



BNL-213684-2020-JAAM

## The VMM readout system

T. Alexopoulos, G. Iakovidis

To be published in "Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment"

March 2020

Physics Department  
**Brookhaven National Laboratory**

**U.S. Department of Energy**  
USDOE Office of Science (SC), High Energy Physics (HEP) (SC-25)

Notice: This manuscript has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy. The publisher by accepting the manuscript for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes.

## **DISCLAIMER**

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

# The VMM readout system

T. Alexopoulos<sup>a</sup>, D. Antrim<sup>b</sup>, C. Bakalis<sup>a,c,\*</sup>, G. De Geronimo<sup>d</sup>, P. Gkoutoumis<sup>a,c</sup>, G. Iakovidis<sup>c</sup>, P. Moschovakos<sup>a,c,1</sup>, V. Polychronakos<sup>c</sup>, A. Taffard<sup>b</sup>

<sup>a</sup> National Technical University of Athens, 15780 Athens, Greece

<sup>b</sup> University of California Irvine, Irvine, CA 92697, USA

<sup>c</sup> Brookhaven National Laboratory, Upton, NY 11973, USA

<sup>d</sup> DG Circuits, NY, USA



## ABSTRACT

The New Small Wheel Upgrade of the ATLAS experiment at CERN, planned to take place at 2020, requires a new generation of front-end electronics that will support its data acquisition requirements. The VMM Application-Specific Integrated Circuit has been in development for the last seven years to serve as the foundation of the New Small Wheel's readout scheme. It has gone through three major revisions and a minor one, the latter being the production version. To facilitate the testing and readout of the VMM, as well as to study its detector performance, a complete readout system has been developed. It consists of flexible Field-Programmable Gate Array logic with extensive functionality, and an efficient software framework providing the user interface. This system, referred to as the "The VMM Readout System", has been used in test-beam campaigns at CERN, as well as in bench calibration and testing measurement scenarios for the past several years, supporting all readout modes and features of the VMM.

## 1. Introduction

The upgrades of the Large Hadron Collider (LHC) at CERN in 2019–20 (LS2) and 2024–26 (LS3) will substantially increase the instantaneous luminosity. To cope with the increased particle flux, the LHC experiments will be upgraded as well. During LS2, each of the so-called "Small Wheels" of the ATLAS muon spectrometer, that cover a region of  $1.0 < |\eta| < 2.7$  for muon tracking and  $1.0 < |\eta| < 2.4$  for triggering, are planned to be replaced by the "New Small Wheel" (NSW). The NSW comprises two gaseous detector technologies: the Micromegas (MM), mainly used for track reconstruction, and the small strip Thin Gap Chambers (sTGC) mainly used for triggering [1]. The 2.4 million readout channels of those chambers will require a new generation of electronics to read them out. The keystone of the NSW's electronics system is the VMM Application-Specific Integrated Circuit (ASIC), which connects to the detector readout elements to provide trigger and tracking muon data to the ATLAS experiment [1,2]. The VMM has undergone several revisions over the past few years. Throughout this process, the need for developing a VMM Readout System (VRS) became a necessity. VRS is able to calibrate, test, and perform detector measurements using the VMM, while implementing the same firmware, hardware, and software framework. In this paper, the various components of the VRS are described.

The VRS supports a plethora of prototype front-end boards that house VMMs and configurable logic chips that interface with the ASIC, i.e. Field-Programmable Gate Arrays (FPGAs). The firmware of the front-end FPGA is adaptable to different readout scenarios, boards and FPGA packages and pinouts. A typical front-end board that can be used within this system is the Micromegas Front-End 1 (MMFE1), which is also presented here. In addition, the VRS Supervisory Board (VSB) is another component of the readout system, that can interface with more than one front-end nodes and maximize readout efficiency in multiple-board implementations. On the back-end, a software suite named VMM Ethernet Readout Software (VERSIO) interfaces to the FPGAs over a commercial Ethernet network using the User Datagram Protocol (UDP). The software acts as an interface for the user and can also accumulate readout data from the VMMs, sent to it via the FPGAs, to build data files for offline data analysis.

## 2. The VMM ASIC

The VMM is intended to be used in the front-end readout electronics of both the MM and sTGC detectors of the NSW Phase-I upgrade project of the ATLAS experiment at CERN [1]. It has also been proposed for several other applications and smaller-scale experiments. Developed at Brookhaven National Laboratory, the VMM is fabricated in the 130 nm

\* Corresponding author at: National Technical University of Athens, 15780 Athens, Greece.

E-mail address: [christos.bakalis@cern.ch](mailto:christos.bakalis@cern.ch) (C. Bakalis).

<sup>1</sup> Currently at CERN, Geneva 1211, Switzerland.

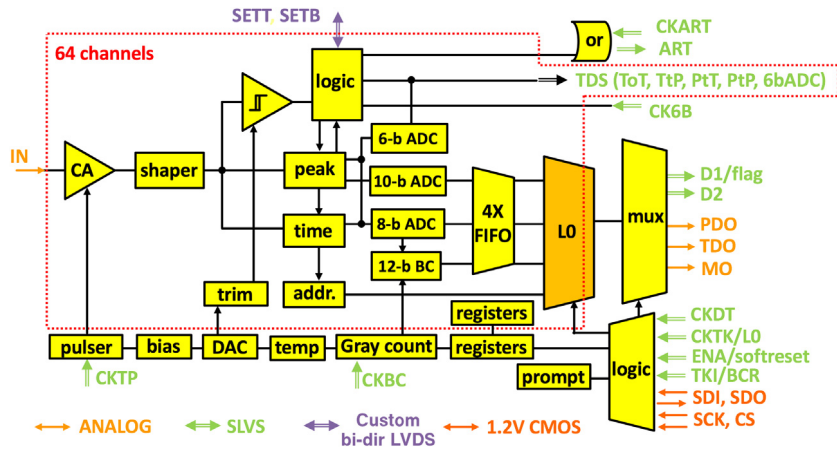


Fig. 1. Architecture of the VMM. The majority of the block diagram describes one of the identical 64 channels of the chip. The CA and shaper process the input pulse, while the ADCs digitize the pulse height and timing information, before storing their results in the channel's FIFO. The LO logic selects data based on the trigger input (LO) and timestamp information of the event. The data are sent to the external readout device via two lines, D1 and D2, synchronized with a clock provided to the CKDT input. Three outputs, useful for monitoring, drive the peak detector (PDO), time detector (TDO) and channel monitor (MO) voltages out of the chip. The test pulse strobe (CKTP) input drives the channel's testing circuitry, while the reference clock is provided via the CKBC pin. The chip can be reset by issuing a pulse in the ENA or TKI pins. The configuration of the ASIC is conducted via the SDI, SDO, SCK and CS pins. Finally, the trigger primitives are sent out of the chip via the ART, ToT, TtP, PtT, PtP and 6bADC outputs, synchronized with the CKART and CK6B clock inputs.

