TMB 2005 + 2013 Design

TAMU & UCLA High Energy Physics
Version 5.02
February 7, 2014

Table of Contents

FIGURES 3
TABLES 3

TMB OVERVIEW 4

CLCT PROCESSING ALGORITHM 5

DDR SIGNAL SYNCHRONIZATION 7

CFEB DDR Receiver Sync Stages 7
ALCT DDR Receiver Sync Stages 8
ALCT DDR Transmitter Sync Stages 9

TTC SEQUENCES 13

Trigger State 13
Bunch Crossing Counter 13
Resync 13

ALCT+CLCT MATCHING ALGORITHM 14

Matching Logic 14
LCT Duplication 14
LCT Quality 14
MPC Format 15

VME REGISTERS 17

Addressing Modes 17
Base Address 17
Boot Register 17
Register Addresses 18
Register Definitions 24
TTC Commands: 84

TMB BOARD STATUS OPERATIONS 85

ID Registers 85
Digital Serial Numbers 85
Power Supply ADC 85
Clock Delays 85
JTAG Chains 86
RAT Module Status Register USER1 86
RAT Module Control Register USER2 87

USER PROM PROGRAMMING 88

Register Initialization 88
VME PROM-0 89
JTAG PROM-1 90

DMB READOUT 93

Full-Readout and Local-Readout Format (Long Header): 93
Long Header-only Format: 94
Short Header-only Format: 94
Miniscope: 95
Blocked CFEB DiStrips List Format: 97
Header Word Descriptions: 98

Sample TMB Raw Hits Dump 112

CONFIGURATION 113

Shunt Settings 113
Switch Settings 114
Fuses 115

SIGNAL SUMMARY 116

CCB 116
ALCT 117
DMB 118
CFEB 119
MPC 120
RPC 121
VME 121
JTAG 121
LEDs 122
TMB Total I/O Count 123

CONNECTORS 124

TMB Connector Summary 124
J0-J4 CFEB0-CFEB4 Connectors 125
J5 ALCT Cable1 Connector (Receiver) 126
J6 ALCT Cable2 Connector (Transmitter) 127
J1-J6 SCSI-II 50-Pin Connector Pin Convention 128
J7 Xilinx LVDS Xilinx X-Blaster Connector 129
P1 Backplane VME64x J1/P1 Connector 130
P2A Backplane CCB+DMB Connector 131
P2A Backplane CCB+DMB Connector Continued 132
P2B Backplane DMB Connector 133
P3A Backplane MPC Connector 134
P3B Backplane RPC+ALCT Connector 135
Backplane Pin Diagram 136
Front Panel Connector Locations 137
CCB Front Panel 138

DOCUMENTATION REVISION HISTORY 140

FIRMWARE CHANGE LOG 143

Figures

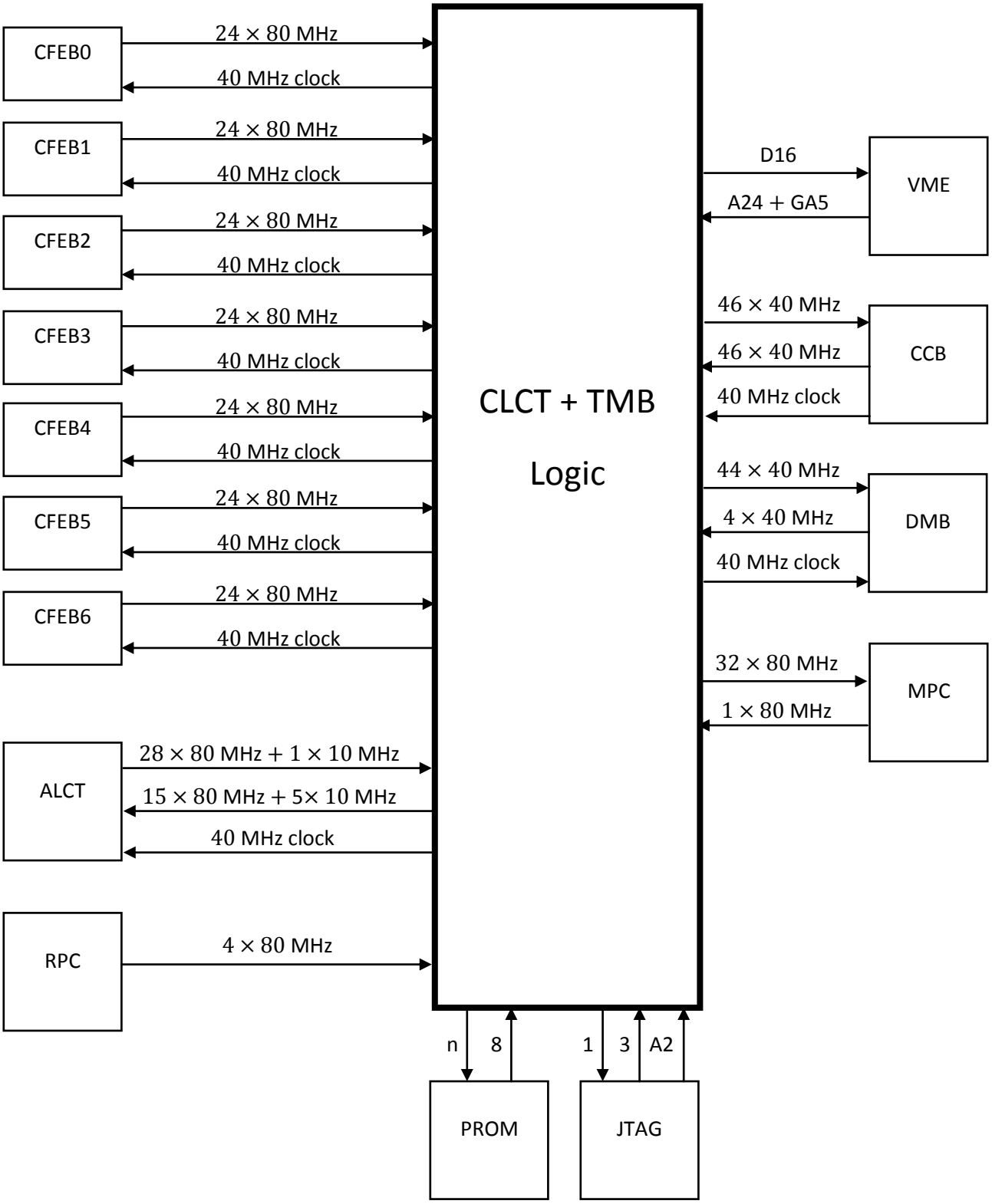
Figure 1 TMB Overview 4
Figure 2 CFEB Muonic Receiver 10
Figure 3 ALCT Muonic Receiver 11
Figure 4 ALCT Muonic Transmitter 12
Figure 5: 50-Pin PCB Connector (Female) 128
Figure 6: 50 Pin Cable Connector (Male) 128
Figure 7: 50 Pin PCB Connector Pin Convention 128

