# ATLAS LAr Calorimeter
# AMC Firmware Specifications

LAr firmware working group
atlas-lar-be-firmware@cern.ch

BERNARD DINKESPILER (CPPM – CNRS – France)
NICOLAS DUMONT-DAYOT (LAPP – CNRS – France)
YUJI ENARI (University of Tokyo – Japan)
RAINER HENTGES (University of Dresden – Germany)
KENNETH JOHNS (University of Arizona - USA)
STEPHANIE MAJEWSKI (University of Oregon - USA)
VLADIMIR ZHULANOV (Novosibirsk - Russia)

v02 // November 20, 2014

# Todo list

# Contents

Chronogram of the main signals

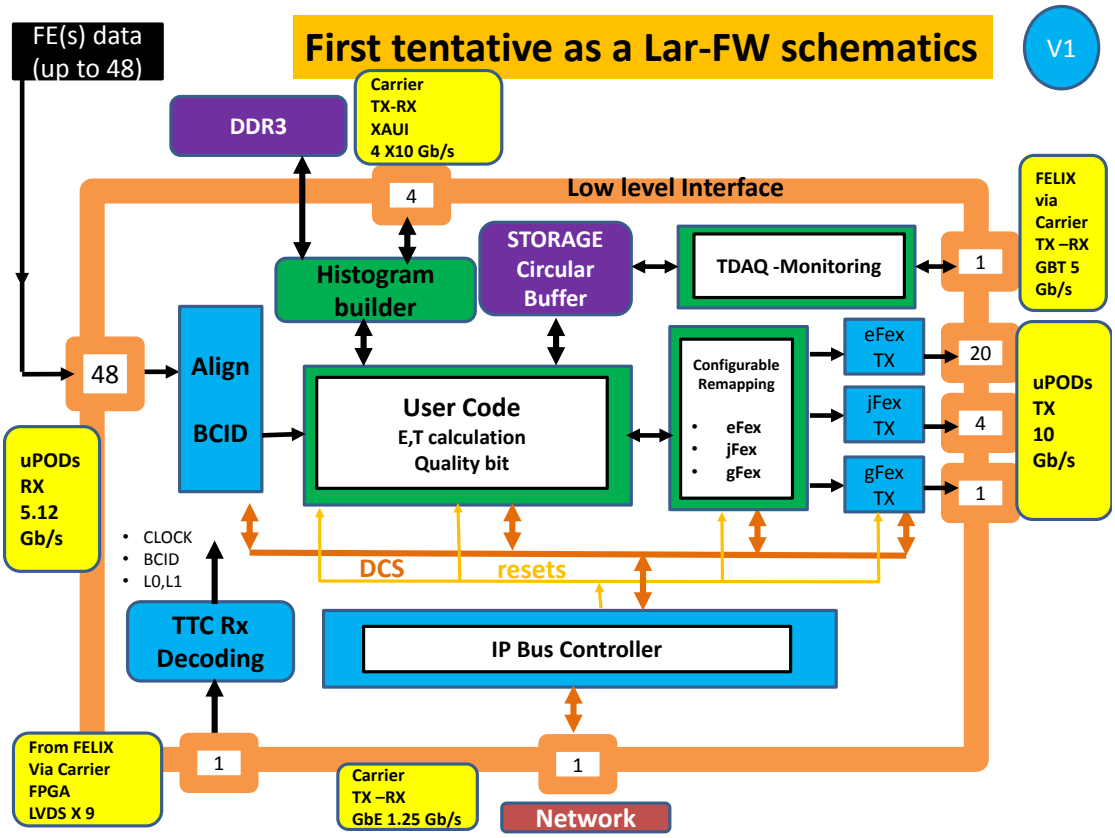
Insert here a chronogram of the control and data signals

# 1 Introduction

The present document aims at defining the functional specifications of the firmware embedded in the main FPGA of the AMC_A10 board, referenced 10AX115R4F40I4SGES (ARRIA 10 family – ALTERA™).

This document is based on the Atlas Liquid Argon Calorimeter Phase I Upgrade Technical Design Report and the work of a group of experts listed as the authors. At first a block diagram is presented showing the different blocks that constitute the firmware. Then, each block is specified in detail and, when applicable, the description of their implementation is presented.

Figure 1.1: Block diagram of the AMC firmware



# 2 Low Level Interface

## 2.1 Goal of the document

This document describes the low level interface part of the Arria10 firmware mounted on the LDPB (Larg Digital Processing Blade) AMC (Advanced Mezzanine Card). So it describes the interfaces between the external components (DDR3, GbE, XAUI and LTDB links ...) and the FPGA fabric.

