

PAPER • OPEN ACCESS

## Versatile firmware for the Common Readout Unit (CRU) of the ALICE experiment at the LHC

To cite this article: O. Bourrion *et al* 2021 *JINST* **16** P05019

View the [article online](#) for updates and enhancements.

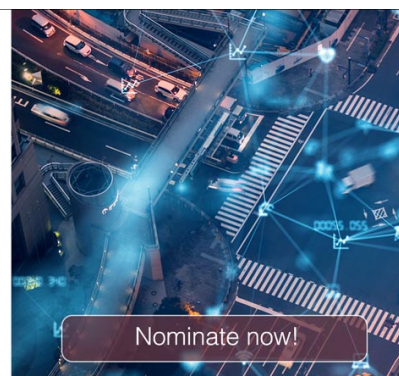


**The Electrochemical Society**  
Advancing solid state & electrochemical science & technology

The ECS is seeking candidates to serve as the  
**Founding Editor-in-Chief (EIC) of ECS Sensors Plus,**  
a journal in the process of being launched in 2021

The goal of ECS Sensors Plus, as a one-stop shop journal for sensors, is to advance the fundamental science and understanding of sensors and detection technologies for efficient monitoring and control of industrial processes and the environment, and improving quality of life and human health.

*Nomination submission begins: May 18, 2021*



## Versatile firmware for the Common Readout Unit (CRU) of the ALICE experiment at the LHC

O. Bourrion,<sup>a,\*</sup> J. Bouvier,<sup>a</sup> F. Costa,<sup>b</sup> E. Dávid,<sup>c</sup> J. Imrek,<sup>c</sup> T.M. Nguyen<sup>c</sup> and S. Mukherjee<sup>d</sup>

<sup>a</sup>Univ. Grenoble Alpes, CNRS, Grenoble INP (Institute of Engineering), LPSC-IN2P3, 38000 Grenoble, France

<sup>b</sup>CERN, 1211 Geneva 23, Switzerland

<sup>c</sup>Wigner Research Centre for Physics, 29-33 Konkoly-Thege Miklós str, H-1121 Budapest, Hungary

<sup>d</sup>Bose Institute, Kolkata 700 054 WB, India

E-mail: [olivier.bourrion@lpsc.in2p3.fr](mailto:olivier.bourrion@lpsc.in2p3.fr)

**ABSTRACT:** As from the run 3 of CERN LHC scheduled in 2022, the upgraded ALICE experiment will use a Common Readout Unit (CRU) at the heart of the data acquisition system. The CRU, based on the PCIe40 hardware designed for LHCb, is a common interface between 3 main sub-systems: the front-end, the computing system, and the trigger and timing system. The 475 CRUs will interface 10 different sub-detectors and reduce the total data throughput from 3.5 TB/s to 635 GB/s. The ALICE common firmware framework supports data taking in continuous and triggered mode and forwards clock, trigger and slow control to the front-end electronics. In this paper, the architecture and the data-flow performance are presented.

**KEYWORDS:** Data acquisition concepts; Detector control systems (detector and experiment monitoring and slow-control systems; architecture; hardware, algorithms; databases); Modular electronics

\*Corresponding author.



---

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Hardware overview</b>	<b>2</b>
<b>3</b>	<b>Data readout</b>	<b>4</b>
3.1	GBT packet type	5
3.2	GBT stream type	6
<b>4</b>	<b>CRU firmware requirements</b>	<b>6</b>
<b>5</b>	<b>Firmware description</b>	<b>7</b>
5.1	Board Support Package (BSP)	8
5.2	GBT wrapper	8
5.3	TTS interface	9
5.4	Detector Data Generator (DDG)	10
5.5	Slow control	10
5.6	Datapath wrapper	12
5.7	PCIe DMA	13
<b>6</b>	<b>Firmware development, simulation and integration</b>	<b>14</b>
<b>7</b>	<b>Resource usage</b>	<b>16</b>
<b>8</b>	<b>Conclusion</b>	<b>17</b>

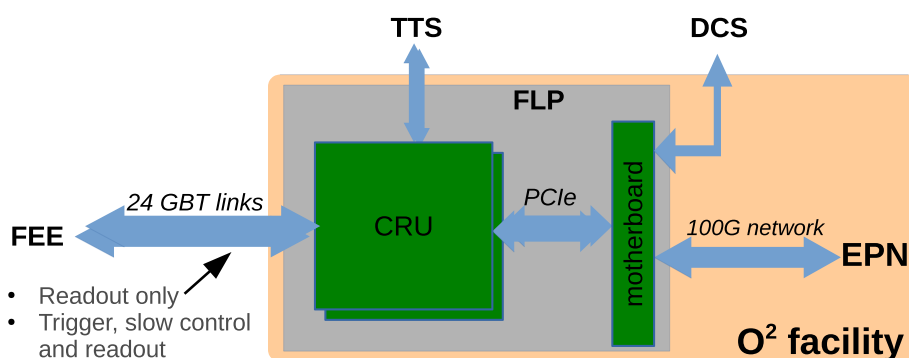
---

## 1 Introduction

The ALICE upgrade addresses the challenge of reading out lead-lead (Pb-Pb) collisions at rate of 50 kHz, proton-proton (pp) and proton-lead (p-Pb) at 200 kHz and higher. This will result in the collection and inspection of a data volume of heavy-ion events roughly 100 times larger than during run 1 and 2. From run 3 on, the majority of ALICE sub-detectors are upgraded to operate in continuous, trigger-less readout mode. This a consequence to the fact that a very low signal-to-background ratio is expected in the low- $p_T$  region, as the rate of collisions of interest will be of the same order as the interaction rate [1]. However, triggered readout is still used by all detectors for commissioning and calibration runs. Also, some detectors which are not upgraded still use the legacy readout systems in triggered mode. The 13 ALICE sub-detectors read out via  $\approx 10000$  readout links produce 3.5 TB/s of data. In order to cope with the continuous readout and the resulting data throughput 10 detectors have been upgraded to use the Common Readout

Unit (CRU). Faithful to the design reuse strategy for the LHC experiments, the PCIe40 electronics designed for the LHCb experiment [2] is used as the ALICE CRU. The CRU is a PCIe-gen3 based FPGA processor board with up to 48 bidirectional optical links (48 in, 48 out).

The CRU is the interface between the front-end electronics (FEE), the Online-Offline facility ( $O^2$ ), the Detector Control System (DCS) and the Trigger and Timing System (TTS), see figure 1. One CRU can interface up to 24 optical links with the front-ends, one trigger link with the TTS and one PCIe interface. The GBT protocol [3, 6] developed by the electronics is used to communicate with the front-end electronics. In ALICE it was chosen to aggregate a limited number of links (24) to limit the risk to loose a large portion of some detectors in case of hardware failure.



