A System-on-a-Chip Based Front-end Electronics Control System for the HL-LHC ATLAS Level 0 Muon Trigger

Aoto Tanaka (U-Tokyo, ICEPP) on behalf of the ATLAS Collaboration

Email : aoto.tanaka@cern.ch

22nd IEEE Real Time Conference (Oct. 12-23, 2020) ID : 211

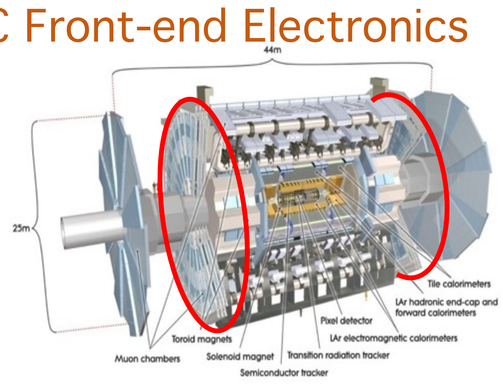
Abstract

For your reference : [1] Technical Design Report of HL-LHC ATLAS [Link]

Exploiting commodity FPGAs in the front-end electronics is a standard solution for readout and trigger electronics for the HL-LHC. It requires the system to be capable of configuring, debugging and testing the firmware from a remote host. Furthermore, the control system is also responsible for monitoring and recovery in case of the SEU errors on the configuration memory of the FPGAs. Realizing such control capability is a unique challenge for the HL-LHC experiments. We have developed a control system to achieve such requirements with a new module exploiting System-on-a-Chip device as the solution for the Level-0 endcap muon trigger system of the LHC-ATLAS experiment. The new module named JATHub module will provide functionalities of flexible control of the system: configuration, testing, monitoring, and debugging the FPGAs on the front-end electronics. In addition, realization of the robust operation scheme for the possible radiation damage on the JATHub module is another essential item of development. We have invented a scheme to mitigate SEU errors on the SoC device and a redundant boot mechanism that will guarantee robust operations even in case of the radiation damages in the flash memory devices during the physics data taking with beam collisions. We have fabricated two prototype boards of the JATHub module. Demonstration study with the proto-modules will verify the concept of the use of SoC for the management of the front-end FPGAs in the experimental cavern of the collider experiments for HL-LHC.

HL-LHC and Upgrade of TGC Front-end Electronics

- HL-LHC
  - Upgrade peak luminosity 5~7.5 $\times 10^{34} cm^{-2} s^{-1}$
  - Start operation from 2027
- Upgrade of TGC front-end electronics for HL-LHC
  - Renew readout・trigger electronics
  - Use FPGAs in the front-end to transfer all hit signals to ATLAS counting room by high speed optical-link
    - TTC signal distribution and register control are also realized with optical-link

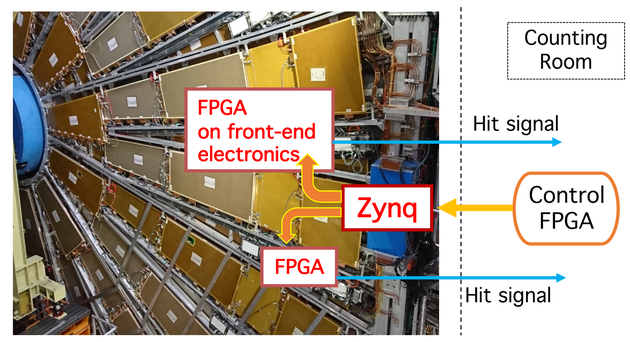


ATLAS and TGC

Control of Front-end FPGAs in HL-LHC TGC

- For the stable operation of the TGC electronics system, FPGAs in the ATLAS cavern need to be controlled from outside
  - Program and debug firmware
  - Build a robust front-end system
    - Recover from Single Event Upset (SEU) caused by radiation damage
    - Reset and reestablish optical-link, triggered by outside

