



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

Yoshifumi Narukawa, on behalf of the ATLAS TDAQ Collaboration
The University of Tokyo

yoshifumi.narukawa@cern.ch

Validation of the FPGA-based Trigger/DAQ system

- In high-energy particle physics experiments, the Trigger and DAQ (TDAQ) system plays a crucial role in maximizing the overall performance of the experiment
- In recent years, with advancements in integrated circuit technology, TDAQ systems have become more sophisticated and complex
- To accurately implement the system and achieve optimal TDAQ performance, it is essential to develop a detailed and comprehensive electronics validation system using actual hardware

-> We have designed and implemented an advanced SoC-based validation system for muon trigger system in Phase-II ATLAS experiment

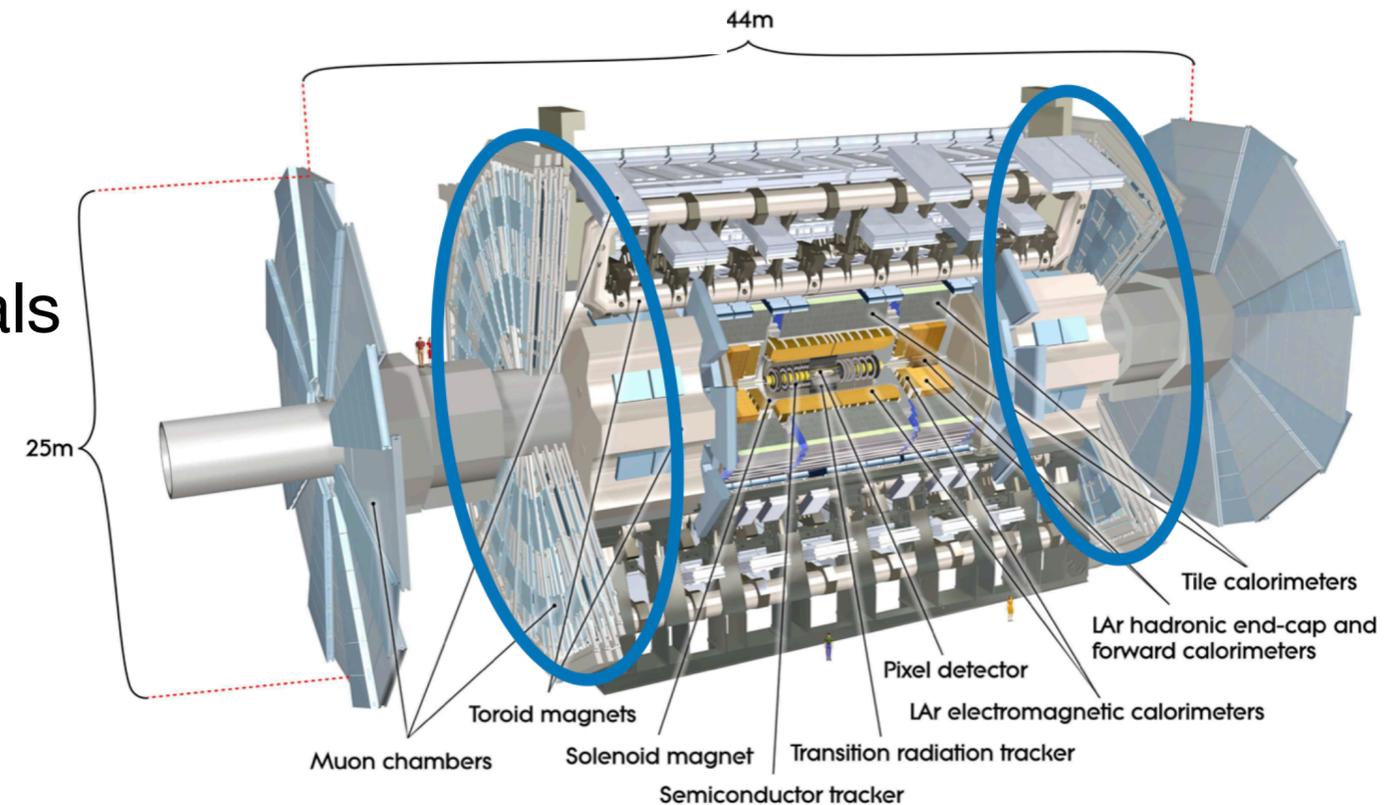
This system makes it possible to demonstrate the trigger algorithm using the large statistical “physical” datasets, such as Monte-Carlo simulation, toy straight tracks, and collision data

Phase II Upgrade for HL-LHC

- The operation of the ATLAS detector at High-Luminosity LHC (HL-LHC) will begin in 2029

	Peak luminosity ($\text{cm}^{-2}\text{s}^{-1}$)	First stage trigger rate (kHz)	First stage trigger latency (μs)
LHC (Run3)	2×10^{34}	100	2.5
HL-LHC	$5 - 7.5 \times 10^{34}$	1000	10

- Level-0 (L0) Muon Trigger system performs realtime muon reconstruction in the endcap region with layer coincidence of Thin Gap Chambers (TGC) signals
- To handle the increased collision rate and L0 trigger rate, **the readout and triggering electronics of the TGC need to be replaced**



Thin Gap Chamber (TGC)

Phase-II TGC electronics overview

ASD (Amplifier - Shaper - Discriminator)

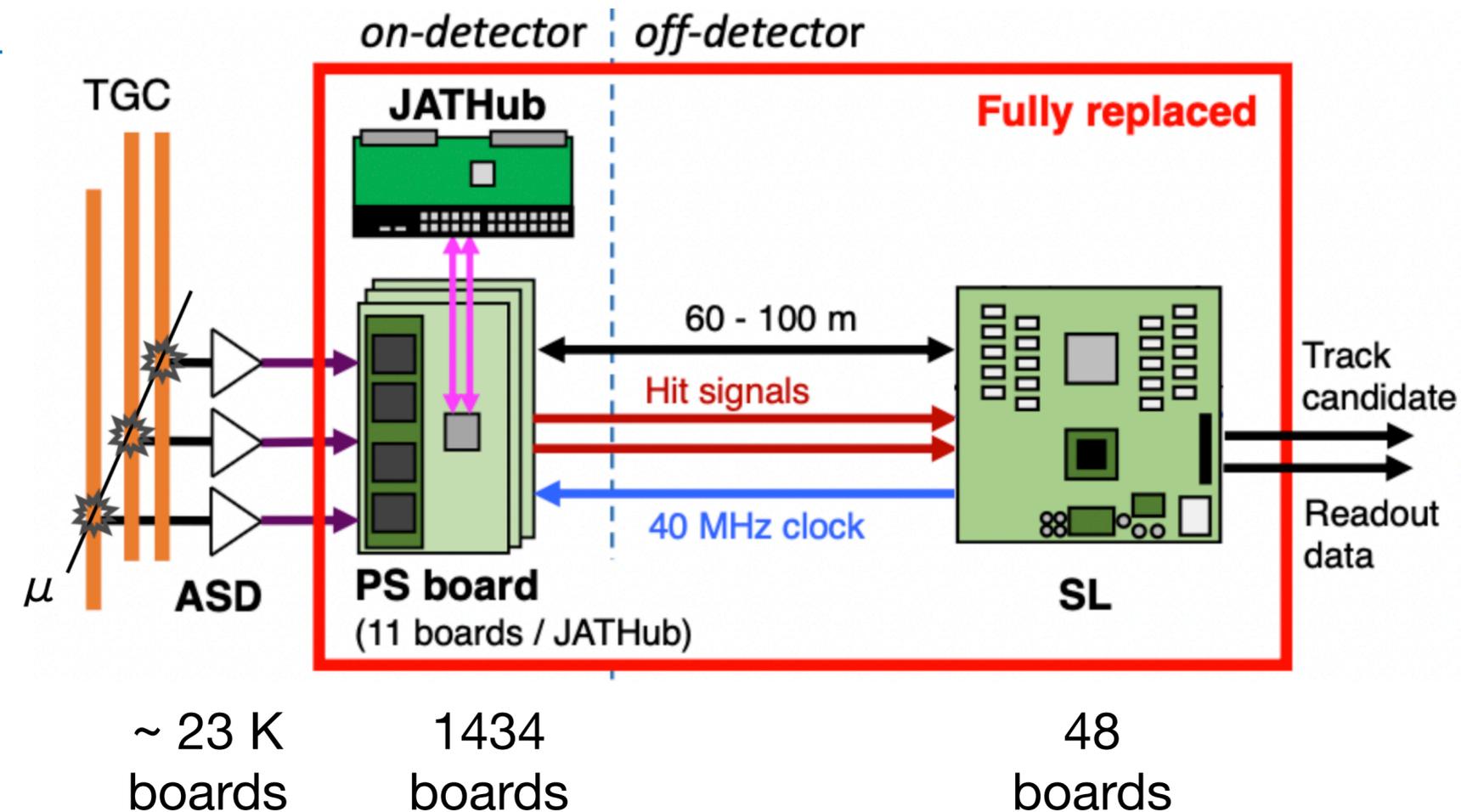
- Charge signals from TGC are amplified, shaped, and discriminated

