There are several Combined_TTC links within the shelf, one between each Fex slot (slots 3-14) and each Hub, also one between each Hub and ROD and 2 links between two HUBs.
The additional feature allow to transmit the Reset signal (Aurora Initialization, Figure 12) from the Readout_Ctrl link and distribute it to the appropriate shelf slot. In total, we have the following number of links:

- 12 links between each FEX slot and each HUB
- 1 link between the ROD and each HUB
- 2 links between two HUBs

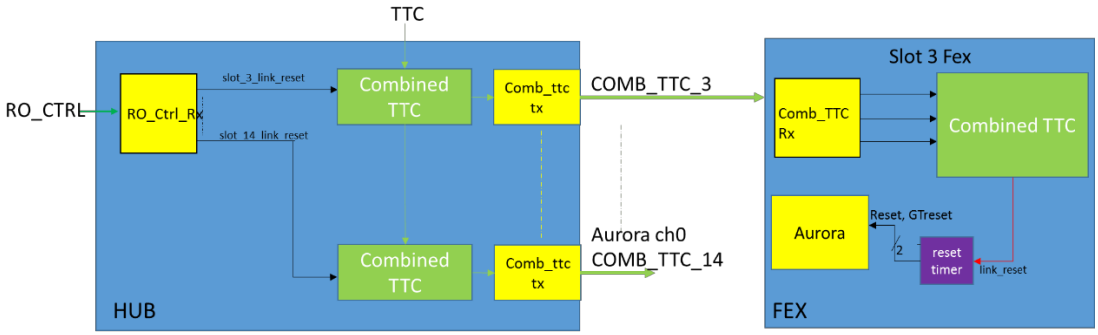


Figure 13. Example of link Reset Routing.

5.1 Combined_TTC/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 links 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.

The Combined_TTC/DATA FW development comprises several stages. In terms of Combined_TTC/DATA link, the first stage assumes to transmit the Combined_TTC/Data stream to the ROD module, also to the FEX slot 3 (see Figure 14 and 15). 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 GBT/TTC component. Next development steps assumes to add (gradually) the remaining receivers in the shelf.

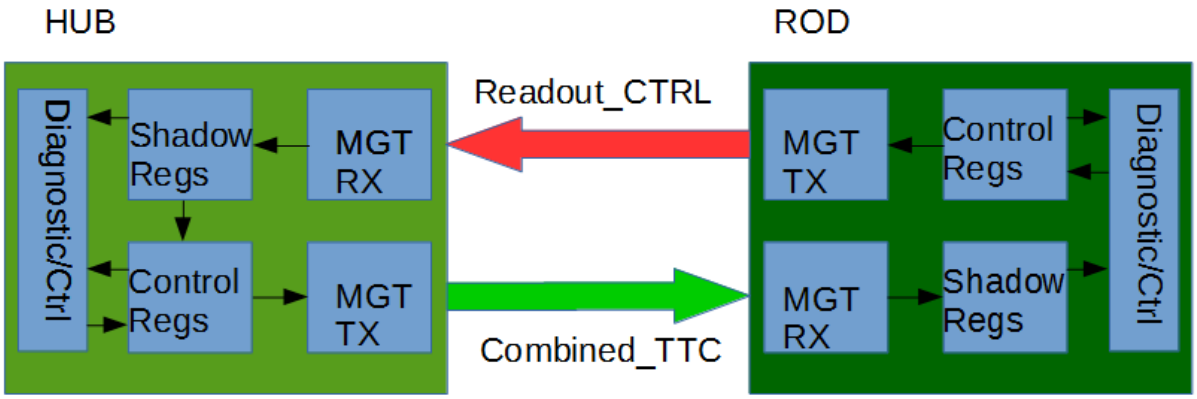


Figure 14. The HUB and ROD setup to test the Readout_Ctrl and Combined_TTC/Data link.

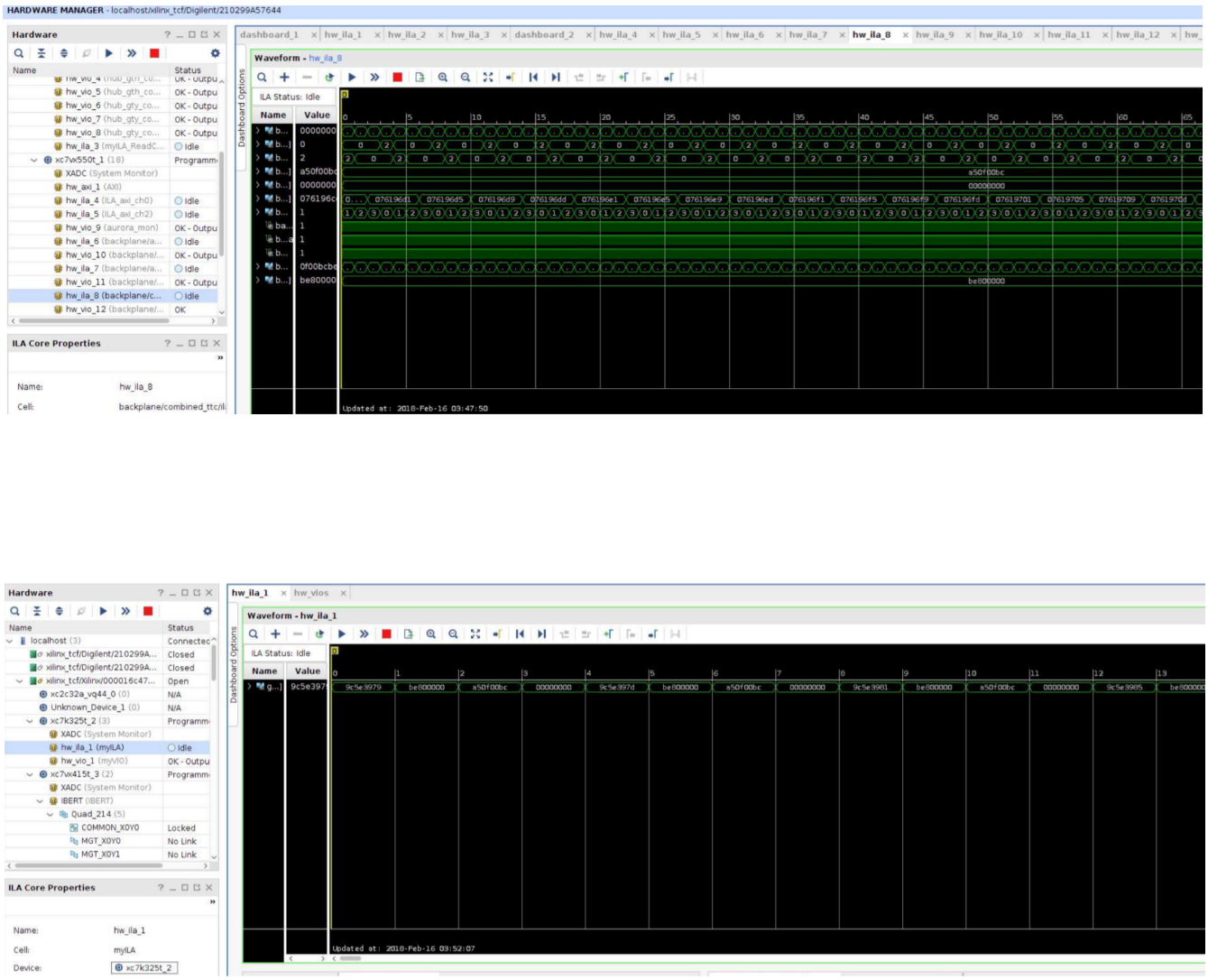


Figure 15. The Combined_TTC Data link test: A correct Combined_TTC stream on the ROD (upper plot) and on the FTM module in slot 3 (lower plot).