Tables

Table 1: Shunts 113
Table 2: Switch Settings 114
Table 3 & 3A: Fuses for TMB2005 and TMB2013 115
Table 4: CCB Signal Summary 116
Table 5: ALCT Signal Summary 117
Table 6: DMB Signal Summary for TMB2005 and TMB2013 118
Table 7: CFEB Signal Summary 119
Table 8: MPC Signal Summary 120
Table 9: RPC Signal Summary 121
Table 10: VME Signal Summary 121
Table 11: JTAG Signal Summary 121
Table 12, 12A & 12B: TMB Front Panel LEDs, Mez-2013 SMT LEDs and Mez-2013 Testpoints 122
Table 13: TMB Total I/O Count 123
Table 14: Connector Summary for TMB2005 and TMB2013 124
Table 15: J0-J4 CFEB0/4-to-TMB Connectors 125
Table 16: J5 ALCT Cable1 Connector [J10 on ALCT board] 126
Table 17: J6 ALCT Cable2 Connector [J11 on ALCT board] 127
Table 18: J8 Xilinx LVDS X-Blaster Connector 129
Table 19: P1 VME64x Connector 130
Table 20: P2A Backplane CCB+DMB Connector 131
Table 21: P2B Backplane DMB Connector 133
Table 22: P3A Backplane MPC Connector 134
Table 23: P3B Backplane RPC+ALCT Connector 135
Table 24: J5 ALCT Cable1 Connector [J10 on ALCT board] 157

TMB Overview

Figure 1 TMB Overview



CLCT Processing Algorithm

bx -½ Latch 1st-in-time CFEB transmission

bx 0 Latch 2nd in-time CFEB transmission
Demultiplex 1st-in-time and 2nd-in-time CFEB Triads
Map cable-signal ordering into 5 CFEBs x 8 DiStrip Triads x 6 Layers (240 signals)
Store CFEB Triads in Raw-hits RAM
Multiplex CFEB Triads with internal pattern-injector RAM
AND Triads with Hot Channel Mask to disable errant DiStrips
Decode Triad start bits (240 processed in parallel)

- Triad decoder state machines run continuously to preclude missing any start bits
- Each ½-strip one-shot can fire again on the same clock cycle that the previous ½-strip pulse ends
- If a 2nd triad arrives for the same DiStrip while busy, the triad is decoded but the one-shot does not fire
- In that case, the triad-skipped counter is incremented

bx 1 Decode Triad strip bits

bx 2 Decode Triad ½-strip bits

bx 3 Fire ½-strip one-shots for 6bx (triad_persist is programmable, 6bx is the default)
OR 160 ½-strips on each layer for layer-trigger mode
Stagger correction (if stagger_hs_csc =1) shifts alternate layers by -1hs

ly0	-0hs	i.e. hs5→ hs5	
ly1	-1hs	i.e. hs6→ hs5	
ly2	-0hs	i.e. hs5 → hs5	key layer ly2
ly3	-1hs	i.e. hs6 → hs5	
ly4	-0hs	i.e. hs5 → hs5	
ly5	-1hs	i.e. hs6 → hs5	

Pattern Finding:

For each of 160 key ½-strips consider the 42 neighboring ½-strips (i.e. on key 5 use the following ½-strips)

```
hs              0123456789A
ly0[10:0]       xxxxxkxxxxx     5+1+5 =11
ly1[ 7:3]          xxkxx        2+1+2 = 5
ly2[ 5:5]            k          0+1+0 = 1
ly3[ 7:3]          xxkxx        2+1+2 = 5
ly4[ 9:1]        xxxxkxxxx      4+1+4 = 9
ly5[10:0]       xxxxxkxxxxx     5+1+5 =11
```

For each of 160 key ½-strips, count layers with hits matching the 9 pattern templates
Pattern ID=1 is a layer-OR trigger, Pattern ID=0 is no-pattern-found

```
Hit pattern LUTs for 1 layer: - = don't care, xx= one hit or the other or both
Pattern       id=2        id=3        id=4        id=5        id=6        id=7        id=8        id=9        idA
Bend dir      bd=0        bd=1        bd=0        bd=1        bd=0        bd=1        bd=0        bd=1        bd=0
              |           |           |           |           |           |           |           |           |
ly0      --------xxx xxx-------- -------xxx- -xxx------- ------xxx-- --xxx------ -----xxx--- ---xxx----- ----xxx----
ly1      ------xx--- ---xx------ ------xx--- ---xx------ -----xx---- ----xx----- -----xx---- ----xx----- -----x-----
ly2 key  -----x----- -----x----- -----x----- -----x----- -----x----- -----x----- -----x----- -----x----- -----x-----
ly3      ---xxx----- -----xxx--- ---xx------ ------xx--- ----xx----- -----xx---- ----xx----- -----xx---- -----x-----
ly4      -xxx------- -------xxx- -xxx------- -------xxx- ---xx------ ------xx--- ---xxx----- -----xxx--- ----xxx----
ly5      xxx-------- --------xxx -xxx------- -------xxx- --xxx------ ------xxx-- ---xxx----- -----xxx--- ----xxx----
              |           |           |           |           |           |           |           |           |
// Extent   0123456789A 0123456789A 0123456789A 0123456789A 0123456789A 0123456789A 0123456789A 0123456789A 0123456789A
// Avg.bend - 8.0 hs    + 8.0 hs    -6.0 hs     +6.0 hs     -4.0 hs     +4.0 hs     -2.0 hs     +2.0 hs      0.0 hs
// Min.bend -10.0 hs    + 6.0 hs    -8.0 hs     +4.0 hs     -6.0 hs     +2.0 hs     -4.0 hs      0.0 hs     -1.0 hs
// Max.bend - 6.0 hs    +10.0 hs    -4.0 hs     +8.0 hs     -2.0 hs     +6.0 hs      0.0 hs     +4.0 hs     +1.0 hs
```

