Readout Unit for the LHCb experiment

Jose F. Toledo, Politechnical University of Valencia
Hans Muller, Francois Bal, Beat Jost, CERN, Geneva, Switzerland

Abstract

The Readout Unit (RU) for the LHCb data acquisition and trigger system is a programmable eventbuilder interface, taking input from up to four Gbit/s data link receivers, and outputting to single PCI nodes of a readout network. It has been conceived as a versatile and programmable prototype module that uses the CERN S-Link as link-receiver protocol [1], 9U- IEEE960 power and mechanics, and PCI bus protocols for the readout network. The FPGA-based receiver stage merges incoming event-fragments into subevent blocks that are stored in an intermediate subevent buffer. Subevents are forwarded by an FPGA-based eventbuilder interface logic that is directly connected via the PCI bus with the protocols of the readout network. An embedded I/O processor provides remote control over all input and output functionalities, via a 100Mbit/s LAN connection to the experiment control system.

1. PURPOSE OF THE READOUT UNIT

The Readout Unit (RU) is being developed to fulfil the functionality of the first stage of the LHCb data acquisition system [2] [Figure 1].

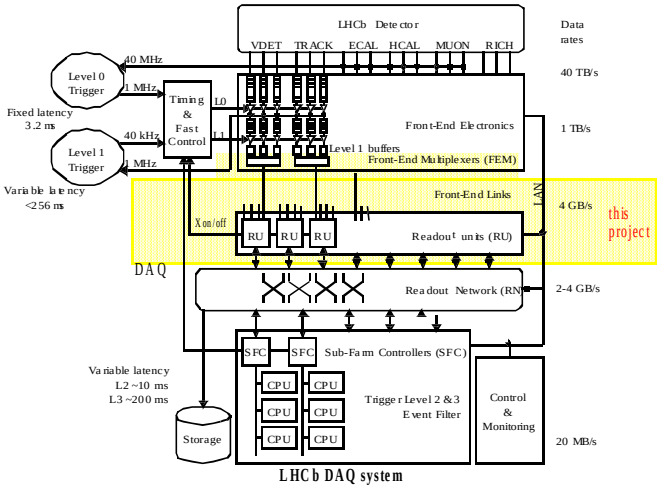

Figure 1: LHCb DAQ system

The overview below [Figure 2] shows it's top level architecture between S-Link receiver inputs and PCI outputs. The role of the merger is to collect event-fragments from the input FiFos and store them in the subevent buffer. The subevent builder assembles these into subevent blocks. For complete event building, subevents are transferred via the subevent builder interface to a PCI bus compliant readout network.

The RU is also equipped with an embedded CPU which is connected via a LAN to the remote control and monitoring system.

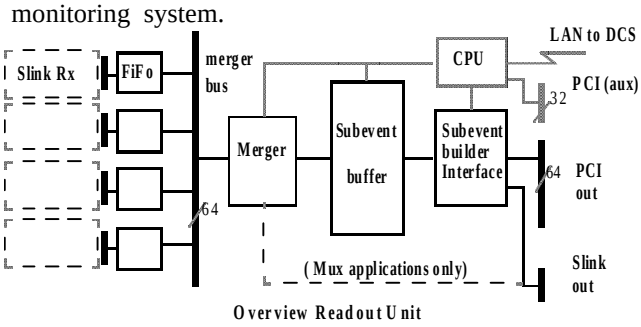

Figure 2: architecture of Readout Unit

1.1 Bandwidth requirements for LHCb

The nominal requirements for a Readout Unit are given by the LHCb level-1 trigger rate and event sizes per input link of the RU. The detector with the highest occupancy will generate max. 1 kByte per event including some formatting overhead. With four input links and a 40 kHz level-1 rate, this corresponds to a 160 Mbyte/s sustained throughput requirement for the RU. The requirements for the individual parts of the RU are summarised in [Table 1] below. Each link input requires 40 Mbyte/s input bandwidth, the output bandwidth is chosen 264 Mbyte/s by using 64-bit 33-MHz PCI (528 Mbytes/s for 66 MHz).