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	Link_up(0)
5	1		5	L1ID		5	control channel		5	Link_up(1)
6	0		6	L1ID		6	control channel		6	Link_up(2)
7	1		7	L1ID		7	control channel		7	Link_up(3)
8	version(0)		8	L1ID		8	control channel		8	Link Enable(0)
9	version(1)		9	L1ID		9	control channel		9	Link Enable(1)
10	version(2)		10	L1ID		10	control channel		10	Link Enable(2)
11	version(3)		11	L1ID		11	control channel		11	Link Enable(3)
12			12	L1ID		12	control channel		12	ROD XOFF
13			13	L1ID		13	control channel		13	0 (ROD reserved)
14			14	L1ID		14	control channel		14	0 (ROD reserved)
15	reserved		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	shelf(0)
20	felix_backpressure(0)		20	L1ID		20	control channel		20	shelf(1)
21	felix_backpressure(1)		21	L1ID		21	control channel		21	shelf (2)
22	felix_backpressure(2)		22	L1ID		22	control channel		22	shelf(3)
23	felix_backpressure(3)		23	L1ID(23)		23	control channel		23	CRC (9-bit)
24	felix_backpressure(4)		24	ECRID(0)		24	control channel		24	CRC (9-bit)
25	felix_backpressure(5)		25	ECRID(1)		25	control channel		25	CRC (9-bit)
26	felix_backpressure(6)		26	ECRID(2)		26	control channel		26	CRC (9-bit)
27	felix_backpressure(7)		27	ECRID(3)		27	control channel		27	CRC (9-bit)
28	felix_backpressure(8)		28	ECRID(4)		28	control channel		28	CRC (9-bit)
29	felix_backpressure(9)		29	ECRID(5)		29	control channel		29	CRC (9-bit)
30	felix_backpressure(10)		30	ECRID(6)		30	control channel		30	CRC (9-bit)
31	felix_backpressure(11)		31	ECRID(7)		31	control channel		31	CRC (9-bit)

Table 4. Combined TTC/Data bit definitions.

5.2 Control 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.

Level 1 Accept (L1A)

L1 Accept is used to indicate when an event has been accepted by the Central Trigger Processor.
The L1ID will be valid during the same broadcast cycle where L1A is active.

L1A is only active for a single broadcast cycle, and so there is the possibility of a Node or ROD missing it in the case of a crc error. If a node module drops an L1A, it would likely fail to produce data, headers, and trailers for that event. This will result in a timeout error occurrence in the ROD.

If the ROD misses the L1A, it will receive unexpected readout packets from any Nodes which have successfully received the L1A. The ROD will place these unexpected packets into a debug stream.

Bunch Counter Reset (BCR)

Entities within the shelf may keep local bunch counters. These are kept in sync using the BCR.
The Bunch Counter is a local 12-bit counter that increments each LHC clock. It counts to a value of 3563 and then rolls over to 0.
The first bunch after that in which BCR is asserted is bunch 0.
BCR is only active for a single broadcast cycle, and so there is the possibility of missing it in the case of a crc error. However, it is possible to design the counter in such a way as to provide further protection. The counter should be reset on (value = 3563) OR (BCR ='1'). If a BCR is missed, the counter should still remain in sync.

Event Counter Reset (ECR)

Entities within the shelf may keep local event counters. These are kept in sync using the ECR.
The Event counter is a local 24-bit counter that increments each L1A. It is reset to '-1' on ECR. The first event after ECR is number 0.
ECR is only active for a single broadcast cycle, and so there is the possibility of missing it in the case of a crc error. However, given that the ECRID is broadcast directly, the needs for a local counter is diminished.

Privileged Readout

This bit indicates that a full data readout has been requested. On each 40 MHz cycle, the delayed L1A, ECR, and BCR are read from the pipeline. If the L1A bit is set, the Privileged Readout FIFO is inspected. If the FIFO is not empty (this is the expected condition), the Privileged Readout 0 or 1 value is read and asserted on the backplane at the same time as the L1A, ECR, and BCR.

Felix Backpressure

These bits are used to convey the backpressure status as passed to the Hub from the Felix Backpressure GBT link. The 12 bits are each carried on individual eLinks within the GBT. The Hub receives the GBT link and inserts the individual backpressure bits into the Combined_TTC stream.

These bits are only used by the ROD. They are used to stop ROD data on any of the (up to) 12 individual Full Mode links going to Felix. These bits should remain asserted for the duration of the time that Felix is asserting backpressure.

L1ID

The L1ID should match the value in a local Event Counter. However, it is broadcast explicitly over this link to enhance channel reliability.
If a CRC error is detected in a message, then all four words must be discarded. In this case, the L1A may be completely missed, thus putting the local entity out of sync with the rest of the system. The L1ID however, may be correctly read in the next message that doesn't contain a crc error.
Other bits within the four-word broadcast will be held active for at least two four-word broadcast periods in order to reduce the probability of missing an active signal due to CRC error.

ECRID

These bits are an extension of the L1ID. In the past, this has been generated on each board independently. Now the L1ID and ECRID bits are combined into a single 32-bit counter and broadcast from the Hub to all other boards.

Control Channel

These bits are reserved for a possible future implementation of a message channel

Link Reset

As the Aurora Receiver, the ROD must control all of the Aurora Data Link initialisation. The Link Reset signal is therefore derived from the Readout_Ctrl link. The Hub should connect to the appropriate Readout_Ctrl Link Reset for the slot to which the Combined_TTC is connected. Each endpoint also has a local reset timer to control Aurora link establishment.
There are four Link Reset bits to allow each of the four jFEX Aurora links (per Hub) to be reset individually. The eFEX should listen to Link Reset 0. Below is an example of routing from Hub to a Slot-3 Fex

ROD XOFF

When active, this signal indicates that the ROD cannot currently accept further data from Fex sources. This signal is taken from the Readout_Ctrl and is fanned-out to the shelf FEX's over each slot's Combined_TTC link.

Link_Up (3:0)

These signals indicate to each FEX whether or not the corresponding ROD Rx Aurora channel is "Up". The eFEX should use only Link_Up(0)

ROD Reserved

These bits are reserved for future ROD functionality to be defined

Shelf Number

The Hub provides the shelf number to all Combined_TTC targets via these bits.

CRC (9-bit)

CRC is included to provide additional robustness on this link. Erroneous resets could cause loss of data. Choice of polynomial is from https://users.ece.cmu.edu/~koopman/crc/index.html

The chosen polynomial is 0x17d in Koopman format of 0x2fb in explicit+1 format.

