RT-10 Workshop on ATCA

Lessons learned in designing high speed, massively parallel DAQ systems using the high availability ATCA Platform

Michael Huffer, mehsys@slac.stanford.edu

May 22-23, 2010

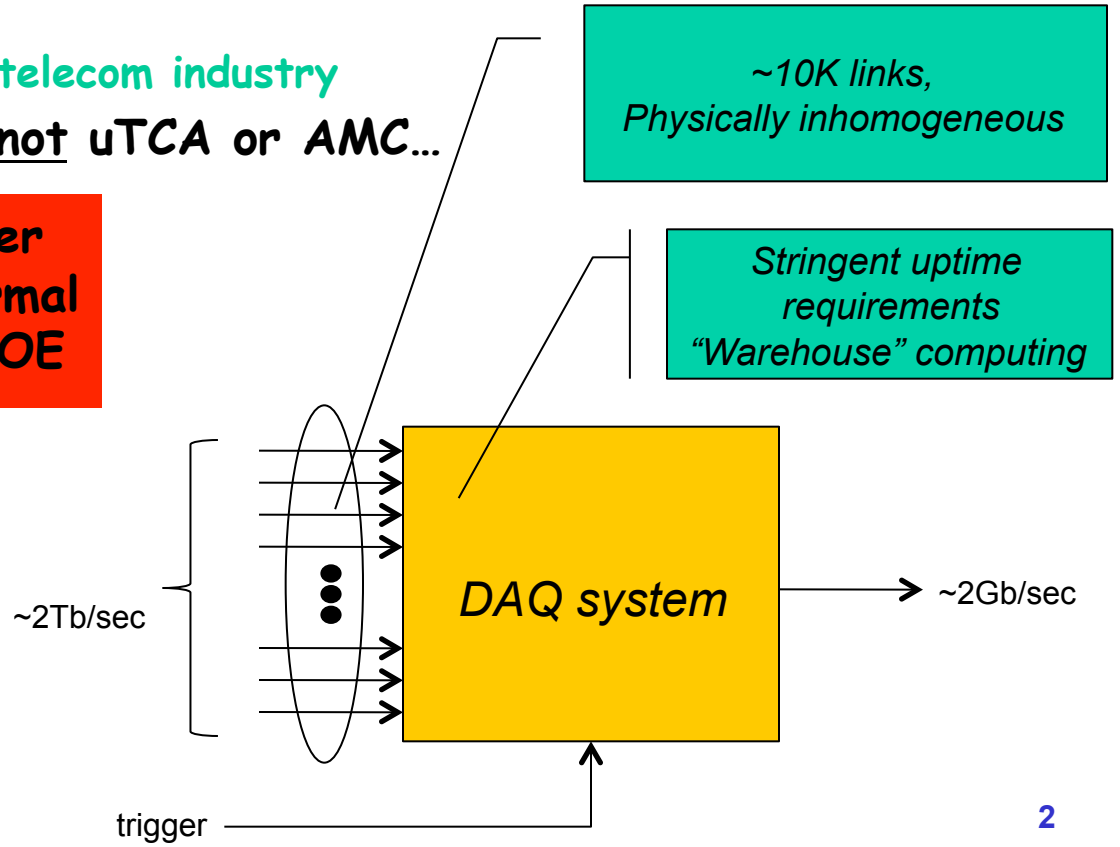
Representing:
Mark Freytag
Gunther Haller
Ryan Herbst
Chris O'Grady
Amedeo Perazzo
Eric Siskind
Matt Weaver

Disclaimers & context

- Do not profess to be an expert on ATCA, simply intend to…
  - Share our experience in developing ATCA products
- Neither am I an evangelist for the standard
  - Many alternatives exist with same attractive features
  - What differentiates ATCA from the pack (IMHO)…
    - its relative maturity
    - backing of PICMG & telecom industry
- Talk will focus on ATCA, <u>not</u> uTCA or AMC…

Reference to any manufacturer does not constitute any formal endorsement by SLAC or DOE

- We are our <u>own</u> customer
  - <u>Not</u> an advantage shared by industry
  - Separating these roles helps one understand why the specification is so complex…



Outline

- SLAC Detector (DAQ) R & D
  - Project genesis & motivation
  - Concepts & building blocks
    - The RCE & CIM
  - The RCE & CIM hosted on ATCA boards
- Why ATCA is a good fit for DAQ
- Shelf selection & board development
- Board design considerations
  - Introduction
  - Power distribution
  - Intelligent Platform Management (IPM)
  - Rear-Transition-Modules (RTM)
  - Transport Interfaces
- Summary

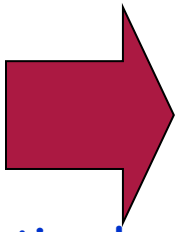
SLAC Detector R & D

- *DAQ/trigger "technology"* for next generation HEP experiments supported by SLAC
  - LCLS
  - LSST
  - SuperB
  - LHC luminosity upgrade (ATLAS)
  - Massively parallel database systems (Petacache)

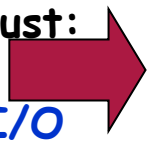
- Goals:
  - "Survey requirements & capture their commonality"
  - Tools, not solutions. "one size does not fit all"
  - Leverage recent industry innovation
  - Technology evaluation & demonstration "hardware"
    - Must be targeted to solving "real problems"

Three building block concepts

- Computational elements. Must:
  - be low-cost
    - $$$
    - footprint
    - power
  - support a variety of computational models
    - CPU
    - DSP
    - gates
  - have both flexible and performanent I/O
- Mechanism to connect these elements. Must:
  - be low-cost
  - provide low-latency/high-bandwidth I/O
  - be based on a commodity (industry) protocol
  - support a variety of interconnect topologies
    - hierarchical
    - peer-to-peer
    - fan-In & fan-Out
- Packaging solution for both element & interconnect
  - Must provide High Availability
  - must allow scaling
  - must support varying physical I/O interfaces
  - preferably based on a commercial standard



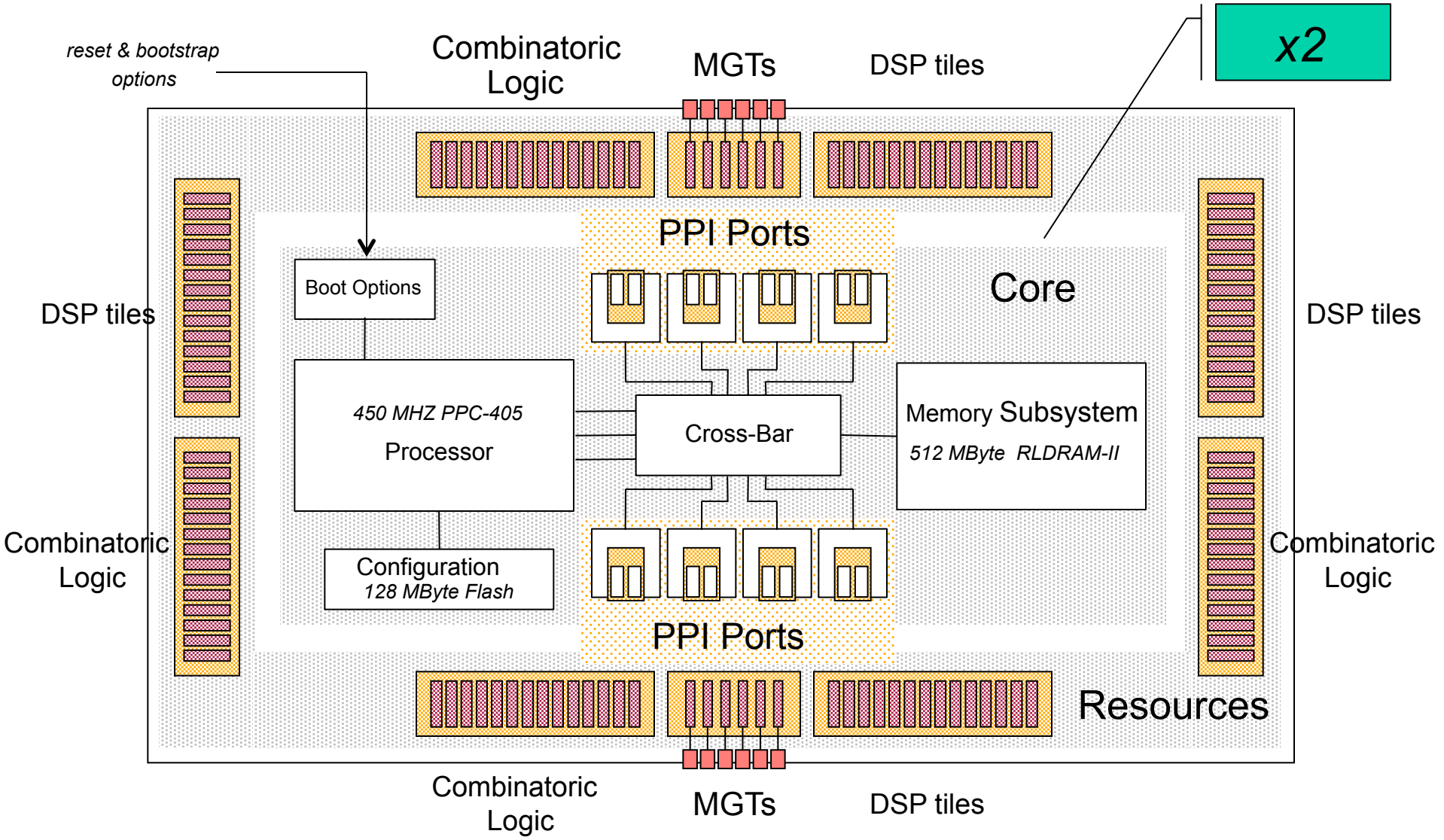
- The Reconfigurable Cluster Element (RCE) based on:
  - System-On-Chip technology (SOC)
    - *Virtex-4 & 5*