## 2.2 DDR3 interfaces

### 2.2.1 Introduction

The ARRIA 10 is connected to 2 external MT41JT128M16JT-125 DDR3 for massive storage of data. This DDR3 is a 16 bits data bus memory with a theoretical maximum data rate of 3.2 $^{Gb}/_{s}$.

The density is 2GBits distributed in 8 banks of 16K row, each row containing 1K column of 16 bits (page of 2Kb).

### 2.2.2 Description

The ARRIA 10 can interface these DDR3 with 2 independent and parallel Altera EMIF (External Memory Interface) IP [3]. This IP can be built with a hard memory controller and a hard PHY that are part of the ARRIA10 silicon. The maximum data rate is obtained with the quarter clock rate for user logic. As only one row per bank can be opened at a given time it is recommended to make accesses from one bank to other for more efficiency.

### 2.2.3 External interfaces

The interface between the ARRIA 10 FPGA and the external memory includes the following signals.

| Port | Width | Direction | Description | IO Standard |
|---|---|---|---|---|
| Global_reset_n | 1 | Input | DDR3 Asynchronous reset active | 1.8 V |
| Pll_ref_clk | 1 | Input | Pll reference clock input | 1.5 V |
| Oct_rzqin | 1 | Input | Calibrated on chip termination | 1.5 V |
| Mem_ck | 1 | Output | Ck clock | Diff 1.5-V SSTL Class I |
| Mem_ck_n | 1 | Output | Ck clock (negative leg) | Diff 1.5-V SSTL Class I |
| Mem_a | 14 | Output | Address | SSTL-15 Class I |
| Mem_ba | 3 | Output | Bank address | SSTL-15 Class I |
| Mem_cke | 1 | Output | Clock enable | SSTL-15 Class I |
| Mem_odt | 1 | Output | On die termination | SSTL-15 Class I |
| Mem_reset_n | 1 | Output | Asynchronous reset | 1.5 V |
| Mem_we_n | 1 | Output | Write Enable command | SSTL-15 Class I |
| Mem_ras_n | 1 | Output | Column Active Strobe command | SSTL-15 Class I |
| Mem_cas_n | 1 | Output | Row Active Strobe command | SSTL-15 Class I |
| Mem_dqs | 2 | Bidir | Data strobe | Diff 1.5-V SSTL Class I |
| Mem_dqs_n | 2 | Bidir | Data strobe (negative leg) | Diff 1.5-V SSTL Class I |
| Mem_dq | 16 | Bidir | Read/write data | SSTL-15 |
| Mem_dm | 2 | Output | Write data mask | SSTL-15 |

Table 2.1: Ports of the DDR3 interface

### 2.2.4 FPGA fabric interfaces

The DDR3 is seen as an Avalon Memory mapped slave by the FPGA fabric. So the FPGA fabric interface should act as an Avalon Memory Mapped master in order to read or write data to the DDR3. The Avalon bus timing is given by the EMIF user clock output. This clock has a frequency constrained by the external DDR3 memory and the hard controller clock rate. The interface between the ARRIA 10 EMIF hard controller and the FPGA fabric is based on the standard Avalon Interface Memory Mapped interface [4] and includes the following signals (this description is preliminary because the ARRIA 10 EMIF local interface is not yet described in the ALTERA External Memory Interface Handbook. This description is based on the actual HPC II controller).

### 2.2.5 Examples

Quarter-rate DDR3 SDRAM reads
Quarter-rate DDR3 SDRAM writes
Waiting for new release of the ALTERA External Memory Interface Handbook for ARRIA 10 EMIF timing.

## 2.3 LTDB/FEX interfaces

### 2.3.1 Introduction

Connected to LTDB uPOD Rx and FEX uPOD Tx

Different speed rate: Rx=5.12Gbps – Tx=6.4/9.6Gbps???

PHY: A10 transceivers

Protocol: Rx=LOCIC, Tx=Unknown for the moment???