PS board (Primary Processor board)

- Assigns hit rising edges to the corresponding bunch crossings
- Sends 256 bits of hit bitmap to SL every 25 ns

Endcap SL (Endcap Sector Logic)

- Trigger : Performs muon track reconstruction and estimates p_T by a layer coincidence of TGC hits
- Readout : Retrieves hit data and results of trigger calculation for each triggered event and transmits it via an optical link to the latter module
- Control : Distributes timing and control signals to the PS boards



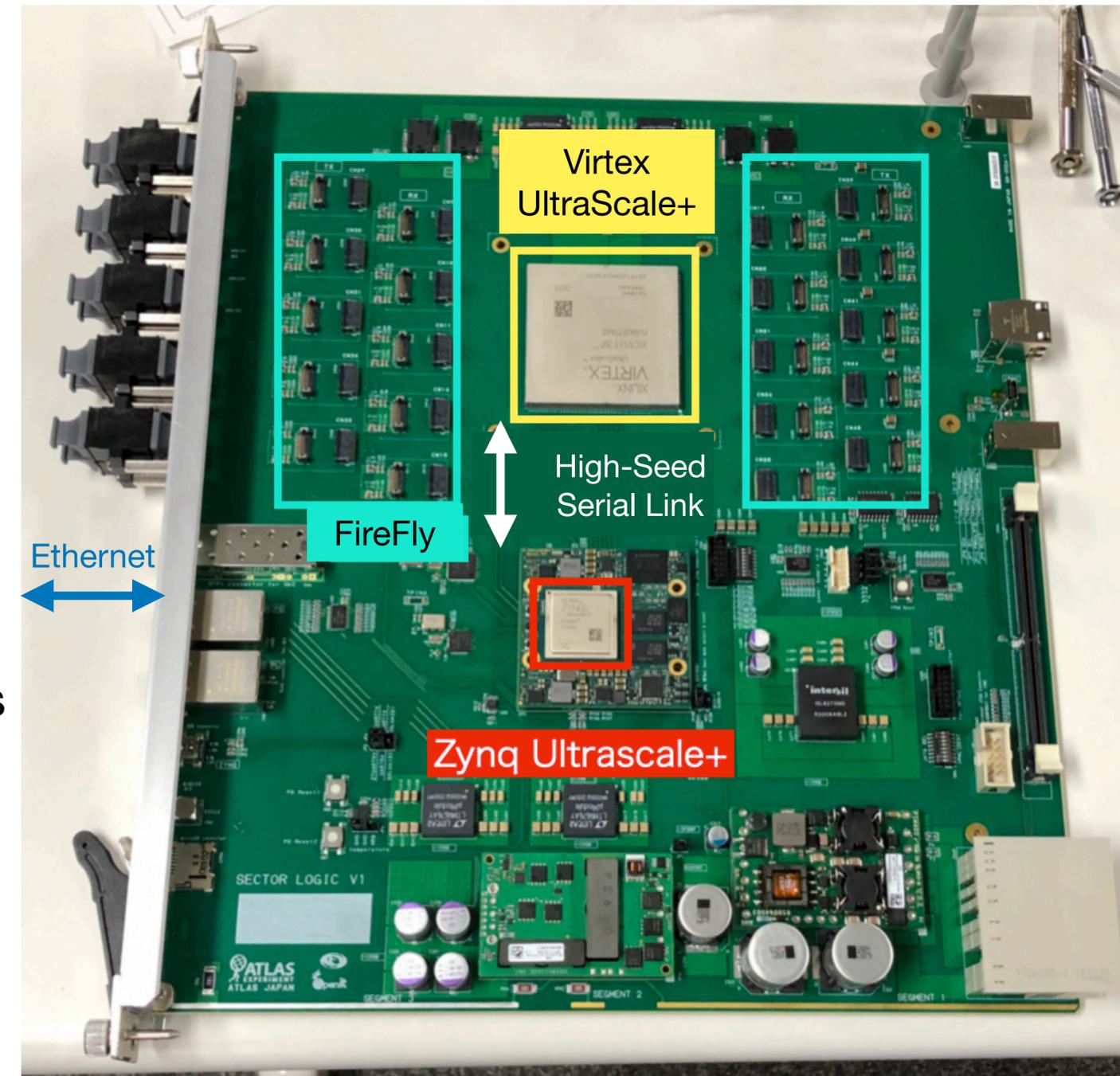
Endcap Sector Logic (SL)

Virtex UltraScale+ FPGA (XC7VU13P-1FLGA2577E)

- Receives signals from the PS boards, calculates muon trigger candidates, and reads out hits and trigger information for L0-selected events
- Large-scale FPGA consisting of four silicon chips
- High-performance multi gigabit SERDES transceivers (GTY) in 32 quad banks

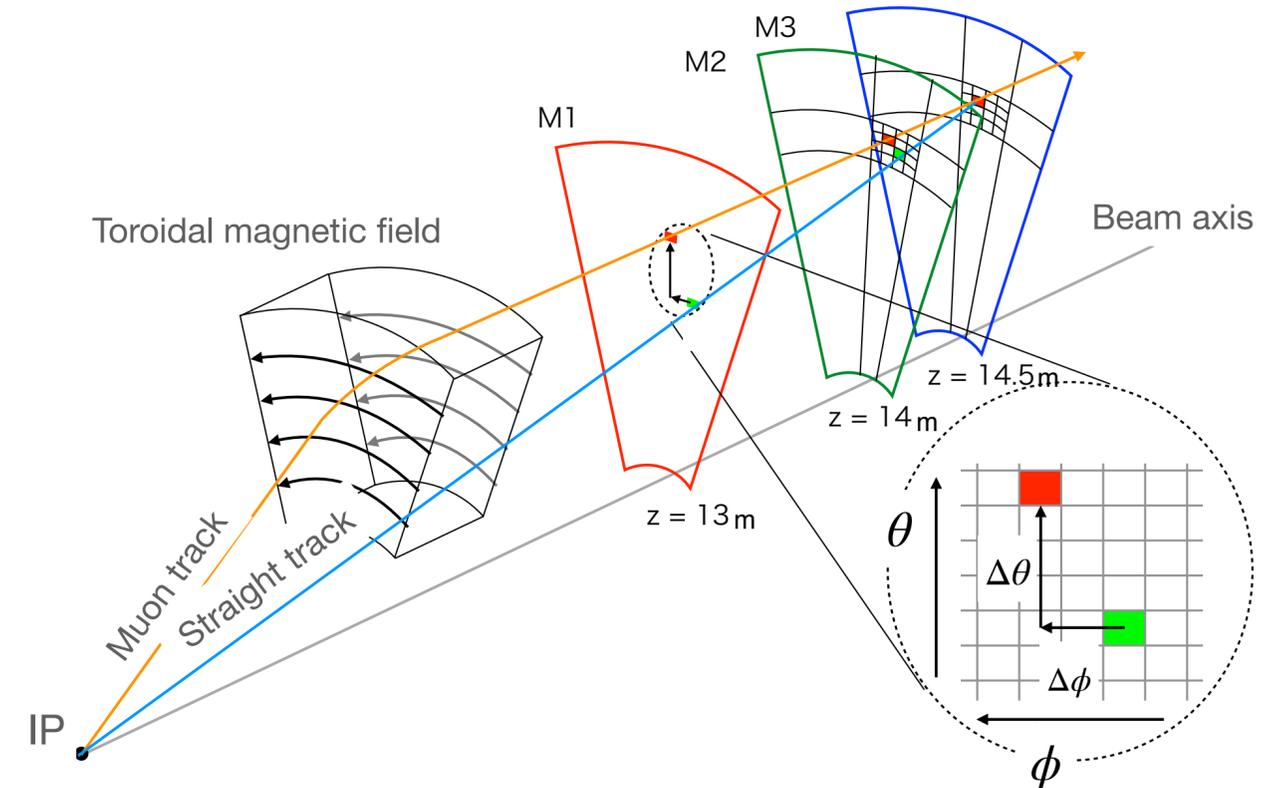
Zynq UltraScale+ MPSoC (XCZU5EV-2SFVC784I)

