Wupper - a PCIe DMA Engine for FELIX

Frans Schreuder, Andrea Borga, Jos Vermeulen, Oussama el Kharraz Alami

May 16, 2025

Contents

Revision History

Revision	Date	Author(s)	Description
5.0	19-01-2021	F. P. Schreuder	Updated documentation to be compatible with phase2 / rm5.0
4.8	11-02-2020	F. P. Schreuder	Regenerated wupper documentation with rm4.9
4.7	06-12-2019	F. P. Schreuder	Updated INT_TEST behaviour and regenerated rm4.9 description
4.6	23-10-2019	F. P. Schreuder	Fixed endianness of TTC ToHost channel in FELIXDataFormats
4.5	09-05-2019	F. P. Schreuder	Regenerated registermap 4.7 documentation
4.4	18-03-2019	F. P. Schreuder	Regenerated rm 4.6 documentation
4.3	06-03-2019	F. P. Schreuder	regenerated register map and wupper documentation
4.2	17-10-2018	F. P. Schreuder	regenerated registermap documentation
4.1	30-08-2018	F. P. Schreuder	regenerated Wupper.pdf with rm4.4
4.0	17-05-2018	F. P. Schreuder	Added FLX-712 value to CARD_TYPE register
3.7	15-02-2017	F.P. Schreuder	Added interrupt #5, connected to xoff (full flags in CR)
3.6	10-10-2016	F.P. Schreuder	Improved Endless DMA description.
3.5	18-05-2016	F.P. Schreuder	Added ToHost Full interrupt and changed interrupt description into table
3.4	29-04-2016	A.O. Borga	Added Section supported tools; added sections labels for referencing; improved referencing layout by adding citations; added DMA descriptors documentation (section 3.1); documented endless DMA (section 3.2) and its handling improved interrupt routine (section 3.4) and interrupt handling
3.3	24-08-2015	F.P Schreuder	Added all bit fields and names of GBT Monitor and Control registers
3.2	14-08-2015	F.P Schreuder	Fixed bit fields in emuram write register
3.1	14-07-2015	A.O. Borga	Added appendix and moved the FELIX specific register map to it, added separate file for the lists of tables and figures
3.0	14-07-2015	A.O. Borga	Uniformed the naming convention to Wupper throughout the document, removed replay cache FIFO, updated all diagrams in section "design"
2.0	17-04-2015	A.O. Borga	Uniformed the naming convention to PCIe Engine throughout the document
1.9	03-2015	J.C. Vermeulen	New section on host software added and moved material from the section on operating the DMA controller to this section. The new section 6 is not yet complete

1.8	2015	F.P. Schreuder	Update of DMA registers
1.7	29-10-2014	A.O. Borga	Major global revision
1.6	28-10-2014	A.O. Borga	Updated PCIe coregen figures, modified appearance of paths throughout the text, fixed typos, updated the simulation and testing sections, added the interrupt handling section, reversed the order of this table
1.5	23-10-2014	F.P. Schreuder	Updated register map, figures and pepo commands, some cosmetic improvements
1.4	23-09-2014	J.C. Vermeulen	Updated figures
1.3	23-09-2014	F.P. Schreuder	Updated pepo commands for memory allocation
1.2	19-09-2014	F.P. Schreuder	Added All pages of Xilinx core wizard
1.1	19-09-2014	F.P. Schreuder	Applied modifications after Andrea's review
1.0	16-09-2014	F.P. Schreuder	created

1 Introduction

Wupper[1] is designed for the ATLAS / FELIX project [1], to provide a simple Direct Memory Access (DMA) interface for the Xilinx Virtex-7 PCIe Gen3 hard block and has later been ported to the Kintex Ultrascale, Virtex Ultrascale+ and Versal Prime series. The core is not meant to be flexible among different architectures, but especially designed for the 256 and 512 bit wide AXI4-Stream interface [4] of the Xilinx Virtex-7 and Ultrascale FPGA Gen3 Integrated Block for PCI Express, and the Ultrascale+ and Versal Prime Gen4 Integrated Block for PCI Express (PCIe) [5, 6, 7, 8].

The purpose of Wupper is therefore to provide an interface to a standard FIFO. This FIFO has the same width as the Xilinx AXI4-Stream interface (256 or 512 bits) and runs at 250 MHz. The user application side of the FPGA design can simply read or write to the FIFO; Wupper will handle the transfer into Host PC memory, according to the addresses specified in the DMA descriptors. Several descriptors can be queued, up to a maximum of 8, and they will be processed sequentially one after the other. The number of descriptors (NUMBER_OF_DESCRIPTORS generic) plays an important role, it determines the total number of descriptors, but also the number of FIFO interfaces in the ToHost direction. The last descriptor is always dedicated for FromHost (DMA memory read from the server) transactions, all other descriptors are dedicated for ToHost transfers (Memory writes from the FPGA into the server memory).

Another functionality of Wupper is to manage a set of DMA descriptors, with an $address$, a $read/\overline{write}$ flag, the $transfersize$ (number of 32 bit words) and an $enable$ line. These descriptors are mapped as normal PCIe memory or IO registers. Besides the descriptors and the enable line (one per descriptor), a status register for every descriptor is provided in the register map.

For synthesis and implementation of the Xilinx specific IP cores, it is recommend to use the latest Xilinx Vivado release as listed in section 2. The cores (FIFO, clock wizard and PCIe) are provided in the Xilinx .xci format, as well as the constraints file (.xdc) is in the Vivado Format.

For portability reasons, no Xilinx project files will be supplied with the core, but a bundle of TCL scripts has been supplied to create a project and import all necessary files, as well as to do the synthesis and implementation. These scripts will be described later in this document.

[1] The person performing the act of bongelwuppen, the Gronings version of the famous Frisian sport of the Fierljeppen (canal pole vaulting) https://nds-nl.wikipedia.org/wiki/Nedersaksische_sp%C3%B6llegies#Bongelwuppen

2 Compatibility

FELIX had been tested on the following platforms and tools:

1. Operating systems:
    - Scientific Linux CERN 6, kernel 2.6
    - Scientific Linux 7, kernel 3.10
2. Xilinx Vivado:
    - 2020.1: migrated 11-2020
    - 2018.1: migrated 05-2019
    - 2015.4: migrated 02-2016
    - 2014.4: initial version
3. Xilinx FPGA:
    - Virtex-7 690T
    - Kintex Ultrascale XCKU115
    - Virtex Ultrascale+ VU9P, VU37P
    - Versal Prime VM1802

3 Core Architecture

Xilinx has introduced the AXI4-Stream interface [4] for the PCIe EndPoint core: a simplified version of the ARM AMBA AXI bus [2]. This interface does not contain any address lines, instead the address and other information are supplied in the header of each PCIe Transaction Layer Packet (TLP). Figure 1 shows the structure of the Wupper_core design. The Wupper_core is divided in two parts:

1. **DMA Control:**
   This is the entity in which the Descriptors are parsed and fed to the engine, and where the Status register of every descriptor can be read back through PCIe. Depending on the address range of the descriptor, the pointer of the current address is handled by DMA Control and incremented every time a TLP completes. DMA Control also handles the circular buffer DMA if this is requested by the descriptor (See 3.2).

   DMA control contains a register map, with addresses to the descriptors, status registers and external registers for the user space register map.

2. **DMA Read Write:**
   This entity contains two processes:

   - *ToHost / Add Header:* In the first process the descriptors are read and a header according to the descriptor is created. If the descriptor is a ToHost descriptor, the payload data is read from the FIFO and added after the header. This process also takes care of switching to the next active DMA descriptor, which is leading for selecting the MUX on the output ports of the ToHostFifo's.
   - *FromHost / Strip Header:* In the second process the header of the received data is removed and the length is checked; then the payload is shifted into the FIFO.

   Both processes can fire an MSI-X type interrupt by means of the interrupt controller when finished.

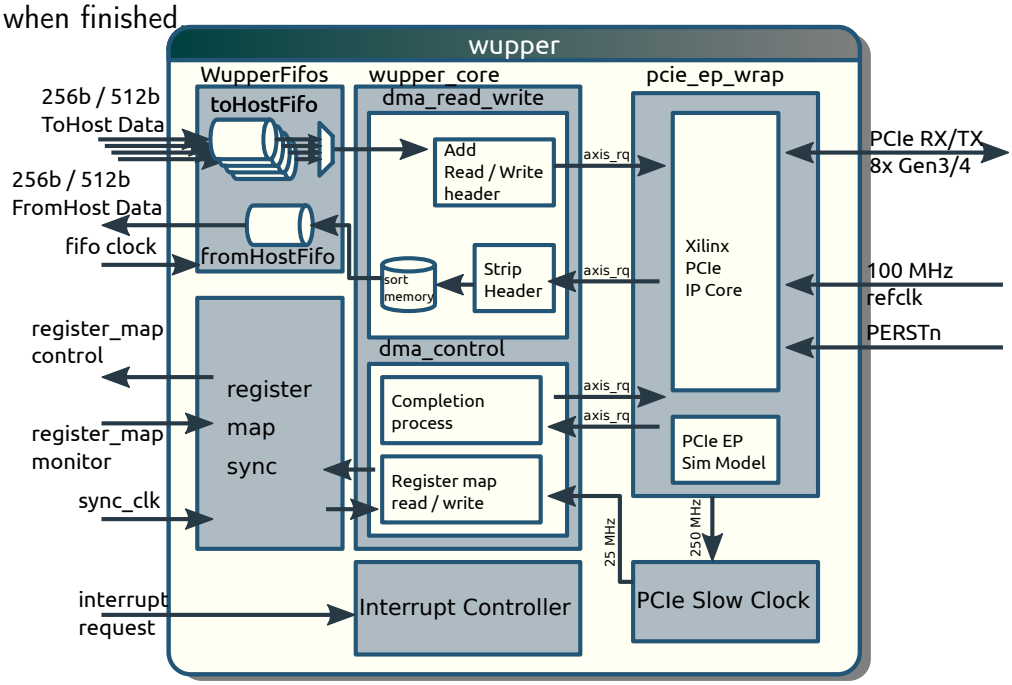

Figure 1: Structure of the Felix PCIe Engine.

Figure 1 shows a synchronization stage for the IO and external registers, The user space registers are stored and processed in the 25 MHz clock domain in order to relax timing closure of the design. The synchronization stage synchronizes the register map again to the clock used in the application design (sync_clk).

The DMA Control process always responds to a request with a certain req_type from the server. It responds only to IO and Memory reads and writes; for all other request types it will send an unknown request reply. If the data in the payload contains more than 128 bits, the process will send a "completion abort" reply and go back to idle state. The maximum register size has been set to 128 bits because this is a useful maximum register size; it is also the maximum payload that fits in one 250 MHz clock cycle of the AXI4-Stream interface.

The add_header process selects the descriptor and sets the ToHostFifo MUX accordingly. Based on the descriptor content, it requests a read or write to/from the server memory. If the descriptor is set to ToHost, it also initiates a FIFO read and adds the data into the payload of the PCIe TLP (Transaction Layer Packet). When the descriptor is set to FromHost this process only creates a header TLP with no payload, to request a certain amount of data from the server memory that fits in one TLP.

The DMA FromHost process checks the size of the payload against the size in the TLP header, the data will be pushed into the FromHost FIFO.

3.1 DMA descriptors

Each transfer To and From Host is achieved by means of setting up descriptors on the server side, which are then processed by Wupper. The descriptors are set in the BAR0 section of the register map (see Appendix 6). An extract of the descriptors and their registers is shown in Table 2 below. The register map in BAR0 has space for a maximum of 8 DMA descriptors, but the actual number of descriptors that are implemented is determined by the generic NUMBER_OF_DESCRIPTORS. The descriptor at NUMBER_OF_DESCRIPTORS-1 is the FromHost descriptor which always has the READ_WRITE bitfield set to 1 (FROMHOST) and the descriptors 0 to NUMBER_OF_DESCRIPTORS-2 are implemented as ToHost descriptors. An additional special FromHost descriptor is implemented at NUMBER_OF_DESCRIPTORS, this is the so called trickle descriptor (see 3.3) which is similar to the other FromHost descriptor, but it ignores the pc_pointer. The number of ToHost FIFOs is automatically determined by the same generic, as well as the ToHost FIFO depth. Setting NUMBER_OF_DESCRIPTORS to 6 (default in phase 2 FELIX) will result in 4 ToHost descriptors and FIFOs (descriptor 0..3) and a single FromHost descriptor / FIFO (descriptor 4).

Address	Name/Field	Bits	Type	Description
0x0000	DMA_DESC_0			
	END_ADDRESS	127:64	W	End Address
	START_ADDRESS	63:0	W	Start Address
0x0010	DMA_DESC_0a			
	PC_POINTER	127:64	W	server Read Pointer
	WRAP_AROUND	12	W	Wrap around
	READ_WRITE	11	R	1: FromHost/ 0: ToHost
	NUM_WORDS	10:0	W	Number of 32 bit words
...				
0x0200	DMA_DESC_STATUS_0			
	EVEN_PC	66	R	Even address cycle server
	EVEN_DMA	65	R	Even address cycle DMA

	DESC_DONE CURRENT_ADDRESS	64 63:0	R R	Descriptor Done Current Address
		...		
0x0400	DMA_DESC_ENABLE	7:0	W	Enable descriptors 7:0. One bit per descriptor. Cleared when Descriptor is handled.

Table 2: DMA descriptors types.

Every descriptor has a set of registers, with the following specific functions:

- DMA_DESC: the register containing the start ($start\_address$) and the end ($end\_address$) memory addresses of a DMA transfer; both handled by the server (software API).
- DMA_DESC_a: integrates the information above by adding (i) the status of the read pointer on the server side ($pc\_pointer$), (ii) the wrap around functionality enabling ($wrap\_around$, see Section 3.2 below), (iii) the FromHost ("1") and ToHost ("0") transfer direction bit ($read\_write$), and (iv) the number of 32 bits words to be transferred (num_words)
- DMA_DESC_STATUS: status of a specific descriptor including (i) wrap around information bits ($even\_pc$ and $even\_dma$), (ii) completion bit ($desc\_done$, (iii) DMA pointer current address ($current\_address$)
- DMA_DESC_ENABLE: the descriptors enable register (dma_desc_enable), one bit per descriptor

3.2 Endless DMA with a circular buffer and wrap around

In $single\,shot$ transfer, the DMA ToHost process continues sending data TLPs (Transaction Layer Packets) until the end address ($end\_address$) is reached. The server can check the status of a certain DMA transaction by looking at the $desc\_done$ flag and the $current\_address$. Another possible operation mode is the so- called $endless\ DMA$: the DMA continues its action and starts over (wrap-around) at start address ($start\_address$) whenever the end address ($end\_address$) is reached. The second mode is enabled by asserting the wrap-around ($wrap\_around$) bit. In this mode the server has to provide another address named server pointer ($PC\_read\_pointer$): indicating where it has last read out the memory. After wrapping around the DMA core will transfer To Host memory until the $PC\_read\_pointer$ is reached. The server read pointer should be updated more often than the wrap-around time of the DMA, however it should not be read too often as that would take up all the bandwidth, limiting the speed of the DMA transfer in progress. A typical rule of thumb to determine what "too often" means is that software should not update the pointer every clock cycle, but rather after processing a block of a few kB of data.

In order to determine whether Wupper is processing an address behind or in front of the server, Wupper keeps track of the number of wrap around occurrences. In the DMA status registers the even_cycle bits displays the status of the wrap-around cycle. In every even cycle (starting from 0), the bits are 0, and every wrap around the status bits will toggle. The $even\_pc$ bit flags a $PC\_read\_pointer$ wrap-around, the $even\_dma$ a Wupper wrap-around. By looking at the wrap-around flags the server can also keep track of its own wrap-arounds. Note that while in the $endless\ DMA$ mode ($wrap\_around$ bit set), the $PC\_read\_pointer$ has to be maintained by the server (software API) and kept

within the start and end address range for Wupper to function correctly. Figure 2 below shows a diagram of the two pointers racing each other, and the different scenarios in which they can be found with respect to each other.

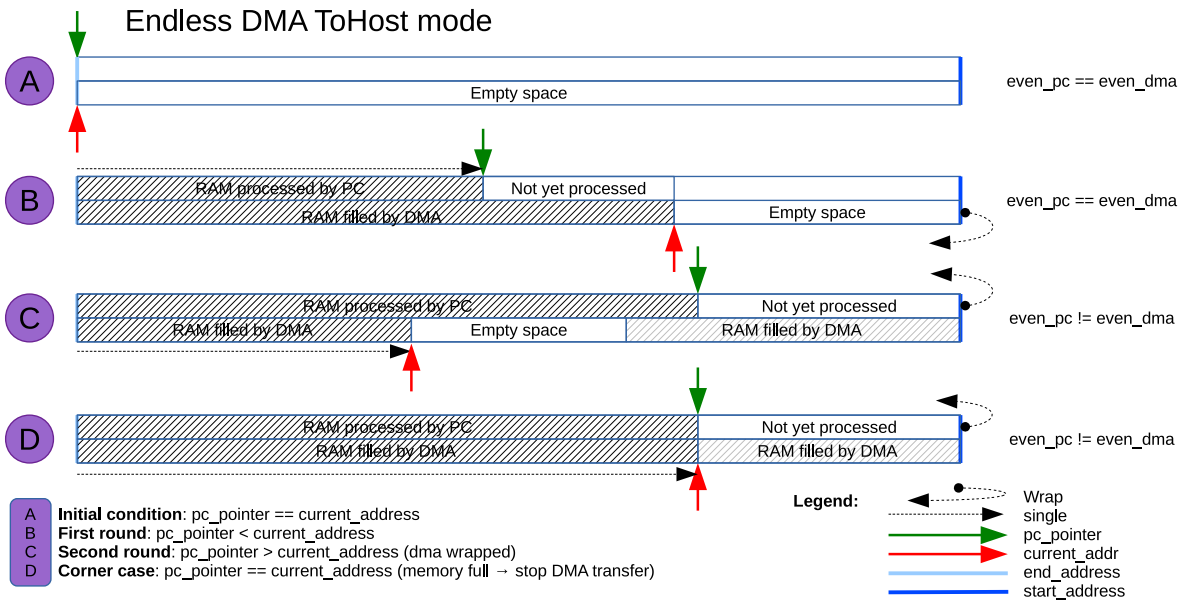

Figure 2: Endless DMA buffer and pointers representation diagram in ToHost mode.

Looking at Figure 2 above, the following scenarios can be described:

- A : start condition, both the server and the DMA have not started their operation.
- B : normal condition, the PC_read_pointer stays behind the DMA's current_address
- C : normal condition, the DMA's current_address has wrapped around and has to stay behind the PC_read_pointer
- D : the server is reading too slow, the DMA is stalled because the server read pointer is not advancing fast enough, the DMA current_address has to stay behind.

If the DMA descriptor is set to FromHost, the comparison of the even bits is inverted, as the server has to fill the buffer before it is processed in the same cycle. In this mode the $pc\_read\_pointer$ is also maintained by the software API, however it is indicating the address up to where the server has filled the memory. In the first cycle the DMA has to stay behind the read pointer, when the server has wrapped around, the DMA can process memory up to $end\_address$ until it also wraps around.