Global Foundries 8RF-DM process (formerly IBM 8RF-DM). The device is packaged in a Ball Grid Array (BGA) with a footprint of  $21 \times 21 \text{ mm}^2$ . It is composed of 64 identical and independent input channels. A block diagram of one of these channels is shown in Fig. 1. Each channel integrates a low-noise charge amplifier (CA) with adaptive feedback, test capacitor, and adjustable polarity (in order to process either positively or negatively charged input signals). The input MOSFET is p-channel. The filter (shaper) is third-order with one real pole and two complex-conjugate poles, designed in Delayed-Dissipative Feedback (DDF) [3]. It has adjustable peaking time in four values (25, 50, 100 and 200 ns) and stabilized, band-gap referenced, baseline. The gain is adjustable in eight values (0.5, 1, 3, 4.5, 6, 9, 12 and 16 mV/fC).

Next to the shapers are the sub-hysteresis discriminators [2] with neighbor-enabling logic and individual threshold trimming, the peak detector, and the time detector. The threshold is adjusted by a global 10-bit Digital-to-Analog Converter (DAC) and an individual-channel 5-bit trimming DAC. The neighbor-channel logic forces the neighboring channels to a triggered one to perform measurements, even if those channels did not exceed the set threshold. The neighbor logic extends also to the two neighboring chips through bidirectional IOs. The peak detector measures the peak amplitude and stores it in an analog memory. The time detector measures the timing using a Time-to-Amplitude Converter (TAC), i.e. a voltage ramp that starts either at threshold crossing, or at the time of the peak and stops at the next falling edge of the reference clock. For the ATLAS implementation, this is the Bunch-Crossing (BC) clock with a frequency of 40.079 MHz. The TAC value is stored in an analog memory and the ramp duration is adjustable in four values (60, 100, 350 and 650 ns). The peak and time detectors are followed by a set of three (6-, 8-, and 10-bit) low-power Analog-to-Digital Converters (ADCs) [4]. Which combination of these ADCs are enabled depends on the selected mode of operation.

The ASIC includes global and acquisition resets and a configurable pulse generator connected to the injection capacitor of each channel, adjustable with a global 10-bit DAC, and triggered by an external pulse. A band-gap reference circuit and a temperature sensor complete the basic features of the VMM. Finally, the ASIC integrates analog monitor capabilities to directly inspect the global DACs, the band-gap reference, the temperature sensor, the analog baseline, the analog pulse, and the channel threshold (after trimming).

## 2.1. Readout modes

The VMM has three modes of operation: a two-phase analog mode, a continuous simultaneous read/write mode, and the Level-0 mode. Also, the chip provides trigger primitives via its fast outputs.

### 2.1.1. Two-phase analog mode

In the two-phase mode, data are registered while the VMM is in acquisition mode and then are read-out once the system is switched to the readout mode. Acquisition is re-enabled after the data extraction phase is completed. The readout advances to the next channel by injecting a token to the relevant input. The token is sparse, passed only among those channels with valid events. Once the procedure is complete, the token is routed to the output for reading-out the next chip, thus allowing for a daisy-chained readout with a single token input. In this mode, the internal ADCs are switched off and the analog signals are read-out through analog buffers with external ADCs.

### 2.1.2. Continuous mode

In continuous mode, the simultaneous read/write of data assures dead-timeless operation that can handle rates up to the maximum of 4 MHz per channel.<sup>2</sup> Higher rates can be achieved by interrupting the 10-bit ADC once the 6-bit ADC has finished, since the 6-bit ADC conversion takes up  $\sim 40 \text{ ns}$  including the channel reset. The peak and time detectors convert the voltages into currents that are routed into the 10- and 8-bit ADC, respectively. The channel is reset once both conversions are completed and the digital values are latched in digital memories. The self-reset of each channel provides continuous and independent operation of all 64 channels. In this mode, each channel of the ASIC provides the 38-bit data stored in a four-event deep de-randomizing FIFO. The data are read-out via two serial data lines (D1 and D2 in Fig. 1).

### 2.1.3. Level-0 mode

The Level-0 (LO) readout mode of the VMM was designed to be compatible with the readout scheme of the ATLAS experiment. It provides the ability to buffer the channel hit data in a deeper memory architecture and allows the VMM to select the data to be read-out after the reception of an external trigger input, namely the LO signal. The LO selection logic applies a configurable latency to the trigger

<sup>2</sup> This is a result of the 250 ns conversion time of the internal ADCs.

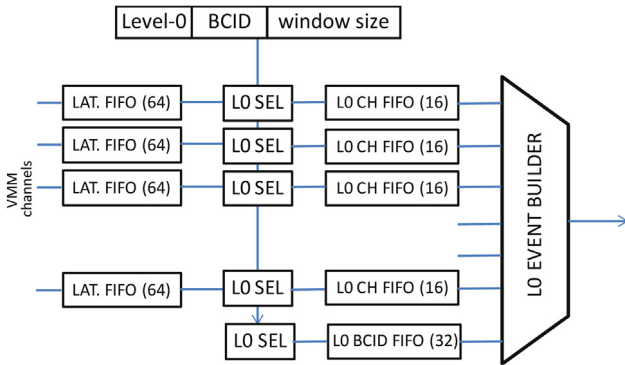


Fig. 2. LO buffer and event selection logic block diagram. The trigger signal, or LO, gets timestamped with a Bunch Crossing ID (BCID) value, then a configurable offset is applied to it. The hits stored in each 64-event-deep latency FIFO (LAT. FIFO) have a timestamp themselves as well. The LO selection logic (LO SEL) checks the hit timestamps against the ones of the trigger. Only hits within a window of configurable width enter the LO CH FIFO before being sent out of the ASIC by the LO Event Builder.

