Photon Control and Data Systems for LCLS

Amedeo Perazzo

representing

Mark Freytag, Gunther Haller, Ryan Herbst, Mike Huffer, Chris O'Grady,
Leonid Sapozhnikov, Eric Siskind, Dave Tarkington, Matt Weaver

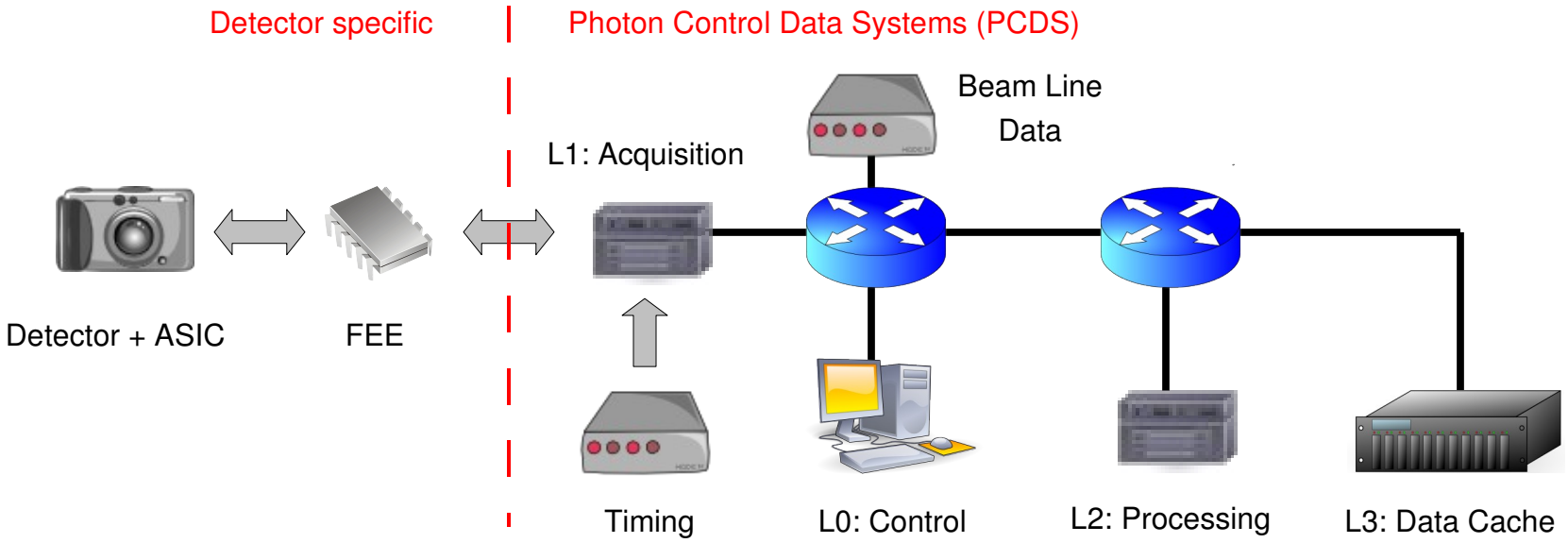
Stanford University/SLAC

XFEL DAQ and Control for Photon Beam Systems Workshop
DESY, March 10th 2008

Contents

- **Data System Architecture**
  - Brief description of the functionality of the various components (6 slides)
- **Control System Architecture**
  - (Extremely) Brief description of the overall control system (1 slide)
- **Networking**
  - Service network organization (2 slides)
  - Control and data traffic network organization (1 slide)
- **Reconfigurable Cluster Element**
  - SLAC custom made board
  - Principal component of the LCLS DAQ system
    - *Common among different experiments*
  - Main focal point of this talk (~12 slides)

Data System Architecture



- **Detector**
  - Experiment specific
  - May be bump-bonded to ASIC or integrated with ASIC
- **Front-End Electronics (FEE)**
  - Provide local configuration registers and state machines
  - Provide ADC if ASIC has analog outputs
  - FEE uses FPGA to transmit to DAQ system

Accelerator Data

- **Timing interface**
  - Timing boards (EVR) receive 120 Hz timing information from accelerator timing generator board (EVG)
    - *1Gb/s optical fibers connect EVG with EVR through custom protocol*
  - EVR distributes timing signals to L1 nodes and FEE
- **Beam Line Data**
  - Time-stamped 120 Hz beam quality data information
  - Information contained in raw Ethernet packets
    - *low latency network*
    - *no UDP or TCP*
  - Used by L1 nodes to veto events