PCI output	64bit@33MHz (66 MHz)
Subevent Buffer	1 Mbyte
Peak input per link	40 Mbyte/s
Integral BW	160 Mbyte/s

Table 1: Bandwidth Requirements

2. THE RU MODULE

The Readout Unit is a 9U motherboard with modular I/O connections for data and control [Figure 3]. The input stage receives data fragments from front-end links via four S-Link receivers. Input FiFos are used to derandomize input data and to convert the output to a 64-bit format. The subevent merger (SEM) logic scans the FiFos, which share a 64 bit very high bandwidth bus, and writes their data to the subevent buffer (SEB), together with a directory entry. The output stage is the eventbuilder interface (EBI) which combines the data from the SEB into subevents which are then transmitted

via the PCI bus to the readout network (RN). The monitoring and control unit (MCU) provides remote control and access to data via a LAN connection. An auxiliary PCI slot allows for extended applications of the RU, such as the use of a Timing and Controls (TTC) receiver card [3].

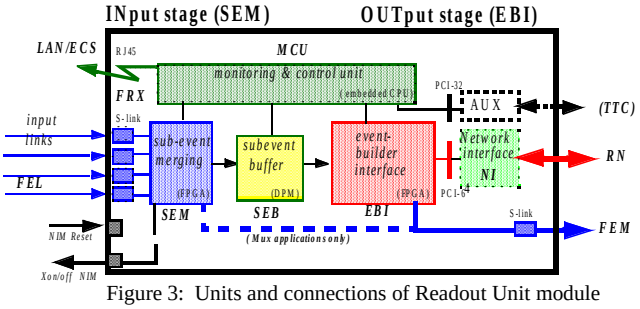

Figure 3: Units and connections of Readout Unit module

2.1 The subevent merger (SEM)

The SEM logic scans the 64-bit-wide FiFo outputs in a round-robin fashion and transfers event data blocks to the dual-port subevent buffer (SEB). After writing a block to the SEB, a pointer to its location is added to the directory table in the SEB.

As shown dotted in [Figure 3] there is a multiplexer option to transmit the input data unchanged directly to a single S-link transmitter output. In normal operation however subevents are built from the stored event-data, which are then sent as one block either to the PCI, or the Slink output.

2.2 Subevent buffer (SEB)

The SEB is physically a dual-port memory between the SEM and the EBI. Most of its capacity[1] is used for storing data, a small part is used a pointer directory. In order to support LHCb's "phased readout protocol" the SEB must be accessible like memory via the PCI network interface for selective readout. For the LHCb "full readout protocol", all events residing in the SEB are transferred to the readout network, using a fully event-driven data flow concept.

2.2 Eventbuilder Interface (EBI)

The EBI logic is the RU's output stage. It generates subevents from the data in the SEB by identifying all blocks which belong to the same event number. Such identified subevents are framed and reformatted as single output entities called subevents. The EBI is also part of the full eventbuilding system of the DAQ system and therefore transfers subevents according to a chosen readout protocol and network technology. For this prototype, the PCI standard industry bus is used as convenient and well-supported network interface between the RU's FPGA logic and the network. The full or phased LHCb eventbuilding protocols can be implemented using PCI either for message passing or for memory-like access.

2.3 The Monitoring and Control Unit (MCU)

A commercial plug-in processor card [4] is used for remote monitoring and control via a LAN connection to the Experiment Control System (ECS). Tasks of the MCU include monitoring the SEB buffer status, error handling and reporting and remote control of the operating parameters of the RU. Apart from this, the MCU can also initialise the PCI bus, re-load FPGA configuration bitmaps, access status registers inside the FPGAs, implement high-level data transport protocols, or emulate RU functions before implementing them in FPGA hardware. The MCU firmware will be developed using a cross-compiler (for Intel i960) on host PCs with calls to low level monitor services.

2.4 The Front-end Link Receivers (FLR)

For the RU prototype we opted for the use of the 32 bit S-Link protocols of the ATLAS experiment at the link receiver side. The use of S-link mezzanine cards allows for a free choice of either using short (copper) or long (fiber) technologies for the input links. At the S-link connector the same 32 bit framing protocol is available for all link technologies.

2.5 Network Interface (NI)

This is a network-specific PCI card which adapts a given readout network technology to the PCI output bus of the RU.

Since standard PCI cards are mechanically not compatible with 9U crate mechanics, a final version of the RU requires that this card is either available as PMC mezzanine or that its logic can be placed on the RU motherboard. An example of a PMC network interface can be found in [5]. For tests with standard PCI cards, a simple PCI-PMC adapter card may be used.