- The Cluster Interconnect (CI)
  - based on 10-GE *Ethernet* switching

- ATCA
  - *Advanced Telecommunication Computing Architecture*
  - crate based, serial backplane

(Reconfigurable) Cluster Element (RCE)



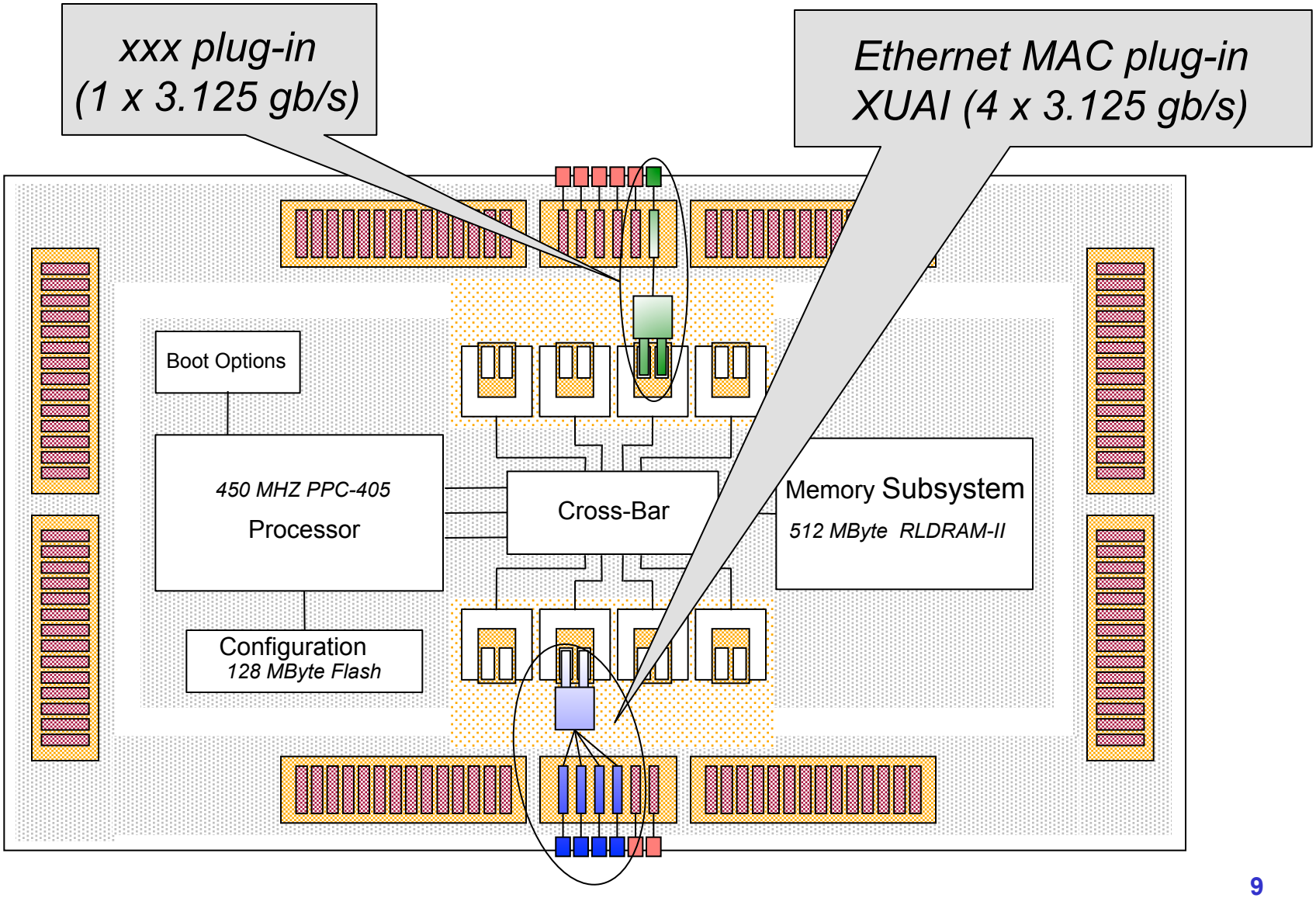
Embedded software & development support

- Cross-development…
  - GNU cross-development environment (C & C++)
  - Remote (network based) as well as local GDB debugger
  - Network (virtual) console and telnet support
- Operating system support…
  - Generic bootstrap loader (O/S independent)
  - Open Source Real-Time kernel (RTEMS)
    - POSIX compliant interfaces
    - Standard I/P network stack
  - Exception handling support
- Object-Oriented emphasis:
  - Class libraries (C++)
    - Plug-In support
    - Configuration Interface

Resources

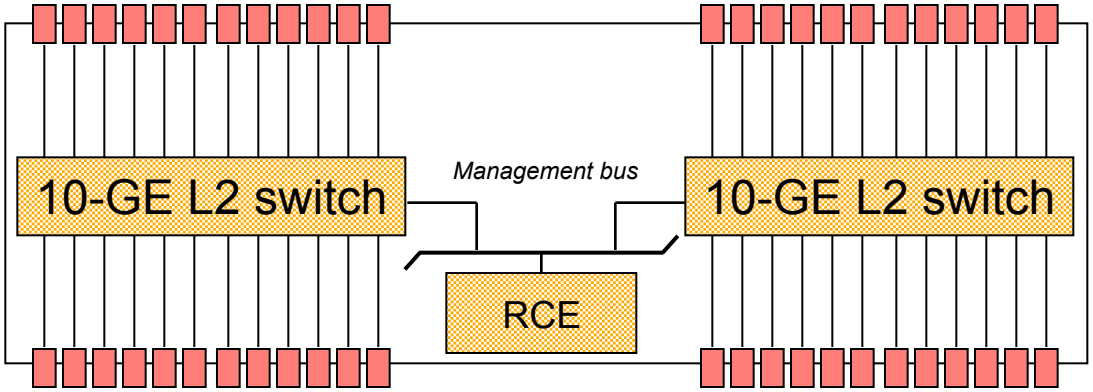
- Multi-Gigabit Transceivers (MGTs)
  - Up to 20 lanes of:
    - SER/DES
    - input/output buffering
    - clock recovery
    - 8b/10b encoder/decoder
    - 64b/66b encoder/decoder
  - Each lanes can operate up to 6.5 gb/s
  - Lanes may be bound together for greater aggregate speed
  - 8 lanes reserved for 10-GE (4 per core)
- Combinatoric logic
    - LUTs & gates
    - flip-flops (block RAM)
    - I/O pins
- DSP support
  - Contains up 240 Multiple-Accumulate-Add (MAC) units
  - Can interface through either plug-in or processor interface

Protocol "Plug-Ins"