The CRC is calculated over all four words of each transmission. When a CRC error occurs, all four of the associated words must therefore be ignored. Because of this, local counters for L1ID and ECRID should not be used. If a reset is missed due to a crc error, then these values would not be updated. L1ID and ECRID should be taken directly from the Combined_TTC link.

5.3 CombinedTTC/Data latency definition

The primary concern for the FEX modules is providing deterministic latency of the data at the receiver end. The user description of the firmware is as follows:

The 128 bits that make a complete frame of TTC / Readout Control Combined Data are provided to the users of this data at the output of a 128 bit wide D Register. This D Register is updated on the positive edge of the FPGA's 40.08 MHz clock that is locked to the 40.08 MHz LHC Backplane Reference Clock. These 128 bits are stable at all times except during the update.

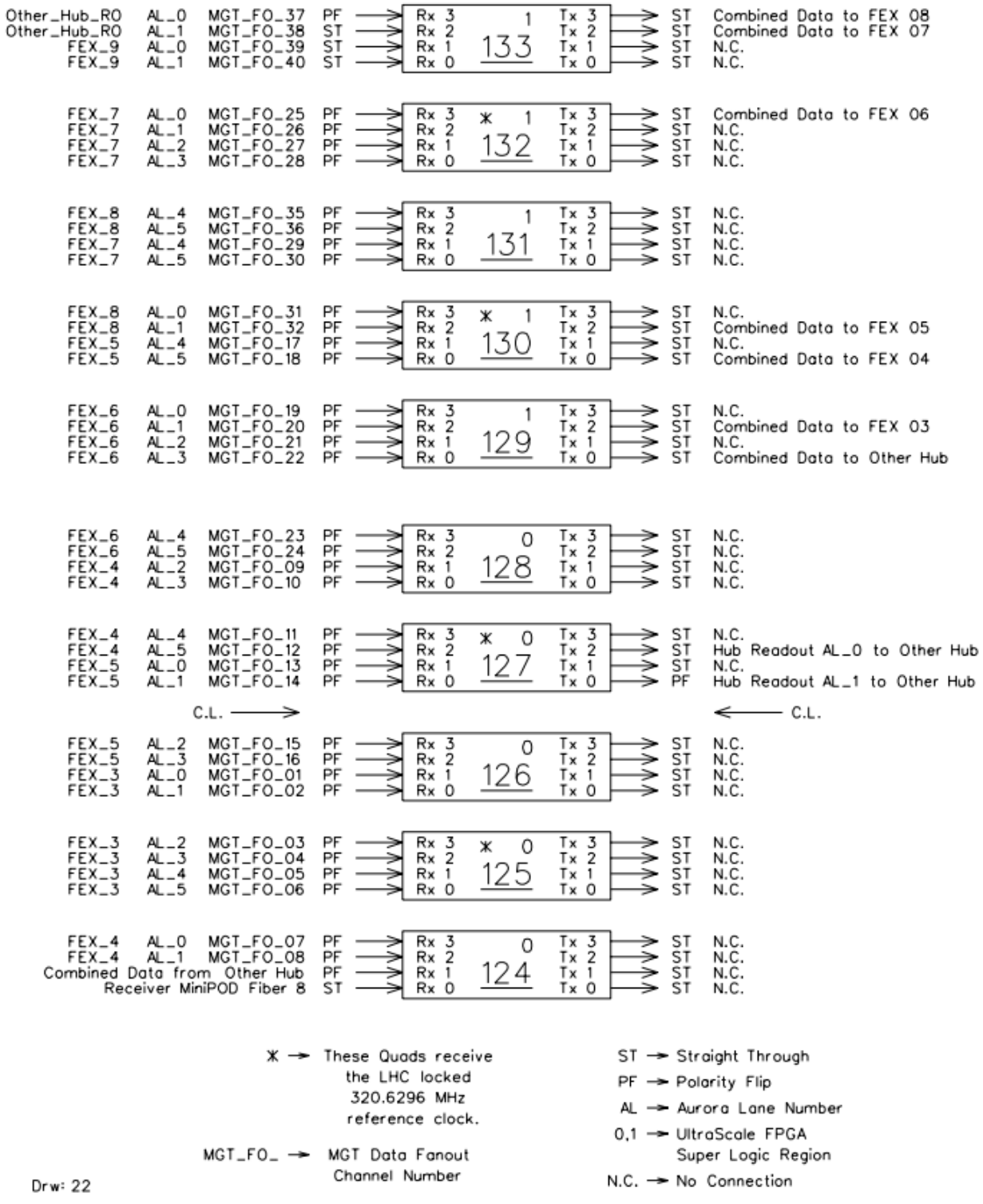


Figure 16. HUB MGT assignments (GTY Transceivers – QUADs 124:133)

