DFE Crate Controller

The DFE Crate Controller upgrade will be designed, assembled and tested at D0.

1. Finalize the design specification.
2. Choose parts, tech begins building schematic symbols and footprints
3. EE to draw and check the schematic (1 month)
4. Begin firmware design and simulate (2 months)
5. Review schematic and approve for layout. Order parts.
6. EE to work with tech on PCB layout (1 month)
7. Review layout and approve for fabrication.
8. PCB fabrication and inspection. (3 weeks)
9. Send boards out for BGA placement and inspection. (2 weeks)
10. Techs to stuff remaining parts the board (2 weeks)
11. Power up testing, program the FPGA, blink LEDs.
12. Debug the serial command link interface.
13. Debug the GbE interface, begin sending/receiving basic packets from standalone Linux PC. EE, with help from G.Savage, R.Rechenmacher.

BACKPLANE and POWER

The new backplane specificiation will be given to a vendor who will draw the schematic, layout the backplane and assemble it. Finished backplanes will be delivered to D0 where a technician will mount them in a new crate. All crates will be assembled here and then one unit will be shipped to BU. The plan is to have one crate at BU, one at D0, two for installation, and two for spares.

1. Agree that 48V power distribution is going to work.
2. Finalize the pinouts and mechanical drawings, send the design specification out to vendors.
3. Order crate chassis/crate parts
4. Select the 48V bulk supply and order.
5. The backplane vendor draws the schematic for the backplane (1 month)
6. FNAL review of this schematic
7. Vendor layout of the backplane (1 month)
8. FNAL review of layout
9. Vendor assembles and tests the backplanes (1 month)
10. FNAL receives the backplanes from the vendor assembles all crate. (FNAL tech)
11. Send a crate and power supply to BU.
12. Work with the D0 infrastructure group to determine how to connect the bulk supplies to the various safety systems in the detector (i.e. RMI interlocks, VESDA, local smoke and water detection systems, remote power control, etc.)

DFEA

BU should deliver DFEA prototypes to FNAL before INTEGRATION TESTING (I) begins. At this time the DFEA boards should be power-up tested and all interconnections on the boards should be tested and verified to be error free (using JTAG testing or equivalent). DFEA production firmware need not be ready for this testing.

INTEGRATION TESTING (I)

The purpose of this testing phase is to check that the pieces fit together and communicate properly at a basic hardware level. We'll test to make sure that the Linux PC can talk to the DFEA board via the DFEC and the backplane. Even if the DFEA production firmware is ready, we won't be testing any physics functionality now. These tasks will probably be duplicated at BU and D0 and run in parallel.

1. Plug DFEC and DFEA boards into backplane, check mechanical alignment and power up.
2. Connect the DFEC to an SCL link and look at clock and control signals being distributed over the backplane. Make sure that skew between multiple DFEA boards is within spec. At this point the DFEA should have some basic firmware that can decode the control bits and check for parity errors, sync gap length, etc. The DFEA should turn some diagnostic LEDs on.
3. Unplug the DFEC from the SCL link and set it to “simulated SCL” mode, where it will run using the 53MHz oscillator. In this mode the DFEC will generate a fake beam structure and assert L1_ACCEPT signals at periodic or random intervals.
4. Connect the DFEC to the Linux box via Ethernet. Write and Read back DFEA registers from the Linux box software. Insure that read/write operations to/from DFEA registers are rock solid.
5. Initialize the DFEA FPGAs from the Linux command line. At this point there will be no EPICs or DFEWare or anything like that. Just a set of command line utilities and/or C functions are all that's needed to test this connection.
6. Test DFEA input/output LVDS links. The DFEA should drive a pattern out on the output links and loop it back to the inputs. The DFEA firmware should check for bit errors on these inputs and blink LEDs.
7. Mount an SLDB transmitter on the DFEA board and send test patterns to the L1muon hardware (done at FNAL). L1muon will run bit-error rate tests to insure the integrity of these links (4-6 per DFEA board).

INTEGRATION TESTING (II)

This testing will still occur on the bench. Now we'll be testing more of the functionality in the DFEA production firmware. Most of this testing will occur at BU, and the DFEC can be set to "free run" mode so an SCL transmitter is not needed. The setup will be: crate, backplane, DFEC, DFEA(s), Linux box with an Ethernet card. Since test vectors are contained in the buffers inside the DFEA FPGAs, a datapump is not required.