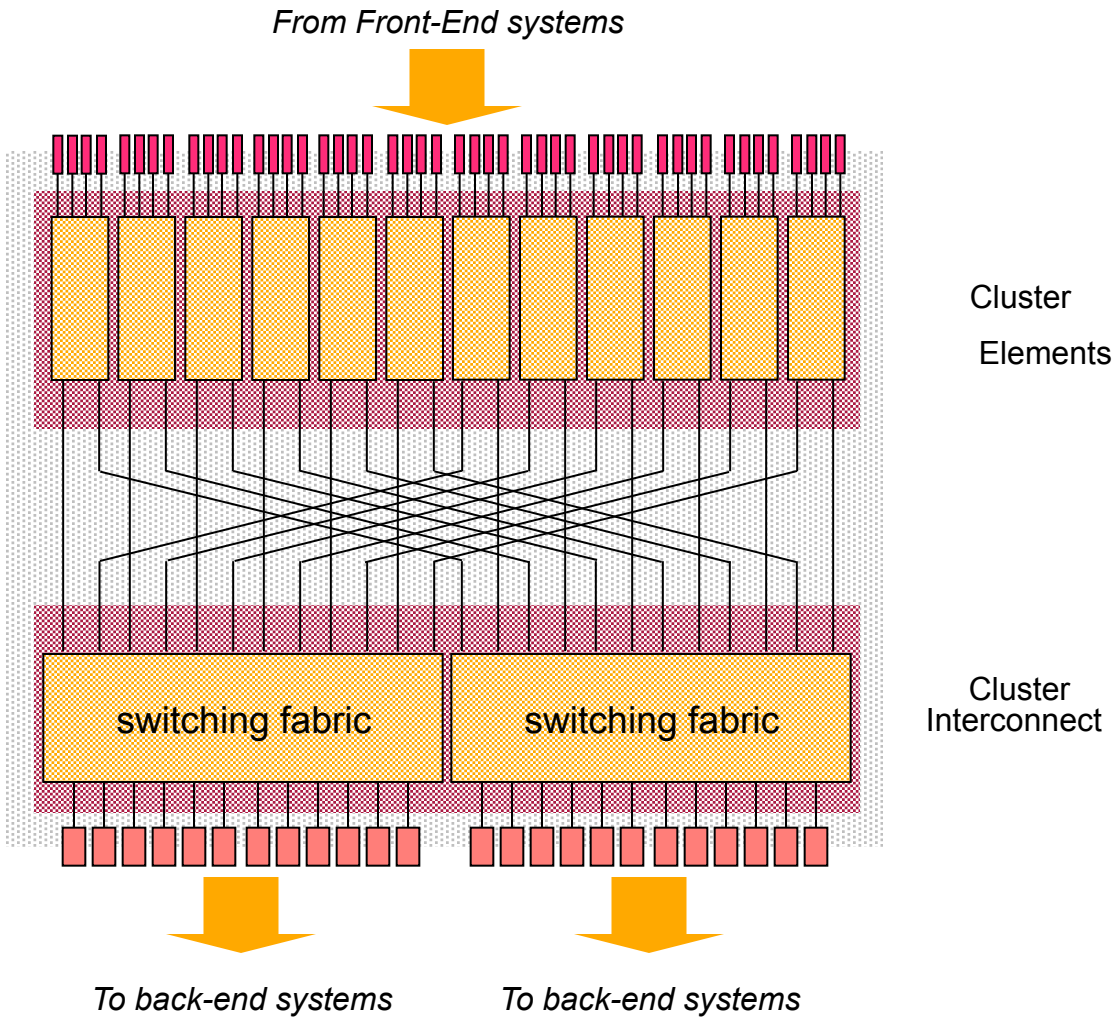
The Cluster Interconnect (CI)

- Managed 24/48 port 10-GE switch
  - Reuses RCE for management

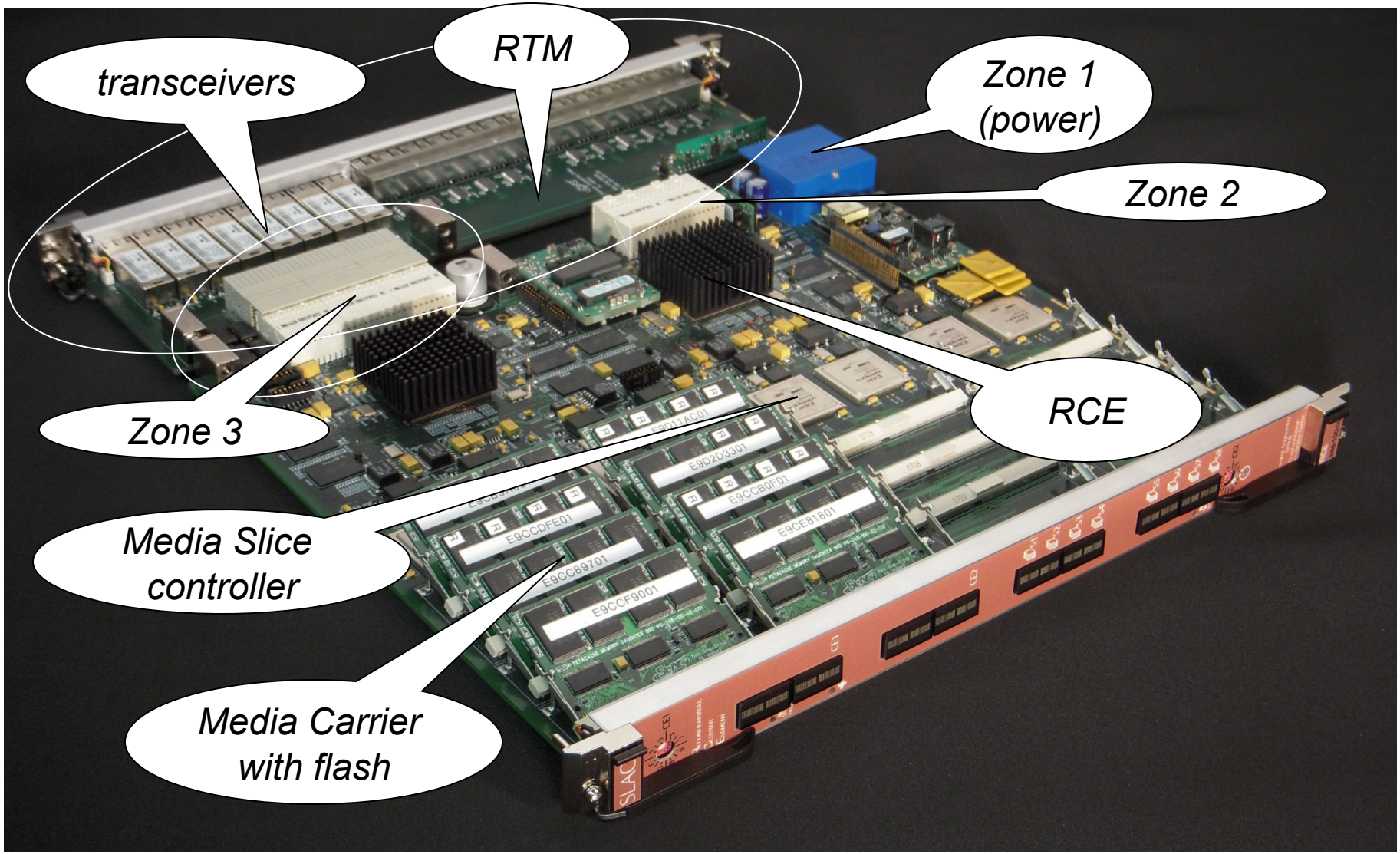


- Switching based on the *Fulcrum* FM224
  - 24 port 10-GE switch
  - is an *ASIC* (packaging in 1433-ball BGA)
  - 10-GE XAUI interface, however, supports multiple speeds…
    - 100-BaseT, 1-GE & 2.5 gb/s
  - less then 24 watts at full capacity
  - cut-through architecture (packet ingress/egress < 200 ns)
  - full Layer-2 functionality (VLAN, multiple spanning tree etc..)
  - configuration can be managed or unmanaged

