CMS DAQ System Design with Zynq MPSoC for Phase-2

Petr Žejdl

on behalf of

CMS-DAQ group

5 October 2023

3nd SoC Workshop

DTH-P2 Board

Acknowledgements: ***D. Gigi***

Outline

- CMS Central DAQ Hardware for Phase-2 Upgrade
  - Prototypes
  - Current status
  - Zynz MPSoC
  - Network Booting
  - Graceful Shutdown
- Summary

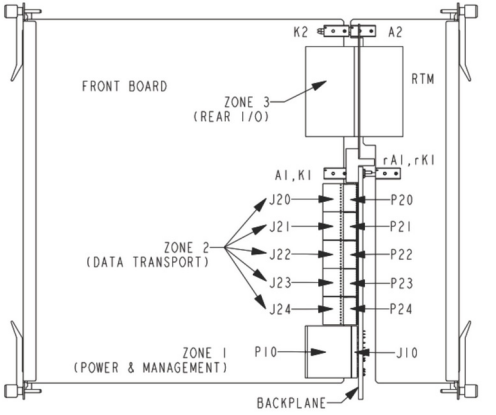
Introduction with some Numbers

	Run 2 & Run 3	Phase 2 (Run 4)	Factor
L1 rate	100 kHz	750 kHz	7.5
Event size	2 MB (design) (1.4 MB measured in Run 2)	~8.4 MB	~4.2
Event Network	1.6 Tb/s	51 Tb/s	~32
HLT Computing	0.7 MHS06	37 MHS06	53
Storage throuhput pp **HI**	2 GB/s 12 GB/s	51 GB/s 51 GB/s	26.0 4.3
Storage capacity	0.3 PB	3.3 PB	11

x32 higher data throughput

CMS ATCA Crates in Phase-2 (HL-LHC) Upgrade

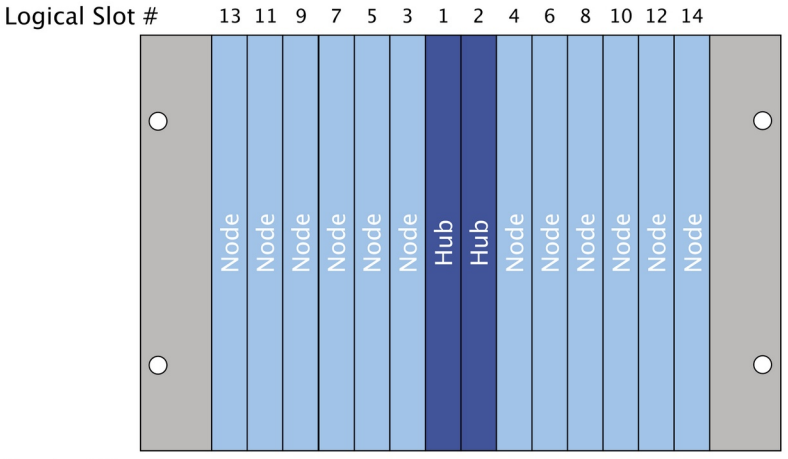
- CMS back-end electronics will be in ATCA crate(s)
  - CMS will use **Schroff** ATCA crate with dual-star backplane
  - About 150 crates hosting approx. 1300 back-end boards
- ATCA imposes design rules and requirements
  - Board consists of **Front Board** and optional **Rear Transition Module (RTM)** [1]
  - Network configuration is obtained via DHCP protocol
    - Based on geographical location identifiers called **Client IDs** [2]

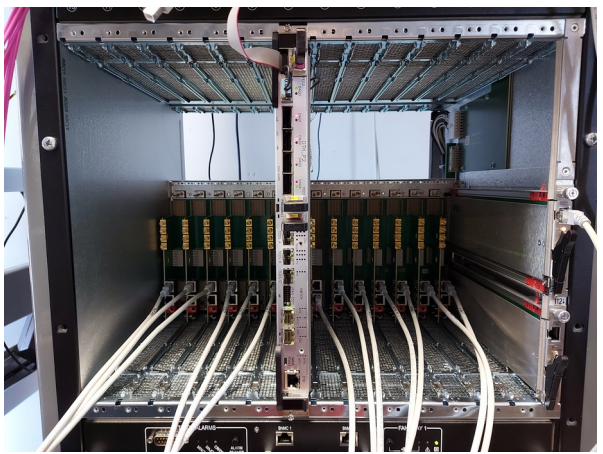


- References
  - [1] PICMG, *"Advanced TCA base specification: Advanced TCA"*
  - [2] PICMG, *"HPM.3, DHCP-Assigned Platform Management Parameters Specification"*

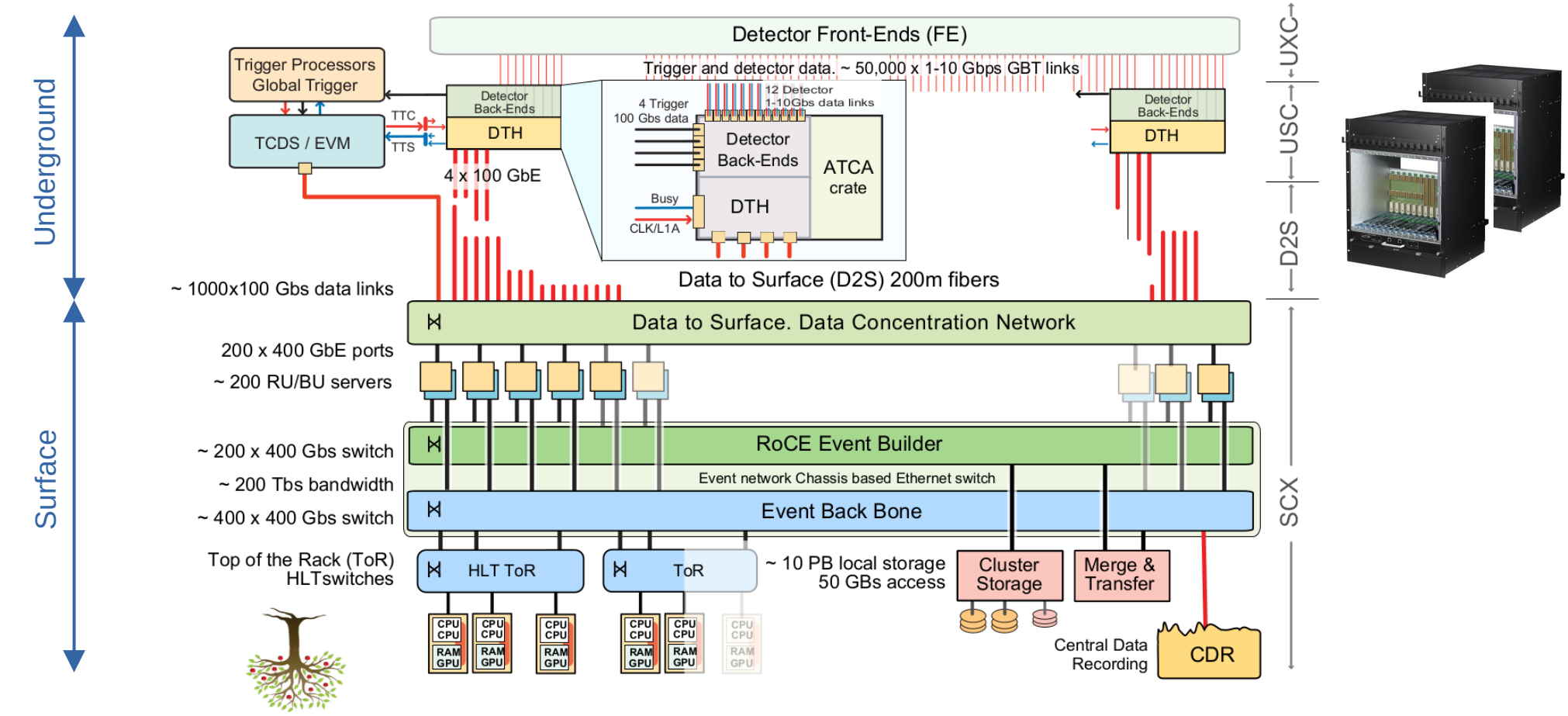
ATCA Crate + DAQ and Timing HUB (DTH-400)

- CMS ATCA Crate
  - 12 **Node** slots, 2 **HUB** slots
    - Each HUB slot has connections to 12 Node slots
- DAQ and Timing HUB (**DTH-400**)
  - Custom board designed by central DAQ of CMS
  - ATCA HUB functionality
    - Provides Gigabit Ethernet connectivity for ATCA crate
  - DAQ functionality
    - Optical readout links from back-end boards
    - 400 Gbit/s bandwidth towards DAQ using TCP/IP streams
  - Timing functionality
    - LHC clock distribution
    - Connection to CMS Trigger and Timing Control and Distribution System (TCDS)
  - About 150 boards foreseen