and forwards out only hit data for which their timestamp is within a window of the trigger’s time-of-arrival plus its latency offset. A block diagram of the logic is depicted in Fig. 2, where data from the VMM channel(s) are forwarded from left to right through various buffers. The first FIFO is a 64-deep latency buffer (LAT. FIFO in Fig. 2) where all hits can be buffered. Each channel has a selector circuit based on the arrival of the external trigger signal which finds events within the configurable window<sup>3</sup> (LO SEL in Fig. 2). Once there is a match, the data are copied to the channel FIFO (LO CH FIFO in Fig. 2) and are ready to be read-out in a round-robin manner, starting with the information of the BC timestamp. The data are forwarded out of the ASIC via two serial lines, synchronous to both edges of an externally-provided 160 MHz data clock, resulting in a total bandwidth of 640 Mb/s. The data are encoded using the 8b/10b protocol [5], leading to an effective bandwidth of 512 Mb/s.

#### 2.1.4. Fast outputs

The VMM has several fast serial outputs that can be used for different applications:

- The Address in Real Time (ART): This is the address (i.e. VMM channel number) of the first in time channel above-threshold. Depending on the chosen configuration parameters on the VMM, the total latency of the streamed ART address can be within  $\sim 15$  ns, or  $\sim 20$  ns plus the peaking time. This latency is the sum of several delays, some of which are the 5 ns delay between the peak and the peak-found signal, and the 5 ns digital latency from the comparator firing to the leading edge of the ART. The 6-bit address with an accompanying flag are transmitted serially, at 320 Mbps data rate, on every clock cycle of the BC clock.
- Direct Data Outputs (DDO): The VMM provides fast information via its 64 direct timing outputs, one for each of its channels. Each channel implements a fast digitization 6-bit ADC, that provides a coarse, but rapid amplitude measurement of the pulse. The conversion starts immediately when the peak-found signal is issued and the data are streamed at each output, synchronized with an externally-provided 160 MHz clock. The DDOs can also operate in four other modes: time-over-threshold (ToT), threshold-to-peak (TtP), peak-to-threshold (PtT), or a 10 ns pulse occurring at peak (PtP).

<sup>3</sup> Maximum size of eight clock cycles.

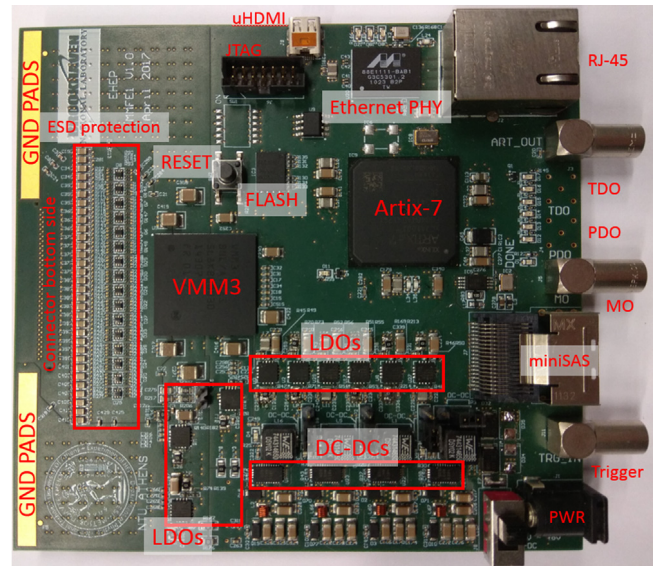


Fig. 3. The MMFE1 board.

### 3. The MMFE1 board

The MMFE1 board [6] houses a single VMM and an Artix<sup>®</sup>-7 FPGA by Xilinx<sup>®</sup>. It is shown in Fig. 3. Communication with the software host (see Section 5) is established using the FPGA and a commercial network Ethernet physical interface (PHY) at speeds of 10 or 100 or 1000 Mbps. The power distribution circuit utilizes step-down regulators with an input voltage range from 3.4 V up to 42 V and an output ripple of less than 10 mV peak-to-peak. The four power rails of the VMM (Vddp, Vdd, Vddad and Vddd) are independently powered by different Low Drop-Out (LDO) regulators to mitigate any ripple effects. Three LEMO connectors provide access to the monitoring, peak-detector and time-detector outputs (MO, PDO, TDO respectively) of the on-board VMM. The board has the ability to output the ART signal and receive an external trigger signal.

The architecture of the MMFE1 board is shown in Fig. 4. The FPGA is accessible through a standard JTAG connector for programming and monitoring. The configuration file is stored in a flash memory which is accessible over the Serial Peripheral Interface (SPI) protocol. A second memory (EEPROM) is used to store the Media Access Control (MAC) and network addresses of the board, each of which are configurable. The EEPROM is accessed by the FPGA via the I<sup>2</sup>C protocol, thus allowing the on-board FPGA to dynamically reconfigure its network address, a crucial aspect for the scalability of the system. A  $\mu$ HDMI connector provides access to additional external signals such as a reference clock, trigger and reset. A miniSAS<sup>4</sup> connector can be used to interface externally with other boards, such as the Level-1 Data Driver Card (L1DDC) [6], that has been developed for the NSW upgrade [1].

The input protection of the VMM is implemented using steering diodes in a rail-to-rail configuration along with series resistors. A number of front-end boards based on the same architecture as the one described here have been fabricated to interface with various types of detectors. The MMFE1 board, for example, was fabricated to validate the functionality of the VMM ASIC using  $10 \times 10$  cm<sup>2</sup> Micromegas [7] prototype chambers.

### 4. Firmware implementation

The on-board FPGA hosts the front-end VRS firmware, a design capable of reading-out, configuring and calibrating the ASIC. It interfaces to the on-board VMM(s) and the software (see Section 5). Written

<sup>4</sup> Commercial mini Serial Attached Small Computer System Interface.