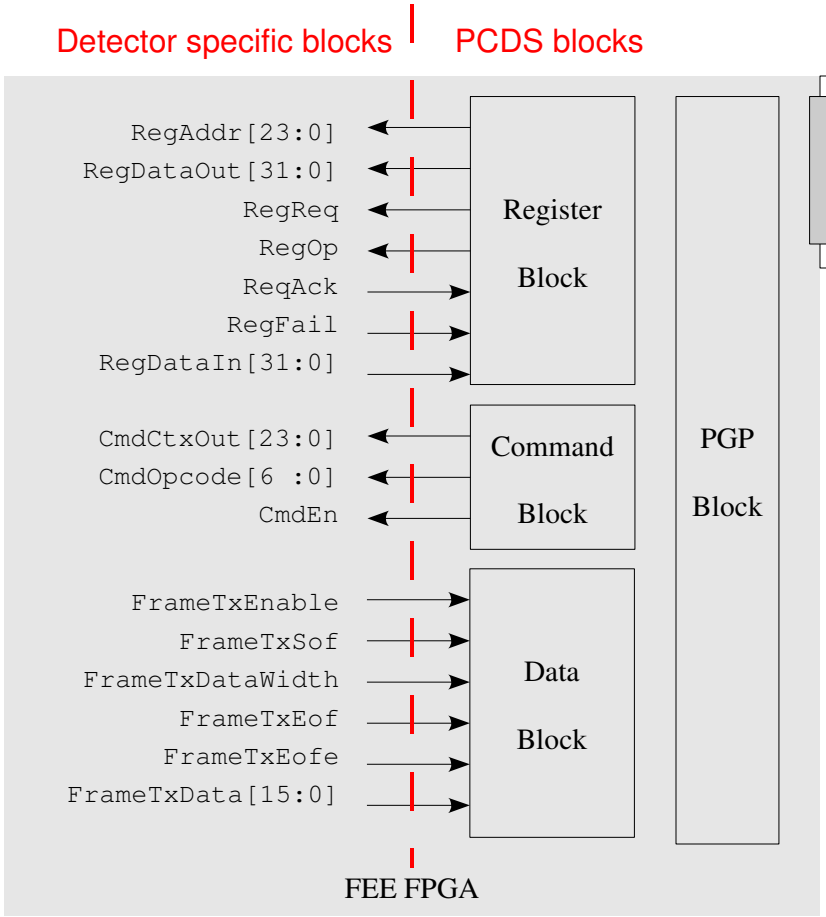
Level 0 Nodes

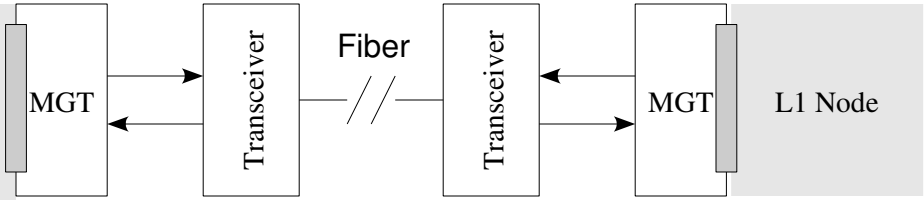
- **Level 0: Control**
  - DAQ operator consoles
- **Provide different functionalities:**
  - Run control
    - *Partition management, data-flow*
  - Detector control
    - *Configuration (modes, biases, thresholds, etc)*
  - Run monitoring
    - *Data quality*
  - Telemetry monitoring
    - *Temperatures, currents, voltages, etc*
- **Manage all L1, L2 and L3 nodes in a given partition (i.e. the set of DAQ nodes used by a specific experiment or test-stand)**

Level 1 Nodes

- **Level 1: Acquisition**
  - Receive 120 Hz timing signals, send trigger to FEE, acquire FEE data
  - Error detection and recovery of the FEE data
  - Control FEE parameters
  - Calibration
    - *Dark image accumulation and averaging*
    - *Transfer curve mapping, gain calculation*
    - *Neighbor pixel cross-talk calculation*
  - Event-build FEE science data with beam-line data
  - Image processing
    - *Pedestal subtraction using calibration constants, cross-talk corrections*
    - *Partial data reduction (compression)*
    - *Rejection using 120 Hz beam-line data*
    - *Processing envisioned both in software and firmware (VHDL)*
  - Send collected data to Level 2 nodes over 10 Gb/s Ethernet

