ALICE Internal Note

C.E.R.N. Wigner R.C.P.

ALICE COMMON READ-OUT UNIT ALICE-CRU

CRU Specification

Issue:	Draft
Revision:	v0.7
Reference:	CRU
Created:	08/31/2015
Last modified:	14/06/2016
Prepared by:	E. David, T. Kiss, F. Costa, J. Imrek for the CRU team

Abstract

This document specifies the ALICE CRU (Common Read-out Unit): the actual PCIe40 CRU hardware, the different communication forms and usage scenarios between the CRU and the detector FEs (Front-Ends) over the GBT link and the internal firmware User Logic Interface in details.

1. Document Title: CRU Specification					
2. Document R	eference Numbe	er: CRU			
3. Issue	4. Revision	5. Date	6. Reason for change		
Working Draft	v0.3	22 January 2016	Initial thinking		
Working Draft	v0.4	01 March 2016	Evolution of the thinking		
Working Draft	v0.5	09 March 2016	Chapter 3.7 updated Chapter 4 updated		
Working Draft	v0.6	14 March 2016	Chapter 3.1 rephrased Chapter 3.2 moved to Glossary at the end Point 3.4.3 updated Chapter 3.7 removed Chapter 4 updated		
Working Draft	0.7	14 June, 2016	All Chapters updated		

Document Status Sheet

Table of Contents

1	Introd	luction	5
2	CRU	Hardware Specification	5
	2.1 Mai 2.1.1 2.1.2 2.1.3 2.1.4 2.1.5 2.1.6 2.1.7 2.1.8 2.2 Pow 2.2.1 2.2.2 2.2.3	n components FPGA Configuration devices Optical transceivers PCIe bridge JTAG management Clocks PCIe connector JTAG connectors <i>ver supply</i> Power supply estimation Power source Power supply tree	.5 .6 .6 .7 .7 .8 .9 .9 .9 10 10 10
	2.3 Coa 2.3.1 2.3.2 2.4 Mea 2.4.1 2.4.2	ling	11 11 11 11 11 12
3	Comn	nunication over the Detector Links (GBT Links)	13
	3.1 Intr 3.2 Trig 3.2.1 3.2.2 3.2.3	oduction ger distribution related features LHC clock distribution Trigger information distribution FE read-out control distribution	<i>13</i> <i>15</i> <i>15</i> <i>15</i> <i>15</i>
	3.3 Dat 3.3.1 3.3.2 3.3.3	a Transport Related Features The Raw GBT Data Stream Using the GBT as Independent Serial Links Using the GBT as a Parallel 80-bit Interface	<i>15</i> 15 15 16
	3.4 Slov 3.4.1 3.4.2 3.5 Sup	v control related features GBTx ASIC communication GBT-SCA communication	22 22 22 22 23
4	Com	nunication over Detector Links (Custom Links)	24
-	4.1 TRI	D detector links	24
5	Comn	nunication over the Trigger Links	25
U	5.1 Intr	oduction	25
	5.2 The	Downstream Link	25
	5.3 The 5.3.1 5.3.2	Upstream Link	27 27 27
6	Comm	nunication over the PCI Express interface (Software Specification)	28
	6.1 Intr	oduction	28

6.2 Th	e CRU Driver	
6.3 Th 6.3.1	e CRU API CRU DAQ API	
6.3.2 7 CRU	User Logic Interfaces	
7.1 Int	roduction	
7.2 Po	rts Specifications	
7.2.1	TTS Downlink Ports (from LTU to CRU direction)	
7.2.2	TTS Uplink Ports (from CRU to LTU direction)	
7.2.3	GBT Downlink Ports (from CRU to FE direction)	
7.2.4	GBT Uplink Ports (from FE to CRU direction)	
7.2.5	PCIe Link Ports	
7.2.6	Avalon-MM Slave Port - PCIe BAR Read/Write access	
7.2.7	GBT-SCA Link Ports	
8 Tern	iinology	

1 Introduction

To be done.

2 CRU Hardware Specification

2.1 Main components



Figure 1: Block diagram of the PCIe40 board

2.1.1 FPGA

The FPGA is an Arria 10 10AX115S4F45I3SGES from Altera. This is the largest one in the Arria 10 family with 1150 kLE. It is pin compatible with a smaller one with 900 kLE (see Figure 2) which will allow a cost optimization if the application can fit in this matrix. This FPGA is also pin compatible with the future Stratix 10 with the same amount of cells which paves the way for a future speed upgrade if required.



Figure 2: Pin compatible migration possibilities for the selected FPGA

Compared with the FPGA implemented on the AMC40 pins we have 1.8 times more cells than before. This should make the implementation more comfortable.

The main features are the following:

- Number of Logical Elements: 1 150 000
- Internal memory: 8.4 MB
- Variable precision DSP blocks: 1518
- 18 x 19 multipliers: 3036
- Fractional PLLs: 32
- I/O PLLs: 16
- High speed serial links at 12.5 Gbits/s: 72
- Maximum speed on serial links : 12.5 Gb/s
- Maximum internal speed : 500 MHz

2.1.2 Configuration devices

An EPQS512 quad serial memory device download the configuration of the FPGA at power-up. This device can be accessed by the JTAG interface or the PCIe interface using the CvP protocol though the FPGA.