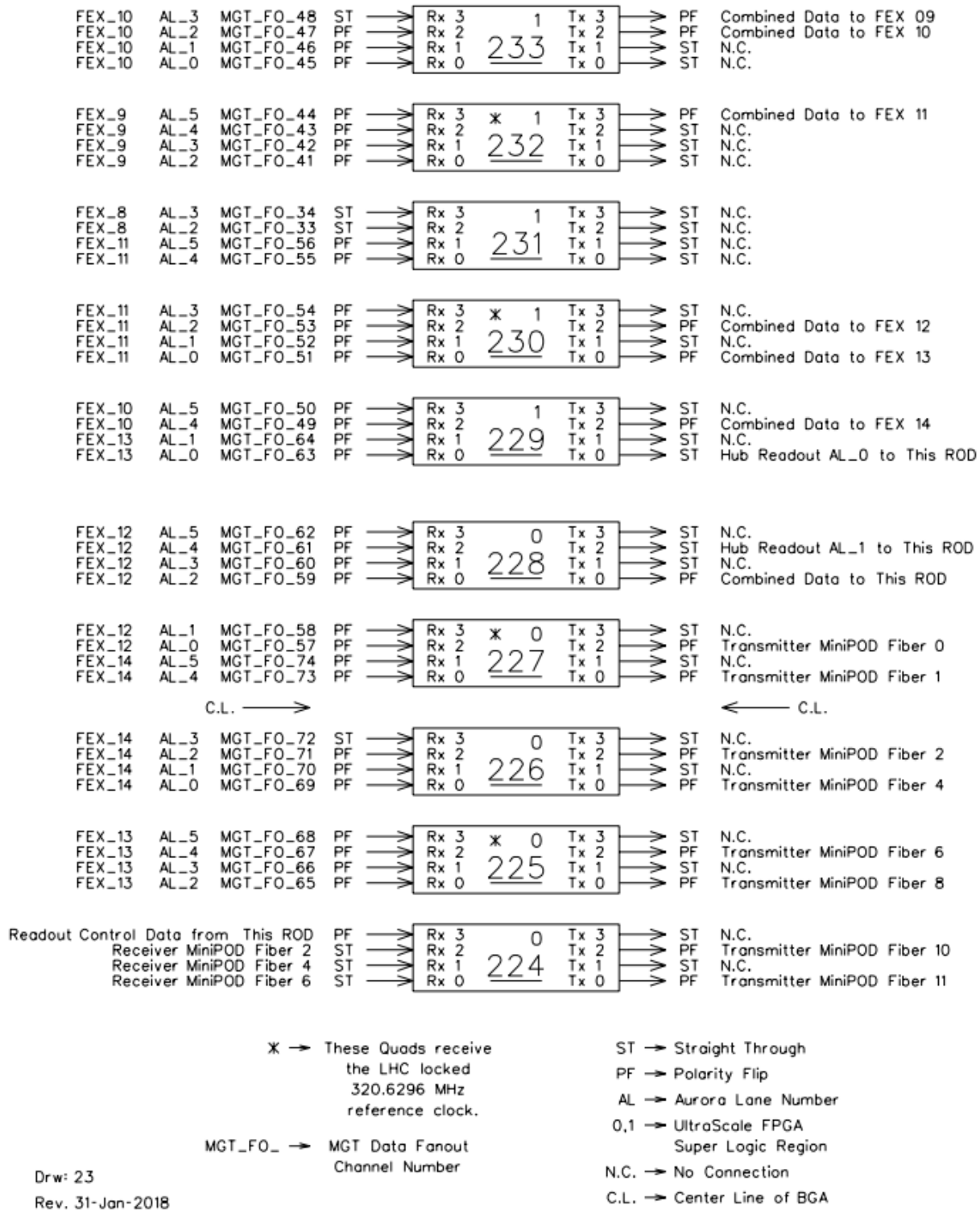
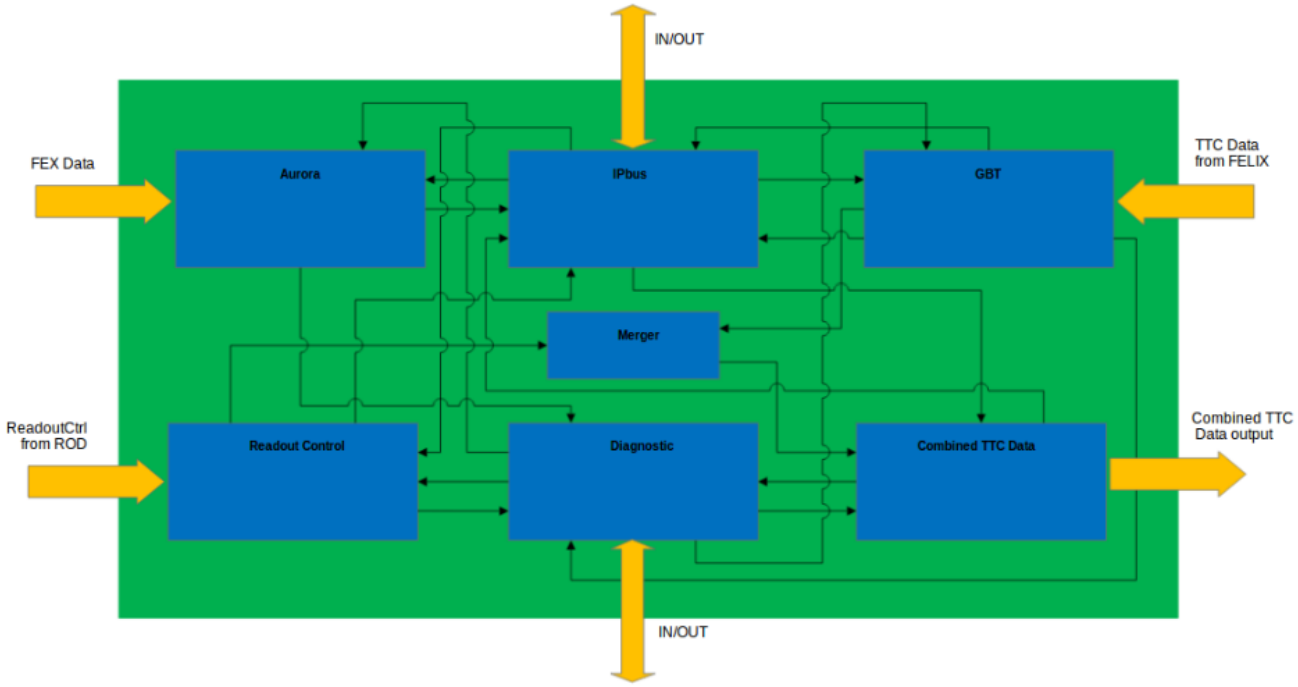


Figure 17. HUB MGT assignments (GTH Transceivers – QUADs 224:233)

HUB Firmware Diagram:



HUB FW development split into several separate stages.

1. IPbus
2. Readout Control @6.4Gbps
3. Combined TTC/Data @6.4Gbps
4. GBT @4.8Gbps
5. Merger
6. Diagnostic
6. Aurora8b/10b @6.4Gbps

Type:	Utilization	Available	Utilization %
LUT	18543	716160	2.58
LUTRAM	1082	154560	0.70
FF	153260	1432320	10.70
BRAM	48	2520	1.90
GT	4	80	5.00
BUFG	39	1200	3.25

Resource usage for HUB FW: Ipbus+ReadoutCtrl+CombinedTTC/Data (2 links only)

Specification for the HUB Safe Configuration

Introduction

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.

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 [4]. The ILA core includes many advanced features of modern logic analyzers, including Boolean trigger equations, and edge transition triggers. Because the ILA core is synchronous to the design being monitored, all design clock constraints that are applied to your design are also applied to the components inside the ILA core.

Virtual Input/Output (VIO)

The LogiCORE™ IP Virtual Input/Output (VIO) core is a customizable core that can both monitor and drive internal FPGA signals in real time [5]. The number and width of the input and output ports are customizable in size to interface with the FPGA design. Because the VIO core is synchronous to the design being monitored and/or driven, all design clock constraints that are applied to your design are also applied to the components inside the VIO core. Run time interaction with this core requires the use of the Vivado® logic analyzer feature.

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) [7]. The system monitors can be placed in the bottom SLR0 (SYSMONE1_X0Y0) and then consecutively in the upper SLR increasing Y locations (SYSMONE1_X0Y1). Monitoring across SLR boundaries is not possible. Temperature, VCCINT, VCCAUX, VCCBRAM measurements are specific to an individual SLR. For the UltraScale FPGAs

SYSMONE1, the System Management Wizard provides I2C functionality to the slave SLRs using the DRP port and additional logic. The SYSMON has numerous operating modes that are user-defined by writing to the control registers, which can be accessed through DRP, JTAG or I2C. It is also possible to initialize these register contents when the SYSMON is instantiated in a design using the block attributes.