bx 4 Result for each of 160 keys is a list of 9 pattern-ID numbers (pid) [2 to A] and corresponding number of layers [0 to 6] with matching hits (nhits)

Find the best 1-of-9 pattern ID numbers for each key by comparing nhits

- Ignore bend direction: left and right bends have equal priority (bit 0 of pid implies bend direction)
- If two pattern IDs have the same nhits, take the higher pattern ID
- A key with no matching hits, would always return pid=A and nhits=0

bx 5 Pre-trigger if any 1-of-160 keys have nhits ≥ hit_thresh_pretrig and pid ≥ pid_thresh_pretrig
Construct 5-bit active-cfeb list for DMB:
cfebs with a key that has nhits ≥ hit_thresh_pretrig and pid ≥ pid_thresh_pretrig
cfebs with a key that has nhits ≥ dmb_thresh_pretrig
cfebs adjacent to a cfeb that has nhits ≥ hit_thresh_pretrig and pid ≥ pid_thresh_pretrrig within adjfeb_dist

bx 6 Finding 1st CLCT:
Construct 7-bit pattern quality for sorting: pat[7:0]
pat[7:5]=nhits[2:0]
pat[4:0]=pid[3:0]
Ignore the bend direction bit (pid[0]), left and right bends have equal priority
Store pat[7:0] for 160 keys for use later to find 2nd CLCT
Start finding best 1-of-160 keys by sorting on the 6-bit number pat[7:1]
If two keys have the same pat[7:1] take the lower key

bx 7 Find 1st CLCT:
Finish finding best 1-of-160 keys by sorting on the 6-bit number pat[7:1]
Store 1st CLCT info: key, pattern ID, and number of hits
For empty events, key=0, pid=A and nhits=0. If clct_blanking=1, then key=pid=hits=0

bx 8 Finding 2nd CLCT:
Construct list of busy keys
Mark keys near 1st CLCT as busy from 1st key-nspan to 1st key+pspan
If clct_sep_src=1, pspan and nspan are set equal to clct_sep_vme, typically 10hs
If clct_sep_src=0, pspan and nspan are read from RAM and depend on the pattern ID number
This allows two non-bending tracks | | to be closer than bending tracks / \
Start finding best 1-of-160 keys by sorting on the 6-bit number pat[7:1]
Skip busy keys
If two keys have the same pat[7:1] take the lower key

bx 9 Find 2nd CLCT:
Finish finding best 1-of-160 keys by sorting on the 6-bit number pat[7:1]
Store 2nd CLCT info: key, pattern ID, and number of hits
For empty events, key=11, pid=A and nhits=0. If clct_blanking=1, then key=pid=hits=0

bx 10 Drift Delay 1bx (waits for CSC drifting)

bx 11 Drift Delay 1bx
If clct0 nhits < hit_thresh_postdrift OR pid < pid_thresh_post_drift, discard event

bx 12 Match to ALCT window 0
If alct matches, jump to bx15 logic, latency is shortened 2bx

bx 13 Match to ALCT window 1
If alct matches, jump to bx15 logic, latency is shortened 1bx

bx 14 Match to ALCT window 2
If ALCT does not arrive, and clct_only mode is enabled, accept CLCT at window 2
If ALCT does not arrive, and not in clct_only mode, discard event

bx 15 Construct two LCTs from CLCT and ALCT data
If event has 2 CLCTs and 1 ALCT, copy 1st ALCT into 2nd ALCT position
If event has 1 CLCT and 2 ALCTs, copy 1st CLCT into 2nd CLCT position
Calculate LCT quality
Multiplex mpc injector ram data

bx 16 Transmit 1st-in-time LCT frame to MPC

bx 16½ Transmit 2nd-in-time LCT frame to MPC

DDR Signal Synchronization

CFEB DDR Receiver Sync Stages

CFEB receiver logic is designed to synchronize incoming comparator data to TMBs main clock, while minimizing CLCT trigger-path latency. It has a programmable 0-16bx delay stage to compensate for differing CFEB cable lengths.

The logic is shown schematically (Figure 2) but the actual TMB logic is written in behavioral Verilog.

U1A Latches incoming CFEB data din on the *falling* edge of clock_iob.
The user has already tuned the associated cfeb[n]_rxd digital phase shifter so that clock_iob always latches stable data.

U1B Latches incoming din on the *rising* edge of clock_iob.
U1A and U1B comprise a single Double Data Rate (DDR) I/O-Block flip-flop, and are only shown here separately for clarity.

U2 Latches data transferred from U1A on the *rising* edge of clock_iob.
U2 now holds din_1st-in-time data, aligned with the rising edge of clock_iob, while U1B holds din_2nd-in-time data, also aligned with the rising edge of clock_iob.

In non-muonic firmware, this would be the end stage, because clock_iob would be the same as TMBs main clock.

In muonic firmware versions, din_1st and din_2nd still need to be synchronized to TMBs main clock. This is done by latching data on both the posedge and negedge of the main clock, then selecting one latch or the other with the posneg bit.