### 2.3.2 Description

PHY: Enhanced or standard transceivers, options, clocking Protocol: LOCIC for LTDB, ??? for FEX

### 2.3.3 External interfaces

Ref clock, Tx/Rx links, IO standard, IO termination...

| Port | Width | Direction | Description |
|---|---|---|---|
| Av_mm_reset_n | 1 | Output | Reset the EMIF Avalon MM user clock domain |
| Av_mm_usr_clk | 1 | Output | Avalon MM User clock domain : quarter rate (default) Ex : User Clock=200MHz for a 800MHz DDR3 clock |
| Av_mm_ready | 1 | Output | Asserted when controller is busy |
| Av_mm_read | 1 | Input | Read request signal |
| Av_mm_write | 1 | Input | Write request signal |
| Av_mm_address | 24 | Input | Memory address at which burst should start. Address must remain stable during first transaction of the burst<br>For quarter rate: Av_mm_address[23:0]<br>Av_mm_add[23:21] : bank address[2:0]<br>Av_mm_add[20:7] : row address[13:0]<br>Av_mm_add[6:0] : column address[9:3]<br>The IP core ignores the least significant bit (LSB) of the column address (multiples of 8) on the memory side, because the local data width is 8 times the memory data bus width. |
| Av_mm_readdata | 128 | Output | Read data : 8 times DDR3 16 bits data bus |
| Av_mm_writedata | 128 | Input | Write data : 8 times DDR3 16 bits data bus |
| Av_mm_burstcount | 7 | Input | Number of burst in each read/write burst |
| Av_mm_byteenable | 16 | Input | Byte enable for write data |
| Av_mm_readdatavalid | 1 | Output | Indicates whether read data is valid |

Table 2.2: Ports of the ARRIA 10 EMIF local interface

### 2.3.4 FPGA fabric interfaces

Avalon Streaming Interface Avalon MM Interface Internal R/W registers connected to Avalon MM Interface

## 2.4 GbE interfaces

### 2.4.1 Introduction

Connected to AMC connector for slow control PHY+MAC

### 2.4.2 Description

PHY: A10 transceivers MAC: triple speed Ethernet ???

### 2.4.3 External interfaces

Ref clock, Tx/Rx links, IO standard, IO termination...

### 2.4.4 FPGA fabric interfaces

Avalon Streaming Interface Avalon MM Interface

### 2.4.5 Registers

Internal R/W registers connected with Avalon MM Interface

## 2.5 XAUI interfaces

### 2.5.1 Introduction

Connected to AMC connector for "personal" fast monitoring PHY+MAC

### 2.5.2 Description

PHY: A10 transceivers + soft XAUI MAC: 10GbE ???

### 2.5.3 External interfaces

Ref clock, Tx/Rx links, IO standard, IO termination...

### 2.5.4 FPGA fabric interfaces

Avalon Streaming Interface Avalon MM Interface

### 2.5.5 Registers

Internal R/W registers connected with Avalon MM Interface

## 2.6 GBT interfaces

### 2.6.1 Introduction

Connected to AMC connector for T-DAQ fast monitoring and TTC decoding PHY + protocol

### 2.6.2 Description

PHY: Enhanced or standard transceivers, options, clocking Protocol: GBT

### 2.6.3 External interfaces

Ref clock, Tx/Rx links, IO standard, IO termination...

### 2.6.4 FPGA fabric interfaces

Avalon Streaming Interface Avalon MM Interface

### 2.6.5 Registers

Internal R/W registers connected to Avalon MM Interface

## 2.7 LVDS interfaces

### 2.7.1 Introduction

Connected to AMC connector for miscellaneous informations

### 2.7.2 Description

LVDS up to 1.6Gbps 2 LVDS TX/RX links

### 2.7.3 External interfaces

Ref clock, Tx/Rx links, IO standard, IO termination...

### 2.7.4 FPGA fabric interfaces

Avalon Streaming Interface Avalon MM Interface

### 2.7.5 Registers

Internal R/W registers connected to Avalon MM Interface

## 2.8 Others

### 2.8.1 Introduction

uPOD I2C, Arria10 ADC and sensing diode, Arria10 GPIO, EPCQ-L Flash

### 2.8.2 Description

### 2.8.3 External interfaces

GPIO, I2C, EPCQ-L

### 2.8.4 FPGA fabric interfaces

Avalon MM Interface

### 2.8.5 Registers

Internal R/W registers connected to Avalon MM Interface

# 3 Reception Stage

## 3.1 Introduction

The 5.12 Gb/s data stream coming from the LTDBs via optical fiber is received on the AMC board by a uPOD optical receiver. The optical stream is turned into an electrical stream which is sent to the RX part of a transceiver in the FPGA (see chapter 2). The output of this RX transceiver consists in a 16 b data path at 320 MHz that is the input to the reception stage.