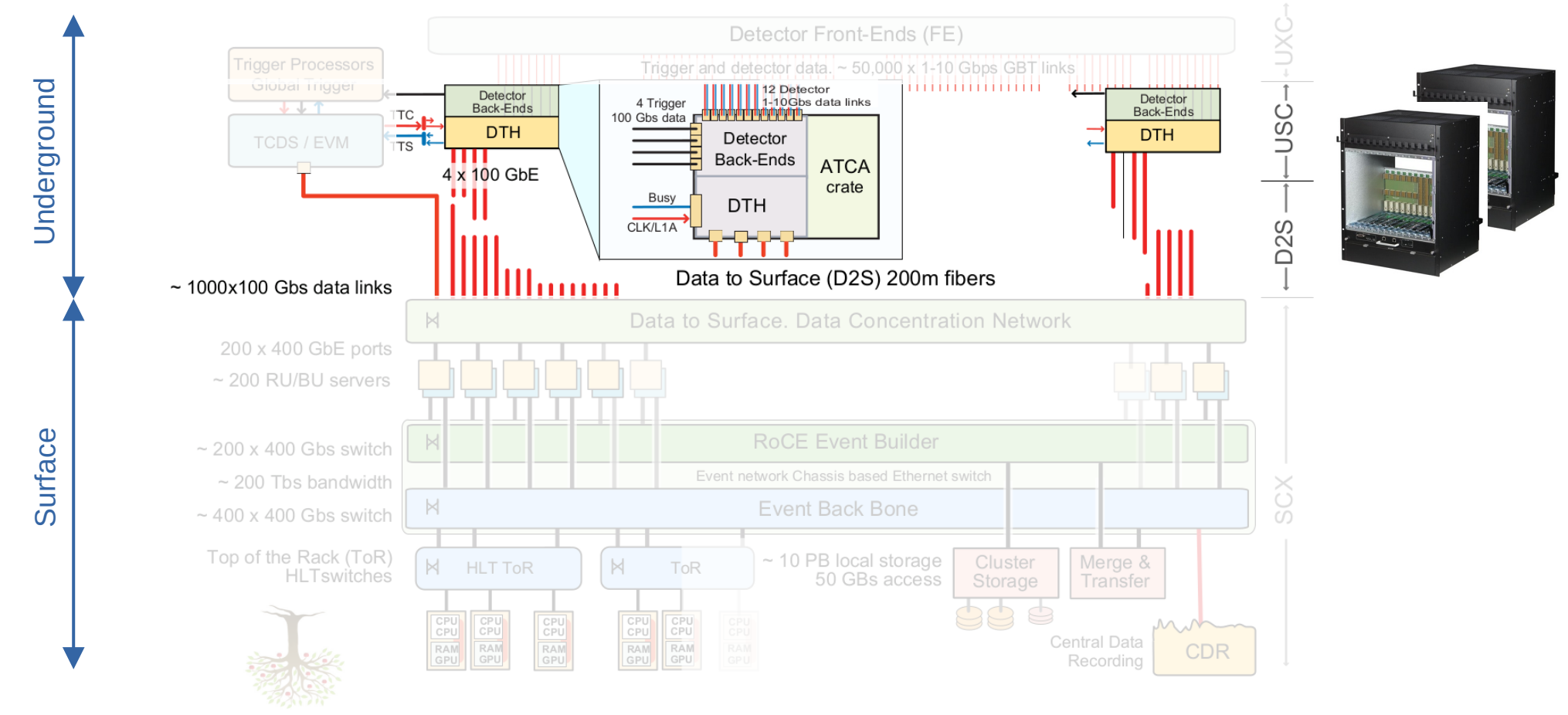




CMS DAQ System for Phase-2 (HL-LHC) Upgrade



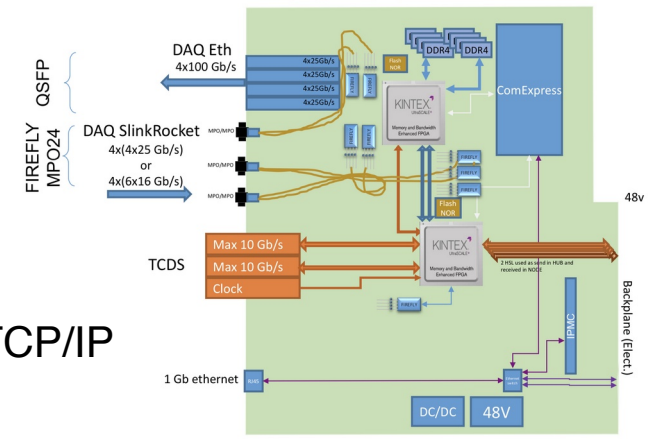
CMS DAQ System for Phase-2 (HL-LHC) Upgrade

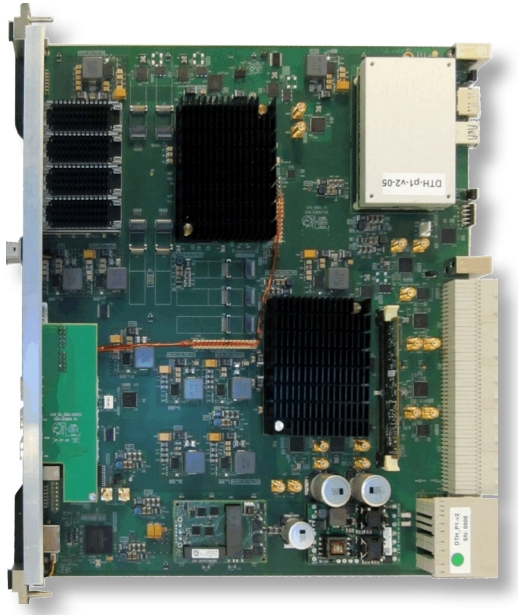


Prototyping

DTH Prototype 1

- Introduced 2nd SoC Workshop in 2021 in CMS Overview by Frans Meijers
- DTH P1 v1
  - With Hybrid Memory Cube (HMC)
    - 400 Gb/s capable
    - Was **discontinued** by Micron!
- DTH P1 v2
  - DAQ readout up to 200 Gb/s over TCP/IP
    - Limited by DDR speed
  - FPGA Kintex UltraScale 15P
  - Board controller is **COM Express** (x86 Computer-On-Module)
  - No Ethernet connectivity for node slots
- Next DTH prototype with Zynq MPSoC





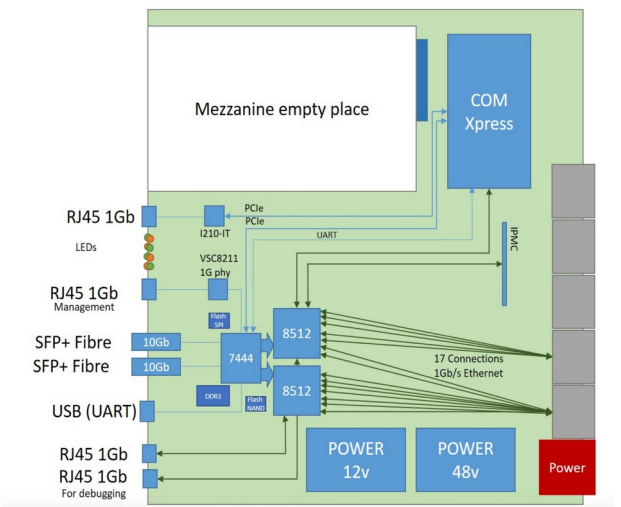
DTH P1

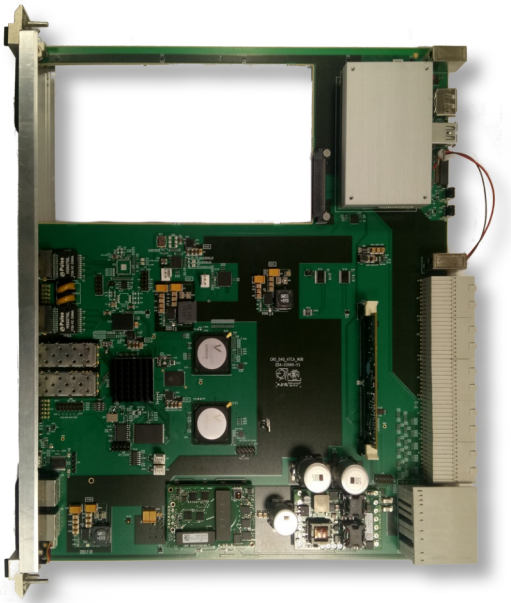
DTH Ethernet Switch Prototype

- Managed Ethernet switch
  - Providing Gigabit Ethernet connectivity for node slots, shelf manager, and IPMC
  - Two 10 GbE uplinks for redundancy
  - ~~Vitesse~~ ~~Microsemi~~ Microchip VSC7444 Ethernet switch ASIC
  - Board controller is **COM Express** (x86 Computer-On-Module)