3. READOUT PROTOCOLS

The transfer of data between the RU's and the 2/3rd level CPU farm computers requires a transfer protocol between the RU's EBI logic and the 2nd/3rd level farm CPU's [6].

The full readout protocol of LHCb is meant to be simple: a continuous, write-only data transfer is maintained from source to destinations. The phased readout protocol only transfers subevents which have been pre-selected by the 2nd level decision. This drastically reduces the amount of data to be transferred over the network, however its implementation, i.e. selective readout, is complex. Both readout protocols are to be studied with the Readout Unit prototypes.

[1] 1 Mbyte is the nominal SEB capacity for the RU prototype

3.1 Full readout

Under the full readout protocol, the EBI output logic autonomously writes subevents via the PCI bus and the readout network (RN) to a destination CPU. The transmission via the RN can be either a memory mapped PCI-to-PCI transfer between sources and destinations (transparent), or a DMA between the SEB and a destination buffer (buffered I/O). The choice depends also on the possibilities of the RN.

Subevents are chained event-blocks in the SEB, with pointers stored in the SEB directory. Due to the use of circular buffers there is no need to implement a garbage collection mechanism. Buffer overflow in the SEB is prohibited via a Xon/Xoff return signal, generated by the input stage of the SEB logic.

3.2 Phased readout

Under this protocol, only small subsets of events are transmitted for making a 2nd level decision whilst the rest of the event is queued in the SEB. Since the remainder of the data is only transferred after a positive level-2 decision, the bandwidth across the RN network can be considerably reduced: non-accepted events are discarded from the SEB. For this protocol, the EBI logic needs to understand a set of messages like selective accept and reject from the RN.

4. SUBEVENT BUILDING

Subevents are built by the EBI output logic from the event-fragments stored in the subevent buffer. The event-data fragments stored in the SEB are identified by their event numbers which is used by the EBI output stage to build and frame subevents. The process of event building from a buffer is very tolerant for arrival spreads and allows in particular receiving several fragments of the same ID via the same link.

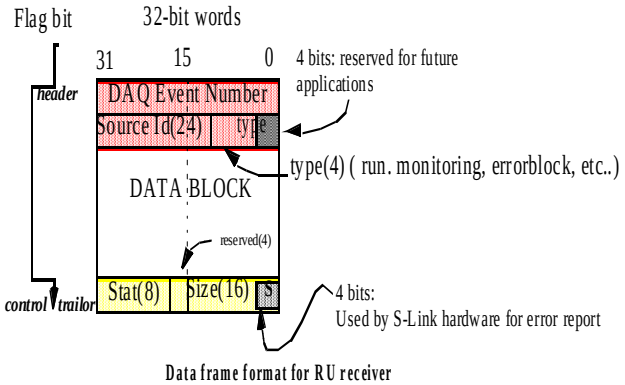

Figure 4: Link input format convention

4.1 Input Data formats

A very simple input format convention is adopted for the Readout Unit prototype [Figure 4]. The framing overhead is minimal and consists of two 32-bit words in the header and one in the trailor. The start and end of the input block is identified via a hardware flag bit. In order to ease subevent building, the event identifier is contained in the header word.

The block size and the error status are contained in the trailor word. Four LSB bits are used by S-link.

3.2 Subevent framing

A recursive and simple way of generating subevent blocks has been adopted. The data blocks of event frames belonging to the saem event number are concatenated behind the same header word as the input frames [Figure 5]. This requires that data blocks have their own consistent format which is transparent to the RU. The size and status fields for subevents are recalculated and appended as trailor word. An error block is inserted between the last data block and the trailor only in case of errors. The latter is indicated in the status word.