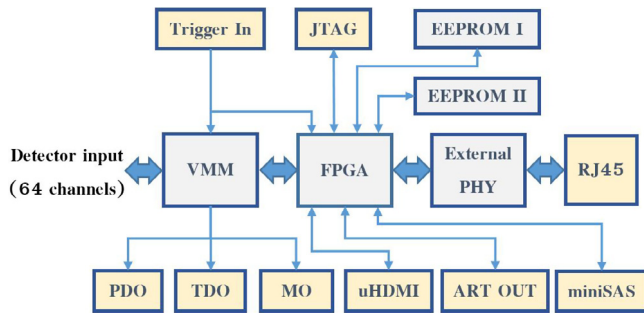


Fig. 4. Block diagram illustrating the architecture of the MMFE1 front-end board. Blocks with yellow color indicate physical connectors while those in gray color indicate on-board components. The VMM is connected to the FPGA via differential lines, while its monitoring outputs drive three LEMO connectors. The trigger signal is propagated to the FPGA and to one of the VMM channels, so that the ASIC can perform precise timing measurements to the trigger signal itself. The FPGA can be accessed via JTAG, while two flash memories store the FPGA’s configuration and network access information. Ethernet connection is achieved via a standard RJ45 and External PHY pair that the FPGA can interface with. Finally, the FPGA can interface with off-board devices via a  $\mu$ HDMI and a miniSAS connector. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

primarily in VHDL, the design can be implemented on several types of front-end boards that house one or multiple VMMs, such as the MMFE1 (see Section 3). The firmware can also be easily ported to different FPGA packages and pinouts. As the scheme may be used either in standalone bench measurements, or in multiple-board applications, scalability and flexibility were two additional factors important in the development process. The main modules of the design are shown in Fig. 6 and are described below.

#### The flow FSM

This component is the supervisory state machine of the FPGA’s logic. It interfaces with all other building blocks of the design and defines the overall state in which the system is, depending on the user’s directive as it is forwarded by the Configuration Block.

#### UDP/ICMP block

Based on the 1 Gbps UDP/IP Stack design from OpenCores [8], this sub-module provides interfacing capabilities with the software host. Modifications have been made to the original design to accommodate some Internet Control Message Protocol (ICMP) functionalities. The block also deploys the Xilinx® GTP Transceiver IP core that interfaces with the on-board PHY chip to provide 1 Gbps Ethernet-over-copper connection with the software host.

#### Configuration block

This module registers the incoming UDP payload data from the software and applies the values in internal configuration registers accordingly. It also forwards any commands for switching the system’s state to the Flow FSM. Furthermore, it receives the VMM configuration data, and transmits it to the ASIC via the SPI protocol.

#### CKTP/CKBC generator

This module generates the BC clock (CKBC) and the test-pulse signal (CKTP). The CKBC is used as the VMM’s reference clock, while the CKTP is driving the VMM internal test pulser. The test-pulse strobe is generated by the FPGA synchronously to the CKBC. The reference clock source may be an on-board oscillator or an external common clock source. The latter is used in multiple-board readout schemes to ensure synchronization between independent boards. This module is also responsible for the timing calibration of the ASIC.

In order to calibrate its timing response, the VMM is configured in the internal pulser mode. The FPGA skews the CKTP with respect to the CKBC where the TAC is latched (see Section 2). This is implemented

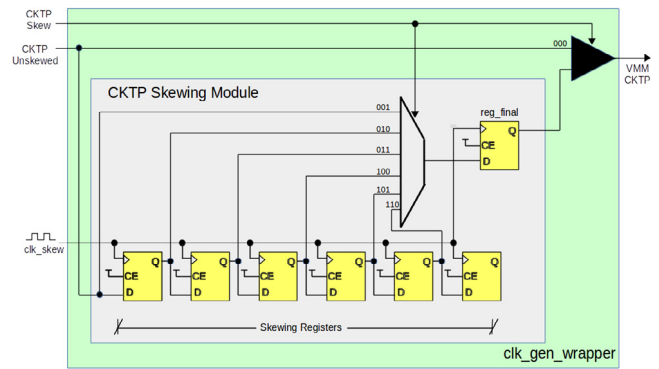


Fig. 5. Behavioral representation of the CKTP skewing design. Only a limited number of the actual twenty-four registers and their associated multiplexer inputs is depicted here. Since the multiplexer is mainly implemented in FPGA LUTs that may be rearranged from implementation to implementation, an extra, physically constrained, register level before the global buffer that fans-out the CKTP ensures reliability and repeatability of the skewing scheme.

in the CKTP/CKBC Generator module which can skew the two signals with a granularity of 1 ns. This is achieved by driving the pulser clock into a shift-register clocked by a 500 MHz clock, comprised of twenty-four registers/delay stages. Each output of the register chain drives a multiplexer that depending on the skew setting, selects the corresponding delay stage. Each register delays the pulser clock by 2 ns, but their total number makes it possible to skew the signal within two periods of the BC clock at a final step size of 1 ns. The data from the calibration procedure are captured by the software. In a later analysis the conversion factor of ADC counts to nanoseconds is derived for each individual channel. The circuit architecture is shown in Fig. 5.

#### Readout wrapper

This part of the logic deploys different sub-blocks (readout cores) that extract data from the VMM. Supporting all possible readout modes of the ASIC, it receives a trigger signal that initiates a readout cycle from the Packet Formation module and forwards this trigger to the VMM accordingly. Depending on the board implementation, the wrapper can instantiate more than one readout cores that can extract data from multiple on-board VMMs.

#### Packet formation

This module lies one hierarchical level above the Readout Wrapper. It forwards trigger signals to it, as received by the Trigger Module. It implements an internal trigger counter that the software uses to group events from multiple boards, when applicable. The module also aggregates the VMM data from the readout core(s). After accumulating all VMM data, it builds the event packet and finally forwards the data to the software via a buffer that interfaces with the UDP/ICMP Block.

#### Trigger module

This is the module that receives a trigger signal from either an external source or from the CKBC/CKTP Generator.<sup>5</sup>

#### XADC module

This module implements the Xilinx® IP core of the same name. It essentially consists of a wrapper of the integrated 12-bit ADC that is driven by the VMM analog outputs. The only part of the logic written in Verilog HDL, the XADC is used when calibrating the threshold and pulser DACs of the VMM, or to sample the voltage variation of the VMM’s channel inputs, a measurement indicative of the noise each

<sup>5</sup> The CKTP is also used as a trigger signal in the case of internal-generated-trigger mode, primarily applicable when acquiring data using the VMM’s test pulser.

