The Read Out Controller for the ATLAS New Small Wheel

**R.-M. Coliban^{a*}, S. Popa^a, T. Tulbure^a, D. Nicula^a, M. Ivanovici^a, S. Martoiu^b,
L. Levinson^c and J. Vermeulen^d**

^a*Transilvania University,*

Brasov, Romania

^b*IFIN-HH,*

Magurele, Romania

^c*Weizmann Institute of Science,*

Rehovot, Israel

^d*Nikhef,*

Amsterdam, Netherlands

E-mail: coliban.radu@unitbv.ro

ABSTRACT: In the upgrade process of the ATLAS detector, the innermost stations of the endcaps (Small Wheels) will be replaced. The New Small Wheel will have two chamber technologies, small-strip Thin Gap Chambers and Micromegas, each providing triggering and precision track measurement. Custom front-end Application Specific Integrated Circuits will be used to read and filter information from both types of detectors. In the context of the New Small Wheel data path, the Read Out Controller ASIC is used for handling, preprocessing and formatting the data generated by the VMM upstream chips. The Read Out Controller will concentrate the data streams from 8 VMMs, filter data based on the ATLAS Level-0 trigger which identifies bunch crossings of interest and transmit the data to FELIX via the L1DDC. The Read Out Controller is composed of 8 VMM Capture modules, a cross-bar and 4 SubROC modules. The output data is sent via up to 4 serial links with a configurable speed of 80, 160, 320 Mbps per link.

KEYWORDS: Read Out Controller; ASIC; ATLAS detector; New Small Wheel.

*Corresponding author.

Contents

1. Introduction	1
2. Level-0 de-randomization circuit for the VMM	2
3. Design of the Read Out Controller	3
3.1 VMM Capture	4
3.2 xBar	5
3.3 sROC	6
4. Validation & Testing	8
5. Conclusions	8

1. Introduction

To cope with the increased luminosity and background rates from LHC Run 3 and beyond, the innermost muon station of the ATLAS [1] endcap, the Small Wheel, is being replaced by the New Small Wheel (NSW) [2]. Two different detector technologies, small-strip Thin Gap Chambers (sTGC) and Micromegas (MM), will each provide both triggering and precision track measurement. In order to meet constraints of bandwidth, latency and radiation tolerance, the front-end electronics will include four new ASICs for digitizing and buffering the detector data: the Trigger Data Serializer (TDS), Address in Real Time (ART), VMM and Read Out Controller (ROC). Figure 1 depicts a block diagram of a front-end board containing the four chips. The VMM is the front-end ASIC that receives the analog signals from the detector. Per channel discriminators and ADCs send data to both the trigger and readout paths. The former is prompt, the latter is buffered awaiting a trigger signal. The TDS and ART are designed to transmit trigger data, while the ROC transmits data to the readout path. The data from the ROC is received by a Gigabit Transceiver Link (GBT) which sends it via fiber to a general purpose network with a high-availability interface, called FELIX (Front End LInk eXchange) [3]. The data, tagged by BCID, is buffered in the VMM until a Level-0 trigger calls for it to be transferred to the ROC. The ROC will aggregate data from up to 8 VMMs and send it either unfiltered (after Phase 1 of the ATLAS detector upgrade) or filter it based on a Level-1 trigger (after Phase 2).

Figure 1: Block diagram of the NSW front-end board.

2. Level-0 de-randomization circuit for the VMM

The de-randomization circuit is part of the future VMM3 front-end ASIC to be used by the ATLAS New Small Wheel (NSW) detector. The VMM front-end produces asynchronous data for each incoming signal which crosses a particular threshold. The outputs for DAQ are 10-bit charge of strips, 1ns time resolution of the peak time. The information is forwarded to the digital de-randomization block. The digital block inside the VMM (Figure 2) merges data from 64 front-end channels, performs data de-randomization and data selection.

Figure 2: Block diagram of the VMM de-randomization circuit.

The purpose of this block is to hold the incoming hit data until it is selected by an external trigger which arrives at a fixed latency after the event. Due to the intrinsic spread of the drift times in the MM gap that is of the order of 100 ns, the hit data corresponding to a given event may be

inside a broader temporal window of a few bunch-crossings (BC) periods. The front-end circuit of the VMM chip has a finite deadtime of the order of 200 ns, the time needed for the digitization of the incoming signal. The maximum possible trigger window is constrained to the same value, such that there can be no consecutive pulses which fall inside the same trigger window.