**Figure 1.** The CRU is the interface between the detector front-end electronics, the  $O^2$  facility, the DCS and the TTS via the Local Trigger Unit (LTU).

The  $O^2$  facility is a computer farm composed of the First Level Processors (FLP) and Event Processing Nodes (EPN). The FLP (Dell PowerEdge R740) exchanges information with the FEE via the CRU, it can host a maximum of three CRUs. The FLP communicates with the EPN through a 100 Gb InfiniBand network.

The front-end interface, ensured via the GBT link, can be used to receive/deliver the following information:

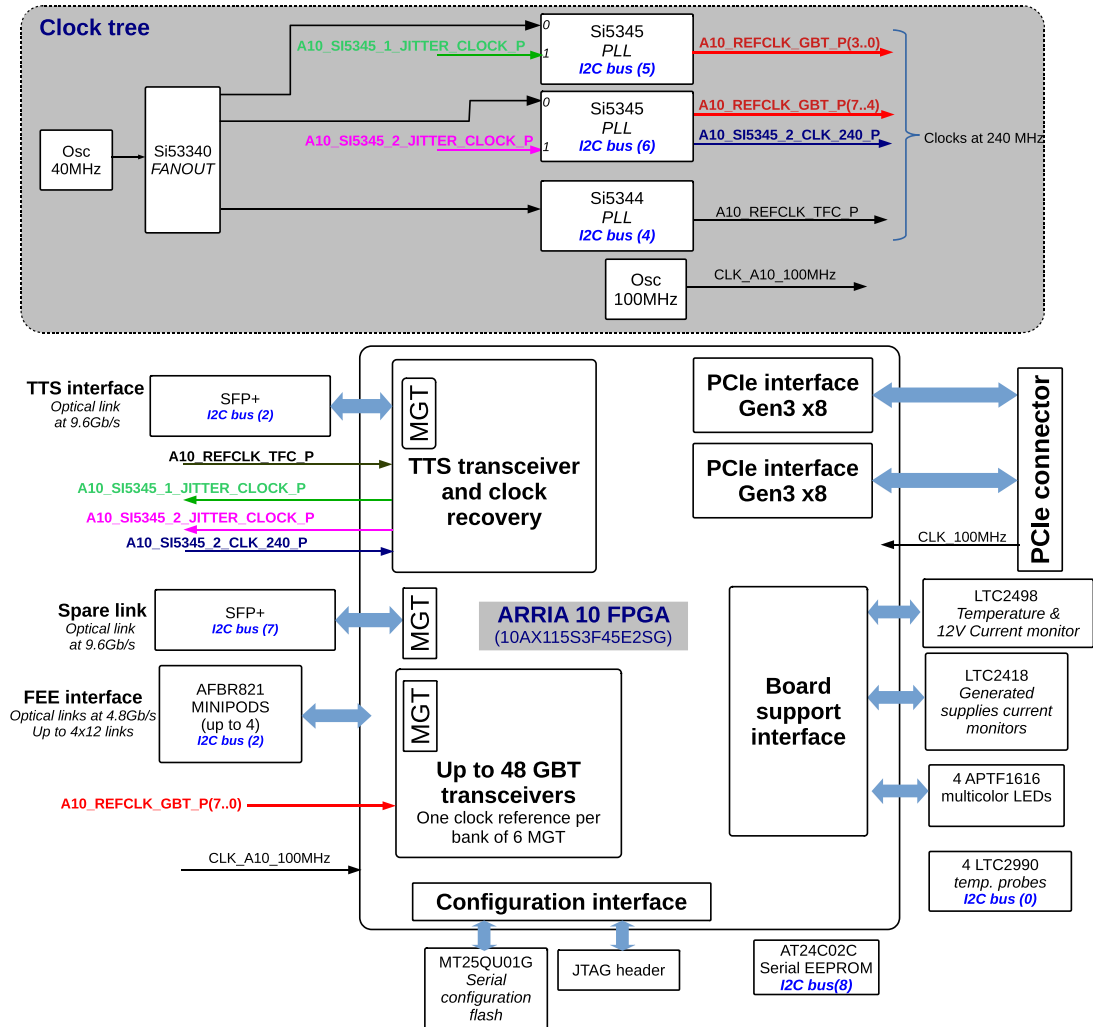
- Readout (physics data), from FEE to CRU.
- Trigger and timing (clock and trigger information), from CRU to FEE.
- Slow control, both directions.

The communication with the TTS is done via a dedicated bidirectional Passive Optical Network (PON) [4]. The PON is time multiplexed in the upstream direction (CRU toward TTS). It allows the reception of the LHC machine clock and of trigger and decision messages in the downstream direction (TTS toward CRU) and acknowledge messages in the upstream direction.

The CRU is connected to the server's motherboard via dual PCIe gen3  $\times 8$ . The DMA is dedicated to the readout of the detector while control messages from the DCS are passed through the Base Address Register (BAR) interface.

## 2 Hardware overview

A functional overview of the hardware highlighting the features used in ALICE CRU can be seen in figure 2. The clock tree is shown as well as the FPGA and its interfaces with the various components of interest.



**Figure 2.** CRU hardware overview. The clock tree is shown as well as the FPGA and its interfaces with the various components of interest. The TTS and the GBT transceivers use dedicated Multi Giga Bit transceivers (MGT).

The clock tree is designed such as to use a common reference for all communication links of the CRU, aside from the PCIe which uses 100 MHz. The board can be operated either in standalone mode with a 40 MHz oscillator available on the board or from a recovered clock extracted from the optical connection with the TTS.

However, the TTS transceiver needs a stable 240 MHz reference clock at startup, locally generated by a SI5344PLL. Once locked to the incoming stream, the recovered clock is sent from the FPGA to a high performance PLL (SI5345) for jitter attenuation. The cleaned clocks are then used to operate the FPGA logic and the GBT. The selection between the local and recovered clock mode is done with the SI5345 PLL via I2C communication. A free running 100 MHz clock is produced on board and used to operate the miscellaneous functions embedded within the FPGA, such as initialisation and hardware monitoring.

The FPGA is an ARRIA10 from Intel (10AX115S3F45E2G). It is connected to two Small Form Pluggable (SFP+) connectors. Only one SFP of the two is used for the TTS connection, the second is a spare connection. Up to  $4 \times 12$ -channel bi-directional 10.3125 Gb/s optical transceivers (mini-pods [5]) ensure the connections to the front-end electronics via GBT links.