Register Command Data Interface



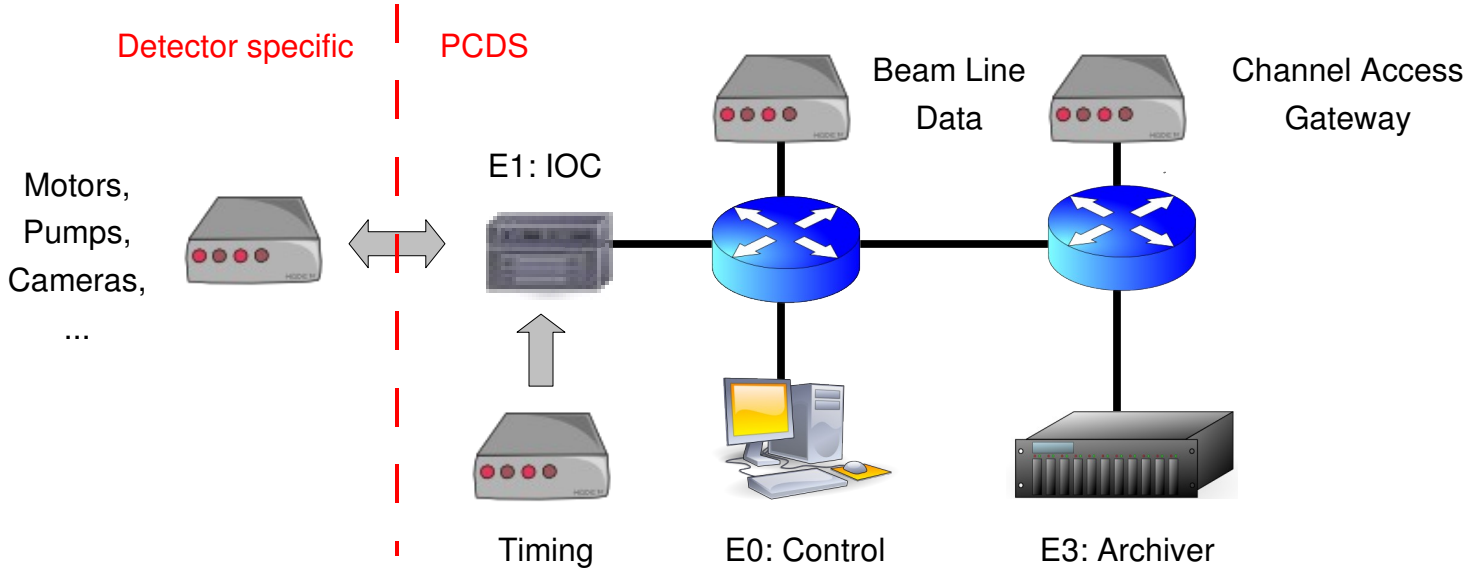


- **Interface defined between FEE and L1**
  - Common interface among different experiments
  - Provide data, command and register interfaces
  - Custom point-to-point protocol (Pretty Good Protocol, PGP) implemented as FPGA IP core
  - FEE FPGA assumed to be Xilinx Virtex-4 FX family with Multi Gigabit Transceivers (MGT)

Level 2 & 3 Nodes

- **Level 2: Processing**
  - High level data processing:
    - *Learn, pattern recognition, sort, classify*
      - e.g. combine 10^5 – 10^7 images into 3D data-set
    - *Alignment, reconstruction*
  - Currently evaluating different ATCA blades for L2 nodes
  - Send processed data to L3 over 10 Gb/s Ethernet
- **Level 3: Data Cache**
  - Provide data storage
    - *Located in server room in experimental hall*
  - Off-line system will transfer data from local cache to tape staging system
    - *Tape staging system located in SLAC central computing facilities*
  - Must be able to buffer up data in local storage during downtimes of staging system
    - *Current requirement is ~4 days of data*