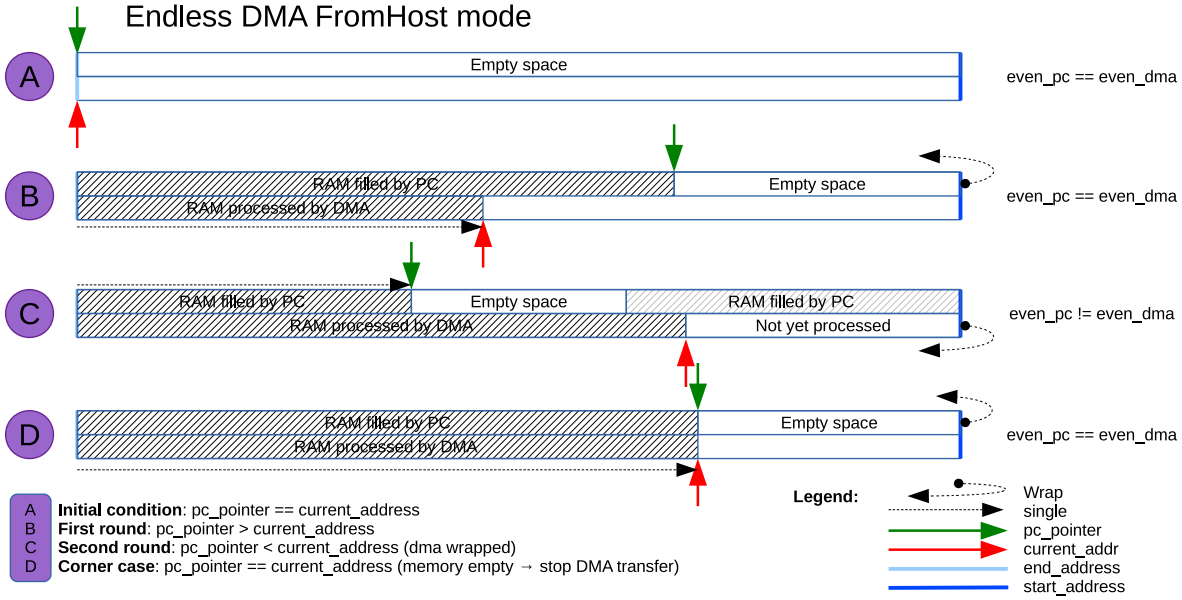

Figure 3: Endless DMA buffer and pointers representation diagram in FromHost mode.

Looking at Figure 3 above, the following scenarios can be described:

- A : start condition, both the server and the DMA have not started their operation.
- B : normal condition, the DMA's current_address stays behind the PC_read_pointer
- C : normal condition, the PC_read_pointer has wrapped around and has to stay behind the DMA's current_address
- D : the server is writing too slow, the DMA is stalled because the server read pointer is not advancing fast enough, the DMA current_address has to stay behind.

3.3 Trickle descriptor

The trickle descriptor is a special FromHost descriptor which is implemented at NUMBER_OF_DESCRIPTORS. This descriptor works exactly as the other FromHost descriptor if WRAP_AROUND is '0' (single shot DMA), but with WRAP_AROUND set to '1', it ignores the PC_POINTER so the throughput is not throttled by the software. Instead the throughput is limited by the target E-Link as defined in the contents of the cmem_rcc buffer containing the trickle commands. If all the data in that buffer is targeted to one 80 Mb/s E-Link the DMA throughput is also limited to 80 Mb/s automatically. Wupper will keep playing back the trickle memory in the host PC until the DMA_DESC_ENABLE bit is cleared by te software.

To gracefully disable the trickle descriptor, letting Wupper complete the complete range of the memory, the WRAP_AROUND bit can first be cleared, and then the DMA_DESC_-ENABLE bit will clear automatically when the transfer has completed.

3.4 Interrupt controller

Wupper is equipped with an interrupt controller supporting the MSI-X (Message Signaled Interrupt eXtended) as described in "Chapter 17: Interrupt Support" page 812 and onwards of [9]. In particular the chapter and tables in "MSI-X Capability Structure".

The MSI-X Interrupt table contains eight interrupts; this number can be extended by a generic parameter in the firmware. All interrupts are mapped to the data_available interrupt of the corresponding ToHost descriptor, formerly known as interrupt number 2 in phase1 (rm-4.x) firmware. All the other interrupt sources have been removed since multiple ToHost descriptors were introduced in rm-5.x. The interrupts are detailed in Table 3.

Table 3: PCIe interrupts.

Interrupt	Name	Description
0	ToHost 0 Available	Fired when data becomes available in the ToHost FIFO 0 (falling edge of ToHostFifoProgEmpty)
1	ToHost 1 Available	Fired when data becomes available in the ToHost FIFO 1 (falling edge of ToHostFifoProgEmpty)
1	ToHost 2 Available	Fired when data becomes available in the ToHost FIFO 2 (falling edge of ToHostFifoProgEmpty)
3	ToHost 3 Available	Fired when data becomes available in the ToHost FIFO 3 (falling edge of ToHostFifoProgEmpty)
4	ToHost 4 Available	Fired when data becomes available in the ToHost FIFO 4
5	crDownXoff	ToHost combined full flags (CR xoff)
6	BUSY change	Fired when the busy LEMO signal changes
7	ToHost Full	Fired when the ToHost FIFO becomes full

All Interrupts are fired when enough data has arrived in the ToHost fifo to fill at least one TLP of data. Once an interrupt has fired, it will not produce an additional interrupt until any write occurs to a register in BAR0. The idea is that this write occurs when the SW_POINTER has been updated by the software.

All the interrupts can also be fired from the register INT_TEST, by setting the bitfield IRQ to the desired interrupt number. This write action will fire a single interrupt.

4 Xilinx PCIe EndPoint Core

Wupper was built around the interface of the Virtex-7 FPGA Gen3 Integrated Block for PCI Express v4.3 [5], and was later ported to other Xilinx PCIe hard blocks:

- Virtex-7 FPGA Gen3 Integrated Block for PCI Express [5]. Wupper was tested on Virtex7 with the VC709 (FLX709) board and the HTG710 (FLX710) boards using the XC7VX690T FPGA. (PCIe Gen3x8)
- UltraScale Devices Gen3 Integrated Block for PCI Express [6]. Wupper was tested with the BNL711 (FLX711) and BNL712 (FLX712) boards, using the KU115 FPGA. (2x PCIe Gen3x8 with a PCIe x16 switch)
- UltraScale+ Devices Integrated Block for PCI Express [7]. Wupper was tested with the VCU128-es1 (FLX128) (VU37P FPGA), the XUPP3R (VU9P FPGA) (FLX800) and the BNL801 board (FLX801) (VU9P FPGA) 2x PCIe Gen4x8 bifurcated. [2]
- Versal ACAP Integrated Block for PCI Express [8]. Wupper was tested on the VMK180 board (VM1802 ACAP), PCIe Gen4x8

This core is using a PCIe hard block in the Virtex-7 FPGA. The hard block is equipped with an AXI4-Stream interface.

4.1 Xilinx AXI4-Stream interface

The interface has the advantage that it has two separate bidirectional AXI4-Stream interfaces. The two interfaces are the requester interface, with which the FPGA issues the requests and the PC replies, and the completer interface where the PC takes initiative.

bus	Description	Direction
`axis_rq`	**R**equester re**Q**uest. This interface is used for DMA, the FPGA takes the initiative to write to this AXI4-Stream interface and the PC has to answer.	FPGA → PC
`axis_rc`	**R**equester **C**ompleter. This interface is used for DMA reads (from PC memory to FPGA), this interface also receives a reply message from the PC after a DMA write.	PC → FPGA
`axis_cq`	**C**ompleter re**Q**uest. This interface is used to write the DMA descriptors as well as some other registers.	PC → FPGA
`axis_cc`	**C**ompleter **C**ompleter. This interface is used as a reply inteface for register reads, as well as a reply header for a register write.	FPGA → PC

Table 5: AXI4-Stream streams.

4.2 Configuration of the core

The Xilinx PCIe EndPoint core is configured as a PCI express Gen3 (8.0GT/s) End Point with 8 lanes and the Physical Function (PF0) max payload size is set to 1024 bytes. AXI-ST Frame Straddle is disabled and the client tag is enabled. All other options are set to

[2] For the VU9P FPGA, PCIe Gen4 is not officially supported, but it was demonstrated to work. It can be enabled only on Vivado 2018.1 using a tcl command or by editing the .xci file

default, the reference clock frequency is 100MHz and the only option for the AXI4-Stream interface is 256 bit at 250MHz, see Figures 4 to 14.

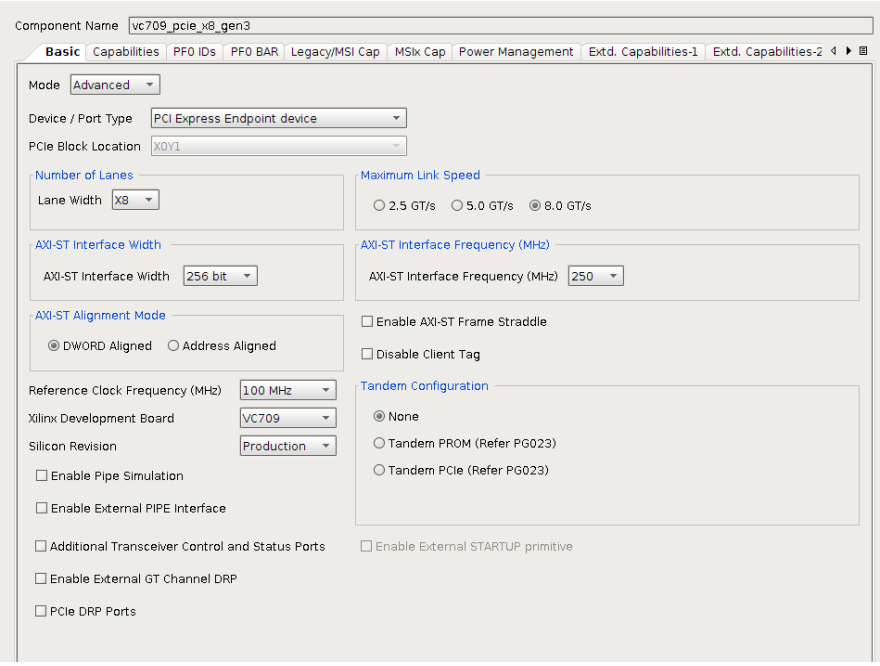

Figure 4: PCIe core configuration in Vivado [Basic].

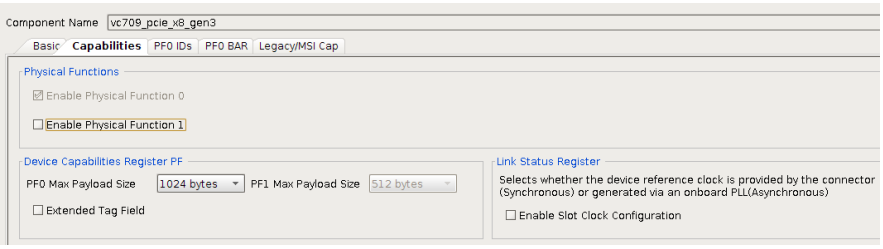

Figure 5: PCIe core configuration in Vivado [Capabilities].

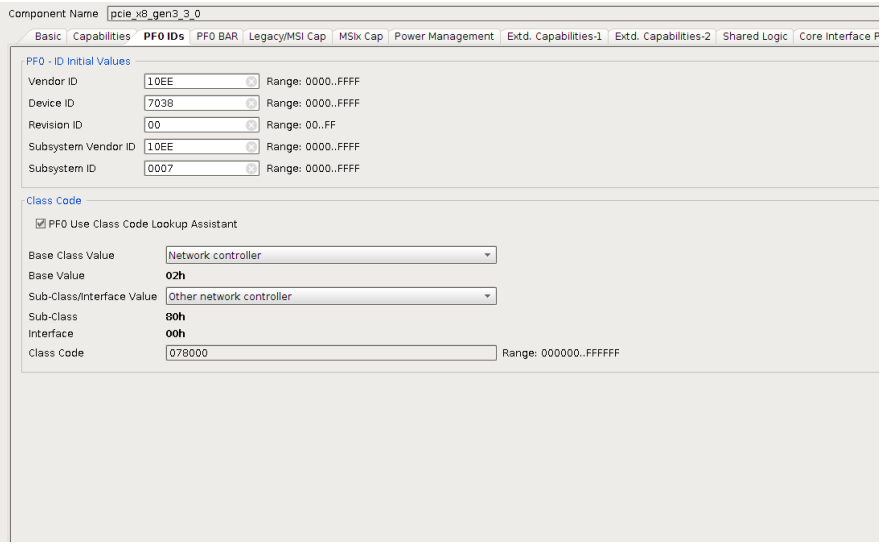

Figure 6: PCIe core configuration in Vivado [PF0 IDs].

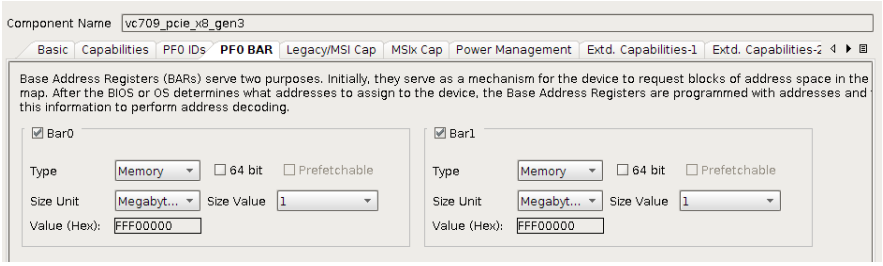

Figure 7: PCIe core configuration in Vivado [PF0 BAR].

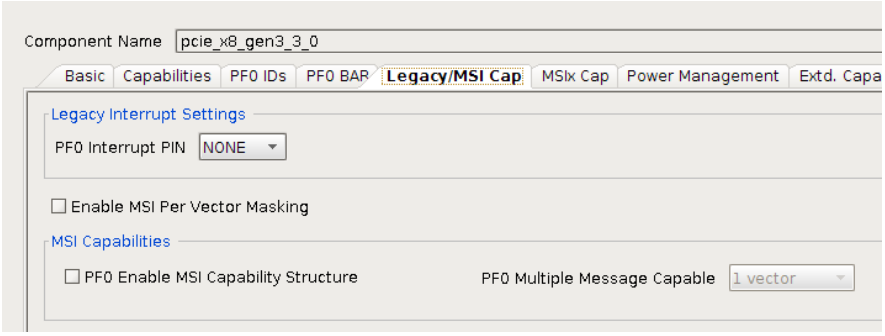

Figure 8: PCIe core configuration in Vivado [Legacy/MSI Cap].

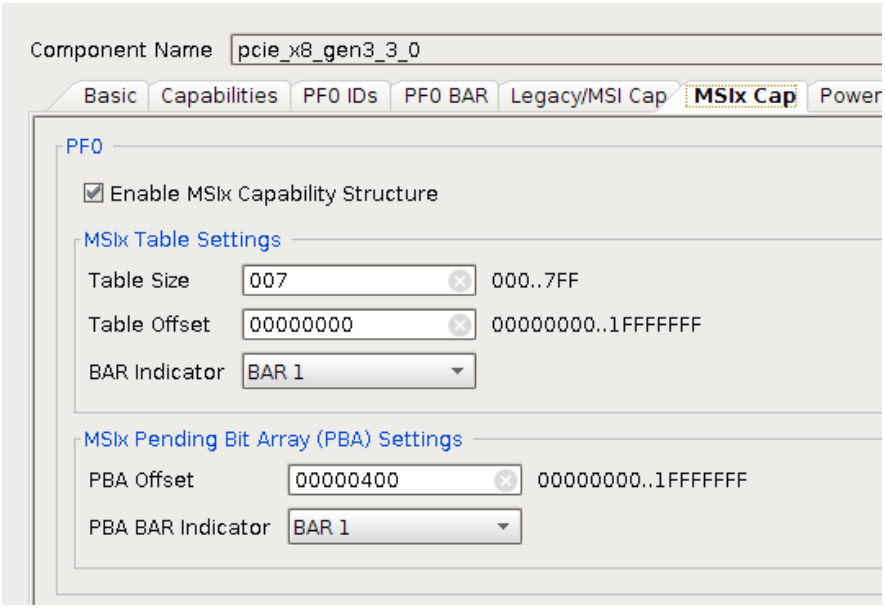

Figure 9: PCIe core configuration in Vivado [MSIx].

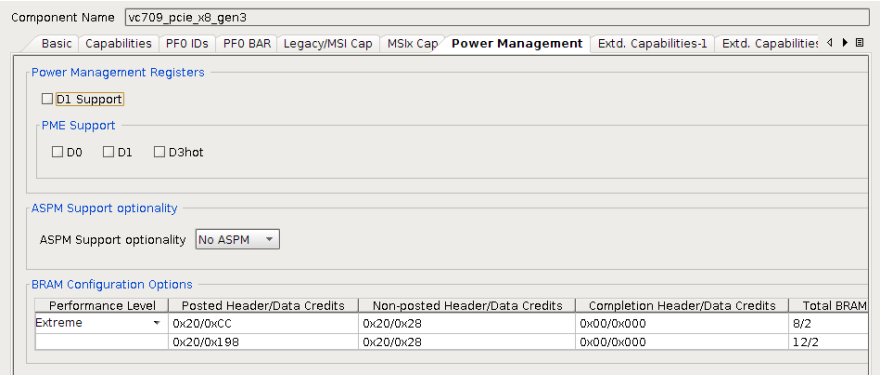

Figure 10: PCIe core configuration in Vivado [Power Management].

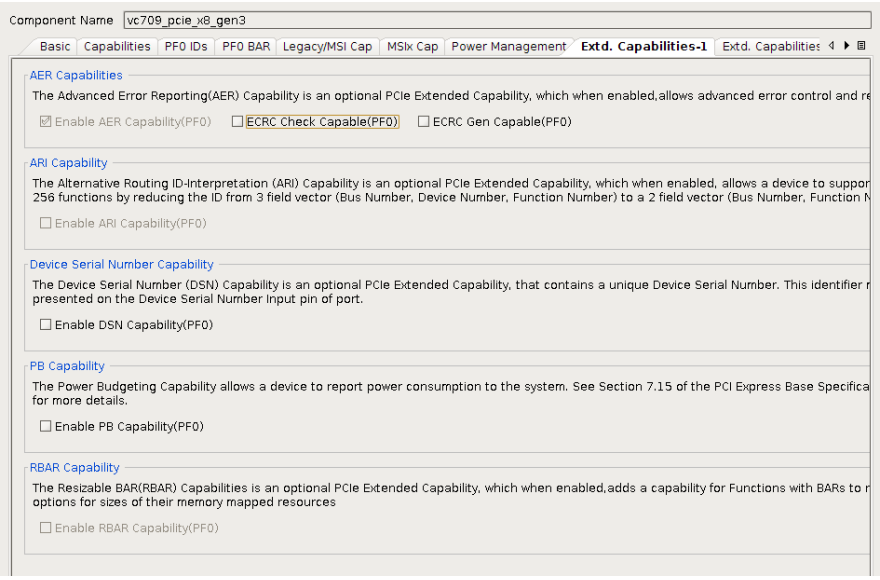

Figure 11: PCIe core configuration in Vivado [Extd. Capabilities 1].

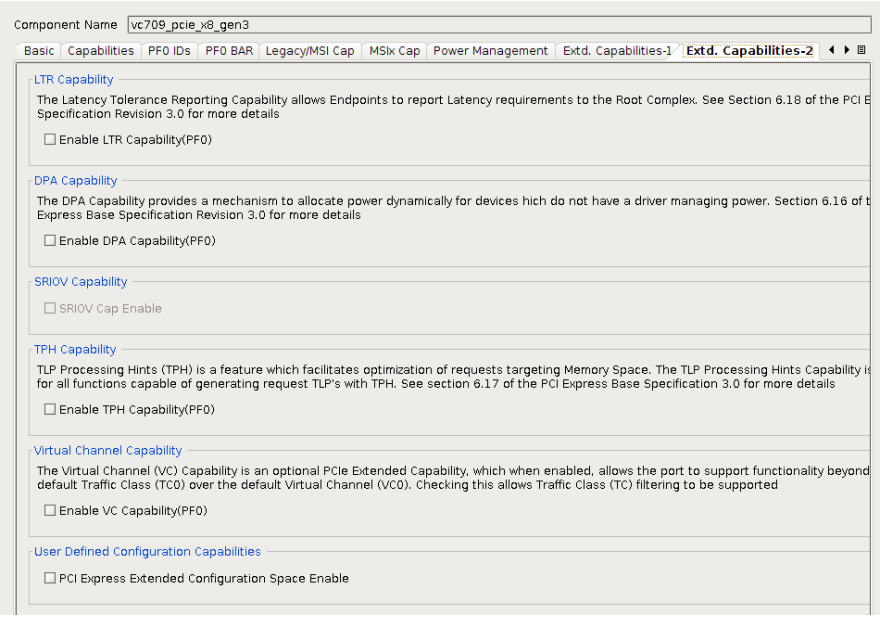

Figure 12: PCIe core configuration in Vivado [Extd. Capabilities 2].