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 IOSTANDARD LVDS DIFF_TERM_ADV TERM_100 DQS_BIAS TRUE EQUALIZATION EQ_LEVEL0	IBUFGDS	IN
3	Ref_4008_MHz_from_FPGA_to_Rec_Dir Ref_4008_MHz_from_FPGA_to_Rec_Cmp	PACKAGE_PIN AV31 PACKAGE_PIN AW31	Controling only, an VIO component added. A default value set to 0.	OUT

		IOSTANDARD LVDS		
4	PLL_4008_Mhz_Lock_Detect_to_FPGA	PACKAGE_PIN B27 IOSTANDARD LVCMOS18	Monitoring only, an ILA component added.	IN
5	PLL_32064_MHz_Lock_Detect_to_FPGA	PACKAGE_PIN B26 IOSTANDARD LVCMOS18	Monitoring only, an ILA component added.	IN
6	Select_Input_Second_40_Fanout	PACKAGE_PIN A26 IOSTANDARD LVCMOS18	Controling only, an VIO component added. A default value set to 0.	OUT
7	FPGA_SW_A_ATC_LOOP_DET FPGA_SW_B_ATC_LOOP_DET FPGA_SW_C_ATC_LOOP_DET	PACKAGE_PIN AV13 PACKAGE_PIN AT14 PACKAGE_PIN AV15 IOSTANDARD LVCMOS33 SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value set to 0.	OUT
8	FPGA_SW_A_LOOP_DETECTED FPGA_SW_B_LOOP_DETECTED FPGA_SW_C_LOOP_DETECTED	PACKAGE_PIN AY15 PACKAGE_PIN AU13 PACKAGE_PIN AV16 IOSTANDARD LVCMOS33	Monitoring only, an ILA component added.	IN
9	FPGA_SW_A_MDC FPGA_SW_B_MDC FPGA_SW_C_MDC	PACKAGE_PIN AW13 PACKAGE_PIN AT16 PACKAGE_PIN AY13 IOSTANDARD LVCMOS33 SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value set to 0.	OUT
10	FPGA_SW_A_MDIO FPGA_SW_B_MDIO FPGA_SW_C_MDIO	PACKAGE_PIN AW15 PACKAGE_PIN AU16 PACKAGE_PIN AY14 IOSTANDARD LVCMOS33 PULLUP TRUE	Monitoring only, an ILA component added.	IN
11	ISO_SLOT_HW_ADRS_0 ISO_SLOT_HW_ADRS_1	PACKAGE_PIN AT12 PACKAGE_PIN AT11	Monitoring only, an ILA component added.	IN

	ISO_SLOT_HW_ADRS_2 ISO_SLOT_HW_ADRS_3 ISO_SLOT_HW_ADRS_4 ISO_SLOT_HW_ADRS_5 ISO_SLOT_HW_ADRS_6 ISO_SLOT_HW_ADRS_7	PACKAGE_PIN AU12 PACKAGE_PIN AU11 PACKAGE_PIN AV11 PACKAGE_PIN AW12 PACKAGE_PIN AW11 PACKAGE_PIN AY12 IOSTANDARD LVCMOS33		
12	SHELF_ADRS_0_TO_FPGA SHELF_ADRS_1_TO_FPGA SHELF_ADRS_2_TO_FPGA SHELF_ADRS_3_TO_FPGA SHELF_ADRS_4_TO_FPGA SHELF_ADRS_5_TO_FPGA SHELF_ADRS_6_TO_FPGA SHELF_ADRS_7_TO_FPGA	PACKAGE_PIN BB15 PACKAGE_PIN BB14 PACKAGE_PIN BA14 PACKAGE_PIN BB13 PACKAGE_PIN BB12 PACKAGE_PIN BB11 PACKAGE_PIN BA12 PACKAGE_PIN BA11 IOSTANDARD LVCMOS33	Monitoring only, an ILA component added.	IN
13	OVERALL_ADRS_0_TO_RES_NET OVERALL_ADRS_1_TO_RES_NET OVERALL_ADRS_2_TO_RES_NET OVERALL_ADRS_3_TO_RES_NET OVERALL_ADRS_4_TO_RES_NET OVERALL_ADRS_5_TO_RES_NET OVERALL_ADRS_6_TO_RES_NET OVERALL_ADRS_7_TO_RES_NET	PACKAGE_PIN BF25 PACKAGE_PIN BE25 PACKAGE_PIN BF26 PACKAGE_PIN BE27 PACKAGE_PIN BF27 PACKAGE_PIN BE28 PACKAGE_PIN BE29 PACKAGE_PIN BE30 IOSTANDARD LVCMOS18 SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value set to 0.	OUT
14	HUB_I2C_TO_FPGA_SCL_0 HUB_I2C_TO_FPGA_SDA_0 Hub_I2C_TO_FPGA_SCL_1 Hub_I2C_TO_FPGA_SDA_1	PACKAGE_PIN BE16 PACKAGE_PIN BF16 PACKAGE_PIN BA29 PACKAGE_PIN BB29 IOSTANDARD LVCMOS18	No monitoring.	IN
15	I2C_Buf_1501_ENABLE I2C_Buf_1502_ENABLE I2C_Buf_1503_ENABLE	PACKAGE_PIN BA16 PACKAGE_PIN BA15 PACKAGE_PIN BB16 IOSTANDARD LVCMOS33 SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value set to 1 for I2C_Buf_1501_ENABLE, others set to 0.	OUT
16	Recvr_MiniPOD_INTR_B Trans_MiniPOD_INTR_B	PACKAGE_PIN AR15 PACKAGE_PIN AN16 IOSTANDARD LVCMOS33	Monitoring only, an ILA component added.	IN

17	Recvr_MiniPOD_RESET_B Trans_MiniPOD_RESET_B	PACKAGE_PIN AR14 PACKAGE_PIN AP13 IOSTANDARD LVCMOS33 SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value set to 1.	OUT
18	Recvr_MiniPOD_SCL Trans_MiniPOD_SCL	PACKAGE_PIN AR12 PACKAGE_PIN AN15 IOSTANDARD LVCMOS33 SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value set to 0.	OUT
19	Recvr_MiniPOD_SDA Trans_MiniPOD_SDA	PACKAGE_PIN AR13 PACKAGE_PIN AP15 IOSTANDARD LVCMOS33	Monitoring only, an ILA component added.	IN
20	ACCESS_SIGNAL_1_FROM_FPGA ACCESS_SIGNAL_2_FROM_FPGA	PACKAGE_PIN AT30 PACKAGE_PIN AT31 IOSTANDARD LVCMOS18 SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value set to 0.	OUT
21	HUB_FPGA_LED50_DRV HUB_FPGA_LED51_DRV HUB_FPGA_LED52_DRV	PACKAGE_PIN BF29 PACKAGE_PIN BF30 PACKAGE_PIN BF31 IOSTANDARD LVCMOS18 SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value set to 0.	OUT
22	Hubs_SMB_Alert_B	PACKAGE_PIN AP16 IOSTANDARD LVCMOS33	Monitoring only, an ILA component added.	IN
23	ALL_HUB_POWER_GOOD_TO_FPGA	PACKAGE_PIN AM16 IOSTANDARD LVCMOS33	Monitoring only, an ILA component added.	IN
24	MGT_FO_EQU_ENB_GRP_1 MGT_FO_EQU_ENB_GRP_2 MGT_FO_EQU_ENB_GRP_3 MGT_FO_EQU_ENB_GRP_4 MGT_FO_EQU_ENB_GRP_5 MGT_FO_EQU_ENB_GRP_6 MGT_FO_EQU_ENB_GRP_7 MGT_FO_EQU_ENB_GRP_8 MGT_FO_EQU_ENB_GRP_9 MGT_FO_EQU_ENB_GRP_10 MGT_FO_EQU_ENB_GRP_11	PACKAGE_PIN C25 PACKAGE_PIN A25 PACKAGE_PIN B25 PACKAGE_PIN A24 PACKAGE_PIN C24 PACKAGE_PIN B22 PACKAGE_PIN D20 PACKAGE_PIN C20 PACKAGE_PIN A23 PACKAGE_PIN A21 PACKAGE_PIN A20	Controling only, an VIO component added. A default value set to 0.	OUT