- Interfaces with TDAQ systems via ethernet. Serves as the control master for the Virtex Ultrascale+ FPGA and PS boards
- High-performance multi gigabit SERDES transceivers (GTH) for communication with Virtex UltraScale+ FPGA
- MPSoC offers flexible functionalities with its processor system for control and debugging
 - which will be exploited in the validation system design

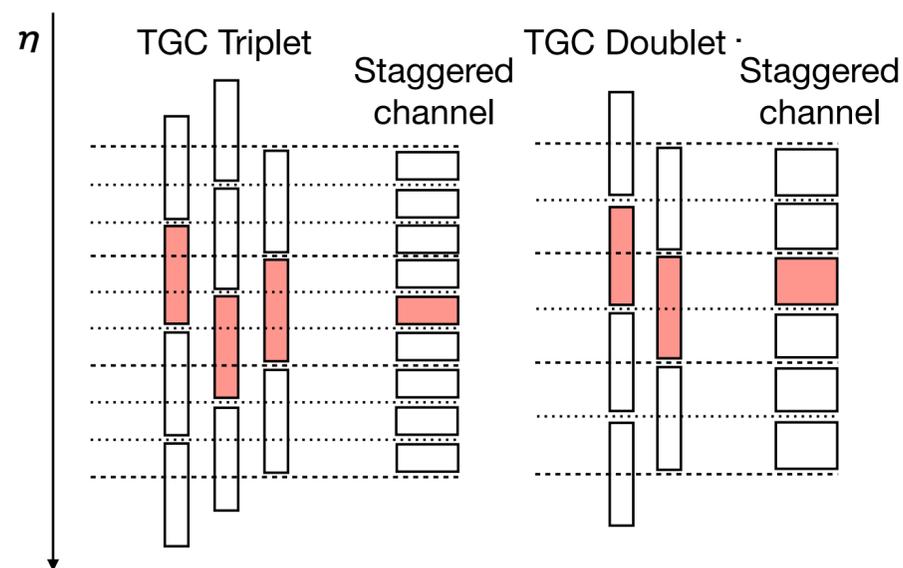


Overview of the L0 endcap muon trigger algorithm

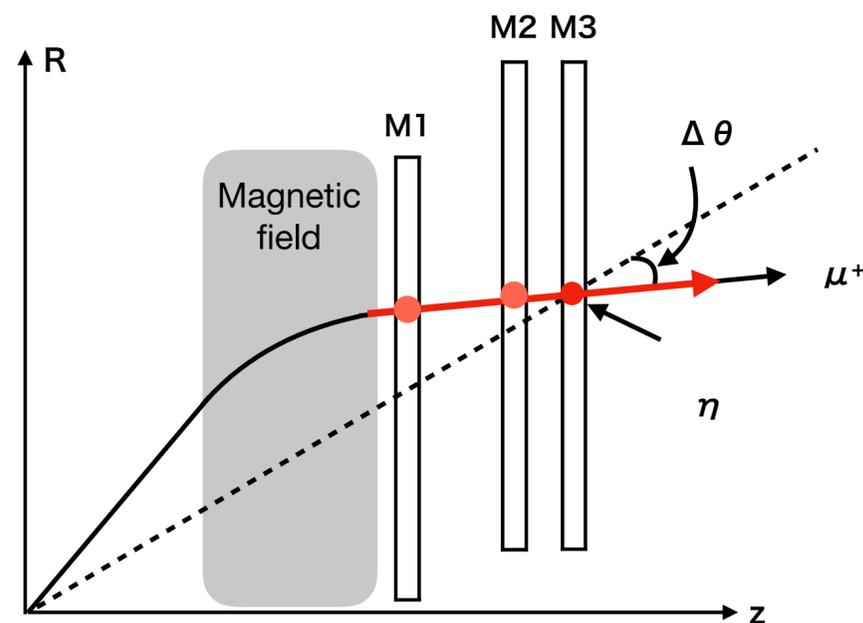
- There are seven layers of TGC chambers, organized into three stations
- Muons coming from the IP are bent in a eta direction by toroidal magnetic field
- **Momentum can be estimated by a point angle measurement with the $(\Delta\theta, \Delta\phi)$ observables**



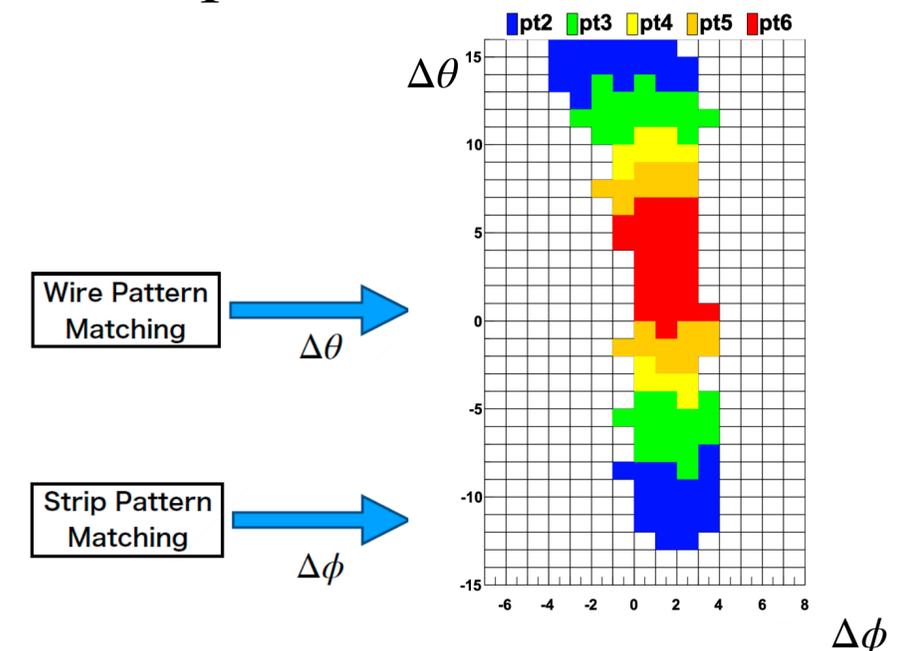
1. Station Coincidence



2. Segment Reconstruction



3. p_T estimation with LUTs



Trigger Validation System

Test Vector Generator

- A software tool which emulate the fiber routing from the detector channel to SL
- It uniformly generate pseudo input of SL from every types of datasets

