A Read-Out Driver for Silicon Detectors in ATLAS

Trevor Vickey

University of Wisconsin, Madison
ATLAS Collaboration

September 26, 2006

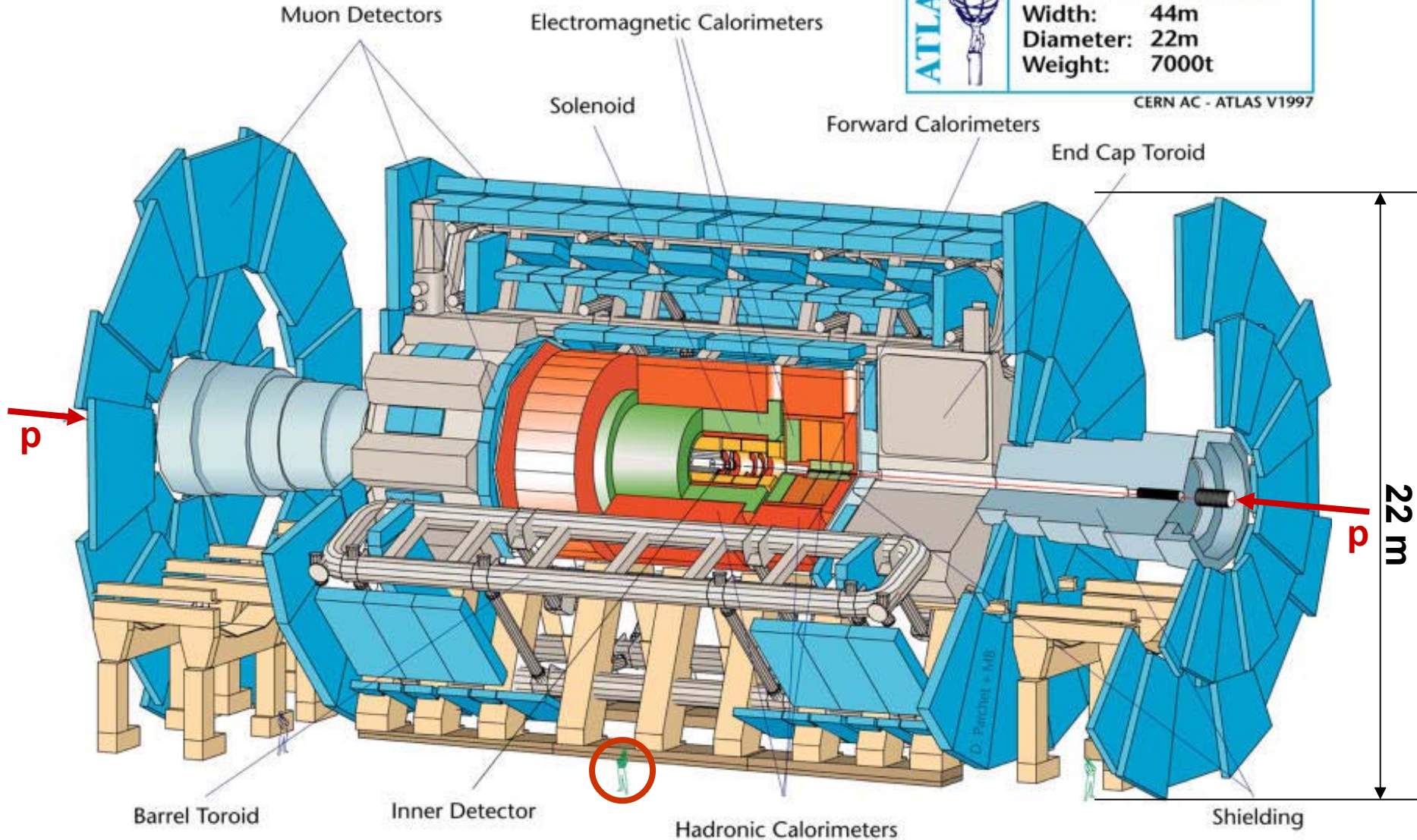12th Workshop on Electronics for the LHC and Future Experiments

A Toroidal LHC ApparatuS (ATLAS)

General purpose experiment at the LHC

2

The ATLAS Inner Detector

Critical for charged-particle tracking, *B* hadron and τ tagging

▶ Consists of three detector sub-systems (and more than 86 million channels)

Transition Radiation Tracker (**TRT**)
(Axial Barrel Straws and Radial End-Cap)

$\eta = 0$

$\eta = 2.5$

p

p

Pixel Detector
(3 Barrels and 3 x 2 End-Cap Disks)

Semiconductor Tracker (**SCT**)
(4 Barrels and 9 x 2 End-Cap Disks)

Pixel Tracker Module

Each Pixel Module has

- 16 Front-End ICs
  - 2880 channels / IC
  - 18 columns x 160 rows
  - Bump bonds to sensor
- One Module Controller Chip
  - Collects data from the 16 FE chips
  - Translates commands into chip signals

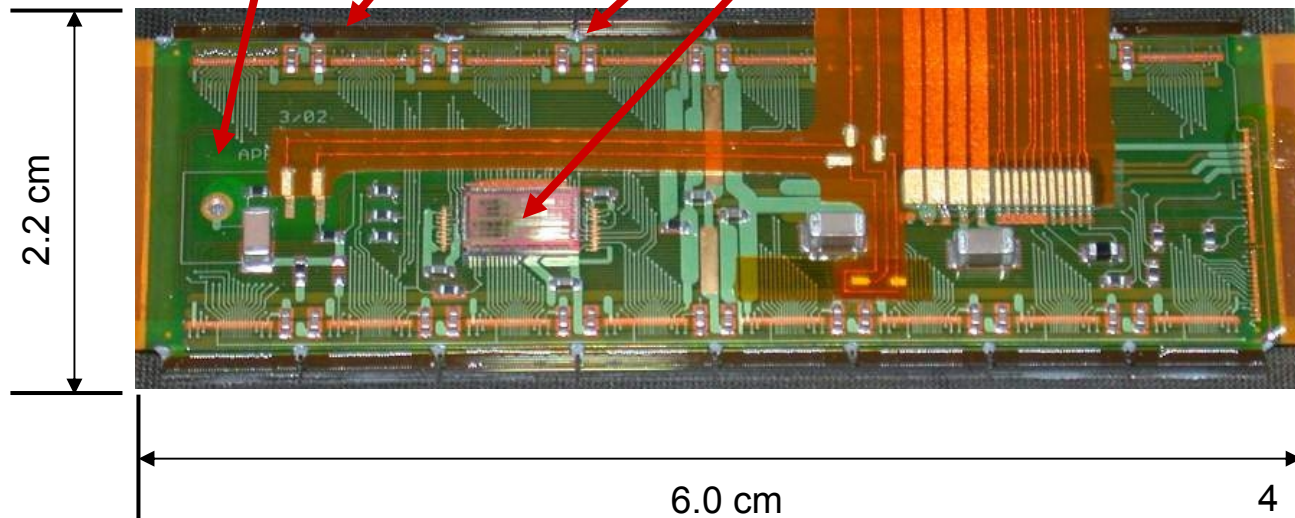## Total Number of Modules

- 1456 (Barrels)
- 144 x 2 (End-caps)

Read-out Rate:

- B-Layer 160 Mbit/s
- Layer-1 and Endcaps 80 Mbit/s
- Layer-2 40 Mbit/s

There are over 80 million channels in the ATLAS Pixel Tracker

ATLAS Pixel Module (46080 channels) Pixel Size: 50 x 400 µm

Flexible PCB   Front-End ICs (x16)   Sensor (between PCB and IC)

Module Controller Chip (MCC)

2.2 cm

6.0 cm

4

Semiconductor Tracker Module

Each side of an SCT Module has

▶ Two silicon sensors
- • Manufactured by Hamamatsu
- • 768 instrumented strips at an 80 µm pitch

▶ An array of six binary readout chips
- • ABCD3TA ASICs
- • Discriminator
- • Pipeline
- • Data Compression Logic
- • Read-out Buffer