1. Setup the DFEA to produce various test patterns (fake events) on the outputs. Capture these outputs in the DFEA buffers and read them from the Linux box. Insure that these patterns are correct and that the DFEA's output capture logic is working properly.
2. Load the DFEA input buffers with test vectors (or real data captured from run IIA hardware)and inject them into the DFEA firmware core at full speed. Capture, readout and verify the DFEAs output records. This could be automated with custom software running on the Linux box. Many events should be tested to insure that the firmware is working properly.
3. Test that the isolation bits, which are shared between neighboring DFEA boards, are working properly and there are no timing issues here.
4. Verify the overall latency of the DFEA and make sure it's within spec.

INTEGRATION TESTING (III)

This testing will occur mostly at D0, and it's mostly software related. The goal of this testing is to make the Linux software interface more robust, so that multiple clients (which may be running on different machines) can talk to the new crates. The low-level functions which provide read/write access to the DFEA boards must be integrated into a framework such as EPICs, which will ultimately handle monitoring, status, alarms, firmware databases and revision control, and firmware downloading. This testing can occur in parallel with INTEGRATION TESTING I and II.

1. Build up the network card driver to support multiple clients. "channelization"? Begin integrating into EPICs framework. Ohio State University should be consulted, as they have done something similar to this already.
2. Determine if the firmware revision control/archiving/downloading code can fit into the existing "DFE Ware" program. There is no longer flash memory in the DFECs – now the firmware files go directly from the Linux host to the DFEA boards. Does this break DFEWare's download and initialize routines? Do we abandon DFEWare, or do we modify it?
3. Determine which DFEA registers need to be routinely scanned by EPICs. Setup EPICs to scan for this status information and set alarms, etc. Write new GUIs to display the status of the DFEA boards.

INTEGRATION TESTING (IV)

In this phase the DFE hardware is moved down to the center platform and connected to the MIXER crate in parallel with the existing DFEA hardware. A passive splitter will be used to "tap off" from the existing DFEA cables, so that the new DFEA crate can get real inputs without disrupting the CTT system.

1. Install the LVDS splitter on a few of the MIXER to DFEA cables.
2. Temporarily install the new DFE crate on the center platform, North Side rear aisleway, behind the existing DFEA crates.
3. Install new cables from splitter to the new DFEA boards. Install enough cables for two sectors (one DFEA board).
4. Install 48V supplies on the West/East platform, tie into the safety system and test. Insure that the new DFEA crates have adequate protection and cooling. Connect power to the new DFE crate and test.
5. Install a spare CTOC board in crate PW03-2 and connect a few of the new DFEA boards to it. Connect the L3 output of this new CTOC board to our L3 readout crate.
6. Connect SCL and Ethernet cables to the new DFEA crate, power up and verify that communications are working properly.
7. Read/write DFEA registers, initialize boards with production firmware.
8. Verify that EPICs is scanning the new DFEA boards and displaying the status bits properly.
9. Set the new DFEA boards to send fake L1/L2 records to the CTOC board. This CTOC board will test the integrity of these records and send them to our L3 readout crate for inspection.
10. Move cables and plug a few new DFEA sectors into an existing CTOC. Now there will be one CTOC board which has inputs from new and old DFEA boards. This CTOC board can compare the relative arrival time of these records and check that everything is within spec. *Once we are confident that things look good here, then we can proceed to the next step.*
11. Disconnect all run IIA DFEA hardware, including power supplies. Install the new DFEA crates in PC03-2 and PC03-1 and re cable. Install all of the new DFEA boards.
12. Power up and initialize the DFEA crates with production firmware. Set all outputs to test pattern mode and use downstream hardware (CTOC, STOV/STSX, L1muon) to verify data integrity and to check for cable installation errors.
13. Check DFEA status bits and insure that all MIXER to DFEA cables are installed properly.
14. Load the DFEA input buffers with test vectors, and verify that all downstream boards are receiving the expected records.
15. Switch the DFEAs over to data mode and check that the CTTT trigger rates are reasonable, begin offline analysis using trigsim/examine,etc.

“LONG-TERM” MAINTENANCE

D0 post-docs will act as a liaison between the physics groups and the BU firmware engineer. BU will provide firmware updates in response to requested changes. Post-docs will be responsible for checking in this firmware in and initializing the DFEA boards. Firmware updates are an iterative process – we must assume that this process will occur for the entire duration of run IIB.