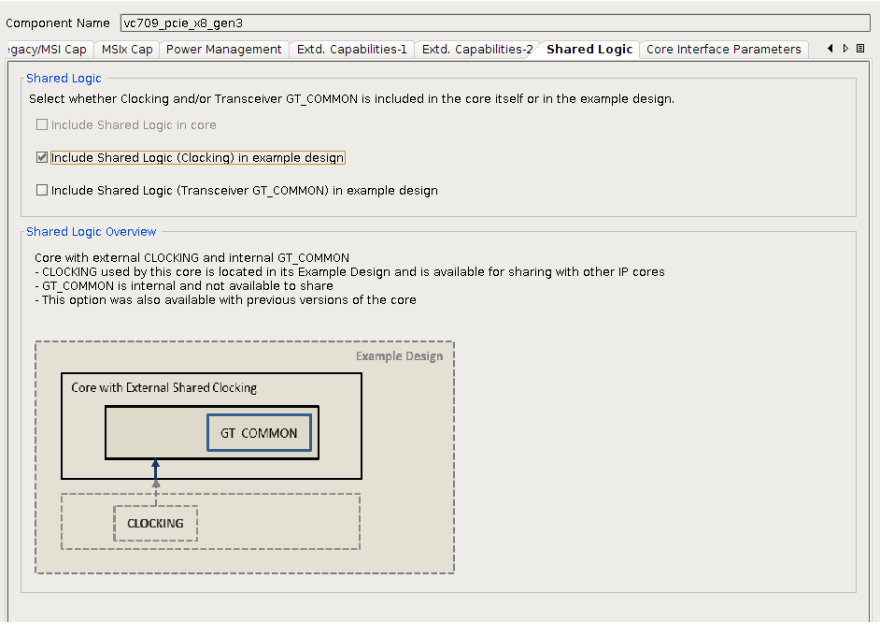

Figure 13: PCIe core configuration in Vivado [Shared LogicMSIx].

Figure 14: PCIe core configuration in Vivado [Core Interface Parameters].

5 Obtaining and building Wupper

Wupper was written in VHDL, however a number of files are processed with the registers defined in registers-x.x.yaml using WupperCodeGen [15]. WupperCodeGen is a Jinja2 based templating engine which generates a template file (for instance dma_control.vhd.template) into the generated source file. The WupperCodeGen software is used to generate the VHDL source, the C/CPP headers as wel as the register-map documentation in this document, see 6

directory	**contents**
firmware/constraints	Contains the XDC files with Vivado constraints including Chipscope ILA definitions.
firmware/output	Empty placeholder where bit files will be generated
firmware/Projects	Empty placeholder where the Vivado projects will be generated
firmware/scripts/FELIX_top	This directory contains scripts to create the vivado project and to run synthesis and implementation, see later this chapter.
firmware/scripts/filesets	This directory contains scripts to define sets of VHDL file for a certain firmware functional block. For instance wupper_fileset.tcl contains all the necessary VHDL, XCI, XDC and BD files to generate Wupper for all hardware platforms.
firmware/scripts/helper	Contains a set of scripts that can be sourced from the scripts in simulation and scripts/FELIX_top in order to easily create projects for the different tools (Vivado, Questasim, Sigasi) and to start an implementation, set the necessary generics, create reports and a bitfile for the required FPGA.
firmware/simulation/Wupper	Contains a testbench (wupper_tb.vhd) and a PCIe endpoint functional model (pcie_ep_sim_model.vhd), both using the UVVM library. These simulation files can used together with the other Wupper sources to demonstrate multi descriptor DMA transfers and Register reads / writes.
firmware/sources/pcie	This directory contains the source files of Wupper that are not generated using WupperCodeGen.
firmware/sources/templates	This directory contains the yaml definition of the register map, the .vhd.template files and the generated .vhd files.
firmware/sources/ip_cores	Contains a Vivado .xci file for the PCIe ip cores

Table 7: Directories in the repository.

Please note that if changes to any of the core are made, a manual copy of the relevant .xci file in *firmware/Projects/felix_top/felix_top.srcs/sources_1/ip* should be made to the relevant folder in */firmware/sources/ip_cores*.

5.1 Create the Vivado Project

The vivado project is not supplied in the GIT tree, instead a .tcl script is provided to generate the project. To create the project, open Vivado without a project, then open the TCL console and run the following commands.

Listing 1: Create Vivado Project.

```
cd felix/firmware/scripts/FELIX_top/
source ./FLX712_import_vivado.tcl
```

FLX712 can be replaced with any of the other supported hardware platforms. A project should now be created in *firmware/Projects/*. **beware that this script will overwrite and recreate the project if it exists already.**

5.2 Running synthesis and implementation

When the project has been created, you can simply press the buttons to run synthesis and implementation of the design, but a tcl script has been created to run these steps automatically. Additionally the script will create the bitfile in the *firmware/output* directory, as well as an .mcs file and an .ltx file, containing the ChipScope ILA probes. All those 3 files have a timestamp in their filename so any previous synthesis output will be maintained. The script can simply be executed if the project is open.

Listing 2: start synthesis / implementation.

```
cd felix/firmware/scripts/FELIX_top/
source ./do_implementation_BNL712.tcl
```

6 Simulation

The directory *firmware/simulation/Wupper* contains all necessary testbenches (wupper_tb.vhd, pcie_ep_sim_model.vhd) to run the simulation in Mentor Graphics Modelsim or Questasim [12].

The directory simulation/UVVMExample contains a file modelsim.ini with some standard information, there is also a script "ci.sh" wich will execute the UVVM based simulation. It assumes that questasim 2019.1 is installed, the Xilinx libraries are compiled in simulation/xilinx_lib and the UVVM library is compiled in simulation/UVVM. The wupper simulation can be started by executing

Listing 3: Run the simulation.

```
cd FELIX/firmware/simulation/UVVMExample
./ci.sh Wupper
```

By default the simulation starts in command line mode. If GUI mode is desired (e.g. to view waveforms), the ci.sh script can be edited, and the "-c" parameter from the vsim command can be removed.

Appendix A FELIX register map, version 5.0

Starting from the offset address of BAR0, BAR1 and BAR2. BAR0 only contains registers associated with DMA.

Bar0						
DMA_DESC						
0x0000	0,1	DMA_DESC_0				
		END_ADDRESS	127:64	W	End Address	0x0000000000000000
		START_ADDRESS	63:0	W	Start Address	0x0000000000000000
0x0010	0,1	DMA_DESC_0a				
		SW_POINTER	127:64	W	Pointer controlled by the software, indicating read or write status for circular DMA	0x0000000000000000
		WRAP_AROUND	12	W	Wrap around	0x0
		FROMHOST	11	R	1: fromHost/ 0: toHost	0x0
		NUM_WORDS	10:0	W	Number of 32 bit words	0x00
...						
0x00E0	0,1	DMA_DESC_7				
		END_ADDRESS	127:64	W	End Address	0x0000000000000000
		START_ADDRESS	63:0	W	Start Address	0x0000000000000000
0x00F0	0,1	DMA_DESC_7a				
		SW_POINTER	127:64	W	Pointer controlled by the software, indicating read or write status for circular DMA	0x0000000000000000
		WRAP_AROUND	12	W	Wrap around	0x0
		FROMHOST	11	R	1: fromHost/ 0: toHost	0x0
		NUM_WORDS	10:0	W	Number of 32 bit words	0x00
DMA_DESC_STATUS						
0x0200	0,1	DMA_DESC_STATUS_0				
		EVEN_PC	66	R	Even address cycle PC	0x0
		EVEN_DMA	65	R	Even address cycle DMA	0x0
		DESC_DONE	64	R	Descriptor Done	0x0
		FW_POINTER	63:0	R	Pointer controlled by the firmwarre, indicating where the DMA is busy reading or writing	0x0000000000000000

		...				
0x0270	0,1	DMA_DESC_STATUS_7				
		EVEN_PC	66	R	Even address cycle PC	0x0
		EVEN_DMA	65	R	Even address cycle DMA	0x0
		DESC_DONE	64	R	Descriptor Done	0x0
		FW_POINTER	63:0	R	Pointer controlled by the firmwarre, indicating where the DMA is busy reading or writing	0x0000000000000000
0x0300	0,1	BAR0_VALUE	31:0	R	Copy of BAR0 offset reg.	0x00000000
0x0310	0,1	BAR1_VALUE	31:0	R	Copy of BAR1 offset reg.	0x00000000
0x0320	0,1	BAR2_VALUE	31:0	R	Copy of BAR2 offset reg.	0x00000000
0x0400	0,1	DMA_DESC_ENABLE	7:0	W	Enable descriptors 7:0. One bit per descriptor. Cleared when Descriptor is handled.	0x00
0x0420	0,1	DMA_RESET	any	T	Reset Wupper Core (DMA Controller FSMs)	0x0
0x0430	0,1	SOFT_RESET	any	T	Global Software Reset. Any write resets applications, e.g. the Central Router.	0x0
0x0440	0,1	REGISTER_RESET	any	T	Resets the register map to default values. Any write triggers this reset.	0x0
0x0450	0,1	FROMHOST_FULL_THRESH				
		THRESHOLD_ASSERT	22:16	W	Assert value of the FromHost programmable full flag	0x0
		THRESHOLD_NEGATE	6:0	W	Negate value of the FromHost programmalbe full flag	0x0
0x0460	0,1	TOHOST_FULL_THRESH				
		THRESHOLD_ASSERT	27:16	W	Assert value of the ToHost programmable full flag	0x000
		THRESHOLD_NEGATE	11:0	W	Negate value of the ToHost programmalbe full flag	0x000
0x0470	0,1	BUSY_THRESHOLD_ASSERT	63:0	W	Tohost or Fromhost busy will be asserted in circular DMA mode when the server PC buffer gets full (space below ASSERT threshold)..	0x0000000006400000
0x0480	0,1	BUSY_THRESHOLD_NEGATE	63:0	W	Tohost or Fromhost busy will be negated in circular DMA mode when the server PC buffer gets less full (space above NEGATE threshold).	0x0000000006E00000
0x0490	0,1	BUSY_STATUS	0	R	A tohost descriptor passed BUSY_THRESHOLD_ASSERT, busy flag set	0x0
0x04A0	0,1	PC_PTR_GAP	63:0	W	This is the minimum value that the pc_pointer in a descriptor has to decrease in order to flip the evencycle_pc bit	0x0000000001000000
0x04B0	0,1	TOHOSTFIFO_EMPTY	3:0	R	Empty flags of the ToHost FIFOs in Wupper	0x0
0x04C0	0,1	TOHOSTFIFO_PEMPTY	3:0	R	Programmable empty flags of the ToHost FIFOs in Wupper	0x0
0x04D0	0,1	FROMHOSTFIFO_FULL	0	R	Full flag of the FromHost FIFO in Wupper	0x0

0x04E0	0,1	FROMHOSTFIFO_PFULL		0	R	Programmable full flag of the FromHost FIFO in Wupper	0x0

Table 8: FELIX register map BAR0.

BAR1 stores registers associated with the Interrupt vector.

Bar1						
INT_VEC						
0x0000	0,1	INT_VEC_0				
		INT_CTRL	127:96	W	Interrupt Control	0x00000000
		INT_DATA	95:64	W	Interrupt Data	0x00000000
		INT_ADDRESS	64:0	W	Interrupt Address	0x0000000000000000
...						
0x00F0	0,1	INT_VEC_15				
		INT_CTRL	127:96	W	Interrupt Control	0x00000000
		INT_DATA	95:64	W	Interrupt Data	0x00000000
		INT_ADDRESS	64:0	W	Interrupt Address	0x0000000000000000
0x0100	0,1	INT_TAB_ENABLE	7:0	W	Interrupt Table enable Selectively enable Interrupts	0x00

Table 9: FELIX register map BAR1.

BAR2 stores registers for the control and monitor of HDL modules inside the FPGA other than Wupper. A portion of this register map's section is dedicated for control and monitor of devices outside the FPGA; as for example simple I2C devices.

Bar2						
Generic Board Information						
0x0000	0,1	REG_MAP_VERSION	15:0	R	Register Map Version, 5.0 formatted as 0x0500	0x0000
0x0010	0,1	BOARD_ID_TIMESTAMP	39:0	R	Board ID Date / Time in BCD format YYMMDDhhmm	0x0000000000
0x0030	0,1	GIT_COMMIT_TIME	39:0	R	Board ID GIT Commit time of current revision, Date / Time in BCD format YYMMDDhhmm	0x0000000000
0x0040	0,1	GIT_TAG	63:0	R	String containing the current GIT TAG	0x0000000000000000
0x0050	0,1	GIT_COMMIT_NUMBER	31:0	R	Number of GIT commits after current GIT_TAG	0x00000000
0x0060	0,1	GIT_HASH	31:0	R	Short GIT hash (32 bit)	0x00000000
0x0070	0,1	STATUS_LEDS	7:0	W	Board GPIO Leds	0xAB
0x0080	0,1	GENERIC_CONSTANTS				
		TRICKLE_DESCRIPTOR_INDEX	35:32	R	Index of the (first if more than one) Trickle descriptor	0x0
		FROMHOST_DESCRIPTOR_INDEX	31:28	R	Index of the (first if more than one) FromHost descriptor	0x0
		TRICKLE_DESCRIPTORS	27:24	R	Number of Trickle descriptors	0x0
		FROMHOST_DESCRIPTORS	23:20	R	Number of FromHost descriptors	0x0
		TOHOST_DESCRIPTORS	19:16	R	Number of ToHost descriptors	0x0
		INTERRUPTS	15:8	R	Number of Interrupts	0x00
		DESCRIPTORS	7:0	R	Number of Descriptors Tohost + FromHost excluding trickle descriptor	0x00
0x0090	0,1	NUM_OF_CHANNELS	7:0	R	Number of GBT or FULL mode Channels	0x00
0x00A0	0,1	CARD_TYPE	63:0	R	Card Type: - 709 (0x2c5): FLX709, VC709 - 710 (0x2c6): FLX710, HTG710 - 711 (0x2c7): FLX711, BNL711 - 712 (0x2c8): FLX712, BNL712 - 128 (0x080): FLX128, VCU128 - 180 (0x0B4): FLX180, VMK180 - 181 (0x0B5): FLX181, BNL181 - 182 (0x0B6): FLX182, BNL182	0x0000000000000000
0x00C0	0,1	GENERATE_GBT	0	R	1 when the GBT Wrapper is included in the design	0x0
0x00D0	0,1	OPTO_TRX_NUM	7:0	R	Number of optical transceivers in the design	0x00
0x00E0	0,1	GENERATE_TTC_EMU	1	R	1 when TTC emulator is generated	0x0

INCLUDE_EGROUPS						
0x0100	0,1	INCLUDE_EGROUP_0				
		TOHOST_32	9	R	ToHost EPATH32 is included in this EGROUP	0x0
		FROMHOST_02	8	R	FromHost EPATH02 is included in this EGROUP	0x0
		FROMHOST_04	7	R	FromHost EPATH04 is included in this EGROUP	0x0
		FROMHOST_08	6	R	FromHost EPATH8 is included in this EGROUP	0x0
		FROMHOST_HDLC	5	R	FromHost HDLC is included in this EGROUP	0x0
		TOHOST_02	4	R	ToHost EPATH02 is included in this EGROUP	0x0
		TOHOST_04	3	R	ToHost EPATH04 is included in this EGROUP	0x0
		TOHOST_08	2	R	ToHost EPATH08 is included in this EGROUP	0x0
		TOHOST_16	1	R	ToHost EPATH16 is included in this EGROUP	0x0
		TOHOST_HDLC	0	R	ToHost HDLC is included in this EGROUP	0x0
...						
0x0160	0,1	INCLUDE_EGROUP_6				
		TOHOST_32	9	R	ToHost EPATH32 is included in this EGROUP	0x0
		FROMHOST_02	8	R	FromHost EPATH02 is included in this EGROUP	0x0
		FROMHOST_04	7	R	FromHost EPATH04 is included in this EGROUP	0x0
		FROMHOST_08	6	R	FromHost EPATH8 is included in this EGROUP	0x0
		FROMHOST_HDLC	5	R	FromHost HDLC is included in this EGROUP	0x0
		TOHOST_02	4	R	ToHost EPATH02 is included in this EGROUP	0x0
		TOHOST_04	3	R	ToHost EPATH04 is included in this EGROUP	0x0
		TOHOST_08	2	R	ToHost EPATH08 is included in this EGROUP	0x0
		TOHOST_16	1	R	ToHost EPATH16 is included in this EGROUP	0x0
		TOHOST_HDLC	0	R	ToHost HDLC is included in this EGROUP	0x0
0x0170	0,1	WIDE_MODE	0	R	GBT is configured in Wide mode	0x0

0x0190	0,1	FIRMWARE_MODE	4:0	R	0: GBT mode 1: FULL-GBT 2: LTDB mode (GBT mode with only IC and TTC links) 3: FEI4 mode 4: ITK Pixel 5: ITK Strip 6: FELIG GBT 7: FULL mode emulator 8: FELIX_MROD mode 9: lpGBT mode 10: 25G Interlaken 11: FELIG LPGBT 12: HGTD_LUMI 13: BCMPRIME 14: FELIG_PIXEL	0x0
0x01A0	0,1	GTREFCLK_SOURCE	1:0	R	0: Transceiver reference Clock source from Si5345 1: Transceiver reference Clock source from Si5324 2: Transceiver reference Clock from internal BUFG (GREFCLK)	0x0
0x01B0	0,1	CR_GENERICS				
		XOFF_INCLUDED	2	R	Xoff bits (usually full mode) can be generated by the FromHost Central Router	0x0
		DIRECT_MODE_INCLUDED	1	R	Indicates that the Direct mode functionality was built in the Central Router	0x0
		FROM_HOST_INCLUDED	0	R	Indicates that the From Host path of the Central router was included in the design	0x0
0x01C0	0,1	BLOCKSIZE	15:0	R	Number of bytes in a block	0x0000
0x01D0	0,1	PCIE_ENDPOINT	0	R	Indicator of the PCIe endpoint on BNL71x cards with two endpoints. 0 or 1	0x0
0x01E0	0,1	CHUNK_TRAILER_32B	0	R	Indicator that the chunk trailer is in the new 32-bit format	0x0
0x01F0	0,1	NUMBER_OF_PCIE_ENDPOINTS	1:0	R	Number of PCIe endpoints on the card. The BNL71x cards have 2 endpoints	0x0
0x0200	0,1	AXI_STREAMS_TOHOST				
		IC_INDEX	23:16	R	The AXIs ID (EPath-ID) of the ToHost IC E-Link	0x00
		EC_INDEX	15:8	R	The AXIs ID (EPath-ID) of the ToHost EC E-Link	0x00
		NUMBER_OF_STREAMS	7:0	R	Total number of AXIs IDs (EPath-IDs) per physical link ToHost	0x00
0x0210	0,1	AXI_STREAMS_FROMHOST				
		IC_INDEX	23:16	R	The AXIs ID (EPath-ID) of the FromHost IC E-Link	0x00
		EC_INDEX	15:8	R	The AXIs ID (EPath-ID) of the FromHost EC E-Link	0x00
		NUMBER_OF_STREAMS	7:0	R	Total number of AXIs IDs (EPath-IDs) per physical link FromHost	0x00

