Contribution ID: **26** Type: **Oral**

Development of methodology and implementation of SoC-based compact single-board validation system for the ATLAS Phase-II level-0 muon trigger system

Wednesday 2 October 2024 14:00 (20 minutes)

Firmware testing on actual hardware is an optimal way to validate large-scale FPGA-based trigger/DAQ systems. For the ATLAS Phase-II level-0 muon trigger system's Sector Logic (SL) firmware, a methodology using prototype ATCA-based SL boards was developed, featuring self-complete DAQ, high-statistics test patterns, and various nature of input test data. The design exploits Zynq SoC on the board for control, injection of BRAM-based test patterns data and flexible DAQ to probe the process. The success is owing to the flexibility of the SL board design around the SoC device. This methodology facilitates precise firmware validation and systematic debugging.

Summary (500 words)

In high-energy particle physics experiments, developing a Data Acquisition system using large-scale FPGA has been a common approach to achieving efficient data taking. Modern large-scale FPGAs allow condensed data processing algorithms and frontend board data I/O, where extensive validation is essential for stable development and operation. The firmware simulation for the large-scale system tends to be time-consuming, and validation on the actual hardware is found to be an optimal solution. For the ATLAS Phase-II level-0 muon trigger system's Sector Logic (SL) firmware, we developed a methodology using prototype SL boards as comprehensive software tools and hardware frameworks around System-on-a-Chip (SoC) devices on the SL board.

Core design features include:

- A self-complete DAQ system enabling standalone single-board operation for DAQ, slow control, and readout.
- Capability of handling significant statistics and various test datasets for algorithm testing, facilitating input-output data comparison and latency validation.
- Ability to handle diverse data types (Monte Carlo simulation, actual collision, and toy trajectory data) for extensive hardware-software comparison studies.

Figure 1 details the hardware system, including the target Virtex Ultrascale+ FPGA (VU13P) and Zynq Ultrascale+ SoC for testing. The Zynq SoC, with Linux OS, serves as the master for slow control, DAQ, and timing. Test patterns stored in VU13P FPGA BRAMs are synchronized with a Test Pulse Trigger (TPT) signal. The Zynq SoC configures test patterns and synchronizes L0 accept signals with TPT. The design includes intermediate trigger data readout for debugging, and the readout algorithm compresses data effectively using AXI Chip2Chip design with high-speed serial data transmission in the Aurora protocol. This enables quick validation, even with many events.

We developed software tools for generating test patterns from various datasets and a bit-wise simulation framework coded in C++. By designing compatible input data formats for the software simulator and actual hardware, extensive hardware-software comparison is possible.

Our design has been implemented for Phase-II ATLAS Level 0 muon trigger in the endcap region. The single-board test system demonstrated testing 50,000 collision events in about one minute. Figure 2 shows a detailed comparison study between actual hardware output (black points) and simulation output (red points) for the identical inputs. It validates the firmware precisely and allows systematic study of minor bugs owing to the large statistics and realistic data input.

As the Zynq SoC is a standard solution typically for slow control in the recent data processing board and is available, this methodology and conceptual design have broad applicability. We would like to share the key features of the hardware design to realize this validation system. The knowledge and experience can be shared with other FPGA-based electronics systems in the presentation.

Author: CAMPLANI, Alessandra (University of Copenhagen (DK))

Presenter: NARUKAWA, Yoshifumi (University of Tokyo (JP))

Session Classification: Programmable Logic, Design and Verification Tools and Methods

Track Classification: Programmable Logic, Design and Verification Tools and Methods