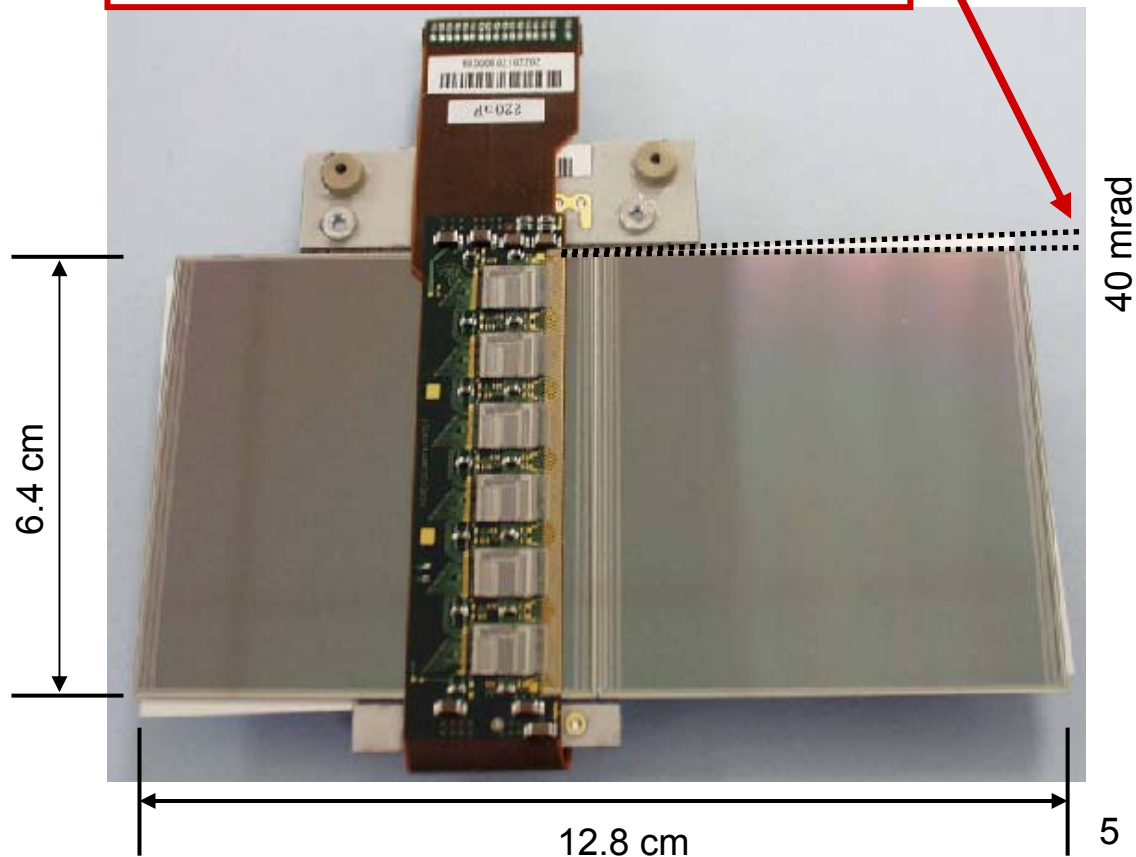## Total number of modules:

▶ 2112 (Barrels)
▶ 988 x 2 (End-caps)

Read-out Rate:

▶ Barrels and End-caps 40 Mbit/s

There are over 6.2 million channels in the ATLAS Semiconductor Tracker

40 milliradian stereo angle between sides provides resolution in the axial direction

ATLAS SCT Module (1536 channels) Pitch: 80 µm

40 mrad

6.4 cm

12.8 cm

5

Data Acquisition Hardware

The ROD (Read Out Driver) is central to the DAQ chain

▶ Hardware common between Pixel and SCT Trackers (firmware differences exist)

Optical Fibers for:
→ **Clock, Config, Trigger**
← **Event Data**

VME Crate

(Single Board Computer)

HOST

SBC

Configuration
Histograms

ROD — **BOC** — 48 — **MODULE**

ROD — **BOC** — 48 — **MODULE**

ROD — **BOC** — 48 — **MODULE**

Up to 16 RODs
(Read Out Driver)
per VME Crate

ROD — **BOC** — 48 — **MODULE**

(Back Of Crate)

TTC — **TIM**

(Timing, Trigger & Control) (TIming Module)

Timing and Trigger

S-Link

ROS

(Read Out Subsystem)

To Event-Builder
and Level-2

6
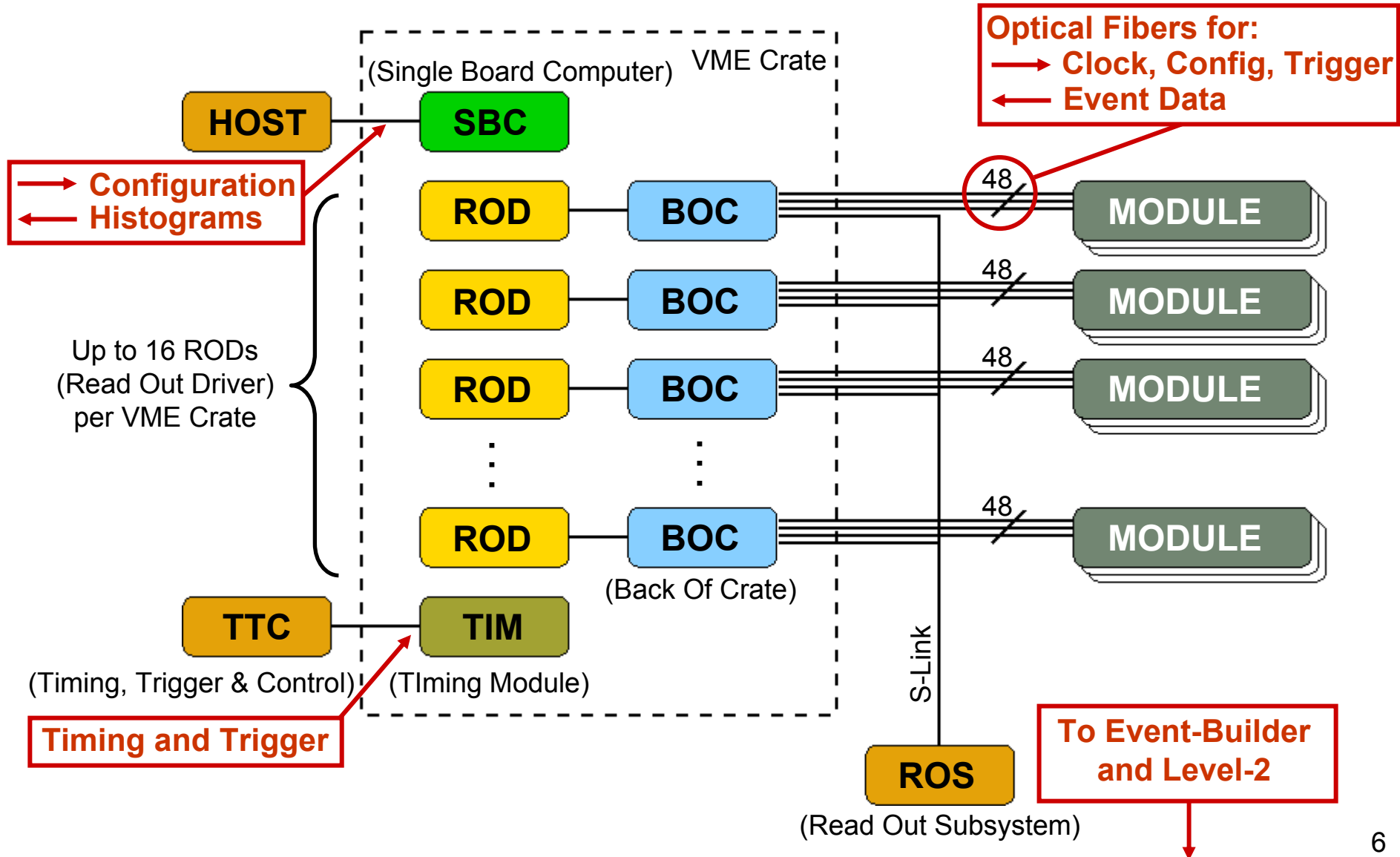
The Silicon Read-Out Driver (ROD)

Primary purpose: Module configuration, Trigger propagation, Data formatting

▶ A hybrid of FPGAs and DSPs

←	**Configuration & Triggers**
→	**Event Data**
←	**Backpressure / Halt Output**

MODULE

BOC
(Back Of Crate)

S-Link

ROS
(Read Out Subsystem)

To Event-Builder and Level-2

Formatters

Event Fragment Builder

Router & Slave DSPs

Controller & Master DSP

ROD

TIM
(TIming Module)