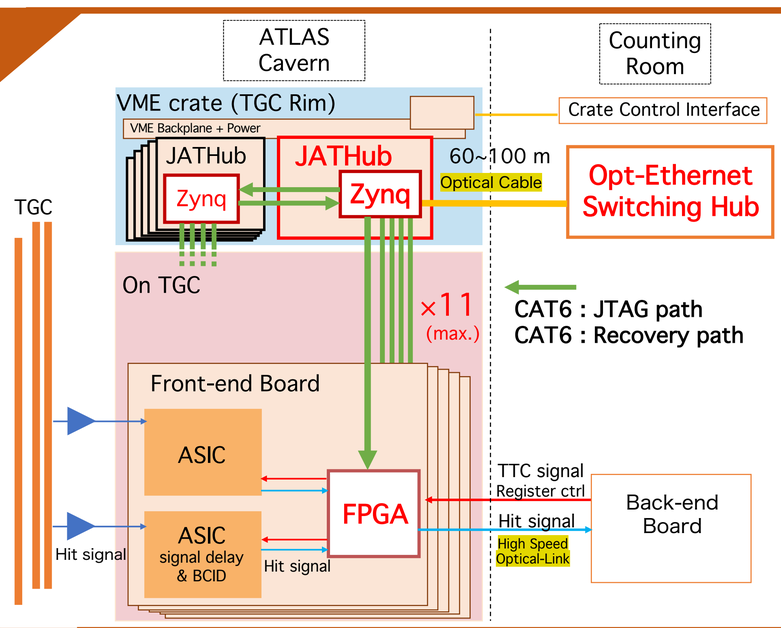
→ **Zynq SoC based control system has been developed**



JTAG Assistance Hub (JATHub) module

Board Design Overview

- Introduced JATHub module to control Front-end Boards, and designed the control scheme
- Design of the JATHub
  - Connectable to max. 11 Front-end Boards with 2 CAT-6 Cables for each board
    - Front-end Board: 1434, JATHub: 148
    - Reuse currently available signal cables
  - Equipped with Zynq SoC as the main driver
  - To be installed in VME crate (inherited from the current system)
    - 6 JATHubs in 1 crate



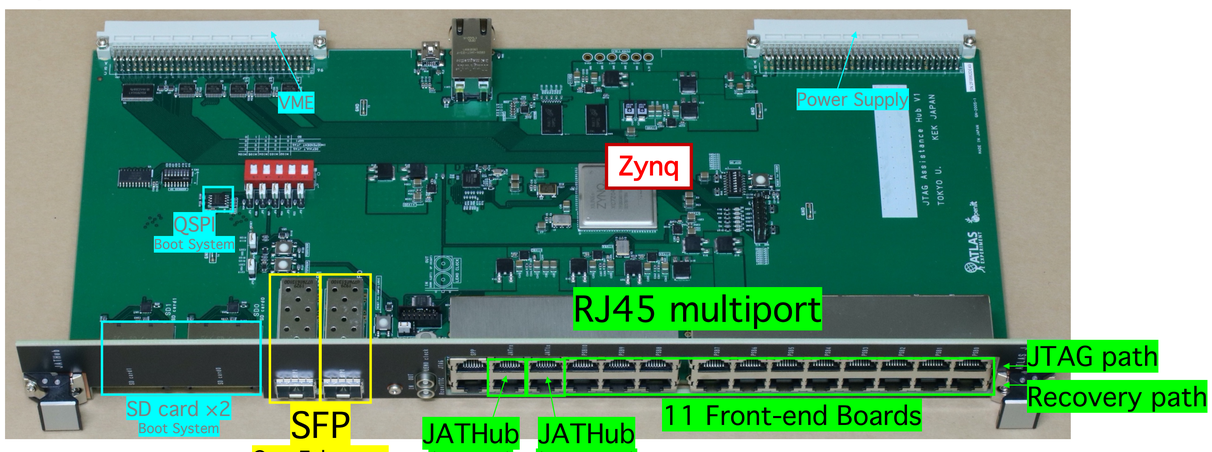
Required Functionalities

- Opt-Ethernet network
- Control of Front-end Boards ×11(max.)
  - Configuration of FPGA by JTAG access
  - Recovery of FPGAs from unrecoverable SEU error
  - Monitoring distributed TTC clock
- Mutually control with neighbouring JATHub
  - Configuration of Zynq by JTAG access
  - Recovery of Zynq from unrecoverable SEU error
- Redundant boot system with SD card ×2, QSPI (backup)
- VME Control