For ALICE, only two mini-pods are equipped to allow the connection to 24 front-end links with the only exception being the TRD detector, which does not use the GBT protocol and where 36 links and 3 mini-pods are needed.

On the back-end side the CRU is connected to the PCIe edge connector and offers a dual gen3  $\times 8$  PCIe interface. This interface is clocked by the reference clock of 250 MHz provided through the connector.

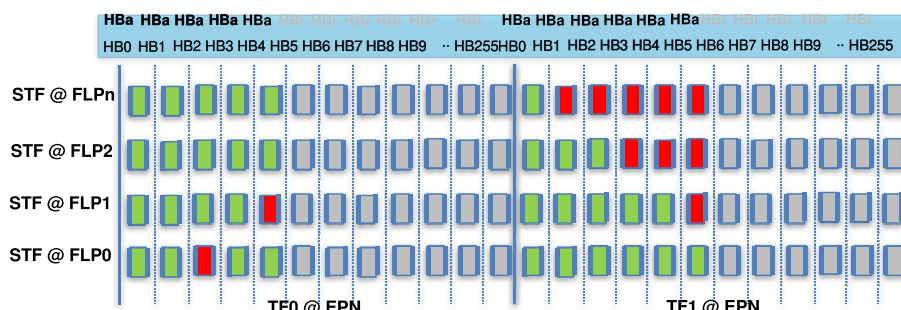
The FPGA is also connected to board support functionalities such as temperature and current sensors, and an EEPROM which contains a unique identifier set by the manufacturer during board assembly. Communication to these peripherals is via I2C or Serial Peripheral Interface (SPI). There is also the possibility to control multi-color LEDs which is useful when trying to quickly locate a specific machine in a server farm for maintenance purpose.

Finally, the FPGA can be configured either from a JTAG probe, which is handy for debugging the firmware in lab, or from a Quad SPI flash. The latter can be reconfigured remotely via the PCIe interface, allowing on-site upgrades.

### 3 Data readout

The ALICE computing upgrade concept consists of transferring all detector data unfiltered (triggerless) to the computing system. Data volume compression will be performed by processing the data on the fly and not by rejecting complete events as do the high-level triggers or event filter farms of most high-energy physics experiments. For event reconstruction at the EPN level, the continuous data stream is sliced in Time Frames (TF) of programmable length of maximum of 22 ms. The times frames are then divided into Heart Beat Frames (HBF) of one orbit duration (89.4  $\mu$ s). Heart beat and time frame triggers indicating the boundaries between HBFs and TFs are distributed to the CRU via the PON network by the Central Trigger Processor (CTP). In this scheme, the task of the CRU is to collect the data continuously and to check the successful heart beat frame transmission to each first level processor. The CTP distributes heart beat triggers defining whether the corresponding HBF should be accepted (HBaccept—HBa) and thus forwarded to the FLP or deleted (HBreject—HBBr). This scheme allows the data throughput to be adjusted to match the available bandwidth during commissioning and dedicated calibration runs. For each HBF, each CRU delivers *an acknowledge* (HBACK) or *not acknowledge* message (HBNACK) to CTP which

assesses the quality of the HBF transmission of all CRUs. In the case incomplete or corrupted HBFs have been transmitted, the CTP can request the corresponding HBF or even those from a full TF to be deleted from the FLP memory. An example of two incomplete time frame transmissions is shown in figure 3.



**Figure 3.** Illustration of the continuous readout (extracted from [8]). Each rectangle represents a heart beat frame (green: successfully received; red: bad reception or missing fragment; grey: deleted HBF). There are up to 256 HBFs per Sub Time Frame (STF) produced by each FLP and the time frame is the collection of STF at the EPN level. In this example, the remaining HBFs of a TF are rejected after a bad reception is detected. HBA: heart beat accept; HBR heart beat reject.

The upgraded detectors are classified either as streaming or as packet detector types and can be read out via the CRU in two readout modes: continuous and triggered. Both readout modes are supported by the two types of detectors, and the experiment will run in one of the two readout modes with a combination of different detector types.

- Packet detectors can generate data formatted in packets before shipping them to the CRU. Every packet, so called GBT packets, contains the time stamp defined as orbit and bunch crossing.
- Streaming detectors cannot organize the data in formatted packets, therefore they generate a stream of data, so called GBT stream, without orbit and bunch crossing information.

As described before, data are transferred to the FLP memory in Heart Beat Frames (HBF). The data coming from the different detectors reach the FLP memory with an identical format.

### 3.1 GBT packet type

The CRU expects to receive HBFs prepared in the FEE and transferred over the GBT links. The HBF consists of one or more blocks of data containing one header, called Raw Data Header (RDH) and the payload. Data coming from detectors delivering GBT packets are forwarded to the DMA engine unmodified by the CRU firmware. The CRU verifies only the correct HBF structure and formatting. For every HBF the CRU sends an acknowledge message to the CTP informing whether the corresponding HBF has been correctly received (data received properly from all links involved in the data taking) and forwarded to the FLP.



### 3.2 GBT stream type

GBT stream type sequences do not have an HBF structure and consist of a continuous stream of detector specific raw data entries. Upon receiving HB triggers, the CRU partitions the data stream in HBFs and forwards them to the DMA engine or deletes the corresponding data packets. In this configuration each GBT link generates a data throughput of up to 4.48 Gbps, regardless the content of the data. In normal operation the detector specific user logic of the CRU performs zero suppression in order to reduce the data throughput.

For short debugging and calibration runs the data compression can be switched off to transfer sequences of uncompressed data, otherwise the amount of data would be too high to be handled by the rest of the data taking chain. For the streaming detectors to collect data in the physics runs, it is mandatory to implement the user logic in the CRU, as described below.

## 4 CRU firmware requirements

The CRU firmware is divided into two parts. The first part is the common firmware which (i) provides the interfaces to PCIe, trigger and timing, and up to 24 front-end links via the GBT protocol, (ii) provides the possibility to read out all detectors in ‘raw-mode’ with no data processing in the CRU, (iii) allows reference clock and trigger signals distribution, and (iv) permits FEE configuration. The second part is the user logic which is only needed for those detector systems that need detector-specific data processing for instance baseline correction or zero suppression.

An important ALICE requirement is to be able to switch between raw-mode and user-logic at any moment without reloading different firmware versions. Self-testing capabilities to ease commissioning and system maintenance are implemented. They are detailed later in this paper.

From a system point of view, the different requirements for the common firmware are due to the different GBT bus mode (packet or stream), the information it is supposed to carry, the integration of a user logic or not, and the type of slow control protocol required to configure the FEE.

It is important to mention that when the data reach the DMA engine in the CRU the format is identical, regardless of the data taking configuration of the card. Having the same data format simplifies the firmware logic of the CRU as well as the software required for physics analysis.