	MGT_FO_EQU_ENB_GRP_12 MGT_FO_EQU_ENB_GRP_13	PACKAGE_PIN A19 PACKAGE_PIN A18 IOSTANDARD LVCMOS18 SLEW SLOW DRIVE 8		
25	TBD_SPARE_LINK_0_Dir TBD_SPARE_LINK_0_Cmp TBD_SPARE_LINK_1_Dir TBD_SPARE_LINK_1_Cmp TBD_SPARE_LINK_2_Dir TBD_SPARE_LINK_2_Cmp TBD_SPARE_LINK_3_Dir TBD_SPARE_LINK_3_Cmp	PACKAGE_PIN AV26 PACKAGE_PIN AW26 PACKAGE_PIN AT27 PACKAGE_PIN AU27 PACKAGE_PIN AY27 PACKAGE_PIN AY28 PACKAGE_PIN AT29 PACKAGE_PIN AU29 IOSTANDARD LVCMOS18SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value set to 0.	OUT
26	ROD_PRESENT_B_TO_FPGA	PACKAGE_PIN AW27 IOSTANDARD LVCMOS18	Monitoring only, an ILA component added.	IN
27	ROD_Power_Control_2_FPGA ROD_Power_Control_3_FPGA ROD_Power_Control_4_FPGA	PACKAGE_PIN AW28 PACKAGE_PIN AV30 PACKAGE_PIN AV29 IOSTANDARD LVCMOS18	Monitoring only, an ILA component added.	IN
28	ROD_Power_Enable ROD_Power_Enable_B	PACKAGE_PIN AR17 PACKAGE_PIN AP17 IOSTANDARD LVCMOS33 SLEW SLOW DRIVE 4	Controling only, an VIO component added. A default value: ROD_Power_Enable set 0 ROD_Power_Enable_B set 1	OUT
29	FPGA_RODs_SMBALERT_B	PACKAGE_PIN BB28 IOSTANDARD LVCMOS18	Monitoring only, an ILA component added.	IN
30	Phys_U21_TXD0 Phys_U21_TXD1 Phys_U21_TXD2 Phys_U21_TXD3 Phys_U21_TX_EN Phys_U21_GTX_CLK Phys_U22_TXD0 Phys_U22_TXD1 Phys_U22_TXD2 Phys_U22_TXD3 Phys_U22_TX_EN Phys_U22_GTX_CLK	PACKAGE_PIN BA36 PACKAGE_PIN AY35 PACKAGE_PIN BB36 PACKAGE_PIN BA35 PACKAGE_PIN BB34 PACKAGE_PIN BA34 PACKAGE_PIN AR36 PACKAGE_PIN AT36 PACKAGE_PIN AR35 PACKAGE_PIN AT35 PACKAGE_PIN AU34 PACKAGE_PIN AT34 IOSTANDARD LVCMOS18	Controling only, an VIO component added. A default value set to 0.	OUT
31	Phys_U21_CLK125_LED_MODE Phys_U21_RXD0_MODE0	PACKAGE_PIN AU33 PACKAGE_PIN BC31	Monitoring only, an ILA component added.	IN

	Phys_U21_RXD1_MODE1 Phys_U21_RXD2_MODE2 Phys_U21_RXD3_MODE3 Phys_U21_RX_DV_CLK125_EN Phys_U21_RX_CLK_PHYAD2 Phys_U22_CLK125_LED_MODE Phys_U22_RXD0_MODE0 Phys_U22_RXD1_MODE1 Phys_U22_RXD2_MODE2 Phys_U22_RXD3_MODE3 Phys_U22_RX_DV_CLK125_EN Phys_U22_RX_CLK_PHYAD2	PACKAGE_PIN BA32 PACKAGE_PIN BB33 PACKAGE_PIN AY33 PACKAGE_PIN BB31 PACKAGE_PIN AV33 PACKAGE_PIN AY34 PACKAGE_PIN AV35 PACKAGE_PIN AV36 PACKAGE_PIN AV34 PACKAGE_PIN AU32 PACKAGE_PIN AW32 PACKAGE_PIN AW33 IOSTANDARD LVCMOS18		
32	Spare_OSC_TO_FPGA_Dir Spare_OSC_TO_FPGA_Cmp	PACKAGE PIN AU28 PACKAGE PIN AV28 IOSTANDARD LVDS DQS_BIAS TRUE	IBUFGDS	IN

Table: "Hub Safe Configuration" signal list, including the location and I/O standard, components which are being used to receive these signals and directionality.

Hub FPGA Signal Types:

Clock and Clock Management Signals:

Logic_Clk_320.64_MHz_to_FPGA_Dir/Cmp
Logic_Clk_40.08_MHz_to_FPGA_Dir/Cmp

These are the main Logic Clock signals for the Hub FPGA. These are AC coupled LHC locked clocks. The FPGA must receive them as differential signals, 100 Ohm terminate them, and DC Bias these FPGA inputs by including the proper combination of DQS_Bias and EQ_Level0 attributes in the setup of these I/O Blocks. Even if these clocks are not used in a minimal safe FPGA design they must still be properly received.

Ref_40.08_MHz_from_Other_Hub_ Dir/Cmp

This is an AC coupled LVDS input to the Hub FPGA. In a shelf with two Hub Modules this clock path is used on the Hub that does not receive an Optical Felix TTC signal. Because this is AC coupled the I/O Block that receives this signal must be setup with the correct combination of the DQS_Bias and EQ_Level0 attributes. This FPGA input should include a 100 Ohm differential terminator. This input needs to be setup and instantiated in all versions of the Hub FPGA firmware.

Ref_40.08_MHz_from_FPGA_to_Rec_ Dir/Cmp

This is an LVDS output from the Hub FPGA that runs to an LVDS receiver and then connects to the reference input on the 40.08 MHz PLL. I believe that we should include this output even in a minimal safe FPGA design even ifvwe only tie the input to the LVDS driver Low.

PLL_40.08_Mhz_Lock_Detect_to_FPGA
PLL_320.64_MHz_Lock_Detect_to_FPGA

These are two Lock Detect signals from the PLL Clock circuits on the Hub Module. When Hi they indicate that the associated PLL has locked onto its reference input. The FPGA needs to always provide 1.8V CMOS receivers for these two signals.

