L2 Performance and Plans

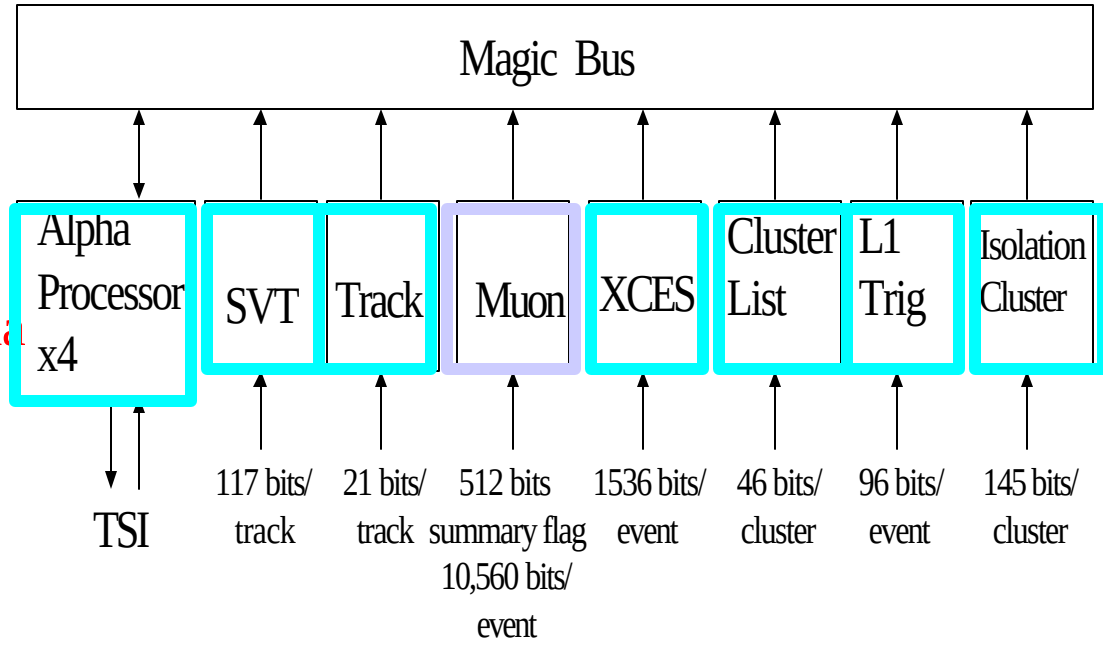
Stephen Miller

August 1, 2002

L2 Review

Description of Operation

- Effective 3 stage pipeline
  - Loading (4 buffers on boards)
  - MBus Transfer
  - Processing
- Data loaded from upstream into interface boards
- Data sent over Magic Bus to Alpha boards. (Data sent to all Alphas)
  - Reces boards readout by single Alpha if needed for electron trigger
- Alpha processor(s) evaluate algorithms (in parallel)
  - L2 Trigger decision sent to TS



Data Loading Performance

Data Loading rate/time

- Time from L1A to all data being on interface board
- L1: 500ns (4 CDF clocks)
- SVT: about 25us (real tracks)
  - Dominated by SVX readout and SVT processing
- Track: Minimum of 4.6usec + 132nsec*Ntracks>22
  - 4.6usec is time to check all XTRP data boards for tracks and send the EE word
  - 1.6usec to start sending first track + 132ns for each additional track
- Clist: 1 us per cluster
  - Dominate time is finding clusters
  - 300ns transfer time per cluster from LOCOS to Clist
- Isolist: 3us per cluster
  - Dominate time is finding clusters
- Muon: 2.5us
- Reces: 6us
  - Dominated by 5us to get data into SMXR

Magic Bus Performance

- Magic Bus Transfer rate/time
  - L1: 400 ns
  - SVT: 300 + 200*(N-1) ns 2 SVT tracks/MBus word
  - Track: 300 + 200*(N-1) ns 5 tracks/MBus word
  - Clist: 260 + 150*(N-1) ns 1 cluster/MBus word (could be 2 clusters)
  - Isolist:800 + 300*(N-1) ns 1 cluster/MBus word
    - 0ns for 0 MBus words
  - Muon: 300 + 200*(N-1) ns (assumed same as tracklist boards)
    - Muon board sends 4 MBus summary words
  - Reces: 0ns (Reces is read by Alpha while processing triggers)
    - Alpha uses 300ns on MBus to read Reces data
  - Magic Bus arbitration based on passing signal from slot to sot (20ns per slot)
    - 900ns of arbitration time for crate with 4 alphas and isolist sending every event
    - Current running has less than 500ns of arbitration time

Magic Bus Performance - cont

- Alpha Transfer rate/time
  - Magic Bus transfer time includes time to empty fifo on Alpha and put data in memory
  - Transfer over PCI Bus
    - Transfer starts as soon as data enters the fifo
    - Actual transfer rate limited by memory access time and other PCI activity
      - E.g. readout of TL2D bank, CPU reading data from previous event
    - Effective bandwidth of 20-80Mbytes/sec
  - Additional time from last MBus transfer to all data being in memory is < 1us