The GBT links are used to send downstream trigger messages and/or the reference clock to the FEE, while upstream they are used for data readout and optionally to acknowledge specific slow control transactions. The CRU firmware supports operation of the GBT links in GBT-mode (80 bits of payload and 32 bits of forward error correction) or in wide-bus (112 bits of payload and no forward error correction). For almost all the GBT detectors the GBT-mode provides sufficient data bandwidth. Only the Time Projection Chamber (TPC), pushes data using the ‘wide-bus’ mode in order to satisfy the bandwidth requirements. As the radiation load for the TPC front-end cards is sufficiently low, the forward error correction the GBT-mode would provide is not required. The requirements of the various detectors can be summarized in table shown in table 1.



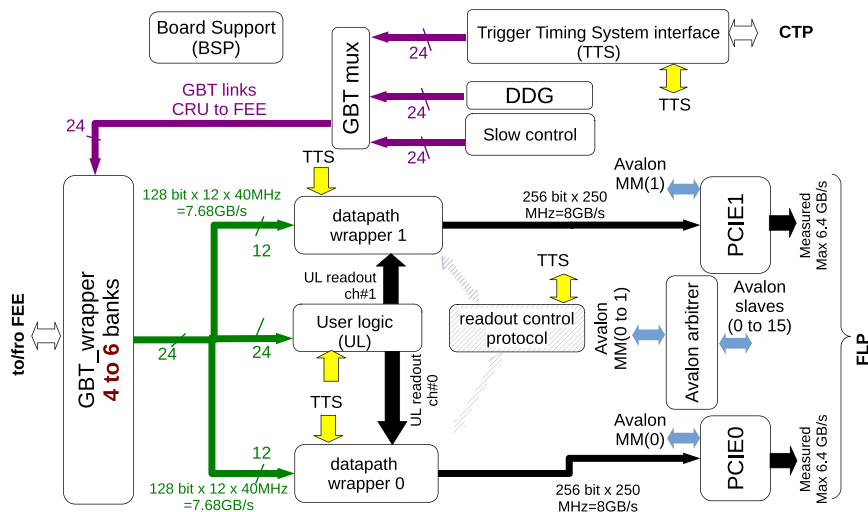
**Table 1.** Table summarizing the requirements of the various detectors. In white features not required; in blue compatible requirements with an adapted common firmware; in green and orange features requiring a specific firmware file generation.

Detector	Number of CRU	TX/RX per CRU	User Logic	GBT with FEC ("gbt-mode")	TTS through CRU	DCS through CRU
CPV (Charged Particle Veto)	1	24/24	N	Y	Y	Y (SCA)
FIT (Fast Interaction Trigger)	2	24/24	N	Y	Y	N
ITS (Inner Tracker System)	26	12/24	N	Y	clock only	Y (single word transfert)
MCH (Muon Tracking Chamber)	32	24/24	Y	Y	Y	Y (SCA)
MFT (Muon Forward Tracker)	11	12/24	N	Y	clock only	Y (single word transfert)
MID (Muon Identifier)	2	24/24	Y	Y	Y	Y (SCA)
TOF (Time Of Flight)	4	24/24	N	Y	Y	N
TPC (Time Projection Chamber)	360	24/24 (50%) 0/24 (50%)	Y	N (static)	Y	Y (SCA)
TRD (Transition Radiation Detector)	37	0/36	Y	not GBT	N	N
ZDC (Zero Degree Calorimeter)	1	24/24	N	Y	Y	N

## 5 Firmware description

An overview of the firmware is given in figure 4. The main parts are shown: the FEE interface (through GBT links), the Trigger and Timing System (TTS) interface, the Board Support Package (BSP), the data path and the PCIe endpoints. Starting from the front-end side on the left, the GBT\_wrapper interface is shown; it is the interface with the FEE.

On the downstream path (CRU to FEE), depending on the detector requirements or test requirements, several sources can be selected to supply the GBT\_wrapper. These are the trigger and timing system interface, the Dedicated Data Generator (DDG) or the slow control.



**Figure 4.** Overview of the common firmware. The main parts are shown: the GBT wrapper, the trigger and timing system interface, the data path and the PCIe endpoints.

## 5.1 Board Support Package (BSP)

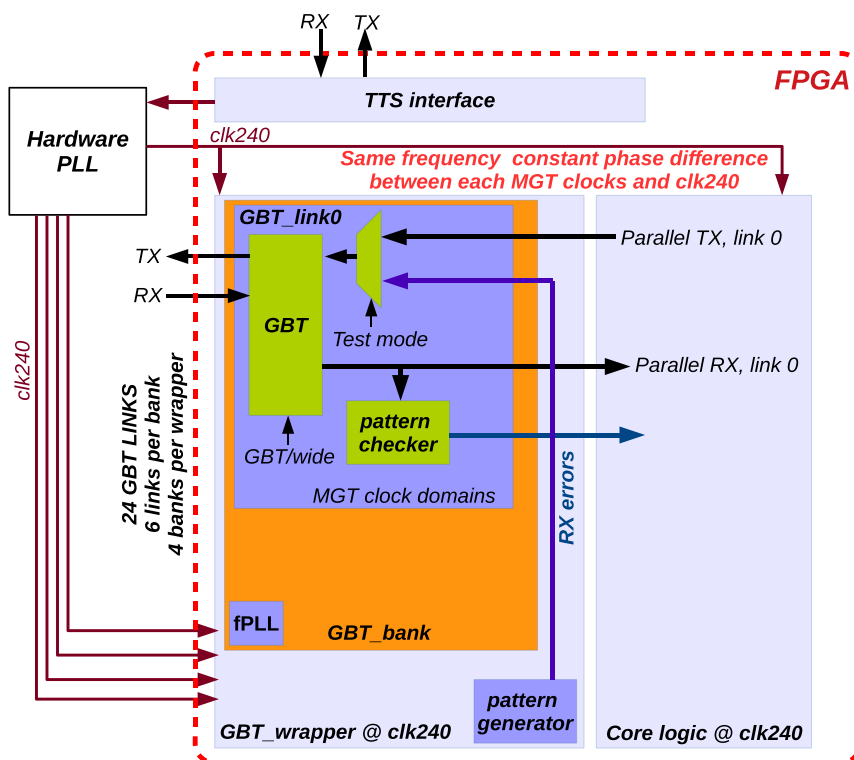
The board support package features several I2C masters that are directly controlled through the PCIe interface. They allow the readout of the various optical transceiver parameters such as temperature or optical power, the settings of external PLLs and the access to the board serial number stored in an EEPROM. Additionally, it permits accessing to the FPGA serial number (fixed by the FPGA manufacturer) and monitoring of the FPGA die temperature.