Testing of the JATHub Prototype

JATHub Prototype

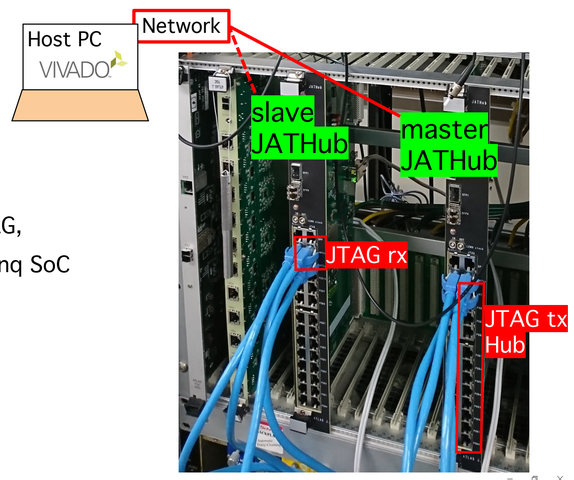
- Fabricated 2 JATHub Prototype Boards (delivered on 2020/4/17)
  - Zynq SoC : XC7Z045-2 FFG900I
  - Main Interface : RJ45 for Front-end Board ×22, SFP for Opt-Ethernet , SD card ×2
  - 14 layered board

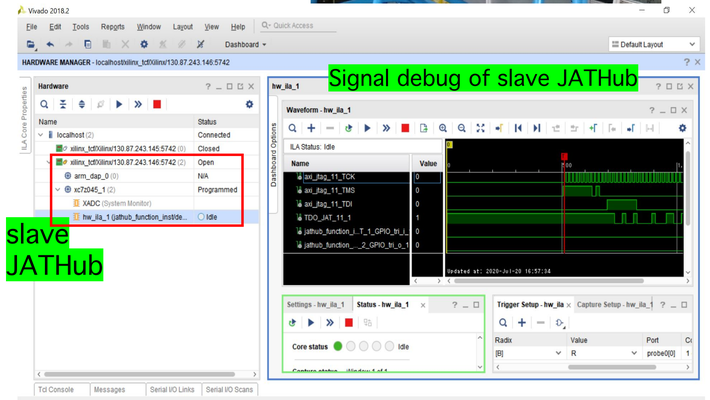


Open source consortium of Instrumentation

JTAG Access to remote FPGAs

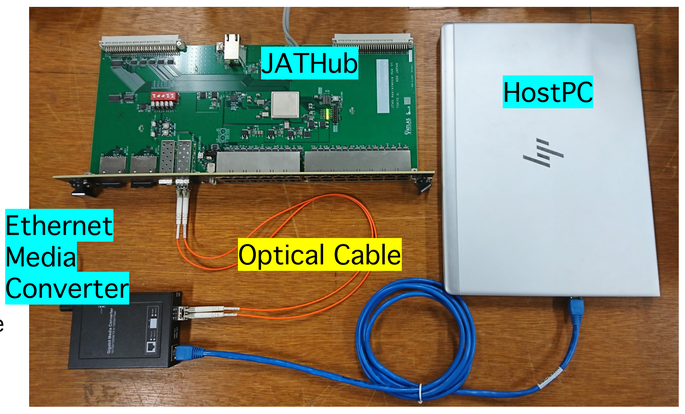
- Access master Zynq SoC with TCP/IP from Host PC, and access other slave Zynq or FPGA from master Zynq SoC with JTAG, by running the Xilinx Virtual Cable (XVC) application in master Zynq SoC
- Enable to use the functions of Vivado running in Host PC for remote FPGAs
  - Program firmware to FPGA (Indirect program of SPI flash is not supported by XVC)
  - IBERT
  - ILA (Integrated Logic Analyzer) for debug
- Developed and implemented this JTAG access with hub function
- Tested this JTAG access function
  - Master JATHub connects with slave JATHub (1m CAT6, TCK 3.125MHz), and slave Front-end Boards (15m CAT6, TCK 2.5MHz)
  - Succeeded to configure, program, and debug slave FPGAs and zynq