SBC
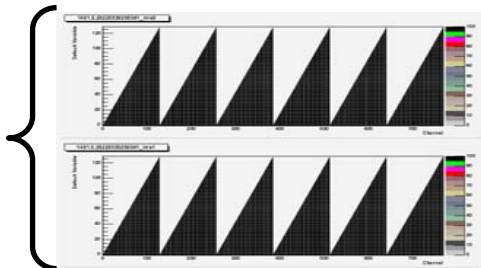(Single Board Computer)
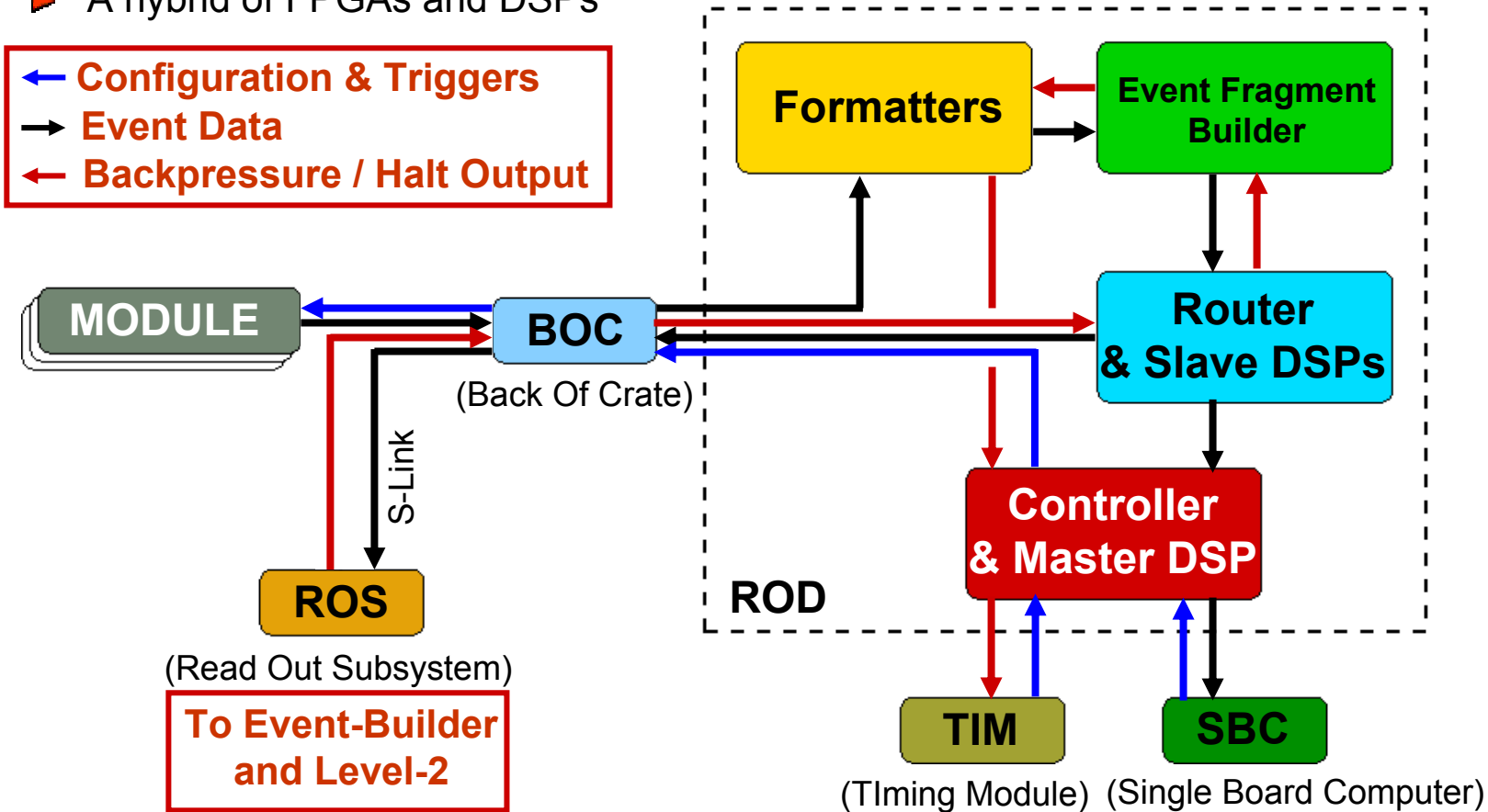
Secondary purpose: Calibrations / Monitoring

▶ FPGAs for time-critical functions (Event Data Path)

▶ DSPs for configuration, ROD control, calibrations and monitoring
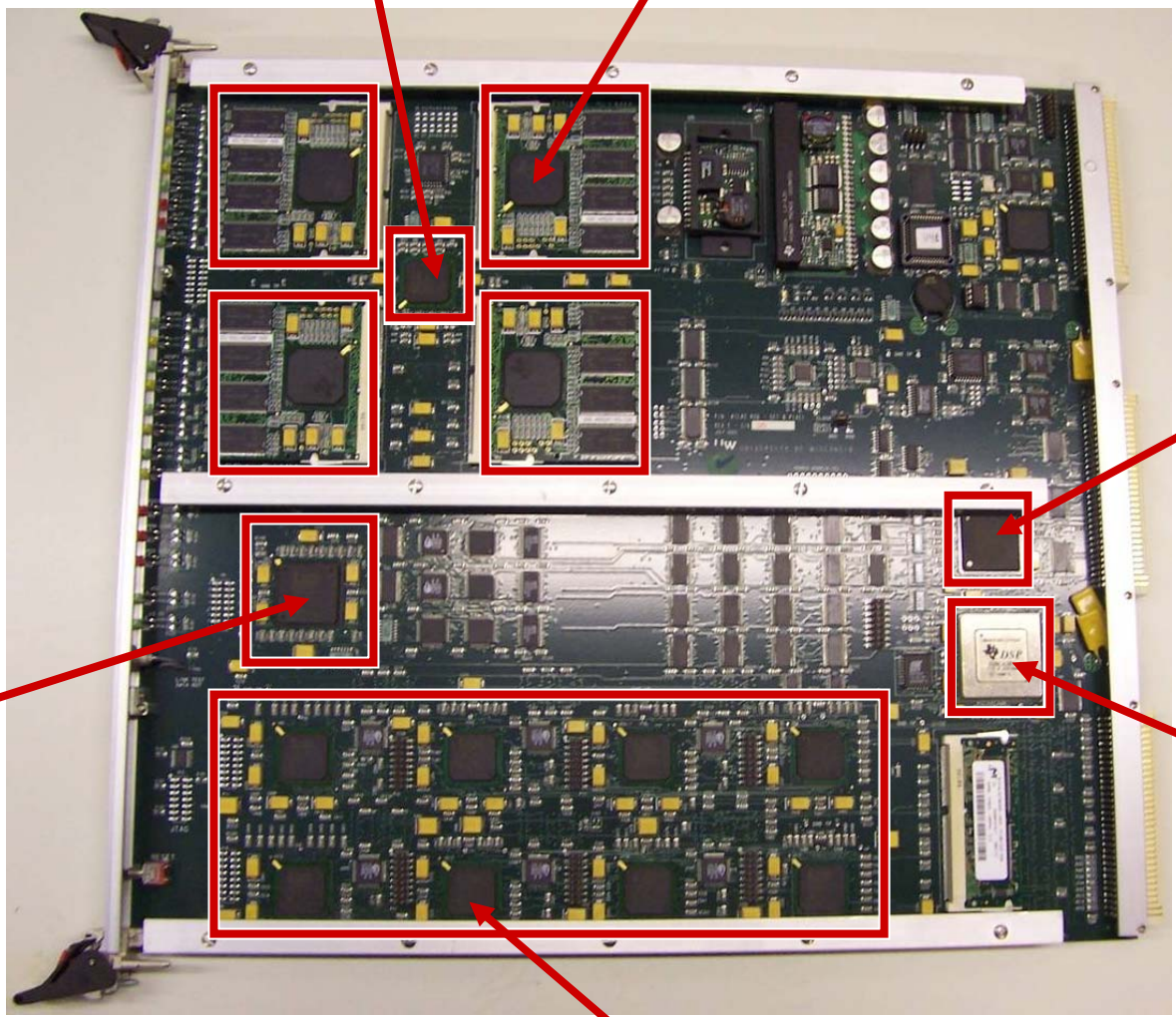
Calibration Histograms

The Silicon Read-Out Driver (ROD)

Router FPGA

Slave DSPs

9U VME

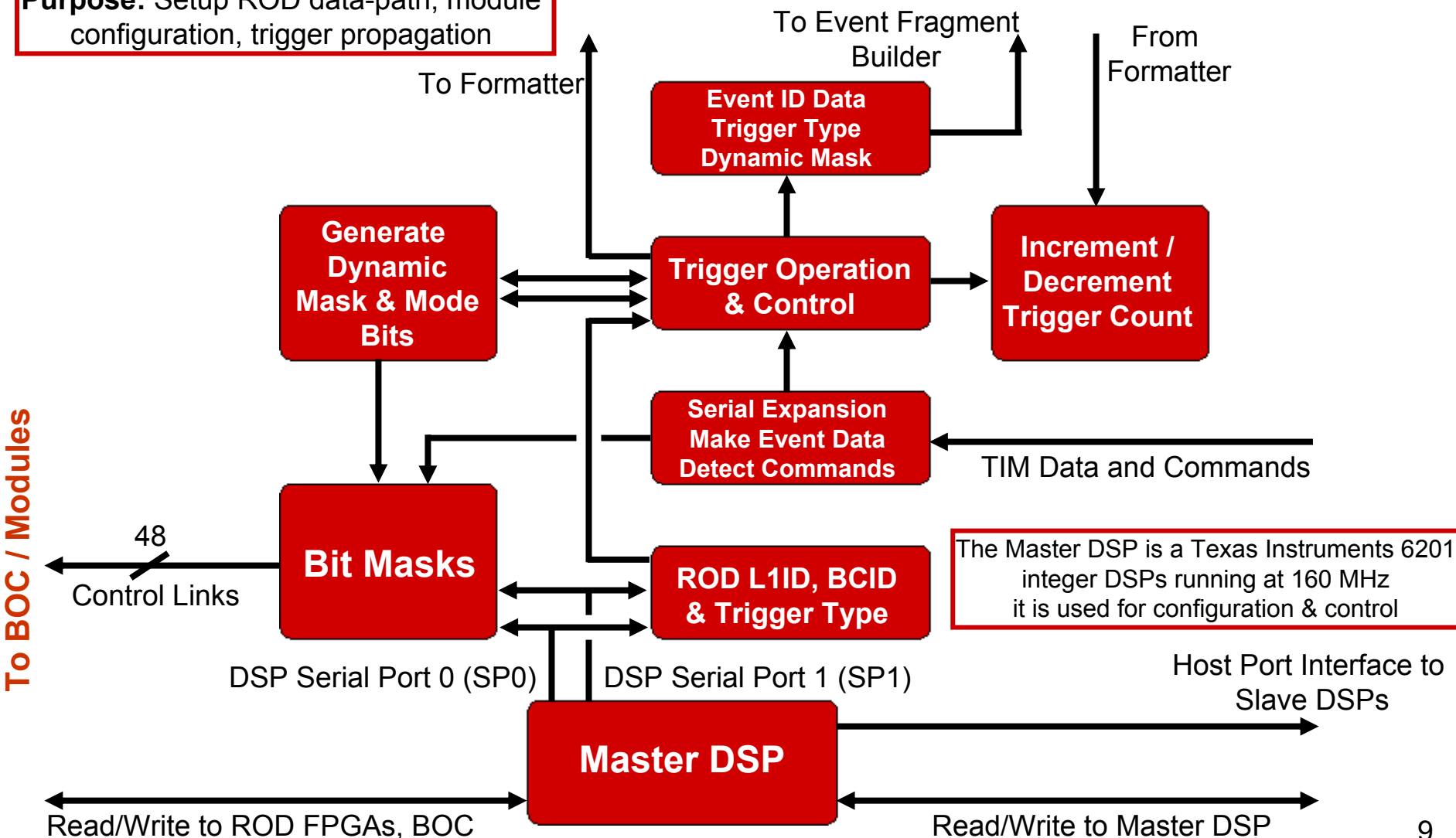**Controller FPGA**

Event Fragment Builder FPGA

Master DSP

Formatter FPGAs

ROD Controller (FPGA) and Master DSP

The Master DSP (MDSP) has ROD and BOC registers connected to one of its EMIFs (External Memory InterFace)