The BSP functionality also includes the reconfiguration of the QSPI flash and the possibility to trigger an FPGA reboot from the slow control (through PCIe interface). The chosen strategy is to have two reserved areas in the flash memory, one for a golden and one for the application firmware. The golden firmware will never be modified outside of the lab, while the application firmware can be modified when deployed on site. At cold startup, the FPGA always boots on the golden firmware. Then, by issuing a PCIe BAR command, specifying the memory offset to use, it is possible to load the application firmware in the FPGA. In the case of a configuration failure (e.g. loss of connection, power cut or faulty firmware) the CRU can be easily recovered.

## 5.2 GBT wrapper

The `GBT_wrapper` is a modified version of the GBT-FPGA developed at CERN [3, 6], see figure 5. The main differences are that (i) it has a user data path operating at 240 MHz (six times the machine clock), (ii) the clock domain crossing between the transceiver domain and the user domain is achieved with timing constraints instead of a phase scan at link start-up, (iii) dynamic switching is possible between GBT-mode and wide-bus mode to cover all detector requirements and (iv) the test data pattern generator is shared between all links to save resources. Moreover, the `GBT_wrapper` permits external (with optical fibers) and internal (inside the FPGA transceivers) loop-back tests which allow the validation of the CRU-FEE communication and the CRU data path operation once installed in the system. In external loop-back the data generator enables the emission of representative data towards the FEE that can be looped back by them into the CRU; while in internal loop-back mode, this feature allows to stress the system without relying on the availability of detector FEE.

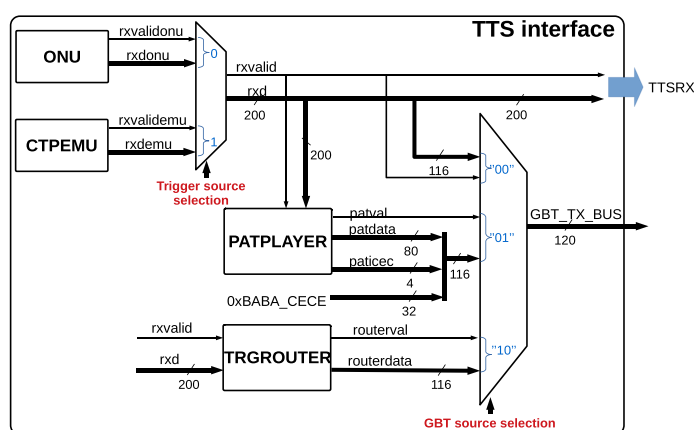
The strategy used to maintain a constant latency and to avoid the phase scanning on the transmission path (CRU to FEE) is (i) to rely on the zero delay buffer provided by the hardware PLL and to feed the extracted transmission delays due to the PCB in the constraint file, and (ii) to use the 6 time faster rate to properly sample and transfer the data from one clock domain to the other (120 bits transferred at 40 MHz). On the receiving, which is used only for readout, a non constant latency can be accepted, and thus a FIFO was implemented to cope with the clock domain transfer from the recovered clock domain to the user part clock domain. This solution was extensively tested across several scenarios and proved to be reliable. The scenarios tested were: CRU reference clock switching between local and remote clock source, warm reboot (FPGA reconfiguration) and cold reboot (CRU and FLP turned off and rebooted).



**Figure 5.** Modified GBT-FPGA inserted in the common firmware overview. It shows the clock recovered from the trigger and timing system transferred to an external jitter cleaner PLL. The cleaned clock is re-injected into each bank of the FPGA used for the GBT connection. A single GBT link within a MGT bank is shown. The pattern generator is shared between all links and operates with the common core clock (clk240).

### 5.3 TTS interface

As shown in figure 6, the TTS interface is composed of four components.



**Figure 6.** Overview of the TTS interface. The 200 bit word contains the trigger information on its lower 116 bits and the trigger decision message on its 80 bits upper part.

The first is the Optical Network Unit (ONU) [9, 10] which recovers the machine clock from the PON and forwards it through the GBT to the FEE. The ONU is also used to receive the trigger and timing message at each clock cycle (trigger bits, bunch crossing number, heart beat ID) from the central system. The 200 bit word contains the trigger information on its lower 116 bits and the trigger decision message on its 80 bits upper part. The CRU uses the upstream direction to send the HBACK or HBNACK trigger message. As the optical network is passive, the upstream communication is time multiplexed and a message can only be sent every 125 ns times the total number of ONUs connected on the PON. The second component is a trigger emulator (ctpemu) that is used for tests and system diagnostic purposes. It can produce trigger messages, like the ones provided through the ONU, and simulate readout flow control by producing HBa and HBr commands. The third component is the pattern player (patplayer) that can generate a programmable sequence to be transmitted to the FEE, it is started by a trigger bit issued either by the ONU or ctpemu. The fourth is the trigger router (trgrouter) which remaps, duplicates and forwards some trigger bits received via the PON from the CTP to the FEE boards via the GBT links.

#### 5.4 Detector Data Generator (DDG)

The DDG is the component in the CRU able to emulate detector behavior and data throughput. This component plays a main role during the test and validation of the CRU firmware and the readout software chain when the detector hardware is not yet widely available. The DDG has different configuration parameters and it can be dynamically configured to produce either streaming or packet type data. The data packet can be produced for different trigger types with fixed or random packet length and inter packet duration, generating a realistic detector throughput. The DDG can be used to test the firmware at any moment in time. Configuring the GBT links using the internal loop-back connection makes the injection of DDG data in the system possible without the requirement to change the physical optical connection at the card input. DDG is a powerful self-test feature to verify the correct behavior of the hardware and the software without relying on external hardware elements like the FEE.

#### 5.5 Slow control

The majority of the detectors are connected to the DCS system through the CRU. In ALICE there are three configuration protocols:

- GBT-EC (External Control) [6], used to send configuration data to the GBT SCA ASIC (Slow Control Adapter) [7] installed on the FEE.
- GBT-IC (Internal Control) [6], used to configure the GBTx ASIC on the FEE.
- GBT-SWT (Single Word Transfer), not using the GBT protocol.