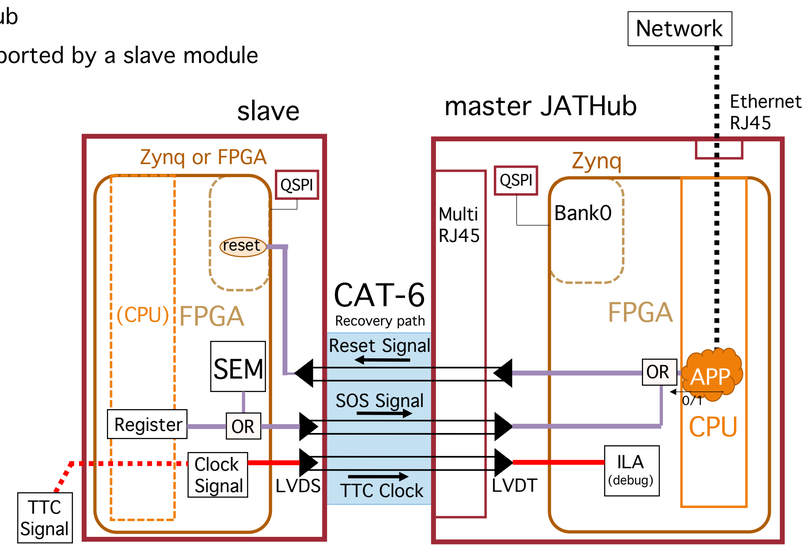
1000Base-X Ethernet via Optical Cable

- Implemented PHY in FPGA part of Zynq, and connected CPU and GTX transceiver (which connects to SFP module)
  → Enable to communicate with Zynq by optical cables
- Tested Opt-Ethernet using the JATHub prototype
  - Connected Host PC and JATHub with Optical Cable
  - Succeeded to establish ethernet network
    - Successfully sent PING and SSH



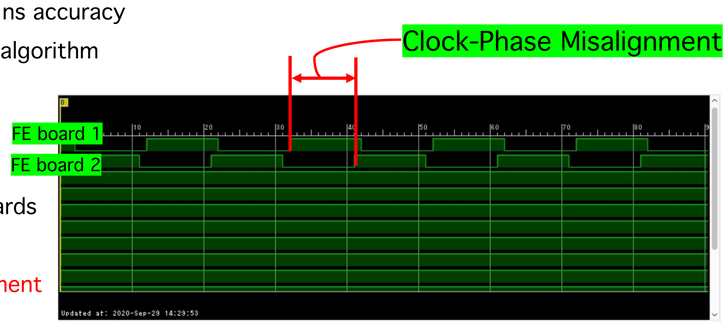
Operation Scheme for SEU on Slave Modules

- Soft Error Mitigation (SEM) controllers are running on the FPGAs and Zynqs, and automatically recover the most of SEU errors
- Reset signal are triggered by JATHub when unrecoverable SEU error is reported by a slave module via a robust communication path
- Implemented the recovery path in master JATHub, with voting logic at various places of the path
- Tested this recovery function
  - Master JATHub connects with slave JATHub, and slave Front-end Boards
  - Succeeded to recover the slave modules



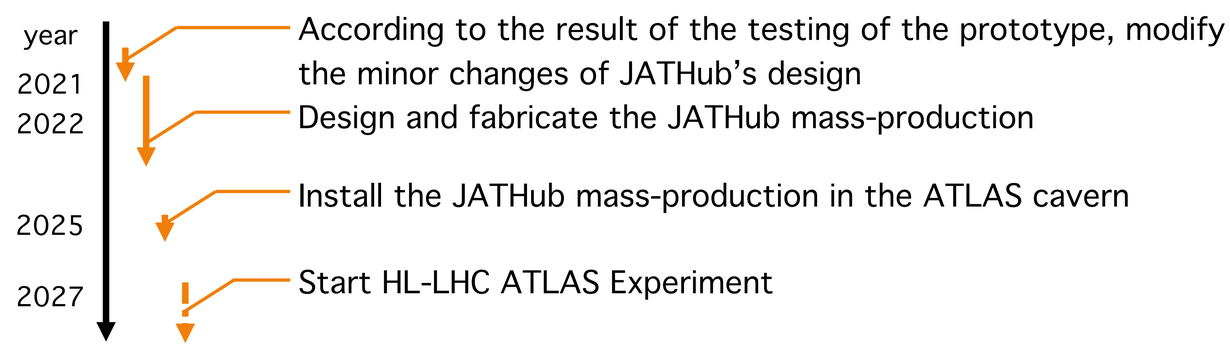
Monitoring Clock Signal from Slave Front-end Board

