PREPARED FOR SUBMISSION TO JINST

CHEF2019 - CALORIMETER FOR THE HIGH ENERGY FRONTIER 2019

25-19 NOVEMBER 2019

KYUSHU UNIVERSITY, FUKUOKA, JAPAN

LHC-ATLAS Phase-1 upgrade: Firmware validation for real time digital processing for new trigger readout system of the Liquid Argon calorimeter

R. Oishi^a on behalf of ATLAS Liquid Argon Group

^a*Department of Physics, Graduate of School of Science, The University of Tokyo,
Hongo Bunkyo yard Tokyo, Japan*

E-mail: reiyo.oishi@cern.ch

ABSTRACT: The third run (Run 3) of the ATLAS experiment at CERN's Large Hadron Collider will begin in 2021 in a high luminosity and high energy environment. In this circumstance, an effective trigger system is required. ATLAS uses a two-stage trigger system. The first stage trigger system (Level 1 trigger) is based on hardware, and the second stage is a software-based high-level trigger on a PC farm. With this system, an overall trigger rate is required to be suppressed to 1 kHz or less from a bunch-crossing rate of 40 MHz. In the liquid argon calorimeter level 1 trigger system, new trigger readouts with a factor of ten higher granularity compared to the Run 2 trigger readout, are introduced to suppress backgrounds. By measuring shower shape with the higher granularity, the electron and photon trigger rate can be suppressed sufficiently. To process all data from the supercells, new readout boards are prepared. One of the boards, dubbed LATOME, converts ADC data to transverse energy with fixed latency and sends three types of energies to the Level 1 trigger system. The trigger system uses computed energies with LATOME firmware, and therefore the development and validation of the LATOME firmware is important for the ATLAS data acquisition system. One of the LATOME firmware module, *User Code*, calculates the transverse energy for all supercells. To ensure all *User Code* blocks are functional, firmware validation is required. These firmware validation schemes with simulation and with the board itself are presented with the LATOME firmware design.

KEYWORDS: Calorimeters, Trigger algorithm, Trigger concepts and systems (hardware and software)

*Copyright 2020 CERN for the benefit of the ATLAS Collaboration.
Reproduction of this article or parts of it is allowed as specified in the CC-BY-4.0 license.*

Contents

1	Introduction	1
2	LATOME firmware	2
3	<i>User Code</i>	2
4	Simulation verification	4
5	Onboard validation	5
6	Conclusion	6

1 Introduction

Run 3 of the LHC [1] will be started in 2021. The planned instantaneous luminosity is up to $2.0 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ at $\sqrt{s} = 14 \text{ TeV}$ with proton-proton collisions with a bunch spacing of 25 ns. The ATLAS experiment [2] at the LHC uses a two-staged trigger system. The first trigger system (Level 1 trigger) uses only limited detector information. Every proton-collision event is processed at the level 1 trigger system with a fixed latency of $2.5 \mu\text{s}$. Then the trigger decision in the second system (the high-level trigger) is performed on a PC farm only for events passing the level 1 trigger. The full granularity of the detector information is used in the high-level trigger system. With the two-staged trigger systems, the trigger rate is required to be suppressed to 1 kHz from the bunch crossing rate of 40 MHz. To cope with the trigger processes in the high luminosity and high energy environment during Run 3, trigger performance needs to be improved. In the liquid argon (LAr) calorimeter [3] trigger system, a new trigger readout system will introduce “supercells” which will allow a readout granularity ten times higher than the Run 2 trigger readout. This new system uses a high-performance advanced mezzanine card, so-called LATOME (LAr Trigger prOcessing MEzzanine), which has an Intel Corporation Arria-10 FPGA. The LATOME firmware receives the ADC data digitized by the on-detector digitizer board and transmits three types of energies to the next stage of the level 1 trigger system. One of the modules in the LATOME firmware, *User Code*, uses digital filtering to calculate transverse energy and extracts the arrival time of the energy deposition in each supercell. Blocks inside the *User Code* are categorized as main-path and sub-path blocks. All blocks have unique functions. Main-path blocks need to work with fixed latency and send output to the next block or a module with reduced timing constraints. Moreover, simple monitoring systems are implemented into sub-path blocks to support the LATOME firmware monitoring process. Simulation schemes of the *User Code* verification have been established with Universal VHDL Verification Methodology (UVVM) [4], which can set up an automatic verification procedure. On-board validation took place at the test-bench of CERN and is reported in this proceedings.