Single-board test system

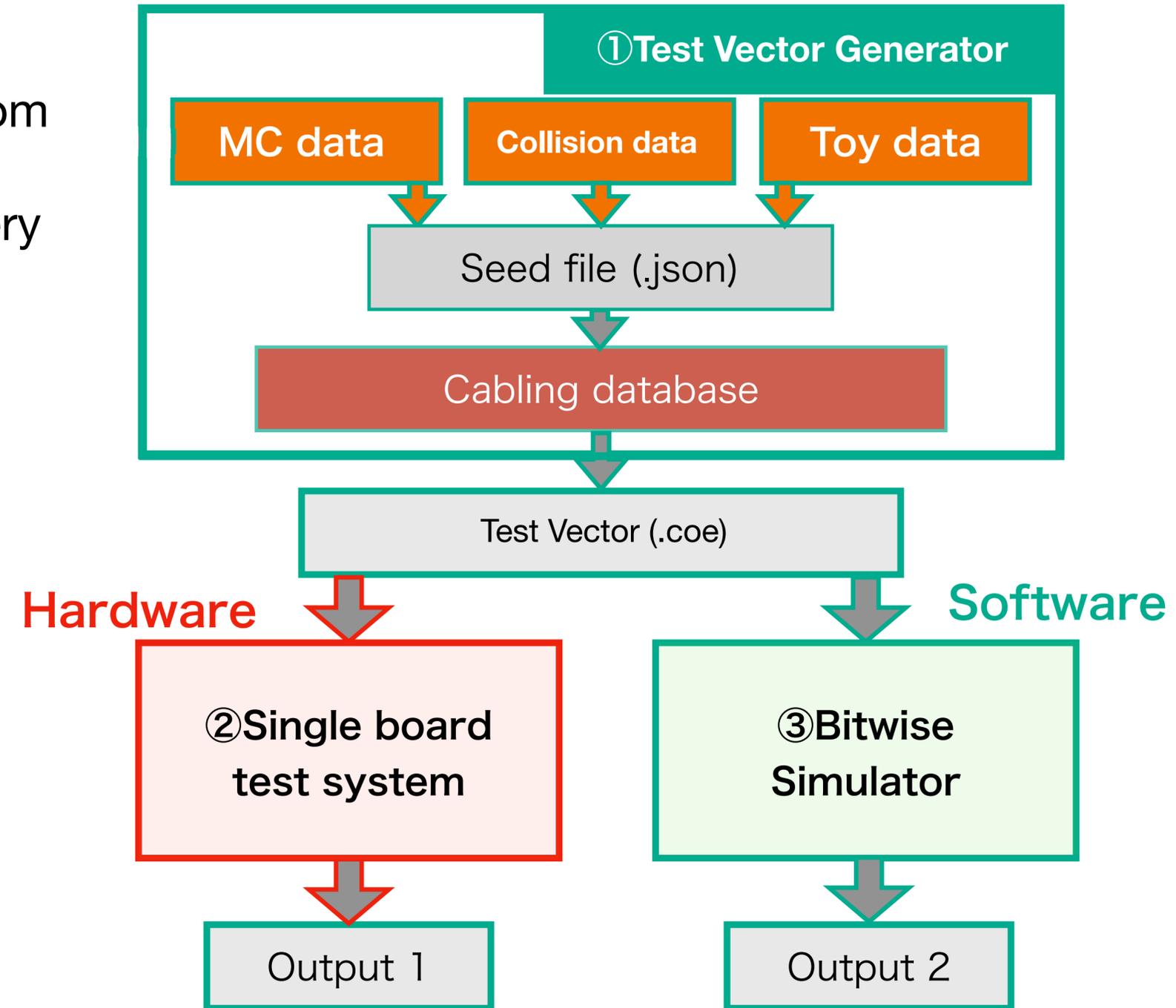
- A SoC-based hardware system that performs timing control, register control, and readout data

Bitwise simulator

- A software simulator that completely emulates trigger logics of hardware at the bit level

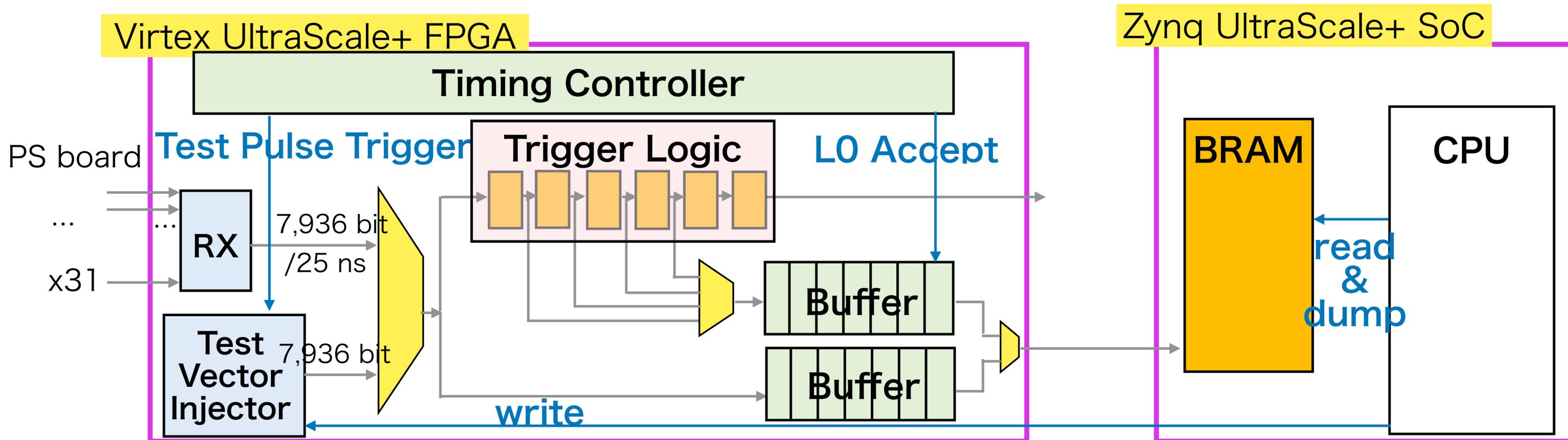
By processing identical input data on two paths and checking the consistency of those outputs

→ We can achieve detailed validation



Overview of single-board test system

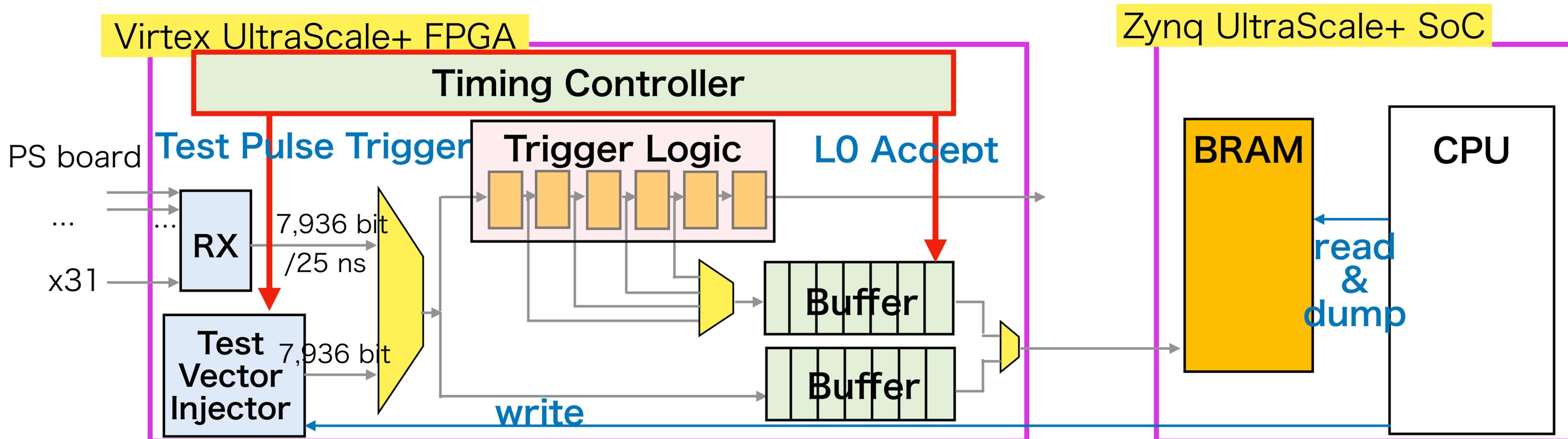
- MPSoC (linux booted) serves as a master of validation system for control and readout
 1. MPSoC writes the test vector into the BRAM inside the test vector injector
 2. Timing Controller outputs the Test Pulse Trigger (TPT) signals
 3. The trigger output is stored in the Buffer, and only the event that receive the L0A signal are passed through to the MPSoC
 4. Data in the BRAM is read out by CPU