VSC7444 Switch ASIC

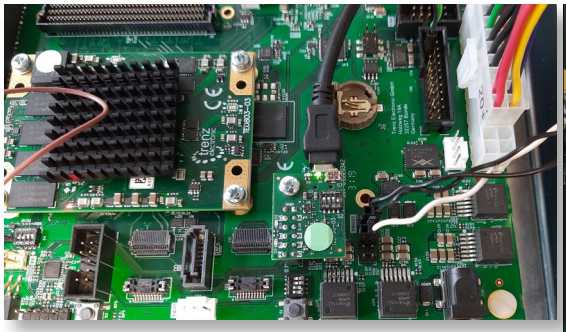
- **10/1 Gb/s Ethernet switch**
- **500 MHz 32-bit MIPS CPU**
- **Network on Chip**
- 1GB SDRAM (on PCB)
- Running U-Boot + Linux kernel 4.19
- Busybox, Buildroot based



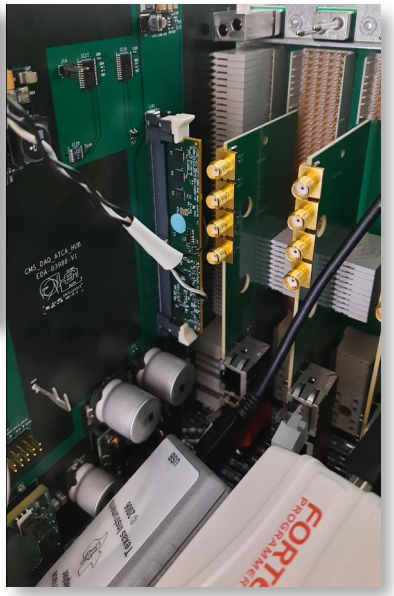


Zynq MPSoC Prototyping

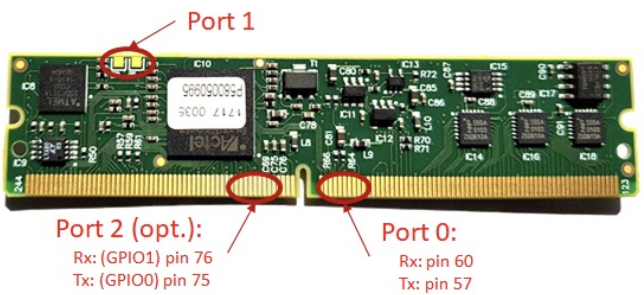
- COM Express on DTH P1
- Using ZCU102 and Trenz for prototyping
- Connected to CERN IPMC
  - Used for obtaining geographical address for DHCP Client ID
  - Via external wires using edge pins on IPMC



UART from Trenz baseboard (XMOD)



UART to CERN IPMC in ATCA

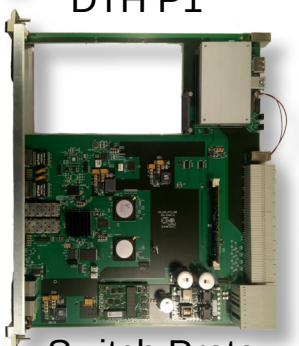


UART availability on CERN IPMC

Current Status

The latest prototype: DTH-P2

DTH P1

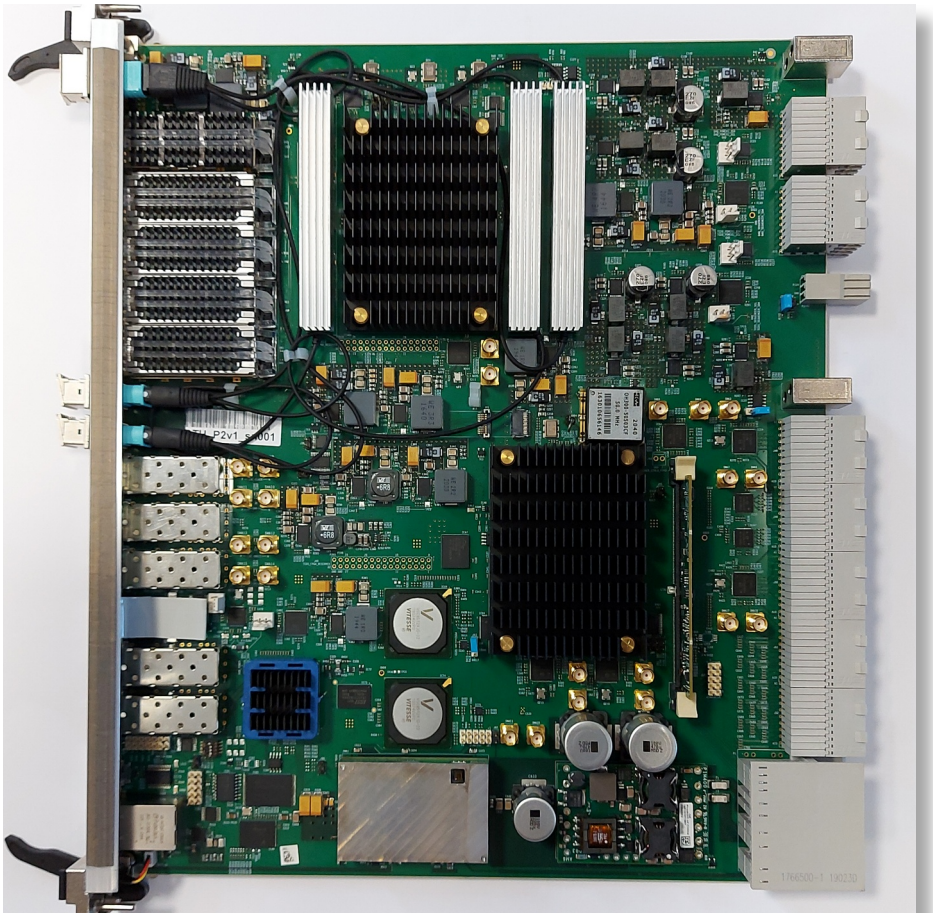


Switch Proto

Zynq MPSoC
on Module

FPGA upgraded from KU15P to VU35P

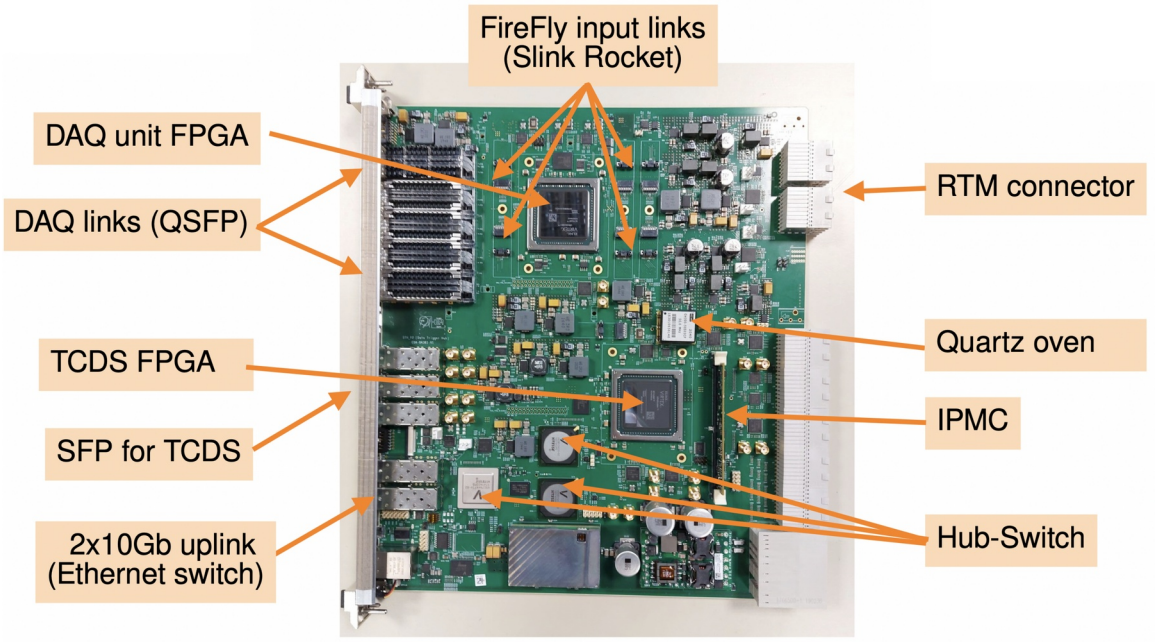
- 8GB of High Bandwidth Memory (HBM) for Network buffers