Select_Input_Second_40_Fanout

This signal is an output from the Hub FPGA. It is used to enable or disable the Hub circuit board from sending out 40.08 MHz Reference Clocks on Zone-2 of the Backplane. Recall that in a shelf with 2 Hub Modules, that only the primary Hub that receives the Optical Felix TTC signal will send out its 40.08 MHz Reference Clocks to the FEX modules. The secondary Hub (the Hub that does not receive an Optical Felix TTC signal) must not send out its Zone-2 Reference Clocks. This is a 1.8V CMOS open-drain output from the Hub Module. When Low it enables the Hub Module to send out its Zone-2 Reference Clocks. When the open-drain output transistor in the FPGA I/O Block is Off then its backplane Reference Clock output drives will be shutdown. In a minimal safe design we want this open-drain output to pull this signal Low so that we can see these Reference Clock outputs.

SPARE_OSC_TO_FPGA_Dir/Cmp

This is a spare differential clock input to the Hub FPGA. As long as we are not installing this part (U562) then nothing special needs to be done with these input pins to the FPGA even in a minimal safe FW design.

Ethernet Switch Chip Management Signals:

Each of the 3 Broadcom Ethernet Switch chips has its own set of 4 management signals with the Hub FPGA. See Hub Circuit Diagram 17.

FPGA_SW_X_ATC_LOOP_DET

This is a control signal from the Hub FPGA to the Ethernet Switch chips. I assume that this signal will eventually be controlled from a bit in an IPBus visible register. A minimal Hub FPGA design could just tie this signal Low. A Slow Slew and 4 mA Drive level is fine for this 3.3V CMOS static level output signal.

FPGA_SW_X_LOOP_DETECTED

This is a status signal from the Broadcom Switch chips to the Hub FPGA. I assume that this status bit will eventually be visible in an IPBus register. The FPGA needs to at least always provide a receiver for this 3.3V CMOS input signal.

FPGA_SW_X_MDC

Because the MDC/MDIO interface on the Broadcom Switch chips is in Slave mode, this clock signal runs from the Hub FPGA to the Switch chip. Eventually I assume that these pins on the Hub FPGA will be controlled by firmware that implements the MDC/MDIO part of a MAC. For minimal safe firmware this line could just be driven Low by a Slow Slew 4 mA 3.3V CMOS output driver.

FPGA_SW_X_MDIO

This is a bi-directional data line between the Hub FPGA and the MDC/MDIO interface in the Broadcom Switch chips. Eventually I assume that these pins on the Hub FPGA will be handled by firmware that implements the MDC/MDIO part of a MAC. In a minimal safe Hub FPGA design I would provide a 3.3V CMOS receiver for this signal with a weak pull-up resistor at its input.

Hardware Address Signals:

ISO_SLOT_HW_ADRS_X

These 8 signals bring the backplane Hardware Slot Address to the Hub FPGA. The IPMC module also receives these signals. Even if the Hub FPGA is not going to use this Hardware Slot number information it must still receive these 8 lines with 3.3V CMOS receivers.

SHELF_ADRS_X_TO_FPGA

These are 8 signals that bring the Shelf Address information from the IPMC to the Hub's FPGA. These lines connect to "User I/O" pins on one of the ARM CPUs in the IPMC module. The Hub Firmware needs to at least always provide 3.3V CMOS receives for these signals.

OVERALL_ADRS_X_TO_RES_NET

These are 8 outputs from the Hub FPGA that provide an Overall Hardware Address to the ROD mezzanine. This Overall Hardware Address is made up of the Slot Hardware Address that comes from the Backplane and the Shelf Hardware Address that comes from the IPMC. In a minimal safe FPGA implementation I would just tie all 8 of these Overall Hardware Address lines Low. These can be 1.8V CMOS Slow Slew 4 mA Drive outputs.

I2C Bus Signals:

Hub_I2C_to_FPGA_SCL
Hub_I2C_to_FPGA_SDA

Note that this I2C bus makes TWO connections to the Hub FPGA. It connects to pins:

BE16 IO_L23P_T3U_N8_I2C_SCLK_65
BF16 IO_L23N_T3U_N9_I2C_SDA_65

that provide for a Slave I2C interface to the FPGA System Monitor. It connects to pins:

BA29 IO_L18P_T2U_N10_AD2P_67
BB29 IO_L18N_T2U_N11_AD2N_67

so that one can implement a Master I2C interface so that for example the Hub FPGA can talk to the Hub's DCDC Converter power supplies. In a minimal safe design both of these I2C signals at both pin pairs should connect to 1.8V CMOS Receivers.

I2C_Buf_150X_ENABLE

These are control signals from the Hub FPGA to the Sensor I2C Bus translator/buffer chips. These translator/buffer chips allow the overall Sensor I2C Bus on the ROD-Hub cards to be divided up into sections. Eventually we may need to control dividing up the Sensor I2C bus either from bits in an IPBus visible register of from a control signal from the ROD to the Hub. For a minimal safe Hub FPGA design these 3 I2C buffer control signals can just be tied Hi which will enable these I2C translator/buffers. These control signals are 3.3V Slow Slew 4 mA Drive outputs from the Hub FPGA.

MiniPOD Management Signals:

Each of the MiniPODs has 4 management signals associated with it. The Hub's Receiver and Transmitter MiniPODs each have their own private set of these 4 management signals. The setup of the MiniPOD management signals on the Hub is very similar to that used on the CMX card.

Recvr_MiniPOD_INTR_B
Trans_MiniPOD_INTR_B

Interrupt signals from the MiniPODs to the FPGA. Low indicates interrupt. The FPGA should always provide 3.3V CMOS receivers for these two signals.

Recvr_MiniPOD_RESET_B
Trans_MiniPOD_RESET_B

These are Reset signals from the FPGA to the MiniPODs. Low indicates Reset. A minimal safe design should at least drive these lines Hi with a Slow Slew 4 mA Drive 3.3V CMOS output. In a full firmware design I assume that these Reset signals will be controlled from bits in an IPBus visible register.

Recvr_MiniPOD_SCL
Trans_MiniPOD_SCL

These are the clock signals for the bi-directional serial data path between the FPGA and the MiniPODs. In a minimal safe design I would drive this signals Low with a Slow Slew 4 mA Drive 3.3V CMOS output.

Recvr_MiniPOD_SDA
Trans_MiniPOD_SDA

These are the bi-directional data lines for the serial data path between the FPGA and the MiniPODs. In a minimal design I would provide a 3.3V CMOS receiver for these signals.

Miscellaneous Select I/O Signals:

ACCESS_SIGNAL_X_FROM_FPGA

These two output signals from the FPGA run to a translator/buffer chip and then to pins in the front panel J2 connector. See Hub Circuit Diagram 53. The purpose of these signals is to allow one to see with an oscilloscope some aspect of the FPGA's operation. Unless needed for something special these 1.8V CMOS output signals can be configured for a Slow Slew rate and modest 4 mA Drive. A minimal safe Hub FPGA design could just tie these 2 signals Low.

HUB_FPGA_LED5X_DRV

These three signals are outputs from the Hub FPGA that control front panel LEDs. These are 1.8V CMOS signals and using a Slow Slew and 4 mA of Drive would be fine.
A minimal safe Hub FPGA design could just tie these 3 output signals Low.