2.1.3 Optical transceivers

The board is able to interface Front-Ends with up to 48 bidirectional optical links at 10 Gbits/s each for a total bandwidth of 0.48 Tbits in each direction, even if the connection with Front-Ends is currently limited to 4.8 Gbits/s only per link. The board will be able to manage the new LP_GBT protocol when available.

This interface is implemented with 4 optical transmitters AFBR_81uVxyZ and 4 receivers AFBR_82uVxyZ from Avago also called MiniPODs as shown in Figure 3. Each of them handle 12 optical links. Each transmitter or receiver is linked to a MTP/MTP connector on the front plate.



Figure 3: MiniPOD optical interfaces from Avago

The board also embeds one SFP+ bidirectional device devoted to TFC interface. It is also possible to insert a PON device to broadcast the clock to several Front-Ends.

2.1.4 PCIe bridge

The Arria 10 FPGA on the PCIe40 card can only manage PCIe Gen3 interfaces with 8 lanes whereas the CPU needs a PCIe Gen3 with 16 lanes interface. The adaptation is made with a PCIe bridge PEX8747 from PLX.

The PEX8747 has 48 PCIe Lanes, implemented as 16 lanes per station across three stations. The stations are connected to one another by the internal non blocking fabric.

The adaptation 8 lanes/16 lanes is done by connecting two 8 lanes ports of the bridge to the FPGA and a 16 lanes port to the PCIe connector of the card as shown in Figure 4.

Two 8 lanes port are not used in this scheme.



Figure 4: PEX8747 configuration

This PCI Express bridge may be removed from the next/final version of the PCIe40 card, as up to date server motherboards can overtake its functionality, merging two times 8 lanes PCIe interfaces in a single slot into a single 16 lanes interface.

2.1.5 JTAG management

Three sources of JTAG allow to program the Arria10 GX FPGA:

- an external 10 pins connector located on the front board
- a JTAG link piloted by the PCIe
- a JTAG link piloted by an embedded USBblaster

JTAG source selection is assured by a hub implemented in a MAX V FPGA. The choice of JTAG source is made either by external straps. The overall JTAG connection is shown in Figure 5.





2.1.6 Clocks

2.1.6.1 LHC clocks



Figure 6: Clock paths on PCIe40 card^{1, 2}

¹ In ALICE CRU implementation the "120 MHz Recovered clock from TFC" will be 240 MHz. Therefore the "Filtered 120 MHz" jitter cleaned clocks fed to the 8 GBT banks will also be 240 MHz.

² On the prototype PCIe40 cards two jitter cleaner components were tested, TI CDCE62005, and SI 5338. Based on measurements SI5338 showed the better performance.

2.1.6.2 External jitter cleaner

The LHC clock is recovered from the input trigger link by internal CDR circuitry of the Arria 10 FPGA. there is an external, on-board jitter cleaner component on the PCIe40 card. This component (SI 5338) is a zero-delay PLL, with an

• output jitter of < 1.5 ps (typ: 0.7 ps) RMS.

It is followed by an 1:8 clock buffer, that fans-out the filtered clock to the 8 GBT banks of the FPGA. (Each GBT bank has a dedicated reference clock input.) The timing specification of that fan-out buffer component is the following:

- additive jitter: < 100 fs, typ: ~ 40 fs @240 MHz
- maximum propagation delay: 450 ps
- maximum output-to-output skew: 20 ps.
- maximum part-to-part skew (i.e. between CRUs): 125 ps.

2.1.6.3 PCIe Clocks

A 100 MHz clock signal coming from the PCIe connector is fanned out to the two PCIe interfaces of the FPGA.

2.1.7 PCIe connector

The PCIe connector is a 82 pins vertical edge card connectors supporting x1, x4, x8, and x16 Link widths to suit different bandwidth requirements.

Pinout and form factor are detailed in xxx (to be done).

The PCIe40 board is compliant with PCI Express CEM r3.0 specification

2.1.8 JTAG connectors

Two 10 pin JTAG headers are mounted inside the board for programming the Arria10 GX through an USB blaster or to program the JTAG hub implemented in the Max V FPGA. It is shown in Figure 7.



Figure 7: Front plate JTAG interface 10 pin header (M40-4011046 from Harwin)

2.2 Power supply

2.2.1 Power supply estimation

The overall power consumption of the PCIe40 board is estimated to be 120 W.

2.2.2 Power source

Because it overpass the maximum 75W power consumption that can be drawn from the PCIe connector, an PCI Express 2 x 4 auxiliary power connector and cable assembly will be used as specified in the PCIe specification.

2.2.3 Power supply tree

The power tree for supplying the FPGA is illustrated in Figure 8. Other devices like PLLs, temperature sensors are connected on the same tree through passive filters.



Figure 8: Power supply tree for supplying the Arria10 GX FPGA

2.3 Cooling

2.3.1 Requirements

An air flow of 2 m/s is required to maintain the FPGA temperature under the maximum bearable limit.

2.3.2 Implementation

To be defined.

2.4 Mechanics

2.4.1 Dimensions

The form factor is Standard Height (111.28 mm) Half Length (167.65 mm) PCI Express add-in card.