Purpose: Setup ROD data-path, module configuration, trigger propagation

To Event Fragment Builder

To Formatter

From Formatter

Event ID Data Trigger Type Dynamic Mask

Generate Dynamic Mask & Mode Bits

Trigger Operation & Control

Increment / Decrement Trigger Count

Serial Expansion Make Event Data Detect Commands

TIM Data and Commands

To BOC / Modules

48

Control Links

Bit Masks

ROD L1ID, BCID & Trigger Type

The Master DSP is a Texas Instruments 6201 integer DSPs running at 160 MHz it is used for configuration & control

DSP Serial Port 0 (SP0)

DSP Serial Port 1 (SP1)

Host Port Interface to Slave DSPs

Master DSP

Read/Write to ROD FPGAs, BOC

Read/Write to Master DSP
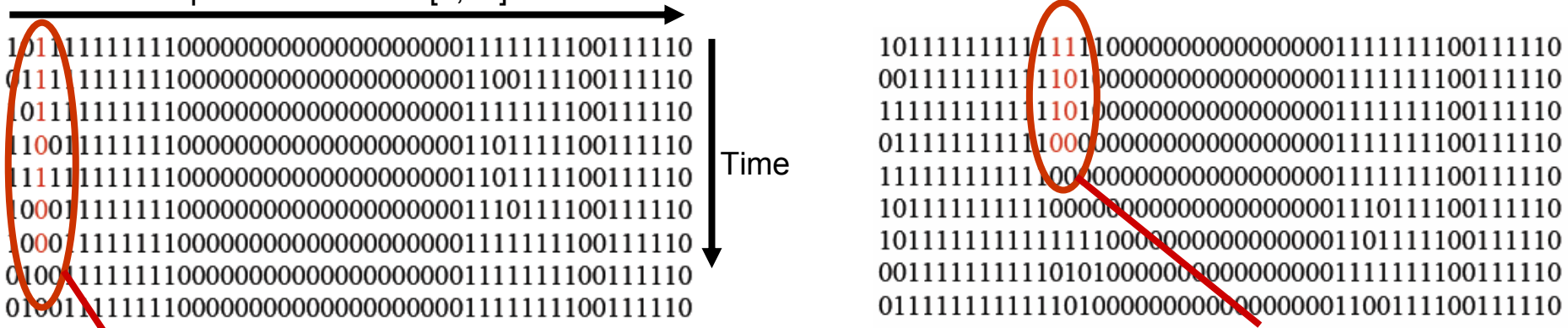
9

Link Data Format

The Pixel MCC and the SCT ABCD3TA each return a bit stream

- ▶ The BOC splits these into individual 40 MHz streams
- ▶ The ROD Input Links receive the data streams from the BOC
  - 40 MHz output: The Pixel Barrel Layer-2 and the SCT
  - 80 MHz output: The Pixel Barrel Layer-1 and the Pixel Endcaps
  - 160 MHz output: The Pixel Barrel B-Layer

Event Header: 11101

Thanks to Andreas Korn (LBNL) for the Input Memory FIFO captures

Input Link Number [0,47]

```
10111111111000000000000000000111111100111110
01111111111000000000000000000110011110011110
01111111111000000000000000000111111100111110
11001111111000000000000000000110111100111110
11111111111000000000000000000110111100111110
10001111111000000000000000000110111100111110
10001111111000000000000000000111111100111110
01001111111000000000000000000111111100111110
01001111111000000000000000000111111100111110
```

```
10111111111111100000000000000000111111100111110
00111111111110100000000000000000111111100111110
11111111111110100000000000000000111111100111110
01111111111110000000000000000000111111100111110
11111111111110000000000000000000111111100111110
10111111111110000000000000000000110111100111110
10111111111111111000000000000000110111100111110
00111111111110101000000000000000111111100111110
01111111111110100000000000000000110011100111110
```

Time

40 MHz Mode

80 MHz Mode

```
10111111111101110000000000000000111111100111110
01111111111110001000000000000000110011100111110
10111111111110000000000000000000111111100111110
11111111111111101000000000000000110111100111110
11111111111111100100000000000000110111100111110
10111111111111111000000000000000110111100111110
10111111111110101000000000000000111111100111110
01111111111111000000000000000000111111100111110
01111111111111000000000000000000111111100111110
```
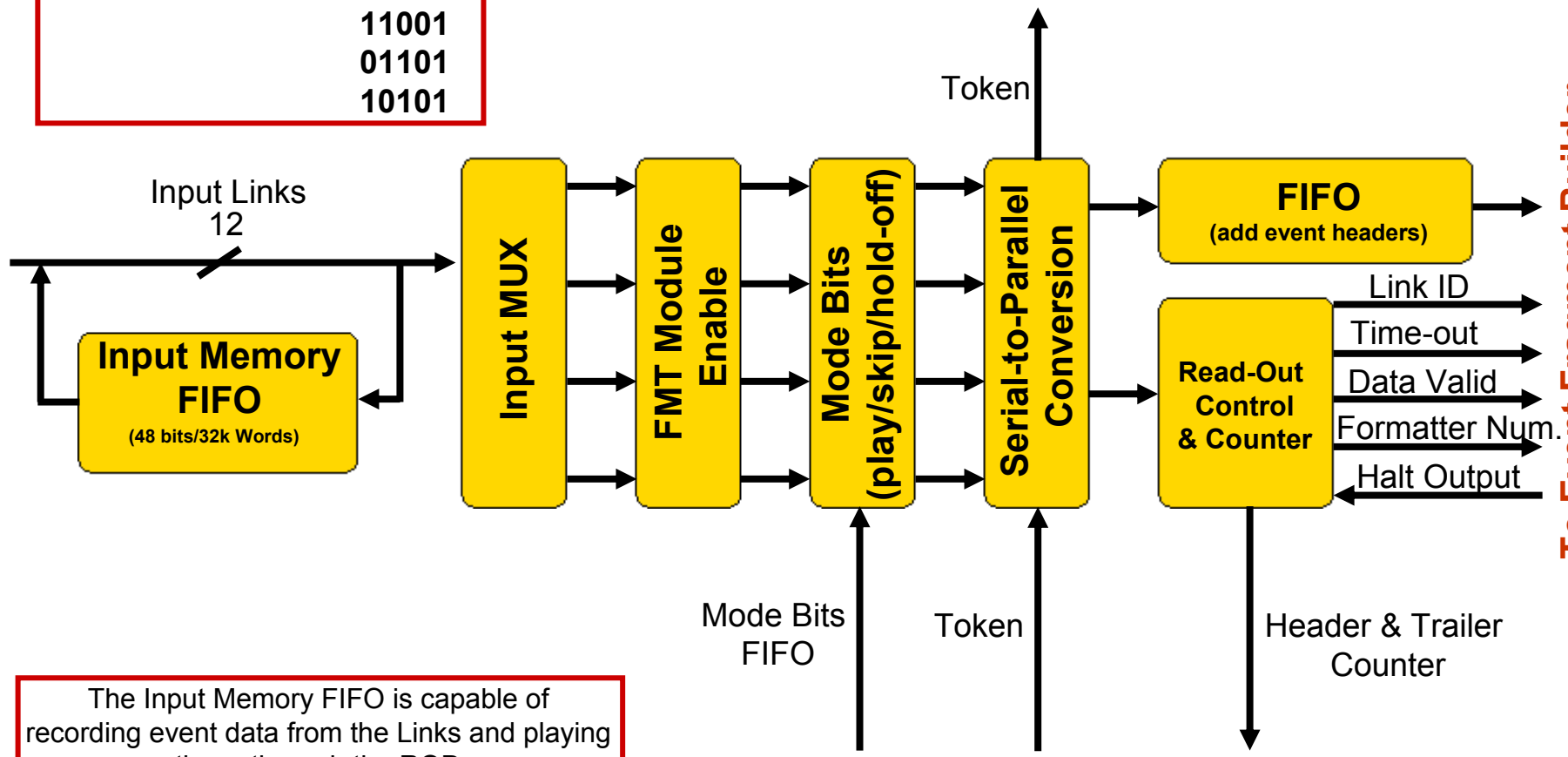
160 MHz Mode