If the posedge flip-flops U11,U12 are out-of-time (creating a dead-spot in the din receive window) then the negedge flip-flops U3,U4 must be in-time. One set of latches, either U11,U12 or U3,U4 will always be in-time for a given phase of clock_iob.

U3,U4 Latch din_1st and din_2nd on the falling edge TMBs main clock_1x.

U5,U6 Select either posedge data from U2,U1B or negedge data from U3,U4.

U7,U8 Delay data by 1-to-16 bx, according to delay[3:0] from a VME register.
Setting address=0 for U7,U8 gives a 1bx delay, so delay-1 is used to form the address.

U14 Subtracts 1 from delay[3:0] to form the shift register address for U7,U8.

U9,U10 Bypass delays U7,U8 if the delay is 0, thus giving a 0-16bx delay span.

U11,U12 Latch data on the rising edge of TMBs main clock_1x, and are the final synchronization stage.

ALCT DDR Receiver Sync Stages

ALCT receiver logic is designed to synchronize incoming data to TMBs main clock. It uses a small number of flip-flops, and is able to fit in a single FPGA column. It uses a different technique than the CFEB sync section, and does not have a delay stage.

The logic is shown schematically (Figure 3), but the actual TMB logic is written in behavioral Verilog.

U1A Latches incoming ALCT data din on the *falling* edge of clock_iob.
The user has already tuned the associated alct_rxd digital phase shifter so that clock_iob always latches stable data.

U1B Latches incoming din on the *rising* edge of clock_iob.
U1A and U1B comprise a single Double Data Rate (DDR) I/O-Block flip-flop, and are only shown here separately for clarity.

U2 Latches ALCT data transferred from U1A on the *rising* edge of clock_iob.
U2 now holds din_1st-in-time data, aligned with the rising edge of clock_iob, while U1B holds din_2nd-in-time data, also aligned with the rising edge of clock_iob.

In non-muonic firmware, this would be the end stage, because clock_iob would be the same as TMBs main clock.

U3,U4 Latch din_1st and din_2nd on either the *rising* or *falling* edge of TMBs main clock_1x. They are clocked at 2x the main clock frequency, but are enabled for only one edge direction, according to the value of the posneg bit. The edges of clock_2x are closely aligned to the edges of clock_1x.

U7 If posneg =1, U7 inverts a logic accessible copy of clock_1x, called clock_lac.
If posneg =0, U7 passes clock_lac un-inverted.

Because clock_lac is delayed by about 2ns after clock_1x, posneg =1 enables U3,U4 to latch data on the edge of clock_2x that corresponds to the *rising* edge of clock_1x, and posneg=0 enables latching on the *falling* edge.

U5,U6 Latch data on the rising edge of TMBs main clock_1x, and are the final synchronization stage.

ALCT DDR Transmitter Sync Stages

The ALCT transmitter logic is designed to synchronize TMBs outgoing data to the ALCT board's clock. It has a programmable 0-16bx delay stage, and a data-path multiplexer for sending test patterns to ALCT.

The logic is shown schematically (Figure 4), but the actual TMB logic is written in behavioral Verilog.

U1,U2 Select the transmitter data source, either ALCT data or the test pattern generator, according to the value of sync_mode.

U3,U4 Delay data by 1-to-16 bx, according to the VME-set delay-1 (the subtractor is not shown).

U3,U4 Delay data by 1-to-16 bx, according to delay[3:0] from a VME register.
Setting address=0 for U3,U4 gives a 1bx delay, so delay-1 is used to form the address.

U15 Subtracts 1 from delay[3:0] to form the shift register address for U3,U4.

U5,U6 Bypass delays U3,U4 if the delay is 0, thus giving a 0-16bx delay span.

U7,U8 Latch data on the rising edge of TMBs main clock_1x, and are the final main-clock stage.

In non-muonic firmware, this would be the end stage, because clock_iob would be the same as TMBs main clock.

U9,U10 Latch data on either the *rising* or *falling* edge of TMBs main clock_1x. They are clocked at 2x the main clock frequency, but are enabled for only one edge direction, according to the value of the posneg bit. The edges of clock_2x are closely aligned to the edges of clock_1x.

U14 If posneg =1, U14 inverts a logic accessible copy of clock_1x, called clock_lac.
If posneg =0, U14 passes clock_lac un-inverted.

Because clock_lac is delayed by about 2ns after clock_1x, posneg =1 enables U9,U10 to latch data on the edge of clock_2x that corresponds to the *rising* edge of clock_1x, and posneg=0 enables latching on the *falling* edge.

U12 Latches 1st-in-time data on the *rising* edge of clock_iob. It is a Double Date Rate I/O Block flip-flop that transmits 1st-in-time data on the rising edge of clock_iob, and 2nd-in-time on the falling edge.

U11 Holds 2nd-in-time data for ½ cycle while waiting for U12 to transmit 1st-in-time data.