## 3.2 Reception Stage task

The reception stage must:

- Detect the border pattern in the input serial stream to "cut out" 16 bit data words with the proper alignment.
- Extract the ADC data from the parallel data stream. These data are scrambled
- Unscramble the ADC data stream
- Extract the CRC from the data stream
- Compute the CRC from the unscrambled data and compare it to the extracted CRC
- Extract the BCID information from the parallel data stream and compute the current BCID.
- Constantly check that the BCID progression is consistent
- Provide the user code block a data stream and some flags:
  - Unscrambled ADC values : 12 bit @ 320 MHz
  - Status flags: synchronization done, BCID=0, ...
  - Error flags: loss of synchronization, BCID error, CRC error ...

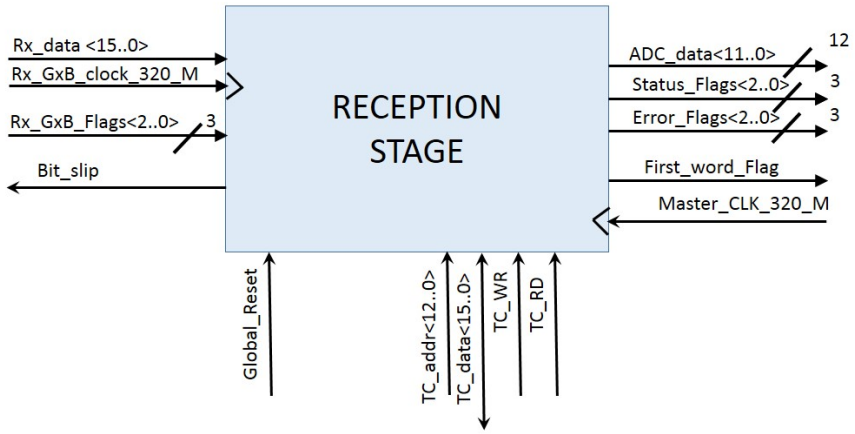



Figure 3.1: Block diagram of the reception stage module

- Provide an alignment mechanism in order to accommodate a set of 48 input fiber streams having different timing reference. The data stream to the user code block must ensure that the data are resynchronized on a unique Master_Clock for the whole FPGA and the BCIDs are the same.
- Provide a data injector based on a RAM buffer for self test purposes.

## 3.3 Reception Stage I/O

The reception stage block diagram is present in figure 3.1 The input and output ports are listed in table 3.1.

| Signal name | direction | function | comment |
|---|---|---|---|
| RX_data⟨15..0⟩ | Input | | |
| RX_GXB_Clock | Input | | |
| RX_GXB_Flag⟨0⟩ | Input | PLL_locked | |
| RX_GXB_Flag⟨1⟩ | Input | RX_is_locked_to_data | |
| RX_GXB_Flag⟨2⟩ | Input | RX_GX_B is Busy | |
| Bit_Slip | Output | RX will skip one bit | |
| ADC_data⟨11..0⟩ | Output | Extracted, unscrambled ADC data | |
| Status_Flag⟨0⟩ | Output | | |
| Status_Flag⟨1⟩ | Output | | |
| Status_Flag⟨2⟩ | Output | | |
| Error_Flag⟨0⟩ | Output | Loss of PLL or data | |
| Error_Flag⟨1⟩ | Output | Bad BCID | |
| Error_Flag⟨2⟩ | Output | Bad CRC | |
| Error_Flag⟨3⟩ | Output | FIFO_error | |
| First_word_FLAG | Output | Next word in FIFO is word 0 of BCID_0 | |
| Master_CLK_320 | input | Main 320 MHz from Main 40 Mhz | |
| GLOBAL_RESET | Input | Module reset | |
| TC_addr⟨13..0⟩ | Input | Address of Test Control location | |
| TC_data⟨15..0⟩ | Input/Output | Value stored in Test Control location | |
| TC_Write | Input | Write strobe | |
| TC_Read | Input | Read Strobe | |

Table 3.1: Ports of the reception stage module

## 3.4 Chronogram

Chronogram of the main signals

## 3.5 Implementation

In order to understand properly the implementation of the reception stage functions, one needs to know the format of the input data. The format of the input data is described in detail in the following publication: "A low_latency, small form factor Optical link for the High Luminosity LHC experiments" [1]

Insert here a chronogram of reference missing signals

The main points are the following:

- A set of 128 bit is transmitted every 25 ns, i.e for each BCID value.
- The pattern "0101" is used as a border pattern in order to be able to align the serial stream.
- The 12 bit ADC data are scrambled using a polynomial algorithm.
- A CRC value is computed on the scrambled data and sent with them.
- The BCID number is not sent directly. It is encoded by means of 2 pseudo random predefined sequences.