Figure 9: Board dimensions

2.4.2 Face plate

The layout of MTP/MTP , SFP+ , JTAG and USB connectors is shown in Figure 10.



Figure 10: PCIe40 face plate

3 Communication over the Detector Links (GBT Links)

The typical detector links are GBT links. Presently it is only the TRD detector that will connect to the CRU with a different link technology. The special case of TRD will be described in a separated chapter.

In the following chapters we will describe the detectors - CRU communication via GBT links.

3.1 Introduction

The aim of this chapter is to specify in details the different communication forms between the Common Read-out Unit (CRU) and the detector front-end electronics (FE) over the bidirectional GBT link.

CERN GBT project home page: https://espace.cern.ch/GBT-Project/



- 4 Sending and receiving packets to and from GBTx internal register block
- Sending and receiving packets to and from GBT-SCA ASIC

Figure 11: Different communication forms between the CRU and the FEs

Figure 11 shows a generic CRU and FE setup with different communication forms. The communication forms organized into three groups based on the functionality:

- Trigger related communications (clock, trigger, read-out)
- Detector data transfer related communications
- Slow control related communications

After evaluating the different detector requirements the following required GBT downlink and uplink feature usages has been identified:

GBT Downlink features:

- 1. LHC clock distribution The CRU delivers the 40 MHz LHC clock with deterministic phase to the FE over the GBT downlink.
- Trigger information distribution The CRU receives the full TTS information from the LTU (up to 192-bit) and is able to deliver some subset of this trigger information (up to 80-bit) over the GBT downlink with deterministic latency.
- FE Read-out Control distribution Some detectors will implement the readout control not locally on the FE cards, but inside the CRU. In this case, instead sending the raw trigger information to the FE they will generate and send any custom read-out control signals with deterministic latency.
- 4. Parallel 80-bit packet based communication Unidirectional packet based communication from CRU to FE.
- 5. Parallel 80-bit single word communication Unidirectional single word based communication from CRU to FE.
- 6. GBTx ASIC control communication GBTx control information sending from CRU to GBTx ASIC on the FE card. (GBT frame IC field)
- GBT-SCA control communication FE control information (HDLC packet) sending from CRU to GBT-SCA ASIC on the FE card. (GBT frame EC field)

GBT Uplink features:

- 1. Raw GBT data stream Interpreted as 80/112-bit parallel word.
- 2. Independent serial links Raw 80/112-bit GBT data stream interpreted as independent serial links with custom protocol.
- 3. Parallel 80-bit packet based communication Unidirectional packet based communication from FE to CRU.
- 4. Parallel 80-bit single word communication Unidirectional single word based communication from FE to CRU.
- GBTx ASIC control communication Receiving the response from the GBTx ASIC on the FE card. (GBT frame IC field)
- 6. GBT-SCA control communication Receiving the response from the GBT-SCA ASIC on the FE card. (GBT frame EC field)

GBT Downlink	TPC	МСН	MID	TOF	FIT	ZDC	ITS	MFT	TRD	СТР
1. LHC clock distribution	х	х	х	x	?	?	х	х		
2. Trigger information distribution				x	?	?				
3. FE read-out control distribution	х	x	x							
4. Parallel packet based communication							x	x		
5. Parallel single word communication							х	x		
6. GBTx ASIC control communication	х	х	х		?	?	х	x		
7. GBT-SCA control communication	х	х	х		?	?	x	х		
GBT Uplink										
1. Raw GBT data stream	x								X (custom)	
1. Independent serial links		х	х							
2. Parallel packet based communication				x	?	?	x	x		x
3. Parallel single word communication							x	x		
4. GBTx ASIC control communication	х	х	x		?	?	x	x		
5. GBT-SCA control communication	х	x	x		?	?	x	х		

Table 1 summarizes the different GBT downlink and uplink feature utilizations by different detectors.

 Table 1: Detectors GBT Links feature chart

3.2 Trigger distribution related features

The following chapters describe the trigger distribution related features in details. Each feature is unidirectional and handled by the GBT downlinks.

3.2.1 LHC clock distribution

The CRU firmware is able to receive the 40 MHz LHC clock information from CTP/LTU and reconstruct the extracted LHC clock with known phase relation to the origin. The firmware is able to synchronize the GBT frame header to the LHC clock rising edge and propagate the LHC clock information to the FE over the GBT downlinks.

The LHC clock distribution feature has no payload overhead, the latency optimized GBT-FPGA implementation itself provides this feature. In addition to this, the CRU clock distribution network ensures that the phase differences between the clocks transported over the GBT link remains stable over power cycles and firmware upgrades.

3.2.2 Trigger information distribution

In this mode the CRU will propagate a subset of the whole TTS information received from CTP/LTU over the GBT downlink (80-bit out of 192-bit). This feature will be implemented in the generic (common) CRU firmware.

This mode can be used when there is a capable onboard logic on the FE side and able to work based on the raw TTS information.

3.2.3 FE read-out control distribution

In this mode the TTS information will be not sent directly to the FE, but instead to a detector specific firmware module inside the CRU firmware and that detector specific logic will control the actual GBT downlink depending on the specific FE needs.