- In practical use, monitor TTC clock (25ns) in 1ns accuracy by a sophisticated clock-phase measurement algorithm (to be developed)
- Tested clock-phase monitoring
  - Monitor 2 clock signals from 2 front-end boards using ILA (2.5ns accuracy)
  - Succeeded to monitor clock-phase misalignment

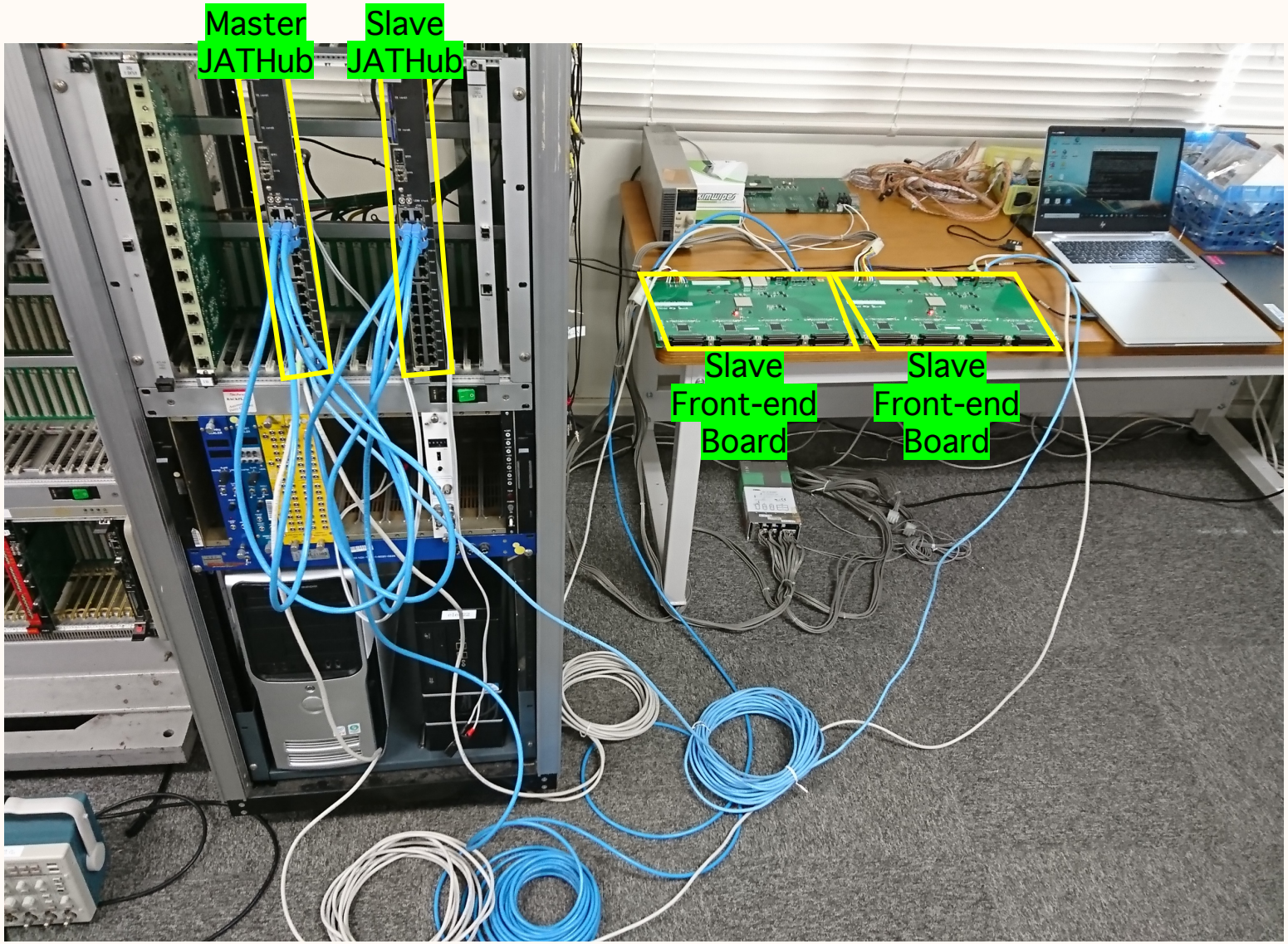


Summary and Future Prospects

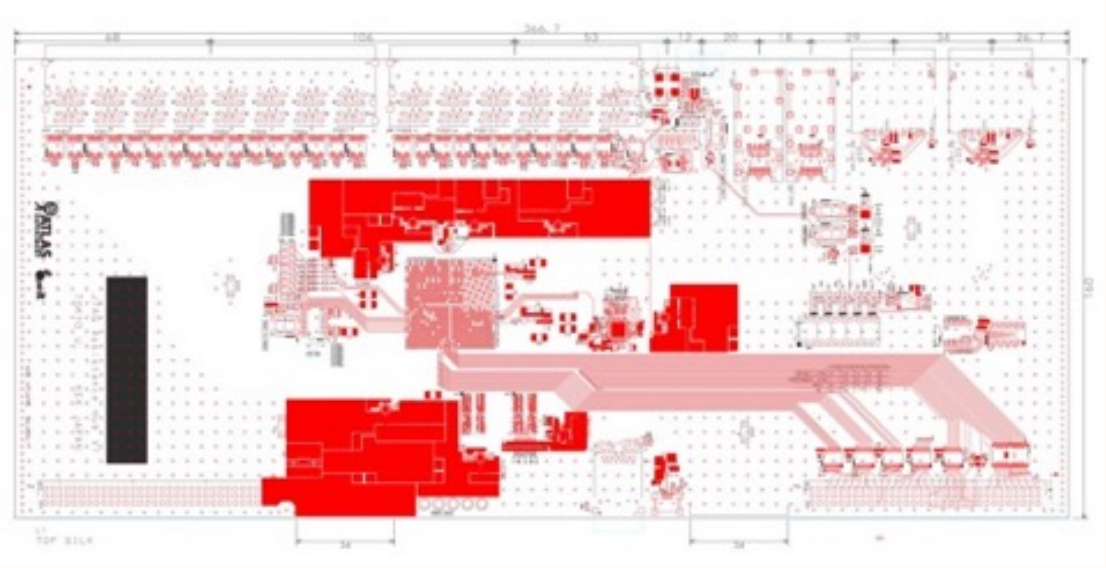
- Designed and developed the front-end control system of HL-LHC ATLAS Muon Trigger
- Designed JATHub module as the center of the control system satisfying the following requirements
  - Remotely control and monitor about 1500 of the FPGAs in ATLAS cavern
  - Ensure the robust operation even in the high-level radiation area
- Fabricated 2 JATHub prototype boards, tested them and confirmed most of the functions of the JATHub work properly as we designed and developed