# 4 User Block Specification

## 4.1 Introduction

User block consists from following sub-blocks:

1. Pedestal subtraction (PED)
2. Filtering algorithm (FIL)
3. Summing for each FEX (SUM)
4. Re-quantization for each FEX (ReQ)
5. Generate contexts of TDAQ monitoring (MON)
6. Generate the histogram for monitoring (HIST)

user code Note : may need to calculate internal conversion to set bin ID.

user code Question: which path to readout?

The user block diagram is shown in 4.1. Data (raw ADC and result of calculation) are handled by the circular buffer. First four blocks are for the energy calculation from the ADC with state machine, and the last two blocks are for monitoring. Each sub-block functionalities and specification are described below.

## 4.2 Pedestal subtraction (PED)

Functionality: Subtract the pedestal and store in the circular buffer

- Input: ADC from circular buffer, pedestal from constants register
- Output: ADC-pedestal
- Accuracy: co-efficient should be 14 bits
- Timing: xxx.

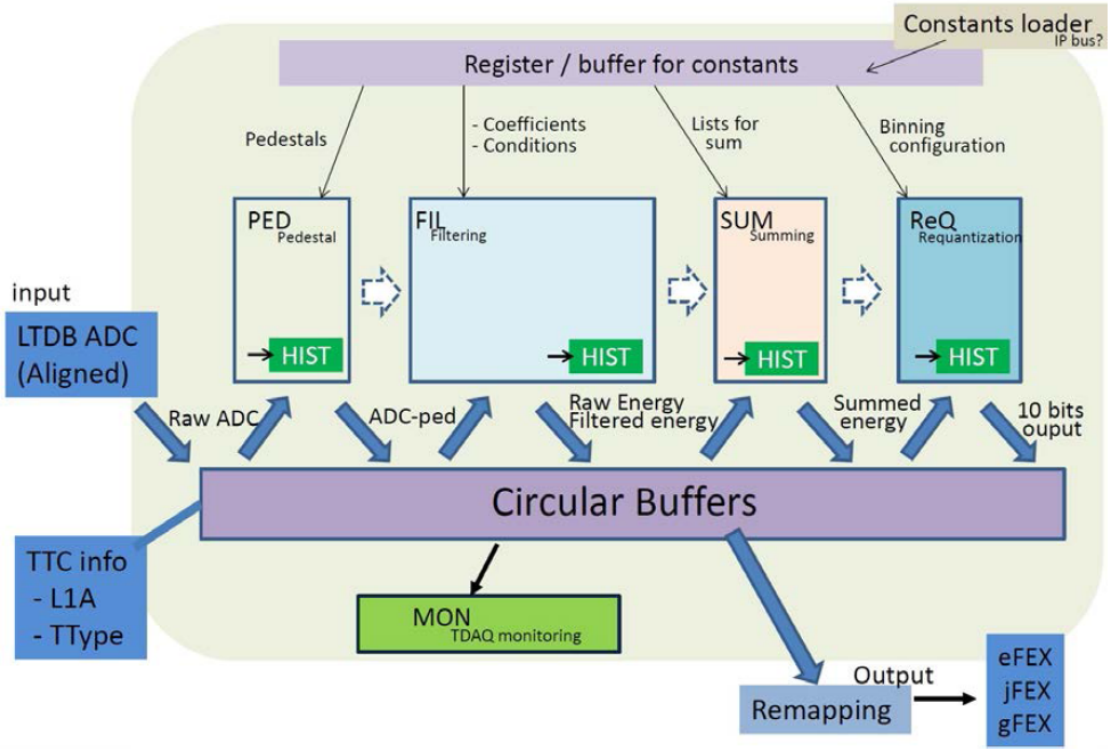



Figure 4.1: Block diagram for USER block for AMC

## 4.3 Filtering algorithm (FIL)

Functionality: Calculate energy based on the filtering algorithm

- Input: ADC-pedestal, filtering co-efficient, condition parameter
- Output; Raw energy (14 bits, before applying condition), Filtered energy (14 bits), quality bits (8 bits??)
- Accuracy: co-efficient should be 14
- Multiplication would be upto 36 bits (may be omit by compilation)
- Sign(1bit)+integer(5 bits)+fraction (11 bits)
- Timing: xxx

## 4.4 Summing for each FEX (SUM)

Functionality: Calculate sums on the energy for jFEX and gFEX.