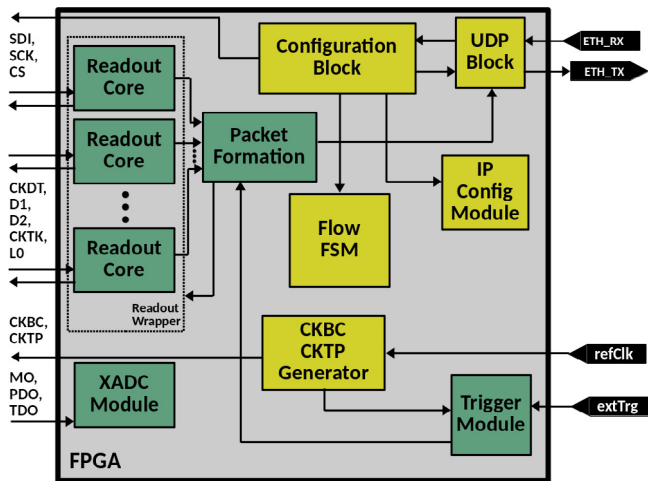


Fig. 6. Architecture of the VMM readout firmware. On the left-hand side the FPGA interfaces with the VMM (see also Fig. 1 for a reference to the VMM pin names), while on the right-hand side the FPGA interfaces with the Ethernet PHY chip and any trigger or reference clock input port. For simplicity, the connectivity of the Flow FSM module with all other sub-modules is not depicted here.

channel is subjected to. Finally, the module is responsible of digitizing the PDO and TDO voltage levels when acquiring data from the VMM in the Two-Phase Analog mode. Whichever the case may be, the samples are forwarded upstream to the software which handles them as typical data.

#### IP configuration modules

When using multiple readout boards within the same network domain, the ability to easily modify each FPGA's host IP and MAC address becomes a necessity. The desired IP address, which is configured by the software, is registered via the SPI or the I<sup>2</sup>C protocol. Upon a power start, the same module extracts the previously written address from the memory and stores the local IPv4 and MAC addresses.

#### 4.1. Readout modules

Due to the importance of the VMM readout procedure carried on by the FPGA, all logic blocks that implement the various data extraction schemes are described in more detail in this section.

All three VMM readout modes, described in Section 2, are adequately supported by the firmware. Three distinct readout components are contained within the Readout Wrapper (see Fig. 6) and depending on the desired data-extraction scheme (readout mode), the corresponding component is instantiated. In cases where the FPGA has to read-out more than a single VMM, the firmware offers the possibility to deploy multiple instances of a given readout component, to extract VMM data in parallel.

In the case of the Two-Phase Analog mode (see Section 2), the pulse and timing amplitudes are sampled by the FPGA's XADC, while the design's readout core deserializes the channel's address and advances the token to the next available channel. This readout scheme relies on the use of the external 12-bit ADC in order to take advantage of its higher resolution, compared to that provided by the VMM's faster internal ADCs. For this reason, the achievable readout rate is generally lower compared to the other two schemes.

The component handling the continuous readout of the VMM (described in Section 2) has two versions. The first one implements a readout operation in which the BC clock is used continuously. This mode is optimal, for example, in cases where the trigger and the VMM events have a precisely specified timing relationship with the reference clock. This can be achieved by the general experimental conditions

(e.g. in the LHC), or by using a global synchronization logic (implemented in the VSB and described below). The second version, issues the reference clock as a strobe, asserting it only upon the reception of a trigger signal. The FPGA issues a configurable amount of clock strobes to the VMM, after a fixed latency with respect to the initiating trigger signal. The VMM channels which their TAC is not saturated within the window defined by the fixed latency are digitized by the internal ADCs and eventually read-out by the FPGA. This mode can be used in an asynchronous operation, provided that the original trigger latency is not too large as to saturate the TACs of the channels. This readout scheme is illustrated in Fig. 7.

The readout component in the firmware, designed to handle the L0 readout mode of the VMM (see Section 2) emulates all necessary signals that are foreseen to be issued by the Trigger, Timing and Control (TTC) system [9] of the ATLAS experiment and reads out the data stream of the ASIC. The design's L0 readout core deploys logic that aligns to and decodes the 8b/10b-encoded stream from the VMM, that is synchronous to the 160 MHz data clock provided by the FPGA. In this readout mode, the firmware emphasizes on minimizing the dead-time associated with the reception of a trigger. Fig. 8 shows the readout efficiency of the system as a function of the trigger rate.

Apart from the three distinct readout modes that the firmware supports, the design's logic can also read-out the ART data, a useful feature to study the ASIC's trigger primitives, if the ART output of the VMM is routed directly to the on-board FPGA. This is the case for the MMFE1 board, as described in Section 3. The ART readout module deserializes the 6-bit ART address with its flag, while it also monitors the state of the FPGA logic's Trigger Module. The module's architecture implements counters that measure the time difference between the reception of an ART address and an external trigger, with 6.25 ns granularity. The address and time-difference information are prepended to the data packet that is transmitted to the software after each trigger for further analysis.

#### 4.2. VRS supervisory board

The VRS Supervisory Board (VSB) is another FPGA-based system that interfaces with other VRS front-end nodes. Its firmware has been implemented successfully in a Xilinx<sup>®</sup> VC709 evaluation board. Connectivity is achieved through a custom FPGA Mezzanine Board (FMC). The design is flexible enough to be ported to any board which is able to connect to several VRS front-end nodes. It can be used to maximize the efficiency in setups comprising front-end boards that are subject to high trigger rates — in a test-beam implementation for instance. It provides a common clock source to the entire system and can receive an external trigger signal, usually originating from a scintillator coincidence, and forwards it to each front-end in a synchronous manner. Upon receiving a trigger, a front-end node issues a busy signal to the VSB. This signal is released once the VMM readout is complete. The VSB can only issue another trigger once all nodes are no longer busy, therefore readout synchronization of the front-end boards is guaranteed.

