DHCAL Back-End

Eric Hazen, John Butler, Shouxiang Wu

Boston University
Outline

- Review of proposed readout
  - DCOL architecture, front-end links, protocol
- Proposed DCOL implementation
  - hardware/firmware event building
- Software event builder requirements
- Protocols
- Prototyping
- Status/plans
DHCAL Readout (Review)
(one plane shown)

Front-End Boards
Kapton Cables
Data Concentrator
Ribbon Cable
Data/Super Concentrator
Data Collector
Trigger Timing Module
VME Crate

Details may change!
Space is available for 12+1 RJ-45 connectors on a 6U panel (using quad shielded connectors)

12 inputs for super-concentrators

1 input from master trigger/timing module
Data Collector (DCOL)
(Review)

- **SERDES**
- **FPGA**
  - **Spartan 3E** (2 required)
- **SDRAM** (if needed)
- **Event Builder, VME64**
- **Flash Memory**
- **CPLD**
- **JTAG**
- **5V Buffers**
- **VME**

**Total of 12 I/O cables**

- **Trigger, Timing input**
  - **RJ-45**

**Low-skew fanout of clock, trigger, timing**

**LVDS**

**RJ-45**
Proposed Link Technology
(Review)

Front-End Electronics

FPGA (Spartan-3E)

Capacitative or GMR Digital Isolator

Shielded Twisted Pair Cable (4 pairs)

Data Collector

FPGA (Spartan-3E)

LVDS

Dedicated Pair for test pulse

Power to far end

Separated power/GND plane for isolated electronics (powered from DCOL end)
3.3-V / 5-V HIGH-SPEED DIGITAL ISOLATORS

FEATURES
- 4000-V_{peak} Isolation
  - UL 1577, IEC 60747-5-2 (VDE 0884, Rev. 2)
  - IEC 61010-1 and CSA Approved
  - 50 kV/μs Transient Immunity Typical
- Signaling Rate 0 Mbps to 150 Mbps
  - Low-Propagation Delay
  - Low-Pulse Skew (Pulse-Width Distortion)
- Low-Power Sleep Mode
- High-Electromagnetic Immunity
- Low-Input Current Requirement
- Failsafe Output
- Drop-In Replacement for Most Opto and Magnetic Isolators

APPLICATIONS
- Industrial Fieldbus
  - Modbus
  - Profinet
  - DeviceNet™ Data Buses
  - Smart Distributed Systems (SDS™)
- Computer Peripheral Interface
- Servo Control Interface
- Data Acquisition

DESCRIPTION
The ISO721, ISO721M, ISO722, and ISO722M are digital isolators with a logic input and output buffer separated by a silicon oxide (SiO₂) insulation barrier. This barrier provides galvanic isolation of up to 4000 V. Used in conjunction with isolated power supplies, these devices prevent noise currents on a data bus or other circuits from entering the local ground, and interfering with or damaging sensitive circuitry.

A binary input signal is conditioned, translated to a balanced signal, then differentiated by the capacitive isolation barrier. Across the isolation barrier, a differential comparator receives the logic transition information, then sets or resets a flip-flop and the output circuit accordingly. A periodic update pulse is sent across the barrier to ensure the proper dc level of the output. If this dc-refresh pulse is not received for more than 4 μs, the input is assumed to be unpowered or not being actively driven, and the failsafe circuit drives the output to a logic high state.

FUNCTION DIAGRAM

Diagram showing the isolation barrier, input and output channels, and various stages of signal processing and isolation.
Data Encoding
(one possible scheme)
(Review)

100ns Frame

Rising edges occur at steady 40MHz rate
Bit encoding by pulse width

First bit always '1', pattern '1000' used for synchronization

This type of code would be easy to implement in a Xilinx FPGA
DLL (delay-locked loop) in FPGA would recover clock with ~0 skew
Delayed clock output would sample bits to recover data stream
Return Data
(from Super-Concentrator to Data Collector)
(Review)

100ns Frame

- 1 0 0 0 -
- 1 1 0 0 -

DAQ data 0
DAQ data 1
Slow Control Data

Remaining 3 bits used to send data at 10Mbit/sec

For data readout, two bits could be assigned to provide 20Mbits/sec.

If more bandwidth is required, a clock rate higher than 40MHz could be used.
Assumptions/Requirements:

- All hits time-stamped to 100ns, synch'd clocks
  - Events built from events with hits matching to within 1-2 clocks
- Two operation modes:
  - Global trigger from beam counters
  - Self-trigger mode for cosmics
- Compatibility with CALICE DAQ is desirable
  - Nicely-packaged events with 1:1 correspondence to global triggers
Data Collector

- Proposed Hardware Implementation:
  - All data received and stored as packets:
    - Timestamp (24 bits), Address (16 bits), Hit data (64 bits)
      - Total 13 bytes (round up to 16 for easy hardware processing?)
  - Packets are grouped in buffers by matching timestamp
    - Buffer large enough to hold max size event (all chips hit)
  - DCOL makes buffers available for VME transfer
    - Buffers read out in time order
    - VME registers provide buffer word count and timestamp
    - Buffer data transferred from fixed address (using 64BLT)
  - Monitoring registers (scalers) count:
    - Packets (per link), triggers, errors, etc.
    - Does this need to be stored on an event-by-event basis?
Event Builder

• Software event builder tasks:
  - Combine buffers into a single event
    • Read buffers from all DCOL with nearly matching timestamps
      - Two single word reads plus a BLT for each buffer
  - Read event number and other info from trigger
  - Read monitoring scalers if needed
  - Ship event to DAQ
  - Is this OK, or should the hardware do more of the work?
Protocols - Draft from ANL

- **Slow control:**
  - 10Mbit/sec stream send/receive on 2nd bit of front-end link 4 bit frame
  - Allow read/write with individual or broadcast access to any DCAL chip or data concentrator

- **Hit data:**
  - 20Mbit/sec stream received on bits 3,4 of frame
  - Individual data packets from each DCAL chip
  - Sent at full speed by data concentrators; no back-pressure
What do we need for DCOL debugging?

- 6U VME crate, PCI bridge, PC
- Full prototype DCOL VME module
- Front-end Simulator
  - Daughterboard or dedicated test board
- Trigger/Clock Simulator
  - Could be combined with above, or another DCOL module with special firmware
Status - Plans

- Finish draft specification for DCOL
  - need a little more input from ANL
- Review with Mr Wu, start PCB design in December.

Open issues:
- FE-link implementation... daughter-board or design-in on front-end?
- What to do about trigger and CALICE CRC module interface?