- Input: (Filtered) Energy, list of channels to be summed
- Output: (Filtered) Energy (14 bits) for jFEX and gFEX, (and copy for eFEX)
- Accuracy: Keep 14 bits.
- Timing: xxx

## 4.5 Re-quantization for each FEX (ReQ)

Functionality: Convert 14 bits to 10 bits on the energy to be sent to FEX.

- Input: Summed energy, configuration for quantization
- Output: Energy for eFEX, jFEX and gFEX.
- Accuracy: 10 bits
- Timing: xxx

## 4.6 Generate contexts of TDAQ monitoring (MON)

Functionality: Extract samples into buffer to be sent to TDAQ path

- For event particular with TType or all L1A.
- Input: Stored data in circular buffer, L1Accept, TType, parameter for monitor
- Output: certain samples for stored data in circular buffer. (ADC, ADC-ped, Energy Raw, Filtered Energy, summed Energy) Option: specify particular channel
- Accuracy: 14 bits.
- Timing: xxx

## 4.7 Generate the histogram for monitoring (HIST)

Functionality: define histograms and fill, and respond to the request to readout.

- Usable for pedestal calculation, monitoring, etc.
- Input: ADC, energy, hist type, flag, read request
- Output: hist type and entries.
- Accuracy: 32 bits per bin?

# 5 Storage

missing storage text

# 6 Reconfigurable Mapping

## 6.1 The mapping goal

While data from every single SC is being processed in the User block independently in the User block it needs grouping before sending it further to the L1Calo. There are three types of the grouping depending on the destination: eFEX, jFEX and gFEX. These grouping include arrangements of the SCs within specific $\eta-\phi$ area and possible summing.

- For eFEX we need to arrange calculated data from 10 SCs belonging to the same Trigger Tower in one group. The data from two TT is sent over one fiber. 16 fibers are enough to send all data from 32 TT being processed in one AMC. Since the duplication of the data is needed, 32 fibers are needed in total.
- For jFEX we need to sum all energies belonging to the same Trigger Tower, to group energy sums from 16 TTs and send the data further. Two fibers are enough to send all data from 32 TT being processed in one AMC. Since the duplication of the data is needed, 4 fibers are needed in total.
- For gFEX we need to sum all energies belonging to the region of 0.2 x 0.2 of $\eta-\phi$ area and send over single fiber.

Since the calorimeter has areas with non-uniform channels map and special cases on borders, we expect that the SCs map on the AMC input will not be similar for all AMC boards. Thus for different AMC boards different mapping scheme might be needed.

One possible way to realize the number of mapping schemes is to implement a number of different FPGA designs (in the mapping sense) that accommodate all possible mapping cases. These versions of the designs will be compiled to different FPGA bitstreams and will be loaded where they should during the system initialization. While this approach looks very simple, it needs controlling that FPGA design timing constrains are met during every design recompilation. And this might be a problem during design tuning when every minor design modification could lead to a number of compilation & checking procedures.

Another approach is so-called Reconfigurable mapping. In this case we have one universal FPGA design. This design supports every possible mapping scheme and the way the mapping is performed in specific AMC depends on some configuration data that is loaded to

the FPGA registers or FPGA configuration RAMs during the system initialization. This configuration data needs to be worked out only once and be changed only in case of rearrangement of data from FE part. The obvious advantage of these approach is the single FPGA design.

## 6.2 The configurable mapping realization proposal

Configurable Remapping: missing text

# 7 TDAQ Monitoring

These are notes at the moment rather than a specification. There are several things I don't know and have started these with a Q.

The AMC card has three types of output: output for the calorimeter trigger (EFEX, JFEX, GFEX), output to TDAQ on receipt of an L1 Accept and output to the monitoring stream for debugging. The latter two outputs are discussed here.

## 7.1 TDAQ

TDAQ data is sent to the ATLAS DAQ system on receipt of an L1 Accept. The input is from a circular buffer defined below. The output is serial pair, encoded in GBT format, which is used as input to the GBT chipset.

In Phase 1, the L1 trigger latency is 2.5 $\mu$s and the maximum L1 rate is 100 kHz. In Phase 1, data is sent in response to an L1 Accept.

In Phase 2, the L0 trigger latency is 6 $\mu$s (with headroom to go to 10 $\mu$s) and the maximum L0 rate is 1 MHz. The L1 trigger latency is 30 $\mu$s (with headroom to go to 60 $\mu$s) and the maximum L1 rate is 400 kHz.

