High Bandwidth Electronics

Ryan Herbst
What are the standard components in a detector system?

- Detector/Amplifier & ADC
- Digital front end
  - Configure and monitor amplifier & ADC
  - Receive, pre-process & deliver data to DAQ
  - Detector specific data processing functions
- Data transport protocols
  - Moving the data from the front end to the DAQ
  - Natural boundary between detector and facility DAQ
- DAQ (detector specific)
  - Online data processing
  - Multi-detector control & monitoring
  - Event building & parallel data processing
- Storage
  - Permanent storage on disk and tape

Will discuss in this presentation:
- Digital Front End
- System on chip technologies
- Data Transport
- DAQ platforms
Digital Front End

- Typically contain a FPGA (Field Programmable Gate Array)
- High bandwidth interface to analog electronics
  - ASIC with integrated ADC
  - External ADCs
- Configure and control amplifier and ADC
- Reorganize data to ease DAQ software processing
  - Smooth scan across the detector
  - Align data to byte boundaries, avoid bit packing
- Timestamp added to data frame
- High speed summary data processing
  - Detector protection
  - L1 trigger contribution
- Natural place to implement complex data reduction
  - FPGAs contain a great deal of computing power
  - Required for some high frequency beams
  - Embedded processor with firmware offload
    - System on chip technologies
  - Limited to single processing node
- Processing power often limited
  - Mechanical size limitations
  - Power & cooling limitations
- Users can be a roadblock to front end data processing
  - Requires up front specification of data processing algorithms
    - Experiment simulation can help in this effort
  - Danger of lost data due to mistakes and bad assumptions
  - Need raw data mode to verify processing
Data Processing Option – System On Chip

- Enables combination of software & hardware data processing in a single package
- Deployed with front end electronics or at DAQ
- Embedded processor
  - Real time operating system with wide memory bus
- Well defined interface for user firmware (Protocol Plug In)
  - Must be platform independent
  - Low latency access to memory
  - Utilize partial FPGA reconfiguration to ease user synthesis
- Well defined interface for user software
  - Access to data
  - Configure and control protocol plug in
  - Limit user software impact on system performance
- External interfaces implemented as protocol plug ins
  - Interface to incoming data
  - Back end data transport
- Building block for parallel processing platform
- Implemented as a flexible platform
  - Choose one of many FPGAs within a given family
  - Choose memory required for platform

- SLAC RCE Node
  - Xilinx Virtex 5 with embedded PPC 440 processor (438Mhz)
  - 8 protocol plug in interfaces
  - 4GB DDR3 memory (312Mhz)
  - 2GB uSD card for software and configuration storage
  - 10Gbps Ethernet backend interface
Data Transport Between Front End & DAQ

• In most cases the front end is physically separated from the DAQ
  - Exceptions where front end is integrated into DAQ chassis
• May distribute clock and trigger to the front end
• Must support long wire runs (>5 meters)
• Penetrate vacuum chamber in some cases
• Optical is the preferred medium when available
  - Provides electrical break between front end electronics and DAQ
  - Allows for higher data rates than copper
  - Allows for longer distance
• Message based protocols work best
  - PCI-Express is a logical bus structure
  - Avoid distributing system memory to the front end
• This is typically the interface which must be integrated into a facility DAQ
  - One size does not fit all but avoid a new custom protocol for each camera
  - Choose one of a small subset of interfaces
• A few examples of data transport protocols
  - UDP
  - Infiniband
  - USB
  - CameraLink
  - Aurora
  - PGP
Data Transport Using UDP

- Most common protocol
- UDP layer easily implemented directly in FPGA firmware
  - DHCP supported added with small embedded processor (Microblaze) or SOC
- Many flavors available
  - 10Mbps - 40Gbps
  - Copper or optical
- Easily attached to computing platform
  - Useful for prototyping and system mobility
  - One of a few options for laptops
- Switches can be added for many to one applications
- Can not be used to transport timing and trigger to front end
- Ethernet frame size limitation
  - Adds overhead and requires software re-assembly
  - Frames are dropped when error is detected
  - Frames can be presented out of order
- Software stack adds latency

XFEL Dual 10Gbps SFP+ Mezzanine Card
Data Transport Using Infiniband

- Primarily used as backend interconnect
  - Server to server
  - Server to storage
- Variety of speeds
  - 2Gbps – 25Gbps per lane
  - Lane bonding up to 12x
- Compatible with Ethernet
  - Ethernet over Infiniband
  - Infiniband over Ethernet
- Heavy hardware offload of protocol layers
- Switches can be added for many to one applications
- Copper links
  - Some optical adapters but implemented as cable replacements
- Difficult to implement in front end
  - Large memories required for some operating modes
  - Limited cores available
- Can not be used to transport timing and trigger to front end
Data Transport Using USB

