



# Plan for Alpha Boards

Stephen Miller

August 1, 2002

L2 Trigger Review



# L2 Processor Status

---

- **Hardware Status**

- **5 RevD Production boards**

- 3 in use at B0 – 1 in official L2 crate, 1 in 2<sup>nd</sup> L2 crate
- 2 in Michigan used for testing
  - 1 identified problem on each board

- **5 Rev C Production Boards**

- 1 Pre-production board at B0 – useful for teststand
- 4 production boards at Michigan with bad vias.
  - Not useful for any system

- **2 Prototype Boards**

- 1 functional board used in tests at Michigan



# L2 Processor Status

---

- Boards work very reliably
  - No hardware failures of components on Alpha board
  - Firmware and software works well
  - Firmware needs improving (next slide)
    - Low level of errors (few per shift) due to firmware



# Alpha Improvements

---

- **PIO fpga – used for single reads/writes on MBus**
  - Currently Alpha waits for interface boards to finish sending data before reading Reces
    - Problem with MBus timing of control signals can interfere with the L1 board
  - FNAL/D0 engineer working on rewrite of fpga design
    - If possible, will use the new design
  - Need to test capability for 64bit burst PCI transfers when reading Reces
- **TSI fpga – used for controlling data transfer, L2-TS handshake, getting L1A**
  - All control functions currently done by software read/writes to fpga registers
    - Includes getting L1A, sending startload, checking mod\_done bits, sending decision to the Trigger Supervisor. Each PCI transaction takes about 150 to 200ns
    - .Gives about 5us of overhead for each event.
  - This functionality has been put into fpga firmware
    - Has been tested without beam and fake tracks (L2\_torture run at 75KHz with 0.3% deadtime)
    - Tests with beam led to problems with DMA configuration fpga



# Alpha Improvements (cont)

---

- **DMA fpga -**
  - Need to configure DMA buffers every event
    - Takes 2us to write to 6 buffers (plus 3us for reading buffers)
    - New version of fpga will configure buffers with a single write
    - Design needs to be debugged
  - Occasional errors during running when PCI reads/writes to DMA fpga give wrong data
    - Seems to be dependent on other PCI bus activity
- **VME reads**
  - VME readout of TL2D takes about 1ms
    - Due to single word transfers plus large data size of TL2D
      - Most of TL2D is scaler information for L1 and L2 trigger bits
  - Can enable Universe chip to do 64bit PCI burst transfers and VME block reads
    - Works fine in standalone test. Corrupts reading of DMA buffers when used in running



# Multiple Alphas

---

- Alphas tested together in crate without beam
  - Cosmic runs with 3 alphas have been run at 20KHz
    - 2 alphas always sent L2 rejects to TS
  - For test 1 Alpha reads data loaded in other alphas to check for errors
    - Boards all received same data
- Multiple Alpha Plan
  - First need to get improved firmware on alpha
  - Repeat no beam tests with pipeline fpga for data loading
  - Likely will wait for shutdown for multiple alpha tests
- Creation of L2 executables still a big issue for multiple alphas
  - Requires code to optimally share triggers between the different exes
  - Lots of work to do on code which makes executable from the database
- Multiple alphas should help tail of processing time
  - Events that require processing multiple complicated triggers
    - E.g. events with multiple L1 triggers being passed



# Maintaining Boards

---

- **Documentation**
  - Extensive documentation for the AlphaPC164 motherboard and Alpha CPU
  - Document written specifying functionality of FPGAS
  - Schematics exist at B0 and as postscript.
- **Debug Software**
  - Some test utilities part of the EBSDK code
  - Alpha based program for VME and MagicBus tests with 1 or more Alphas
  - Standalone C programs on Linux boxes to read/write to alpha registers/memory
- **Repair**
  - Short term repair of Alphas hasn't been necessary
  - Would plan on doing repairs at Michigan
  - Repair of Alphas in long term might be difficult



# Summary

---

- Alpha boards function very well
  - Some low level errors due to fpga operation need to be fixed
  - Most failures of alpha have been software related
- Several improvements needed for optimal operation
  - Improved fpga designs for PIO, DMA, TSI will significantly reduce processing time for each event
  - Improved DMA fpga should also allow faster VME readout
- Plan to move to Multiple Alpha operation during shutdown
  - Biggest effort will be in software and executable creation