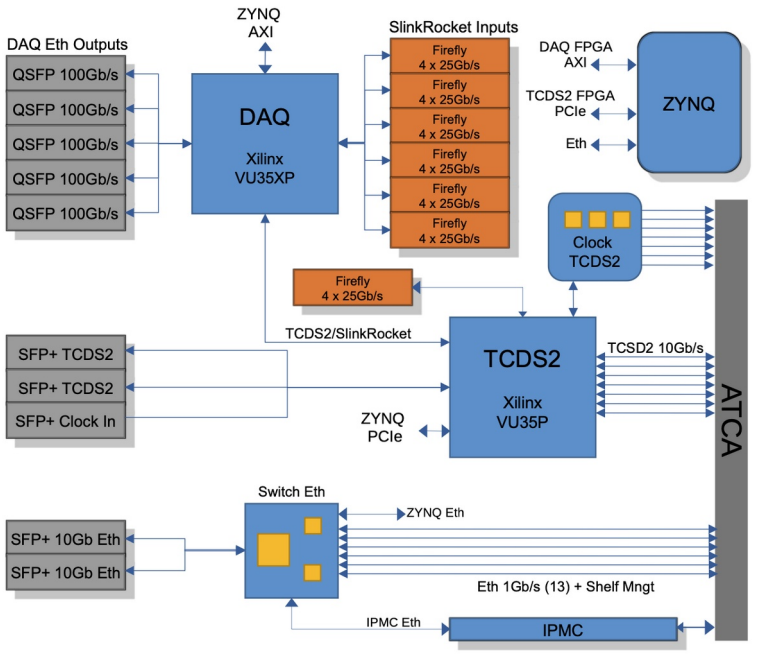


DTH P2

The latest prototype: DTH-P2

- Specified for sustained throughput 400 Gb/s over TCP/IP
- 24x 25 Gb/s input links (FireFly) from back-end boards (oversubscribed)
- 5x 100 GbE output links (QSFP28) towards DAQ over TCP/IP streams



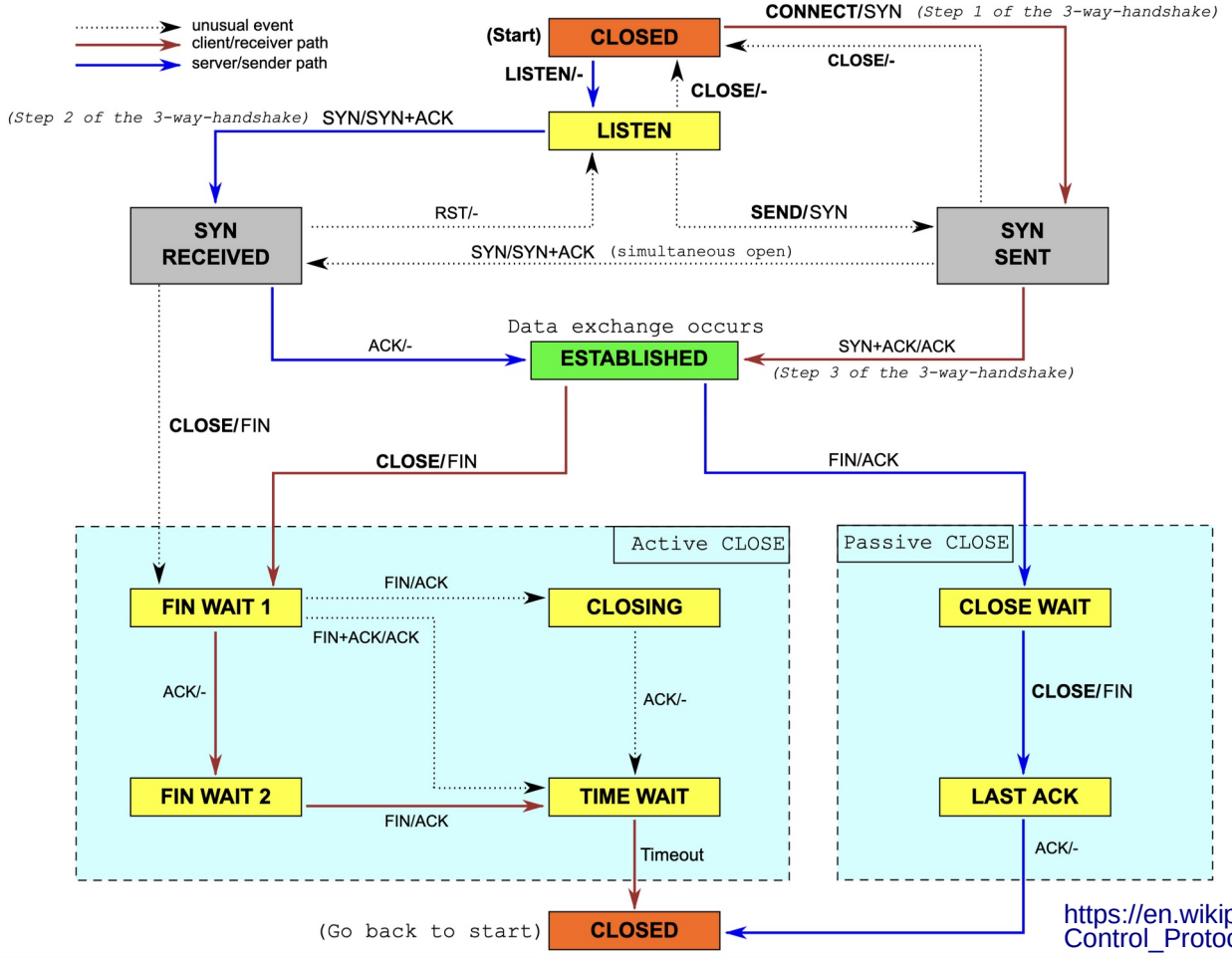


DTH Functionality

- Timing functionality
  - Connection to CMS Trigger and Timing Control and Distribution System (TCDS)
    - LHC high precision clock distribution
    - Distribution of TTC (trigger) signals and collection of back-end status (TTS) via backplane
  - More information about TCDS for Phase-2 in paper for RT2022 by Jeroen Hageman
- DAQ functionality
  - Optical readout links from back-end boards (over front-panel)
  - Orbit aggregation: Event fragments aggregated by orbit (~300 kB)
    - Larger blocks/packets have lower network transport and processing overhead
  - Aggregated bandwidth of 400 Gb/s over TCP/IP streams towards central DAQ
  - High Bandwidth Memory (HBM) used for TCP buffer with theoretical bandwidth 409 GB/s
- Extension: DAQ-800 board is being developed
  - Node board (not HUB), 800 Gb/s aggregated bandwidth, 2x DAQ FPGAs
  - For subsystems with larger bandwidth requirements