10

ROD Formatter (FPGA)

8 Formatters on the ROD; capable of header error detection

Good Header: **11101**
Acceptable Headers: **11100**
11111
11001
01101
10101

Purpose: Serial-to-parallel conversion (word data), detect module packet errors

From BOC / Modules

Input Links
12

Input Memory FIFO
(48 bits/32k Words)

Input MUX

FMT Module Enable

Mode Bits (play/skip/hold-off)

Token

Serial-to-Parallel Conversion

FIFO
(add event headers)

Read-Out Control & Counter

Link ID
Time-out
Data Valid
Formatter Num.
Halt Output

To Event Fragment Builder

Mode Bits
FIFO

Token

Header & Trailer
Counter

To/From Controller

The Input Memory FIFO is capable of recording event data from the Links and playing these through the ROD
(a useful debugging feature)

11

ROD Event Fragment Builder (FPGA)

Engine 0

Halt Output

To/From Formatter

Formatter Num. (2 bits)
Link Num. (4 bits)
Data (32 bits)
Time Out
Data Valid

Increment / Decrement L1ID

L1/BC ID Check

Error Check

Format & Count

Link Num. (6 bits)
Time-out Error
L1ID Error
BCID Error
Data (32 bits)

FIFO

Purpose: Collect Formatter output, check L1 and BC IDs, count errors, generate Event Header & Trailer

Output FIFO Data (43 bits)

EventMemA

Error Summary Word

To/From Controller

Event Data & Trigger Type

L1 & BC IDs

Dynamic Mask

Header/Trailer Generator

Header and Trailer Data (32 bits)

EventMemC

Error Summary Word

EventMems A, B and C are useful for debugging work

EventMemB

To/From Router

Engine 1

To/From Formatter

Data Valid
Time Out
Data (32 bits)
Link Num. (4 bits)
Formatter Num. (2 bits)

Increment / Decrement L1ID

L1/BC ID Check

Error Check

Format & Count

Data (32 bits)
BCID Error
L1ID Error
Time-out Error
Link Num. (6 bits)

FIFO

Output FIFO Data (43 bits)

Halt Output

12

Error Counting and Handling

One of the primary purposes of the Formatters and the EFB:

▶ To identify and count errors found inside of the event data

▶ The types of errors are stored inside of the event header and trailer words:

**Example for the SCT event format
is shown**

Name	Bits [15:0] or [31:16]
Event Header	001PTLBKxMMMMMMM
Event Trailer	010ZHVxxxxxxxxxx
Single Hit	1FFFFCCCCCCCxfx0
Double Hit	1FFFFCCCCCCCsfx1

B = BC ID Error
C = Cluster base address
F = Front-End number
f = Error in the first hit
H = Header/Trailer Limit Error
K = Condensed mode
L = L1 ID Error
M = Link Number
P = Preamble Error
s = Error in the second hit
T = Time-out Error
V = Data Overflow Error
x = (filled with 0 by the ROD)
Z = Trailer Bit Error

ROD Router (FPGA) & Slave DSPs

Halt Output

XOn / XOff S-Link

Event Header and
Trailer Data (32 bits)

Dsp Halt Output

S-Link Data

Read, format and direct the event data

Output FIFO (43 bits)

Purpose: Route formatted data to Level-2 and/or Slave DSPs (histogram)

Data Valid

Error Format Data

2 Traps	2 Traps	2 Traps	2 Traps
Event Type: ATLAS, ROD or TIM	Event Type: ATLAS, ROD or TIM	Event Type: ATLAS, ROD or TIM	Event Type: ATLAS, ROD or TIM
DMA Transfer Engine 256 32-bit word blocks	DMA Transfer Engine 256 32-bit word blocks	DMA Transfer Engine 256 32-bit word blocks	DMA Transfer Engine 256 32-bit word blocks
SDSP 0	SDSP 1	SDSP 2	SDSP 3

Texas Instruments 6713 floating point DSPs running at 220 MHz for monitoring and calibration histogramming

ROD Communication: Registers

These communication registers are blocks of 32-bit words at the start of the DSP's internal memory

The Master DSP regularly checks Communication Registers

- This is done while inside of the main loop (infinite) of the DSP code
- MDSP polls these registers watching for requests from the Host (user)

Quite a few different types of Communication Registers

- General Status (number of current tasks running, is event trapping engaged, etc.)
- Dedicated Histogramming Registers (scan statistics, coordinate actions)
- Event Trapping Command & Status Registers (set trapping parameters, event statistics)
- General Purpose Command Register (receive host requests)
- Memory-map Register (set at initialization; points to the memory map in external memory)
- Task Status Register (indicates current state of tasks running on the ROD)
- Registers specific to Inter-DSP communication
- Counter Register (ticks over with every iteration of the main loop)
- Reserved Misc. registers for debugging purposes

ROD Communication: Lists and Primitives

ROD FPGA registers are mapped in the Master DSP memory space

▶ A system of primitives allows the MDSP to remain in control of its memory while receiving commands from the host

▶ Primitives exist for:
- Reading and writing FPGA registers
- Reading and writing regions of Master DSP memory
- Load or modify Silicon Module configurations
- Starting the Slave DSPs

▶ The MDSP can send primitive lists to the Slave DSPs
- Start calibration histogramming

▶ In general, primitives are executed once by the DSP

▶ DSP software is versatile enough to handle new primitives
- These are simple and easy to write

Primitive replies (if any)

▶ Stored inside of a special "reply" buffer

▶ Reply buffer is received by the Host after primitive execution

Primitive structure

List Header
- List Length
- List Index
- List Primitive Count
- List Version

Primitive 1
- Primitive Length
- Primitive Index
- Primitive ID
- Primitive Version
- Primitive Payload Data

Primitive N
- Primitive Length
- Primitive Index
- Primitive ID
- Primitive Version
- Primitive Payload Data

List Trailer
- List Length
- List Checksum

Primitive List State Machine

Primitive List execution: Host-to-MDSP and MDSP-to-SDSP

Primitive List Execution
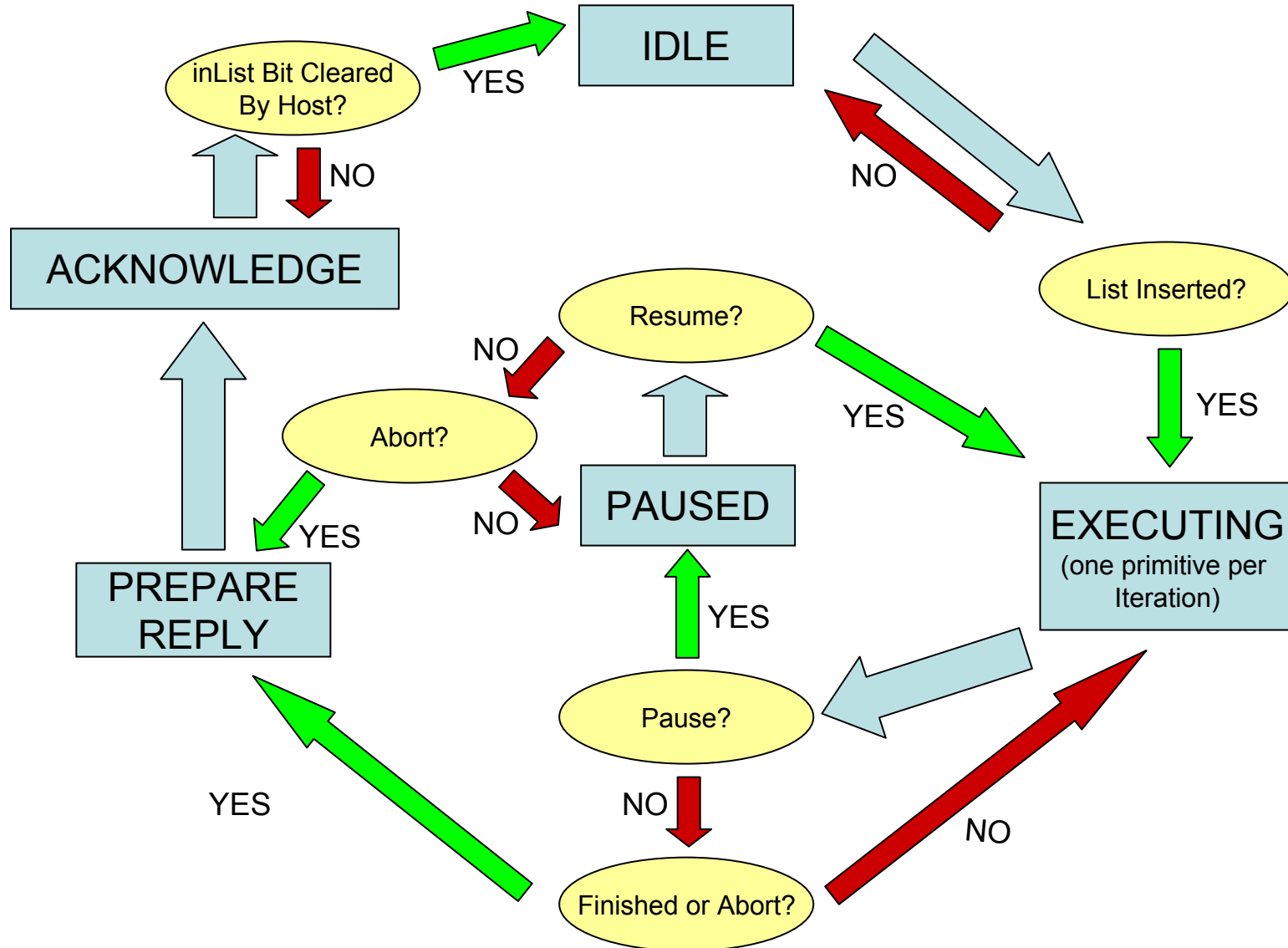
Text Buffer State Machine