0x0220	0,1	FROMHOST_DATA_FORMAT	1:0	R	0: The data format is as it was in phase1, supporting only multiples of 2 bytes 1: FromHost header uses a 5-bit length field as described in FLX-1355 2: FromHost header is 32-bit and the packet length is 256-bit (32 bytes) including the header FLX-1601 3: FromHost header is 32-bit and the packet length is 512-bit (32 bytes) including the header FLX-1601	0x0
0x0230	0,1	FULLMODE_HALFRATE	0	R	If set to 1 the FULL mode firmware is running at 4.8Gb instead of the default 9.6Gb	0x0
0x0240	0,1	SUPPORT_HDLC_DELAY	0	R	The HDLC encoders can offload a 1us delay as described in FLX-1826	0x0
CR To Host Controls And Monitors						
0x0800	0,1	TIMEOUT_CTRL				
		ENABLE	32	W	1 enables the timout trailer generation for ToHost mode	0x1
		TIMEOUT	31:0	W	Number of 40 MHz clock cycles after which a timeout occurs.	0xFFFFFFFF
0x0810	0,1	MAX_TIMEOUT	31:0	R	Maximum allowed timeout value	0x00000000
0x0820	0,1	CRTOHOST_FIFO_STATUS				
		CLEAR	any	T	Any write to this register clears the latched FULL flags	0x0
		FULL	47:24	R	Every bit represents the full flag of a channel FIFO	0x000000
		FULL_LATCHED	23:0	R	like FULL but a latched state, clear by writing to this register	0x000000
0x0830	0,1	CRTOHOST_DMA_DESCRIPTOR_1				
		WR_EN	any	T	Any write to this register assigns the DMA ID to the AXIS_ID set in CRTOHOST_DMA_DESCRIPTOR_2.AXIS_ID	0x0
		DESCR	2:0	W	Target descriptor	0x0
0x0840	0,1	CRTOHOST_DMA_DESCRIPTOR_2				
		DESCR_READ	13:11	R	Read back the value of the descriptor assigned to AXIS_ID	0x0
		AXIS_ID	10:0	W	ID of the AXI stream (E-Path ID) to associate with CRTOHOST_DMA_DESCRIPTOR_1.DESCR	0x00
CRTOHOST_INSTANT_TIMEOUT_ENA_GEN						
0x0850	0,1	CRTOHOST_INSTANT_TIMEOUT_ENA_00	41:0	W	Enable instant timeout after the first data arrives in CRToHost.	0x0000000000
...						
0x09C0	0,1	CRTOHOST_INSTANT_TIMEOUT_ENA_23	41:0	W	Enable instant timeout after the first data arrives in CRToHost.	0x0000000000
0x09D0	0,1	DISCARD_DATA_FOR_DESCR				

		FIFO_FULL	15:8	W	Discard data for a given DMA channel when Wupper FIFO is full, even if DMA is enabled	0x00
		DMA_DISABLED	7:0	W	Discard data for a given DMA channel when Wupper FIFO is full, and the descriptor is not enabled	0xFF
CR From Host Controls And Monitors						
0x1000	0,1	CRFROMHOST_FIFO_STATUS				
		CLEAR	any	T	Any write to this register clears the latched FULL flags	0x0
		FULL	47:24	R	Every bit represents the full flag of a channel FIFO	0x000000
		FULL_LATCHED	23:0	R	like FULL but a latched state, clear by writing to this register	0x000000
BROADCAST_ENABLE_GEN						
0x1010	0,1	BROADCAST_ENABLE_00	41:0	W	Enable path to be included in a broadcast message.	0x0000000000
		...				
0x1180	0,1	BROADCAST_ENABLE_23	41:0	W	Enable path to be included in a broadcast message.	0x0000000000
0x1190	0,1	CRFROMHOST_RESET	any	T	Central Router FromHost Controls and Monitors	0x0
Decoding Controls And Monitors						
PATH_HAS_STREAM_ID						
0x2000	0,1	LINK_00_HAS_STREAM_ID				
		EGROUP6	55:48	W	EPATH (Wide mode or lpGBT) is associated with a STREAM ID	0x00
		EGROUP5	47:40	W	EPATH (Wide mode or lpGBT) is associated with a STREAM ID	0x00
		EGROUP4	39:32	W	EPATH is associated with a STREAM ID	0x00
		EGROUP3	31:24	W	EPATH is associated with a STREAM ID	0x00
		EGROUP2	23:16	W	EPATH is associated with a STREAM ID	0x00
		EGROUP1	15:8	W	EPATH is associated with a STREAM ID	0x00
		EGROUP0	7:0	W	EPATH is associated with a STREAM ID, use only bit0 for FULL mode.	0x00
		...				
0x2170	0,1	LINK_23_HAS_STREAM_ID				
		EGROUP6	55:48	W	EPATH (Wide mode or lpGBT) is associated with a STREAM ID	0x00
		EGROUP5	47:40	W	EPATH (Wide mode or lpGBT) is associated with a STREAM ID	0x00
		EGROUP4	39:32	W	EPATH is associated with a STREAM ID	0x00
		EGROUP3	31:24	W	EPATH is associated with a STREAM ID	0x00
		EGROUP2	23:16	W	EPATH is associated with a STREAM ID	0x00

		EGROUP1	15:8	W	EPATH is associated with a STREAM ID	0x00
		EGROUP0	7:0	W	EPATH is associated with a STREAM ID, use only bit0 for FULL mode.	0x00
DECODING_LINK_STATUS_ARR						
0x2180	0,1	DECODING_LINK_ALIGNED_00	57:0	R	Every bit corresponds to an E-link on one (lp)GBT or FULL-mode frame. For FULL mode only bit 0 is used	0x00000000000000
...						
0x22F0	0,1	DECODING_LINK_ALIGNED_23	57:0	R	Every bit corresponds to an E-link on one (lp)GBT or FULL-mode frame. For FULL mode only bit 0 is used	0x00000000000000
DECODING_EGROUP_CTRL_GEN						
DECODING_EGROUP						
0x2300	0,1	DECODING_LINK00_EGROUP0_CTRL				
		ENABLE_TRUNCATION	59	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		EPATH_ALMOST_FULL	58:51	R	FIFO full indication	0x00
		REVERSE_ELINKS	50:43	W	enables bit reversing for the elink in the given epath	0x00
		PATH_ENCODING	42:11	W	Encoding for every EPATH, 4 bits per E-path 0: direct mode 1: 8b10b mode 2: HDLC mode 3: TTC 4: ITk Strips 8b10b 5: ITk Pixel 6: Endeavour 7-15: reserved	0x11111111
		EPATH_WIDTH	10:8	W	Width in bits of all EPATHS in an EGROUP 0:2, 1:4, 2:8, 3:16, 4:32	0x0
		EPATH_ENA	7:0	W	Enable bits per EPATH	0x00
...						
0x2360	0,1	DECODING_LINK00_EGROUP6_CTRL				
		ENABLE_TRUNCATION	59	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		EPATH_ALMOST_FULL	58:51	R	FIFO full indication	0x00
		REVERSE_ELINKS	50:43	W	enables bit reversing for the elink in the given epath	0x00

		PATH_ENCODING	42:11	W	Encoding for every EPATH, 4 bits per E-path 0: direct mode 1: 8b10b mode 2: HDLC mode 3: TTC 4: ITk Strips 8b10b 5: ITk Pixel 6: Endeavour 7-15: reserved	0x11111111
		EPATH_WIDTH	10:8	W	Width in bits of all EPATHS in an EGROUP 0:2, 1:4, 2:8, 3:16, 4:32	0x0
		EPATH_ENA	7:0	W	Enable bits per EPATH	0x00
		...				
DECODING_EGROUP						
0x27D0	0,1	DECODING_LINK11_EGROUP0_CTRL				
		ENABLE_TRUNCATION	59	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		EPATH_ALMOST_FULL	58:51	R	FIFO full indication	0x00
		REVERSE_ELINKS	50:43	W	enables bit reversing for the elink in the given epath	0x00
		PATH_ENCODING	42:11	W	Encoding for every EPATH, 4 bits per E-path 0: direct mode 1: 8b10b mode 2: HDLC mode 3: TTC 4: ITk Strips 8b10b 5: ITk Pixel 6: Endeavour 7-15: reserved	0x11111111
		EPATH_WIDTH	10:8	W	Width in bits of all EPATHS in an EGROUP 0:2, 1:4, 2:8, 3:16, 4:32	0x0
		EPATH_ENA	7:0	W	Enable bits per EPATH	0x00
		...				
0x2830	0,1	DECODING_LINK11_EGROUP6_CTRL				
		ENABLE_TRUNCATION	59	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		EPATH_ALMOST_FULL	58:51	R	FIFO full indication	0x00
		REVERSE_ELINKS	50:43	W	enables bit reversing for the elink in the given epath	0x00

		PATH_ENCODING	42:11	W	Encoding for every EPATH, 4 bits per E-path 0: direct mode 1: 8b10b mode 2: HDLC mode 3: TTC 4: ITk Strips 8b10b 5: ITk Pixel 6: Endeavour 7-15: reserved	0x11111111
		EPATH_WIDTH	10:8	W	Width in bits of all EPATHS in an EGROUP 0:2, 1:4, 2:8, 3:16, 4:32	0x0
		EPATH_ENA	7:0	W	Enable bits per EPATH	0x00
MINI_EGROUP_TOHOST_GEN						
0x2840	0,1	MINI_EGROUP_TOHOST_00				
		ENABLE_AUX_TRUNCATION	15	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		ENABLE_IC_TRUNCATION	14	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		ENABLE_EC_TRUNCATION	13	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		AUX_ALMOST_FULL	12	R	Indicator that the AUX path FIFO is almost full	0x0
		AUX_BIT_SWAPPING	11	W	0: two input bits of IC e-link are as documented, 1: two input bits are swapped	0x1
		AUX_ENABLE	10	W	Enables the AUX channel	0x1
		IC_ALMOST_FULL	9	R	Indicator that the IC path FIFO is almost full	0x0
		IC_BIT_SWAPPING	8	W	0: two input bits of IC e-link are as documented, 1: two input bits are swapped	0x1
		IC_ENABLE	7	W	Enables the IC channel	0x1
		EC_ALMOST_FULL	6	R	Indicator that the EC path FIFO is almost full	0x0
		EC_BIT_SWAPPING	5	W	0: two input bits of EC e-link are as documented, 1: two input bits are swapped	0x0
		EC_ENCODING	4:1	W	Configures encoding of the EC channel	0x2
		EC_ENABLE	0	W	Enables the EC channel	0x1
...						
0x29B0	0,1	MINI_EGROUP_TOHOST_23				
		ENABLE_AUX_TRUNCATION	15	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		ENABLE_IC_TRUNCATION	14	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		ENABLE_EC_TRUNCATION	13	W	Enable truncation mechanism in HDLC decoder for chunks > 12 bytes	0x0
		AUX_ALMOST_FULL	12	R	Indicator that the AUX path FIFO is almost full	0x0
		AUX_BIT_SWAPPING	11	W	0: two input bits of IC e-link are as documented, 1: two input bits are swapped	0x1

		AUX_ENABLE	10	W	Enables the AUX channel	0x1
		IC_ALMOST_FULL	9	R	Indicator that the IC path FIFO is almost full	0x0
		IC_BIT_SWAPPING	8	W	0: two input bits of IC e-link are as documented, 1: two input bits are swapped	0x1
		IC_ENABLE	7	W	Enables the IC channel	0x1
		EC_ALMOST_FULL	6	R	Indicator that the EC path FIFO is almost full	0x0
		EC_BIT_SWAPPING	5	W	0: two input bits of EC e-link are as documented, 1: two input bits are swapped	0x0
		EC_ENCODING	4:1	W	Configures encoding of the EC channel	0x2
		EC_ENABLE	0	W	Enables the EC channel	0x1
0x29C0	0,1	TTC_TOHOST_ENABLE	0	W	Enables the ToHost Mini Egroup in TTC mode	0x1
0x29D0	0,1	DECODING_REVERSE_10B	0	W	Reverse 10-bit word of elink data for 8b10b E-links 1: Receive 10-bit word in ToHost E-Paths, MSB first 0: Receive 10-bit word in ToHost E-Paths, LSB first	0x1
0x29E0	0,1	DECODING_ENDIANNESS_FULL_MODE	0	W	Specify the byte order in FULL mode 1: Big-endian 0: Little-endian	0x0
YARR_DEBUG_ALLEGROUP_TOHOST_GEN						
0x29F0	0,1	YARR_DEBUG_ALLEGROUP_TOHOST_00				
		REF_PACKET	63:32	W	Reference packet to be matched	0x02000000
		CNT_RX_PACKET	31:0	R	Count packets of a given value	0x00000000
...						
0x2AA0	0,1	YARR_DEBUG_ALLEGROUP_TOHOST_11				
		REF_PACKET	63:32	W	Reference packet to be matched	0x02000000
		CNT_RX_PACKET	31:0	R	Count packets of a given value	0x00000000
PATH_ERRORS						
0x2AB0	0,1	LINK_00_ERRORS				
		EGROUP6	27:24	R	Errors for Egroup6	0x0
		EGROUP5	23:20	R	Errors for Egroup5	0x0
		EGROUP4	19:16	R	Errors for Egroup4	0x0
		EGROUP3	15:12	R	Errors for Egroup3	0x0
		EGROUP2	11:8	R	Errors for Egroup2	0x0

		EGROUP1	7:4	R	Errors for Egroup1	0x0
		EGROUP0	3:0	R	Errors for Egroup0	0x0
...						
0x2C20	0,1	LINK_23_ERRORS				
		EGROUP6	27:24	R	Errors for Egroup6	0x0
		EGROUP5	23:20	R	Errors for Egroup5	0x0
		EGROUP4	19:16	R	Errors for Egroup4	0x0
		EGROUP3	15:12	R	Errors for Egroup3	0x0
		EGROUP2	11:8	R	Errors for Egroup2	0x0
		EGROUP1	7:4	R	Errors for Egroup1	0x0
		EGROUP0	3:0	R	Errors for Egroup0	0x0
0x2C30	0,1	INTERLAKEN_CONTROL				
		HEALTH_INTERFACE	12	R	Automatically detect the lane number in the interlaken descrambler	0x1
		PACKET_LENGTH	11:0	W	Lenth of an interlaken metaframe	0x7E8
INTERLAKEN_STATUS_GEN						
0x2C40	0,1	INTERLAKEN_LANE_00_STATUS				
		CLEAR_STATUS	any	T	Decoding block	0x0
		DECODER_ERROR_SYNC	8	R	Sticky error bit, clear with CLEAR_STATUS	0x0
		DESCRAMBLER_ERROR_BADSYNC	7	R	Sticky error bit, clear with CLEAR_STATUS	0x0
		DESCRAMBLER_ERROR_-STATEMISMATCH	6	R	Sticky error bit, clear with CLEAR_STATUS	0x0
		DESCRAMBLER_ERROR_NOSYNC	5	R	Sticky error bit, clear with CLEAR_STATUS	0x0
		BURST_CRC24_ERROR	4	R	Sticky CRC error bit, clear with CLEAR_STATUS	0x0
		META_CRC32_ERROR	3	R	Sticky CRC error bit, clear with CLEAR_STATUS	0x0
		HEALTH_LANE	2	R	Health bit for this lane	0x0
		DESCRAMBLER_ALIGNED	1	R	This channels descrambler is aligned	0x0
		DECODER_ALIGNED	0	R	This channels decoder is aligned	0x0
...						
0x2CF0	0,1	INTERLAKEN_LANE_11_STATUS				
		CLEAR_STATUS	any	T	Decoding block	0x0

		DECODER_ERROR_SYNC	8	R	Sticky error bit, clear with CLEAR_STATUS	0x0
		DESCRAMBLER_ERROR_BADSYNC	7	R	Sticky error bit, clear with CLEAR_STATUS	0x0
		DESCRAMBLER_ERROR_-STATEMISMATCH	6	R	Sticky error bit, clear with CLEAR_STATUS	0x0
		DESCRAMBLER_ERROR_NOSYNC	5	R	Sticky error bit, clear with CLEAR_STATUS	0x0
		BURST_CRC24_ERROR	4	R	Sticky CRC error bit, clear with CLEAR_STATUS	0x0
		META_CRC32_ERROR	3	R	Sticky CRC error bit, clear with CLEAR_STATUS	0x0
		HEALTH_LANE	2	R	Health bit for this lane	0x0
		DESCRAMBLER_ALIGNED	1	R	This channels descrambler is aligned	0x0
		DECODER_ALIGNED	0	R	This channels decoder is aligned	0x0
SUPER_CHUNK_FACTOR_GEN						
0x2DE0	0,1	SUPER_CHUNK_FACTOR_LINK_00	7:0	W	number of chunks glued together	0x01
...						
0x2E90	0,1	SUPER_CHUNK_FACTOR_LINK_11	7:0	W	number of chunks glued together	0x01
DECODING_LINK_CB_GEN						
0x2EA0	0,1	DECODING_LINK_00_CB				
		DESKEWED	61:4	R	Every bit corresponds to an E-link on one (lp)GBT frame. Register indicates whether the E-link has been de-skewed in the channel. E-link are grouped in a channel according to CBOPT	0x00000000000000
		CBOPT	3:0	W	Channel bonding option 0: no bonding 3: Bonding 0/1/2 3/4/5 other values: reserved	0x0
...						
0x2F50	0,1	DECODING_LINK_11_CB				
		DESKEWED	61:4	R	Every bit corresponds to an E-link on one (lp)GBT frame. Register indicates whether the E-link has been de-skewed in the channel. E-link are grouped in a channel according to CBOPT	0x00000000000000
		CBOPT	3:0	W	Channel bonding option 0: no bonding 3: Bonding 0/1/2 3/4/5 other values: reserved	0x0
0x2F60	0,1	DECODING_MASK64B66BKBLOCK	3:0	W	Mask User K-Block based on its block number (see sp011)	0xA

0x2F70	0,1	DECODING_DISEGROUP	6:0	W	Disable egroups for debugging purposes	0x0
0x2F80	0,1	FULLMODE_32B_SOP	0	W	When set to 1, use 32-bit 0x0000003C as start of chunk, otherwise only 8-bit 0x3C (FULL mode only)	0x0
0x2F90	0,1	DECODING_HGTD_ALTIROC	0	W	Set to 1 to use HGTD Altiroc K characters in the 8b10b decoders (LPGBT firmware mode)	0x0
0x2FA0	0,1	DECODING_HGTD_LUMI_CONF				
		DEBUG_DATASOURCE	36	W	enable local data source for debugging	0x0
		RAW_MODE	35	W	enable RAW mode (just forwarding 6b8b data)	0x0
		LHC_TURNS	34:25	W	number of LHC turns to aggregate	0x64
		TRIG_LAT	24:16	W	trigger latency for per-event luminosity	0x00
		SYNC_WORD	15:0	W	sync word for luminosity stream	0x4778
Encoding Controls And Monitors						
0x3000	0,1	ENCODING_REVERSE_10B	0	W	Reverse 10-bit word of elink data for 8b10b E-links. 1 MSB first, 0 LSB first	0x1
ENCODING_EGROUP_CTRL_GEN						
ENCODING_EGROUP						
0x3010	0,1	ENCODING_LINK00_EGROUP0_CTRL				
		ENABLE_DELAY	63	W	Enable inter-packet delay generation in HDLC encoder	0x0
		TTC_OPTION	62:59	W	Selects TTC bits sent to the E-link	0x0
		EPATH_ALMOST_FULL	58:51	R	Indiator that the EPATH FIFO is almost full	0x00
		REVERSE_ELINKS	50:43	W	enables bit reversing for the elink in the given epath	0x00
		EPATH_WIDTH	42:40	W	Width of the Elinks in the egroup 0: 2 bit 80 Mb/s 1: 4 bit 160 Mb/s 2: 8 bit 320 Mb/s	0x0
		PATH_ENCODING	39:8	W	Encoding for every EPATH, 4 bits per E-Path 0: No encoding 1: 8b10b mode 2: HDLC mode 3: ITk Strip LCB 4: ITk Pixel 5: Endeavour 6: reserved 7: reserved greater than 7: TTC mode, see firmware Phase 2 specification doc	0x11111111
		EPATH_ENA	7:0	W	Enable bits per E-PATH	0x00

...

0x3050	0,1		ENCODING_LINK00_EGROUP4_CTRL			
		ENABLE_DELAY	63	W	Enable inter-packet delay generation in HDLC encoder	0x0
		TTC_OPTION	62:59	W	Selects TTC bits sent to the E-link	0x0
		EPATH_ALMOST_FULL	58:51	R	Indiator that the EPATH FIFO is almost full	0x00
		REVERSE_ELINKS	50:43	W	enables bit reversing for the elink in the given epath	0x00
		EPATH_WIDTH	42:40	W	Width of the Elinks in the egroup 0: 2 bit 80 Mb/s 1: 4 bit 160 Mb/s 2: 8 bit 320 Mb/s	0x0
		PATH_ENCODING	39:8	W	Encoding for every EPATH, 4 bits per E-Path 0: No encoding 1: 8b10b mode 2: HDLC mode 3: ITk Strip LCB 4: ITk Pixel 5: Endeavour 6: reserved 7: reserved greater than 7: TTC mode, see firmware Phase 2 specification doc	0x11111111
		EPATH_ENA	7:0	W	Enable bits per E-PATH	0x00

...