**Q:** In Phase 2, what is to be done? Send data on receipt of L0 Accept? Send data on receipt of L1 Accept? Transfer L0 Accept data into a second buffer for later transmission on L1 Accept? The latter is not ideal since synchronization is then lost.

For the moment, we will focus on Phase 1. TDAQ data from each AMC card is sent to a GBT, where it is subsequently sent to FELIX and then on to the ATLAS DAQ system. Several data formats must be accommodated.

**Q:** Is it necessary to have all possible data formats in the firmware or can there be separate firmware builds for the different data formats?

In normal operation, the data sent to TDAQ are the transverse energy (ET) outputs from the FIR filter. The ET outputs are 10 bits but they can be stored in 16 bit words. The bandwidth is 320 SuperCells (SC) x 16 bits x 100 kHz = 0.5 Gbps. This easily fits in the approximately 5 Gbps bandwidth of the GBT. One should also send a few words of header information including the local bunch crossing identification (BCID) corresponding to the L1 Accept. The data will be sent in the GBT mode (format) of the GBT (need to verify) and transmitted to the physical GBT chipset over a differential pair.

A second TDAQ mode is for calorimeter noise readout. In this mode, both the raw SC data and filtered transverse energy data are transmitted. Both are sent for 10 bunch crossings (BC's).

**Q:** What determines the 10 BC's? Why not 5? 2? 20?

The data is collected at 1 Hz L1 Accept rate (RNDM) so the amount of data to be transmitted is 320 SC x 2 (raw and FIR) x 16 bits x 10 BC's x 1 Hz = 0.1 Gbps. This is easily within the GBT bandwidth. The data would be accompanied by a header including the local BCID.

A third TDAQ mode is used to compare the transverse energy FIR calculations to that determined using the readout of individual calorimeter cells. One could imagine using heavily prescaled jet triggers in special runs for L1 Accept for this case. The data to be sent would be both raw SC data and filtered transverse energy data. The bandwidth would be similar to second TDAQ mode.

**Q:** In the TDR, it is proposed to apply cuts of ET (SC) > ET(thresh) on the data. While this can be done it is an additional complication in the firmware in terms of packing data to be sent to the GBT. It needs to be understood whether this is a requirement and whether it is desirable for a firmware standpoint.

**Q:** All, or at least many, of the use cases need to be specified in writing by the LAr group before the firmware is written.

### 7.1.1 The pseudo-firmware is ...

## 7.2 Monitoring

The function of the monitoring stream is to provide validation of the data processing on the AMC cards. The monitoring data is meant to be sent over the 10 GbE ATCA Zone 2 fabric though an ATCA switch to a host PC or farm. It is not part of the normal ATLAS TDAQ system. The input is from a circular buffer described below. The output is to four XAUI lanes that connect the AMC FPGA to the Carrier Card FPGA. The transmitted data is in the form of jumbo UDP packets. The aggregate bandwidth is 10 Gbps.

In normal operation, monitoring data is sent to a host PC or farm. The "stimulus" for sending data could a prescaled L1 Accept trigger or a RNDM one. One monitoring mode would send the raw data of SuperCells, the results of the transverse energy and time calculations and local BCID in response to the "stimulus". There would be UDP and data header information sent as well. The data to be sent would be two BC's before and after the peak BC. In this case the bandwidth would be 320 SC x 3 (raw, transverse energy, time) x 5 (BC's) x 16 bits x S, where S is the stimulus rate. An S of 10 Hz gives a bandwidth of 0.76 Gbps. While this is below the 10 GbE bandwidth, recall that four AMC's feed the Carrier Card FPGA/Zone 2 fabric and there are multiple Carrier Cards in the ATCA crate.

Another mode of operation would see an oscilloscope type of data output. In this case, data from 32 BC's would be sent for a sample of SuperCells. The data to be sent might be 80 SC x 2 (raw and transverse energy) x 16 bits x 32 BC's x S, where S is the stimulus rate. An S of 10 Hz gives a bandwidth of 0.82 Gbps.

**Q:** All, or at least many, of the use cases need to be specified in writing by the LAr group before the firmware is written. The hard part of this firmware is packing the data into jumbo UDP packets. This is driven by the S, the number of SuperCells and the different forms of data to be sent. An additional cut on ET (SC) > ET(thresh) has also been proposed for the data. While this can be done it is an additional complication in the firmware in terms of packing data in UDP packets.