The selection mechanism uses the timing information embedded in each data sample to select data hits corresponding to each trigger window. In order to preserve the temporal order of the hits in the queues, the data is stored by the digital block in separate FIFOs for each input channel. A selection circuit at the output of each queue examines the timestamp information of the oldest data sample in the queue and decides whether it has to be copied to the output, if it falls inside a trigger window, or discarded if it is too old to be triggered by any future trigger. A particular data sample may fall inside the trigger windows of two or more consecutive triggers, if the triggers occur close in time. The selection mechanism is designed to take this situation into account and replicate the data sample in all trigger packets concerned.

The output stream of data is segmented into data packets corresponding to each trigger signal. These packets are stored into a back-end buffer until they are transmitted to the ROC via a two-wire serial interface with 8b/10b encoding.

A queue management mechanism is implemented on this FIFO in order to preserve the packet synchronization in case of an overflow condition, by sending special empty packets to the backend system until normal operation is restored.

3. Design of the Read Out Controller

The Read Out Controller (ROC) will be an ASIC designed to forward data from the VMM chips towards FELIX, while accommodating the trigger architectures designed for both Phase 1 and Phase 2 of the ATLAS detector upgrade. The ROC will aggregate data from up to 8 VMM chips. Each VMM data connection is a 640 Mbps 2-bit serial link, with the data being *8b/10b encoded*.

The format of the data packets is depicted in Figure 3 (top). Each packet consists of a header (identified by a MSB value of zero) containing a timestamp consisting of two data words: orbit and Bunch Crossing ID (BCID), followed by zero or more hit words, representing the data that corresponds to the Level-0 trigger. One or more *comma* characters, *K28.5*, are used to separate data from different events. In Phase 2, the ROC will filter the data based on an orbit-BCID pair

Figure 3: Format of the VMM output / ROC input data packets (top) and format of the ROC output data packets (bottom).

corresponding to a Level-1 trigger; in Phase 1 there will be no such filtering. The data will be repackaged and forwarded on up to 4 E-Links. The format of the output packet is depicted in Figure 3 (bottom). If no hit data corresponding to the orbit-BCID pair is found in the VMMs, a null header event is sent. Otherwise, the output data will consist of a header, one or more hit data words and a trailer.

In addition to VMM data filtering and forwarding, the ROC will also forward TTC information to the VMM chips, and also provide clock signals to other front-end ASICs.

The block diagram of the Read Out Controller is depicted in Figure 4. The VMM Capture module receives data packets from a VMM on the serial interface and stores them in a FIFO in order to be read by the sROC modules. The xBar module is a switching fabric interconnecting 8 VMM Capture modules to 4 sROCs in a configurable manner. The sROC aggregates the data from the corresponding VMMs and forms packets which are sent via an E-link serial interface. The configuration of the chip and error reporting is done by the Config block, which communicates with an SCA ASIC via an I2C interface. The TTC information from FELIX is processed and forwarded by the TTC Decode module, while the clock signals are generated by the ePLL block. The design of the Config and TTC Decode modules is still in progress. The ePLL design is taken from the GBT project [4]

Figure 4: Block diagram of the Read Out Controller.

3.1 VMM Capture

The VMM Capture module is composed of the following blocks: a deserializer (DES), a COMMA ALIGN module, a wrapped 8b/10b decoder (VMM DEC), an 8b to 32b word assembler (ASSEMBL) and a FIFO module; the blocks, along with the data path, are depicted in Figure 5.

The DES module captures the data on the serial lines connected to the VMM, which are DDR on a 160 MHz clock; the odd bits are on one line and the even bits on the other. The (8b/10b encoded) data being expected to be received LSB-first. For example, for a 10-bit word *abcdefghij* (*a* is the LSB), the serial input format is the one presented in Figure 6.

Figure 5: Block diagram of the VMM Capture module.

Figure 6: VMM Capture input format.

The COMMA ALIGN module monitors the 10-bit output of the DES for the presence of the comma character *K28.5*. Once this character is identified in the input stream, the 10-bit data is aligned to the proper boundaries, thus achieving synchronization with the VMM and enabling the data transfer between the VMM and ROC. The module can sense if the data alignment is lost due to transmission errors (no *comma* characters received in a predefined time interval) and is capable of achieving resynchronization.

The VMM DEC (VMM Decode) module decodes the 10-bit input, with the result being on the 8-bit output. The module is capable of signaling if an illegal character, a running disparity error or decoding error was encountered. 8b/10b encoding [5] [6] is used in the communication between the ASICs in order to have an out-of-band byte alignment and event framing using the *comma* symbol.