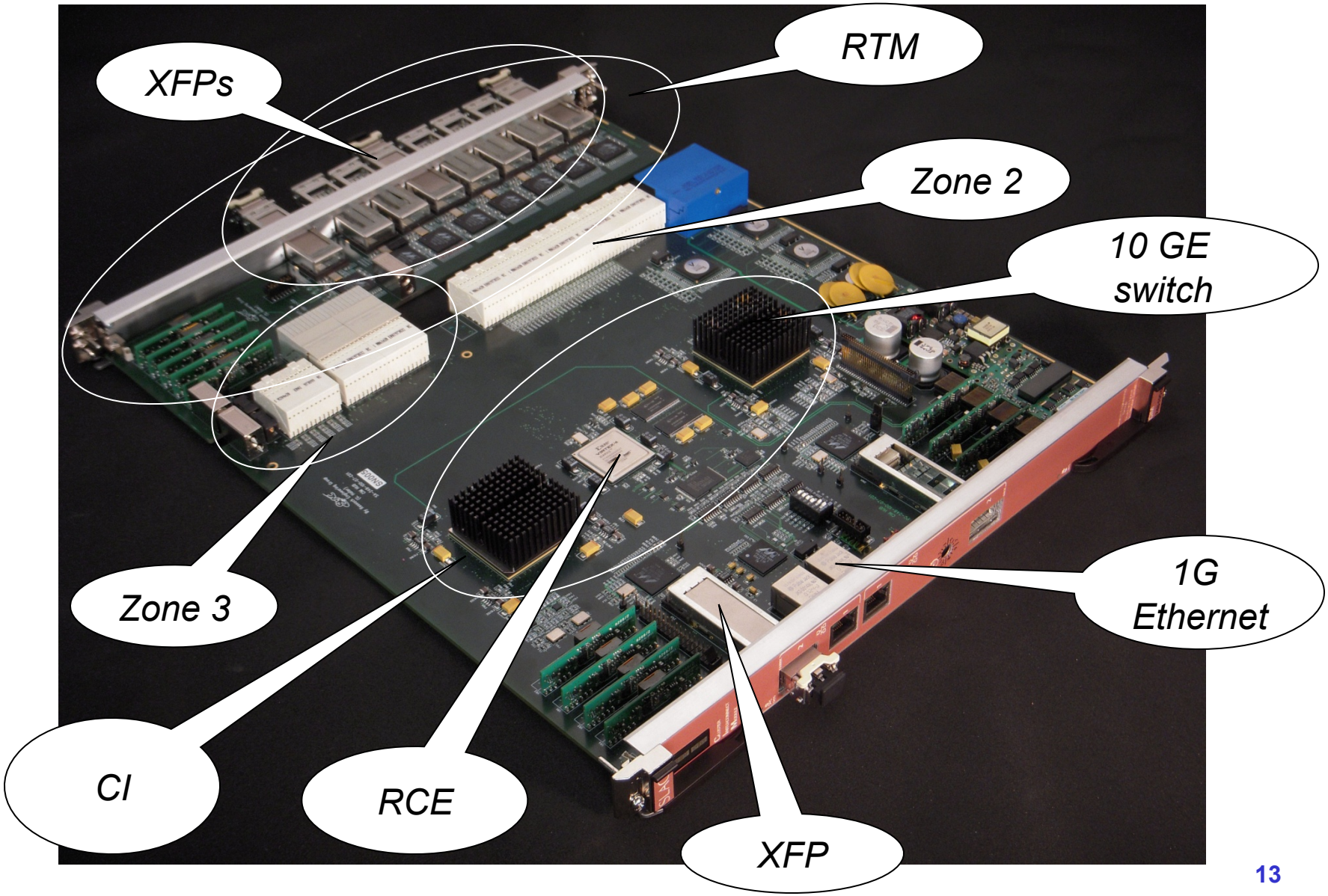
A cluster of 12 elements (24 nodes)



Node Board (RCE board + RTM)



Hub board (Cluster Interconnect board + RTM)



Why ATCA for large scale DAQ?

- Its attractive features:
  - Backplane & packaging available as a *commercial* solution
  - (Relatively) generous form factor
    - 8U x 1.2" pitch
  - Emphasis on High Availability
    - lots of redundancy
    - hot swap capability
    - well-defined environmental monitoring & control (IPM)
      - pervasive industry use
  - External power input is low voltage DC
    - allows for rack aggregation of power
- Its very attractive features:
  - The concept of a Rear Transition Module (RTM) allows…
    - all cabling on rear (module removal without interruption of cable plant)
    - separation of data interface from the mechanism used to process data
  - High speed, serial backplane
    - protocol agnostic
    - provision for different interconnect topologies

Shelf selection

- (loosely) ATCA Shelf corresponds to a classic "Crate"
- Shelves vary in size and orientation (horizontal or vertical):
  - 2 slots (minimum)
  - 5 slots (good development size)
  - 14 slots (maximum 19" rack)
  - 16 slots (Euro-standard)
- Smaller shelves (< 6) typically have mesh topologies
  - Degenerates to Star topology
- Smaller shelves are typically oriented horizontally
- Standard expects line voltage of 48 VDC
  - Handicap in bench-top (development) environment
- "Dumb" or "smart" shelf manager?
- Beware of differentiation between physical and logical slots
  - Physical slot numbering is invariant of board function, while logical slot numbering depends on board function

ASIS 5-slot Shelf

Front

fans

front board (node)

Shelf manager

front board (hub)

Power supplies

Back

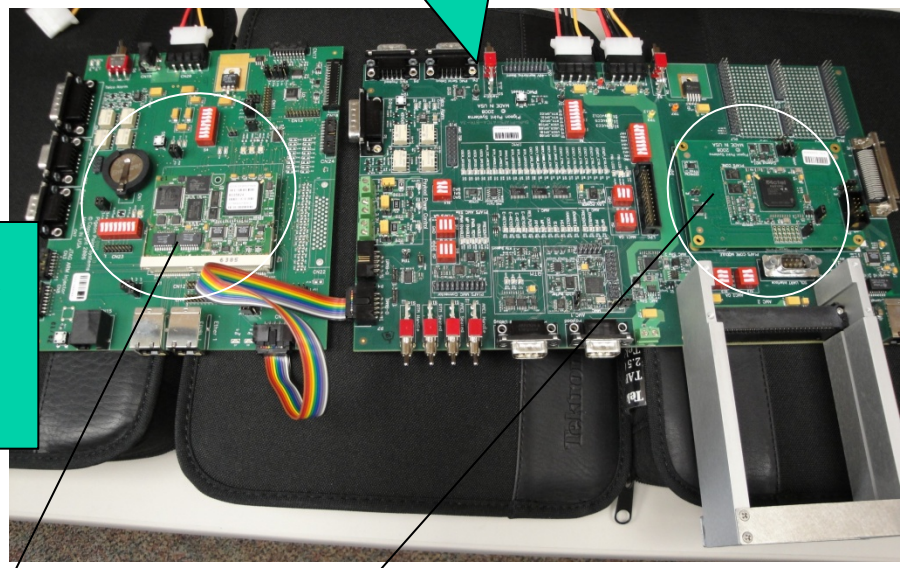
RTM (node)

RTM (hub)

