DHCAL Data Collector  
Loop-Back Test  
May 7, 2007

This note describes some test firmware developed to permit self-contained testing of the front-end links on a DCOL module. No external hardware is required, except for a custom RJ-45 cable and VME crate.

The loop-back firmware uses the TTM connector at the top of the front panel. It can be connected to any of the 12 front-end links using a special cable (described below). It also includes the normal DCOL functionality (except that only standalone mode is available since the TTM connector is used for the loop-back test).

The loop-back firmware is available in the firmware directory:

http://physics.bu.edu/~wusx/download/DCOL/

As of this writing the current file was called dcol_loop.mcs and dated 5/2/07. It is recommended that the loop-back firmware be stored in the backup sector of the board. This can be accomplished using the 'dcol.cc' utility program with the command:

    p dcol_loop.mcs 1

**Setup for Loop-Back Test**

First, obtain a loop-back test cable. The cable should have standard RJ-45 connectors on both ends, with wiring as follows:

- Wire pins (1,2) and (7,8) straight through using two pairs.
- Leave pins 3, 4, 5, 6 unconnected.

1. If you have stored the loop-back firmware in flash sector 1 as suggested above:
   - write 0xabcdef01 to offset 0x18 (“FPGA_prg”)
   - read from offset 0 (“CMD”). Bit 31 should be set indicating sector 1 loaded
   - read from offset 4 (“CSR”). Firmware revision (upper 8 bits) should be zero.

2. Reset and configure the board:
   - write 1 to offset 0 (“CMD”)
   - write 0xf to offset 4 (“CSR”)

   Enable run mode, circular mode, standalone mode and disable sorting

3. Enable one front-end link
   - write to offset 8 (“DCON_enable”) to enable one input
   - Bit 0 = top input
   - Bit 1 = 2nd input, etc
**Slow Control Transmit Test**

Slow control data can be sent to the loop-back module using the normal DCOL interface. For example, to send 10 bytes to front-link 1 (the top-most one):

- Write 0x80112233 to offset 0x2000
  - Write 0x44556677 to offset 0x2004
  - Write 0x88990000 to offset 0x2008
- Write 3 to offset 0x28
- Read offset 0x28, check that it is zero to indicate transmission complete

A couple of notes:

- The first byte transmitted is always in the upper 8 bits of the first word. In this case, it is 0x80.
- The high bit of the first byte must always be set to '1' (this is the start bit)

You should then see the same data received by the loop-back module at offset 0x800, 0x804 and 0x808:

  - Read 0x800, expect 0x80112233
  - Read 0x804, expect 0x44556677
  - Read 0x808, expect 0x88990000

**Slow Control Receive Test**

The loop-back module can send fixed-format, 4 byte packets back to the DCOL on the slow control link. This simulates the return of data from the front-end if a slow control read is performed. To trigger this feature:

Write 0x80000000 to offset 0x28

You can view the received data in the slow control receive buffer at 0x2800 (plus an offset for the input number). The format of the 4 bytes is as follows, as packed into a 32-bit word:

0xhhnnmmcc

Where the first byte is as follows:

- bit 7 always '1'
- bits 6-3 connector number (0-11)
- bits 2-0 always '0'

The 2nd and 3rd bytes (nn, mm) are the contents of a counter which increments once for each byte sent. The 4th byte (cc) is a checksum of the other 3 bytes.

**Hit Data Test (single packet)**

The loop-back module can send fixed-format, 16-byte packets to the DCOL as hit data. This simulates the transmission of hit data from the front-end. To trigger this feature:

Write 0x40000000 to offset 0x28

The first byte is 0x80, with the input connector number (0-11) stored in bits 3-6. The 2nd through 15th bytes are the contents of a counter which increments once for each byte sent. The 16th byte is a
checksum of the other 15 bytes.
The received data may be seen in the circular event buffer at 0x800000.

**Hit Data Test (continuous)**
The loop-back module can send a continuous stream of 16-byte hit data packets. To trigger this feature:

Write 0x60000000 to offset 0x28

The packets have the same format as described above. Note that this test can fill the buffer if allowed to run for more than about 20 seconds. To stop the transmission:

Write 0 to offset 0x28