Control System Architecture

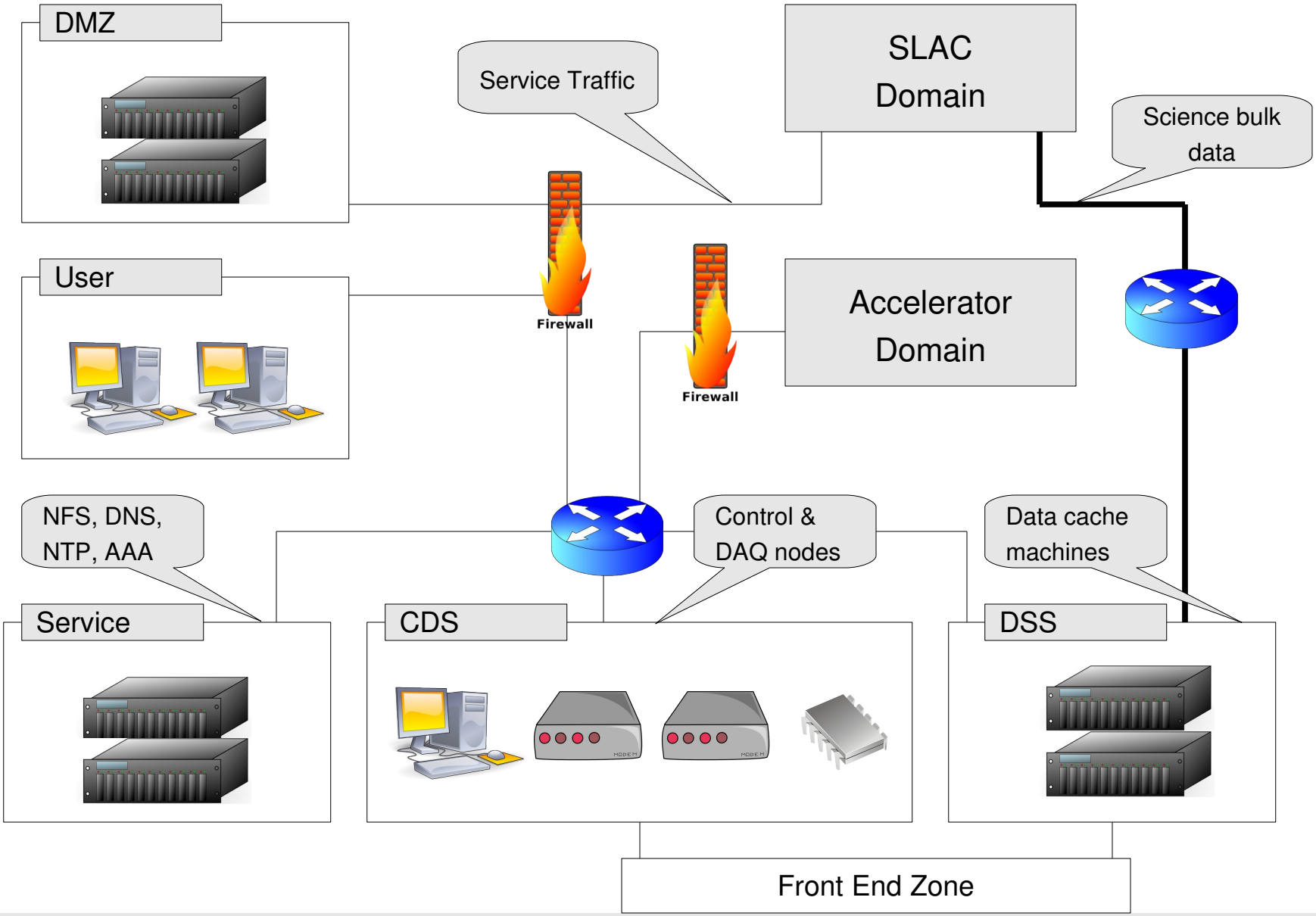


- **Control system is EPICS based**
  - Each experiment control system connects to accelerator controls and to photon-beam controls through a channel access gateway (CAG)
    - *CAG as proxy for external clients and to filter traffic between different systems*
- **Adopted different architectures for the Input-Output Controllers (IOC)**
  - Most IOCs are MVME5500 SBCs (PowerPC) running RTEMS
    - *Plus some cPCI PP410 SBC (x86) running Linux*
    - *Plus some 1U servers (x86) soft IOCs running Linux*

Networking

- **PCDS network organized in 2 zones**
  - Back-end: provides networking services to the PCDS enclave
  - Front-end: control and data acquisition traffic
- **Network organization driven mainly by new DOE security rules**
- **Back End Zone**
  - Must allow Control & DAQ to be operational, for limited amount of time, when connection to SLAC domain is down
  - Divided into five subnets:
    - *DMZ: limited access from SLAC machines*
    - *USER: development and Internet access*
    - *SERVICE: provides NFS, DNS, NTP and AAA services to CDS and DSS*
    - *CDS (Control & DAQ Subnet): service subnet for Control & DAQ nodes*
    - *DSS (Data Storage Subnet): service subnet for the Data Storage machines*
  - Switching based on commercial off-shelf machines
    - *Switches configured to also provide routing and firewall capabilities*

Back End Zone Network Diagram



Networking (II)

- **Front End Zone**
  - Provides the infrastructure for the control and data acquisition traffic
  - Divided into three subnets:
    - *DAQ: science data, partition management, run monitoring and telemetry traffic*
      - connects DAQ operator consoles (L0), readout nodes (L1), processing nodes (L2) and data cache machines (L3)
    - *EPICS: control traffic*
      - connects control operator consoles (E0), IOCs (E1), EPICS archiver (E3) and the channel access gateway
    - *BLD: low latency beam-line data traffic*
  - Switching based on custom Cluster Interconnect Modules (CIM)
    - *Low latency, high-speed ATCA switches*
    - *Provide connectivity between modules inside ATCA chassis and among separate crates*
    - *More on the CIM at the end of this presentation...*

Reconfigurable Cluster Element (I)

- **The RCE is the most interesting among the different Level 1 node types**
  - SLAC custom made ATCA board
- **Based on System On Chip (SOC) Technology**
  - Currently implemented with Xilinx Virtex 4 devices, FX family
    - *Targeting XC4VFX60*
  - Xilinx devices provide
    - *Reconfigurable FPGA fabric*
    - *DSPs (200 for XC4VFX60)*
    - *Generic CPU (2 PowerPCs 405 running at 450 MHz for XC4VFX60)*
    - *TEMAC: Xilinx TriMode Ethernet Hard Cores*
    - *MGT: Xilinx Multi-Gigabit Transceivers 622Mb/s to 6.5Gb/s (16 for XC4VFX60)*
- **Power consumption**
  - About 72 watts for fully populated board
    - *Configuration with 2 RCE per board and 4 flash memory slices per RCE*
    - *~30W per board+RTM, ~15W per FX60 (RCE), ~1.5W per FX20 (slice)*

Reconfigurable Cluster Element (II)

- **FPGA fabric**
  - Interfaces to:
    - *memory subsystems*
    - *JTAG debug port*
    - *custom multi-function display*
    - *various I/O channels*
  - Generic DMA Interface (PIC) designed as set of VHDL IP cores
    - *Up to 16 PIC channels*
  - PIC in conjunction with Multi-Gigabit Transceivers and protocol cores, provide many channels of generic, high speed, serial I/O
    - *10Gb Ethernet*
    - *PGP*
  - PIC in conjunction with TriMode Ethernet Hard Cores also provide commodity network interfaces
    - *1Gb Ethernet*

Reconfigurable Cluster Element (III)

- **System Memory Subsystem**
  - 512 MB of RAM (currently 128 MB)
    - *Memory controller provides 8 GB/s overall throughput*
    - *Uses Micron RLDRAM II*
- **Platform Flash Memory Subsystem**
  - Stores firmware code for FPGA fabric
- **Configuration Flash Memory Subsystem**
  - 128 MB configuration flash
  - Dedicated file system for storing software code and configuration parameters (up to 16 selectable images)
- **Storage Flash Memory Subsystem (optional)**
  - Up to 1TB per RCE persistent storage flash (currently 256GB per RCE)
    - *Low latency/high bandwidth access through I/O channels using PGP*
    - *Uses Samsung K9NBG08 (32 Gb per chip)*