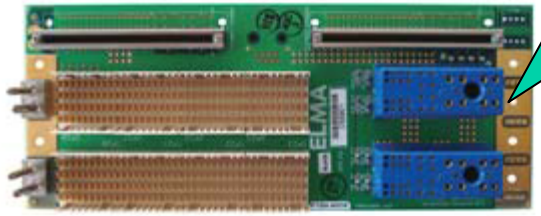
Development…

Board extender

IPM Controller Development system



"naked 2 slot back-plane



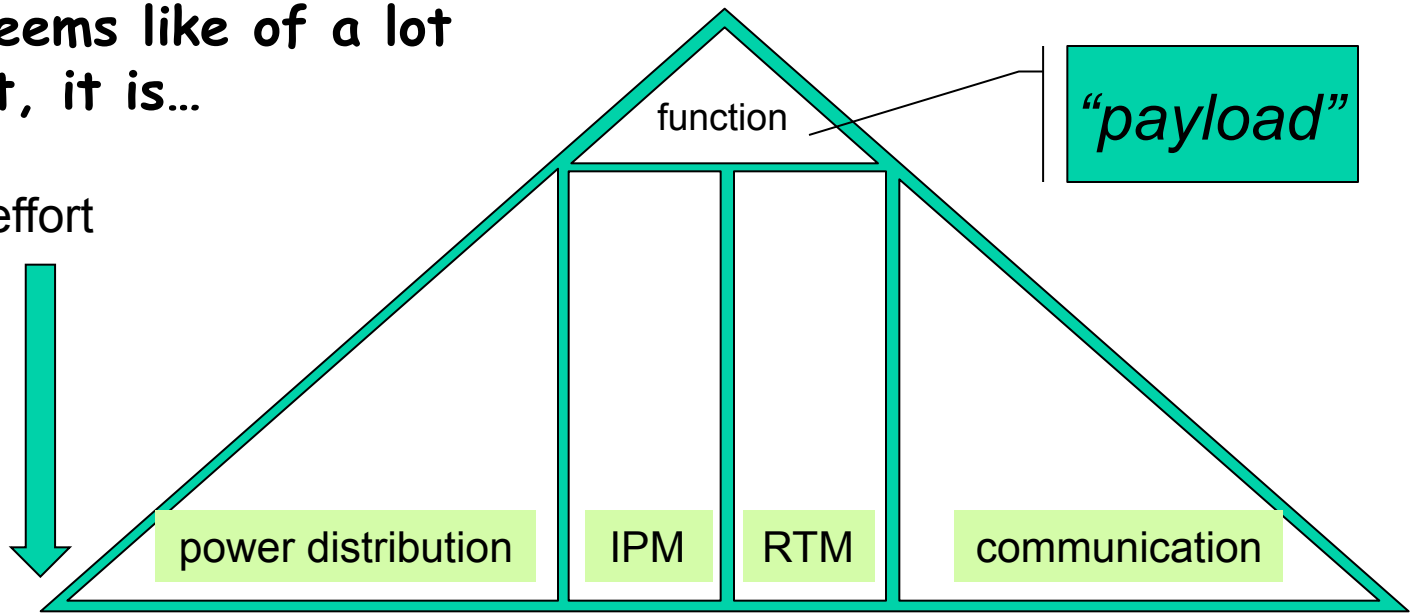
controller

shelf manager

Photos Courtesy ELMA-BUSTRONIC

Introduction

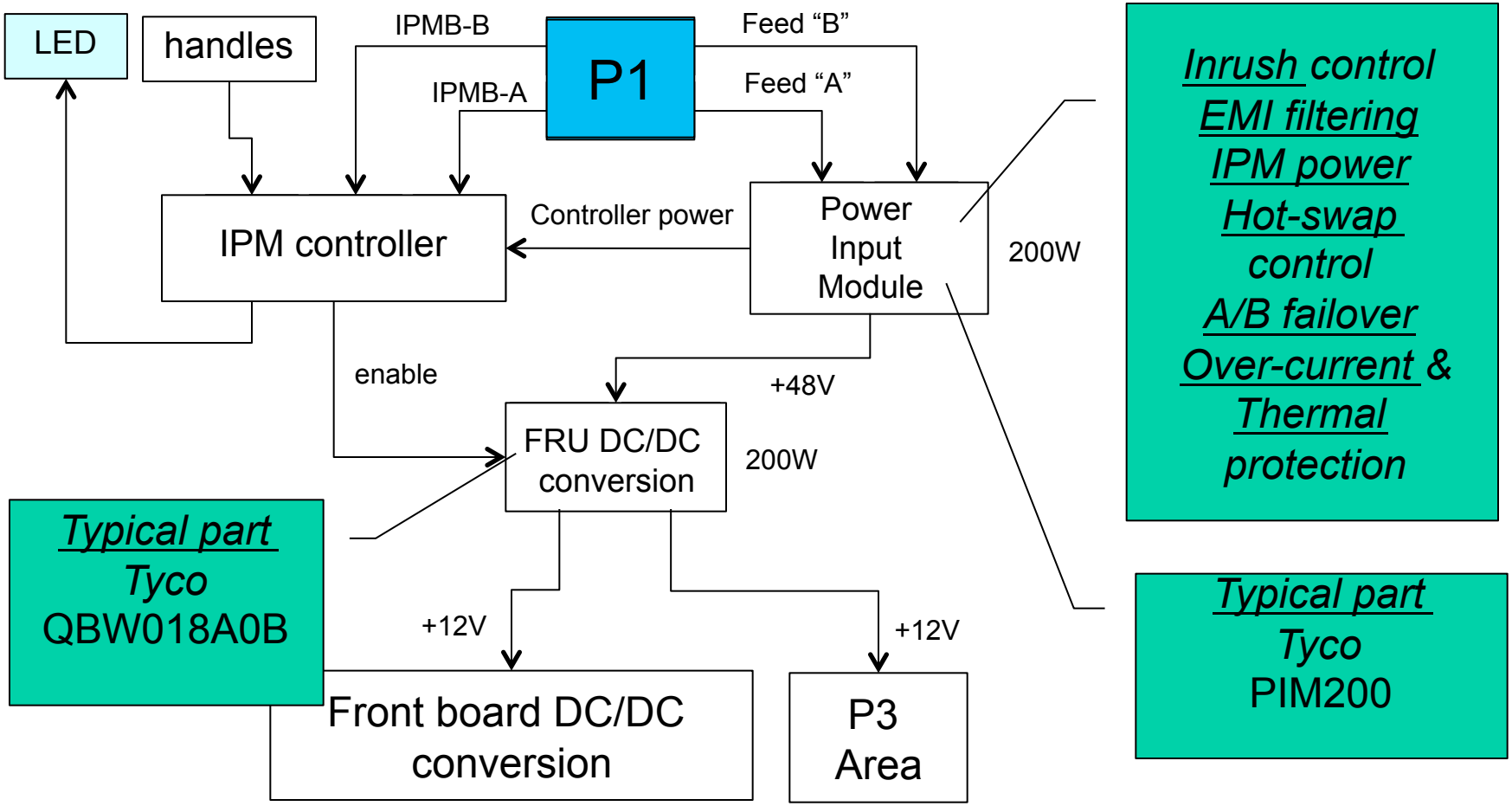
- If its seems like of a lot of effort, it is…



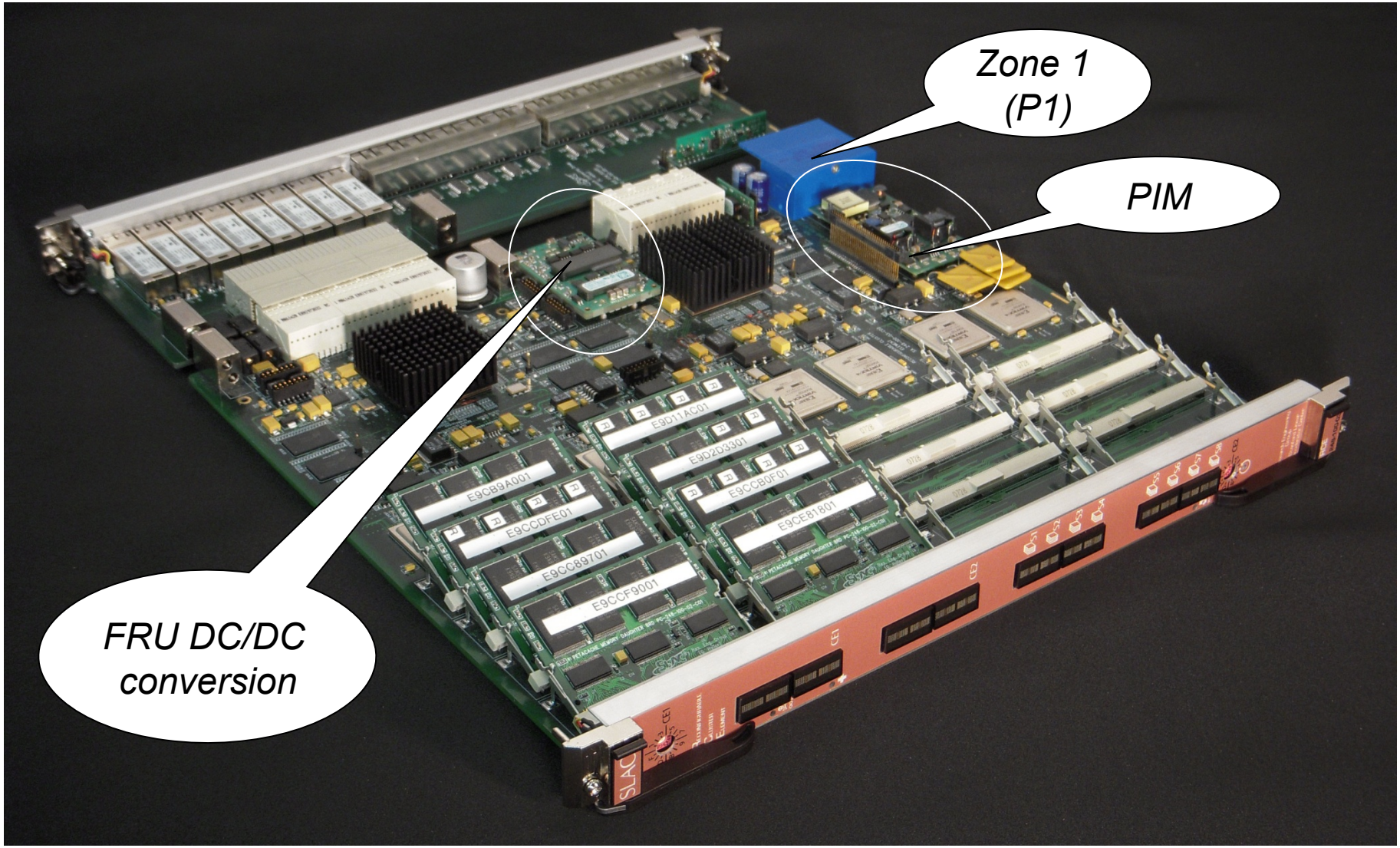
- Dderives from ATCA philosophy of "Think globally, act locally"
  - Support multi-tenancy philosophy
  - Maintain Inter-operability within an ecosystem
- Good news
  - Industry provides (at least) partial solutions
  - "cookie cutter approach will serve the developer well