2 LATOME firmware

Signals from the LAr calorimeter are digitized with a sampling rate of 40 MHz by the new LAr Trigger Digitizer Board (LTDB), installed in the front-end crate. The digitized data is transmitted via 70-100 m long optical fibers to LAr digital processing board (LDPB) installed in the back-end electronics. Each LDPB is composed of one carrier board and four LATOME cards embedded with Arria 10 GX FPGA. Each LATOME receives ADC data from the LTDB with 40 fibers at 204.8 Gbps (5.12 Gbps/fiber) and transmits three types of energies with 40 fibers at 448 Gbps (11.2 Gbps/fiber) to the next stage of the level 1 trigger system. Up to 320 supercells will be processed with one LATOME firmware, thus each fiber delivers data from eight supercells at 320 MHz. This supercell information is arranged in the LATOME firmware to decrease a clock frequency to 240 MHz and manage saturated pulses properly. A remapping algorithm is configurable and depends on the detector position. After arranging supercell mapping, transverse energies and timing of the energy depositions are computed in each supercell by using two filtering algorithms. Computed transverse energies are summed up over three types of $\eta - \phi$ regions and they are then transmitted to the next system. LATOME firmware has also an IP bus controller and a monitoring module. The IP bus changes LATOME firmware configurations and implements some simple monitoring systems. These simple monitoring systems can be read or written via IP bus at up to 1 Gbps. The monitoring module extracts processed data over a XAUI (10 Giga ethernet) card and sends it to a dedicated PC through the carrier board and ATCA networks. LATOME firmware modules are summarized in Table 1. They relate to each other as shown in Figure 1(a)

Table 1: LATOME firmware modules

Category	Module	Description
Main-path	Input stage	Extract ADC data
	Configurable remap	Decrease clock frequency and change supercell mapping
	<i>User Code</i>	Calculate transverse energy and extract the timing of energy deposition
	Output summing	Sum up transverse energy over specified regions
Sub-Path	IP bus controller	Read / write data directory and change LATOME firmware configuration
	Monitoring	Read monitoring data at 10 Gbps

3 User Code

One of the LATOME firmware modules, *User Code*, computes transverse energies and extracts the timing of energy depositions with two digital filtering algorithms for every supercell. The first filtering algorithm, an optimal filter [5] which can reduce noise effect, is used to compute transverse energy (E_T) and the products of transverse energy and a pulse phase ($E_T\tau$) from ADC data as follows.

$$E_{Tj} = \sum_{i=0}^3 a_i \left(ADC_j^i - Pedestal_j - Baseline_j^i \right) \quad (3.1)$$

$$E_T\tau_j = \sum_{i=0}^3 b_i \left(ADC_j^i - Pedestal_j - Baseline_j^i \right) \quad (3.2)$$

Index i, j indicates the timing and supercell id, respectively. Pedestal and baseline shift needs to be subtracted from ADC data as shown in Equation 3.1 and 3.2. Filtering coefficients and pedestal values are obtained in an earlier stage. They depend only on supercell ID. Due to the LHC proton filling schemes, baseline shifts affect the energy calculation. They are calculated as follows.

$$Baseline_j^i = \frac{\sum_{k=0}^{2^{10}-1} ADC_j^{i+k}}{2^{10}} - Pedestal_j \quad (3.3)$$