JATHub Testing with 2 Front-end Boards

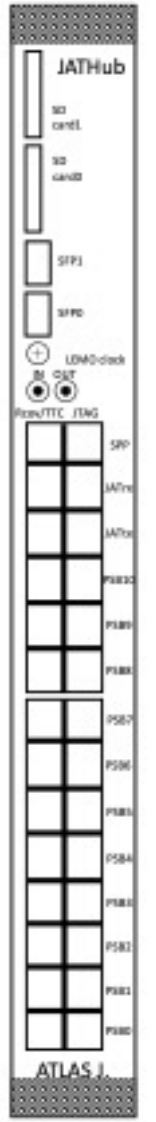


JATHub Prototype Fabrication



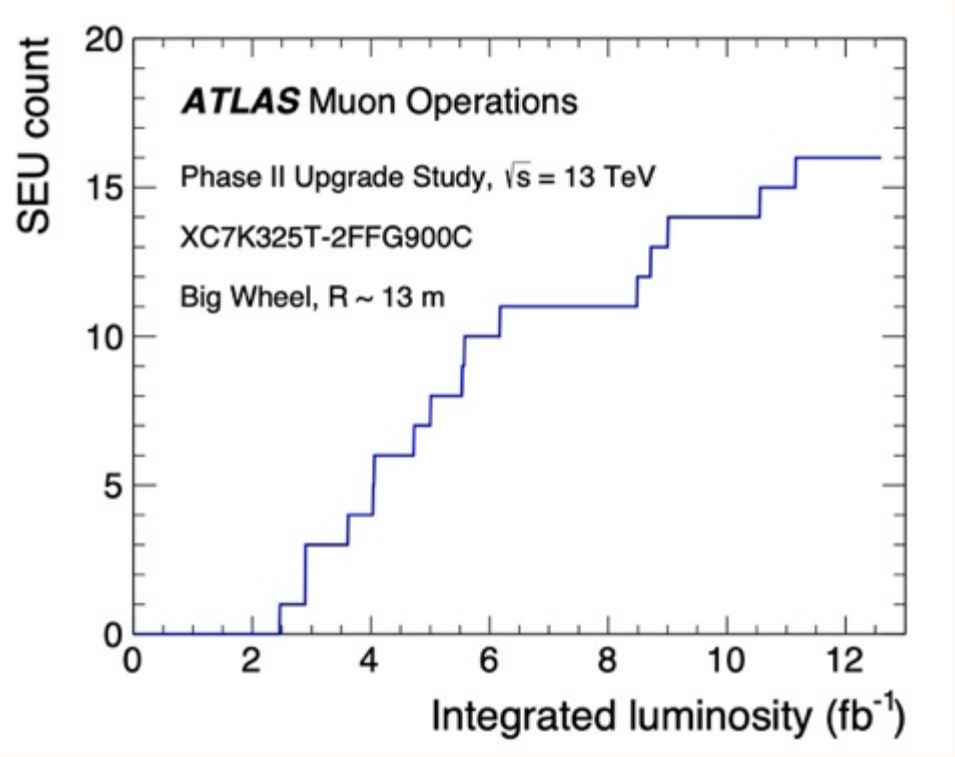
Layout Design

Schematics



Front Panel Design

SEU Counts at Front-end Board in HL-LHC



SEU frequency in ONE FPGA
Run2 : 1 time per ~10 hours (result)
HL-LHC : 1 time per ~3 hours (Calc.)

- SEU count test for a FPGA
  at the location of Front-end board in Run 2(2018)
  [Link]ATLAS public results
- Peak Luminosity in HL-LHC

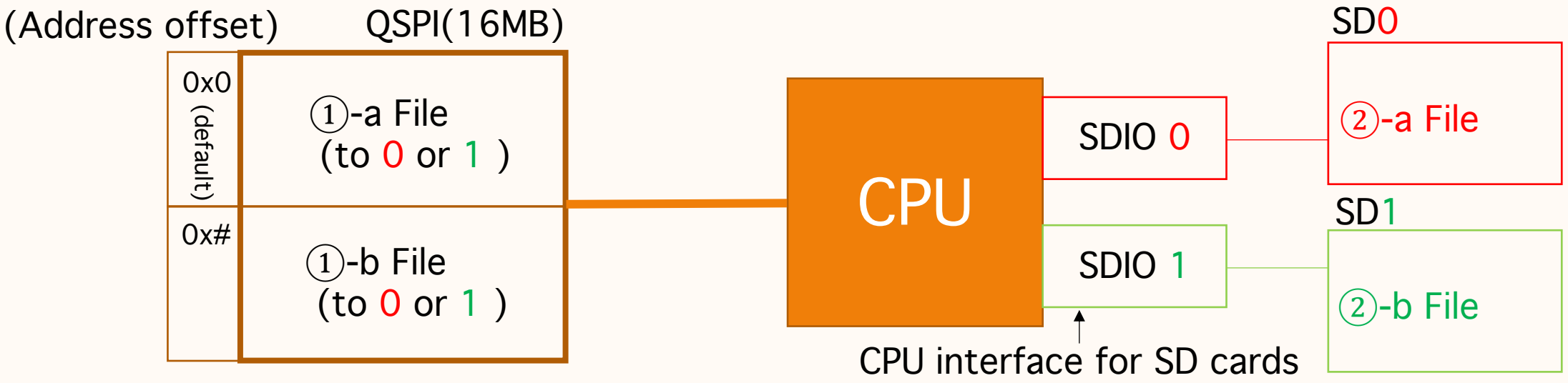
  → $7.5\times10^{34}cm^{-2}s^{-1}$
- Calculated frequency of SEU in FPGAs of all Front-end boards in HL-LHC ATLAS cavern