This mode can be used when there is no capable logic on the FE to but only a dedicated ASIC like logic and the more complex readout control can be moved to the CRU side.

3.3 Data Transport Related Features

The following chapters describe the data transport related features in details. Two main modes have been identified: raw data stream mode and the packet based mode.

3.3.1 The Raw GBT Data Stream

In this mode the raw GBT payload (80/112-bits @ 40 MHz) is not processed by the core CRU firmware but passed directly to the detector specific user logic module inside the CRU firmware. This mode allows any custom communication form over the GBT uplink between the FE and the CRU.

3.3.2 Using the GBT as Independent Serial Links

In this mode each E-Links field in the GBT payload contains a fragment from an incoming serial packet coming from a front-end ASIC or FPGA. Since the serial packet format is detector specific this GBT

payload will be handed to the detector specific user logic module inside the CRU firmware as in the case of the "Raw data stream" mode.

3.3.3 Using the GBT as a Parallel 80-bit Interface

3.3.3.1 Distinction of Data and Control Words over the GBT

Each transmitted 80-bit GBT payload can be considered as a 80-bit GBT word at a parallel interface of 80 bits. In addition, each such GBT words transmitted over the link can be classified as *valid data word* or else by the inherent *"data valid"* feature of the GBT link.

Based on this mechanism, *data words* and *control words* can be distincted. If the data valid bit is 1 then the 80-bit GBT word must be treated as a data word, part of the data transfer. If the data valid is 0 then the 80-bit GBT word must be treated as a control word where the exact control word *type* is encoded in the "Control Code" field of the word.

Figure 12 and Figure 13 shows the GBT DATA and CONTROL word structure.



Figure 12 The 80-bit GBT word marked as DATA



Figure 13 The 80-bit GBT word marked as CONTROL

Currently four control codes are defined: IDLE, SOP, EOP and SWT. (See *Figure 14*, *Figure 15*, *Figure 16*, and *Hiba! A hivatkozási forrás nem található.* in the following Points).

3.3.3.2 CRU Packet Based Protocol

Packets will be defined by sending *packet delimiters*, namely Start of Packet (SOP) and End of Packets (EOP) with the *packet payload* in between. With different kind of (flavours of) delimiters different kind of (flavours of) packets can be defined, e.g. data packets, control packets, service packets, etc. Payloads of different kind of packets can be then forwarded to different kind of buffers, e.g. data packets to data buffers, control packets to control buffers, a.s.o.

In all cases, if no packets are sent, IDLE control words shall be sent over the GBT links. Presently there is no limitation on the length of the packets sent over the GBT link, and also the number of IDLES are not restricted, and can be even zero.

Presently only one kind of packets are defined, namely *data packets*, but there are reserved values of control codes to be able define more kind of SOPs and EOPs if need be in the future.



Figure 14: IDLE control word format

Bits	Name	Description	
[3:0]	Control Code	IDLE = 0	
[79:4]		Reserved (must be zero)	



Figure 15: Start of Packet (SOP) control word format

Bits	Name	Description
[3:0]	Control Code	SOP = 1
[19:4]	Length	Length of the packet (optional)
[35:20]	TTS Busy	Busy information bits
[79:36]		Reserved (must be zero)



Figure 16 End of Packet (EOP) control word format

Bits	Name	Description
[3:0]	Control Code	EOP = 2
[19:4]	Length	Length of the packet (optional)
[51:20]	Checksum	Checksum of the packet (optional)
[52:52]	End Flag	End of packet flag (1 = Yes, 0 = No)
[79:53]		Reserved (must be zero)

Presently only one kind of packets are defined, namely *data packets*, but there are reserved values of control codes to be able to define more kind of SOPs and EOPs, if need be in the future.

3.3.3.3 Packet based data transfer from the FEE to the CRU

In this point it is specified how the CRU packet based protocol can be used for physics data transmission from the FEEs to the CRUs. In this point, the term 'packets' refer to *data packets*, one possible type of packets that can be defined in the CRU packet based protocol. (Presently this is the only kind of packets defined.)

The CRU shall also be able to receive ALICE standard, formatted data packets and transfer them into the memory of the FLP servers. This will be possible by the means of the CRU packet based protocol, embedding the ALICE standard, formatted data packets into the payload of the CRU data packets delimited by SOP and EOP control words.

Note: This protocol is similar to the existing proprietary ALICE DDL protocol where the FEEs have been able to send well formed, ALICE standard, packetized data over the DDL links toward the server computers. These ALICE standard data packets have had a well defined internal structure. Data payload is transmitted after an ALICE wide common data header (CDH), and the payload (packets) had a maximum possible length.

In the upgraded system, Single Data Headers (SDH) will replace the former ALICE Common Data Headers (CDH). The format of these ALICE wide standard data headers (SDH) will be defined by the Computing Work Group 4 of the O2 project.

Figure 17 shows an example of FEE-CRU data packet transmission. The CRU packets must start on GBT word aligned (80-bit) position. The packets start with an SOP control word, followed by the CRU packet payload. In case of transmission of standard ALICE data packets, the CRU payload starts with one or more header words (ALICE CDH/SDH), followed by the physics data as the actual payload. Then the CRU packet ends with an EOP control word.