Power Distribution (1)

- Assumes RTM hot-swap is dependent on Front-Board
  - The P1 connector sources most of the inputs, however…
  - The IPM controller mediates all power transitions



Power Distribution (2)



IPM from the board's perspective

- ATCA assumes your boards are sited in a data center in shelves outside both your interest & control
- Multi-tenant model: Boards (actually their FRUs) "rent" their slot from a shelves' "landlord"
  - Landlord communicates with boards through Shelf Manager (ShMC)
  - Boards communicate with landlord through their Controller (IPMC)
- Protocol is IPM (Intelligent Platform Management)
  - Pervasive usage outside ATCA
  - *Specification* is IPMI (IPM Interface)
  - Shelf *Bus* is IPMB (connections between slots & Shelf Manger)
    - IPMB is a super-set of I^2C distributed on back-plane
    - Shows up on the board's P1 connector
- Classic view of IPM is remote operation of board…
  - Monitor & configure board's payload
  - Power cycle
  - Reboot & reset payload
- However, IPM manages board "hot-swap" (a purely local function)

What *is* an (IPM) Controller?

- Physically:
  - Embedded processor (behavioral model of IPM is quite rich)
    - Lot's of software (could also involve some firmware)
    - Modest amount of persistent storage (code + configuration)
  - Discrete, digital I/O
  - ADCs and DACs
  - I^2C interface (at least two, typically three or more)
- Logical functions:
  - Manages IPMB communication
  - Manages hot-swap process
  - Support for E-keying
  - Monitors power and cooling
  - Responds to watch-dog timer (by resetting)
  - Maintains inventory information
  - Manages payload interface
    - Form & function not defined by the specification

(IPM) Controller (solutions)

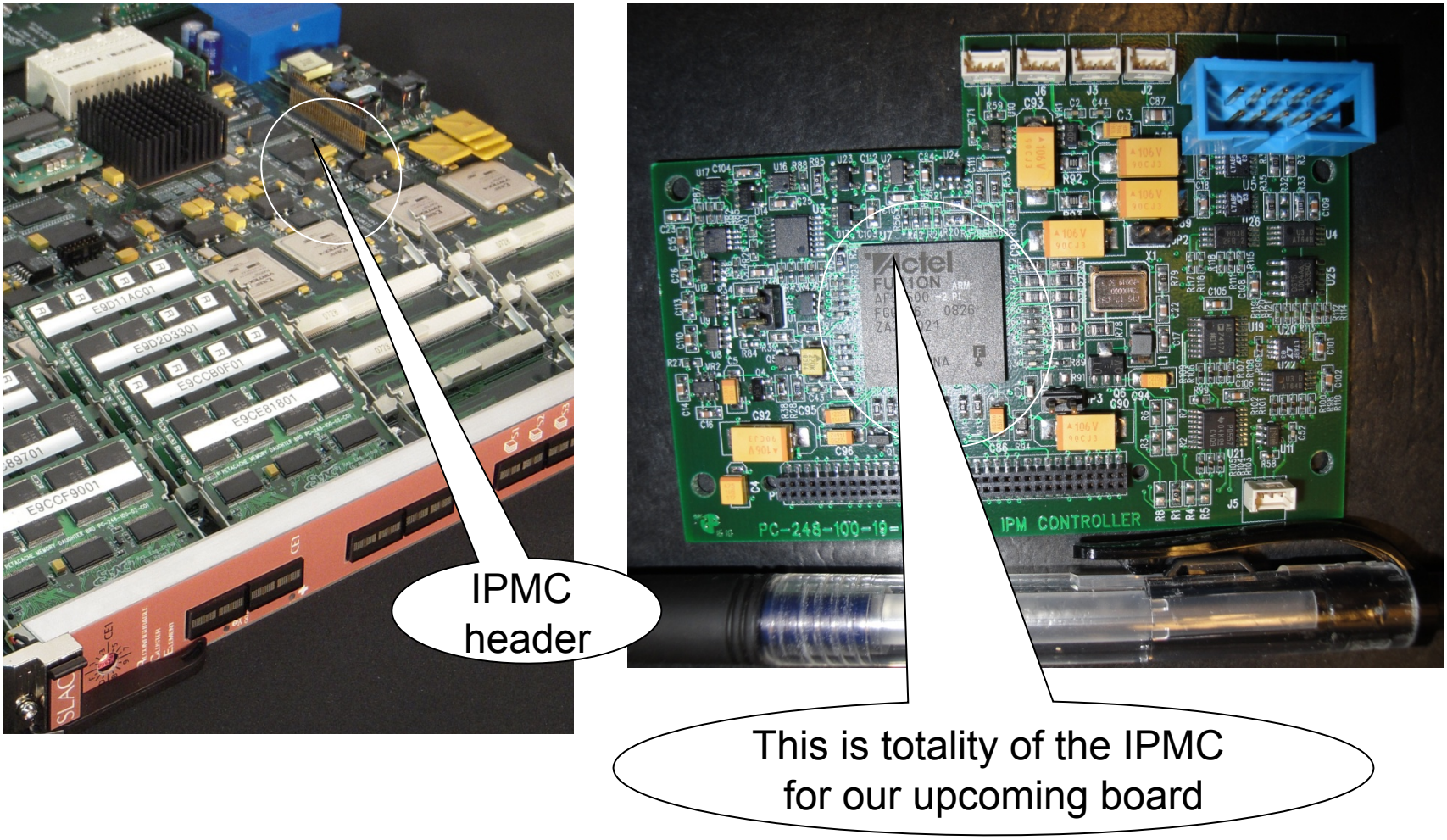
- Build…
  - Pretty daunting (significant NRE and maintenance costs)
  - Eliminates any potential royalty & licensing issues
- Buy (most likely only a partial solution)
  - Depending on payload sophistication could provide complete solution
  - Upfront costs could be prohibitive
  - Potential royalty & licensing issues
- Procrastinate:
  - Bring all necessary signals to header or stay-clear area on board
  - Allow deferring solution at least through development cycle

We bought from Pigeon-Point-Systems

- Company was purchased a year ago (or so) by ACTEL
  - Cost and licensing much more flexible
- PPS solution is hosted on ACTEL's (Smart) Fusion FPGA product
  - This is a SOC, incorporating:
    - ADCs, DACs and digital I/O
    - Hardware signal conditioning
    - Processor, hard or soft (ARM Cortex-7)
    - Configuration & user flash
    - I^2C interfaces
- What are you buying?
  - Development license
    - Source code (software + firmware)
    - Development board
    - Reference design (schematic + software/firmware)
  - Right to distribute royalty free (binary) any board design
- Does not include the (FPGA) devices themselves…

IPMC daughter-board

- Current boards simply map reference design to daughter-board…



The RTM (1)

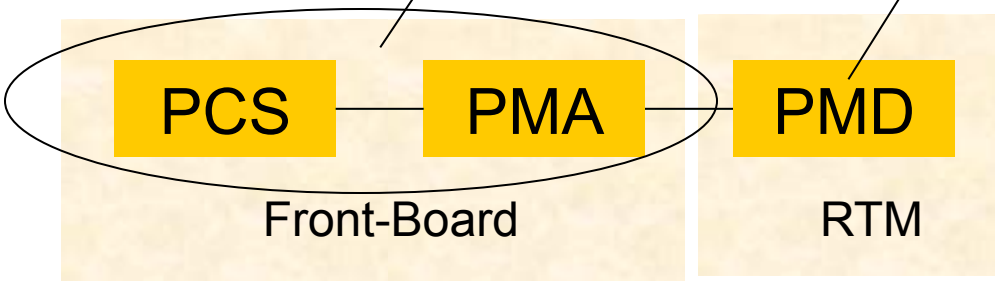
- Physical manifestation of the principle of separation of interface from implementation.
- Ensures cable plant management and hot-swap do not collide
- When is an RTM appropriate?
  - Design requires more area than realizable with Front-Board
  - Design requires a significant amount of external I/O
- What should not go into an RTM?
  - Anything which consumes a lot of power
    - Minimal cooling available is 5W (!)
    - Increased at shelf manufactures discretion (30-40W typical)
    - Access to power limited to P3 area through Front-Board
  - Anything which costs $$$ & represents complexity
    - RTM represents a small & relatively confined footprint
    - Access to back-plane limited to P3 area through Front-Board
- In short, to maximize re-use:
  - Intellectual investment should be in Front-Board
  - RTMs (for a given Front-Board) should be a "dime-a-dozen"