ENCODING_EGROUP

0x3380	0,1		ENCODING_LINK11_EGROUP0_CTRL			
		ENABLE_DELAY	63	W	Enable inter-packet delay generation in HDLC encoder	0x0
		TTC_OPTION	62:59	W	Selects TTC bits sent to the E-link	0x0
		EPATH_ALMOST_FULL	58:51	R	Indiator that the EPATH FIFO is almost full	0x00
		REVERSE_ELINKS	50:43	W	enables bit reversing for the elink in the given epath	0x00
		EPATH_WIDTH	42:40	W	Width of the Elinks in the egroup 0: 2 bit 80 Mb/s 1: 4 bit 160 Mb/s 2: 8 bit 320 Mb/s	0x0

		PATH_ENCODING	39:8	W	Encoding for every EPATH, 4 bits per E-Path 0: No encoding 1: 8b10b mode 2: HDLC mode 3: ITk Strip LCB 4: ITk Pixel 5: Endeavour 6: reserved 7: reserved greater than 7: TTC mode, see firmware Phase 2 specification doc	0x11111111
		EPATH_ENA	7:0	W	Enable bits per E-PATH	0x00
		...				
0x33C0	0,1	ENCODING_LINK11_EGROUP4_CTRL				
		ENABLE_DELAY	63	W	Enable inter-packet delay generation in HDLC encoder	0x0
		TTC_OPTION	62:59	W	Selects TTC bits sent to the E-link	0x0
		EPATH_ALMOST_FULL	58:51	R	Indiator that the EPATH FIFO is almost full	0x00
		REVERSE_ELINKS	50:43	W	enables bit reversing for the elink in the given epath	0x00
		EPATH_WIDTH	42:40	W	Width of the Elinks in the egroup 0: 2 bit 80 Mb/s 1: 4 bit 160 Mb/s 2: 8 bit 320 Mb/s	0x0
		PATH_ENCODING	39:8	W	Encoding for every EPATH, 4 bits per E-Path 0: No encoding 1: 8b10b mode 2: HDLC mode 3: ITk Strip LCB 4: ITk Pixel 5: Endeavour 6: reserved 7: reserved greater than 7: TTC mode, see firmware Phase 2 specification doc	0x11111111
		EPATH_ENA	7:0	W	Enable bits per E-PATH	0x00
MINI_EGROUP_FROMHOST_GEN						
0x33D0	0,1	MINI_EGROUP_FROMHOST_00				
		ENABLE_DELAY	13	W	Enable inter-packet delay generation in HDLC encoder	0x0
		AUX_ALMOST_FULL	12	R	Indicator that the AUX Path FIFO is almost full	0x0
		AUX_BIT_SWAPPING	11	W	0: two input bits of AUX e-link are as documented, 1: two input bits are swapped	0x1
		AUX_ENABLE	10	W	Enables the AUX channel	0x1
		IC_ALMOST_FULL	9	R	Indicator that the IC Path FIFO is almost full	0x0

		IC_BIT_SWAPPING	8	W	0: two input bits of IC e-link are as documented, 1: two input bits are swapped	0x1
		IC_ENABLE	7	W	Enables the IC channel	0x1
		EC_ALMOST_FULL	6	R	Indicator that the EC Path FIFO is almost full	0x0
		EC_BIT_SWAPPING	5	W	0: two output bits of EC e-link are as documented, 1: two output bits are swapped	0x0
		EC_ENCODING	4:1	W	Configures encoding of the EC channel	0x2
		EC_ENABLE	0	W	Configures the FromHost Mini egroup	0x1
...						
0x3540	0,1	MINI_EGROUP_FROMHOST_23				
		ENABLE_DELAY	13	W	Enable inter-packet delay generation in HDLC encoder	0x0
		AUX_ALMOST_FULL	12	R	Indicator that the AUX Path FIFO is almost full	0x0
		AUX_BIT_SWAPPING	11	W	0: two input bits of AUX e-link are as documented, 1: two input bits are swapped	0x1
		AUX_ENABLE	10	W	Enables the AUX channel	0x1
		IC_ALMOST_FULL	9	R	Indicator that the IC Path FIFO is almost full	0x0
		IC_BIT_SWAPPING	8	W	0: two input bits of IC e-link are as documented, 1: two input bits are swapped	0x1
		IC_ENABLE	7	W	Enables the IC channel	0x1
		EC_ALMOST_FULL	6	R	Indicator that the EC Path FIFO is almost full	0x0
		EC_BIT_SWAPPING	5	W	0: two output bits of EC e-link are as documented, 1: two output bits are swapped	0x0
		EC_ENCODING	4:1	W	Configures encoding of the EC channel	0x2
		EC_ENABLE	0	W	Configures the FromHost Mini egroup	0x1
ENCODING_EGROUP_CTRL_FEI4_GEN						
ENCODING_EGROUP_FEI4						
0x3550	0,1	ENCODING_LINK00_EGROUP0_FEI4_CTRL				
		PHASE_DELAY1	11:9	W	phase delay of output data, with 320 Bb/s e-link 8 phases per BC	0x0
		MANCHESTER_ENABLE1	8	W	enable manchester encoding	0x0
		AUTOMATIC_MERGE_DISABLE1	7	W	Disable automatic merging	0x0
		TTC_SELECT1	6	W	TTC/FromHost select (if automatic merging is disabled)	0x0
		PHASE_DELAY0	5:3	W	phase delay of output data, with 320 Bb/s e-link 8 phases per BC	0x0
		MANCHESTER_ENABLE0	2	W	enable manchester encoding	0x0
		AUTOMATIC_MERGE_DISABLE0	1	W	Disable automatic merging	0x0

		TTC_SELECT0	0	W	TTC/FromHost select (if automatic merging is disabled)	0x0
		...				
0x3590	0,1	ENCODING_LINK00_EGROUP4_FEI4_CTRL				
		PHASE_DELAY1	11:9	W	phase delay of output data, with 320 Bb/s e-link 8 phases per BC	0x0
		MANCHESTER_ENABLE1	8	W	enable manchester encoding	0x0
		AUTOMATIC_MERGE_DISABLE1	7	W	Disable automatic merging	0x0
		TTC_SELECT1	6	W	TTC/FromHost select (if automatic merging is disabled)	0x0
		PHASE_DELAY0	5:3	W	phase delay of output data, with 320 Bb/s e-link 8 phases per BC	0x0
		MANCHESTER_ENABLE0	2	W	enable manchester encoding	0x0
		AUTOMATIC_MERGE_DISABLE0	1	W	Disable automatic merging	0x0
		TTC_SELECT0	0	W	TTC/FromHost select (if automatic merging is disabled)	0x0
		...				
ENCODING_EGROUP_FEI4						
0x38C0	0,1	ENCODING_LINK11_EGROUP0_FEI4_CTRL				
		PHASE_DELAY1	11:9	W	phase delay of output data, with 320 Bb/s e-link 8 phases per BC	0x0
		MANCHESTER_ENABLE1	8	W	enable manchester encoding	0x0
		AUTOMATIC_MERGE_DISABLE1	7	W	Disable automatic merging	0x0
		TTC_SELECT1	6	W	TTC/FromHost select (if automatic merging is disabled)	0x0
		PHASE_DELAY0	5:3	W	phase delay of output data, with 320 Bb/s e-link 8 phases per BC	0x0
		MANCHESTER_ENABLE0	2	W	enable manchester encoding	0x0
		AUTOMATIC_MERGE_DISABLE0	1	W	Disable automatic merging	0x0
		TTC_SELECT0	0	W	TTC/FromHost select (if automatic merging is disabled)	0x0
		...				
0x3900	0,1	ENCODING_LINK11_EGROUP4_FEI4_CTRL				
		PHASE_DELAY1	11:9	W	phase delay of output data, with 320 Bb/s e-link 8 phases per BC	0x0
		MANCHESTER_ENABLE1	8	W	enable manchester encoding	0x0
		AUTOMATIC_MERGE_DISABLE1	7	W	Disable automatic merging	0x0
		TTC_SELECT1	6	W	TTC/FromHost select (if automatic merging is disabled)	0x0
		PHASE_DELAY0	5:3	W	phase delay of output data, with 320 Bb/s e-link 8 phases per BC	0x0
		MANCHESTER_ENABLE0	2	W	enable manchester encoding	0x0

		AUTOMATIC_MERGE_DISABLE0	1	W	Disable automatic merging	0x0
		TTC_SELECT0	0	W	TTC/FromHost select (if automatic merging is disabled)	0x0
YARR_DEBUG_ALLEGROUP_FROMHOST_GEN						
0x3910	0,1	YARR_DEBUG_ALLEGROUP_FROMHOST1_00				
		RD53A_AZ_EN	48	W	Auto zeroing module enable	0x0
		CNT_TRIG_CMD	47:16	R	Number of issued triggers via cmd	0x00000000
		ERR_GENCALTRIG_DLY	15:8	R	Number of mismatches between CNT_GENCALTRIG_DLY and REF_DLY_GENCALTRIG	0x00
		REF_DLY_GENCALTRIG	7:0	W	Reference distance between GenCal and First Trigger	0x0F
0x3920	0,1	YARR_DEBUG_ALLEGROUP_FROMHOST2_00				
		CNT_CMD	47:16	R	Number of issued commands	0x00000000
		REF_CMD	15:0	W	Cmd type to be counted. See RD53 Manual for list of allowed commands	0x6666
...						
0x3A70	0,1	YARR_DEBUG_ALLEGROUP_FROMHOST1_11				
		RD53A_AZ_EN	48	W	Auto zeroing module enable	0x0
		CNT_TRIG_CMD	47:16	R	Number of issued triggers via cmd	0x00000000
		ERR_GENCALTRIG_DLY	15:8	R	Number of mismatches between CNT_GENCALTRIG_DLY and REF_DLY_GENCALTRIG	0x00
		REF_DLY_GENCALTRIG	7:0	W	Reference distance between GenCal and First Trigger	0x0F
0x3A80	0,1	YARR_DEBUG_ALLEGROUP_FROMHOST2_11				
		CNT_CMD	47:16	R	Number of issued commands	0x00000000
		REF_CMD	15:0	W	Cmd type to be counted. See RD53 Manual for list of allowed commands	0x6666
0x3A90	0,1	YARR_FROMHOST_CALTRIGSEQ_WE	0	W	enable to store CalPulse+Trigger Sequence into memory	0x0
0x3AA0	0,1	YARR_FROMHOST_CALTRIGSEQ_-WRDATA	15:0	W	CalPulse+Trigger Sequence to be stored in memory	0x0000
0x3AB0	0,1	YARR_FROMHOST_CALTRIGSEQ_-WRADDR	4:0	W	memory address to store CalPulse+Trigger Sequence	0x0
0x3AC0	0,1	HGTD_ALTIROC_FASTCMD				
		ALTIROC3_IDLE	14	W	0 for ALTIROC2 10101100, 1 for ALTIROC3 11110000	0x0
		USE_CAL	13	W	When set to 1, CAL will be sent on L1A, then after TRIG_DELAY BC clocks a TRIGGER. When 0, TRIGGER will be sent on L1A.	0x1
		SYNCLUMI	12	W	Set to 1 to trigger a SYNCLUMI command, rising edge of this bit. Clear in software	0x0

		GBRST	11	W	Set to 1 to trigger a GBRST command, rising edge of this bit. Clear in software	0x0
		TRIG_DELAY	10:0	W	Number of BC clocks between CAL and TRIGGER command if USE_CAL is set to 1	0x05
Frontend Emulator Controls And Monitors						
0x4000	0, 1	FE_EMU_ENA				
		EMU_TOFRONTEND	1	W	Enable GBT dummy emulator ToFrontEnd	0x0
		EMU_TOHOST	0	W	Enable GBT dummy emulator ToHost	0x0
0x4010	0, 1	FE_EMU_CONFIG				
		WE	54:47	W	write enable array, every bit is one emulator RAM block	0x00
		WRADDR	46:33	W	write address bus	0x000
		WRDATA	32:0	W	write data bus	0x00000000
0x4020	0, 1	FE_EMU_READ				
		SEL	35:33	W	Select ramblock to read back	0x0
		DATA	32:0	R	Read back ramblock at FE_EMU_CONFIG.WRADDR	0x00000000
0x4030	0, 1	FE_EMU_LOGIC				
		L1A_TRIGGERED	33	W	1 Send a chunk on every L1A, 0 use the IDLES to determine the rate	0x0
		ENA	32	W	Enable logic based FrontEnd emulator, instead of RAM based.	0x0
		IDLES	31:16	W	Number of IDLE bytes between chunks.	0x0000
		CHUNK_LENGTH	15:0	W	Chunk length in bytes	0x0000
DECODING_BCM_PRIME_L1 A_CONTROLS_GEN						
0x4200	0, 1	DECODING_BCM_PRIME_LINK_00_L1A				
		DELAY	9:5	W	The data in fiber is delayed N clock cycles to match with TTC L1A	0x5
		WINDOW	4:0	W	The L1A signal is extended to cover multiple BCID's	0x5
...						

0x4370	0, 1	DECODING_BCM_PRIME_LINK_23_L1A				
		DELAY	9:5	W	The data in fiber is delayed N clock cycles to match with TTC L1A	0x5
		WINDOW	4:0	W	The L1A signal is extended to cover multiple BCID's	0x5
0x4380	0, 1	DECODING_BCM_PRIME_ONLY_L1A	0	W	If enabled, the BCM_PRIME firmware, will only readout data when an L1A is sent.	0x0
0x4390	0, 1	DECODING_BCM_PRIME_EMU_BCID	0	W	If enabled, the BCM_PRIME firmware will use internally generated BCIDs instead of the TTC one.	0x0
0x43A0	0, 1	DECODING_BCM_PRIME_PUBLISH_-ZEROS	0	W	If enabled, the BCM_PRIME firmware publish empty data-events if they are matched with L1A	0x0
Link Wrapper Controls						
0x5000	0	LINK_FULLMODE_LTI	23:0	W	Set to 1 to enable LTI format TTC distribution (8b10b at 9.6Gb) in the FULLMODE flavour, one bit per channel. Set to 0 for 4.8Gb GBT distribution	0x000000
0x5400	0	GBT_CHANNEL_DISABLE	47:0	W	Disable selected lpGBT, GBT or FULL mode channel	0x000000000000
0x5410	0	GBT_GENERAL_CTRL	63:0	W	Alignment chk reset (not self clearing)	0x0000000000000000
0x5420	0	GBT_MODE_CTRL				
		RX_ALIGN_TB_SW	2	W	RX_ALIGN_TB_SW	0x0
		RX_ALIGN_SW	1	W	RX_ALIGN_SW	0x0
		DESMUX_USE_SW	0	W	DESMUX_USE_SW	0x0
0x5480	0	GBT_RXSLIDE_SELECT	47:0	W	RxSlide select [47:0]	0x000000000000
0x5490	0	GBT_RXSLIDE_MANUAL	47:0	W	RxSlide select [47:0]	0x000000000000
0x54A0	0	GBT_TXUSRRDY	47:0	W	TxUsrRdy [47:0]	0xFFFFFFFFFFFF
0x54B0	0	GBT_RXUSRRDY	47:0	W	RxUsrRdy [47:0]	0xFFFFFFFFFFFF
0x54C0	0	GBT_SOFT_RESET	47:0	W	SOFT_RESET [47:0]	0x000000000000
0x54D0	0	GBT_GTTX_RESET	47:0	W	GTTX_RESET [47:0]	0x000000000000
0x54E0	0	GBT_GTRX_RESET	47:0	W	GTRX_RESET [47:0]	0x000000000000
0x54F0	0	GBT_PLL_RESET				
		QPLL_RESET	59:48	W	QPLL_RESET [11:0]	0x000
		CPLL_RESET	47:0	W	CPLL_RESET [47:0]	0x000000000000

0x5500	0	GBT_SOFT_TX_RESET				
		RESET_ALL	59:48	W	SOFT_TX_RESET_ALL [11:0]	0x000
		RESET_GT	47:0	W	SOFT_TX_RESET_GT [47:0]	0x000000000000
0x5510	0	GBT_SOFT_RX_RESET				
		RESET_ALL	59:48	W	SOFT_TX_RESET_ALL [11:0]	0x000
		RESET_GT	47:0	W	SOFT_TX_RESET_GT [47:0]	0x000000000000
0x5520	0	GBT_ODD_EVEN	47:0	W	OddEven [47:0]	0x000000000000
0x5530	0	GBT_TOPBOT	47:0	W	TopBot [47:0]	0x000000000000
0x5540	0	GBT_TX_TC_DLY_VALUE1	47:0	W	TX_TC_DLY_VALUE [47:0]	0x333333333333
0x5550	0	GBT_TX_TC_DLY_VALUE2	47:0	W	TX_TC_DLY_VALUE [95:48]	0x333333333333
0x5560	0	GBT_TX_TC_DLY_VALUE3	47:0	W	TX_TC_DLY_VALUE [143:96]	0x333333333333
0x5570	0	GBT_TX_TC_DLY_VALUE4	47:0	W	TX_TC_DLY_VALUE [191:144]	0x333333333333
0x5580	0	GBT_DATA_TXFORMAT1	47:0	W	DATA_TXFORMAT [47:0]	0x000000000000
0x5590	0	GBT_DATA_TXFORMAT2	47:0	W	DATA_TXFORMAT [95:48]	0x000000000000
0x55A0	0	GBT_DATA_RXFORMAT1	47:0	W	DATA_RXFORMAT [47:0]	0x000000000000
0x55B0	0	GBT_DATA_RXFORMAT2	47:0	W	DATA_RXFORMAT [95:0]	0x000000000000
0x55C0	0	GBT_TX_RESET	47:0	W	TX Logic reset [47:0]	0x000000000000
0x55D0	0	GBT_RX_RESET	47:0	W	RX Logic reset [47:0]	0x000000000000
0x55E0	0	GBT_TX_TC_METHOD	47:0	W	TX time domain crossing method [47:0]	0x000000000000
0x55F0	0	GBT_OUTMUX_SEL	47:0	W	Descrambler output MUX selection [47:0]	0x000000000000
0x5600	0	GBT_TC_EDGE	47:0	W	Sampling edge selection for TX domain crossing [47:0]	0x000000000000
0x5610	0	GBT_TXPOLARITY	47:0	W	0: default polarity 1: reversed polarity for transmitter of GTH channels	0x000000000000
0x5620	0	GBT_RXPOLARITY	47:0	W	0: default polarity 1: reversed polarity for the receiver of the GTH channels	0x000000000000

0x5630	0	GTH_LOOPBACK_CONTROL	2:0	W	Controls loopback for loopback: read UG476 for the details. NOTE: the TXBUFFER is disabled, near end PCS loopback is not supported. 000: Normal operation 001: Near-End PCS Loopback 010: Near-End PMA Loopback 011: Reserved 100: Far-End PMA Loopback 101: Reserved 110: Far-End PCS Loopback	0x0
0x5640	0	LPGBT_FEC	47:0	W	0: FEC5 1: FEC12	0x000000000000
0x5650	0	LPGBT_DATARATE	47:0	W	0: 10.24 Gbps 1: 5.12 Gbps	0x000000000000
0x5700	0	GBT_TOHOST_FANOUT				
		LOCK	48	W	Locks this particular register. If set prevents software from touching it.	0x0
		SEL	47:0	W	ToHost FanOut/Selector. Every bitfield is a channel: 1 : GBT_EMU, select GBT Emulator for a specific CentralRouter channel 0 : GBT_WRAP, select real GBT link for a specific CentralRouter channel	0x000000000000
0x5710	0	GBT_TOFRONTEND_FANOUT				
		LOCK	48	W	Locks this particular register. If set prevents software from touching it.	0x0
		SEL	47:0	W	ToFrontEnd FanOut/Selector. Every bitfield is a channel: 1 : GBT_EMU, select GBT Emulator for a specific GBT link 0 : TTC_DEC, select CentralRouter data (including TTC) for a specific GBT link	0x000000000000
0x5720	0	FULLMODE_AUTO_RX_RESET				
		ENABLE	32	W	Enable the Automatic RX Reset mechanism	0x1
		TIMEOUT	31:0	W	Number of 40 MHz clock cycles until an unaligned link results in a reset pulse	0x00100000
TCLINK_CNTRL_GEN						
0x5730	0	TCLINK_CONTROL_00				
		OFFSET_ERROR	63:16	W	Error-offset for phase-control Recommended to freeze with an initial value read	0x000000000000
		CLOSE_LOOP	15	W	Close TCLink loop (enables compensation)	0x0
		TX_PI_PHASE_CALIB	14:8	W	UI alignment Tx PI calibrated phase	0x0
		TX_UI_ALIGN_CALIB	7	W	UI alignment Tx PI activate	0x0
		TX_FINE_REALIGN	6	W	Repeats fine alignment procedure	0x0