Implementation of single-board test system

Timing Controller

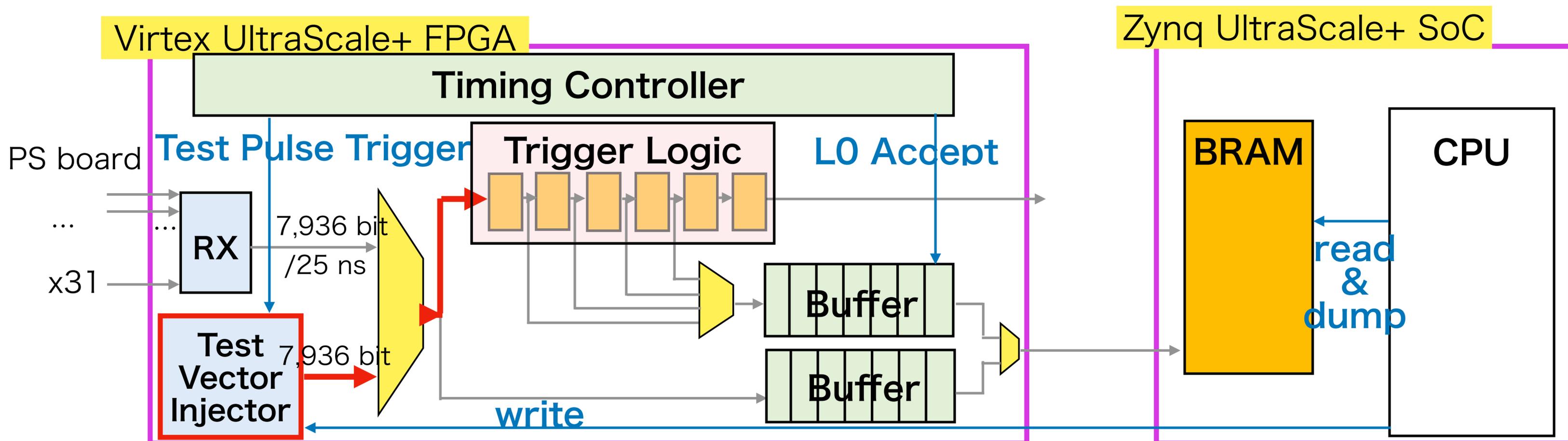
- Emulates the role of the central trigger and generates timing signals such as test pulse trigger and Level-0 Accept (L0A) signals
- L0A signals are generated after a certain period following the test pulse trigger signals, corresponding to the L0 latency



Implementation of single-board test system

Test Vector Injector

- Stores test vectors of 7936 bits in a BRAM and injects them simultaneously at 40 MHz in sync with the test pulse trigger signal
- The BRAM has a depth of 60, and it can be rewritten repeatedly from CPU, which enables test with high statistics

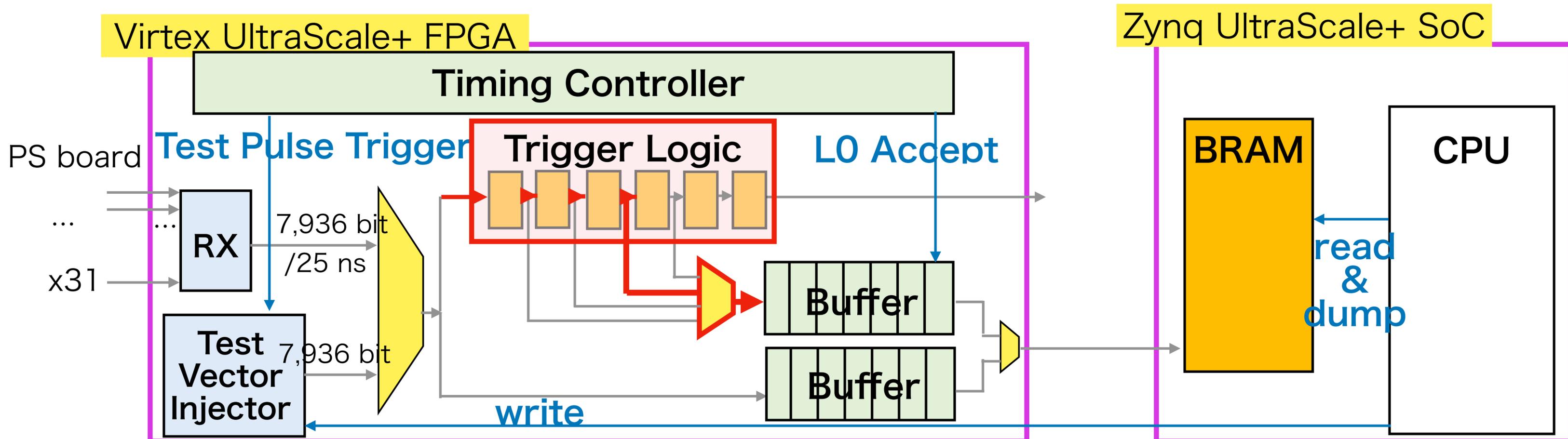


Implementation of single-board test system

Trigger Readout Selector

- Selects which trigger module's output to read out by simple register control
- The depth of the buffer is set to different value tailored to each trigger stage

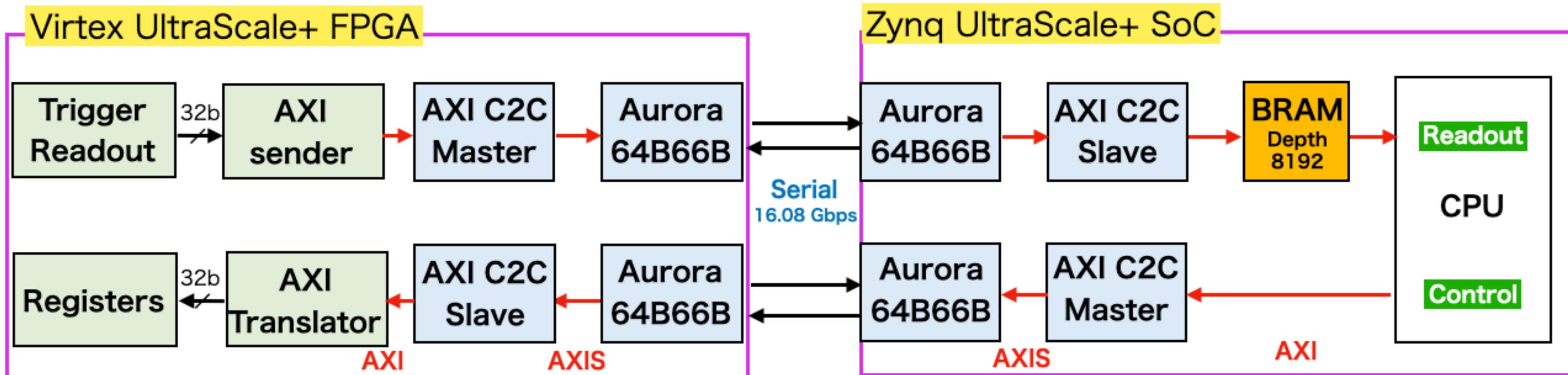
->By confirming that the correct data is being read out through L0A, it is also possible to verify that the trigger logic is operating with the expected latency