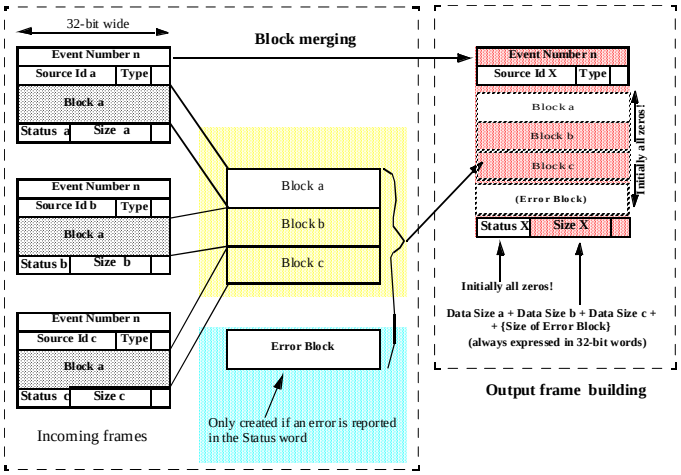

Figure 5: Recursive subevent framing of input events

5. TECHNICAL IMPLEMENTATION

The RU module 's technical layout is shown is shown below [Figure 6].

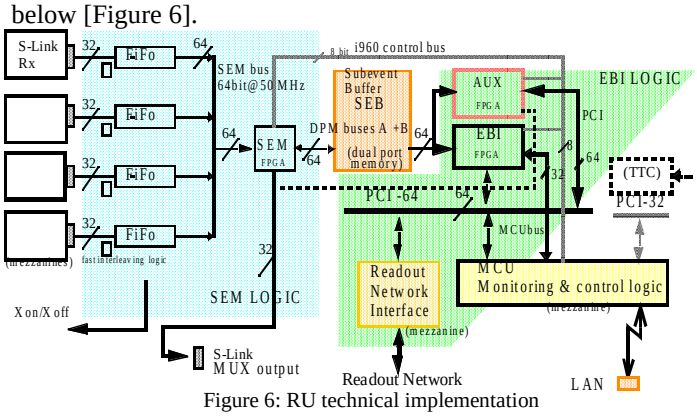

Figure 6: RU technical implementation

5.1 Internal bus connections

At the input side, the 32 bit S-Link words are converted via interleaved 32 bit FiFos to a 64 bit SEB bus

which is read out by the SEM FPGA with a clock rate of 50 MHz to provide a raw bandwidth of 400 Mbyte/s.

Input data are stored in a 64-bit-wide dual port memory interfaced between the SEM input and the EBI output stage. The latter consists of two parallel FPGAs (EBI and AUX) which share the 64-bit SEB bus at their inputs and the 64-bit PCI bus at their outputs. The reason for the parallel use of two FPGA's, apart from lower cost of the FPGAs is the increased PCI bandwidth which can be achieved with two tandem PCI masters[2]. The MCU is connected to the 64-bit PCI bus and its 32-bit memory bus is connected to the EBI FPGA. The latter allows that the MCU can map the SEB into its local memory space. A dedicated, 8-bit-wide i960 bus interconnects all FPGAs with the MCU. This allows for FPGA configuration and access to control of status registers via the MCU, and further via a remotely connected server. The MCU provides a secondary, 32-bit PCI port which may be used for auxiliary PCI cards.

5.2 Motherboard layout

The RU prototype is implemented as 9U motherboard which only takes power from an IEEE-960 crate mechanics. All 6 mezzanines for data input (4) and for data output (2) are located on either front-or backpanel of the motherboard. A preview on the module (currently under design) is shown below [Figure 7]

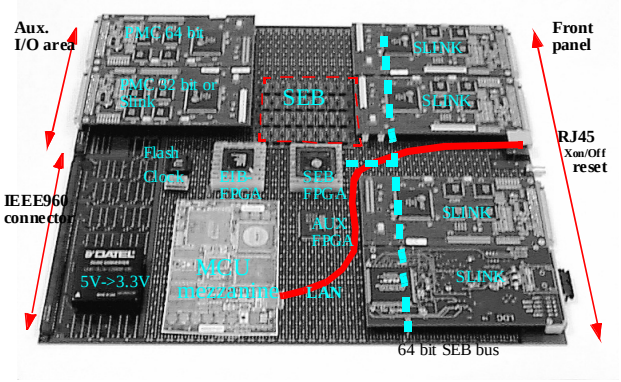

Figure 7: A preview of the 9U motherboard

5.3 Board design tools

The schematic capture was completed using Cadence Concept. The 12-layer board is designed using Cadence Allegro.
For critical buses design like for the 64 bit FiFo bus and the 66-MHz PCI bus, a signal integrity study using Cadence SpecctraQuest and IBIS models from different vendors was carried out. The results of this study also influenced the placement of components in the PCB [Figure 8].