A useful feature of the VSB firmware is that it can perform a precise (time-wise) trigger selection. In an asynchronous trigger operation where the VMMs are running in synchronous mode (e.g. in L0 mode or in continuous mode with a free-running BC clock), the VSB trigger selection logic distributes an external trigger only if it is edge-aligned with the global reference clock, with an accuracy of ~1.5 ns. This functionality offers the possibility to emulate a fully synchronous trigger system, thus facilitating precise timing measurements.

Another feature of the VSB is that it implements an ART data aggregating and processing module, capable of evaluating the VMM trigger primitives from three different front-ends. The logic can deserialize the ART stream from the front-end nodes and the data are combined if there is an ART event from each source. Eventually, the ART address combinations are checked against a look-up-table implemented in the FPGA's blockRAM primitives. The table yields information on all possible valid ART address combinations that represent an actual event.

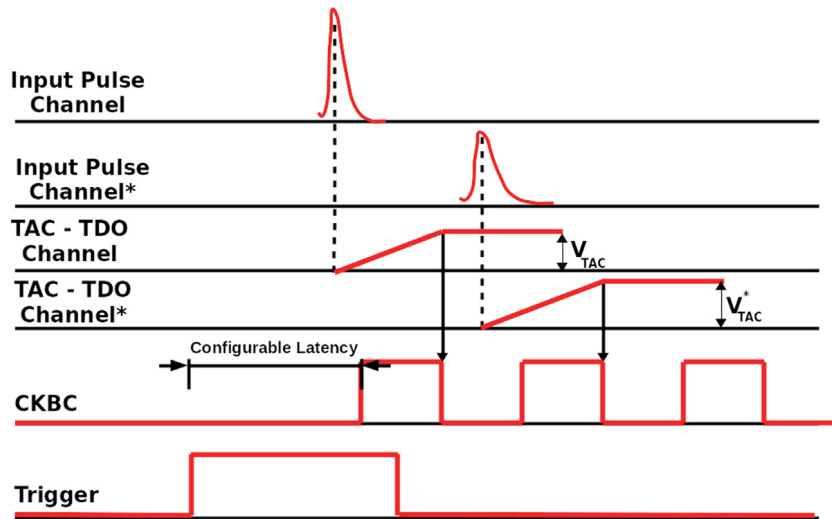


Fig. 7. Strobing the CKBC after a fixed latency with respect to an input trigger to precisely define the timing of hits.

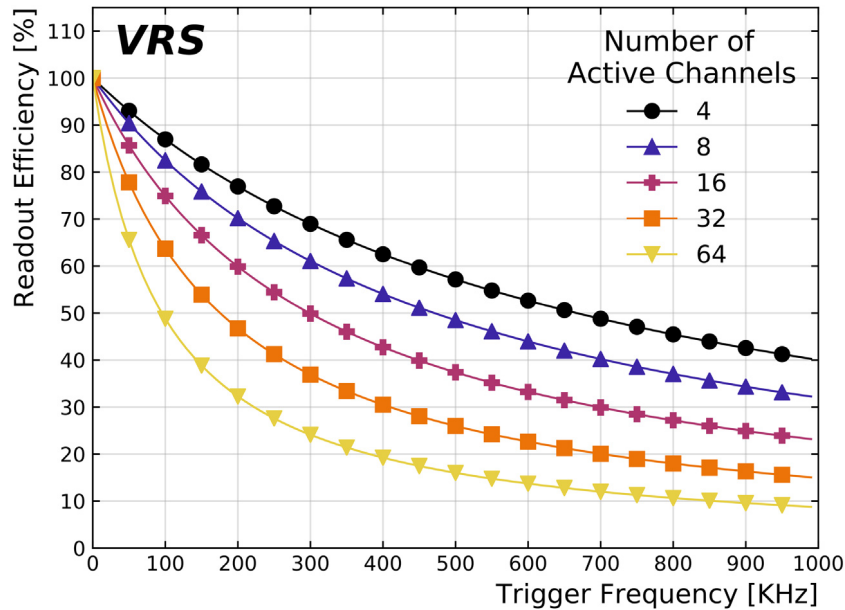


Fig. 8. Readout efficiency of the VRS readout system as a function of the trigger rate, for various channel occupancies, when running in L0 readout mode.

These combinations may be loaded by the user into the look-up-table via UDP upon startup. The logic makes a trigger decision at a fixed latency of 650 ns, and may produce a trigger signal that can be driven to the front-end nodes in order to extract data from the VMMs, thus resulting in a self-triggered system (see Fig. 9).

### 5. The VMM ethernet readout software

User control of the front-end electronics is provided by a high-level software tool referred to as 'VERSO': VMM Ethernet Readout Software. VERSO is developed entirely in C++ and uses the Qt framework [10] for developing the graphical user interface (GUI) which is shown in Fig. 10. VERSO has two main responsibilities: orchestrating configuration processes and data-acquisition (DAQ). Both functionalities are performed using a custom-made protocol implemented within the UDP standard. VERSO also implements a suite of automated functionalities for performing calibration of the front-end electronics or for data-taking with a fixed number of triggers. A block diagram illustrating the configuration, DAQ, and calibration functionalities of VERSO is shown in Fig. 11.

The configuration of the front-end electronics is performed using register-value pairs, with the software communicating the values of the firmware operating parameters (e.g. the readout mode, clock frequencies, trigger signal source, relative clock phases, etc...) set by the user to registers on the FPGA(s). The full VMM configuration is also sent in this manner, with registers assigned for receiving the SPI data for each VMM on the front-end. Upon configuration of each register, the firmware transmits a confirmation UDP packet with a unique transaction ID to VERSO to acknowledge successful receipt of each command sent from the software. This ensures that the requested action took place on the hardware and that the UDP command packet issued from VERSO was not lost. If a successful acknowledgment is not received, VERSO logs this and can re-attempt transmission of the lost command.