Text Buffer reply: MDSP-to-Host and SDSP-to-MDSP

▶ Error, Info, Diagnostic and Transfer (MDSP is a way-station for SDSP text buffers)

Text Buffer Transmission
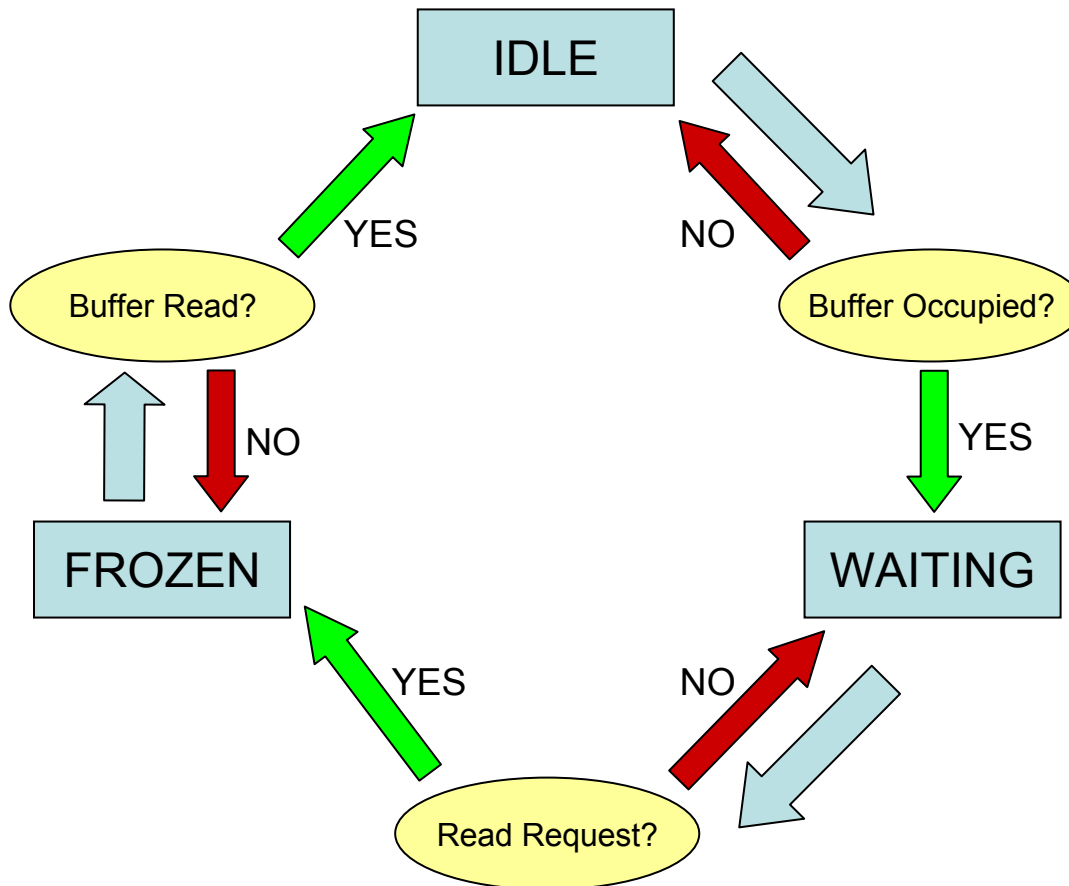
ROD Communication: Tasks

Primitives are typically executed once, Tasks execute ROD functions over an extended period of time

▶ These operate independent of the Primitive Lists
  - The ROD can still process Primitive Lists while various Tasks are running

▶ Tasks start and stop by the Host (or MDSP) sending Primitives
  - Run until they complete (or are halted manually)

▶ Examples of tasks:
  - Event Trapping Task (e.g., in preparation for calibration histogramming)
  - Histogramming Task (issuance of triggers, processing and binning events)

Pixel Calibration Scans

A couple of common Pixel Scans:

- ▶ **Threshold** (on chip charge-injection for each individual pixel; scan the number of hits for each injected charge to obtain the discriminator threshold)
- ▶ **Noise** (a threshold scan without charge injection to measure noise)

Threshold Scan result shown for a single Pixel Module
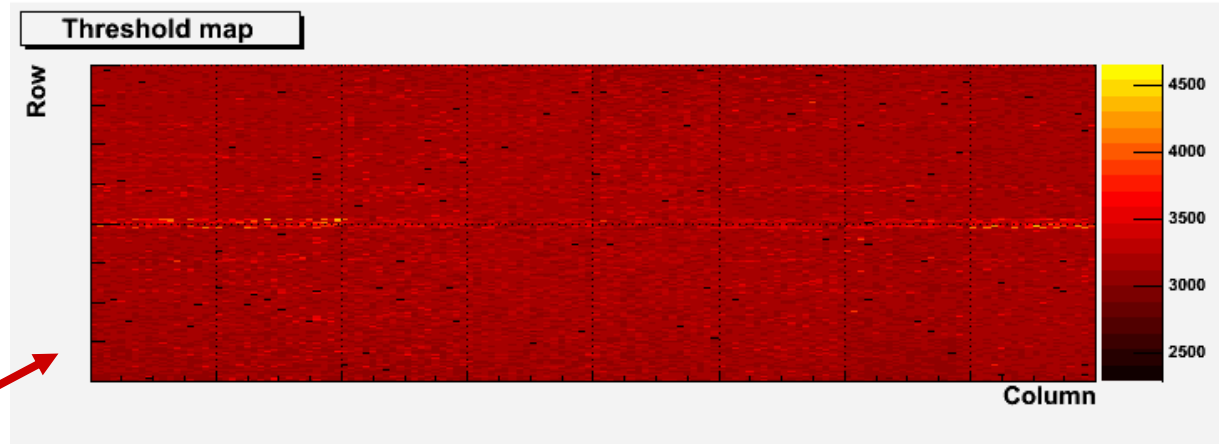
Threshold for each pixel (note the 16 FE ICs)
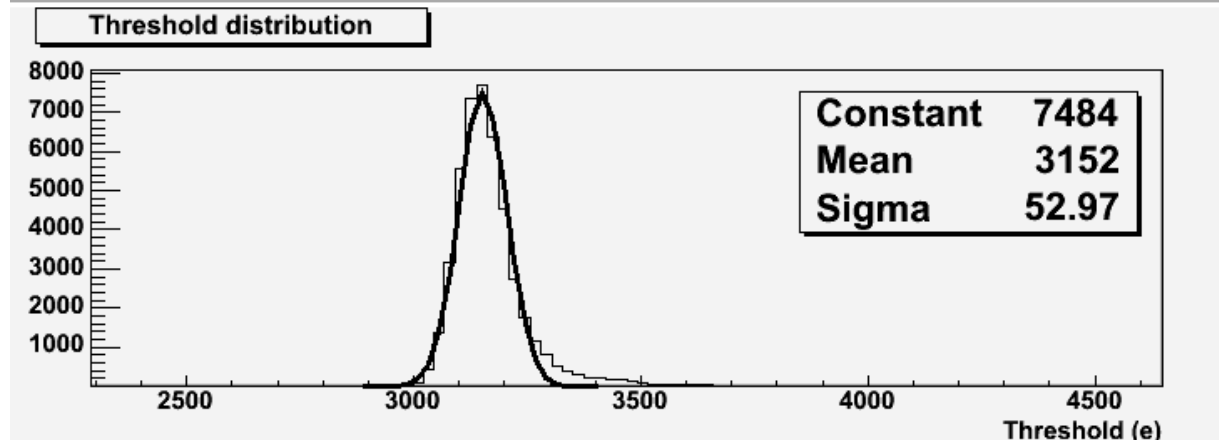
SCT Calibration Scans

Some of the commonly run SCT Calibration Scans:

- ► **Rx Threshold Test** (optimize the Rx threshold value in the BOC data-receiver chip)
- ► **NMask Test** (demonstrate that the chip mask register functions properly)
- ► **Pipeline Test** (scan the chip pipeline for defects; stuck on or off)
- ► **Full Bypass Test** (does the chip bypass feature function properly)
- ► **Noise Occupancy Test** (a threshold scan without charge injection to measure noise)
- ► **Synchronous Trigger Noise Test** (triggers distributed synchronously across the system)