  → Approximately 1 time per 10 seconds
- [note] In this time, SEM(Soft Error Mitigation) in the FPGA automatically recovered the function of the FPGA on the Front-end Board.
- Because ATLAS cavern in HL-LHC will be in higher radiation area than in Run 2, unrecoverable SEU errors which will be corrected by the JATHub are considered even though the SEM is available.

JATHub Boot System with 2 SDs & QSPI

- Boot Files for Zynq SoC can be divided into ①, ②
  ① Initial Boot File (~0.5 MB)、② Second Boot File (~25 MB)
  - ① Program the same 2 Initial Boot Files (①-a, ①-b) into QSPI, attaching offset value for each
  - ② Store the same 2 Second Boot Files (②-a, ②-b) into each SD card(SD0, SD1)

- By default, when CPU is powered on, CPU reads ①-a File with the smallest offset value at first
  - According to the read pointer (0/1) written in ①-a File (eg. SD0), CPU reads ② File (eg. ②-a)



System of Boot Redundancy on the JATHub

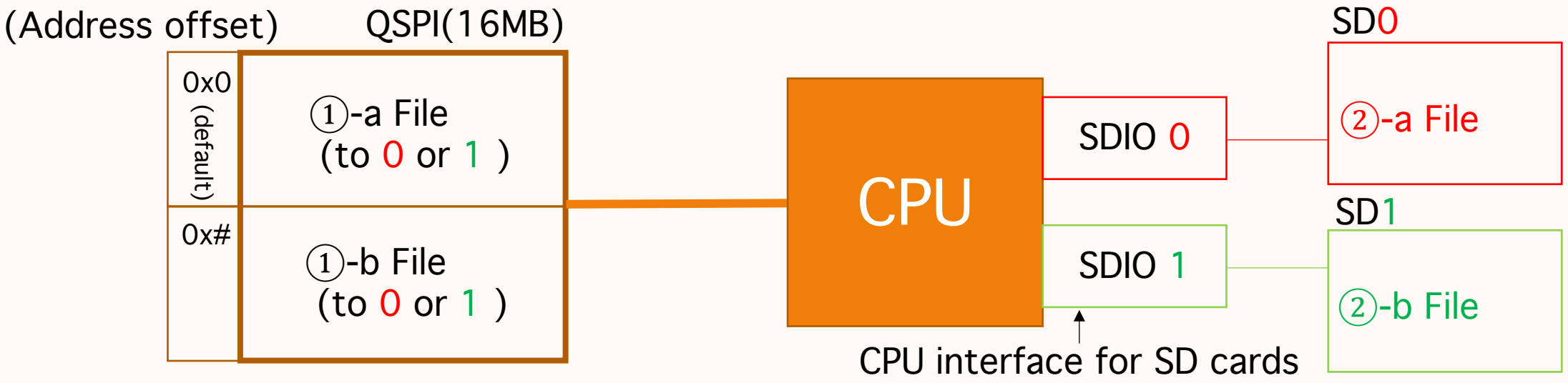
- When CPU fails to read ①-a File

  → CPU will automatically start reading ①-b File
  ("Boot header searcher" will find the file with the second smallest offset value)

- When CPU fails to read ②-a File

  - Remotely, change the read pointer to SD1 in ①-a File, reprogram ①-a File in QSPI

  → CPU will read ②-b File in SD 1, after CPU loads the changed ①-a File



For Your Reference

1, ATLAS Collaboration, Technical Design Report for the Phase-II Upgrade of the ATLAS Muon Spectrometer, Tech. Rep. CERN-LHCC-2017-017. ATLAS-TDR-026, CERN, Geneva (Sep 2017) Link