Technical detail: Inter-chip communication

Two bi-directional serial links: for control and readout

1. In the readout path, the AXI sender converts the 32-bit trigger data into the AXI protocol format
2. AXI C2C Master converts them into AXI Stream and Aurora 64B/66B encodes these data into serial format to communicate with the MPSoC using the GTH giga bit transceiver
3. On the MPSoC side, data is decoded and transferred through the reverse process, and finally dumped into the BRAM
4. In the control path, conversely, the MPSoC acts as the master of the AXI bus and performs read and write operations to registers in the FPGA



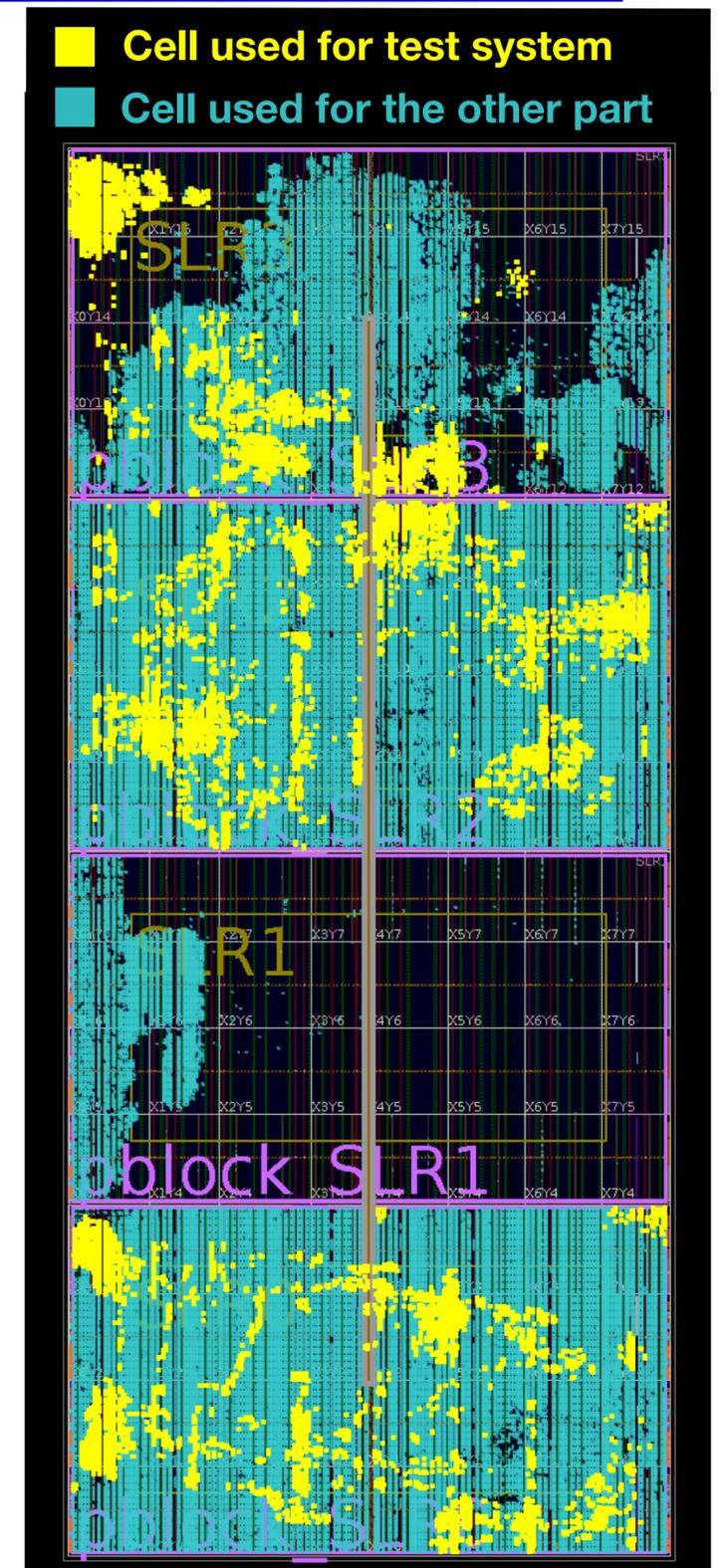
Resource usage

- Even when combined with the other functionality of the SL, there is a sufficient room for resource usage at this stage
(* Firmware development of the SL is currently ongoing)

Resource usage for single-board test system

	LUT (1728000)	REG (3456000)	CLB (216000)	BRAM (2688)	URAM (1280)
SLR0	0.51	0.26	1.54	5.47	0.00
SLR2	0.52	0.28	1.70	5.84	0.00
SLR3	0.42	0.32	1.27	2.96	0.00
Total	1.45	0.86	4.51	14.27	0.00

(%)

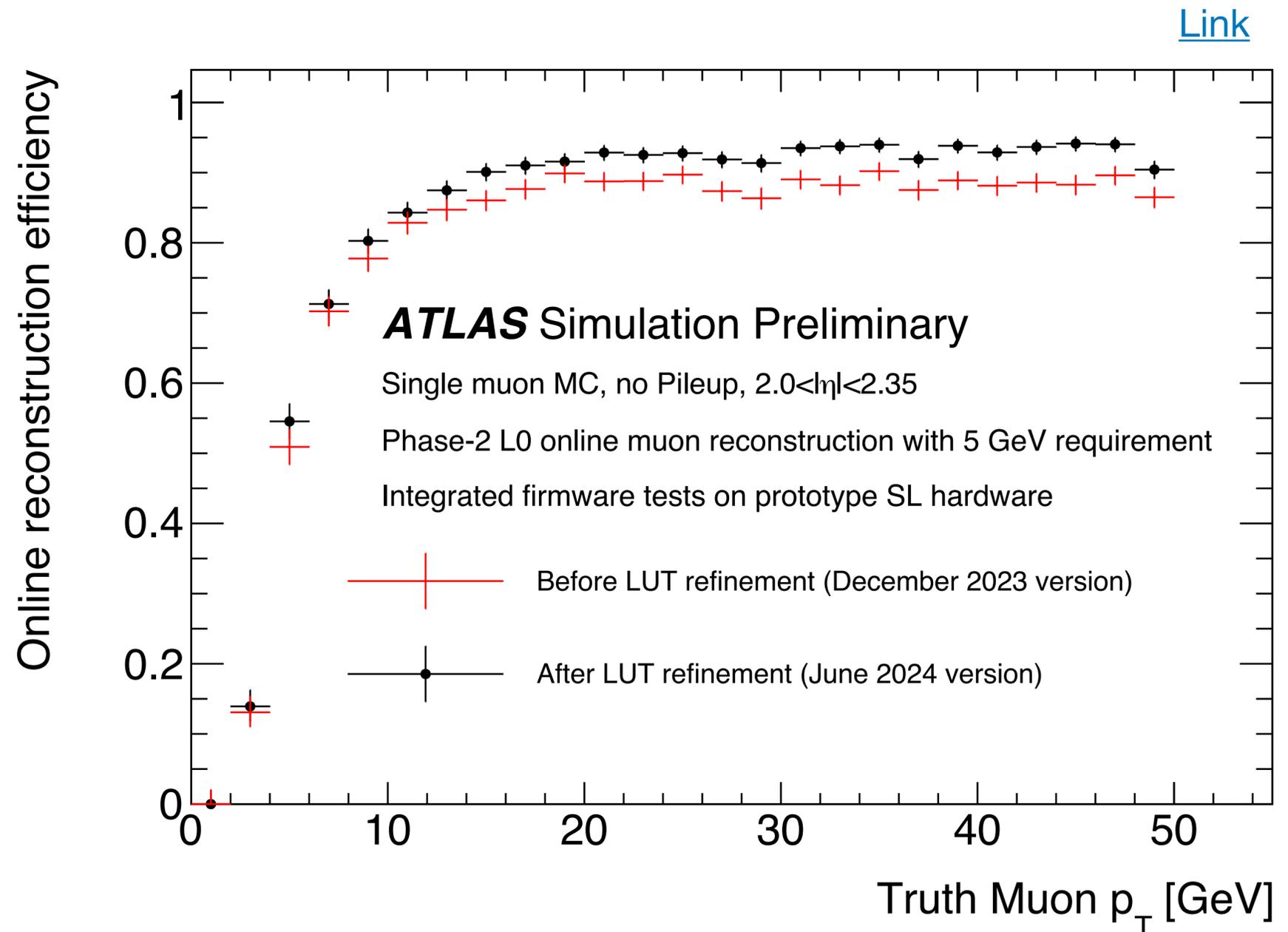


Trigger performance evaluation using single-board test system

- We successfully evaluate the trigger efficiency using single-board test system
-> providing an excellent opportunity for testing, validation, and debugging of the firmware