N-Mask Scan result shown for a single SCT Module
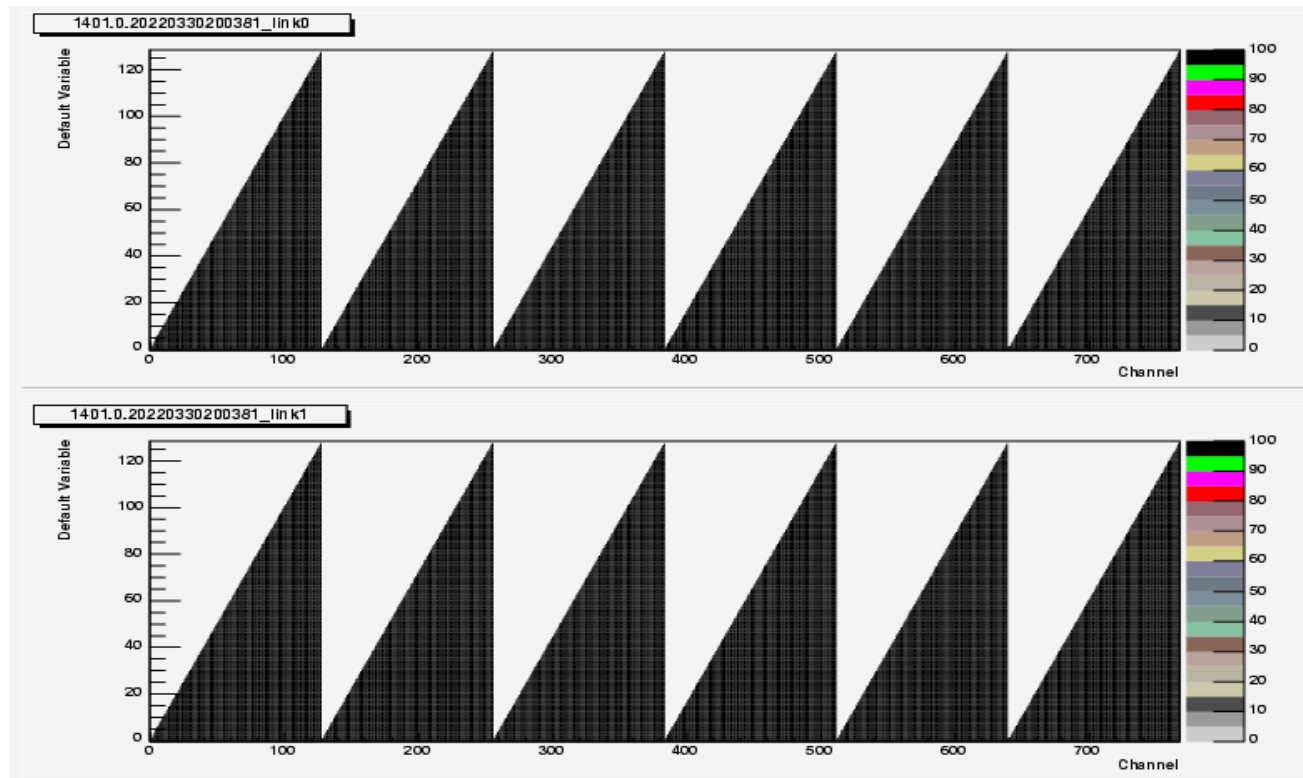
ROD use during detector production / assembly

Calibration scans have been routinely used during module production

- ▶ Verify module functionality before and after mounting or transport
- ▶ Classification of modules
  - • Rank by the number of defects; only mount and install the very best

This phase also gave feedback to ROD developers (and still continues to!)

- ▶ A few DSP software and FPGA Firmware problems have been uncovered this way
  - • Some unique bugs appeared when stressing the ROD with a full complement of modules

Cosmics Data-Taking

TRT/SCT combined run in May 2006

- ▶ ROD performed well; successfully collected more than 400k events
- ▶ Useful DAQ/Detector shake-down, data for efficiency and alignment studies

On-ROD Histogramming used for timing in the SCT during cosmics running (i.e., delay the signal by the correct amount)
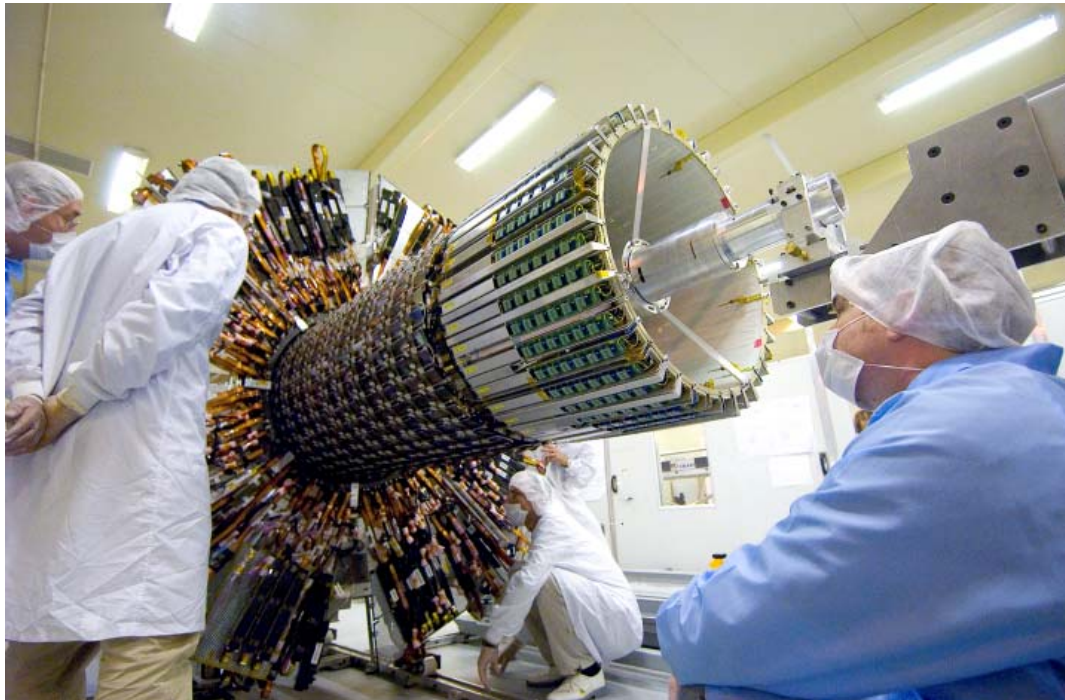
A total of 504 SCT Modules into 12 RODs were used during the combined cosmics data-taking run

Scintillator

SCT

Noise
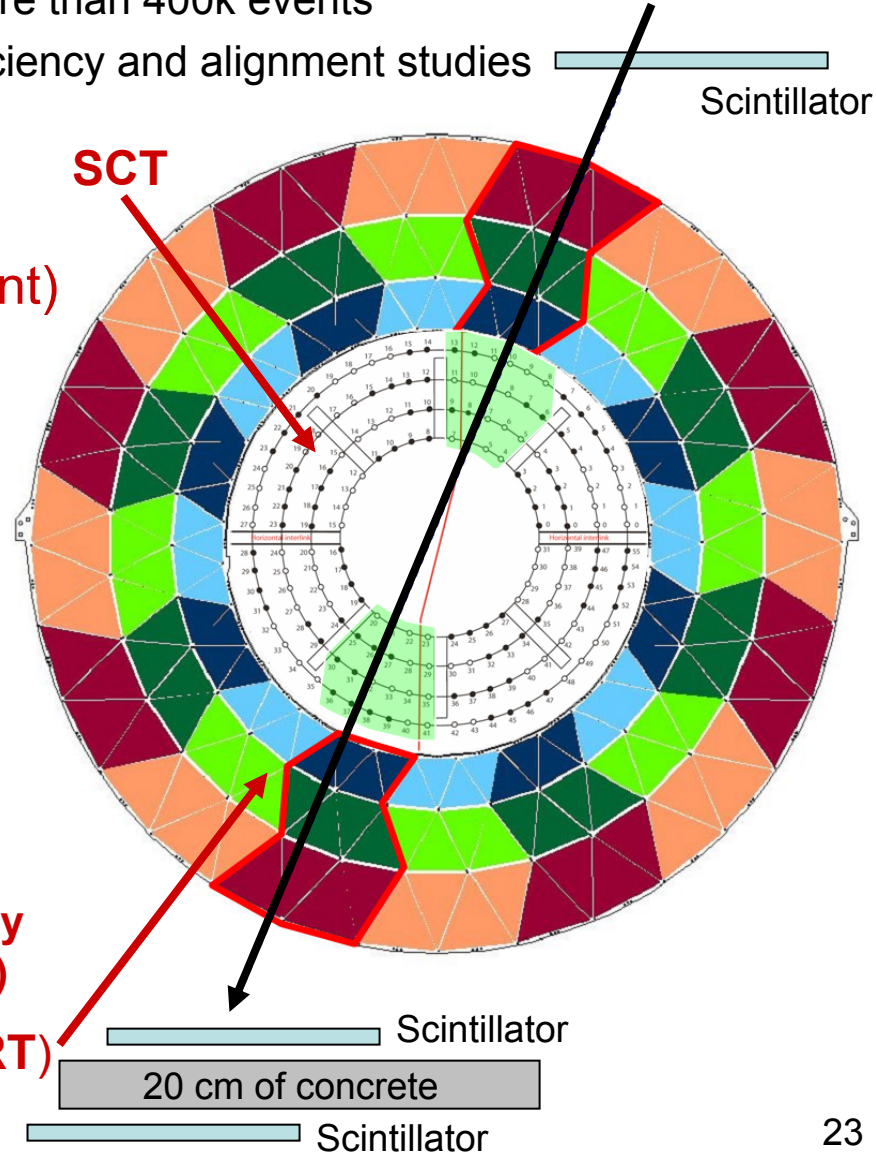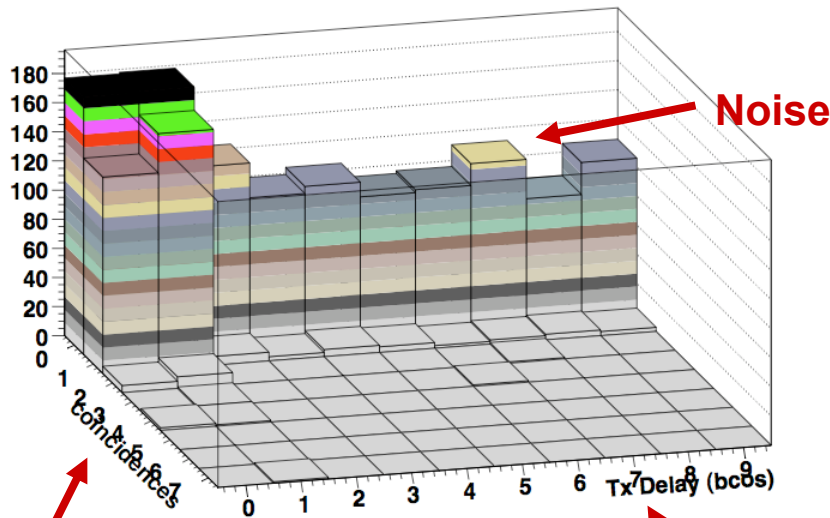
Number of coincident hits on top and bottom strips