While the GBT-EC and IC protocols are part of the GBT design developed at CERN, the SWT is ALICE specific and can be used only by detectors that host an FPGA in the FEE, see table 1. The SWT protocol has been introduced to increase the bandwidth for the slow control operation. It uses the GBT-DATA path to deliver up to 3200 Mb/s (80 bits per 40 MHz clock cycle) to the FEE whereas the GBT-EC path provides 80 Mb/s. In practice the slow control read/write speed

is limited to 36 Mb/s by the time taken for software to access the PCIe BAR. To send the detector configuration data over the GBT link using the SWT protocol, the CRU must be configured to switch to the SWT traffic with the GBT-MUX component (see figure 4). This is a static selection, as there is no dynamic packet switching for the downstream path. In the opposite direction, from the FEE towards CRU, the SWT and FEE information is interleaved within the same link.

In order to distinguish GBT words that contain physics information from the control words, like the SWT, the CRU uses two types of information, the *isdatasel* flag decoded from the GBT header and a part of the GBT word embedded in the GBT data field.

When the detector sends physics data the flag *isdatasel* is set to 1 and the whole 80 bit data field is used to transfer the information. When this flag is 0 the CRU considers the GBT word a control word and it uses the four most significant bits of the data field to distinguish between the different control words (figure 7):

- IDLE: 0x0. This word is used to pause the data transfer in case the FEE has no data to send to the CRU.
- SOP: 0x1. This word identifies the start of packet during a GBT data transfer from a GBT packet type detector.
- EOP: 0x2. This word identifies the end of packet during a GBT data transfer.

isdatasel	[79-76]	[75-0]	
0	SOP 0x1	0x0	} PACKET 1
1	DATA		
1	DATA		
1	DATA		
0	EOP 0x2	0x0	} PACKET 2
0	IDLE 0x0	0x0	
0	IDLE 0x0	0x0	
0	SOP 0x1	0x0	
1	DATA		} PACKET 2
1	DATA		
1	DATA		
0	SWT 0x3	0x0	
0	IDLE 0x0	0x0	
0	EOP 0x2	0x0	

**Figure 7.** GBT stream from a packet detector FEE to a CRU. The physics words are marked by the flag *isdatasel* set to 1 and the control words: IDLE, SOP, EOP, SWT are marked by the flag set to 0 and with a different header stored in the GBT data field.

- SWT: 0x3. Single word transfer, it is used to identify a configuration word.

The CRU extracts the SWT information from the data stream before it reaches the DMA engine and stores it in a dedicated FIFO which is accessed by DCS.

### 5.6 Datapath wrapper

Two datapath\_wrapper blocks are implemented in the firmware. They receive the trigger messages, collect and aggregate the FEE data provided by the GBT link or use directly the user logic input if implemented. They also provide monitoring information to the readout control protocol component (see figure 8).

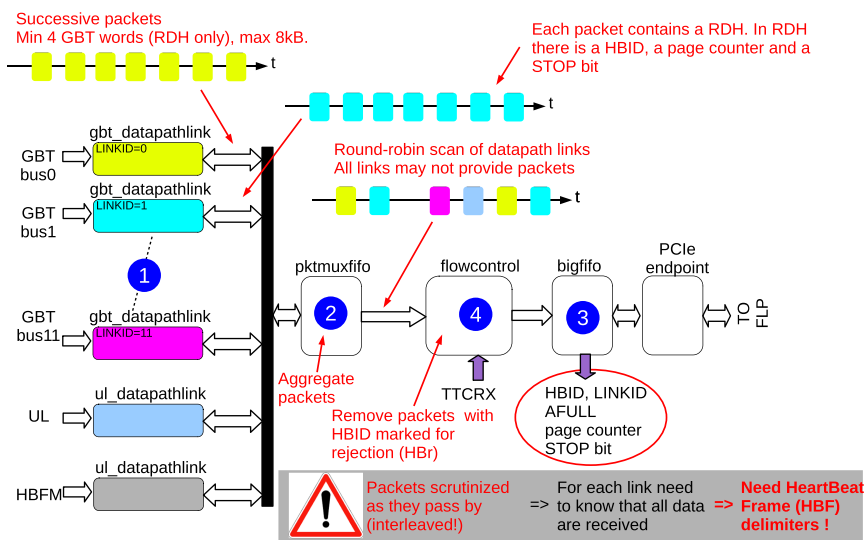


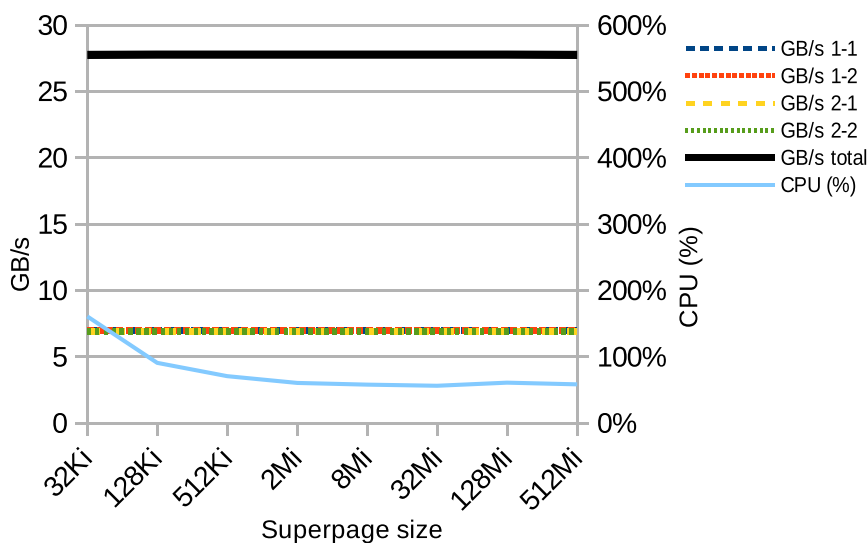
Figure 8. Overview of the datapath\_wrapper.

The first task of each `datapath_wrapper` is to receive in parallel the data either from up to 12 GBT busses, from the readout protocol component (trigger acknowledge or decision message) and/or from one user logic link. The `gbt_datapathlink` is compatible with stream or packet type format. When selecting the stream mode this component constructs data packets, by chopping the data stream and inserting the Raw Data Header (RDH). The RDH describes the readout packet content and contains the Heart Beat ID (HBID), the link ID, the page counter and the stop bit as well as other information. For each link ID the page counter gives the packet identification within the corresponding HBF transmitted and the stop bit indicates whether the last packet for this HBF is being transmitted. The `ul_datapathlink` receives already correctly formatted packets, either from the user logic or from the readout protocol block, i.e. heart beat acknowledge and decision messages (HBFM). At the output of this first stage the packets have a maximum size of 8 kB. The second stage, named `pktmuxfifo`, performs data aggregation by scanning in a round-robin based manner possible data sources (`gbt_datapathlink` and `ul_datapathlink`) and collects data packets. At the output of this stage, the packets from the

various links are interleaved. If required by the CTP, this is followed by the removal of all packets from the data flow with a HBr message (number 4 in figure 8). Then, the packets are stored in a large buffer (bigfifo, 16 kwords of 256 bits) to be made available to the PCIe endpoint. While being stored, the packets are scrutinized and useful parameters (HBID, LINKID, FIFO status) are presented to the `readout control` protocol component. The `readout protocol` uses the information provided by both `datapath_wrapper` blocks to check the interleaved packets. The HBF reception is declared successful only if for each LINKID included in the readout, start (page counter is 0 in RDH) and stop packets (stop bit is 1 in RDH) were received in consecutively and properly stored in the `bigfifo` buffers before a pre-defined timeout for reception elapsed. Then a HBACK or HBNACK message is transmitted to the CTP, which assembles the messages from all CRUs and updates the HBa/HBr messages to communicate whether a given HBF should be maintained or be deleted in the FLP [8, 11].