		PS_STROBE	5	W	Shifts phase of transmitter serial data	0x0
		PS_INC_NDEC	4	W	Shifts phase of transmitter serial data	0x0
		MASTER_MGT_RX_READY	3	W	MGT rx is ready (used as reset)	0x0
...						
0x58A0	0	TCLINK_CONTROL_23				
		OFFSET_ERROR	63:16	W	Error-offset for phase-control Recommended to freeze with an initial value read	0x000000000000
		CLOSE_LOOP	15	W	Close TCLink loop (enables compensation)	0x0
		TX_PI_PHASE_CALIB	14:8	W	UI alignment Tx PI calibrated phase	0x0
		TX_UI_ALIGN_CALIB	7	W	UI alignment Tx PI activate	0x0
		TX_FINE_REALIGN	6	W	Repeats fine alignment procedure	0x0
		PS_STROBE	5	W	Shifts phase of transmitter serial data	0x0
		PS_INC_NDEC	4	W	Shifts phase of transmitter serial data	0x0
		MASTER_MGT_RX_READY	3	W	MGT rx is ready (used as reset)	0x0
Link Wrapper Monitors						
0x6600	0	GBT_VERSION				
		DATE	63:48	R	Date	0x0000
		GBT_VERSION	47:32	R	GBT Version	0x0000
		GTH_IP_VERSION	31:16	R	GTH IP Version	0x0000
		RESERVED	15:3	R	Reserved	0x000
		GTHREFCLK_SEL	2	R	GTHREFCLK SEL	0x0
		RX_CLK_SEL	1	R	RX CLK SEL	0x0
		PLL_SEL	0	R	PLL SEL	0x0
0x6680	0	GBT_TXRESET_DONE	47:0	R	TX Reset done [47:0]	0x000000000000
0x6690	0	GBT_RXRESET_DONE	47:0	R	RX Reset done [47:0]	0x000000000000
0x66A0	0	GBT_TXFSMRESET_DONE	47:0	R	TX FSM Reset done [47:0]	0x000000000000
0x66B0	0	GBT_RXFSMRESET_DONE	47:0	R	RX FSM Reset done [47:0]	0x000000000000
0x66C0	0	GBT_CPLL_FBCLK_LOST	47:0	R	CPLL FBCLK LOST [47:0]	0x000000000000
0x66D0	0	GBT_PLL_LOCK				
		QPLL_LOCK	59:48	R	QPLL LOCK [11:0]	0x000

		CPLL_LOCK	47:0	R	CPLL LOCK [47:0]	0x000000000000
0x66E0	0	GBT_RXCDR_LOCK	47:0	R	RX CDR LOCK [47:0]	0x000000000000
0x66F0	0	GBT_CLK_SAMPLED	47:0	R	clk sampled [47:0]	0x000000000000
0x6700	0	GBT_RX_IS_HEADER	47:0	R	RX IS HEADER [47:0]	0x000000000000
0x6710	0	GBT_RX_IS_DATA	47:0	R	RX IS DATA [47:0]	0x000000000000
0x6720	0	GBT_RX_HEADER_FOUND	47:0	R	RX HEADER FOUND [47:0]	0x000000000000
0x6730	0	GBT_ALIGNMENT_DONE	47:0	R	RX ALIGNMENT DONE [47:0]	0x000000000000
0x6740	0	GBT_OUT_MUX_STATUS	47:0	R	GBT output mux status [47:0]	0x000000000000
0x6750	0	GBT_ERROR	47:0	R	Error flags [47:0]	0x000000000000
0x6760	0	GBT_GBT_TOPBOT_C	47:0	R	TopBot_c [47:0]	0x000000000000
0x6800	0	GBT_FM_RX_DISP_ERROR1	47:0	R	Rx disparity error [47:0]	0x000000000000
0x6810	0	GBT_FM_RX_DISP_ERROR2	47:0	R	Rx disparity error [96:48]	0x000000000000
0x6820	0	GBT_FM_RX_NOTINTABLE1	47:0	R	Rx not in table [47:0]	0x000000000000
0x6830	0	GBT_FM_RX_NOTINTABLE2	47:0	R	Rx not in table [96:48]	0x000000000000
GT_FEC_ERR_CNT_GEN						
0x6840	0	GT_FEC_ERR_CNT_00	31:0	R	Counts the number of FEC errors in the given channel.	0x00000000
...						
0x69B0	0	GT_FEC_ERR_CNT_23	31:0	R	Counts the number of FEC errors in the given channel.	0x00000000
GT_AUTO_RX_RESET_CNT_GEN						
0x69C0	0	GT_AUTO_RX_RESET_CNT_00				
		CLEAR	any	T	Any write to this register clears the counter value	0x0
		VALUE	31:0	R	Counts the number of AUTO RX RESET events that happend on the FULLMODE, GBT or lpGBT link	0x00000000
...						
0x6B30	0	GT_AUTO_RX_RESET_CNT_23				
		CLEAR	any	T	Any write to this register clears the counter value	0x0
		VALUE	31:0	R	Counts the number of AUTO RX RESET events that happend on the FULLMODE, GBT or lpGBT link	0x00000000
TCLINK_MON_GEN						
0x6B40	0	TCLINK_MONITOR_1_00				

		ERROR_CONTROLLER	62:15	R	Error-signal for controller Signed complement 2 number.	0x000000000000
		LOOP_CLOSED	14	R	TCLink loop is closed (compensation is enabled)	0x0
		TX_ALIGNED	13	R	Transmitter alignment procedure finished Use as reset for transmitter user logic	0x0
		PS_DONE	12	R	Phase shift is done	0x0
		TX_PI_PHASE	11:5	R	Tx PI phase after alignment	0x0
0x6B50	0	TCLINK_MONITOR_2_00				
		PHASE_DETECTOR	63:32	R	Phase detector response	0x00000000
		TX_FIFO_FILL_PD	31:0	R	Phase detector current value	0x00000000
0x6B60	0	TCLINK_MONITOR_3_00				
		LOOP_NOT_CLOSED_REASON	58:54	R	Reason why the TCLink loop is not closed	0x0
		PHASE_ACC	53:38	R	phase accumulated output (integrated output)	0x0000
		OPERATION_ERROR	37	R	error output indicating that a clk_en_i pulse has arrived before the done_i signal arrived from the previous strobe_o request	0x0
		DEBUG_TESTER_ADDR_READ	36:27	W	read address for reading stocked TCLink phase accumulated results	0x00
		DEBUG_TESTER_DATA_READ	26:11	R	data of stocked TCLink phase accumulated results	0x0000
		PS_PHASE_STEP	10:7	R	number of units to shift the phase of the receiver clock	0x0
...						
0x6F90	0	TCLINK_MONITOR_1_23				
		ERROR_CONTROLLER	62:15	R	Error-signal for controller Signed complement 2 number.	0x000000000000
		LOOP_CLOSED	14	R	TCLink loop is closed (compensation is enabled)	0x0
		TX_ALIGNED	13	R	Transmitter alignment procedure finished Use as reset for transmitter user logic	0x0
		PS_DONE	12	R	Phase shift is done	0x0
		TX_PI_PHASE	11:5	R	Tx PI phase after alignment	0x0
0x6FA0	0	TCLINK_MONITOR_2_23				
		PHASE_DETECTOR	63:32	R	Phase detector response	0x00000000
		TX_FIFO_FILL_PD	31:0	R	Phase detector current value	0x00000000
0x6FB0	0	TCLINK_MONITOR_3_23				
		LOOP_NOT_CLOSED_REASON	58:54	R	Reason why the TCLink loop is not closed	0x0
		PHASE_ACC	53:38	R	phase accumulated output (integrated output)	0x0000

		OPERATION_ERROR	37	R	error output indicating that a clk_en_i pulse has arrived before the done_i signal arrived from the previous strobe_o request	0x0
		DEBUG_TESTER_ADDR_READ	36:27	W	read address for reading stocked TCLink phase accumulated results	0x00
		DEBUG_TESTER_DATA_READ	26:11	R	data of stocked TCLink phase accumulated results	0x0000
		PS_PHASE_STEP	10:7	R	number of units to shift the phase of the receiver clock	0x0
0x6FC0	0	GBT_PLL_LOL_LATCHED				
		CLEAR	any	W	Any write to this bitfield clears the latched LOL bits	0x0
		QPLL_LOL_LATCHED	59:48	R	Asserted when CPLL lock is lost, clear by writing to CLEAR	0x000
		CPLL_LOL_LATCHED	47:0	R	Asserted when CPLL lock is lost, clear by writing to CLEAR	0x000000000000
0x6FD0	0	GBT_ALIGNMENT_LOST				
		CLEAR	any	T	Any write to this bitfield clears the latched ALIGNMENT_LOST bits	0x0
		ALIGNMENT_LOST	47:0	R	Asserted when GBT_ALIGNMENT_DONE bit is 0, clear by writing to CLEAR	0x000000000000
TTCBUSY Controls And Monitors						
TTC_DEC_CTRLMON						
0x7000	0	TTC_DEC_CTRL				
		B_CHAN_DELAY	30:27	W	Number of BC to delay the L1A distribution to the frontends	0x0
		BCID_ONBCR	26:15	W	BCID is set to this value when BCR arrives	0x000
		BUSY_OUTPUT_STATUS	14	R	Actual status of the BUSY LEMO output signal	0x0
		ECR_BCR_SWAP	13	W	ECR and BCR signals are swapped at the output of the TTC decoder (needed only for LAr TTC)	0x0
		BUSY_OUTPUT_INHIBIT	12	W	forces the Busy LEMO output to BUSY-OFF	0x0
		TOHOST_RST	11	W	reset toHost in ttc decoder	0x0
		TT_BCH_EN	10	W	trigger type enable / disable for TTC-ToHost	0x1
		XL1ID_SW	9:2	W	set XL1ID value, the value to be set by XL1ID_RST signal	0x00
		XL1ID_RST	1	W	giving a trigger signal to reset XL1ID value	0x0
		MASTER_BUSY	0	W	L1A trigger throttling	0x0
0x7010	0	TTC_DEC_MON				
		TH_FF_COUNT	15:5	R	ToHostData Fifo counts	0x00
		TH_FF_FULL	4	R	ToHostData Fifo status 1:full 0:not full	0x0
		TH_FF_EMPTY	3	R	ToHostData Fifo status 1:empty 0:not empty	0x0

		TTC_BIT_ERR	2:0	R	double bit, single bit and comm error in TTC data	0x0
TTC_BUSY_ACCEPTED_G						
0x7020	0,1	TTC_BUSY_ACCEPTED00	56:0	R	busy has been asserted by the given ELINK. Reset by writing to TTC_BUSY_CLEAR	0x00000000000000
...						
0x7190	0,1	TTC_BUSY_ACCEPTED23	56:0	R	busy has been asserted by the given ELINK. Reset by writing to TTC_BUSY_CLEAR	0x00000000000000
0x71A0	0	TTC_EMU				
		FULL	2	R	TTC Emulator memory full indication	0x0
		SEL	1	W	Select TTC data source 1 TTC Emu | 0 TTC Decoder	0x0
		ENA	0	W	Clear to load into the TTC emulator's memory the required sequence, Set to run the TTC emulator sequence	0x0
0x71B0	0	TTC_DELAY	3:0	W	Controls the TTC Fanout delay value, in 25ns (1BC) units	0x0
0x74B0	0	TTC_BUSY_TIMING_CTRL				
		PRESCALE	51:32	W	Prescales the 40MHz clock to create an internal slow clock	0x0000F
		BUSY_WIDTH	31:16	W	Minimum number of 40MHz clocks that the busy is asserted	0x000F
		LIMIT_TIME	15:0	W	Number of prescaled clocks a given busy must be asserted before it is recognized	0x000F
0x74C0	0	TTC_BUSY_CLEAR	any	T	clears the latching busy bits in TTC_BUSY_ACCEPTED	0x0
0x74D0	0	TTC_EMU_CONTROL				
		BUSY_IN_ENABLE	33	W	Enable internal BUSY input to stop L1A on BUSY	0x1
		BROADCAST	32:27	W	Broadcast data	0x0
		ECR	26	W	Event counter reset	0x0
		BCR	25	W	Bunch counter reset	0x0
		L1A	24	W	Level 1 Accept	0x0
0x74E0	0	TTC_EMU_L1A_PERIOD	31:0	W	L1A period in BC. 0 means manual L1A with TTC_EMU_CONTROL.L1A	0x00000000
0x74F0	0	TTC_EMU_ECR_PERIOD	31:0	W	ECR period in BC. 0 means manual ECR with TTC_EMU_CONTROL.ECR	0x00000000
0x7500	0	TTC_EMU_BCR_PERIOD	31:0	W	BCR period in BC. 0 means manual BCR with TTC_EMU_CONTROL.BCR	0x00000DEC
0x7510	0	TTC_EMU_LONG_CHANNEL_DATA	31:0	W	Long channel data for the TTC emulator	0x00000000
0x7520	0	TTC_EMU_RESET	any	T	Any write to this register resets the TTC Emulator to the default state.	0x0
0x7530	0	TTC_L1ID_MONITOR	31:0	R	Monitor L1ID and XL1ID.	0x00000000
0x7540	0	TTC_ECR_MONITOR				

		CLEAR	any	T	Counts the number of ECRs received from the TTC system, any write to this register clears the counter	0x0
		VALUE	31:0	R	Counts the number of ECRs received from the TTC system, any write to this register clears the counter	0x00000000
0x7550	0	TTC_TTYPE_MONITOR				
		CLEAR	any	T	Counts the number of TType received from the TTC system, any write to this register clears the counter	0x0
		VALUE	31:0	R	Counts the number of TType received from the TTC system, any write to this register clears the counter	0x00000000
0x7560	0	TTC_BCR_PERIODICITY_MONITOR				
		CLEAR	any	T	Counts the number of times the BCR period does not match 3564, any write to this register clears the counter	0x0
		VALUE	31:0	R	Counts the number of times the BCR period does not match 3564, any write to this register clears the counter	0x00000000
0x7570	0	TTC_BCR_COUNTER				
		CLEAR	any	T	Counts the number of times BCR is issued, any write to this register clears the counter	0x0
		VALUE	31:0	R	Counts the number of times BCR is issued, any write to this register clears the counter	0x00000000
0x7580	0	TTC_EMU_TP_DELAY	31:0	W	Number of BC that the testpulse should be sent before the L1A, 0 means no test pulse is sent	0x00000040
0x7590	0	TTC_L1A_DELAY	5:0	W	In Phase1 the L0A bit is generated from L1A, but with a variable delay between 0 and 63 BC cycles from L0A to L1A	0x0
0x75A0	0	TTC_CDRLOCK_MONITOR				
		CLEAR	any	T	Clears the latching cdrlock, LOL and LOS bitfields	0x0
		CDRLOCK_LOST	5	R	asserted when CDRLOCKED has been 0, Clear by writing to CLEAR bitfield	0x0
		CDRLOCKED	4	R	Set to 1 if the clock can be successfully recovered from the TTC signal	0x0
		ADN_LOL_LATCHED	3	R	Latched Loss of lock from ADN2814, Clear by writing to CLEAR bitfield	0x0
		ADN_LOS_LATCHED	2	R	Latched Loss of signal from ADN2814, Clear by writing to CLEAR bitfield	0x0
		ADN_LOL	1	R	Loss of lock from ADN2814	0x0
		ADN_LOS	0	R	Loss of signal from ADN2814	0x0
XOFF_BUSY Controls And Monitors						
0x8000	0, 1	XOFF_FM_CH_FIFO_THRESH_LOW	3:0	W	Controls the low threshold of the channel fifo in FULL mode on which an Xon will be asserted, bitfields control 4 MSB	0xB

0x8010	0, 1	XOFF_FM_CH_FIFO_THRESH_HIGH	3:0	W	Controls the high threshold of the channel fifo in FULL mode on which an Xoff will be asserted, bitfields control 4 MSB	0xB
0x8020	0, 1	XOFF_FM_LOW_THRESH_CROSSED	23:0	R	FIFO filled beyond the low threshold, 1 bit per channel	0x000000
0x8030	0, 1	XOFF_FM_HIGH_THRESH				
		CLEAR_LATCH	any	T	Writing this register will clear all CROSS_LATCHED bits	0x0
		CROSS_LATCHED	47:24	R	FIFO filled beyond the high threshold, 1 latch bit per channel	0x000000
		CROSSED	23:0	R	FIFO filled beyond the high threshold, 1 bit per channel	0x000000
0x8040	0, 1	XOFF_FM_SOFT_XOFF	23:0	W	Set any bit in this register to assert XOFF for the given channel, clearing bits will assert XON	0x000000
0x8050	0, 1	XOFF_ENABLE	23:0	W	Enable XOFF assertion (To Frontend) in case the FULL mode CH FIFO gets beyond thresholds. One bit per channel	0x000000
0x8060	0, 1	DMA_BUSY_STATUS				
		CLEAR_LATCH	any	T	Any write to this register clears TOHOST_BUSY_LATCHED	0x0
		ENABLE	4	W	Enable the DMA buffer on the server as a source of busy	0x0
		TOHOST_BUSY_LATCHED	3	R	A tohost descriptor has passed BUSY_THRESHOLD_ASSERT in the past, busy flag was set	0x0
		TOHOST_BUSY	0	R	A tohost descriptor passed BUSY_THRESHOLD_ASSERT, busy flag set	0x0
0x8070	0, 1	FM_BUSY_CHANNEL_STATUS				
		CLEAR_LATCH	any	T	Any write to this register will clear the BUSY_LATCHED bits	0x0
		BUSY_LATCHED	47:24	R	one Indicates that the given FULL mode channel has received BUSY-ON	0x000000
		BUSY	23:0	R	one Indicates that the given FULL mode channel is currently in BUSY state	0x000000
0x8080	0, 1	BUSY_MAIN_OUTPUT_FIFO_THRESH				
		BUSY_ENABLE	24	W	Enable busy generation if thresholds are crossed	0x0
		LOW	23:12	W	Low, Negate threshold of busy generation from main output fifo	0x3FF

		HIGH	11:0	W	High, Assert threshold of busy generation from main output fifo	0x4FF
0x8090	0, 1	BUSY_MAIN_OUTPUT_FIFO_STATUS				
		CLEAR_LATCHED	any	T	Any write to this register will clear the	0x0
		HIGH_THRESH_CROSSED_LATCHED	2	R	Main output fifo has been full beyond HIGH THRESHOLD, write to clear	0x0
		HIGH_THRESH_CROSSED	1	R	Main output fifo is full beyond HIGH THRESHOLD	0x0
		LOW_THRESH_CROSSED	0	R	Main output fifo is full beyond LOW THRESHOLD	0x0
ELINK_BUSY_ENABLE						
0x80A0	0	ELINK_BUSY_ENABLE00	56:0	W	Per elink (and FULL mode link) enable of the busy signal towards the LEMO output	0x00000000000000
...						
0x8210	0	ELINK_BUSY_ENABLE23	56:0	W	Per elink (and FULL mode link) enable of the busy signal towards the LEMO output	0x00000000000000
XOFF_STATISTICS						
0x8220	0,1	XOFF_PEAK_DURATION00	63:0	R	Maximum occurred duration of XOFF on the given channel in 25ns bins since reset	0x0000000000000000
0x8230	0,1	XOFF_TOTAL_DURATION00	63:0	R	Total occurred duration of XOFF on the given channel in 25ns bins, divide by number of Xoffs to calculate the average since reset	0x0000000000000000
0x8240	0,1	XOFF_COUNT00	63:0	R	Total number of XOFF events per channel that occurred since a reset.	0x0000000000000000
...						
0x8670	0,1	XOFF_PEAK_DURATION23	63:0	R	Maximum occurred duration of XOFF on the given channel in 25ns bins since reset	0x0000000000000000
0x8680	0,1	XOFF_TOTAL_DURATION23	63:0	R	Total occurred duration of XOFF on the given channel in 25ns bins, divide by number of Xoffs to calculate the average since reset	0x0000000000000000
0x8690	0,1	XOFF_COUNT23	63:0	R	Total number of XOFF events per channel that occurred since a reset.	0x0000000000000000
0x86A0	0, 1	BUSY_TOHOST_ENABLE	0	W	Enable the busy ToHost Virtual Elink	0x0
LTITTCBUSY Controls And Monitors						
0x8800	0	LTITTC_ALIGNMENT_DONE	0:0	R	RX ALIGNMENT DONE	0x0
0x8810	0	LTITTC_CPLL_FBCLK_LOST	0:0	R	CPLL FBCLK LOST	0x0
0x8820	0	LTITTC_PLL_LOCK				
		QPLL_LOCK	1:1	R	QPLL LOCK	0x0