Figure 17 Example of packet transmission

The main properties of the CRU packet based protocol are the following:

- Unidirectional Due to the very diverse GBT downlink/uplink usage scenarios requiring a bidirectional communication (e.g. acknowledges) between the FE and the CRU would be a limiting factor.
- Requires a capable FPGA on the FE card which is able to build standard ALICE data

Figure 18 shows the packet sending sequence from the FE to the CRU in detailed steps.



Figure 18 FE to CRU data packet transfer

Figure 19 shows the packet sending sequence from the CRU to the FE in detailed steps.



Figure 19: CRU to FE data packet transfer

3.3.3.4 Unsegmented Continuous data streaming to the CRU

Continuous vs. triggered read-out of detectors

The continuous read-out mode of some detectors will be a substantial change from current practice. In this case the physics data stream will not be delimited (segmented) by physics triggers but it is composed of free-running continuous data stream(s) transmitted off the detector. However the online system cannot handle infinite data streams without internal time structure.

To solve this problem, artificial heart beat triggers or 'heart beats' will be generated by the trigger system at constant frequency (e.g. 10 kHz), used for chopping the continuous data flow into manageable Heart Beat Frames (HBF).



Heart Beat Frames(HBF): stream of data delimited by two heart beats



Packetized vs. continuous input streams to the CRU

From the CRU point of view, this classification looks a bit different. Some of the detectors (e.g. TOF), that has enough data flow processing capability locally on the FE and/or read-out cards, sends time segmented data flow off the detector (to the CRU) by sending data packets (with ALICE-standard content) at every trigger signal. This can happens both for physics triggers ("triggered" read-out of detectors) and for heart beat triggers ("continuous" read-out of detectors). Within the packets the data is tagged with event identification.

Other detectors (e.g. TPC), where there is no such local intelligence on the FE and read-out cards, can send continuous, unsegmented flow of data to the CRU only, in which the incoming flow is not delimited neither by physics or heart-beat triggers. In this case it is the responsibility of the CRU to segment and packetize the incoming continuous stream and tag the data in the packets adding event identification.



The concept of this conversation of unsegmented, continuous input streams to data packets with event identification can be seen in Figure 20.

Figure 21: Conversion of continuous input stream to data packets

This input data handler in the CRU reads out the GBT user interface at a 40 MHz interface clock. At each clock cycle an 80-bit (GBT in GBT mode) or 112-bit (GBT in Wide Bus mode) GBT word (data word) is read. As the writing clock to the data handler (40 MHz) is lower than the writing clock from the data handler to the CRU User Logic, (240 MHz), it is possible to add/insert extra words (e.g. delimiters, headers with event identifications, etc) in to the continuous flow, without loosing any data.

3.3.3.5 Single Control Word Transactions (SWT)

While the packed based data transfer can be optimal for the transmission of large detector data packets in data taking, there are some other cases when a simpler protocol would be more suitable for the communication between the CRU and the detectors FEE. Typical example is control communication between FEE and CRU, e.g. sending commands to the FEE, or writing/reading of FEE registers from the CRU. For such purposes, a single word transaction is also included in the CRU specification. (This can also be considered as a special case of the packet based communication where the 'packets' have no payloads and a special "SOP" in itself makes the communication. That is, the transaction is reduced to transmitting a single control word only, this is where the Single Word Transactions (SWT) comes from.)

With the 4-bit control codes of the SWT, and the 76-bit parameters, a large variety of single control words (i.e. single word transactions) can be defined on the user level.



Figure 22: Single Word Transaction (SWT) control word format

Bits	Name	Description
[3:0]	Control Code	SWT = 3
[79:4]	Transaction Parameters	Custom transaction parameters (incl. transaction ID)

Figure 23 shows the single word transaction steps between the CRU and the FE.



Figure 23: CRU to FE Single Word Transaction

3.4 Slow control related features

The following chapters describe the supported slow control related features. The GBT frame contains two dedicated bit field for the slow control purposes: the 2-bit IC filed for GBTx ASIC communication and the 2-bit EC field for GBT-SCA communication. The common CRU firmware supports both channel and also enables to the FE to utilize the packet based protocol for slow control.

3.4.1 GBTx ASIC communication

In the case when the GBTx ASIC configSelect (pin K7) tied to 0 there is a possibility to access the GBTx ASIC internal registers from CRU firmware through the 2-bit IC field. The CRU firmware and tools fully support this feature.

3.4.2 GBT-SCA communication

The GBT frame contains 2 dedicated bits for external slow control applications (the EC field). This 2bits @ 40 MHz field is translated to a bidirectional 80 Mbps E-Link:

- 80 MHz ePort clock output: SCCLKN (pin R6), SCCLKP (pin P6);
- 80 Mbps ePort data input: SCINN (P13), SCINP (P12);
- 80 Mbps ePort data output: SCOUTN (P3), SCOUTP (P4)

The CRU firmware and tools fully support the communication with GBT-SCA over this dedicated E-Link.