The DAQ functionality provided by VERSO can handle the readout of several front-end boards. The event building is based on a trigger counter that is transmitted with each data fragment upon the reception of a trigger signal at the front-end. An 'event', therefore, is the collection of such data fragments that have the same trigger number. The trigger signal can either be an internal one configured by VERSO and

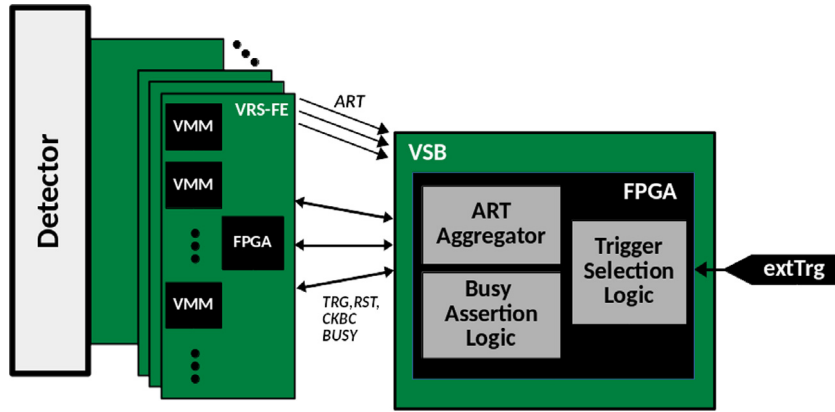


Fig. 9. Block diagram of the VRS Supervisory Board depicting its interfaces with the front-end boards and any external trigger input. The logic blocks of the VSB firmware include the logic that inhibits input trigger signals unless they coincide with the positive edge of the system clock, the ART aggregating module that processes the incoming ART stream and may issue a L0 trigger signal accordingly, and finally the logic that keeps the front-ends synchronized, which would otherwise be agnostic of the readout state of other nodes.

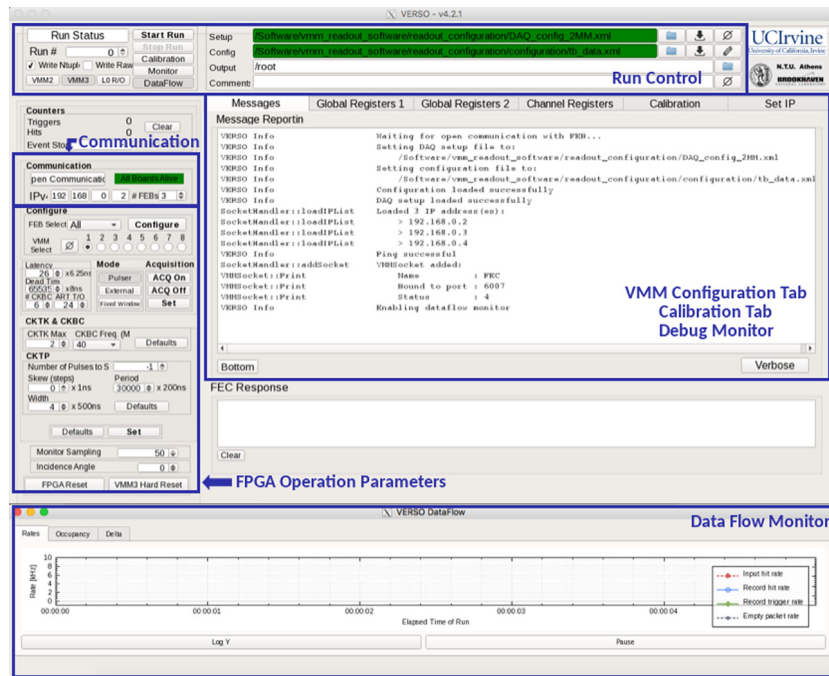


Fig. 10. VERSO user interface.

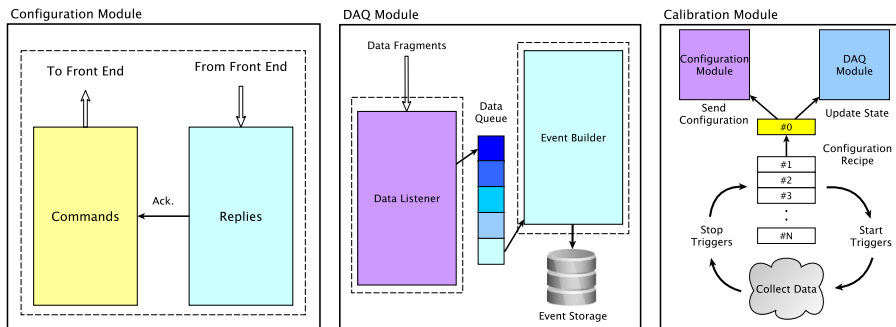


Fig. 11. Diagrams illustrating the logical, high-level separation of the various modules contained within VERSO. Dashed lines indicate independent threads of execution. Unfilled arrows indicate communication with the FPGA via network. *Left*: Configuration module responsible for sending control and configuration commands to the FPGA housed on the front-end electronics. Each command sent expects an acknowledgment reply from the FPGA. *Middle*: DAQ module responsible for processing incoming data fragments from the front-end electronics, buffering them, and building full event fragments to finally store on disk. *Right*: Calibration module responsible for running various front-end calibration routines by processing “configuration recipes” to control instances of the Calibration and DAQ modules.

generated by the firmware (e.g. the test pulse clock, CKTP, described in Section 4) or an external signal.

As the UDP protocol cannot ensure a fixed latency, events based on the same trigger number are not guaranteed to be captured by VERSO within the same UDP packet. The event building of VERSO reflects this asynchronous readout by employing a single-producer/single-consumer threading model wherein a reading thread is tasked with the reception of the data-fragments from the front-end(s) and a second processing thread is tasked with processing all those data-fragments associated with each unique trigger number received. The data fragments received by the reading thread are indexed by their associated trigger-number and are buffered in a queue for eventual removal by the processing thread. The queue is implemented as a lock-free concurrent queue [11] which ensures that the two threads never prevent access to the queue for adding or removing data-fragments. This makes for fully-efficient readout even at the maximal data-taking rates observed during test-beam scenarios. Bench measurements have also shown VERSO to be 100% efficient (i.e. no lost or missed data fragments) in reading out data from a single front-end having all channels triggered internally at the maximum allowed rate (Fig. 8). The processing of an event consists of the decoding of the raw data fragments transmitted by the front-end(s) and building high-level data structures representing each event. These structures are continuously stored on disk throughout a given data-taking period in the ROOT : : TTree format to be used for offline data analysis.