Transmission Delay (bunch-crossings)

Transition Radiation Tracker (TRT)

Scintillator

20 cm of concrete

Scintillator

23

Conclusions

The Silicon Read-Out Driver (ROD)

- 9U VME board hybrid of DSPs and FPGAs
  - Primary functions: module configuration, trigger propagation and data formatting
  - Secondary functions: module calibration and monitoring
- ~260 production boards have been assembled (~160 for Pixels and ~100 for the SCT)
  - Many of these are now installed in their final crates inside of the ATLAS counting room

The ROD has already been used extensively

- During module production and detector assembly at various sites around the world
- Test-beam runs at CERN (most recently Summer 2004)
- Cosmics data taking during the combined TRT/SCT run in May 2006
  - Soon with the Pixel Endcap!

Looking forward to larger-scale multi-crate tests

- The SCT Barrels have been lowered into the ATLAS Collision Hall (testing in Winter 2006)

Focus of future work

- Speeding up the DSP code (shorten the time taken for configuration and calibration scans)
- Implementation of data-quality/detector monitoring during physics data-taking
- Automated response for module recovery/reconfiguration (in conjunction with DAQ/DCS)

Additional Slides

SCT and TRT Detectors in the Collision Hall
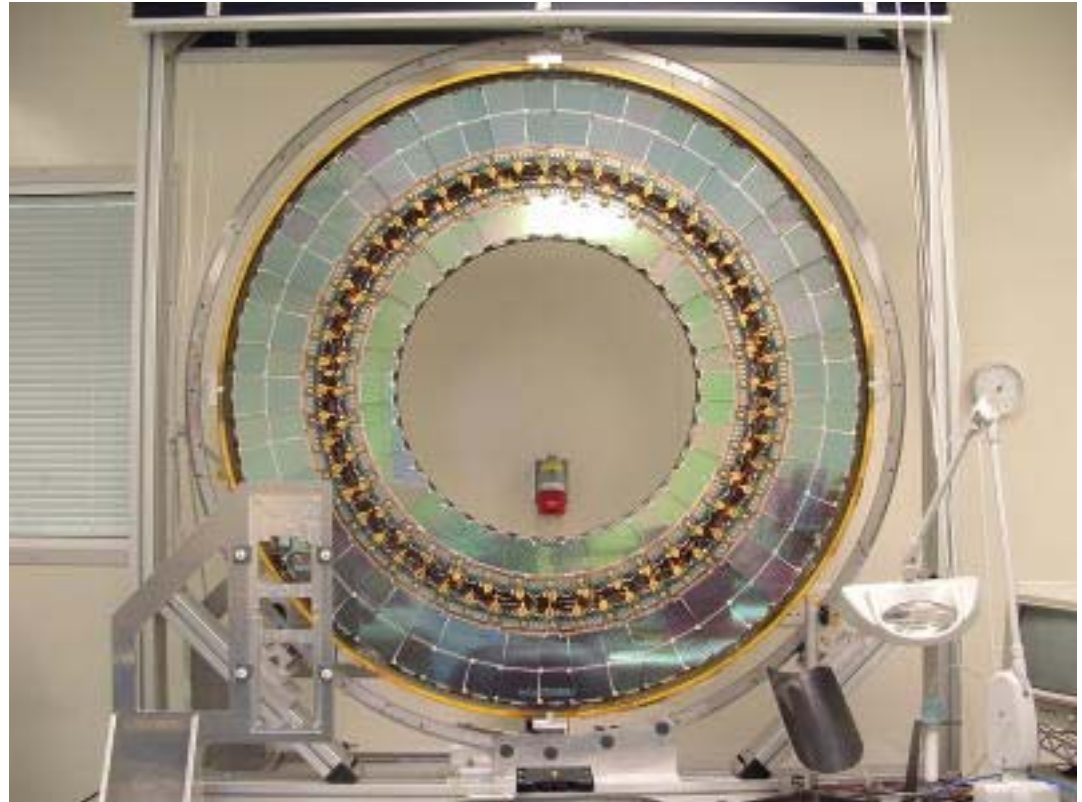
Barrels inserted into the cryostat in late August 2006
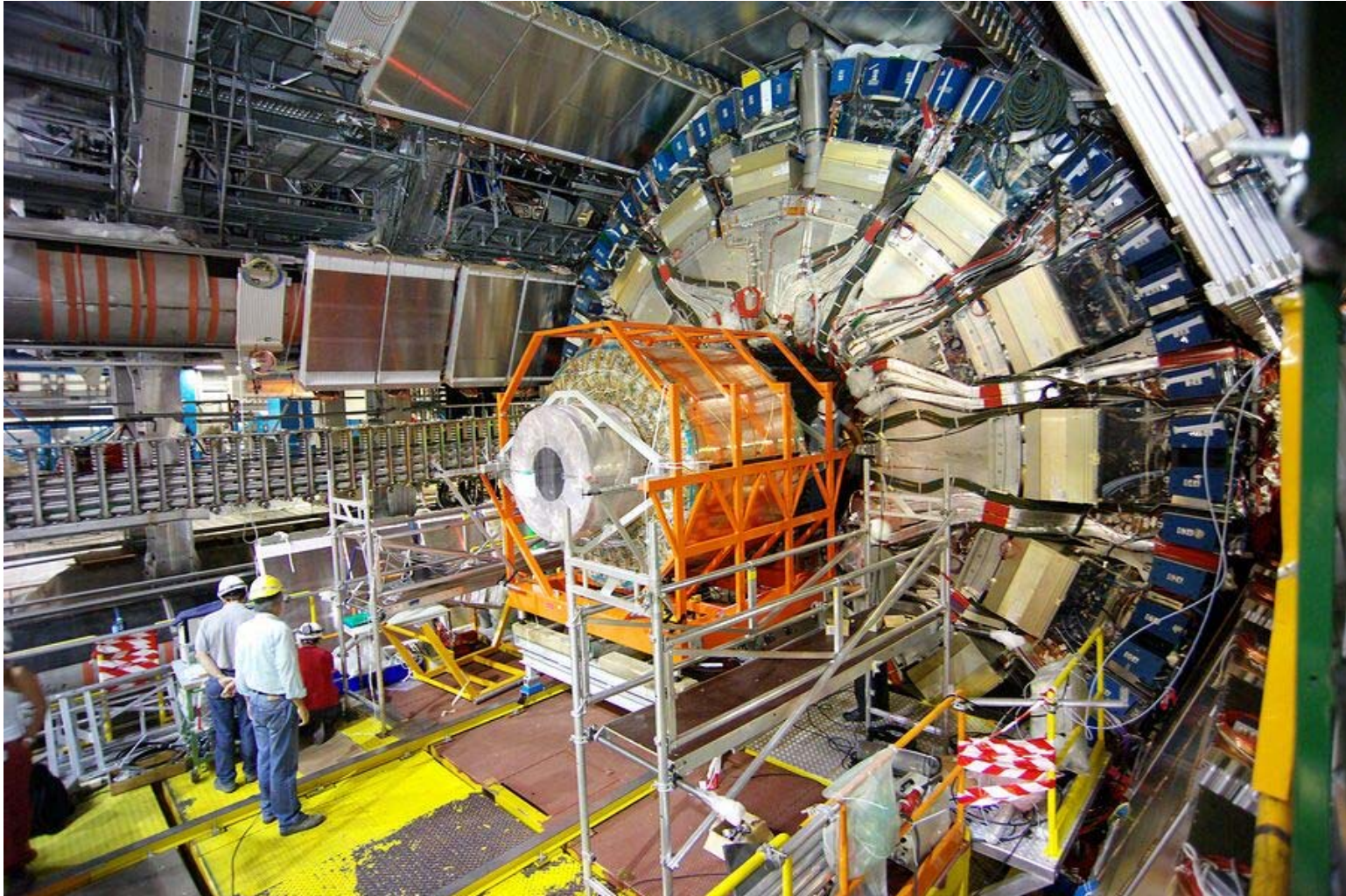
SCT RODs in USA-15

Production RODs in their final location for ATLAS running

The SCT Endcap

Module and Disk

The Pixel Endcap

Sector and Disks