

Trigger system design plan.

Define the data path from pixel detector to trigger output and the processes that must be implemented along that path. Define the data flow rates. Define how the processing reduces the amount of data at each step.

The trigger processing hardware divides into the FPGA and DSP processing.

For the hardware, split up the FPGA tasks and DSP tasks into modules that can be individually simulated and designed. This requires defining the data format in and out and the protocol between the modules. Numbers in { } brackets below indicates the priority attached to this element during the PAC review.

Pixel tasks.

Pixel Readout [Hardware – Pixel group, Data flow analysis – Gustavo]

(This is done to the raw data before it gets to the trigger system)

Take data from pixel chips.

Distribute events to fiber links to smooth flow rates and reduce links count.

Crossing number expansion to 16 bits? Add turn number to crossing number.

Detect and flush large events? Compress?

Pixel Trigger tasks.

Pixel Processor (FPGA) [Gustavo, Ken]

Event ordering.

Large data buffer with allocated space for event data and store by event. Timeout before event is passed on. Need to direct events to specific processing units but it can be a simple algorithm. Timeout is one turn or 159 132 ns crossings or 21 usec. No hit data can arrive later than this because the hardware cannot tag it correctly later than this.

L1 Buffer management.

(The trigger group will rely on or borrow liberally from the DAQ group who will design a 'standard' L1 buffer.)

Either the two data streams split right at the front of this buffer or the event ordering and L1 buffering can be done in the same memory. DAQ output design needs to be incorporated here so buffer fits DAQ model.

Cluster finding by row.

Serial processing, no buffering.

Include tags that may simplify later processing such as large events.

Translation table for row-col to x-y

Serial processing, no buffering.

Include tags that may simplify later processing such as seed region hits and hits near a centerline that should be replicated for different processing across a centerline.

Are tag fields added in pixel readout?

(Testing tool) Pixel hit generator FPGA for testing.

An FPGA that can generate a data stream probably from a file of simulation data and possibly a pattern generator.

FPGA Tracker - Hit Processor [Ted, Ken]

Seed region selector and event and tag checker.

Variable seed regions but changeable on initialization only. Should already be tagged from translation step. Data is replicated here before it is routed to the segment processing

Segment processing including CAM hit finder. {4}

Smallest unit is a two hit connector. This design needs considerable simulation and a prototype.

Tracker Switcher [??] {5}

Routing switching and scheduling tasks. The switch technology should come from the DAQ group. There may need to be a buffer here if crossings are queued. If the processing pool is larger than minimum then the data can go directly to a waiting element.

DSP - Track Processor and Vertex Processor.

Real Time “OS/monitor” including command and error messages. [Dave and Erik]

Need a list of Operating System features identifying those that are required, those that are luxuries and those that are not needed.

Application Software [Erik]

Algorithm simulated first then run in evaluation or prototype hardware.

Trigger studies.

How to test and ‘calibrate’.

Definition of response out and information passed to L2.

Hardware Platform [Greg] {2}

DSP family and specific processor selection process needs definition.

Within-a-family architecture selection by benchmarking simulations.

Can we buy commodity boards or is this a custom design?

Dual port buffers on the input and output can reduce processor load but may not be necessary.

Control and monitoring link. [Dave, Ken and Greg] {1}

Hardware definition of the link. USB, serial, host port are possible.

Define the protocol on the link and list messages necessary in order to help specifying the link.

Structure that allows support of 2000 nodes. Consider the possibility of a Master PC with USB port.

USB hubs for tree nodes. Microprocessor based USB nodes that have 8/16/32 outputs to individual DSPs.

Infrastructure [Greg and Vince]

Trigger system monitor CPU including protocol and messages.

Define protocol and list all the necessary messages.

USB ports to DSPs. Ethernet ports to system control/monitoring computer.

JTAG Ports to FPGAs

Need a list of errors generated by event checking in the hardware.

Packaging, cooling, Power supplies. [Vince in collaboration with all hardware designers]

Estimate of number of racks needed.

Define how packaging is driven by the C&M link and data links.

Muon Trigger tasks.

Algorithm Development [Mike]

Fortran implementation, using Doris Kim's (brute force) code as reference partitioning of compute, across the FPGA/DSP boundary

Muon Readout [Mike]

confirm that data are readout in radially inward order

confirm that the crossing are time-clumped (need not be time-ordered, but data can not be time-shuffled)

confirm that DCB's/fibers are octant-arm specific (and if not, what kind of inverse-DCB do we need?)

Wire Processor - FPGA [Mike]

can this function/task be implemented in the Pixel Processor/FPGA Tracker(s)? or will the Muon Trigger require its own front-line FPGA boards?

- Event ordering
- L1 Buffer Management
- adjacent wire compression
- list building ?
- initial matching ??
- more matching ???

(this task gains items as more is partitioned from the DSPs)

DSP Tracker [Mike]

Real Time OS/monitor

determine if DSP/BIOS and RTDX can do the following:

- must be able to download and (re)boot code
- must be able to set breakpoints and support remote debug
- must be able to provide run-time signaling (monitor) and control - should make

application development easier by providing "wheels" (rather than requiring reinvention of same)

Application [Mike]

C-implementation of entire (Fortran) algorithm

Optimization (DSP performance)

Reduction, offloading work to FPGAs

(this task loses items as more is partitioned to the FPGAs)

Hardware [Mike]

(mostly dictated by Pixel trigger)

L1 Buffers: what intermediate data should be recorded?

Control and monitoring link [Mike]

(mostly dictated by Pixel trigger)

JTAG? Of special significance if DSP/BIOS and RTDX are used protocol, unless "defined" by OS/monitor system structure that supports 2000 nodes (see remote test, below)

Infrastructure [Mike]

Trigger system monitor

integrated FPGA and DSP support

Test

single board, single crate, (sub)system

JTAG pods? And support software? For test, not configuration...

FPGA for-testing-only configuration(s)

DSP diagnostics (mostly for the system, but also for the chip)

("self benching", where possible)

remote test issues

how much of the above can be done from the control room?

Infiltration {Mike}

maintain high level of communications with Pixel Trigger group

recognize similarities with Pixel Trigger architecture

recognize differences with Pixel Trigger architecture, and respond

(seek to minimize hardware development, in exchange for soft/firm)

Numbers and notes

- Bunch crossing rate 132 ns. If bunch crossing number is 16 bits it can count 8.6 ms. If it is 24 bits it can count 2.214 sec.
- Data granularity is crossing number not events. There can be multiple events per crossing.
- The readout is capable of 5 hits per BC0, 132 ns.
- Pixel chips are 18 columns X 160 rows or 2880 pixels.
- Pixel chips may be 20 columns X 160 rows or 3200 pixels.
- Data rate estimated at 1 Gbit/sec/plane times 62 planes is 1 Tbit/sec, 1 Gbit/ms, 1Mbit/us.