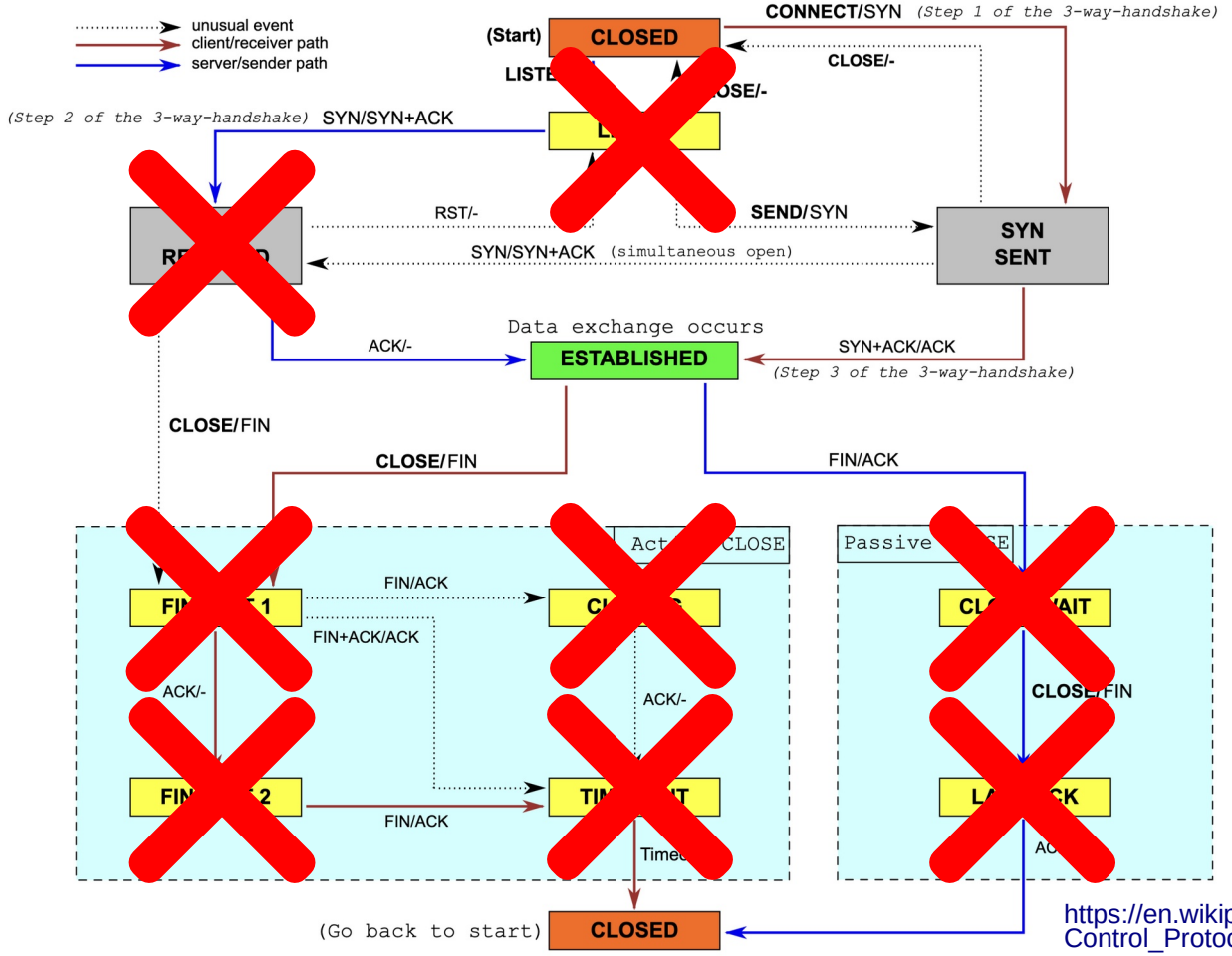
TCP/IP for DTH

TCP State Diagram



https://en.wikipedia.org/wiki/Transmission_Control_Protocol

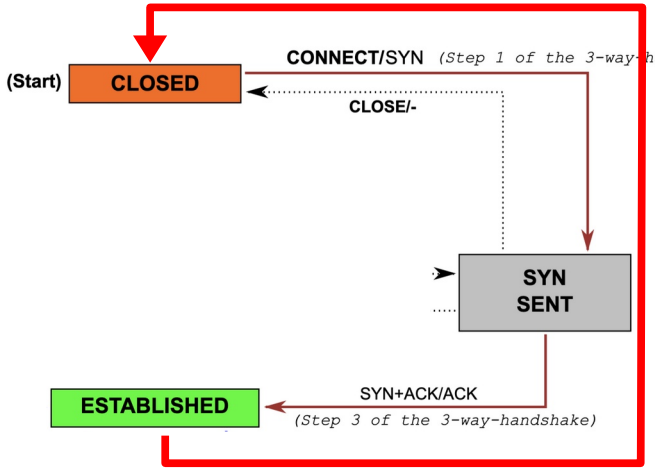
TCP State Diagram Simplified



https://en.wikipedia.org/wiki/Transmission_Control_Protocol

TCP State Diagram Final

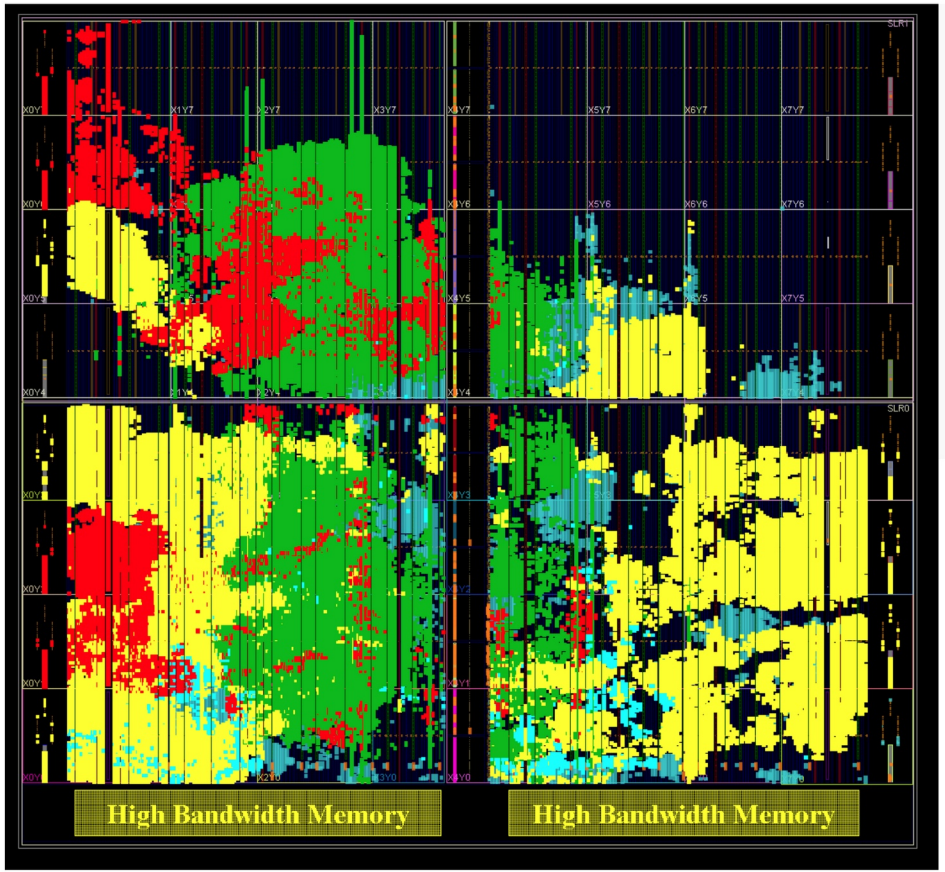
- Simplification to the TCP/IP protocol (for feasible FPGA implementation)
  - Implemented client part only: FPGA opens connection to PC
  - Implemented sender part only: Data goes from FPGA to PC
    - Only acknowledgements go back (part of the protocol)
- All simplifications → compatible with RFC793
  - Using standard Linux TCP/IP stack for receiving
  - **Reliable loss-less transmission**
  - Built-in **flow-control** that follows the receiver buffer occupancy
- In production since Run-2, running at 10 Gb/s
  - References:
    10 Gbps TCP/IP streams from the FPGA for High Energy Physics



Final State Diagram

DAQ FPGA Resource Utilization

Virtex UltraScale 35P



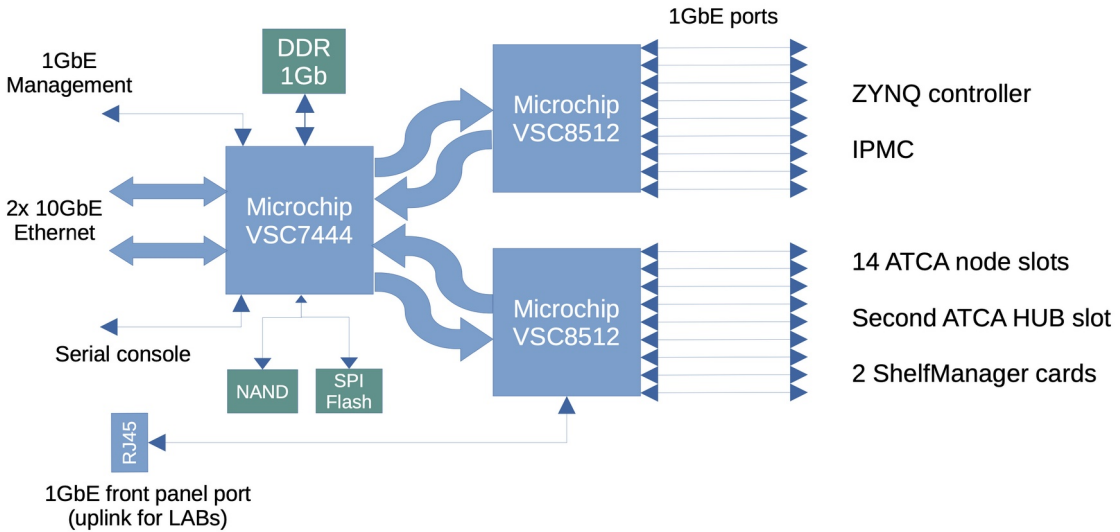
- Back-end Inputs (x16), Emulator, Orbit aggregation
- TCP/IP logic (x16), HBM read
- 100 Gigabit Ethernet (4x 100 GbE Interfaces)
- Rest: AXI, I2C, JTAG, ...