Processor Performance

- Algorithm Processing
  - unpack L1, scalers: 1.9 us
  - run algorithms (UoD): 12.8 us
  - error checking: 1.1 us
- L2-TS interface
  - start TSI handshake: 2.0 us
  - build TL2D: 3.3 us (1.1 us L2R, 71.3 us L2A)
  - finish TSI handshake: 2.0 us
- Board configuration overhead
  - Configuration time not parallel with either processing time or MBus transfer time
    - These times are dominated by PCI reads/writes to registers on fpgas
  - Alpha DMA config: 5.2 us
  - Check for L1A: 0.8 us

Current Performance Summary

- Data loading time is 25us
  - About 25us per event to load svt tracks into interface board
    - Total silicon readout time also determines when next L1A can start to be readout by silicon
  - Without silicon, all data loaded onto interface boards in 5-6us after L1A
- Magic Bus transfer time adds 0.5 to 1 us additional time
  - The time to send svt data to memory after interface board loading
  - Total MBus time is 5us,but boards send data as soon as available
- Processing time is 25us
  - About 15us for unpacking data, running algorithms, (includes long tail)
  - About 10us of overhead for configuring alpha, handshaking with TS
    - Configuration time of 6 us occurs before MBus transfer or processing starts

Deadtime

- Definition: Deadtime occurs when no L1A is possible because all 4 L2 Buffers are full.
  - L2 Buffers are used to hold all data after a L1A and before DAQ readout is finished.
  - Deadtime due to L2 time, DAQ readout time, backpressure from L3/CSL.
  - Even for L2/Readout time << L1A period, still get some deadtime due to Poisson fluctuations in the L1A rate
- L2 deadtime
  - L2 crate is a pipeline: 1 L2 buffer used for MBus, 1 L2 buffer for processing
    - Only 2 L2 buffers left for L1A (and loading), but 1 is usually being used for readout
  - TDR specification of 10us loading+transfer, 10 us processing
    - CDF3495 predicts 2.5% L2 deadtime at L1A rate of 45Khz
      - Single L2 stage with 20us gave 10% deadtime

Deadtime Limits

- Use Saltzberg simulation to estimate
- Comparison to recent run
  - run 148950 (silicon in, lum=2.1e31)
  - L1A=9000 Hz
  - L2A=200 Hz
  - fplatency=29000 ns
  - mbloading=3000
  - cputime=25000
  - tl2dmaking=71 usec
  - readoutlatency=1006 msec
  - pred dead: 2.4%; 0.6%, 1.4%, 0.4% (total; l2, vme, l2orvme)
  - obs dead: 4.9%; 1.3%, 2.6%, 0.7% (total; l2, vme, l2orvme)

Deadtime Limits (cont)

- With current configuration predict
  - L1A=10 kHz ==>1.5% L2 dead
  - L1A=20 kHz ==>9% L2 dead
  - L1A=30 kHz ==>22% L2 dead
- With improved times: Loading 20us, Processing 15us, L2Atime 15us
  - L1A=20 kHz ==>2.5% L2 dead
  - L1A=30 kHz ==>8.5% L2 dead
  - L1A=40 kHz ==>17% L2 dead 11%(6%) L2 deadtime with 10us(1us) Processing

Deadtime Summary

- Latency of slowest stage defines brick wall
- Sum of latencies gives time a buffer is filled
- Number of empty buffers sets distance between start of deadtime and brick wall
- With 2(3) stage pipeline, plus buffer used for readout, have very few empty buffers to absorb fluctuations
  - Shortening any stage will reduce the deadtime

L2 Improvements

- Implemented improvements
  - TTL MBus arbitration: Reduced MBus time from 34us to 25/3us (real/fake svt tracks)
  - Unpack on Demand + Compiler optimization: Reduced processing time from 44us to 14us !!
- Plan for improvements
  - Pipeline fpga (save 5us)
    - improved pipeline performance, better L2-TS handshake
  - DMA – code mod (3us) , new dma fpga (1.5us)
  - Multiple alphas – will help long processing tails
  - Only read Reces on some L2A (will save about 0.5% deadtime at high rates)
  - Software – drop L1 scalers (1.5us), precheck L1 bits (2us), algorithms
  - Isolist clusters only for high Et clusters – saves loading, MBus, unpack time
  - 2 Clist clusters per MBus word – saves MBus, unpacking time

Expected “Final” Run2a Performance

- Data loading
  - Will be dominated by SVX+SVT time. No estimate of final performance.
  - Using only Pass1 clusters for Isolist, will remove long tail for loading time
- MBus transfer – No significant improvements, but none necessary
  - Clist sending 2 clusters per MBus word will have negligible effect
  - Firmware could be optimized on all interface boards, but not worth the effort
- Processing time – Still room for improvement
  - Software changes will get average algorithm time below 10us
    - Multiple alphas will keep tail down as luminosity increases (but add 1us overhead)
  - L2-TS handshake currently at 4us. Need to understand time
    - New firmware will remove about half of that
  - Configuration overhead will be reduced to less than 2us with new firmware
    - Extreme firmware change puts it all in fpga, with time of 0.5us
  - Total processing+configuration time of 15us is realistic.
    - 10us total time is not impossible but would be hard to do.