[Example of usage](#)

- Use a large statistics single muon MC dataset to evaluate the trigger efficiency (500 K events, $0 < p_T < 50$ GeV)
- We observed that the efficiency is lower than expected from the simulation (~94%)
- We investigated the output of each module and identified issues in the logics and LUTs
- By applying the fix, efficiency was improved



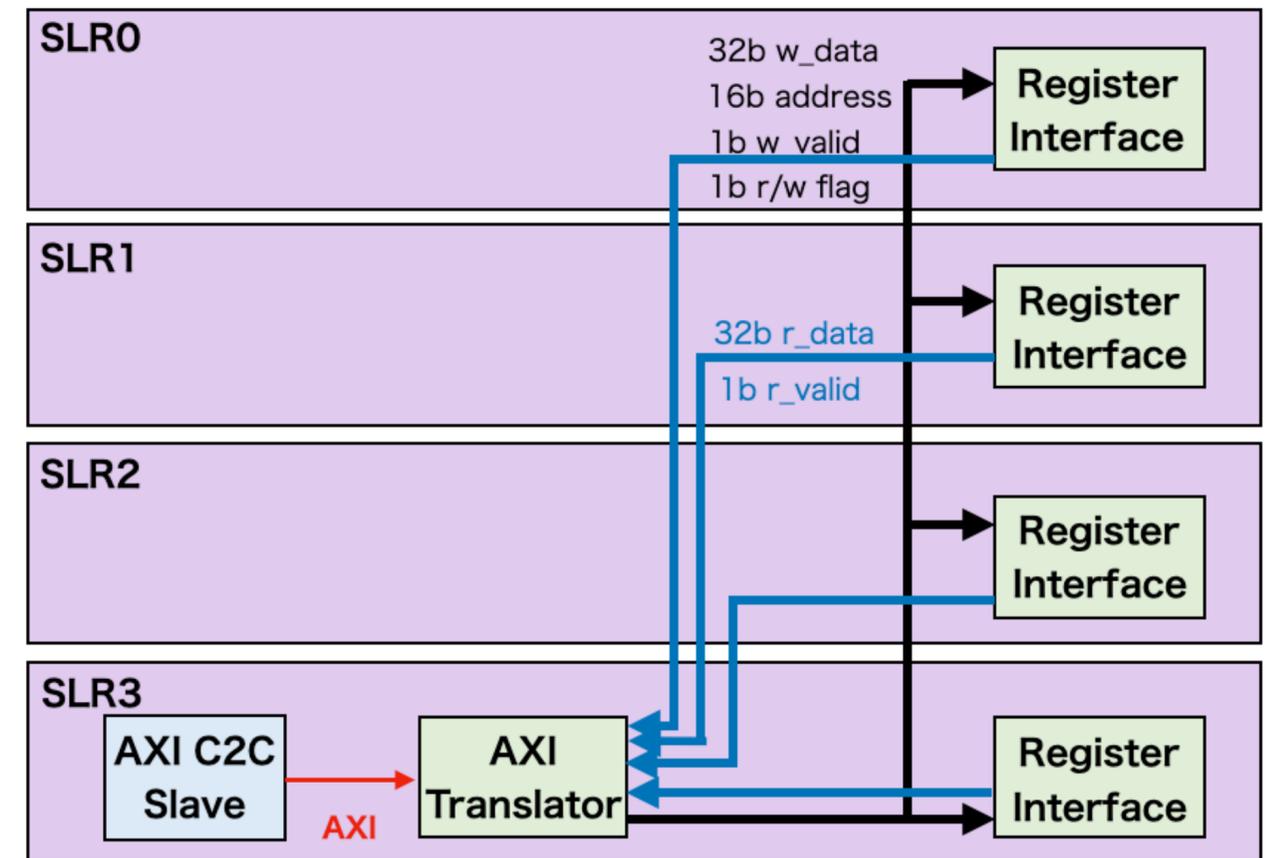
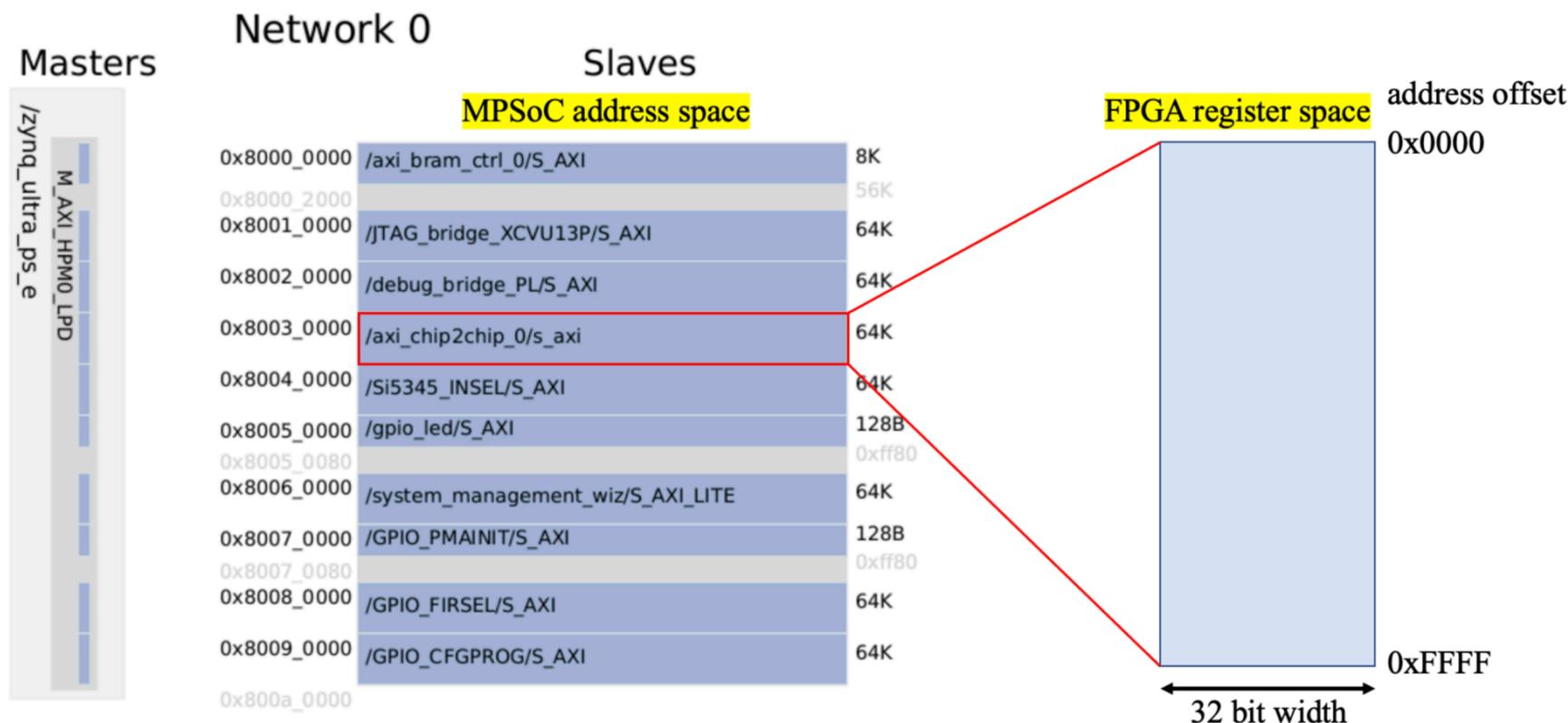
Summary