3.5 Supported FE Architectures

The CRU firmware supports the most common detector front-end architectures built of CERNdeveloped radiation tolerant building blocks (e.g. Versatile Link VTRx and VTTx, GBTx ASIC, GBT-SCA), custom front-end ASICs, (e.g. SAMPA, ALPIDE, etc.) and/or commercial programmable logic devices (e.g. Xilinx, Altera, Microsemi FPGAs).



Figure 24: Typical front-end card architecture

Figure 24 illustrates the typical FEE architecture. The CRU provides 12/24/36 fully bidirectional GBT links but it is not mandatory for the FE cards to utilize both directions of them. A typical FE card will be based around at least one bidirectional VTRx module and several additional VTTx transmit only module.

The primary GBTx ASICs configuration/reconfiguration is supported through the GBT's IC (Internal Control) channel. The additional GBTx ASICs should be connected to the GBT-SCA I2C master ports to enable the configuration/reconfiguration through the CRU and in this case the configuration/reconfiguration goes through the GBTs EC (external control) channel.

4 Communication over Detector Links (Custom Links)

After the upgrade of the ALICE read-out and trigger system, the upgrading detectors will connect to the Common Read-out Units with GBT links. Presently there is one exception, the TRD detector which will connect to the CRU with legacy links.

4.1 TRD detector links

TRD will preserve its 2.5 Gb/s, 8B/10B unidirectional optical links between the detector FEE to the CRUs. Instead of the GBT-FPGA firmware modules, detector specific TRD link receiver modules will be implemented in the CRU FPGA by the detector team, that will then convert the incoming data flow to the same uplink ports of the CRU GBT links - User Logic interface. Downlink ports of the User Logic - GBT links interface will not be used as there are no downlinks to the TRD detectors.

T. b. d. in more details.

5 Communication over the Trigger Links

5.1 Introduction

During Run3 there will be a "dedicated trigger link" between the CTP/LTU and the CRU. The main purpose of this link is to provide the LHC clock and trigger information for the CRU and handle the detector busy state and detector data transfer throttling on the system level.

Currently the candidate for this trigger link is the CERN developed TTC-PON system which is based on the 10G PON technology.

5.2 The Downstream Link

The TTC-PON Downstream link (OLT -> ONU direction) is used to distribute the 40 MHz LHC clock, trigger and throttling information from LTU to CRU.

The main parameters of Downstream link:

- 9.6 Gb/s (6.4 Gb/s user)
- 1:N splitting
- broadcasted PON frames (240-bits @ 40 MHz)
- 160-bit user payload per bunch clock period
- ~ 86 ns latency



Figure 25: Downstream PON frame

The PON frame stream is continuously transmitted from the LTU. Each PON frame contains 24 8B/10B encoded symbols where 20 symbols are allocated for user purposes (see Figure 25). The structure of the transmitted trigger and throttling information is specified by the Central Trigger Team as the following (see Table 2 and Table 3):

Data	PON Byte	Payload	Content
[07]	1	[07]	ТТуре
[07]	2	[815]	ТТуре
[07]	3	[07]	BCID
[07]	4	[811]	BCID
[07]	5	[07]	Orbit
[07]	6	[815]	Orbit
[07]	7	[1623]	Orbit
[07]	8	[2431]	Orbit

 Table 2: PON Frame Fields

Bit	Name	Description
1	Orbit	Orbit
2	НВа	HB accept
3	HBr	HB reject
4	PP	Prepulse
5	Cal	Calibration
6	Start/End	Bit is 1 for start of data, 0 for end of data (used with bits 7 and 8)
7	SOT/EOT	Bit is 1 for Triggered Data
8	SOC/EOC	Bit is 1 for Continuous Data
9	TOF	TOF special trigger to signal transferring data from FM to DRM
10-16		Spare

Table 3: Trigger Types

5.3 The Upstream Link

The TTC-PON Upstream link (ONU -> OLT direction) is used to propagate the busy and throttling information (Figure 26) back from the CRU to the LTU.

5.3.1 The main parameters of the trigger upstream link:

- 2.4 Gb/s shared bandwidth, 8 Mbps (64 ONU's)
- 48-bit user payload
- Waiting time per ONU: 8 us (64 ONUs), 4 us (32 ONUs)
- 12 dB dynamic range

5.3.2 Busy / drop / read-out throttle control

The CRU is a key component of this read-out control mechanism. The control loop is implemented by the CRUs, the trigger uplinks (via the LTU - not shown in the Figure 26), the Central Trigger Processor (CTP), and the trigger downlinks (via the LTU - not shown in the Figure 26). Drop decisions are made on the level of data packets, namely the Heart Beat Frames (HBF) for the continuous read-out of detectors. *Figure 26* shows different types of messages in the control loop.

A detailed description of the busy signaling, packet drop and read-out throttle mechanism in the upgraded ALICE read-out system is specified in the ALICE Technical Note ALICE-TECH-2016-001, "The detector read-out in ALICE during Run 3 and 4", working draft, v1.5 (June 3, 2016), A. Kluge, P. Vande Vyvre; CERN, EP Department.



Figure 26: Busy / Drop / Throttling Signal and Message Flow

6 Communication over the PCI Express interface (Software Specification)

6.1 Introduction

Figure 25 shows the different software layers and the underlying hardware interfaces needed to communicate through the Common read-out Units.