RTM (2)

- Consider a typical communication protocol's Physical layer…
  - 10-GE
  - PCIe
  - SATA
  - Fibre/channel
  - Infiniband
- Composed from three sub-layers:
  - Physical Media Access (PMA)
  - Physical Coding Sub-layer (PCS)
  - Physical Media Dependences (PMD)

Excellent candidates for re-use

CX-4,
XENPACK,
SFP+,
etc…



The RTM – Zone 3 considerations

- P3 connectors are your generic interface to your Front-Board
  - Proper choice of connector & pin-outs ensures re-use
  - Pay attention to mechanical tolerances
- Connectors will need to support following functionality…
  - Power & ground
  - JTAG (Reprogramming of board logic) Connect to:
    - Front-Panel of Front-Board
  - Hot swap support (handles and blue LED) Connect to:
    - IPM controller
  - Housekeeping (payload configuration & monitoring). Connect:
    - Front-Board payload
    - IPM controller
  - RTM "type" identifier. Connect to:
    - IPM controller (& possibly Front-Board payload)
  - External I/O (High-speed serial and/or parallel). Connect:
    - Front-Board payload
- In Short: same functionality issues as in Zone 1 & 2
  - Reuse ZD (ADF) connectors
  - Consider multigig RT

The RTM – Hot swap?

- Specification defines hot-swap of the RTM as:
  - Activation/Deactivation independent of Front-Board,…
  - Not whether or not it can inserted/removed "live"

- Hot-swap (as defined) is hard, requiring (at a minimum)…
  - treatment of the RTM as separate (managed) FRU
  - careful consideration of connection's electrical properties
  - attention to interaction with front-board payload

- Recommendation:
- Treat Front-Board & its RTM as a single unit
- Allow RTM live insertion/removal dependently with Front-Board
  - *Front-Board* activation/deactivation actives/deactivates *RTM*
  - *RTM* activation/deactivation actives/deactivates *Front-Board*

Transport Interfaces (Introduction)

- The specification assumes boards need to interact with one another…
  - Consequently defines four different types of board communication:
    - The *Base* Interface
    - The *Fabric* Interface
    - The *Clock* (Synchronization) interface
    - The *Update* Channel Interface
- The overriding philosophy is to be as protocol agnostic as possible
  - Specification only determines connectivity between boards
  - Assumes data is transmitted/received serial-differential
  - Assumes communication is full-duplex (independent transmit/ receive)
- However, there are exceptions as encapsulated by these two rules:
  - Boards must provide IP support on either Base or Fabric Interface
  - The base interface must satisfy an Ethernet MAC
    - 10/100/1000Base-T

The Channel (from the board's perspective)

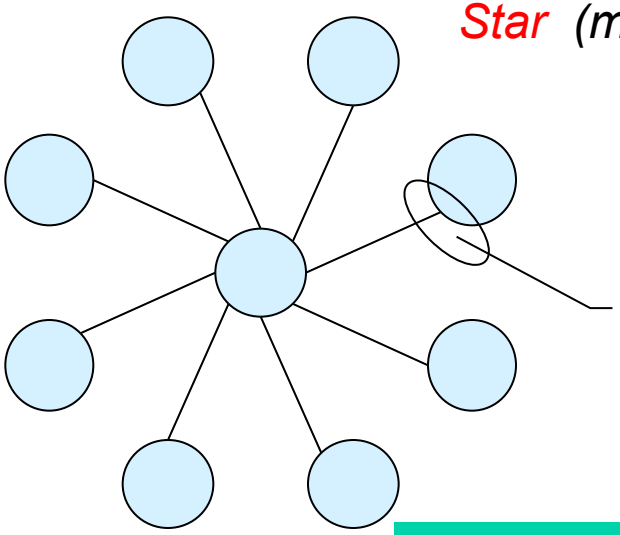
- All transport interfaces are characterized in units of Channels
- Channels correspond to pin assignments on Zone 2 connectors
- Capacity & number of channels varies by Interface Type
  - The capacity of a channel is measured in units of either:
    - Signal/pairs (two "wires")
    - Ports (2 signal/pairs, 1 for transmission & 1 for reception)
- Channels on one board connect to channels on other boards.
  - Connection topology also varies by interface type.

Interface Type	Capacity		Maximum number	Topology
	pairs	ports		
Base	4	2	16	Point-to-point
Fabric	8	4	15	Point-to-point
Update	10	5	1	Point-to-point
Synchronization Clock	6	3	6	Bussed

Shelf topologies

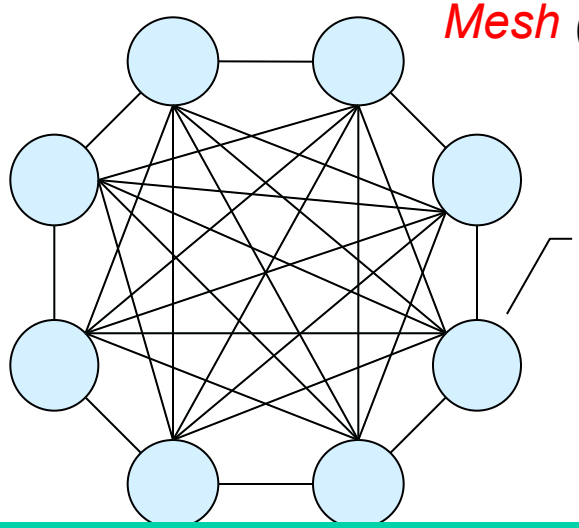
- Determines how channels on one board are mapped to channels on another board.
  - Topologies for *Base*, *Update*[1] and *Clock* interfaces are fixed
  - Topology for the fabric interface varies. However…
    - ATCA restricts the choices to a small, finite set…
    - with its members derived from 2 basic configurations:

Star (many-to-one)



channel

Mesh (any-to-all)

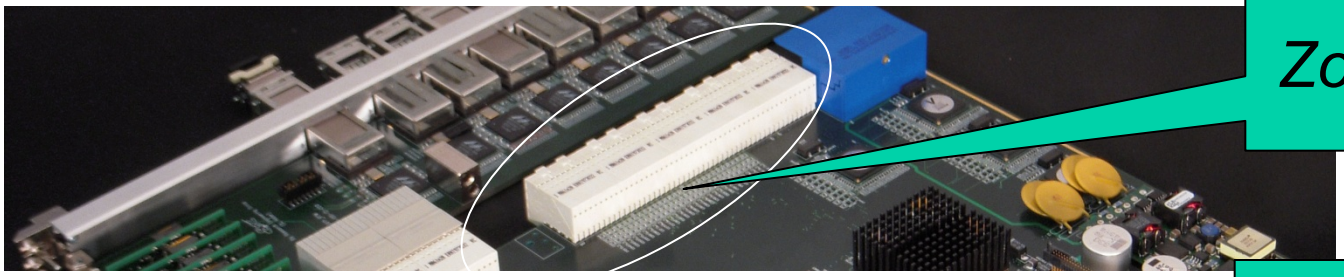


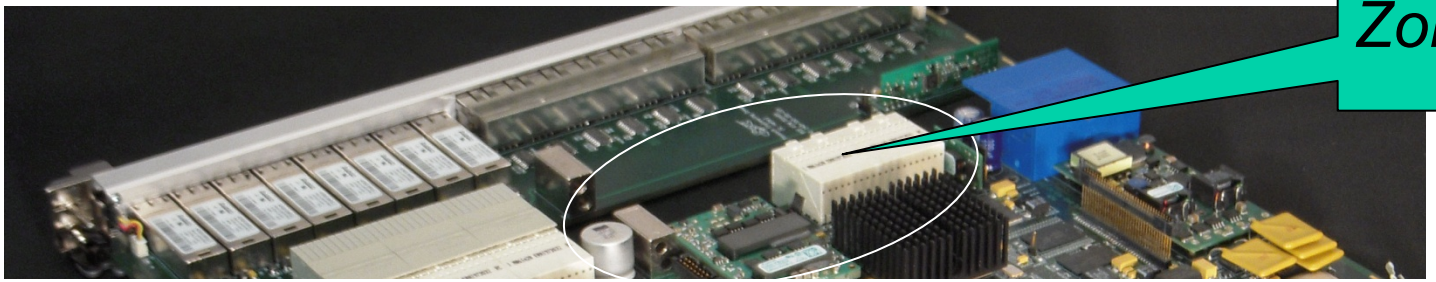
Board (slot)

- Star topology forces ATCA to differentiate board capability
  - *Node* capable
  - *Hub* capable
  - *Node & Hub* capable