Figure 2 CFEB Muonic Receiver

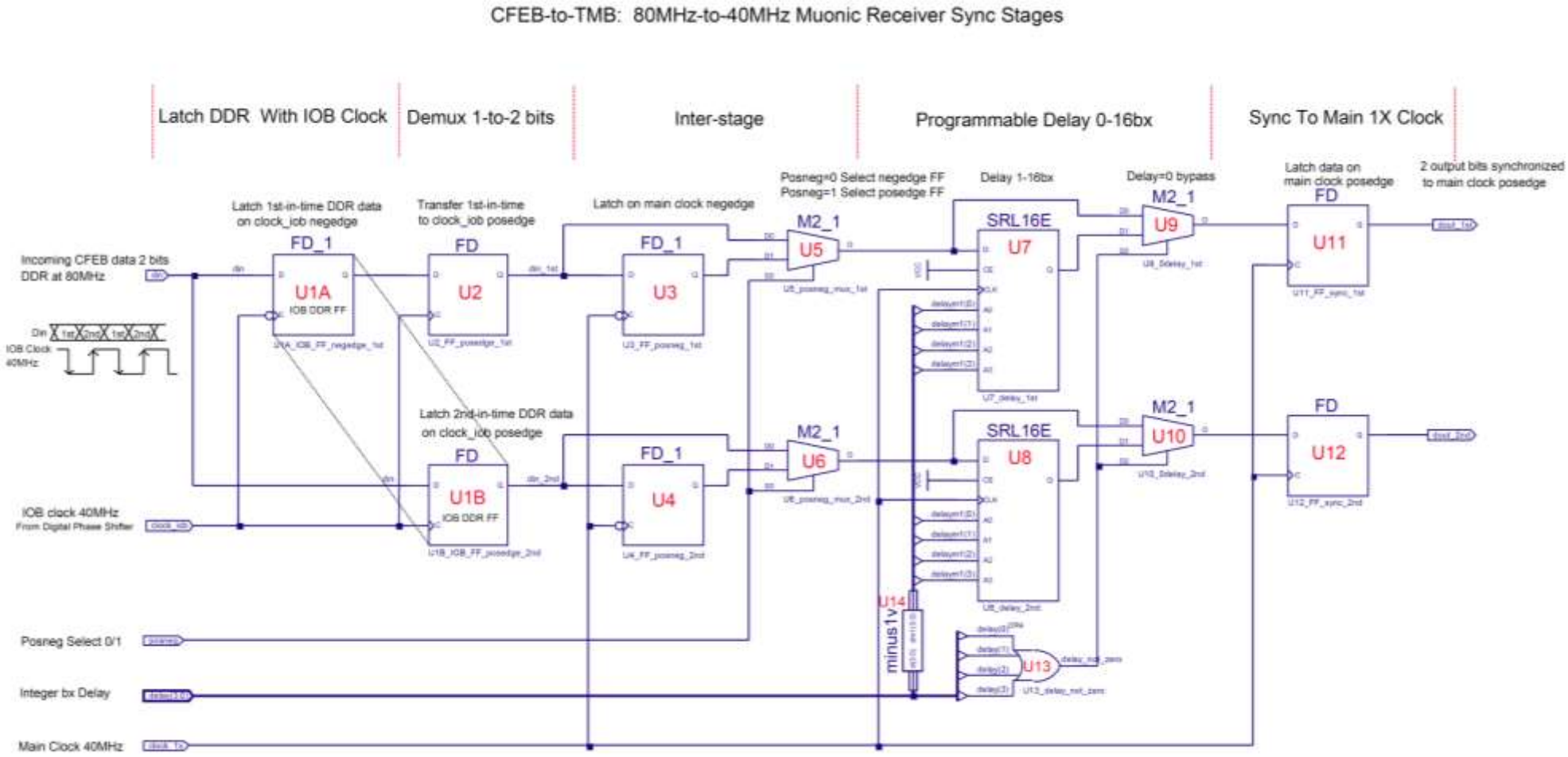


Figure 3 ALCT Muonic Receiver

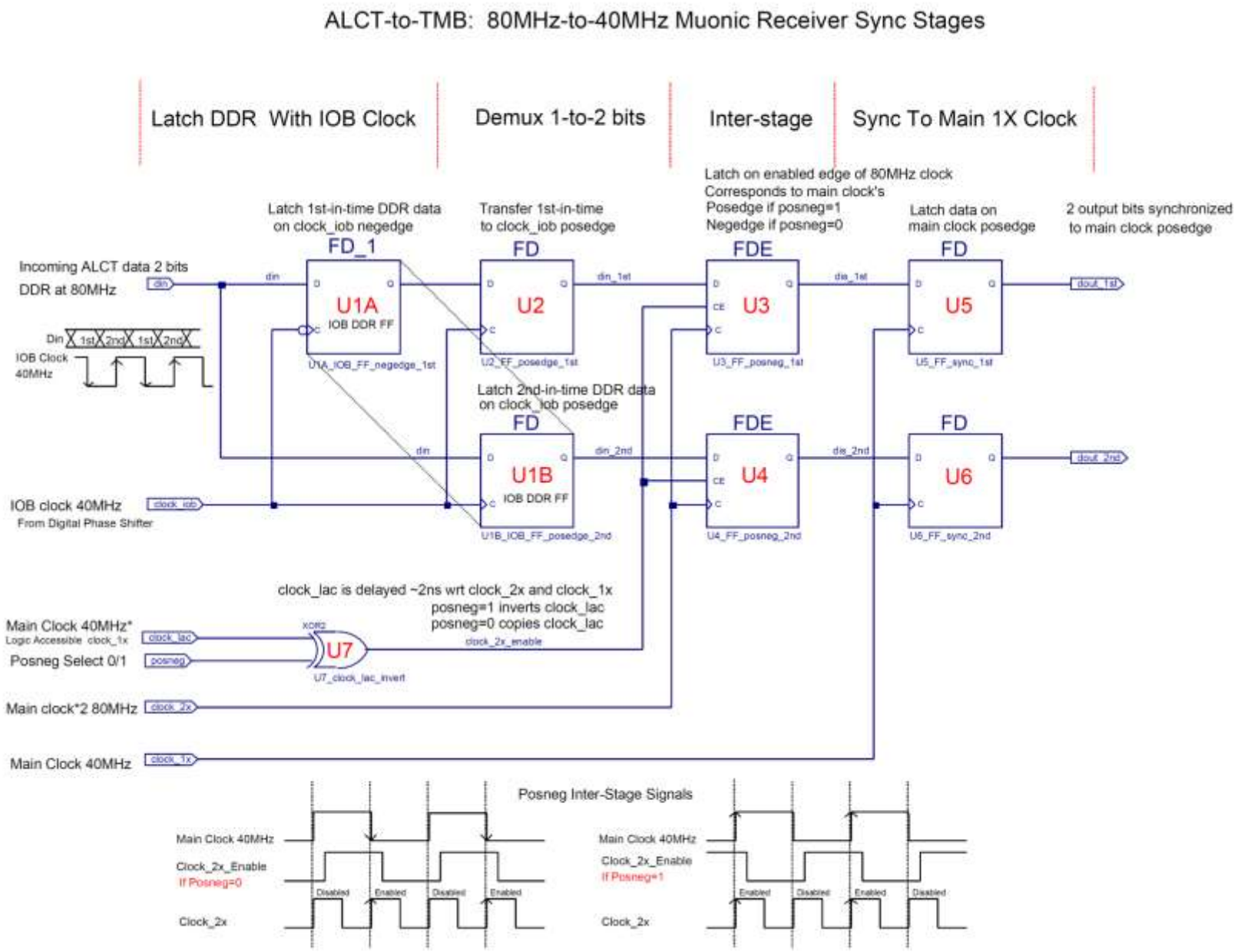
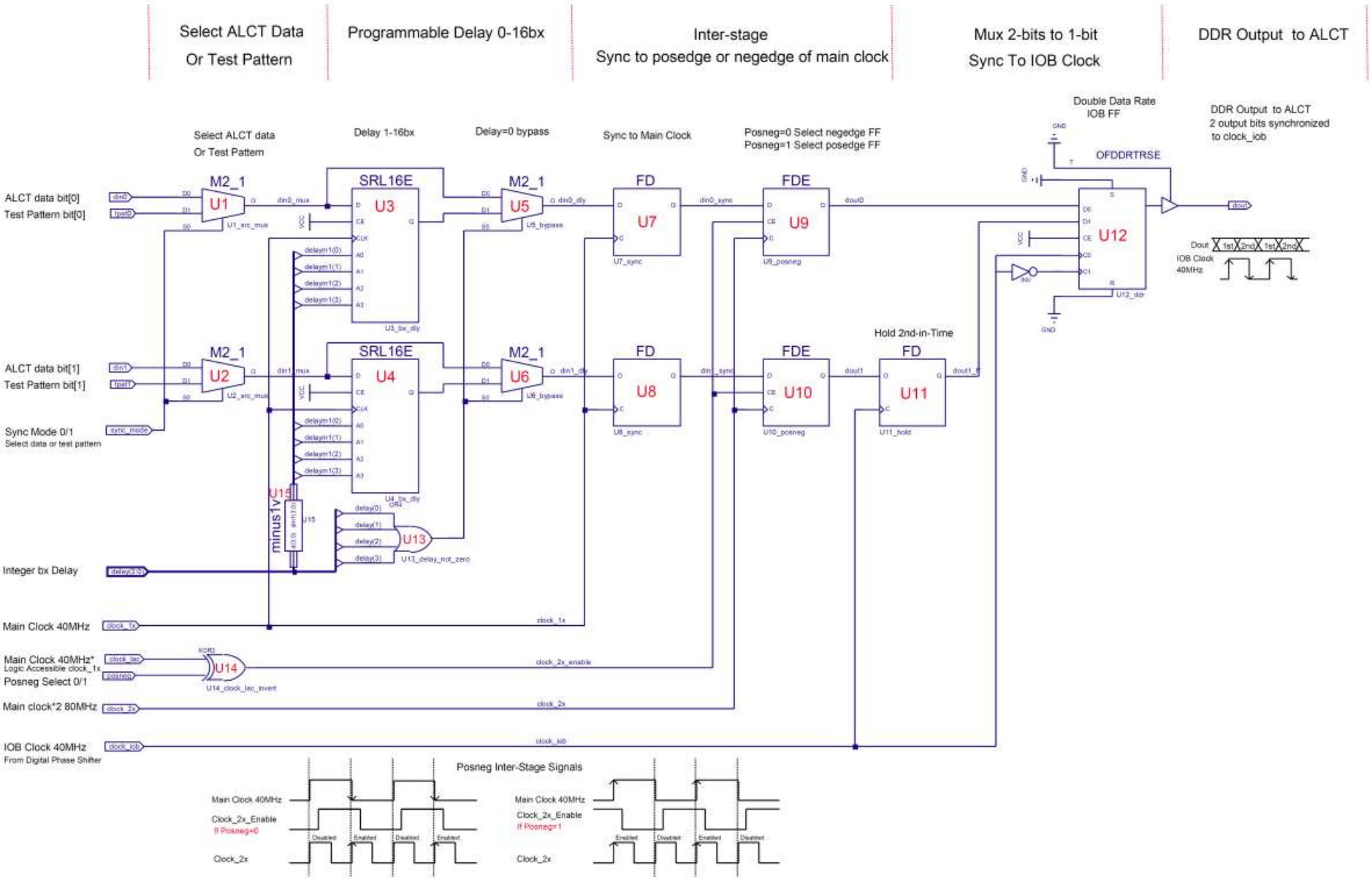


Figure 4 ALCT Muonic Transmitter



TTC Sequences

Trigger State

Start/Stop Triggering Sequence:

On hard_reset go to the StopTrigger state: TMBs CLCT processing machine goes to "idle".

On ttc_start_trigger, wait for the next BX0 (if ttc starts are allowed)
On the next ttc_bx0, resume triggering.

On ttc_stop_trigger (if ttc stops are allowed) or ttc_resync go the StopTrigger state.

Bunch Crossing Counter

Bunch Counter Reset

On ttc_resync or ttc_bxreset, preset the BXN, and hold the count.
On the next BX0 resume counting bunch crossings.

On subsequent BX0s, check that BXN is again at the preset value,
if not, set sync_error.

TMB further checks that when BXN is at the preset value, that a BX0
arrived, if not, set sync_error.

Resync

L1A Counter Reset

On ttc_resync or ccb_evcntreset, clear L1A event counters.

Buffer Reset

---

ttc_resync clears TMBs buffer-pointers, aborts any readout in progress,
and returns various state machines to their idle states.
(a similar condition to after power-up or hard-reset).