The baseline shift is computed as an average of ADC data and pedestal would be subtracted from it to decrease bit width of baseline shift as shown in Equation 3.3. The second filtering algorithm extracts the timing of the energy deposition with the relation between E_T and $E_T\tau$. All of the *User Code* blocks are driven with 240 MHz and every filtering process must be executed with exactly 27 clock ticks (=112.5 ns). 22 clock ticks are used for data from multiple ADCs and the coefficients and reset of the *User Code* latency are used to extract the timing of energy depositions. The *User Code* also has simple monitoring blocks. They store useful information for *User Code* monitoring like a pulse peak and the sum of ADC values for one LHC cycle. All of the *User Code* blocks are summarized in Table 2. They relate to each other as shown in Figure 1(b).

Table 2: *User Code* blocks

Category	Block	Description
Main-Path	BAS	Cancel out baseline shift
	FIR, SAT	Calculate transverse energy with optimal filtering
	SEL	Extract timing of energy deposition
	COM	Treatment for saturation pulses
Sub-Path	SUM	Calculate the sum of ADC data for 1 LHC cycle
	ADC, PEAK	Detect a pulse peak by using ADC shape and energy

The *User Code* has configuration signals described in Table 3. Some blocks are completely disabled with these configurations. They are controlled with the IP bus without recompiling firmware. The development of the *User Code* firmware has been completed and the final version of the *User Code* was released. The LATOME firmware occupies about 90 % of the FPGA logic and memory resources and thus the implemented functions are verified individually. Currently, the *User Code* firmware is under detailed validation.

Table 3: Global Control Signals

global control signal	Description
<i>hardpoint</i>	Change the resolution of transverse energy.
<i>baseline</i>	Baseline cancelation procedure implementation enable.
<i>selection</i>	Extraction of energy deposition enable.
<i>saturation</i>	Saturation treatment enable.
<i>combine</i>	Neighbor supercell treatment enable.

Figure 1: (a) LATOME firmware structure. (b) *User Code* structure. *User Code* sends ADC data / Pedestal subtracted ADC data / Transverse energy / Transverse energy with saturation treatment as monitoring data.

4 Simulation verification

To confirm that the *User Code* logic is functional, simulation schemes have been developed. The verification of the *User Code* logic is performed by comparing the outputs from the *User Code* with those from a corresponding *User Code* model, which is independently prepared using other software scripts. The *User Code* sends four types of data to the monitoring module and one type of data, processed with the *User Code* main-path blocks, to the Output summing (Osum) module. Simulation schemes are constructed with Universal VHDL Verification Methodology (UVVM) [4]. UVVM is an open-source methodology and library for making very structured VHDL-based testbenches efficiently and automatically. Input data to these simulation schemes are generated by the AREUS [6] software. AREUS has been developed for detailed trigger simulation studies of the full readout chain of the LAr calorimeter. By using the AREUS data, *User Code* logic can be verified in conditions similar to the Run 3 experiment. Simulation verification procedures are described below.

1. Specifying an AREUS data and the *User Code* model emulating bit operations of the corresponding *User Code* block.
2. The *User Code* model generates ROM initialization files and initializes ROM in the checkers.
3. The *User Code* processes the AREUS data and outputs five kinds of data to the checkers.
4. The checkers compare firmware outputs with values predicted by the *User Code* model.

The above processes are used to verify all the logic and latency of the *User Code*'s main path. For sub-path blocks, the IP bus reads the register contents and the checker compares with the output from the *User Code* model. All the *User Code* blocks have been verified using several AREUS datasets.