- Large number of inexpensive interface chips available
- Supported on all desktop and laptop platforms
- Unpredictable Linux kernel support
  - Some versions have buffering flow control issues
  - Unpredictable mapping of peripheral to tty
  - USB resets can cause device to attached on different tty
- Relatively slow
  - USB2 = 480Mbps
  - USB3 = 5Gbps
- Copper interface ties front end electrically to PC or DAQ
- Extremely susceptible to impulse noise
  - Power cycles required to reset and recover device
- Can not be used to transport timing & trigger to front end
- Little advantage over using UDP with many disadvantages
Data Transport Using Camera Link

- Standard interface for many commercial cameras
  - Supported in LCLS
- Medium speed interface
  - Up to 5.44Gbps with multiple cables
- Fiber optic extenders available
  - Optical based frame grabbers
  - No clear standard for optical
- Support for trigger transport
- Can be implemented in front end firmware
- Serial channels for configuration
  - Not a defined protocol
- Camera Link HS is in the early stages
  - Native optical support
  - Higher data rates 2.4Gbps per lane
  - All of the features of existing protocol
  - Promising interface
Data Transport Using Aurora

- Xilinx generated general purpose serial protocol
- Implemented on Xilinx integrated SERDES (MGT/GTP/GTX)
  - Copper or optical
  - Speed defined by SERDES and internal data path
  - 8B/10B or 64/66
- Supports lane bonding for faster rates
- Unlimited frame size
- Built in flow control support
- Lacks data protection
  - Requires user to implement CRC or checksum
- One frame at a time
  - Requires external arbitration to support multiple interfaces
- Can not be used to transport trigger
- Missing important features
Data Transport Using PGP

- PGP = Pretty Good Protocol
- Architecture independent – can be deployed on any 8B/10B SERDES
  - Copper or optical
  - Speed defined by SERDES and internal data path
- Unlimited frame size
- 4 virtual channels, each with separate firmware interface
  - 4 frames in flight at any given time
  - Avoid head of line blocking for configuration messages
- Built in flow control support
- Cell based protocol
  - Large frames segmented into 512byte cells for transport
  - Guaranteed cell ordering
  - CRC protected cells, errors forwarded to end point
  - Per VC arbitration at the cell level
- Low overhead (96% efficient after 8B/10B conversion)
- Can be used to transport timing and trigger
  - Low latency, deterministic trigger transport interface
- Unidirectional & bidirectional protocol
  - Example: 1 downstream link and 4 upstream links
  - Upstream and downstream links can be different line rates
- Supports lane bonding for wider data path
- Standard interface for many experiments
  - LCLS, SID, LSST, ATLAS & Others

SLAC PGPCard
4 lane x 3.125Gbps
Full duplex 9.6Gbps
Data Transport Application Messages

- Listed protocols move data across a link
  - Application layer messages for configuration, monitoring & data
- Register access messages
  - Read & write 32-bit values to/from a specific register address
  - Read and write blocks of data
    - Multiple registers in one transfer
    - Block of memory
  - Synchronous interface for accessing external devices
    - Do not respond until operation is complete or timeout occurs
    - Avoid sleep statements in software to match front end latencies
- Command messages
  - Send resets or software triggers
- Data transport
  - Differentiate image data from register access responses
- A common messaging scheme across multiple protocols
  - Camera Link
  - PGP
  - UDP
- Lower layer protocol may be chosen dynamically
  - Load a different software class and set a firmware synthesis flag
- A common application layer messaging protocol would ease DAQ/detector integration
  - Implemented on a small set of data transport protocols

Diagram:

- Storage → DAQ → Digital Front End → ADC → Detector & Amplifier
- DAQ Software
- Protocol Layer (UDP/PGP/CL)
- Physical Layer
- Physical Layer
- Protocol Layer (UDP/PGP/CL)
- Reg Cntrl → CmdCntrl → Data
- Camera Specific Firmware

High Bandwidth Electronics
What is the DAQ?