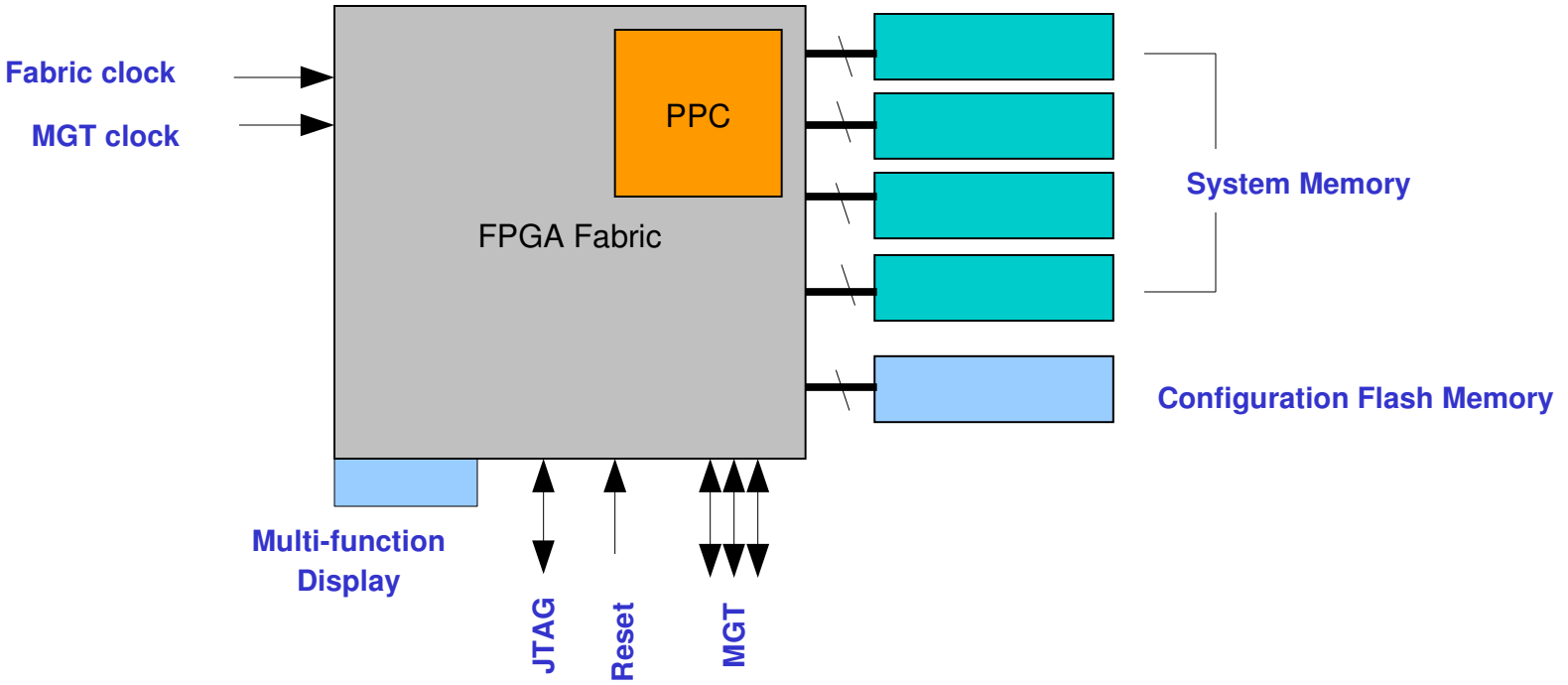
Pretty Good Protocol

- **Pretty Good Protocol (PGP)**
  - Serial point-to-point connectivity
    - *Physical interface is 2 LVDS pairs/lane*
  - Small footprint
  - Features: clock recovery, full duplex, reliable frame transmission and reception, deterministic (and small) latency
  - Implemented as IP protocol core interfaced to PIC
    - *Extensible in both bit-rate and number of lanes*
  - Used to interface the RCE to its storage flash memory subsystem
    - *Four lanes at 3.125Gb/s (~1GB/s per RCE)*
  - Used to interface the RCE to the front-end electronics
    - *Up to four channels at 3.125Gb/s (~1GB/s per RCE or ~2GB/s per RCE board)*