During the calibration procedure, the internal triggering and channel pulsing capabilities of the VMM are used. From the VERSO GUI, the user specifies a set of configuration parameters to step through (e.g. VMM channel pulser amplitudes), collecting a fixed number of internally triggered and pulsed events per step per channel, VMM, and front-end board. The procedure consists of these configuration steps being loaded into memory as a complete ‘configuration recipe’ and being simultaneously passed to running instances of the configuration and DAQ modules (see Fig. 11). Upon receipt of a new step in the configuration recipe, the configuration module terminates the front-end data acquisition, sends the new front-end configuration, and re-enables front-end data acquisition which then begins the transmission of data fragments to VERSO. At each configuration step, the DAQ module is provided a condensed representation of the front-end configuration. This is subsequently stored on disk alongside the event data described above, so that the events from a given calibration data-taking instance can be associated with the active parameters at the time of recording. This additional information is necessary for performing the calibration of the front-end electronics.

## 6. Conclusions

A compact readout system for the VMM ASIC has been developed. VRS was originally prepared for the validation of the ASIC for the ATLAS NSW upgrade, but the developments have gone further, establishing a flexible, robust, and performant system for the readout of chambers. The hardware built is versatile enough to interface with different detectors, while the firmware supports all available VMM readout modes. Moreover, the software allows for efficient data-taking and has embedded calibration procedures. It can operate the ASIC in few simple steps and provides event building from multiple sources. The VRS system provides synchronization and self-triggering capabilities across multiple boards. The system has been validated with VMM internal pulsing but also in several test-beam applications. The system is used successfully by multiple groups as a validation system for the use of VMM in different applications.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## CRediT authorship contribution statement

**T. Alexopoulos:** Conceptualization, Investigation, Writing - review & editing, Supervision, Project administration, Funding acquisition. **D. Antrim:** Conceptualization, Software, Validation, Investigation, Writing - original draft, Writing - review & editing, Visualization. **C. Bakalis:** Conceptualization, Software, Validation, Investigation, Formal analysis, Writing - original draft, Writing - review & editing, Visualization. **G. De Geronimo:** Conceptualization, Methodology. **P. Gkoutoumis:** Conceptualization, Software, Validation, Investigation, Writing - original draft, Writing - review & editing, Visualization. **G. Iakovidis:** Conceptualization, Software, Validation, Formal analysis, Investigation, Supervision, Writing - original draft, Writing - review & editing, Visualization. **P. Moschovakos:** Conceptualization, Software, Validation, Investigation, Writing - original draft, Writing - review & editing, Visualization. **V. Polychronakos:** Conceptualization, Writing - review & editing, Supervision, Project administration, Funding acquisition. **A. Taffard:** Writing - review & editing, Supervision, Funding acquisition.

## Acknowledgments

This work was funded in part by the U. S. Department of Energy, Office of Science, High Energy Physics under Contracts DE-SC0012704, DE-SC0009920.

This present work was partially co-funded by Greece and the European Union through the Operational Program ‘‘Competitiveness Entrepreneurship Innovation’’ of the Framework EPAnEK 2014–2020, ‘‘Detector Development and Technologies for High Energy Physics (DETAnet)’’.

The authors would like to acknowledge the contribution to the project from other researchers of the collaboration, such as L. Kurilenko (U. of Seattle), W. Lu (BNL), S. Martoiu (U. of Bucharest) and R. Pinkham (U. of Michigan).

## References

- [1] CERN, *New Small Wheel Technical Design Report, no. CERN-LHCC-2013-006. ATLAS-TDR-020*, CERN, Geneva, 2013.
- [2] G. De Geronimo, J. Fried, S. Li, J. Metcalfe, N. Nambiar, E. Vernon, V. Polychronakos, VMM1 - An ASIC for micropattern detectors, *IEEE Trans. Nucl. Sci.* 60 (3) (2013) 2314–2321, <http://dx.doi.org/10.1109/TNS.2013.2258683>.
- [3] G. De Geronimo, S. Li, Shaper design in CMOS for high dynamic range, *IEEE Trans. Nucl. Sci.* 58 (5) (2011) 2382–2390, <http://dx.doi.org/10.1109/TNS.2011.2162963>.
- [4] G. De Geronimo, J. Fried, G. Smith, B. Yu, E. Vernon, C. Britton, W. Bryan, L. Clonts, S. Frank, ASIC for small angle neutron scattering experiments at the SNS, *IEEE Trans. Nucl. Sci.* 54 (3) (2007) 541–548, <http://dx.doi.org/10.1109/TNS.2007.893718>.
- [5] A.X. Widmer, P. Franaszek, ADC-balanced, partitioned-block, 8B/10B transmission code, *IEEE Trans. Nucl. Sci.* 27 (5) (1983) 440–451, <http://dx.doi.org/10.1147/rd.275.0440>.
- [6] P. Gkoutoumis, *Design and Development of the Level-1 Data Driver Card (L1DDC) for the New Small Wheel upgrade of the ATLAS experiment at CERN, no. CERN-THESIS-2019-039*, National Technical University of Athens, 2019.
- [7] T. Alexopoulos, J. Burnens, R. de Oliveira, G. Glonti, O. Pizzirusso, V. Polychronakos, G. Sekhniaidze, G. Tsiolitis, J. Wotschack, A spark-resistant bulk-micromegas chamber for high-rate applications, *Nucl. Instrum. Methods Phys. Res. A* 640 (1) (2011) 110–118, <http://dx.doi.org/10.1016/j.nima.2011.03.025>.
- [8] Opencores website. URL <https://opencores.org/>.
- [9] S. Ask, D. Berge, P. Borrego-Amaral, D. Caracina, N. Ellis, P. Farthouat, P. Gälln , S. Haas, J. Haller, P. Klofver, A. Krasznahorkay, A. Messina, C. Ohm, T. Pauly, M. Perantoni, H.P.L. Junior, G. Schuler, D. Sherman, R. Spiwojs, T. Wengler, J.M. de Seixas, R.T. Teixeira, The ATLAS central level-1 trigger logic and TTC system, *J. Instrumentation* 3 (08) (2008) P08002, <http://dx.doi.org/10.1088/1748-0221/3/08/p08002>.
- [10] Qt. URL <https://www.qt.io/>.
- [11] A Single-Producer, Single-Consumer Lock-Free Queue for C++. URL <https://github.com/cameron314/readerwriterqueue>.