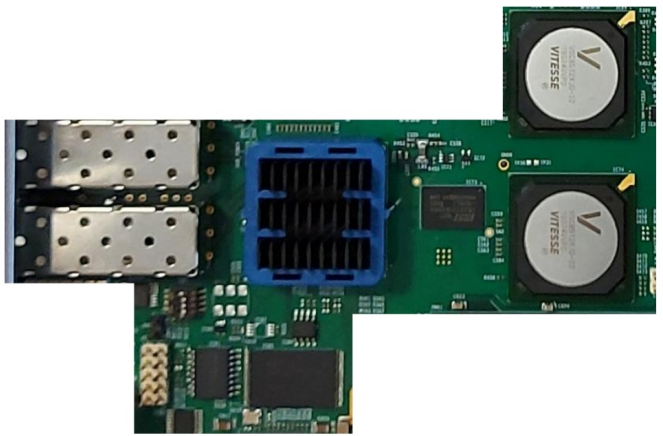
Resource	Utilization	Available	Utilization %
LUT	220552	871680	25.30
LUTRAM	4048	403200	1.00
FF	349070	1743360	20.02
BRAM	856	1344	63.69
URAM	96	640	15.00
DSP	10	5952	0.17
IO	133	416	31.97
GT	33	64	51.56
BUFG	76	672	11.31
MMCM	3	8	37.50
PLL	1	16	6.25

HUB Functionality: Managed Ethernet Switch

- Microchip VSC7444 Switch ASIC with VSC8512 copper PHY
  - 1 Gigabit Ethernet for Node slots, Shelf manager, Zynq and IPMC
  - 2x SFP+ 10 Gigabit Ethernet uplink
  - 1x RJ45 Gigabit Ethernet uplink for LABs
  - Plan to use managed Switch OS software from Microchip

Network-on-Chip



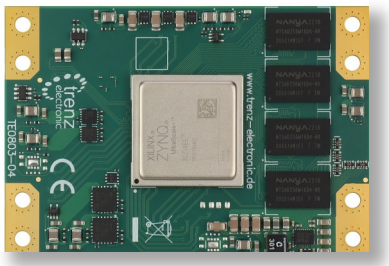


10GbE Switch part on PCB

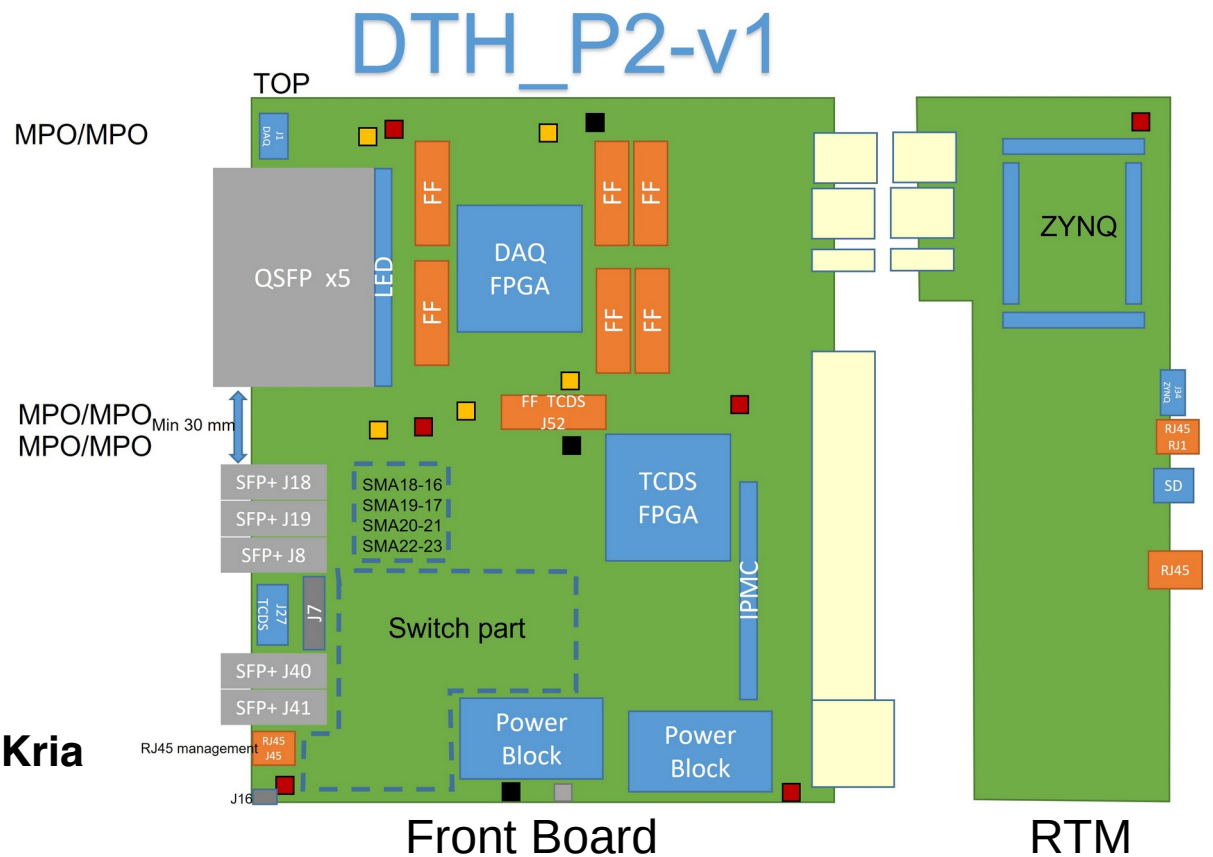
Where is Zynz MPSoC?

RTM with Zynq MPSoC

- Zynq on RTM (Rear Transition Module)
  - Using add-on **module** (SoM**)**
  - Trenz TE0803-04-4GE21-L
    - XCZU4EG-2SFVC784E
    - 4 GB DDR4

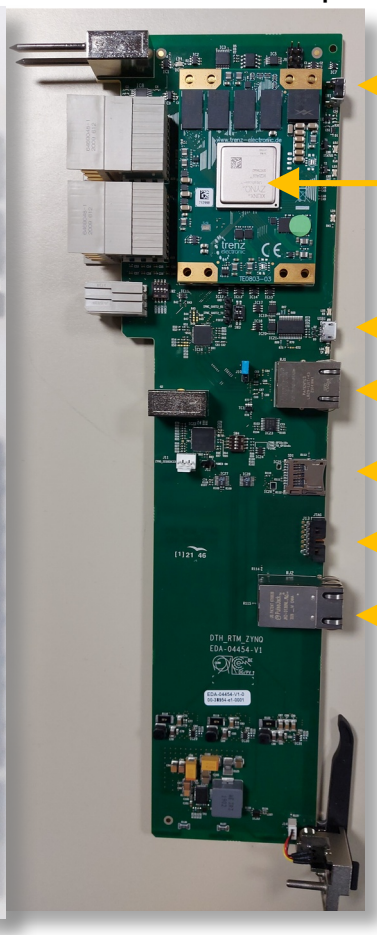


- Flexibility for the future
  - **Can change the module**
  - **Can change the vendor, e.g. Kria**