## 5.7 PCIe DMA

The main data stream flows through the DMA interface which is used to move data from the FEE to the FLP memory. The CRU communicates with the FLP server through a PCIe gen. 3  $\times$  16 interface implemented in the FPGA as a dual endpoint PCIe gen3  $\times$  8. The DMA engine of the CRU is capable of sustaining a total data throughput of 110 Gb/s. In order to achieve this performance, both endpoints must work in parallel, each one handling a maximum of 55 Gb/s (figure 9).



**Figure 9.** The graph shows the measured throughput performance of two CRUs installed in one server and of the CPU usage versus the super-page size. There are in total four endpoints, each one absorbing the maximum allowed data throughput of 55 Gb/s. Note the CPU (Intel Xeon Silver 4210) usage decreases with increasing size of the super-page.

The nominal DMA throughput provides sufficient margin to collect data from the most demanding detector in ALICE, the TPC. Each of the 360 TPC CRUs receives data from 20 GBT links



as input for a total throughput of roughly 89.6 Gb/s. Unfortunately, the data must be aligned to 32 bit words boundary, and therefore the actual throughput is 102.4 Gb/s. The incoming data are compressed by the TPC user logic before being delivered to the DMA engine. The expected output of the TPC user logic is 20 Gb/s.

In the firmware the CRU input data stream is divided between the two PCIe endpoints in order to avoid dynamic switching of the data flow between the two PCIe endpoints. Thus, half of the GBT links implemented in the CRU are connected to one single endpoint via its datapath-wrapper instance that receives data from a maximum of 12 GBT links. In this way, the data throughput is evenly distributed between the two PCIe gen3  $\times$ 8 interfaces.

The communication with the software [12] happens through the PCIe BAR interface. There are two BAR interfaces: BAR0 and BAR2. BAR0 is dedicated to DMA operations, it passes the page descriptors and monitors the status of the data taking, while BAR2 is used to access the card configuration and to monitor the other components of the firmware. The CRU transfers the physics data into the server memory using DMA transfers. In order to do so the CRU needs to know in which buffer data should be stored. For that, the software prepares (or allocates) the buffers in the FLP memory and then passes the addresses of each of these buffers to the CRU. This information is called the DMA page descriptors. Therefore, to maximise the throughput by reducing wait states, the descriptors are prepared in advance by software and delivered through a FIFO to the CRU DMA engine. If the FIFO holding the page descriptors becomes empty, the DMA transaction are paused and data are dropped in full DMA pages until new page descriptors are provided.

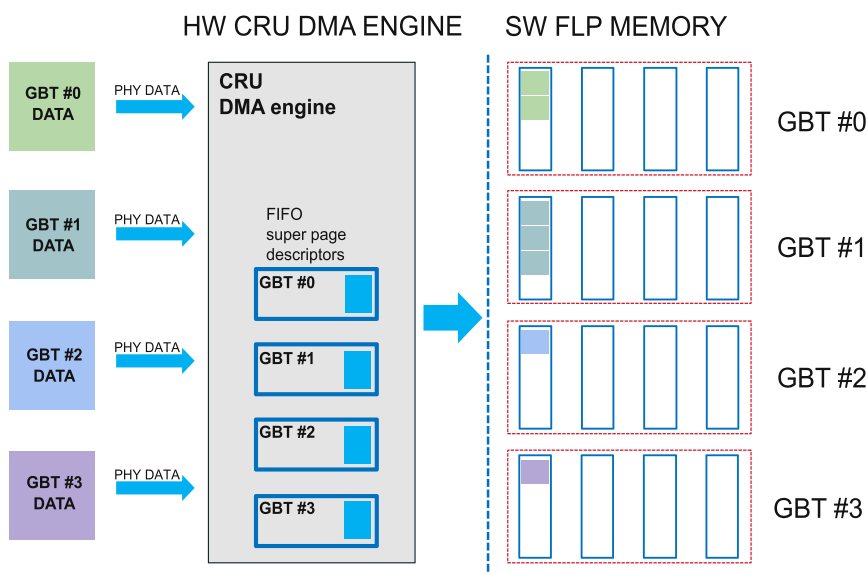
To reduce the software interaction with the hardware, the concept of super-pages has been introduced. A super-page is a buffer of contiguous spaces in memory of usually 1 or 2 MB size (configurable based on the event size of the detector). The software stores the address of this buffer in the CRU descriptor FIFO and the DMA engine fills up the buffer with the data coming from the input links. Once the super-page is full a new descriptor is fetched and data is stored in a new super-page. It is the responsibility of the CRU firmware to divide the super-page in smaller DMA pages, of a maximum size of 8 kB.

The CRU can collect the data from a maximum of 24 GBT links. During data taking the CRU is loaded with 128 super-page descriptors for each GBT link active in the data taking. Data coming from different links are organized in dedicated super-pages, therefore the software fetches data from specific GBT links in dedicated link memory buffers (as shown in figure 10).

Data in the FLP memory is grouped before being sent to the EPN for further analysis. The grouping is done at the boundary of time frames. To simplify the job of the software the HBFs are stored by the CRU in dedicated super-pages. In this way it is simpler to group the different buffers belonging to a single time frame and send them over RDMA (Remote DMA) to the EPN farm. Assigning a dedicated buffer for each GBT link removes any dependency between them, improving the stability of the system and the total throughput.

## 6 Firmware development, simulation and integration

As the firmware implementation is quite complex and requires different configurations with different parameters depending on the sub-detector application a dedicated simulation strategy was



**Figure 10.** DMA with four GBT links. Data are moved into dedicated memory buffers.

developed. The high-speed serial interfaces were simulated and validated on their own as they required much simulation power and would delay significantly a full simulation. Dedicated simulation test-benches were developed for the PCIe DMA interface and for the modified GBT-FPGA which was simulated against the reference design in its Xilinx implementation flavor. The ONU interface was not simulated because it used an IP developed and validated by CERN.