5 Onboard validation

From the simulation verification described in Section 4, it was confirmed that the *User Code* logic, with given latencies, functions well. However, simulation processes do not take into account stability factors, such as the propagation delay time, of the *User Code*. Resource occupancy of the LATOME firmware is tight and therefore it is important to test the *User Code* stability with an actual LATOME board under the same conditions as the Run 3 experiment. On-board validation took place at CERN with two LATOME cards. One LATOME is used as an injector and sends ADC data in the LTDB format. The up to *User Code* firmware (removing the Osum module) is uploaded in the other LATOME. Due to the disabled Osum module, only monitoring data is available. To validate each *User Code* block, it is needed to check if outputs to the monitoring module from the *User Code* are consistent with those from the corresponding *User Code* model. This is the same procedure as the simulation verification. Monitoring data is recorded in binary after the level 1 accepts are generated. Only one out of four types of monitoring data can be recorded at a time. As described in Table 3, *User Code* depends on the configuration, so that each block is validated with proper configurations and monitoring data types. The *User Code* model needs to be configured with the same configuration signals. They are extracted with the IP bus on-line (Figure 2(a)). Sub-path block validation also took place with a similar system. Computed data can be read via the IP bus and were compared with model outputs (Figure 2(b)). The most significant *User Code* features, the transverse energy calculation and timing extraction, were validated by recording E_T as monitoring data and using global control *selection*, one of the configurations of *User Code*. When the global control signal of *selection* is set to False (Figure 3(a)), the output of the Finite Impulse Response (FIR) block is recorded as monitoring data. Hence, the energy calculation can be validated. As shown in Figure 3(b), when the global control *selection* is set to True, only the selected energy will be recorded as monitoring data, thus the timing extraction can be validated. Most of the *User Code* blocks are already confirmed to be functional.

Figure 2: (a) Onboard validation for main-path blocks. Monitoring data is used to check each main-path block's output. By changing *User Code* configurations and monitoring data type, every block can be validated. (b) Onboard validation for sub-path blocks. IP bus extracts sub-path block's contents directory. Blue and green modules are constructed with firmware and software, respectively. All the *User Code* onboard validations take place on a software-based checker.

Figure 3: (a) When global control *selection* is set to False, the SEL block never extracts the timing of energy deposition. Thus all the calculated energies are recorded as monitoring data. Furthermore, multiplication by DSP block can be validated by changing the global control *hardpoint*. If it is set to True, LSB of the transverse energy corresponds to 12.5 MeV. This is a dedicated situation for the FIR block validation. (b) When global control *selection* is set to True, the SEL block extracts timing information. Only some energies passing selection criteria are recorded.

6 Conclusion

An important component of the LATOME firmware, the *User Code*, computes the transverse energy and extracts the timing of the energy deposition with a fixed latency of 112.5 ns for all supercells. Each LATOME firmware receives data from up to 320 supercells at 204.8 Gbps and transfers three types of energies at 448 Gbps. Additionally, some monitoring systems and configurable remapping procedures are included in LATOME firmware. As a consequence of these variable features, LATOME firmware occupancies over 90 % of FPGA resources. Thus, all the firmware logic and stability validations are necessary. The *User Code* logic was verified with simulation by using UVVM automatically and efficiently. Its stability was validated by using real LATOME cards under the same conditions as expected in the ATLAS Run 3 at LHC. Every scheme was executed by comparing *User Code* outputs with values independently prepared from the *User Code* model.

References

- [1] Lyndon Evans and Philip Bryant, JINST 3, S08001 (2008).
- [2] The ATLAS Collaboration, JINST 3, S08003 (2008).
- [3] The ATLAS Collaboration, *ATLAS LAr Calorimeter Phase-I upgrade TDR*, CERN-LHCC-2013-017.
- [4] Bitvis AS, Universal VHDL Verification Methodology, https://github.com/UVVM/UVVM/blob/master/GETTING_STARTED.md.
- [5] W.E. Cleland, E.G. Stern, *Signal Processing considerations for liquid ionization calorimetry in a high rate environment*, Nucl. Instrum. Meth. A 338 (1994) 467.
- [6] Nico Madysa, *AREUS A Software Framework for ATLAS Readout Electronics Upgrade Simulation*, ATL-LARG-PROC-2018-008.