Figure 27: CRU Software Stackup Overview

6.2 The CRU Driver

The CRU uses the PDA (Portable Driver Architecture) driver. It resides mostly in user space and only has a small, easily maintainable adapter in the kernel.

It provides the low-level enumeration, access and control of the device:

- Access to PCI registers (BAR)
- DMA memory handling, with support for:
 - Persistent buffers
 - User space buffers
 - o Buffer sharing
 - Scatter-gather lists
 - o IOMMU protection and consecutive mapping
 - Wrap mapping
 - NUMA control
- Interrupt handling for MSI and INTx interrupts
- Synchronization support for multi-threading

6.3 The CRU API

The CRU API layer is a purely user space library. It is based upon RORC & DDL library code, adapted to use PDA functions.

The CRU API layer provides:

- Controlled access to the driver features
- Initialization, configuration and control of the device, such as: configuration, checking FIFO status, data generator control.
- Low-level DMA data transfer management functions
- Low-level C functions for accessing the register based interfaces provided by the CRU firmware (for example functions accessing the GBT-SCA I2C masters and JTAG port)
- Definitions of CRU registers, status codes, command codes, etc.

6.3.1 CRU DAQ API

The CRU DAQ API rests on top of the CRU API layer and provides a high-level C++ interface to access the data transfer functions of the CRU. These are available through an object representing a CRU device, which the user can create by specifying a serial number. This object can then be used to

- Open DMA channels
- Push pages
- Wait for pages to arrive
- Access memory associated with arrived pages
- Mark arrived pages as read, allowing them to be pushed to again

In addition, the interface provides some low-level API functions to

- Read and write registers.
- Reset the card
- Retrieve the raw memory address of a DMA buffer

While the lower-level C layers rely on return codes for error handling, the C++ API uses exceptions. DMA configuration parameters are also verified for errors or conflicts before being passed on to the lower layers.

6.3.2 CRU DCS API

It is mandatory for the safety of the experiment that the DCS must be able to control and check the status of the FEE even when the data taking is not active. For this reason, a dedicated communication channel between DCS and CRU is established. The DCS system will configure and monitor the FEE using the CRU card connected to several GBT links. A software interface provides the communication channel for the DCS to control the CRU and execute the following main functions (more functions may come in future, following the evolution of the system):

- Download configuration data to the FEE.
- Read and monitor FEE parameters (temperature, current or voltage).

. The CRU DCS API together with the FLP DCS software will provide the needed infrastructure to create the connection.

The CRU DCS API will give access to the following low level operations:

- WRITE a register to the CRU.
- READ a register from the CRU.

The CRU low level API interface will be hidden to the DCS framework. The low details of the interface to open the communication with the card and initialize the data transfer will be handled completely by the CRU DCS API. The interface will also provide information to communicate the status of the current command execution. In case of error during the configuration of the FEE the software will propagate it to the DCS system.



Figure 28: CRU Software Stackup Overview

6.3.2.1 FLP DCS software

The FLP DCS software is a bridge between the FLP and the DCS system.

It provides:

- Communication over the network with the DCS framework, using DIM.
- READ/WRITE access to the CRU, using the CRU DCS API.

This software is the same running in all the FLP, it doesn't have knowledge of the detector configuration, it only provides the communication protocol to transfer data from the DCS to the CRU.

7 CRU User Logic Interfaces

7.1 Introduction



Figure 29: CRU Firmware user logic interfaces

The main CRU firmware User Interfaces will be the (see Figure 29):

- 1. TTS Links: The TTS Downlink provides the User Logic with the raw trigger information and the TTS Uplink accepts the Busy information from the User Logic.
- 2. GBT Links: The GBT Downlinks delivers the payload produced by the User Logic to the frontend and the GBT Uplinks delivers the payload sent by the front-end to the User Logic.
- 3. PCIe Link: Provides the high speed DMA communication interface between the User Logic and the FLP Server.
- 4. Slow Control Links: The Avalon-MM Slave interface allows the User Logic accessibility from the FLP server. The firmware level GBT-SCA interface allows direct SCA access from the User Logic in parallel with the access from the FLP server.

7.2 Ports Specifications

7.2.1 TTS Downlink Ports (from LTU to CRU direction)

Port Name	Direction	Clock Domain	Description
tts_rx_clk_i	Input	N/A	Trigger RX clock
tts_rx_clk_rst_i	Input	tts_rx_clk_i	Trigger RX clock reset
tts_rx_clk_src_i	Input	tts_rx_clk_i	Trig. RX clock source
tts_rx_valid_i	Input	tts_rx_clk_i	Trigger bits are valid
tts_rx_data_i[N-1:0]	Input	tts_rx_clk_i	Trigger data bits

7.2.2 TTS Uplink Ports (from CRU to LTU direction)

Port Name	Direction	Clock Domain	Description
tts_tx_fifo_clk_o	Input	N/A	Trigger TX FIFO write clock
tts_tx_fifo_full_i	Input	tts_tx_fifo_clk_o	FIFO full signal
tts_tx_fifo_wr_o	Output	tts_tx_fifo_clk_o	FIFO write signal
tts_tx_fifo_data_o[15:0]	Output	tts_tx_fifo_clk_o	FIFO data bits