The core of the design was embedded into a dedicated test-bench. Simulation models emulated the data flow from the GBT-detectors (wide or GBT mode) while data sent to the PCIe DMA interface was recorded in a file. The CTP messages were generated by the internal CTP emulator. The FPGA Intel Avalon bus [13] required for configuration was connected to all the simulated components using it. An Avalon master model was used to set the various registers during the simulation. Thus, during the simulation, the configuration sequences required by the sub-detectors were simulated as they take place in real setups. This allowed a reduction of the development time, but also the possibility to reproduce in simulation error cases that were detected by users. Consequently, error resolution was faster, particularly for rarely occurring errors. To ease the simulation, a hexadecimal address table was declared in a VHDL common package and used as reference for these configuration simulations. Custom software extracted the address table information from the VHDL common package for use in the software running on the FLP.

The common firmware is distributed via a git repository to all users. Makefiles included in the distribution allow users to easily simulate or compile the firmware. A reference user logic generating known data is included in the reference design. It can be configured to generate pre-defined and random packet sizes and data rates. This feature along with the DDG is a complementary tool for stressing in situ the firmware and validating the readout software.

In ALICE the CRU firmware deployment remains under the CRU team responsibility. Sub-detectors add their user-logic code to a branch of the common firmware. The CRU team validates the code before integrating it into the master branch and finally deploying it. The branch is then

merged in the global repository, the firmware is compiled and delivered to the FLP in the ALICE system by the central team.

To keep track of the firmware after deployment and avoid misunderstanding, the firmware parts (common and user if any) are compiled after being committed in the repositories. In the provided scripts a routine generating tagging information is executed. The generated VHDL parameters are included in the firmware and fed to status registers before actually starting the compilation. Thus after compilation, by accessing these registers, it is possible to retrieve the compilation date and the git hash of the compiled code.

## 7 Resource usage

A significant effort was made to find a good trade-off between the minimization of the FPGA resources and providing as much adaptability as possible to cover all detectors requirements, while providing embedded system test features. The common part of the firmware uses about 123k (29%) Adaptive Logic Module (ALM) and 1084 (40%) RAM blocks of the available resources. The allocation of resources to the main functional blocks is shown in table 2 for the case where GBT links can be dynamically switched between GBT-mode and wide-mode.

**Table 2.** Table summarizing the FPGA resource usage of the various firmware components and their relative contribution to the total.

Main block name	Sub-block portion	ALM		RAM block (M20k)	
		count	% of total used	count	% of total used
GBT_wrapper	total	54269	44,1	0	0,0
	1 link (out of 24)	2143	1,7	0	0,0
PCIE0	total	12848	10,5	93	8,6
PCIE1	total	12278	10,0	93	8,6
BSP	total	1995	1,6	2	0,2
	i2c masters	1401	1,1	0	0,0
datapath wrapper 0	total	13498	11,0	441	40,7
	1 link (out of 14)	860	0,7	13	1,2
	bigfifo	62	0,1	258	23,8
	pktmuxfifo	1181	1,0	0	0,0
datapath wrapper 1	total	13308	10,8	441	40,7
DDG	total	706	0,6	0	0,0
slow control	total	1405	1,1	0	0,0
Example user_logic	total	533	0,4	0	0,0
serial flash and remote update	total	600	0,5	0	0,0
Readout control protocol	total	1023	0,8	1	0,1
	gbtmux	2564	2,1	0	0,0
TTS interface	total	5545	4,5	0	0,0
	pat player	335	0,3	0	0,0
	ctpemu	831	0,7	0	0,0
	onu	3604	2,9	0	0,0
Miscellaneous	total	2355	1,9	13	1,2
Firmware	total	122927	100,0	1084	100,0

From the table, it can be seen that the highest resource user is the GBT wrapper, which accounts for 44% of the total. Fortunately, for the most demanding detectors in terms of resource usage for the user logic, which is TPC, the firmware can be compiled in wide-bus mode, thus removing the code error corrector at compilation time it is possible to save up to 30k ALMs.

## 8 Conclusion

An adaptable common firmware was developed to cover the needs of the upgraded detectors of ALICE [14]. It was shown that by carefully designing the firmware features such as data path reader, CTP emulator, DDG, pattern player and by making them configurable, it allowed the development and validation effort to be shared between the firmware and the associated readout software. The different readout modes were validated by different detector setups in various running conditions. At this moment 565 CRUs are produced. The board distribution is over and the installation is done. The validation of the detector specific firmwares, the ones featuring user logic, is in progress. The commissioning shall be over by the end of 2021.

## References

- [1] ALICE Collaboration, Upgrade of the online-offline computing system *CERN-LHCC-2015-006, ALICE-TDR-019*.
- [2] J.P. Cachemiche et al., *The PCIe-based readout system for the LHCb experiment*, 2016 *JINST* **11** P02013.
- [3] M. Barros Marin et al., *The GBT-FPGA core features and challenges*, 2015 *JINST* **10** C03021.
- [4] I. Papakonstantinou et al., *Passive optical networks for the distribution of timed signals in particle physics experiments TWEPP09 proceedings* <https://doi.org/10.5170/CERN-2009-006>.
- [5] Broadcom, AFBR-811 optical transceivers, [manufacturer website](#).
- [6] *The GBT project website*, <https://espace.cern.ch/GBT-Project/GBT-FPGA/default.aspx>.
- [7] A. Caratelli et al., *The GBT-SCA, a radiation tolerant ASIC for detector control and monitoring applications in HEP experiments*, 2015 *JINST* **10** C03034.
- [8] A. Kluge and P. Vande Vyvre, *The detector read-out in ALICE during run 3 and 4 ALICE-TECH-2016-001*.
- [9] D.M. Kolotouros et al., *A TTC upgrade proposal using bidirectional 10G-PON FTTH technology*, 2015 *JINST* **10** C04001.
- [10] E.B.S. Mendes et al., *The 10G TTC-PON: challenges, solutions and performance*, 2017 *JINST* **12** C02041.
- [11] F. Costa et al., *The detector read-out in ALICE during run 3 and 4*, *J. Phys.: Conf. Ser.* **898** (2017) 032011.
- [12] F. Costa et al., *The readoutcard userspace driver for the new ALICE O2 computing system*, [arXiv:2010.16327](https://arxiv.org/abs/2010.16327).
- [13] Avalon interface specifications, MNL-AVABUSREF, [Altera website](#).
- [14] *Project gitlab*, <https://gitlab.cern.ch/alice-cru/cru-fw>.