- Establishing a detailed and comprehensive validation system using actual hardware is critical to realize the next-generation TDAQ system
 - The SoC-based single-board test system is one solution
 1. Take a sufficient amount and variety of data, such as MC simulation, toy straight track, and actual collision data
 2. Monitor the trigger output (or intermediate output) and perform comparisons with the input data or with simulations
 3. Verify that the logic circuits are implemented with the expected latency
 - We have completed the design and implementation of this test system for the Phase-2 TGC Endcap SL and are now fully utilizing it in studies aimed at maximizing trigger performance
- ➔ This validation method and implementation techniques are widely applicable to electronics systems with SoC**

Back up

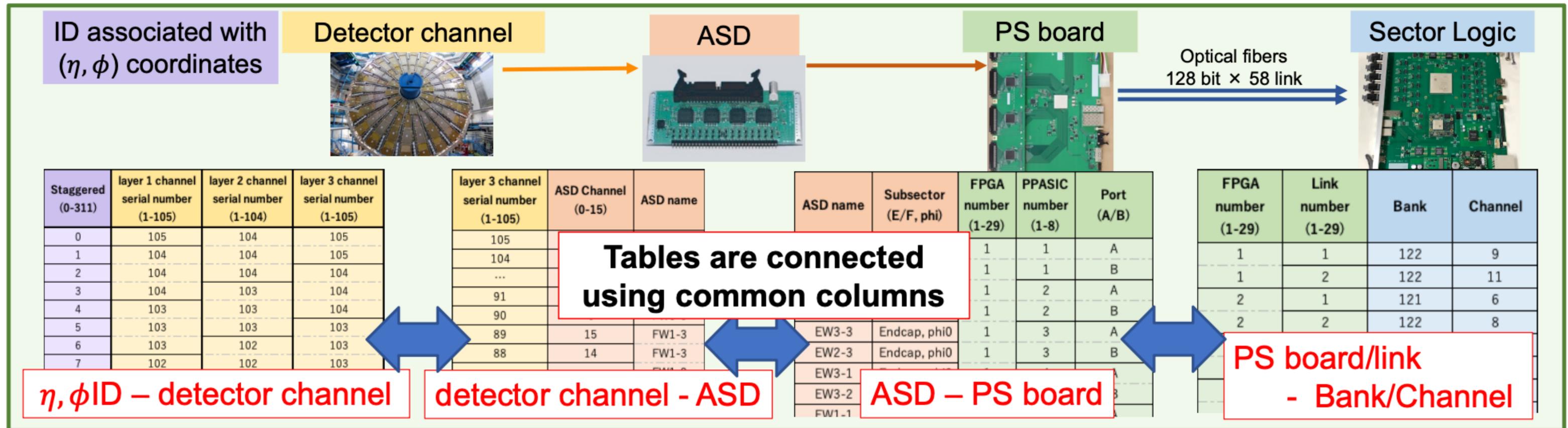
Technical detail: Register Control

- In the MPSoC, the modules connected via the AXI protocol from the CPU, are mapped to the physical address space of CPU
- We designed to map the addresses assigned to AXI C2C to the address of FPGA registers -> CPU can seamlessly operate the registers on the FPGA as if accessing physical memory
- On the FPGA side, the AXI Translator mediates the AXI C2C and the registers. By connecting each register using the custom protocol instead of the AXI protocol, the bus width for crossing SLRs can be minimized



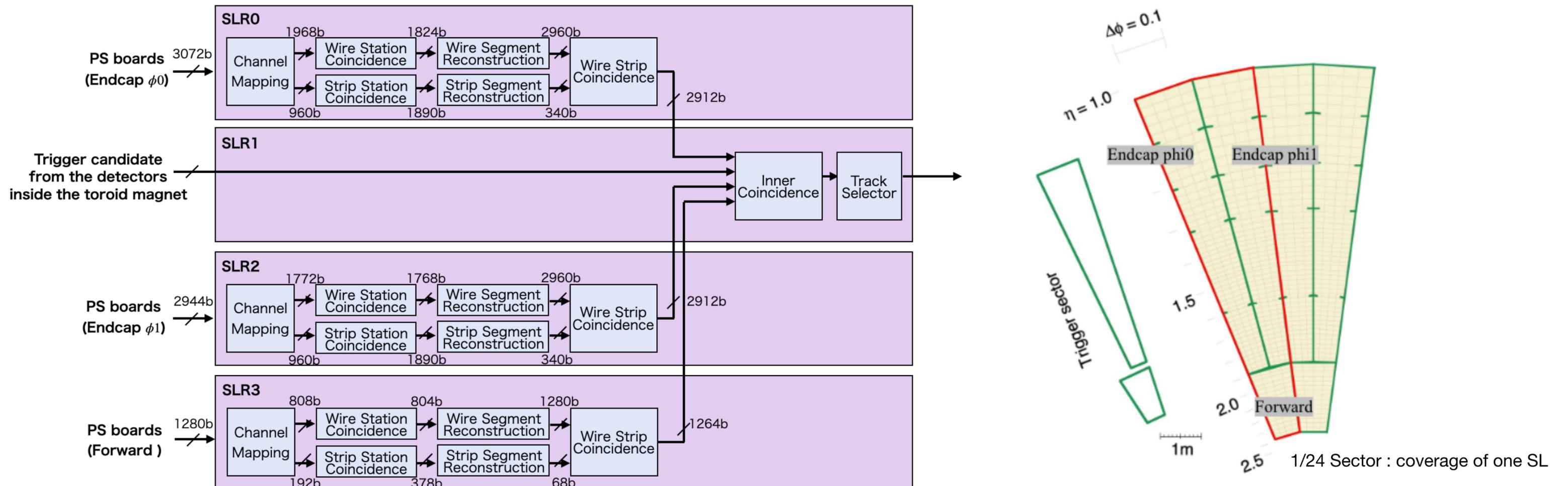
Preparation of test vector with relational database

- The test vector generator utilizes a relational database to provide the cable mapping between TGC chamber channels and the SL input hit bitmap
- A hit bitmap of 7396, serving as pseudo input data from 31 PS boards per SL per bunch crossing, is prepared
- Uniformly generate test vector from every types of datasets which include TGC hit channel information, such as MC simulation, toy straight track, and actual collision data



Implementation of Trigger Logic

- 1/24 sector is divided into three trigger sectors with the boundaries of the detector
 - The three trigger sectors are assigned to different FPGA Super Logic Regions (SLR) and are processed in parallel
 - In each trigger sector, trigger calculation is divided to multiple unit. For each unit, LUT are prepared. (e.g., 360 units for segment reconstruction, 78 units for Wire Strip reconstruction)
- ➔ For optimal performance, it is important to comprehensively check that the logic and LUTs are correctly implemented across all units



Technical detail: Trigger Readout

- The trigger readout circuit effectively retrieves thousands of bits of output
 - Inter-chip communication link is established in SLR3
 - Signal lines crossing between SLRs should be minimized to reduce pressure on the FPGA's timing constraints
 - > We install the readout circuits in parallel across each SLR, serialise the trigger output into 32-bit, and then collect those in the Event Builder at SLR3
- The candidate selector compresses data by selecting only the units where coincidences passed
 - > We can reduce the time required to read out a single event (~ 10,000 events per minute)

ex)
Strip Segment
Reconstruction