7.2.3 GBT Downlink Ports (from CRU to FE direction)

Port Name	Direction	Clock Domain	Description
gbt_tx_clk_i[N-1:0]	Input		Input clocks
gbt_tx_isdata_o[N-1:0]	Output	gbt_tx_clk_i	Is data flags
gbt_tx_data_o[N*112-1:0]	Output	gbt_tx_clk_i	GBT payload bits
gbt_tx_ack_i[N-1:0]	Input	gbt_tx_clk_i	GBT payload bits accepted

7.2.4 GBT Uplink Ports (from FE to CRU direction)

Port Name	Directio n	Clock Domain	Description
gbt_rx_clk_i [N-1:0]	Input		Input clocks
gbt_rx_valid_i[N-1:0]	Input	gbt_rx_clk_i	GBT payloads valid bits
gbt_rx_isdata_i[N-1:0]	Input	gbt_rx_clk_i	Is data flags
gbt_rx_data_i[N*112-1:0]	Input	gbt_rx_clk_i	GBT payload bits

7.2.5 PCIe Link Ports

Port Name	Direction	Clock Domain	Description
pcie_clk_250mhz_i	Input		PCIe input clock
pcie_data_o[511:0]	Output	pcie_clk_250mhz_i	FIFO data output bits
pcie_wr_o	Output	pcie_clk_250mhz_i	FIFO write enable signal
pcie_afull_i	Input	pcie_clk_250mhz_i	FIFO almost full signal
pcie_full_i	Input	pcie_clk_250mhz_i	FIFO full signal

Port Name	Direction	Clock Domain	Description
mms_clk_i	Input	N/A	Input clock
mms_reset_i	Input	mms_clk_	Reset
mms_address_i [15:0]	Input	mms_clk_	Address bits
mms_write_i	Input	mms_clk_	Write signal
mms_writedata_i [31:0]	Input	mms_clk_	Write data
mms_read_i	Input	mms_clk_	Read signal
mms_readdata_o[31:0]	Output	mms_clk_	Read data
mms_rvalid_o	Output	mms_clk_	Read data valid

7.2.6 Avalon-MM Slave Port - PCle BAR Read/Write access

7.2.7 GBT-SCA Link Ports

This interface is not yet specified. The main purpose of this interface will be to provide a direct GBT-SCA access from the User Logic on CRU firmware level with deterministic latency to allow time critical tasks (like the TPC Safety Monitor).

8 Terminology

Arria 10	State of the art mid-range FPGA family of ALTERA
ASIC	Application Specific Integrated Circuits
Avalon-MM	(Memory mapped) control interface of the proprietary SoC bus of ALTERA
Avalon streaming	Data streaming interface of the proprietary SoC bus of ALTERA
BAR	Base Address Register
CDH	(ALICE) Common Data Header
Core CRU firmware frame	work
	Core CRU firmware functionality provided by cntral CRU team as framework for the CRU User Logic developements
CRU	Common Read-out Unit
СТР	Central Trigger Processor
DCS	Detector Control System
User Logic	Common/generic or detector specific CRU firmware modules that add functionalities to the core CRU firmware framework. Detector specific user logic are developed by some detector CRU teams providing such detector specific functionalities like processing of data (e.g. cluster finding), etc.
DMA	Direct Memory Access
FPGA	Field Programmable Gate Arrays
FLP	First level processing node
GBT	Gigabit Bi-directional Trigger and Data Link
GBT downlink	GBT link CRU -> FE direction
GBT uplink	GBT link FE -> CRU direction
GBTx	Radhard ASIC for the CERN GBT link
GBT-SCA	Radhard ASIC for the CERN GBT link
GBT-FPGA	Firmware implementation of the GBT link interface
Common/Generic CRU Fi	mware
	Standalone CRU firmware provided by the central CRU team for detectors sending standard ALICE packets and not needeing detector specific functionalities in the CRU. It consist of the core CRU firmware framework and a generic/common User Logic developed be the central CRU team.
HDLC	High-Level Data Link Control protocol
LTU	Local Trigger Unit (of the trigger system)
MiniPODs	Avago 12-lane parallel optical transmitters and receivers
02	ALICE Online and Offline Project (upgrade project for Run 3 and Run 4)
PCle	PCI Express
SoC	System on Chip
SDH	(ALICE) Single Data Header
TTS	Trigger and Timing System (a.k.a trigger system)
TTS downlink	Trigger link LTU -> CRU direction
TTS uplink	Trigger Link CRU -> LTU direction

The following ALICE detectors plan to use CRU after the upgrade of the ALICE their read-out and trigger system:

ACO	ALICE Cosmic Ray Detector
FIT	Fast Interaction Triggr detector (of ALICE)
ITS	Inner Tracking System detectors (of ALICE)
МСН	Muon Chambre detector (of ALICE)
MFT	Muon Forward Tracking (of ALICE)
MID	Muon Identifier detector (of ALICE)
TOF	Time of Flight detector (of ALICE)
TPC	Time Projection Chambre detector (of ALICE)
TRD	Transition Radiation Detector (of ALICE)
ZDC	Zero Degree Calorimeter detector (of ALICE)