The ASSEMBL module assembles 8-bit words received from the decoder into 32-bit words, corresponding to the data words in Figure 3 (top), which are written in the FIFO; the header word is zero-padded. An additional output bit is present in the data bus, which is used to indicate the end of the packet. The module uses the received *comma* characters in order to delimit the incoming packets and thus is capable of correcting start bit errors (header is received with MSB = 1). The module can also recover from a state of error generated by receiving an incomplete packet (with a last hit data word of 8, 16 or 24 bits), resulting in only a single corrupted data packet sent forward.

The FIFO module stores the 33-bit words and implements the clock domain crossing, between 160 MHz and 40 MHz.

3.2 xBar

The xBar module connects 8 VMM capture modules to 4 sROC modules in a full-mesh network. The data issued by the VMM capture modules is directed to sROC together with the *FIFO empty* control signals. The read signals from sROC are directed to VMM Capture FIFOs. The xBar module is fully combinational (see Figure 7) and consists of 4 MUXes on sROC side (for data and *FIFO empty* - not depicted in Figure), and 8 MUXes on the VMM Capture side and 4 MUXes on the sROC side (for *FIFO read*). The selection is based on configuration inputs as follows: each VMM Capture receives 4 selection bits, corresponding to the destination sROC and each sROC receives 3 selection bits, corresponding to the destination VMM Capture index.

The connectivity between the VMM Capture modules and the sROCs can be configured in any desired way. The options which are more likely to be used for the sTGC and MM detectors are presented in Table 1.

Table 1: Common connectivity options for xBar.

Configuration	sROC 0	sROC 1	sROC 2	sROC 3
"2222"	VMM 0, 7	VMM 1, 6	VMM 2, 5	VMM 3, 4
"422"	VMM 0, 7, 1, 6		VMM 2, 5	VMM 3, 4
"44"	VMM 0, 7, 1, 6		VMM 2, 5, 3, 4	
"224"	VMM 0, 7	VMM 1, 6	VMM 2, 5, 3, 4	
"8"	VMM 0...7

Figure 7: Schematic of the xBar module.

3.3 sROC

The sROC module reads data from up to 8 VMM capture modules, extracts the hit data which corresponds to the orbit-BCID pair of a Level-1 trigger received from the TTC Decode module and forms packets which are encoded using 8b/10 and sent over a serial E-link, with a configurable speed of 320, 160 or 80 Mbps. The block diagram of the module is depicted in Figure 8.

The TTC FIFO stores data received from the TTC Decode module. The format is: {orbit (4b), BCID (12b), L1ID (16b), L0ID (16b)}. The orbit-BCID pair is used by the PACKET BUILDER in order to filter the data read from the VMM Capture FIFOs. The L1ID and L0ID appear in the output packet header and trailer, respectively (see Figure 3). After the Phase 1 upgrade, the trigger information stored in the TTC FIFO is the same as that stored in the VMM ASICs, corresponding to the Level-0 trigger; thus, there will be no hit data filtering. However, after the Phase 2 upgrade, the information stored in the FIFO will be only a fraction of the VMM TTC information, corresponding to the Level-1 trigger, thus some of the data buffered in the VMM Capture module will be discarded.

The PACKET BUILDER module is connected to the corresponding VMM Capture modules via the xBar. The module contains a finite state machine (FSM) which controls the process of reading and filtering the data from the VMM Capture FIFOs. Outputs were modeled in a Mealy fashion to minimize processing latency of the state machine. Essentially, when an orbit-BCID is received on the TTC channel, the module reads all the connected VMM Capture FIFOs in a round robin fashion, discarding the packets with a smaller BCID than the one received from the TTC. A packet will be formed with all the hits with the same orbit-BCID pair as the one received in the TTC FIFO.

The state diagram of the FSM inside the PACKET BUILDER is depicted in Figure 9. In the initial state (S_WAIT_TTC), the module waits for data to be available in the TTC FIFO. When there is data, it is loaded and the internal registers are initialized such as to start checking the VMM Capture FIFOs (S_LOAD_TTC). After this step, the header in the currently connected FIFO is analyzed based on the orbit-BCID pair received on the TTC channel (S_CHECK). If the data in the FIFO is older than the TTC data, the corresponding packets are discarded from the buffer (S_FLUSH). If no packet is found that corresponds to the TTC orbit-BCID pair, an *event miss* error is asserted and the FSM proceeds to the next VMM Capture module in the list. Similarly, if there is no hit data after a header with a matching orbit-BCID pair, the FSM will go to the next VMM. In the

Figure 8: Block diagram of the sROC module.