Occupancy

- Estimate of occupancy based on data
  - BC + any other L1 trigger
  - CLC trigger – defines MinBias sample
  - HighPt – based on a few HighPt triggers
  - Brackets show RMS value
  - Pass1 clusters required for isolist

	BC+L1	CLC	HighPT
Ntrack	8.4 [7.4]	1.1 [2.3]	8
Nsvt	1.2 [1.3]	0.2	1
Ncluster	1.9 [1.7]	0.2	5
Npass1 Cluster	2.6%		10-20%

Occupancy at High Luminosity

- Extrapolate to 4x10^32 with 10 interactions per crossing
- Use HighPt occupancy + 10xCLC
- HighTail = average +3xRMS
- HighPt RMS = HighPt average

	10^{31} Mean	10^{31} High Tail	$4x10^{32}$ Mean	$4x10^{32}$ High Tail
Ntrack	8.4	31	19	76
Nsvt	1.2	5	3	12
Nclust	1.9	7	7	28

Data Loading at High Luminosity

- Use data loading times and expected occupancy
- Isolist time for High Luminosity only based on pass1 clusters
- L1 time fixed at 0.2us
- Muon time fixed at 2.5us
- Reces time fixed(?) at 6us

	10^{31} Mean	10^{31} High Tail	$4x10^{32}$ Mean	$4x10^{32}$ High Tail
Track	4.6us	5.8us	4.6us	11.7us
Svt	25us	??		
Clist	2us	7us	7us	28us
Isolist	6us	21us	4us	17us

MBus Transfer at High Luminosity

Use measured MBus times and exected occupancy

- Clist time for HighL based on 2 clusters per MBus word
- Isolist time for High Luminosity only based on pass1 clusters
- L1 time fixed at 0.4us
- Muon time fixed at 0.9us
- Arbitration time of 0.9us

	10^{31} Mean	10^{31} High Tail	$4x10^{32}$ Mean	$4x10^{32}$ High Tail
Track	0.5	1.5	0.9	3.3
Svt	0.3	0.7	0.5	1.5
Clist	0.4	1.2	0.7	2.2
Isolist	1.1	2.6	1.1	2.3
Total	4.5us	8.2us	5.4us	11.5us

Performance at 4x10^32

- Processing time extrapolated to High Luminosity
  - See Tom's talk for details
  - Processing time = 30us at high luminosity
  - Processing time difficult to estimate
    - Algorithm times can grow quadratically with number of objects in loop
    - Will be running a different set of algorithms
- Data loading time still dominated by SVX/SVT
- MBus transfer time not a limiting factor. Still below 10us
- Processing time (includes data unpacking) will increase the most at high luminosity/occupancy

Possible Processor Upgrade

- D0 is building Beta board as replacement for the Alpha
  - Commercial PCI processor board plugs into custom 9U board with VME, MBUS
    - See talk by Hirosky for details
- Possible to use a CDF version of Beta boards
  - Advantages
    - Designed to be hardware compatible with the Alpha boards
    - Allows simple migration to new processors – boards will work in existing crate
    - Uses development effort by D0 for hardware and software
    - Can still use existing trigger algorithms, data unpacking code
    - **Processor Upgrade based on commercial CPU board**
  - Requires minimal changes to make CDF version
    - Need connection to CDF clock and P2 CDF signals, LVDS signals for TS handshake
  - Propose making CDF specific version of the 9U board
    - Request purchasing 2 D0 boards for development and testing
      - 1 day with a soldering iron and hot glue gun to convert board to CDF format
      - Or run only as processor – keep alpha for controlling start_load, interfacing to TS

Run 2B - Reliability

- Interface boards in crate are working well
  - But 2 broken boards in last 7 months
    - 1 bad chip on L1 board
    - 1 stuck bit (solder fix) on Clist board
  - Currently have 2 spares per interface board
  - Interface boards are responsibility of the same group that provides the upstream data source
    - Muon, Cluster – Michigan
    - L1 – Yale/Argonne
    - Isolist, Reces - Argonne
    - Exception is UCLA tracklist boards
- Alpha Processors have worked well with no failures
  - Prototype board still works after 3 years in test stand
  - Only have 5 boards which is only 1 spare with 4 board configuration
  - Boards will be difficult to fix without expert

Conclusions

- Data collection time dominated by object formation
  - Time for SVT, CAL, ISOCAL >> Interface board to alpha transfer time
  - Magic Bus Transfer Latency dominates over bandwidth
    - Many data sources with few objects per source
- Processing time can be significantly reduced with processor upgrade
  - Upgrade necessary to meet Run2B physics goals
  - Using Beta boards is easiest processor upgrade option
- Long Term Reliability
  - L2 hardware has worked well with few problems
  - Alphas highly reliable, but long term repair capability will be difficult to maintain
  - Interface boards closely coupled to upstream data sources
    - Long Term maintenance needs to be ensured by relevant institutions