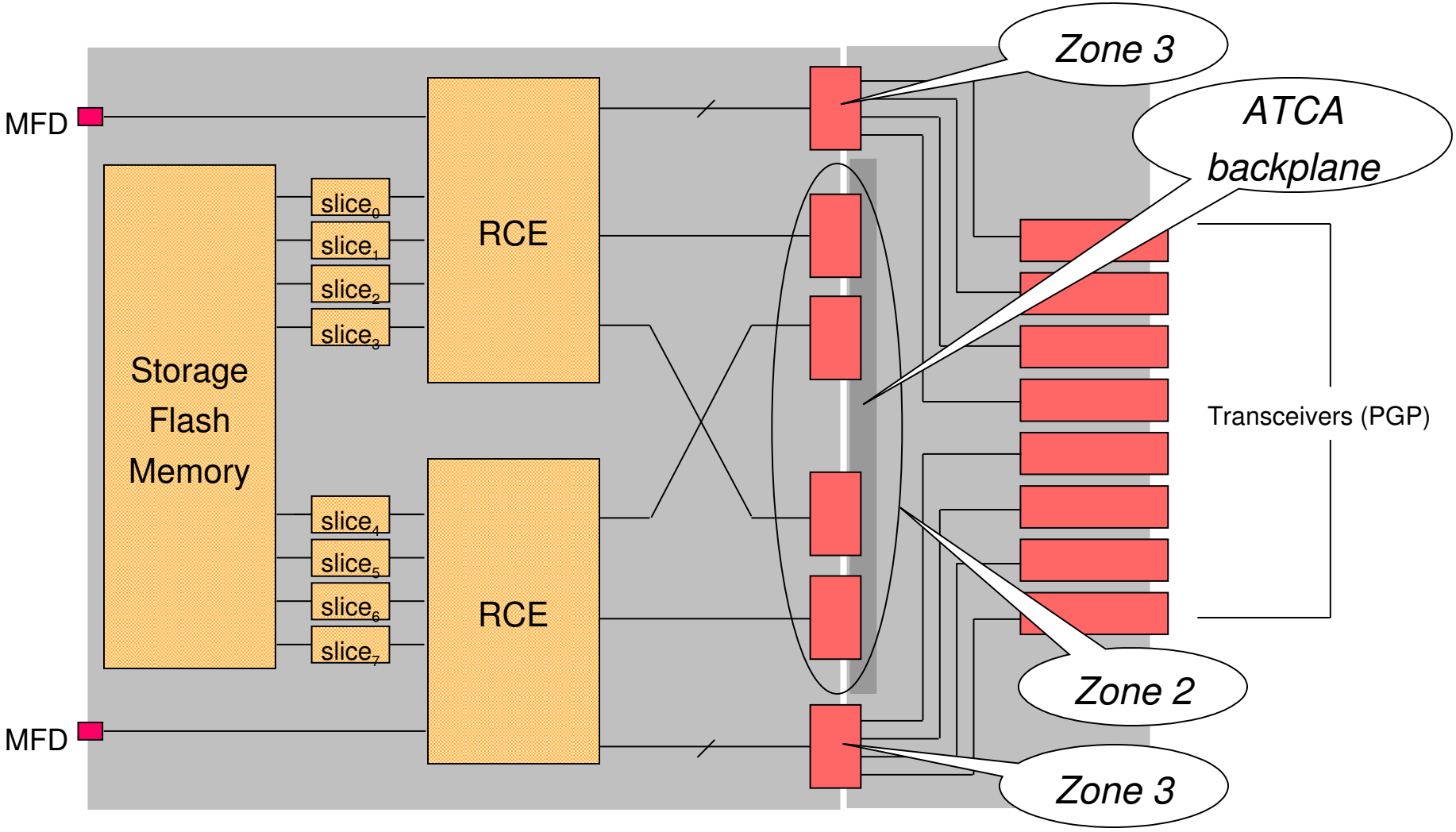
RCE Software

- **Software**
  - Ported open source Real-Time kernel
    - *Adopted RTEMS: Real Time Operating Systems for Multiprocessor Systems*
  - Written BSP mainly in C++
    - *Plus some C and assembly*
  - Written 10Gb Ethernet driver and PGP drivers for bulk data
  - 1Gb management interface driver
  - Built interface to RTEMS TCP/IP network stack (BSD)
  - Developed specialized network stack for zero-copy Ethernet traffic