ALCT+CLCT Matching Algorithm

Matching Logic

LCT Duplication

```
// Fill in missing ALCT if CLCT has 2 muons, missing CLCT if ALCT has 2 muons
wire    no_alct  = !alct0_vpf;
wire    no_clct  = !clct0_vpf;

wire    one_alct = alct0_vpf && !alct1_vpf;
wire    one_clct = clct0_vpf && !clct1_vpf;

wire    two_alct = alct0_vpf && alct1_vpf;
wire    two_clct = clct0_vpf && clct1_vpf;

wire    dupe_alct = one_alct && two_clct;
wire    dupe_clct = one_clct && two_alct;

wire    [MXALCT-1:0]   alct_dummy = clct0_real[18:17] << 11;              // Inserts clct bxn into
wire    [MXCLCT-1:0]   clct_dummy = 0;                                    // frame for clct_only events

wire    [MXALCT-1:0]   alct0 = (no_alct  ) ? alct_dummy : alct0_real;     // Substitute dummy alct
wire    [MXALCT-1:0]   alct1 = (dupe_alct) ? alct0_real : alct1_real;

wire    [MXCLCT-1:0]   clct0 = (no_clct  ) ? clct0_real : clct0_real;     // Do not
wire    [MXCLCT-1:0]   clct1 = (dupe_clct) ? clct0_real : clct1_real;

wire first_vpf         = alct0_vpf || clct0_vpf;                          // First muon exists
wire second_vpf        = alct1_vpf || clct1_vpf;                          // Second muon exists
```

LCT Quality

```
module lct_quality (ACC,A,C,A4,C4,P,CPAT,Q);

// Ports
  input  ACC;     // ALCT accelerator muon bit
  input  A;       // bit: ALCT was found
  input  C;       // bit: CLCT was found
  input  A4;      // bit (N_A>=4), where N_A=number of ALCT layers
  input  C4;      // bit (N_C>=4), where N_C=number of CLCT layers
  input  [3:0] P; // 4-bit CLCT pattern number that is presently 1 for n-layer triggers, 2-10 for current
patterns, 11-15 "for future expansion".
  input  CPAT;    // bit for cathode .pattern trigger., i.e. (P>=2 && P<=10) at present
  output [3:0] Q; // 4-bit TMB quality output

// Quality-by-quality definition
  reg [3:0] Q;

  always @* begin

  if      ( !ACC && A4 && C4 && P==10 )          Q=15; // HQ muon, straight
  else if ( !ACC && A4 && C4 && (P==9 || P==8) ) Q=14; // HQ muon, slight bend
  else if ( !ACC && A4 && C4 && (P==7 || P==6) ) Q=13; // HQ muon, more
  else if ( !ACC && A4 && C4 && (P==5 || P==4) ) Q=12; // HQ muon, more
  else if ( !ACC && A4 && C4 && (P==3 || P==2) ) Q=11; // HQ muon, more
  //                                             Q=10; // reserved for HQ muons with future patterns
  //                                             Q=9;  // reserved for HQ muons with future patterns
  else if (  ACC       &&  A4       &&  C4 && CPAT ) Q=8; // HQ muon, but accel ALCT
  else if (          A && !A4       &&  C4 && CPAT ) Q=7; // HQ cathode, but marginal anode
  else if (            &&  A4 &&  C && !C4 && CPAT ) Q=6; // HQ anode, but marginal cathode
  else if (          A && !A4 &&  C && !C4 && CPAT ) Q=5; // marginal anode and cathode
  //                                               Q=4; // reserved for LQ muons with 2D information in
the future
  else if (          A        &&  C        && P==1 ) Q=3; // any match but layer CLCT
  else if (         !A        &&  C                ) Q=2; // some CLCT, no ALCT (unmatched)
  else if (          A        && !C                ) Q=1; // some ALCT, no CLCT (unmatched)
  else                                             Q=0; // should never be assigned
  end
```

MPC Format

```
assign  mpc0_frame0[6:0]       =       alct0_key[6:0];
assign  mpc0_frame0[10:7]      =       clct0_pat[3:0];
assign  mpc0_frame0[14:11]     =       lct0_quality[3:0] * lct0_vpf;
assign  mpc0_frame0[15]        =       lct0_vpf;

assign  mpc0_frame1[7:0]       =       {clct0_cfeb[2:0],clct0_key[4:0]};
assign  mpc0_frame1[8]         =       clct0_bend;
assign  mpc0_frame1[9]         =       clct_sync_err & tmb_sync_err_en[0];
assign  mpc0_frame1[10]        =       alct0_bxn[0];
assign  mpc0_frame1[11]        =       clct_bx0 * lct0_vpf;
assign  mpc0_frame1[15:12]     =       csc_id[3:0]* lct0_vpf;

assign  mpc1_frame0[6:0]       =       alct1_key[6:0];
assign  mpc1_frame0[10:7]      =       clct1_pat[3:0];
assign  mpc1_frame0[14:11]     =       lct1_quality[3:0] * lct1_vpf;
assign  mpc1_frame0[15]        =       lct1_vpf;

assign  mpc1_frame1[7:0]       =       {clct1_cfeb[2:0],clct1_key[4:0]};
assign  mpc1_frame1[8]         =       clct1_bend;
assign  mpc1_frame1[9]         =       clct_sync_err & tmb_sync_err_en[1] & lct1_vpf;
assign  mpc1_frame1[10]        =       alct1_bxn[0];
assign  mpc1_frame1[11]        =       clct_bx0 * lct1_vpf;
assign  mpc1_frame1[15:12]     =       csc_id[3:0] * lct1_vpf;

Transmission to MPC is         1st Frame[31:0] = {mpc1_frame0[15:0],mpc0_frame0[15:0]}
                               2nd Frame[31:0] = {mpc1_frame1[15:0],mpc0_frame1[15:0]}
```