		CPLL_LOCK	0:0	R	CPLL LOCK	0x0
0x8830	0	LTITTC_RXCDR_LOCK	0:0	R	RX CDR LOCK	0x0
0x8840	0	LTITTC_RXRESET_DONE	0:0	R	RX Reset done	0x0
0x8850	0	LTITTC_RX_BYTEISALIGNED	0:0	R	LTITTC link not aligned	0x0
0x8860	0	LTITTC_RX_DISP_ERROR	3:0	R	Rx disp error in byte 3,2,1,0 of LTITTC link	0x0
0x8870	0	LTITTC_RX_NOTINTABLE	3:0	R	Character in byte 3,2,1,0 of LTITTC link not in 8b10b table	0x0
LTITTC_CTRLMON						
0x8880	0	LTITTC_CTRL				
		LTITTC_GTH_LOOPBACK_CONTROL	11:9	W	GTH_LOOPBACK_CONTROL for LTITTC Link	0x0
		LTITTC_SOFT_RESET	8	W	SOFT_RESET	0x0
		LTITTC_QPLL_RESET	7	W	QPLL_RESET	0x0
		LTITTC_CPLL_RESET	6	W	CPLL_RESET	0x0
		LTITTC_SOFT_TX_RESET	5	W	SOFT_TX_RESET_ALL	0x0
		LTITTC_SOFT_RX_RESET	4	W	SOFT_RX_RESET_ALL	0x0
		LTITTC_GENERAL_CTRL	3:2	W	Alignment chk reset (not self clearing)	0x0
		LTITTC_CHANNEL_DISABLE	1	W	clear toHostData	0x0
		TOHOST_RST	0	W	clear toHostData	0x0
0x8890	0	LTITTC_MON				
		BUSY_OUTPUT_STATUS	3	R	Actual status of the BUSY LEMO output signal	0x0
		LTITTC_BIT_ERR	2:0	R	Alignment comma not received correctly. Place holder	0x0
LTITTC_BUSY_ACCEPTED_G						
0x88A0	0,1	LTITTC_BUSY_ACCEPTED00	56:0	R	busy has been asserted by the given ELINK. Reset by writing to TTC_BUSY_CLEAR	0x00000000000000
...						
0x8A10	0,1	LTITTC_BUSY_ACCEPTED23	56:0	R	busy has been asserted by the given ELINK. Reset by writing to TTC_BUSY_CLEAR	0x00000000000000
0x8A20	0	LTITTC_SL0ID_MONITOR				
		CLEAR	any	T	Counts Set L0ID input bits	0x0
		VALUE	31:0	R	Counts Set L0ID input bits	0x00000000
0x8A30	0	LTITTC_SORB_MONITOR				
		CLEAR	any	T	Counts SetOrbit input bits	0x0

		VALUE	31:0	R	Counts SetOrbit input bits	0x00000000
0x8A40	0	LTITTC_GRST_MONITOR				
		CLEAR	any	T	Counts GRST input bits	0x0
		VALUE	31:0	R	Counts GRST input bits	0x00000000
0x8A50	0	LTITTC_SYNC_MONITOR				
		CLEAR	any	T	Counts the Sync input bits	0x0
		VALUE	31:0	R	Counts the Sync input bits	0x00000000
0x8A60	0	LTITTC_TTYPE_MONITOR				
		CLEAR	any	T	Counts the number of TType received from the LTITTC system, any write to this register clears the counter	0x0
		VALUE	46:15	R	Counts the number of TType received from the LTITTC system, any write to this register clears the counter	0x00000000
		REFVALUE	15:0	W	Counts the number of TType received from the LTITTC system, any write to this register clears the counter	0x0000
0x8A70	0	LTITTC_L0ID_ERR_MONITOR				
		CLEAR	any	T	Counts the number of times the internal l0id /= input L0ID	0x0
		VALUE	31:0	R	Counts the number of times the internal l0id /= input L0ID	0x00000000
0x8A80	0	LTITTC_BCR_ERR_MONITOR				
		CLEAR	any	T	Counts the number of times the BCR period does not match 3564, any write to this register clears the counter	0x0
		VALUE	31:0	R	Counts the number of times the BCR period does not match 3564, any write to this register clears the counter	0x00000000
0x8A90	0	LTITTC_CRC_ERR_MONITOR				
		CLEAR	any	T	Counts the number of time the internally computed crc /= input CRC	0x0
		VALUE	31:0	R	Counts the number of time the internally computed crc /= input CRC	0x00000000
House Keeping Controls And Monitors						
0x9000	0	HK_CTRL_I2C				
		CONFIG_TRIG	1	W	i2c_config_trig	0x0
		CLKFREQ_SEL	0	W	i2c_clkfreq_sel	0x0
0x9010	0	HK_CTRL_FMC				
		CLEAR	any	T	Write to this bitfield clears the latched SI5345_LOL status, SI5345_LOL_LATCHED	0x0
		SI5345_LOL_LATCHED	14	R	Latched version of SI5345_LOL, clear by writing to CLEAR bitfield	0x0

		SI5345_INTR_B	13:12	R	Connects to SI5345_INTR_B pins	0x0
		SI5345_FINC_B	11:10	W	Connects to FINC_B pins of SI5345	0x1
		SI5345_FDEC_B	9:8	W	Connects to FDEC_B pins of SI5345	0x1
		SI5345_LOL	7	R	Loss of lock pin, not connected on VC709	0x0
		SI5345_INSEL	6:5	W	Selects the input clock source 0 : FPGA (FMC LA01) 1 : FMC OSC (40.079 MHz) 2 : FPGA (FMC LA18)	0x0
		SI5345_A	4:3	W	Si5345 I2C address select 2 LSB (0x0:default, dev id 0x68)	0x0
		SI5345_OE	2	W	Si5345 active low output enable (0:enable)	0x1
		SI5345_RSTN	1	W	Si5345 active low reset (0:reset)	0x1
		SI5345_SEL	0	W	Si5345 programming mode 1 : I2C mode (default) 0 : SPI mode	0x1
0x9300	0	MMCM_MAIN				
		CLEAR	any	T	Clears the LOL_LATCHED status	0x0
		LOL_LATCHED	4	R	Main MMCM has lost lock, clear by writing to the CLEAR bitfield	0x0
		LCLK_SEL	3	W	1: LCLK 0: TTC	0x1
		MAIN_INPUT	2:1	R	Main MMCM Oscillator Input 2: LCLK fixed 1: TTC fixed 0: selectable	0x0
		PLL_LOCK	0	R	Main MMCM PLL Lock Status	0x0
0x9310	0	LMK_LOCKED	0	R	LMK Chip on BNL-711 locked	0x0
0x9320	0	FPGA_CORE_TEMP	11:0	R	XADC temperature monitor for the FPGA CORE for FLX709, FLX710 temp (C)= ((FPGA_CORE_TEMP* 503.975)/4096)-273.15 for FLX711 temp (C)= ((FPGA_CORE_TEMP* 502.9098)/4096)-273.8195	0x000
0x9330	0	FPGA_CORE_VCCINT	11:0	R	XADC voltage measurement VCCINT = (FPGA_CORE_VCCINT *3.0)/4096	0x000
0x9340	0	FPGA_CORE_VCCAUX	11:0	R	XADC voltage measurement VCCAUX = (FPGA_CORE_VCCAUX *3.0)/4096	0x000

0x9350	0	FPGA_CORE_VCCBRAM	11:0	R	XADC voltage measurement VCCBRAM = (FPGA_CORE_VCCBRAM *3.0)/4096	0x000
0x9360	0	FPGA_DNA	63:0	R	Unique identifier of the FPGA	0x0000000000000000
0x9420	0	I2C_WR				
		I2C_WREN	any	T	Any write to this register triggers an I2C read or write sequence	0x0
		DATA_BYTE3	34:27	W	Data byte 3 used when RW16BIT is set	0x00
		RW16BIT	26	W	Set to 1 to Write 3 bytes (ADDR + 16 data bits) or read 16 data bits.	0x0
		I2C_FULL	25	R	I2C FIFO full	0x0
		WRITE_2BYTES	24	W	Write two bytes	0x0
		DATA_BYTE2	23:16	W	Data byte 2	0x00
		DATA_BYTE1	15:8	W	Data byte 1	0x00
		SLAVE_ADDRESS	7:1	W	Slave address	0x0
		READ_NOT_WRITE	0	W	READ/<o>WRITE</o>	0x0
0x9430	0	I2C_RD				
		I2C_RDEN	any	T	Any write to this register pops the last I2C data from the FIFO	0x0
		I2C_EMPTY	8	R	I2C FIFO Empty	0x0
		I2C_DOUT	7:0	R	I2C READ Data	0x00
0x9800	0	INT_TEST				
		TRIGGER	any	T	Fire a test MSIx interrupt set in IRQ	0x0
		IRQ	3:0	W	Set this field to a value equal to the MSIX interrupt to be fired. The write triggers the interrupt immediately.	0x0
0x9810	0	CONFIG_FLASH_WR				
		FAST_WRITE	57	W	Write command only. Only used for fast programming.	0x0
		FAST_READ	56	W	Status reading without command writing. Only used for fast programming.	0x0
		PAR_CTRL	55	W	Choose use FW or uC to select the Flash partition. 1 FW | 0 uC.	0x0
		PAR_WR	54:53	W	Choose Flash partition. Valid when PAR_CTRL is 1.	0x0
		FLASH_SEL	52	W	1 takes control over flash, 0 gives JTAG control over flash	0x0
		DO_INIT	51	W	Untested feature, don't use it yet.	0x0
		DO_READSTATUS	50	W	Reads status from flash	0x0
		DO_CLEARSTATUS	49	W	Clears status reading from flash, back to normal flash operation	0x0

		DO_ERASEBLOCK	48	W	Erased the current block of the flash, this register has to be cleared by software	0x0
		DO_UNLOCK_BLOCK	47	W	Unlock writes to the current block, this register has to be cleared by software	0x0
		DO_READ	46	W	Reads the 16 bits from current address, this register has to be cleared by software	0x0
		DO_WRITE	45	W	Writes the 16 bits to current address, this register has to be cleared by software	0x0
		DO_READDEVICEID	44	W	DIN should return 0x0089, this register has to be cleared by software	0x0
		DO_RESET	43	W	Can be used in the future, currently disconnected in firmware	0x0
		ADDRESS	42:16	W	Address for read and write operations (25 bits, upper 2 bits are controlled by uC)	0x000000
		WRITE_DATA	15:0	W	Value of data to write towards flash	0x0000
0x9820	0	CONFIG_FLASH_RD				
		PAR_RD	19:18	R	Show which Flash partition is selected.	0x0
		FLASH_REQ_DONE	17	R	Request done	0x0
		FLASH_BUSY	16	R	Flash operation busy	0x0
		READ_DATA	15:0	R	Value of data read from flash	0x0000
0x9830	0	SI5324_STATUS				
		LOL	15:8	R	Loss of Lock Si5324	0x00
		LOS	8:0	R	Loss of Signal Si5324	0x00
0x9840	0	TACH_CNT	19:0	R	Readout of the Fan tachometer speed of the BNL712 board	0x00000
0x9850	0	RXUSRCLK_FREQ				
		VALID	38	R	Indicates that the frequency measurement is valid	0x0
		CHANNEL	37:32	W	Select the Transceiver channel to measure the clock from.	0x0
		VAL	31:0	R	Frequency in Hz of the selected channel	0x00000000
Generators						
0xA000	0	FELIG_L1ID_RESET	any	T	Any write to this register clears the FELIG L1ID	0x0
FELIG_DATA_GEN_CONFIG_ARR						
0xA020	0	FELIG_DATA_GEN_CONFIG_00				
		CHUNK_LENGTH	50:35	W	FELIG data generator chunk-length in bytes.	0x0000
		RESET	34:28	W	FELIG data generator reset. One bit per group, 0:normal operation, 1:egroup emulation held in reset.	0x0

		SW_BUSY	27:21	W	FELIG elink busy state. One bit per group, 0:normal operation, 1:elink enter busy state.	0x0
		DATA_FORMAT	20:7	W	FELIG data generator format, 2 bits per e-group. 00 8b10b, 01 direct, 10 Aurora	0x000
		PATTERN_SEL	6:0	W	FELIG data payload type. One bit per group, 0:byte counter, 1:USERDATA	0x0
		...				
0xA190	0	FELIG_DATA_GEN_CONFIG_23				
		CHUNK_LENGTH	50:35	W	FELIG data generator chunk-length in bytes.	0x0000
		RESET	34:28	W	FELIG data generator reset. One bit per group, 0:normal operation, 1:egroup emulation held in reset.	0x0
		SW_BUSY	27:21	W	FELIG elink busy state. One bit per group, 0:normal operation, 1:elink enter busy state.	0x0
		DATA_FORMAT	20:7	W	FELIG data generator format, 2 bits per e-group. 00 8b10b, 01 direct, 10 Aurora	0x000
		PATTERN_SEL	6:0	W	FELIG data payload type. One bit per group, 0:byte counter, 1:USERDATA	0x0
FELIG_ELINK_CONFIG_ARR						
0xA1A0	0	FELIG_ELINK_CONFIG_00				
		ENDIAN_MOD	34:28	W	FELIG elink data input endian control. One bit per egroup. 0:little-endian (8b10b), 1:big-endian.	0x0
		INPUT_WIDTH	27:21	W	FELIG elink data input width. One bit per egroup. 0:8-bit (direct), 1:10-bit (8b10b).	0x0
		OUTPUT_WIDTH	20:0	W	FELIG elink data output width. 3 bits per egroup. 0:2b, 1:4b, 2:8b, 3:16b, 4:32b	0x00000
		...				
0xA310	0	FELIG_ELINK_CONFIG_23				
		ENDIAN_MOD	34:28	W	FELIG elink data input endian control. One bit per egroup. 0:little-endian (8b10b), 1:big-endian.	0x0
		INPUT_WIDTH	27:21	W	FELIG elink data input width. One bit per egroup. 0:8-bit (direct), 1:10-bit (8b10b).	0x0
		OUTPUT_WIDTH	20:0	W	FELIG elink data output width. 3 bits per egroup. 0:2b, 1:4b, 2:8b, 3:16b, 4:32b	0x00000
FELIG_ELINK_ENABLE_ARR						
0xA320	0	FELIG_ELINK_ENABLE_00	39:0	W	FELIG elink enable. One bit per elink. 0:disabled, 1:enabled.	0x0000000000
		...				
0xA490	0	FELIG_ELINK_ENABLE_23	39:0	W	FELIG elink enable. One bit per elink. 0:disabled, 1:enabled.	0x0000000000

0xA4A0	0	FELIG_GLOBAL_CONTROL				
		FAKE_L1A_RATE	63:36	W	Sets the internal fake L1 trigger rate. [25ns/LSB]	0x0000000
		PICXO_OFFSET_PPM	35:14	W	When OFFSET_EN is 1, this directly sets the output frequency, within the given adjustment range.	0x00000
		TRACK_DATA	12:12	W	FELIG GT core control. Must be set to enable normal operation.	0x0
		RXUSERRDY	11:11	W	FELIG GT core control. Must be set to enable normal operation.	0x0
		TXUSERRDY	10:10	W	FELIG GT core control. Must be set to enable normal operation.	0x0
		AUTO_RESET	9:9	W	FELIG GT core control. If set the GT core automatically resets on data error.	0x0
		PICXO_RESET	8:8	W	FELIG GT core control. Manual PICXO reset.	0x0
		GTTX_RESET	7:7	W	FELIG GT core control. Manual GT TX reset	0x0
		CPLL_RESET	6:6	W	FELIG GT core control. Manual CPLL reset.	0x0
		X3_X4_OUTPUT_SELECT	5:0	W	X3/X4 SMA output source select.	0x0
FELIG_LANE_CONFIG_ARR						
0xA4B0	0	FELIG_LANE_CONFIG_00				
		B_CH_BIT_SEL	63:42	W	When OFFSET_EN is 1. this directly sets the output frequency. within the given adjustment range.	0x00000
		A_CH_BIT_SEL	41:35	W	Selects the bit from the received FELIX data from which to extract the L1A.	0x0
		LB_FIFO_DELAY	34:30	W	When the GTH or GTB loopback is enabled, this controls the loopback latency in clock cycles.	0x0
		ELINK_SYNC	7:7	W	When set, synchronizes the elink word boundaries. Must be set back to 0 to resume normal operation.	0x0
		PICXO_OFFEST_EN	6:6	W	FELIG TX frequency override. 0:frequency tracking enabled, 1:TX frequency set by PICXO_OFFSET_PPM.	0x0
		PI_HOLD	5:5	W	FELIG phase-interpolator hold. 0:frequency tracking enabled, 1:freeze TX frequency.	0x0
		GBT_LB_ENABLE	4:4	W	FELIG GBT direct loopback enable. 0:disabled, 1:enabled.	0x0
		GBH_LB_ENABLE	3:3	W	FELIG GTH direct loopback enable. 0:disabled, 1:enabled.	0x0
		L1A_SOURCE	2:2	W	FELIG L1A data source select. 0:from local counter, 1:from FELIX.	0x0
		GBT_EMU_SOURCE	1:1	W	FELIG emulation data source select. 0:state-machine emulator, 1:ram-based emulator.	0x0
		FG_SOURCE	0:0	W	FELIG link check data source selection control. 0:normal operation, 1:PRBS link checker (not elink emulation data)	0x0
...						