RCE Block Diagram



RCE Board Block Diagram

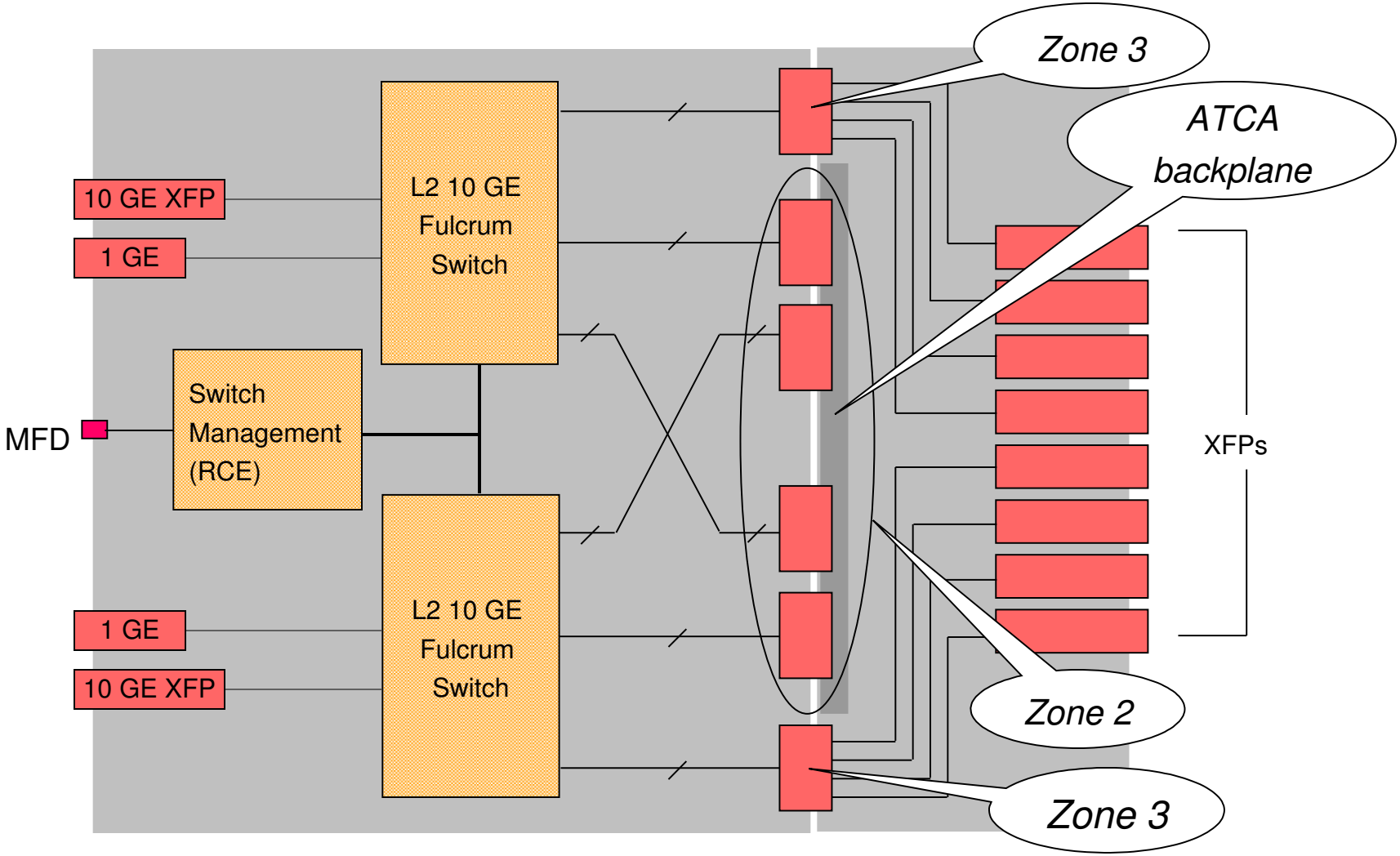


RCE Board with RTM

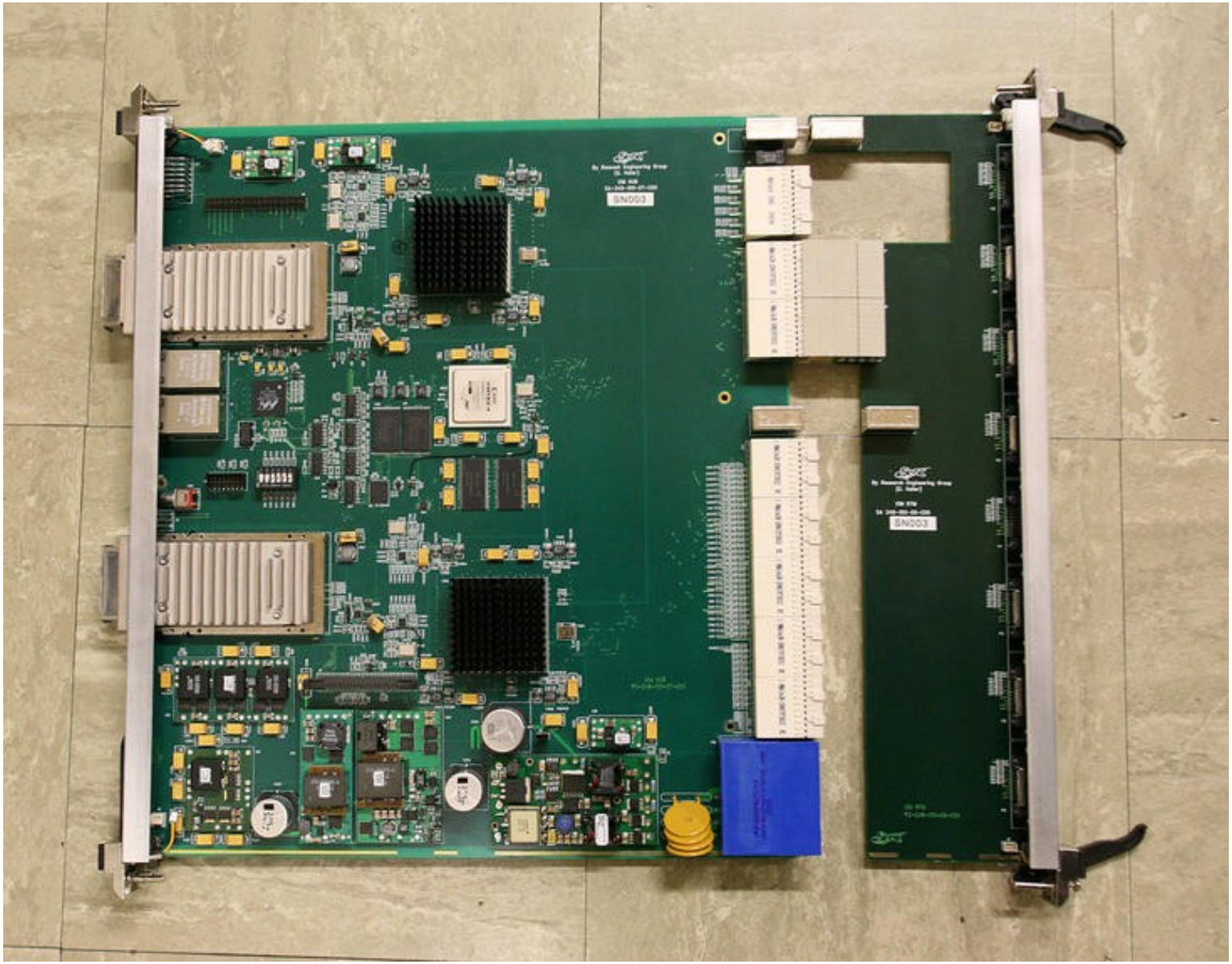
Cluster Interconnect Module

- **ATCA network card**
  - SLAC custom made board
  - Based on two 24-port 10Gb Ethernet switch ASICs from Fulcrum
    - *Up to 480 Gb/s total bandwidth*
  - Managed via Virtex-4 device
    - *Currently XC4VFX20*
  - Fully managed layer-2, cut-through switch
  - Interconnect up to 14 in-crate RCE boards (i.e. 28 RCEs)
  - Interconnect multiple crates for additional scalability
  - Power consumption: ~50W per CIM (~100W per board)
- **Fully configurable**
  - Designed to optimize crates populated with RCE boards
    - *Ability to use ATCA redundant lanes for additional bandwidth if desired*
    - *Ability to use 2.5Gb/s connections in place of standard 1Gb/s Ethernet*
  - At the same time may be configured to connect standard ATCA blades