DTH P2 + RTM

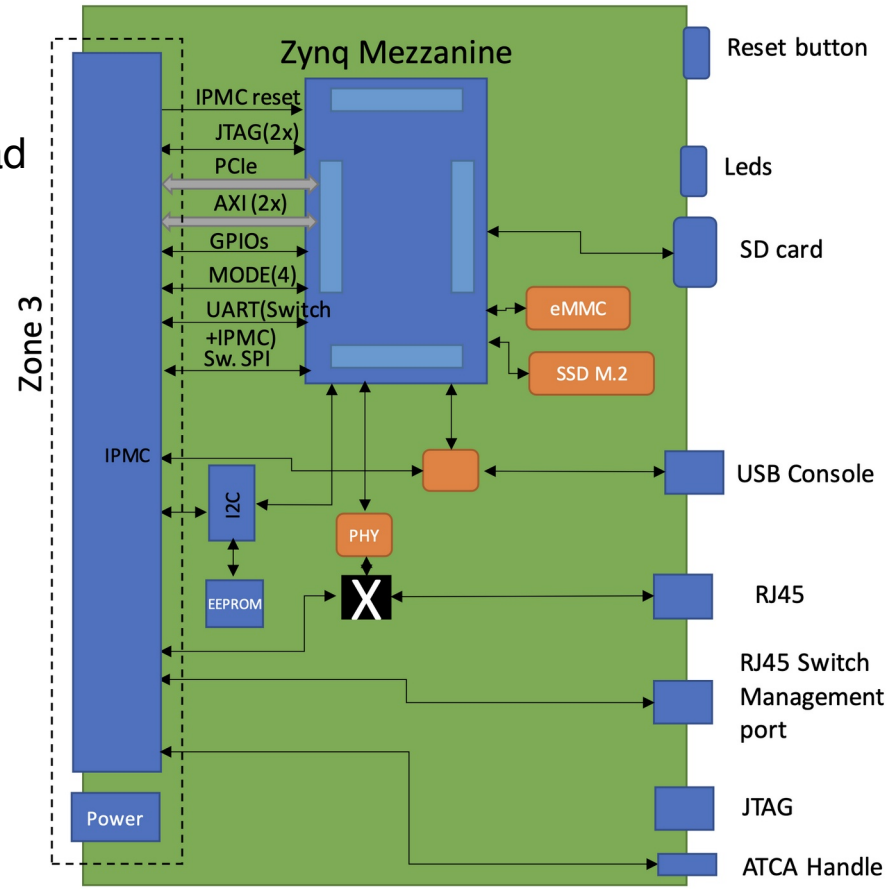
RTM without front panel



- Reset button
- Trenz Module
- Zynq Console (serial over USB)
- RJ45 Zynq Ethernet (Optional)
- microSD card
- JTAG
- RJ45 Switch Management

RTM with Zynq MPSoC

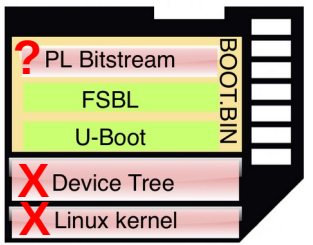
- IPMC (on DTH) connected to Zynq
  - Zynq boot mode select, reset line
  - Dedicated serial line implementing ATCA Payload Interface
    - For obtaining geographical location used in network configuration (DHCP ClientID)
    - Reference:
      Ralf Spiwoks - SoC IG - 16 February 2021
- DAQ and TCDS FPGA connected to Zynq
  - AXI over Chip2Chip / Aurora bus
  - 2x JTAG connected
    - Xilinx Virtual Cable (XVC) running in Linux OS, allows remote debugging over Ethernet
- Version 2 is being tested
  - Contains eMMC and SSD M.2



RTM (v2) Block Diagram

CMS DAQ Design Choices for SoC(s)

- Geographically Aware Network Configuration (as specified by ATCA specs)
  - Geographical location identifier **DHCP Client ID** is used to obtain network configuration
  - Client ID contains shelf address and slot number that are obtained from IPMC
  - Benefit: Consistent IP address and host name, no dependency on board physical address
  - **More information** in **Marc Dobson's** talk on Friday
- Full Network Boot
  - **Minimum files** on SD card of Flash memory, read only access
  - Linux kernel and firmware(s) are fetched from network servers
  - Linux root file system is mounted over NFS
  - Benefits:
    - Network **servers are available independently** od SoC (e.g. when SoC is down or crashed)
    - Easy to deploy/rollback new firmware versions or configurations
    - Large software installations and/or OS updates possible to do quickly on servers
  - Reference: "CMS DAQ ... Design Considerations... in ATCA Crates" - 2nd SoC - 2021

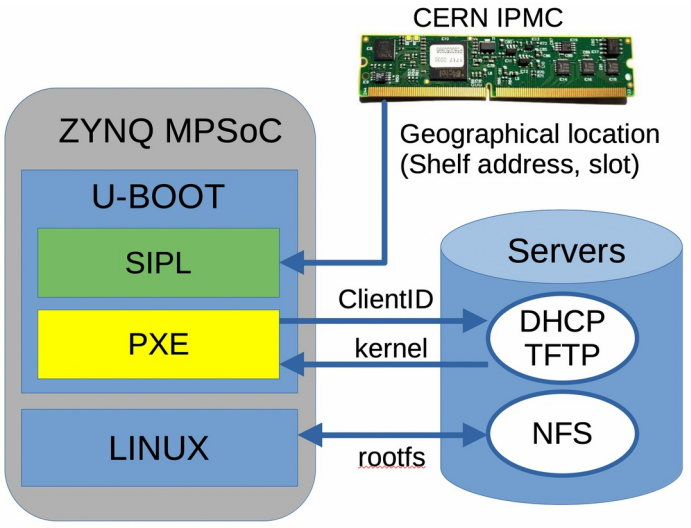


Geographically Aware Network Configuration

- U-Boot is patched with
  - SIPL: Serial Interface Protocol Lite for exchanging information between Zynq and IPMC
    - **Developed by ATLAS L1CT team**
    - https://gitlab.cern.ch/soc/u-boot-sipl
  - DHCP Client ID: Support for Client ID in DHCP, PXE commands in U-Boot
    - **Developed by CMS DAQ team**
    - https://gitlab.cern.ch/hardware/zynq/u-boot-xlnx-ipmc/-/tree/clientid
- Patches are available as Petalinux template
  - https://gitlab.cern.ch/soc/petalinux-template/-/tree/master/
  - Used in tutorials:
    - Tutorial 1: ***"Building Linux Boot Files Using Templates for Multiple SoC Projects"*** by Giulio Muscatello
    - Tutorial 2: ***"Using GitlabCI Parallel Builds for Multi-board PetaLinux Projects"*** by Kareen Arutjunjan

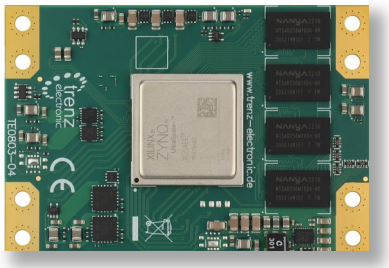
Full Network Boot (Simplified)

- U-Boot
  - SIPL contacts IPMC and forms Client ID
  - PXE
    - Uses Client ID to configure network via DHCP protocol
    - Loads Linux kernel from TFTP server and starts booting
- Linux kernel
  - Uses internal DHCP client to configure network and NFS
  - Mounts root file system over NFS
  - Note:
    - Unfortunately Client ID implementation is broken in the kernel
    - Dnsmasq DHCP server is used as a temporary workaround
      - Remembers MAC address from where ClientID came
      - Then it replies to kernel's DHCP request

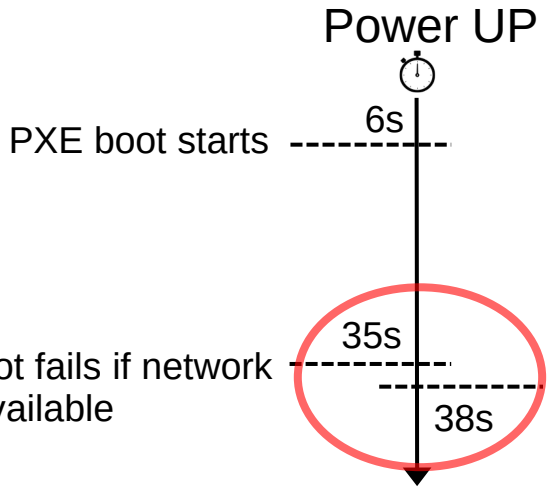


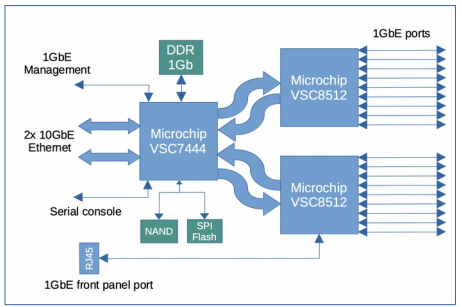
Simplified booting sequence

Network Availability after Power UP



ZYNQ MPSoC on Trenz Module





Network switch ASIC with embedded CPU providing crate Ethernet connectivity

- First PXE boot fails after power up because network is not available in time
  - The default U-BOOT script is stops executing and booting “hangs”
- Failover mechanism added with *boot.scr* script on microSD card
  - Seamless integration, no changes required to the default U-BOOT script
  - Check for network availability added (with timeout)
  - If booting fails the failover mechanism will reset the board
  - *Similar mechanism may be necessary for every network-booted SoC in the ATCA crate*