Figure 9: Packet Builder FSM state diagram.

case that at least one of the VMM Capture buffers contains valid data, a header word is written in the PACKET FIFO (S_SEND_HEADER). Whenever matching hit data is found in the current FIFO, it is read in burst mode and written in the PACKET FIFO. When the FSM has gone through all the VMMs, two cases are possible: either (i) no matching hits were found - then a null header is written in the PACKET FIFO (S_SEND_NULL_HEADER) or (ii) there are hits for the corresponding orbit-BCID pair - then the packet checksum is computed (S_PREPARE_TRAILER) and the packet trailer is written in the PACKET FIFO (S_SEND_TRAILER). The FSM then resumes waiting for new data in the TTC FIFO.

The STREAMER module creates the sROC output data stream, which is divided into 32-bit words. The module reads the data in the Packet FIFO and forwards it, while adding start-of-packet (K28.4) and end-of-packet (K28.6) characters where appropriate. The module also sends "comma" characters (K28.5) continuously if there are no full packets in the Packet FIFO. The sROC ENC module receives 32-bit data words from the Streamer and outputs 40-bit words, which represent the 8b/10b encoding of the input. There is only one internal 8b/10b encoder used, while the input data is time-multiplexed. The FEEDER module receives 40-bit data words from the sROC Encoder and outputs an 8-bit word every 40 MHz clock cycle to the E-LINK TX. The E-Link output is a differential serial line which can transmit data at 320, 160 or 80 Mbps. The E-LINK TX module manages the clock domain crossing between the 40MHz clock and the 160 MHz clock and serializes the data received from the FEEDER.

4. Validation & Testing

In order to validate the design of the ROC, each block was simulated and verified before integrating it in the top level module. After integration into the ROC, the VMM output was emulated and two types of tests were carried: (i) Tests with correct input data streams, to test the functionality of the ROC; (ii) Tests with input data streams containing errors, in order to emulate various possible scenarios (misaligned data, 8b/10b coding errors, incomplete packets, erroneous start-of-packet) and to test the robustness of the ROC and the correct error signaling in the various blocks such as COMMA ALIGN and VMM DEC. The input traffic was emulated using a Poisson distribution, our custom-design traffic generators being fully configurable (average data rate, average packet size, errors and so on). The modules designed and implemented so far have been validated after the tests.

The Verilog implementation was simulated using ModelSim. The physical design stage is being performed with Cadence tools, using the IBM 130-nm technology library.

5. Conclusions

The Read Out Controller is being implemented according to specifications [7]. The ROC consists of 8 VMM Capture modules, an xBar and 4 sROC modules. It aggregates the data from a number of VMMs and delivers the resulting packets on up to 4 serial links. Simulations proved the correct implementation of the ASIC functionality so far, according to the specifications.

Acknowledgments

This work is partially funded by the Romanian Ministry of Education and Research, RO-CERN collaboration, project "ATLAS experiment at LHC", contract no.7/ January 2012.

References

- [1] ATLAS Collaboration. The ATLAS Experiment at the CERN Large Hadron Collider. *Journal of Instrumentation*, 3(8):S08003, 2008.
- [2] ATLAS Collaboration. New Small Wheel Technical Design Report. 2013, CERN-LHCC-2013-006, ATLAS-TDR-020, <http://cdsweb.cern.ch/record/1552862>.
- [3] Jorn Schumacher et al. FELIX: a High-Throughput Network Approach for Interfacing to Front End Electronics for ATLAS Upgrades. in Proceedings of the CHEP 2015 conference, Okinawa, Japan, April 2015, ATL-DAQ-PROC-2015-014, <http://cdsweb.cern.ch/record/2016626>.
- [4] P. Moreira et al. The GBT Project. in Topical Workshop on Electronics for Particle Physics, Paris, France, 21 - 25 Sep 2009, pp.342-346, <http://cdsweb.cern.ch/record/1235836>.
- [5] A.X. Widmer and P.A. Franaszek. A dc-balanced, partitioned-block, 8b/10b transmission code. *IBM Journal of Research and Development*, 27(5):440–451, Sept 1983.
- [6] P.A. Franaszek and A.X. Widmer. Byte oriented dc balanced (0,4) 8b/10b partitioned block transmission code, December 4 1984. US Patent 4,486,739.
- [7] The NSW Readout Working Group. Requirements for the NSW VMM3 readout ASIC the NSW Readout Controller ASIC, April 2015. Design Review Report, unpublished