0xA620	0	FELIG_LANE_CONFIG_23				
		B_CH_BIT_SEL	63:42	W	When OFFSET_EN is 1. this directly sets the output frequency. within the given adjustment range.	0x00000
		A_CH_BIT_SEL	41:35	W	Selects the bit from the received FELIX data from which to extract the L1A.	0x0
		LB_FIFO_DELAY	34:30	W	When the GTH or GTB loopback is enabled, this controls the loopback latency in clock cycles.	0x0
		ELINK_SYNC	7:7	W	When set, synchronizes the elink word boundaries. Must be set back to 0 to resume normal operation.	0x0
		PICXO_OFFEST_EN	6:6	W	FELIG TX frequency override. 0:frequency tracking enabled, 1:TX frequency set by PICXO_OFFSET_PPM.	0x0
		PI_HOLD	5:5	W	FELIG phase-interpolator hold. 0:frequency tracking enabled, 1:freeze TX frequency.	0x0
		GBT_LB_ENABLE	4:4	W	FELIG GBT direct loopback enable. 0:disabled, 1:enabled.	0x0
		GBH_LB_ENABLE	3:3	W	FELIG GTH direct loopback enable. 0:disabled, 1:enabled.	0x0
		L1A_SOURCE	2:2	W	FELIG L1A data source select. 0:from local counter, 1:from FELIX.	0x0
		GBT_EMU_SOURCE	1:1	W	FELIG emulation data source select. 0:state-machine emulator, 1:ram-based emulator.	0x0
		FG_SOURCE	0:0	W	FELIG link check data source selection control. 0:normal operation, 1:PRBS link checker (not elink emulation data)	0x0
FELIG_MON_TTC_0_ARR						
0xA630	0	FELIG_MON_TTC_0_00				
		L1ID	63:40	R	Live TTC data monitor.	0x000000
		XL1ID	39:32	R	Live TTC data monitor.	0x00
		BCID	31:20	R	Live TTC data monitor.	0x000
		RESERVED0	19:16	R	Live TTC data monitor.	0x0
		LEN	15:8	R	Live TTC data monitor.	0x00
		FMT	7:0	R	Live TTC data monitor.	0x00
...						
0xA7A0	0	FELIG_MON_TTC_0_23				
		L1ID	63:40	R	Live TTC data monitor.	0x000000
		XL1ID	39:32	R	Live TTC data monitor.	0x00
		BCID	31:20	R	Live TTC data monitor.	0x000
		RESERVED0	19:16	R	Live TTC data monitor.	0x0

		LEN	15:8	R	Live TTC data monitor.	0x00
		FMT	7:0	R	Live TTC data monitor.	0x00
FELIG_MON_TTC_1_ARR						
0xA7B0	0	FELIG_MON_TTC_1_00				
		RESERVED1	63:48	R	Live TTC data monitor.	0x0000
		TRIGGER_TYPE	47:32	R	Live TTC data monitor.	0x0000
		ORBIT	31:0	R	Live TTC data monitor.	0x00000000
...						
0xA920	0	FELIG_MON_TTC_1_23				
		RESERVED1	63:48	R	Live TTC data monitor.	0x0000
		TRIGGER_TYPE	47:32	R	Live TTC data monitor.	0x0000
		ORBIT	31:0	R	Live TTC data monitor.	0x00000000
FELIG_MON_COUNTERS_ARR						
0xA930	0	FELIG_MON_COUNTERS_00				
		SLIDE_COUNT	63:32	R	Counts the number of rx slides commanded by the GBT logic. Should be static once a link is established.	0x00000000
		FC_ERROR_COUNT	31:0	R	When FG_DATA_SELECT is 1, this counter reports the number of detected data errors.	None
...						
0xAAA0	0	FELIG_MON_COUNTERS_23				
		SLIDE_COUNT	63:32	R	Counts the number of rx slides commanded by the GBT logic. Should be static once a link is established.	0x00000000
		FC_ERROR_COUNT	31:0	R	When FG_DATA_SELECT is 1, this counter reports the number of detected data errors.	None
FELIG_MON_FREQ_ARR						
0xAAB0	0	FELIG_MON_FREQ_00				
		TX	63:32	R	FELIG regenerated TX clock frequency[Hz].	0x00000000
		RX	31:0	R	FELIG recovered RX clock frequency[Hz].	0x00000000
...						
0xAC20	0	FELIG_MON_FREQ_23				
		TX	63:32	R	FELIG regenerated TX clock frequency[Hz].	0x00000000

		RX	31:0	R	FELIG recovered RX clock frequency[Hz].	0x00000000
0xAC30	0	FELIG_MON_FREQ_GLOBAL				
		XTAL_100MHZ	63:32	W	FELIG local oscillator frequency[Hz].	0x00000000
		CLK_41_667MHZ	31:0	W	FELIG PCIE MGTREFCLK frequency[Hz].	0x00000000
FELIG_MON_L1 A_ID_ARR						
0xAC40	0	FELIG_MON_L1A_ID_00	31:0	R	FELIG's last L1 ID.	0x00000000
		...				
0xADB0	0	FELIG_MON_L1A_ID_23	31:0	R	FELIG's last L1 ID.	0x00000000
FELIG_MON_PICXO_ARR						
0xADC0	0	FELIG_MON_PICXO_00				
		VLOT	53:32	R	Value indicates TX clock (recovered RX clock) to RX reference clock frequency offset.	0x00000
		ERROR	20:0	R	Value indicates RX to TX frequency tracking error.	0x00000
		...				
0xAF30	0	FELIG_MON_PICXO_23				
		VLOT	53:32	R	Value indicates TX clock (recovered RX clock) to RX reference clock frequency offset.	0x00000
		ERROR	20:0	R	Value indicates RX to TX frequency tracking error.	0x00000
0xAF40	0	FELIG_RESET				
		LB_FIFO	63:48	W	One bit per lane. When set to 1, resets all loopback FIFOs.	0x0000
		FRAMEGEN	47:24	W	One bit per lane. When set to 1, resets all FELIG link checking logic.	0x000000
		LANE	23:0	W	One bit per lane. When set to 1, resets all FELIG lane logic.	0x000000
0xAF50	0	FELIG_RX_SLIDE_RESET	23:0	W	One bit per lane. When set to 1, resets the gbt rx slide counter.	0x000000
FELIG_ITK_STRIPS_DATA_GEN_CONFIG_ARR						
0xAF60	0	FELIG_ITK_STRIPS_DATA_GEN_CONFIG_00				
		ITKS_FIFO_CTL	19:17	W	data fifo control 2:rst 1:rd 0:wr.	0x0
		ITKS_FIFO_DATA	16:0	W	itks emu data 16:last word 15-0:data word	0x0000
		...				
0xB0D0	0	FELIG_ITK_STRIPS_DATA_GEN_CONFIG_23				
		ITKS_FIFO_CTL	19:17	W	data fifo control 2:rst 1:rd 0:wr.	0x0

		ITKS_FIFO_DATA	16:0	W	itks emu data 16:last word 15-0:data word	0x0000
FELIG_MON_ITK_STRIPS_ARR						
0xB0E0	0	FELIG_MON_ITK_STRIPS_00	2:0	R	data fifo status 2:write done 1:full 0:empty.	0x0
			...			
0xB250	0	FELIG_MON_ITK_STRIPS_23	2:0	R	data fifo status 2:write done 1:full 0:empty.	0x0
FELIG_DATA_GEN_CONFIG_USERDATA_ARR						
0xB260	0	FELIG_DATA_GEN_CONFIG_00_-USERDATA	15:0	W	Sets static payload word. When FELIG_DATA_GEN_CONFIG.PATTERN_SEL=1.	0x0000
			...			
0xB3D0	0	FELIG_DATA_GEN_CONFIG_23_-USERDATA	15:0	W	Sets static payload word. When FELIG_DATA_GEN_CONFIG.PATTERN_SEL=1.	0x0000
0xB800	0	FMEMU_EVENT_INFO				
		L1ID	63:32	R	32b field to show L1ID	0x00000000
		BCID	31:0	R	32b field to show BCID	0x00000000
0xB810	0	FMEMU_COUNTERS				
		WORD_CNT	63:48	W	Number of 32b words in one chunk	0x0020
		IDLE_CNT	47:32	W	Minimum number of idles between chunks	0x0003
		L1A_CNT	31:16	W	Number of chunks to send if not in TTC mode	0x0100
		BUSY_TH_HIGH	15:8	W	Assert BUSY-ON above this threshold	0x14
		BUSY_TH_LOW	7:0	W	De-assert BUSY-ON below this threshold	0x0F
0xB820	0	FMEMU_CONTROL				
		L1A_BITNR	63:56	W	Bitfield for L1A in TTC frame	0x30
		XONXOFF_BITNR	55:48	W	Bitfield for Xon/Xoff in TTC frame	0x20
		EMU_START	47:47	W	Start emulator functionality	0x0
		TTC_MODE	46:46	W	Control the emulator by TTC input or by RegMap (1/0)	0x0
		XONXOFF	45:45	W	Enable Xon/Xoff functionality (1/0)	0x1
		INLC_CRC32	44:44	W	0: No checksum 1: Append the data with a CRC32	0x0
		BCR	43:43	W	Reset BCID to 0	0x0
		ECR	42:42	W	Reset L1ID to 0	0x0

		CONSTANT_CHUNK_LENGTH	41:41	W	Data source select 0: Random chunk length 1: Constant chunk length	0x0
		INT_STATUS_EMU	40:32	R	Read internal status emulator	0x00
		FFU_FM_EMU_T	16	W	For Future Use (trigger registers)	0x0
		FE_BUSY_ENABLE	0	W	Enable the BUSY mechanism if L1A counter passes threshold	0x1
0xB830	0	FMEMU_RANDOM_RAM_ADDR	9:0	W	Controls the address of the ramblock for the random number generator	0x00
0xB840	0	FMEMU_RANDOM_RAM				
		WE	any	T	Any write to this register (DATA) triggers a write to the ramblock	0x0
		CHANNEL_SELECT	39:16	W	Enable write enable only for the selected channel	0x000000
		DATA	15:0	W	DATA field to be written to FMEMU_RANDOM_RAM_ADDR	0x0000
0xB850	0	FMEMU_RANDOM_CONTROL				
		SELECT_RANDOM	20	W	1 enables the random chunk length, 0 uses a constant chunk length	0x0
		SEED	19:10	W	Seed for the random number generator, should not be 0	0x200
		POLYNOMIAL	9:0	W	POLYNOMIAL for the random number generator (10b LFSR) Bit9 should always be 1	0x240
0xB860	0	FMEMU_CONFIG_WRADDR	9:0	W	write enable for the FMEmu ram block	0x00
0xB870	0	FMEMU_CONFIG				
		WE	any	T	Any write to register WRDATA triggers a write to the ramblock	0x0
		CHANNEL_SELECT	55:32	W	Enable write enable only for the selected channel	0x000000
		WRDATA	31:0	W	DATA field to be written to FMEMU_RANDOM_RAM_ADDR	0x00000000
Wishbone						
0xC000	0	WISHBONE_CONTROL				
		WRITE_NOT_READ	32	W	wishbone write command wishbone read command	0x0
		ADDRESS	31:0	W	Slave address for Wishbone bus	0x00000000
0xC010	0	WISHBONE_WRITE				
		WRITE_ENABLE	any	T	Any write to this register triggers a write to the Wupper to Wishbone fifo	0x0
		FULL	32	R	Wishbone	0x0
		DATA	31:0	W	Wishbone	0x00000000
0xC020	0	WISHBONE_READ				
		READ_ENABLE	any	T	Any write to this register triggers a read from the Wishbone to Wupper fifo	0x0

		EMPTY	32	R	Indicates that the Wishbone to Wupper fifo is empty	0x0
		DATA	31:0	R	Wishbone read data	0x00000000
0xC030	0	WISHBONE_STATUS				
		INT	4	R	interrupt	0x0
		RETRY	3	R	Interface is not ready to accept data cycle should be retried	0x0
		STALL	2	R	When pipelined mode slave can't accept additional transactions in its queue	0x0
		ACKNOWLEDGE	1	R	Indicates the termination of a normal bus cycle	0x0
		ERROR	0	R	Address not mapped by the crossbar	0x0
IP Bus						
0xC800	0	IPBUS_WRITE_ADDRESS	31:0	W	Address of the IPBus Write RAM	0x00000000
0xC810	0	IPBUS_WRITE_DATA				
		WRITE_ENABLE	any	T	Any write to this register triggers a write to the Wupper to IPBus inout RAM	0x0
		DATA	63:0	W	IPbus data to write to RAM	0x0000000000000000
0xC820	0	IPBUS_READ_ADDRESS	31:0	W	Address of the IPBus Read RAM	0x00000000
0xC830	0	IPBUS_READ_DATA	63:0	R	IPbus data from Read RAM	0x0000000000000000
0xC840	0	IPBUS_PKT_DONE	0	R	IPbus packet ready to read	0x0
ITK_STRIPS_CTRL						
0xD000	0,1	GLOBAL_STRIPS_CONFIG				
		TEST_MODULE_MASK	63:59	W	(for tests only) contains R3 mask for the simulated trigger data	0x0
		TEST_R3L1_TAG	58:52	W	(for tests only) contains R3 or L1 tag for the simulated trigger data	0x0
		TTC_GENERATE_GATING_ENABLE	51	W	Global control for gating signal generation. Enables generating trickle gating signal in response to TTC BCR. TRICKLE_TRIG_RUN must also be enabled for the trickle configuration to work. (See also BC_START, and BC_STOP fields)	0x0
		TTC_GATING_OVERRIDE	50	W	Overrides and disables gating signal generation when set to '1' (use if the elink is deadlocked and commands don't reach it).	0x0
		INVERT_AMAC_IN	4	W	Invert the polarity of all FELIX AMAC_IN elinks	0x0
		INVERT_AMAC_OUT	3	W	Invert the polarity of all FELIX AMAC_OUT elinks	0x0
		INVERT_DIN	2	W	Invert the polarity of all FELIX 8-bit IN 8b10b elinks	0x0
		INVERT_R3L1_OUT	1	W	Invert the polarity of all FELIX R3L1 elinks	0x0
		INVERT_LCB_OUT	0	W	Invert the polarity of all FELIX LCB elinks	0x0

0xD010	0,1	GLOBAL_TRICKLE_TRIGGER	any	T	writing to this register issues a single trickle trigger for every LCB link connected to this FELIX device	0x0
0xD020	0,1	STRIPS_R3_TRIGGER	any	T	(for tests only) simulate R3 trigger (issues 4-5 sequential triggers)	0x0
0xD030	0,1	STRIPS_L1_TRIGGER	any	T	(for tests only) simulate L1 trigger (issues 4-5 sequential triggers)	0x0
0xD040	0,1	STRIPS_R3L1_TRIGGER	any	T	(for tests only) simulate simultaneous R3 and L1 trigger (issues 4-5 sequential triggers)	0x0
MRO Dregisters						
0xF000	0	MROD_CTRL				
		OPTIONS	15:8	W	Extra options for MROD	0x00
		ENASPARE1	7:7	W	Enable spare1	0x0
		ENAMANSLIDE	6:6	W	Enable Manual Slide in Rx Locking	0x0
		ENAPASSALL	5:5	W	Enable PassAll in EmptySuppress	0x0
		ENATXCOUNT	4:4	W	Enable SimpleCount in TxDriver for locking	0x0
		GOLTESTMODE	3:0	W	GOL Test Mode (emulate CSM): 0: Run Data Emulator when 1; 0: stop, load emulator fifo 1: Enable Circulate when 1; 0: send fifo data only once 2: Enable Triggered Mode when 1; 0: run continueously (no TTC) 3: Enable pattern generator	0x0
0xF010	0	MROD_TCVRCTRL				
		SLIDEMAX	23:16	W	Maximum RXSLIDES before fire a TCVR reset	0xFF
		SLIDEWAIT	15:8	W	RXclk delay in TCVR for next RX_SLIDE operation	0x20
		FRAMESIZE	7:0	W	Number of 32 data words in 1 frame	0x14
0xF020	0	MROD_EP0_CSMENABLE	23:0	W	EP0 CSM Data Enable channel 23-0	0x000000
0xF030	0	MROD_EP0_EMPTYSUPPR	23:0	W	EP0 Set Empty Suppression channel 23-0	0x000000
0xF040	0	MROD_EP0_HPTDCMODE	23:0	W	EP0 Set HPTDC Mode channel 23-0	0x000000
0xF050	0	MROD_EP0_CLRFIFOS	23:0	W	EP0 Clear FIFOs channel 23-0	0x000000
0xF060	0	MROD_EP0_EMULOADENA	23:0	W	EP0 Emulator Load Enable channel 23-0	0x000000
0xF070	0	MROD_EP0_TRXLOOPBACK	23:0	W	EP0 Transceiver Loopback Enable channel 23-0	0x000000
0xF080	0	MROD_EP0_TXCVRRESET	23:0	W	EP0 Transceiver Reset all channel 23-0	0x000000
0xF090	0	MROD_EP0_RXRESET	23:0	W	EP0 Receiver Reset channel 23-0	0x000000
0xF0A0	0	MROD_EP0_TXRESET	23:0	W	EP0 Transmitter Reset channel 23-0	0x000000

0xF0B0	0	MROD_EP1_CSMENABLE	23:0	W	EP1 CSM Data Enable channel 23-0	0x000000
0xF0C0	0	MROD_EP1_EMPTYSUPPR	23:0	W	EP1 Set Empty Suppression channel 23-0	0x000000
0xF0D0	0	MROD_EP1_HPTDCMODE	23:0	W	EP1 Set HPTDC Mode channel 23-0	0x000000
0xF0E0	0	MROD_EP1_CLRFIFOS	23:0	W	EP1 Clear FIFOs channel 23-0	0x000000
0xF0F0	0	MROD_EP1_EMULOADENA	23:0	W	EP1 Emulator Load Enable channel 23-0	0x000000
0xF100	0	MROD_EP1_TRXLOOPBACK	23:0	W	EP1 Transceiver Loopback Enable channel 23-0	0x000000
0xF110	0	MROD_EP1_TXCVRRESET	23:0	W	EP1 Transceiver Reset all channel 23-0	0x000000
0xF120	0	MROD_EP1_RXRESET	23:0	W	EP1 Receiver Reset channel 23-0	0x000000
0xF130	0	MROD_EP1_TXRESET	23:0	W	EP1 Transmitter Reset channel 23-0	0x000000
MRO Dmonitors						
0xF800	0	MROD_EP0_CSMH_EMPTY	23:0	R	EP0 CSM Handler FIFO Empty 23-0	0x000000
0xF810	0	MROD_EP0_CSMH_FULL	23:0	R	EP0 CSM Handler FIFO Full 23-0	0x000000
0xF820	0	MROD_EP0_RXALIGNBSY	23:0	R	EP0 Receiver Aligned monitor 23-0	0x000000
0xF830	0	MROD_EP0_RXRECDATA	23:0	R	EP0 Receiver Data monitor 23-0	0x000000
0xF840	0	MROD_EP0_RXRECIDLES	23:0	R	EP0 Receiver Idle monitor 23-0	0x000000
0xF850	0	MROD_EP0_TXLOCKED	23:0	R	EP0 Transmitter Locked monitor 23-0	0x000000
0xF860	0	MROD_EP1_CSMH_EMPTY	23:0	R	EP1 CSM Handler FIFO Empty 23-0	0x000000
0xF870	0	MROD_EP1_CSMH_FULL	23:0	R	EP1 CSM Handler FIFO Full 23-0	0x000000
0xF880	0	MROD_EP1_RXALIGNBSY	23:0	R	EP1 Receiver Aligned monitor 23-0	0x000000
0xF890	0	MROD_EP1_RXRECDATA	23:0	R	EP1 Receiver Data monitor 23-0	0x000000
0xF8A0	0	MROD_EP1_RXRECIDLES	23:0	R	EP1 Receiver Idle monitor 23-0	0x000000
0xF8B0	0	MROD_EP1_TXLOCKED	23:0	R	EP1 Transmitter Locked monitor 23-0	0x000000

Table 10: FELIX register map BAR2.

References

[1] Atlas Felix project twiki (authorized users only)
https://twiki.cern.ch/twiki/bin/viewauth/Atlas/FELIXdevelopment

[2] ARM AMBA AXI bus standard specification page
http://www.arm.com/products/system-ip/amba/amba-open-specifications.php

[3] Xilinx website about the PCI Express core
http://www.xilinx.com/products/intellectual-property/7_Series_Gen_3_PCI_Express.htm

[4] UG761: Xilinx AXI Bus documentation
http://www.xilinx.com/support/documentation/ip_documentation/axi_ref_guide/latest/ ug761_axi_reference_guide.pdf

[5] PG023: Virtex-7 FPGA Gen3 Integrated Block for PCI Express v4.3
https://www.xilinx.com/support/documentation/ip_documentation/pcie3_7x/v4_3/pg023_v7_pcie_gen3.pdf

[6] PG156: UltraScale Devices Gen3 Integrated Block for PCI Express v4.4
https://www.xilinx.com/support/documentation/ip_documentation/pcie3_ultrascale/v4_4/pg156-ultrascale-pcie-gen3.pdf

[7] PG213: UltraScale+ Devices Integrated Block for PCI Express v1.3
https://www.xilinx.com/support/documentation/ip_documentation/pcie4_uscale_plus/v1_3/pg213-pcie4-ultrascale-plus.pdf

[8] PG343: Versal ACAP Integrated Block for PCI Express v1.0
https://www.xilinx.com/support/documentation/ip_documentation/pcie_versal/v1_0/pg343-pcie-versal.pdf

[9] M. Jackson, R. Budruk, *PCI Express Technology - Comprehensive Guide to Generations 1.x, 2.x and 3.0*, 1st Edition, MindShare Technology Series, 2012

[10] HDL Works EASE
http://hdlworks.com/products/ease/index.html

[11] Avoiding repeating passwords for CVS and SVN - ATLAS
https://confluence.slac.stanford.edu/display/Atlas/Avoiding+repeating+passwords+for+CVS+and+SVN

[12] Mentor Graphics Questasim
http://www.mentor.com/products/fv/questa/

[13] UG908: Programming and Debugging using Vivado
http://www.xilinx.com/support/documentation/sw_manuals/xilinx2014_2/ug908-vivado-programming-debugging.pdf

[14] J. Corbet, G. Kroah Hartman, A. Rubini, *Linux Device Drivers*, 3rd Edition, O'Reilly, 2005
http://www.oreilly.com/openbook/linuxdrive3/book/index.html

[15] J. Schumacher, M Donzelmann, *WupperCodeGen*
https://gitlab.cern.ch/atlas-tdaq-felix/wuppercodegen

List of Figures

List of Tables