Zynq MPSoC Graceful Shutdown

- Extracting ATCA board by pulling the handle
  - Pulling activates the hot swap switch
  - IPMC interrupts the power
  - Zynq is forcibly shut down …
- Simple extension to graceful shutdown
  - Important for un-mounting file systems, etc.
  - Two wires between Zynq and IPMC
    - **zynq_shutdown_request**
      - External interrupt triggers ZYNQ shutdown sequence
    - **zynq_shutdown_ack**
      - Set when shutdown completed, IPMC waits for this signal before interrupting power
- Functionality already existing in PMU firmware of Zynq MPSoC
  - Tested and works
  - Details in backup slide

Extraction handles with hot swap switch

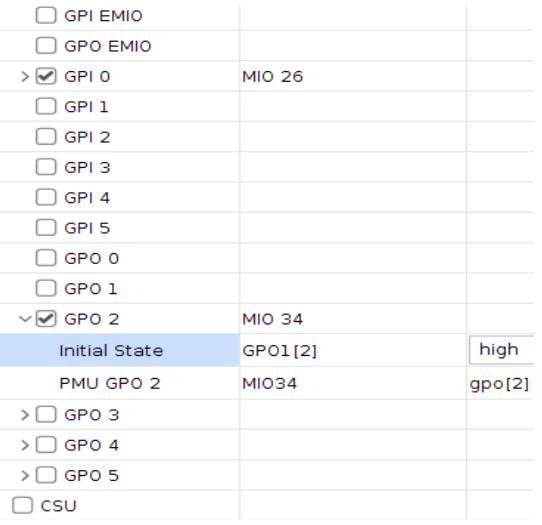
Summary

- DAQ and Timing HUB (DTH) second prototype has been fully tested
  - All necessary board functionalities have been verified
  - Focus on the functionality in DAQ and TCDS firmwares
  - Adding support for more input/output streams (up to 24)
- DAQ-800 board is being developed for bandwidth demanding subsystems
- Zynq MPSoC on Module located on RTM (Rear Transition Module)
  - SoC separated from DTH board, gives maximum flexibility for the future
  - RTM second prototype being developed with eMMC and SSD
  - Full network boot implemented
  - The implementation of DHCP ClientID in network configuration is being finalized
    - **Excellent collaboration with Atlas L1CT / Ralf and Giulio, thanks!**
  - Focus is being moved **towards infrastructure and network services** for SoCs at CMS
    - See talks from Kareen Arutjunjan and Marc Dobson

Backup

Zynq MPSoC Graceful Shutdown Implementation

- PMU Firmware has built-in functionality
- MIO pins routed to PMU
  - PMU input issues a shutdown request to the Linux kernel
  - PMU output changes its state after the Linux kernel is shut
- User configuration
  - 6x dedicated MIO inputs available to PMU
  - 6x dedicated MIO outputs available to PMU
  - Final state of the output after shutdown
- Tested with PetaLinux 2021.2 and works



PS I/O Configuration in Vivado

Configuration in *<project-name>/project-spec/meta-user/recipes-bsp/pmu-firmware/pmu-firmware_%.bbappend*

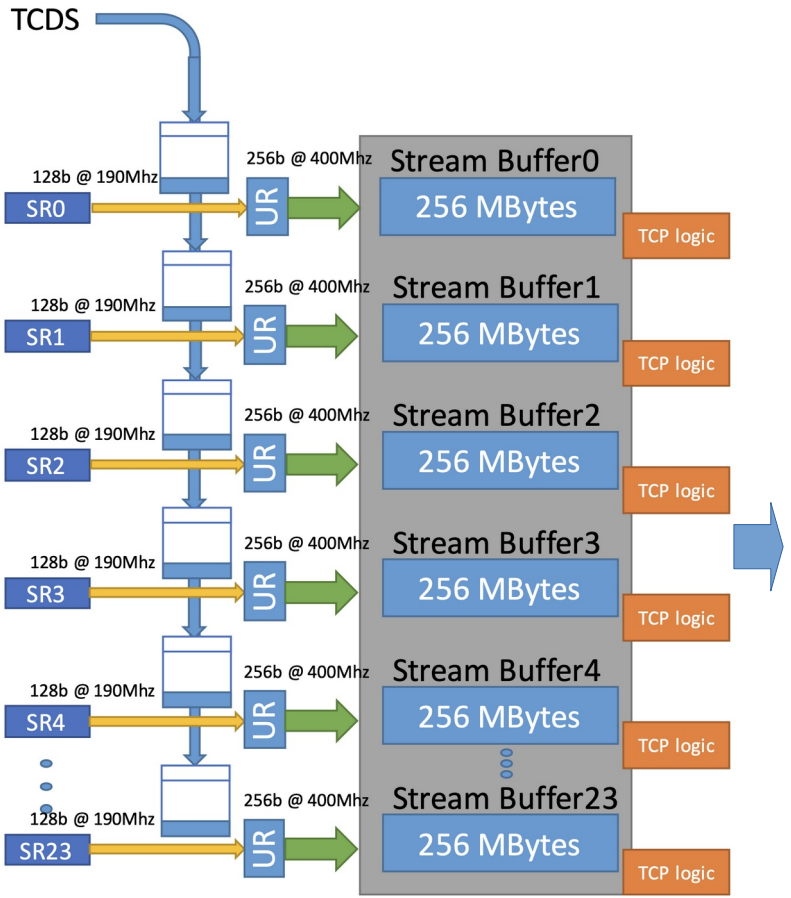
```
YAML_COMPILER_FLAGS_append=" -DPMU_MIO_INPUT_PIN=0 -DBOARD_SHUTDOWN_PIN=2 -DBOARD_SHUTDOWN_PIN_STATE=0"
```

MIO26

MIO34

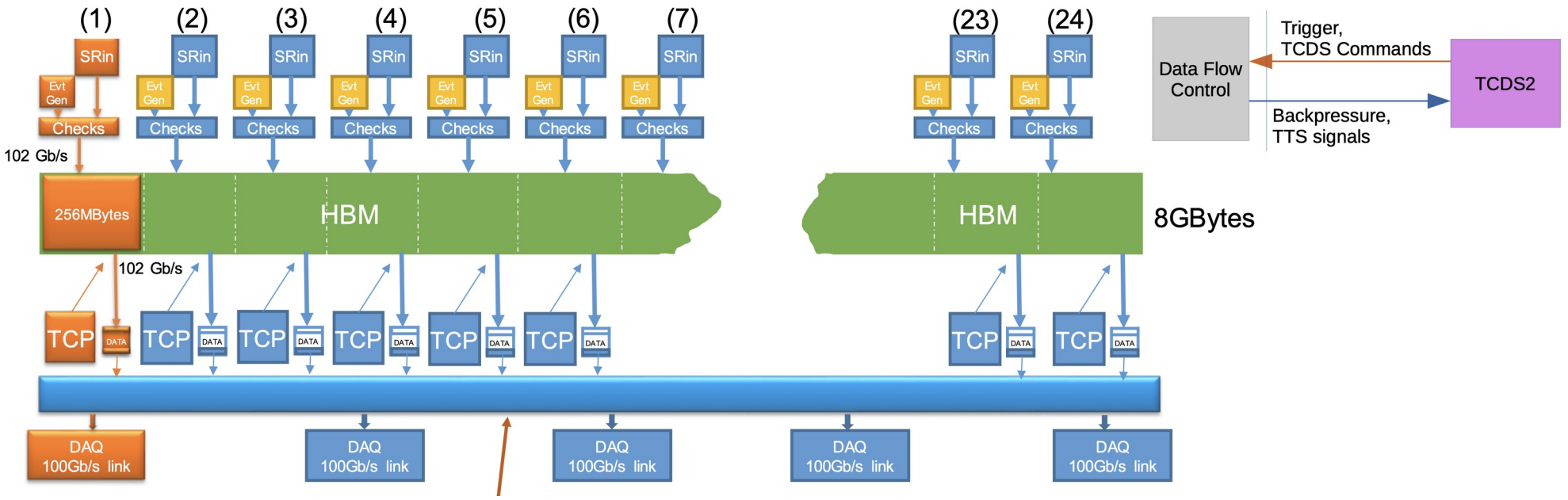
Orbit aggregation functionality in DTH

- Each input link from BE
  - Synchronized and checked with TCDS
  - Multiple fragments are aggregated by orbit
    - ~75 events at 750 kHz
    - Event builder will operate with ~300 kB orbit blocks @ ~10 kHz
- TCP streams
  - 1-24x TCP streams are statically distributed over 1-5x 100 GbE interfaces
  - Stream assignment depends on the bandwidth required by the sub-detector
  - HBM memory used as TCP socket buffer
    - 256 MB per stream



HBM Buffer Structure

24x 25 Gb/s input links from back-end boards



TCP/IP streams are statically distributed over 1-5x 100 GbE interfaces, depending on throughput required by the sub-detector