Hubs_SMB_Alert_B

This is an input signal to the Hub's FPGA from the 7 DCDC Converter power supplies on the Hub Module. When Low it indicates that one or more of these Hub power supplies is in trouble (or at least wants attention). Normally this signal should be Hi. This is a 3.3V CMOS signal and the FPGA must always have a receiver for it even if it does not make use of this signal.

ALL_HUB_POWER_GOOD_TO_FPGA

This is an input signal to the Hub's FPGA from the Power Control circuits on the Hub Module. This is a 3.3V Select I/O signal in Bank 84. When Low this signal indicates that there is some kind of power supply problem on the Hub Module. When Hi this signal indicates: that all 7 DCDC Converters report that they are operating correctly, and that the Isolated +12V supply is Enabled, and that the output from the Linear MGT_AVAUX and BULK_2V5 supplies is good. The Hub FPGA must always instance a 3.3V CMOS receiver for this signal even if it does not make use of this signal. This signal comes through a 2.7k Ohm isolation resistor from a 5V source.

MGT_FO_EQU_ENB_GRP_XX

These 13 control signals are outputs from the Hub FPGA that enable or disable the equalization in the MGT Fanout chips. All fanout chips that service a given data source either have their equalization enabled or disabled. I assume that eventually these 13 equalization enable signals will be separately controllable from bits in an IPBus visible register. In a minimal safe Hub FPGA design all of the Equalization Enable signals can just be tied HI. I would drive them as 1.8V CMOS outputs with Slow Slew and 8 mA of Drive. The loading on these signals will only be about 150 uA but the high Drive level may help if we need to ramp up the FAN_1V8 rail voltage.

TBD_SPARE_LINK_X_Dir/Cmp

These are 8 To Be Determined spare lines between the Hub and ROD FPGAs. Currently none of them have an assigned function. If needed they can be operated as either 8 separate 1.8V CMOS signals or as 4 separate LVDS signals. As with all CMOS signals we can not just leave them floating. As unassigned spare signals the agreement with ROD is that we will run them as 8 separate CMOS signals and that the Hub will drive them and the ROD will receive them. Further the agreement with ROD is that the Hub will keep these signals tri-stated until the ROD asserts it Power Control #2 signal HI to the Hub. See Hub Circuit Diagram 42. Even in a minimal safe firmware design the Hub needs to implement this functionality. These 8 lines should be driven with Slow Slew 4 mA Drive 1.8V CMOS output blocks.

Phys_U21 and Phys_U22 signals:

Each Phys Chip makes 16 connections with the Hub's FPGA. In the real Hub operation these 16 signals will be managed by MAC IP firmware. This RGMII connection between the FPGA and the Phys Chips includes signals that may require fast slew rates, higher drive current, and DCI back (series) termination. These RGMII signals have been laid out carefully on the Hub circuit to minimize cross-talk, reflections, and signal loss. Careful consideration must be given to how the RMGII signals are handled in the FPGA I/O Blocks to provide both a good RGMII interface and to minimize interference with other parts of the Hub Module. For a minimal safe Hub FPGA design we still should handle these CMOS signals in a defined rational way. These signal should be driven with Slow Slew 1.8V CMOS outputs tied Low:

Phys_U2X_TXD0
Phys_U2X_TXD1
Phys_U2X_TXD2
Phys_U2X_TXD3
Phys_U2X_TX_EN
Phys_U2X_GTX_CLK
Phys_U2X_MDC

We should provide 1.8V CMOS receivers for the following signals:

Phys_U2X_RXD0__MODE0
Phys_U2X_RXD1__MODE1
Phys_U2X_RXD2__MODE2

Phys_U2X_RXD3__MODE3
Phys_U2X_RX_CLK__PHYAD2
Phys_U2X_RX_DV__CLK125_EN
Phys_U2X_CLK125__LED_MODE
Phys_U2X_MDIO
Phys_U2X_INT_B

ROD Present and ROD Power Control Signals:

These 7 signals have to do with letting the Hub know whether or not a ROD is installed on it, Controlling (i.e. enabling) the power on the ROD, and letting the Hub know whether or not all ROD power is good and that the ROD is ready for normal trigger system operation.

ROD_PRESENT_B_TO_FPGA

This signal is a pull-down resistor on the ROD. When the ROD is NOT present this signal goes HI on the Hub. The Hub must always provide a 1.8V CMOS receiver for
this signal even if it does not use it.

ROD_Power_Control_2_FPGA

This is an input to the Hub's FPGA. When HI this signal indicates that all power supplies on the ROD are operating correctly. The Hub must always provide a 1.8V CMOS receiver for this signal even if it does not use it.

ROD_Power_Control_3_FPGA

This is an input to the Hub's FPGA. When HI this signal indicates that the ROD is Configured and fully ready for normal L1Calo operation. The Hub must always provide a 1.8V CMOS receiver for this signal even if it does not use it.

ROD_Power_Control_4_FPGA

This is a spare power control signal. In a minimal safe firmware design the Hub should provide a 1.8V CMOS receiver for this signal with a weak pull-up resistor at its input.

ROD_Power_Enable
ROD_Power_Enable_B

These two signal are outputs from the Hub FPGA that through some hardwired logic on the Hub tell the ROD when it may turn ON its power supplies. These are 3.3V Slow Slew 4 mA Drive CMOS outputs. A minimal safe design (that just locks OFF the ROD power) should set ROD_Power_Enable Low and set ROD_Power_Enable_B Hi.

FPGA_RODs_SMBALERT_B

This is an input to the Hub's FPGA that when LOW indicates a power supply problem on the ROD. During normal operation this signal should always be HI. Pull up current to 1V8 is provided by a 1k Ohm resistor on page 25 of the ROD schematics. A normal 1.8V CMOS receiver should be used. The FPGA needs to always provide a receiver for this signal.

References:

1. Technical Specification, ALAS Level-1 Calorimeter Trigger Upgrade, FEX System Switch Module (FEX Hub) Prototype, Dan Edmunds,Yuri Ermoline, Brian Ferguson, Wade Fisher, Philippe Laurens, Spencer Lee, Pawel Plucinski; 2018

2. Specification for Readout Control & Combined TTC/Data Serial links in L1Calo, Ian Brawn, Ed Flaherty, Weiming Qian, David Sankey, Saeed Taghavi; 2018.

3. Aurora 8B/10B v11.0, LogiCORE IP Product Guide, Vivado Design Suite, PG046 October 5, 2016.

4. Virtual Input/Output v3.0, LogiCORE IP Product Guide, Vivado Design Suite, PG159 October 4, 2017.

5. Integrated Logic Analyzer v6.1 , LogiCORE IP Product Guide, Vivado Design Suite, PG172 April 6, 2016.

6. Vivado Design Suite User Guide, Getting Started, UG910 (v2017.1) April 5, 2017

7. UltraScale Architecture System Monitor, User Guide, UG580 (v1.8) June 15, 2017.

8. The GBT-FPGA core: features and challenges, M. Barros Marin et al 2015 JINST 10 C03021

9. GBT-FPGA Tutorial, FPGA Working group – TWEPP 2016