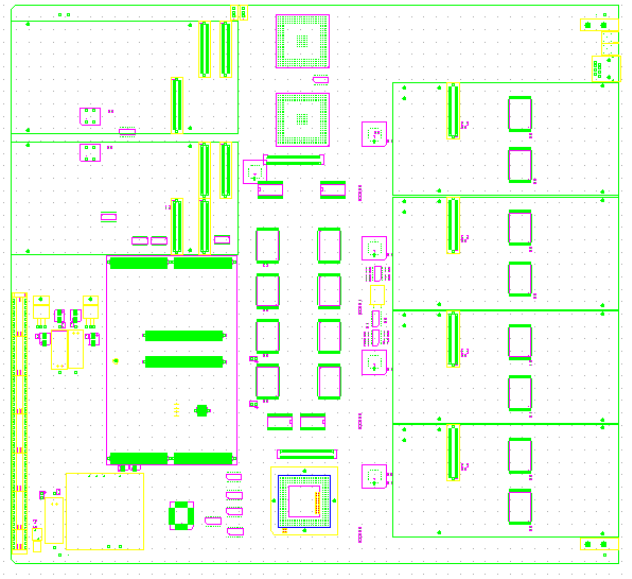

Figure 8: Floorplanning with Cadence Allegro

5.4 FPGA technology

Lucent Technologies' ORCA 3TP12 device [7] has been chosen for both FPGAs (EBI and AUX) in the output stage. This device has a 64-bit 66-MHz embedded master/target PCI hard core interface, as well as an 8-bit embedded hard core interface to i960 processors. The latter interfaces directly to the MCU's local bus and allows for remote configuration, program readback and access to internal FPGA. Approximately 35k equivalent logic gates are available for the EBI logic.

The SEM FPGA in the input stage (ORCA 3T55) is also a member of Lucent Technologies' 3T family. The differences with the previous device are the available user logic size (up to 55k equivalent gates). This device has also an i960 interface but no PCI core [8].

5.5 FPGA development tools

Like for previous design projects we use a VHDL design methodology for ORCA FPGAs from Lucent Technologies. After a VHDL simulation with Cadence LEAPFROG, the VHDL output is synthesised by Exemplar's LEONARDO to generate an EDIF output which is used by Lucent's Foundry tools to place and route all required FPGA resources. The bitstream output is programmed into a Flash Eprom, which during power-up transfers its contents to all FPGAs in the Readout Unit.

[2] In a future RU version we expect that a single FPGA can be used.

6. STATUS AND OUTLOOK

The Readout Unit project is a joint activity of the CERN EP-ED electronics group and the LHCb experiment. A series of meetings [9] have led to the design decisions for the first prototypes to be built in 1999. Since technology choices for input links, readout network and control system are pending, we have opted for a modular approach using S-Link mezzanines for the link input, PCI for the readout network and T-base 100 LAN technology for the control system.

This module is meant to be reproduced in small quantity and serves mainly the purpose for developing the final RU concept. A second-generation module will be started when technology and protocol choices have been taken.

REFERENCES

1. For information on S-Link, see the Slink homepage http://www.cern.ch/HSI/s-link/
2. LHCb collaboration, LHCb Technical Proposal, CERN/LHCC 98-4, Chapter 13
3. Estudio y Desarrollo de Nuevos Sistemas de Temporizacion, Trigger y Control para los Futuros Detectores de LHC del CERN, doctoral thesis on the TTCsr mezzanine Jorge Ferrer, CERN / Politechnical University of Valencia (to appear in English)
4. PXECORE embedded CPU mezzanine module from CompuLab. See http://www.compulab.co.il
5. Design of a high performance PCI interface for an SCI network, COMPUTING & CONTROL engineering journal, December 1998, F.Mora et al.
6. The LHCb Trigger and Data Acquisition System, presented at the IEEE Real Time Conference June 14-18, 1999 Santa Fe, NM RT 99, Santa Fe, 15 June 1999. B.Jost et al.
7. Lucent Technologies' ORCA 3PT12 data sheet. http://www.lucent.com/micro/fpga/hardware.html.
8. Lucent Technologies' ORCA 3C/T FPGA data sheet. http://www.lucent.com/micro/fpga/hardware.html.
9. For history and design documents, see the Readout Unit home page. http://nicewww.cern.ch/~hmuller/lhcb.htm