Node v Hub boards

- Difference in number of connectors in Zone 2
- E-keying requirements will vary
- Constraints on slot usage
  - Hub boards must occupy (logical) slots 1 & 2
  - Node boards won't function (predictably) in hub slots
  - portability issues with Update Channel usage (see below)





Dual-Star Fabric

- Intended for redundancy (fail-over).
- Node boards must reside in (logical) slots 1 & 2
- Note: Logical slot numbering is shelf dependent
  - Slots 1& 2 must be designated with RED (not always true)
- Arguably most popular (& inexpensive) configuration

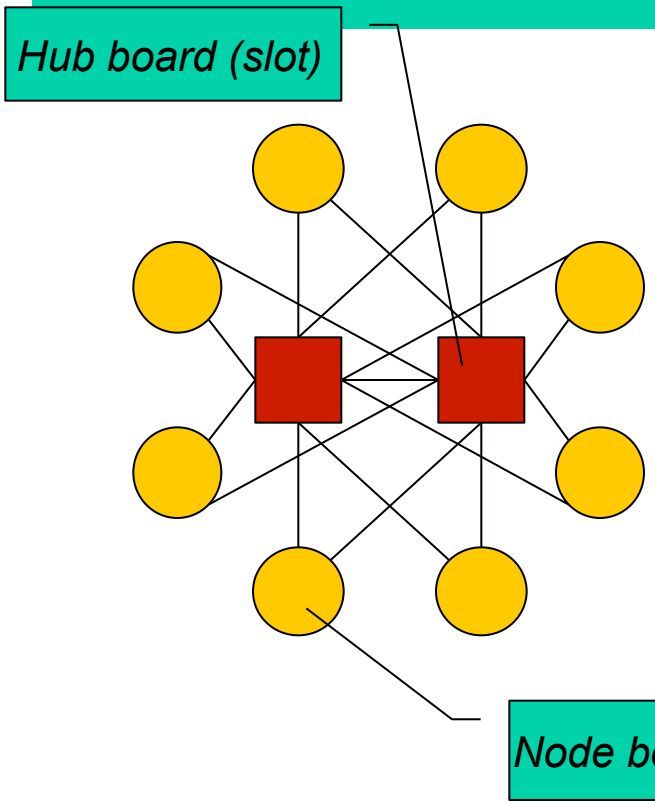
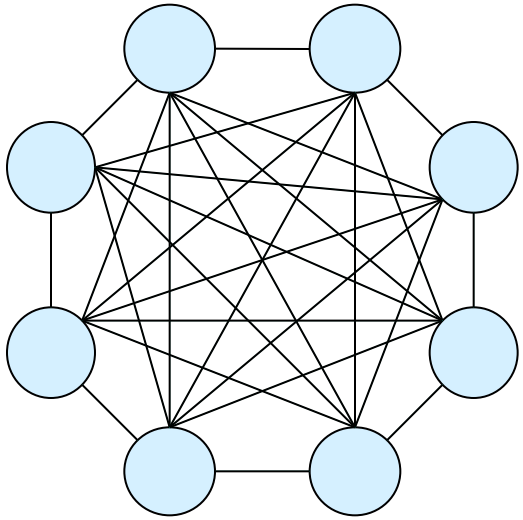


Photo courtesy ELMA-BUSTRONIC

Mesh Fabric (full & replicated)

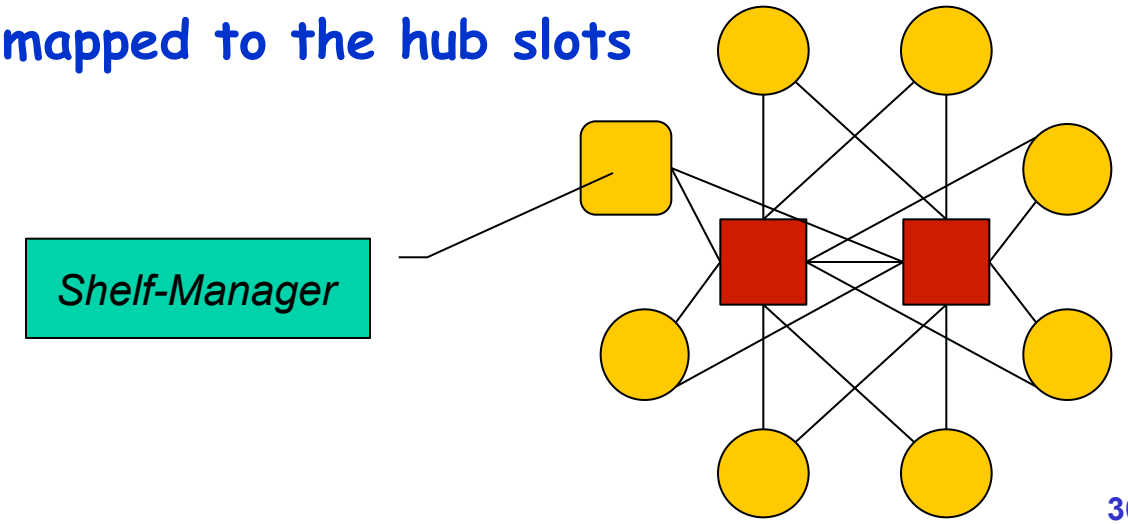
- More connectivity, more connectors, … more expensive
- Meshes can used as Stars
- Replication doesn't change connectivity, it changes *capacity*
  - Added in units of channels
    - 1X, 2X, 3X
  - Comes at the cost of reducing slot count
  - The invariant is the number of channels per board (15)

Photo courtesy ELMA-BUSTRONIC



Base Interface and topology

- Independent of fabric topology always Dual-Star topology
- In general same rules apply as *Fabric* Dual-Star. For example:
  - Nodes always map channels 1 & 2 to the hub slots (1 & 2)
- However, some differences…
  - Must implement *Ethernet* 10/100/1000Base-T
  - Drivers do not require isolation from back-plane prior to system management enable
    - Specifically auto-negotiation is unconstrained
    - Implies base-interface does not require E-Keying
  - Shelf Manager is mapped to the hub slots



Update Interface

- Intended to help "glue" different slots together
  - Support for (virtual) multi-width boards (2X and 3x)
- Same electrical requirements as Base & Fabric interfaces
- Which slots does a back-plane glue together?
  - Specification is somewhat ambiguous. To quote:
    - "Update Channels **should** be routed between physically adjacent slots"
    - "Logical slots 1 & 2 **should** have the Update Channel routed between them"
- In fact, different manufactures for different back-planes map the Update Channel in different ways

- Board developers should be cautious in its usage
  - Must your board must be portable across all back-planes?
- As always E-keying is vital to correct operation…

Synchronization Clock Interface

- Provides synchronous timing distribution to the boards of a shelf…
- Seen by the back-plane as six (6) individual busses.
  - Each bus constitutes one multi-drop, differential pair
- For redundancy the six buses are divided into three groups:
  - CLK1 (A & B) - Telecom specific
    - 8 KHZ clock A/B failover (digital telephony)
  - CLK2 (A & B) - Telecom specific
    - 19.44 MHZ A/B failover (SONET reference clock)
  - CLK3 (A & B) - User defined
    - A & B can be used independently, but limited to 100 MHZ
- Comment: One of the few places where industry neutrality is violated
- Each bus is implemented to the specifications of MLVDS
  - Multi-drop LVDS
    - both National Semiconductor & TI provide a wealth of resources
  - Excellent reference: National Semiconductor Application Note 1503
    - "Designing an ATCA Compliant M-LVDS Clock Distribution Network"
- Usage is problematic in a multi-tenant environment due to bus topology
- Again, E-keying is vital to success…

Summary

- SLAC detector R & D (DAQ) strategy based on modular building blocks
  - (Phase I) architecture is now relatively mature. Resulted in…
    - Both hub & node reference boards (& corresponding RTMs)
    - Validation of ATCA as viable DAQ platform
  - (Phase II) under development
    - Will result in a single (full mesh enabled) reference board
- ATCA is non-trivial. Complexity is due to:
  - Eco-system Interoperability
  - Multi-tenant architecture
  - High Availability
- Practice role separation (The manufacturer is not the customer)
- Consider these issues early in design process…
  - What transport interfaces & with what protocols?
    - Node v Hub boards
  - IPMI (licensing and royalty issues as well as technical choices)
    - Consider procrastination
  - Does your front-board need an RTM?
    - Intelligent usage of the P3 area is the key to front-board reuse
- Its not as hard as you might believe…
  - Industry provides solutions to many ATCA specific functions
  - The first board is the hardest (cookie cutter approach is quite viable)
    -