CIM Board Block Diagram

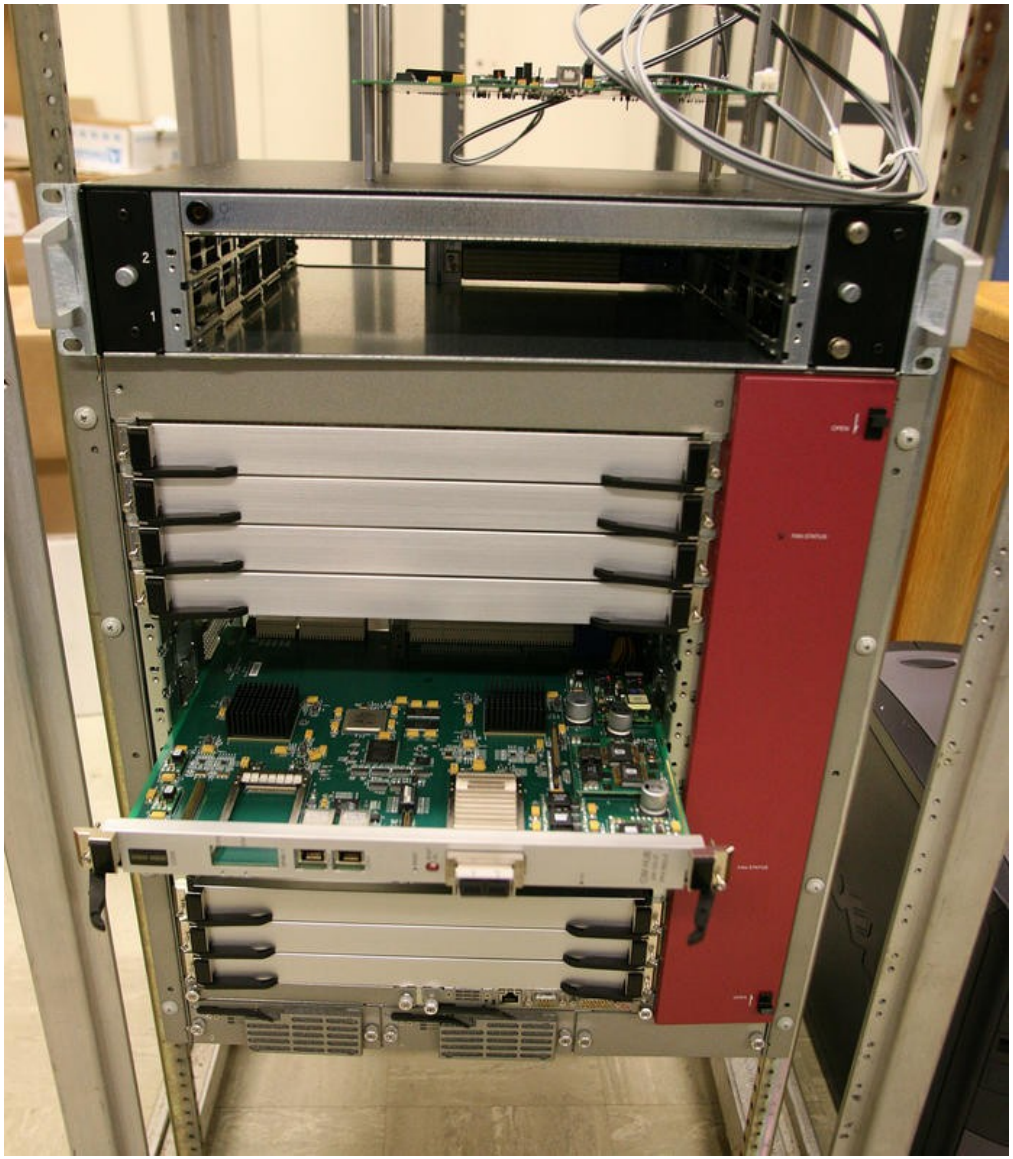


CIM Board with RTM



Front-end Development Board

ATCA 14-slot Chassis



Conclusions and Path Forward

- **LCLS common custom DAQ hardware devices fully prototyped**
  - RCE, CIM and front-end development board
- **Interface with FEE defined**
  - Uses common (among different experiments) communication protocol
    - *Implemented as IP cores interfaced to detector specific user logic in FEE FPGA*
  - Need to develop detector specific user logic for ASIC operations
- **Begun definition of the interface with off-line system**
  - Currently investigating HDF5 as common data format for data cache
- **Software development in progress on: DAQ partition management, user interface, calibration framework and data-flow**
- **Continue development on EPICS drivers for the control system of first LCLS experiment (AMOS)**