### 7.2.1 The pseudo-firmware is ...

## 7.3 Circular Buffers

# 8 Slow Control Firmware Module

## 8.1 Slow Control interface

The LAr firmware working group has decided to use the IP bus protocol over Ethernet as transport mechanism for the slow control interface, the connection between the Advanced Mezzanine Cards (AMCs) and the ATLAS Detector Control System (DCS). Since our colleagues from the Compact Muon Solenoid (CMS) experiment already developed an IP bus controller firmware, we should check if their firmware fits our needs. Their IP bus controller interfaces an IP bus over Ethernet to an adjustable, synchronous System on Chip (SoC) bus. This SoC bus protocol is based on the Wishbone SoC protocol and it is almost in compliance with the Wishbone specifications [5]. All modules which needs to exchange data with the DCS should be connected to this point-to-point SoC bus. An additional module which will monitor some common Field Programmable Gate Array (FPGA) wide values e.g. voltage, temperature, etc, should also be implemented.

## 8.2 ATLAS Detector Control System

The general organization of the Atlas DCS is described in [6]. The Finite State Machine (FSM) mechanism is described in chapter 7.1 therein. It is a framework used to model the behavior of a system by means of limited number of states, transitions between states, actions and events. The DCS is designed to

- be flexible and platform-independent,
- be fault-tolerant and allow remote diagnostics,
- keep the dependence of low-level control procedures on external services such as computer networks as small as possible.

The modules of the LAr firmware should observe the ATLAS FSM states via the IP Bus controller and react on changes if necessary.

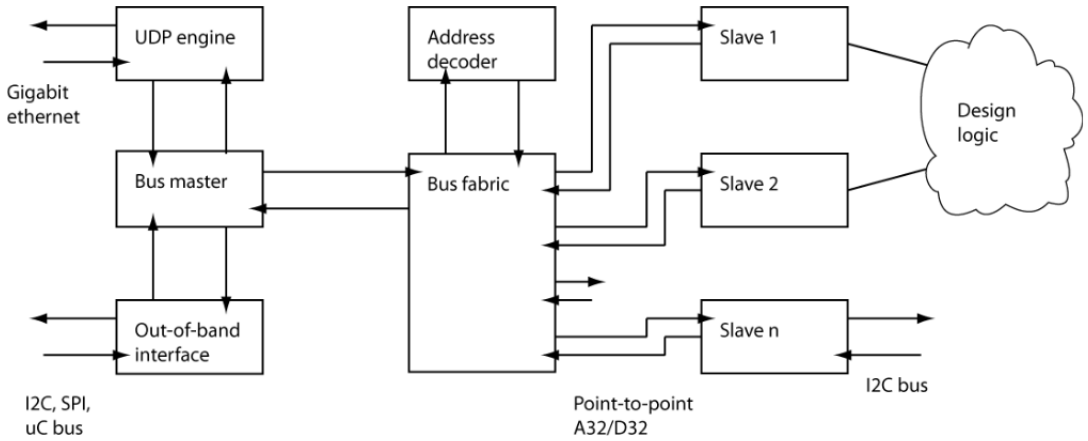



Figure 8.1: Schematic view of the SoC bus topology. The bus is implemented as a set of point-to-point signals. The figure is taken from the CACTUS firmware documentation [9]

## 8.3 IP Bus Protocol and CACTUS firmware

The proposed IP Bus Controller can be found at Code Archive for the The UpgradeS (CACTUS). The archive also contains a documentation of version 2.0 of the IP bus protocol [7]. The firmware directory [8] is located at CERN and contains implementations for different development boards and some example slave designs which we could use as a basis to interface our modules. Figure 8.1 shows a schematic view of the SoC bus topology.

# List of Figures

# List of Tables

# Bibliography

[1] missing reference 14

[2] FPGA ARRIA10 documentation:

[3] ARRIA10 DDR3 EMIF, download: emi_fd_a10_emif.pdf 3

[4] Avalon Interface, download: mnl_avalon_spec.pdf 5

[5] Wishbone specifications, download: wbspec_b4.pdf 25

[6] A. Barriuso Poy, H. Boterenbrood, H.J. Burckhart, *et al.*, The detector control system of the ATLAS experiment, JINST **3** (2008) P05006. download: 1748-0221_3_05_P05006.pdf 25

[7] The IPbus Protocol, download: ipbus_protocol_v2_0.pdf 26

[8] link to the CACTUS IP bus controller firmware directory 26

[9] Notes on Firmware Implementation of an IPbus SoC Bus, download: IPbus_firmware_notes.pdf 26