- Data processing node
- Parallel processing
- Access to beam information
- System level timing and trigger distribution
- Integrate multiple detector front ends
  - Event building
  - Integrate beam information into data stream
  - Configuration and control
- Complex facility DAQ systems interface to a variety of different detectors
  - Many different detectors types at the same time
- When does a detector need a custom DAQ vs facility DAQ
  - In some cases a small scale detector specific DAQ node is necessary
  - Development DAQ
  - Standalone use
  - Detector specific DAQ node should appear to the facility DAQ as a front end
    - Or define a clear separation between detector and DAQ
    - Ensure compatibility with the target facilities
- Many choices for the data acquisition platform
DAQ Based On Commodity Computing Node

- Rack mount PC based
- Inexpensive and easy to add additional nodes
- Small rack profile
- Can function as a mobile standalone system
  - Easy to add local storage
- Great for simple data checking and forwarding to storage
- Minimal options for hardware acceleration
- Front end interface implemented with adapter cards
DAQ Based Upon Integrated Chassis

- Allows tight coupling of computing power and offload processing
  - Easy to mix hardware processing blades with embedded computing blades
- Scalable
  - Chassis contains the interconnect
  - Shared power & cooling resources
  - Shared management resources
- Most chassis include support for a Rear Transition Module (RTM)
  - Allows for integration of the front end electronics if desired
- Flexible choices for software processing
  - Linux blade server
  - Real Time Operating System (RTOS)
  - System on chip
- Specific chassis choice is not important with proper architecture
  - Architecture can move across platforms
  - Avoid busses
- Options to be shown:
  - VME
  - CPCI
  - ATCA
  - uTCA.4

High Bandwidth Electronics
Integrated Chassis Option – VME

- Parallel bus designed around the Motorola 68000
- 3.3V & 5V power distribution
- RTM support
- Lots of industry cards available
- Old bus technology with a very large install base
  - Mostly chosen due to need to be compatible with existing card base
- Backplane timing support added with VXI extension
- Serial interconnect option added with VSX extension
  - Bulk of connector space exists to support legacy VME bus
  - Dual star backplane
  - PCI-Express
  - Infiniband
  - Ethernet
- Lots of custom backplanes

JLAB Trigger Card
Integrated Chassis Option – Compact PCI

- Industrial application of PCI bus in a chassis
- 6U and 3U high boards
- RTM support
- 3.3V & 5V distribution
- Little support for timing and trigger distribution
- Old bus technology
- Lots of custom backplanes
  - CPCI plus system specific interconnect
- CompactPCI Serial introduced
  - Dual star PCI-Express
  - Full mesh Ethernet
  - Single 12V supply
Integrated Chassis Option - ATCA

ATCA – Advanced Telecommunications Computing Architecture

- Large card dimensions
  - Front card is 280mm deep x 322mm high
  - RTM card is 76mm deep x 322mm high
- Dual-star or full mesh data interconnect
  - PCI-Express
  - Infiniband
  - Ethernet (most common)
- Dual star management interconnect (1Gbps Ethernet)
  - Can be repurposed (timing distribution)
- Redundant 48V power
- Extremely flexible architecture
- Support for clock & trigger distribution
  - Telecom oriented
- Large commercial market
- Widely used
  - LCLS
  - ATLAS
  - Heavy Photon Search
  - LSST
  - XFEL
Integrated Chassis Option – uTCA.4 (uTCA For Physics)

- Based upon uTCA which is the integration of ATCA AMCC cards in a dedicated chassis
  - uTCA is used in medical and military
- 12V power distribution
- Dual wide AMCC front card
  - 150mm x 180mm
- Addition of rear transition module (RTM)
  - Same form factor as front card
- Highly defined RTM connector
  - Limits flexibility
- Integrated shelf manager
- Dual star interconnect
  - PCI-Express (most common)
  - Infiniband
  - Ethernet
- Support for clock and trigger distribution
- Heavy shelf overhead to board space
- Chassis specific to uTCA.4 standard
  - Cost comparable with ATCA yet has less board area
  - Little commercial adoption
Detector and DAQ designs can benefit from the transition to high speed serial based interconnect technologies.
Modern chassis technologies allow for scalable DAQ systems.
System On Chip (SOC) technologies increase the amount of computation power available in the front end and DAQ nodes.
Complex data processing in the front end can reduce data sets but will require a development platform for users as well as experiment level simulations.
Detectors deployed at facilities generating large data sets (FELs) should design their detectors to be easily integrated into facility DAQ.
Detector designers should limit themselves to a small subset of common interface technologies.
Detector designers and DAQ developers can benefit by defining a standard set of protocols for configuration, monitoring and data transport.

• Thank You