Signal	1st in Time	2nd in Time	P3Apin	Test Point
mpc_tx[00]	alct_first_key[0]	clct_first_key[0]	A1	TP331-1
mpc_tx[01]	alct_first_key[1]	clct_first_key[1]	B1	TP331-2
mpc_tx[02]	alct_first_key[2]	clct_first_key[2]	D1	TP331-3
mpc_tx[03]	alct_first_key[3]	clct_first_key[3]	E1	TP331-4
mpc_tx[04]	alct_first_key[4]	clct_first_key[4]	A2	TP331-5
mpc_tx[05]	alct_first_key[5]	clct_first_key[5]	B2	TP331-6
mpc_tx[06]	alct_first_key[6]	clct_first_key[6]	D2	TP331-7
mpc_tx[07]	clct_first_pat[0]	clct_first_key[7]	E2	TP331-8
mpc_tx[08]	clct_first_pat[1]	clct_first_bend	A3	TP332-1
mpc_tx[09]	clct_first_pat[2]	clct_first_sync_err	B3	TP332-2
mpc_tx[10]	clct_first_pat[3]	alct_first_bxn[0]	D3	TP332-3
mpc_tx[11]	lct_first_quality[0]	clct_first_bx0_local	E3	TP332-4
mpc_tx[12]	lct_first_quality[1]	csc_id[0]	A4	TP332-5
mpc_tx[13]	lct_first_quality[2]	csc_id[1]	B4	TP332-6
mpc_tx[14]	lct_first_quality[3]	csc_id[2]	D4	TP332-7
mpc_tx[15]	first_vpf	csc_id[3]	E4	TP332-8
mpc_tx[16]	alct_second_key[0]	clct_second_key[0]	A5	TP341-1
mpc_tx[17]	alct_second_key[1]	clct_second_key[1]	B5	TP341-2
mpc_tx[18]	alct_second_key[2]	clct_second_key[2]	D5	TP341-3
mpc_tx[19]	alct_second_key[3]	clct_second_key[3]	E5	TP341-4
mpc_tx[20]	alct_second_key[4]	clct_second_key[4]	A6	TP341-5
mpc_tx[21]	alct_second_key[5]	clct_second_key[5]	B6	TP341-6
mpc_tx[22]	alct_second_key[6]	clct_second_key[6]	D6	TP341-7
mpc_tx[23]	clct_second_pat[0]	clct_second_key[7]	E6	TP341-8
mpc_tx[24]	clct_second_pat[1]	lct_second_bend	A7	TP342-1
mpc_tx[25]	clct_second_pat[2]	clct_second_sync_err	B7	TP342-2
mpc_tx[26]	clct_second_pat[3]	alct_second_bxn[0]	D7	TP342-3
mpc_tx[27]	lct_second_quality[0]	clct_second_bx0_local	E7	TP342-4
mpc_tx[28]	lct_second_quality[1]	csc_id[0]	A8	TP342-5
mpc_tx[29]	lct_second_quality[2]	csc_id[1]	B8	TP342-6
mpc_tx[30]	lct_second_quality[3]	csc_id[2]	D8	TP342-7

mpc_tx[31]	second_vpf	csc_id[3]	E8	TP342-8

VME Registers

Addressing Modes

TMB2005 responds to A24D16 VME addressing modes:

Address Modifier 39_{16}, A24 non-privileged mode
Address Modifier $3D_{16}$, A24 supervisor mode

It does not respond to byte-addressing modes, so all valid addresses must be even numbers.

Base Address

TMB2005s "base address" bits A[23:19] select which TMB is being addressed by the VME crate controller. The base address is determined either by the 5 VME-backplane-slot Geographic Address bits or by the Local Address set by two on-board hexadecimal rotary switches. Shunt SH62 selects between Geographic [1-2] and Local [2-3] modes.

A[23:19] = VME Crate Slot Geographic Address, (Slot= 2 to 21)$_{10}$	SH62 [1-2]
A[23:19] = Hexadecimal Switch Address SW2x16+SW1	SH62 [2-3]

Multiple TMBs can be addressed simultaneously using a Global Address:

A[23:19] = 26_{10} Addresses all TMBs in parallel
A[23:19] = 27_{10} Address all peripheral crate modules

Boot Register

When geographic addressing is used, the S2/S1 hexadecimal switches should be set to 1Ah, which allows the hardware Boot Register to respond to the slot 26_{10} global address.

The Boot Register responds to all even VME addresses + base between 70000h and 7FFFEh to allow block-mode VME writes.

Register Addresses

Address Hexadecimal Add to Base	Register Name	Description
70000	ADR_BOOT	Hardware Bootstrap Register
00	ADR_IDREG0	ID Register 0
02	ADR_IDREG1	ID Register 1
04	ADR_IDREG2	ID Register 2
06	ADR_IDREG3	ID Register 3
08	ADR_VME_STATUS	VME Status Register
0A	ADR_VME_ADR0	VME Address read-back
0C	ADR_VME_ADR1	VME Address read-back
0E	ADR_LOOPBK	Loop-back Register
10	ADR_USR_JTAG	User JTAG
12	ADR_PROM	PROM
14	ADR_DDDSM	3D3444 State Machine Register + Clock DCMs
16	ADR_DDD0	3D3444 Delay Chip 0
18	ADR_DDD1	3D3444 Delay Chip 1
1A	ADR_DDD2	3D3444 Delay Chip 2
1C	ADR_DDDOE	3D3444 Delay Chip Output Enables
1E	ADR_RATCTRL	RAT Module Control
20	ADR_STEP	Step Register
22	ADR_LED	Front Panel +On-Board LEDs
24	ADR_ADC	ADCs
26	ADR_DSN	Digital Serials
28	ADR_MOD_CFG	TMB Configuration
2A	ADR_CCB_CFG	CCB Configuration
2C	ADR_CCB_TRIG	CCB Trigger Control
2E	ADR_CCB_STAT0	CCB Status
30	ADR_ALCT_CFG	ALCT Configuration
32	ADR_ALCT_INJ	ALCT Injector Control
34	ADR_ALCT0_INJ	ALCT Injected ALCT0
36	ADR_ALCT1_INJ	ALCT Injected ALCT1
38	ADR_ALCT_STAT	ALCT Sequencer Control/Status
3A	ADR_ALCT0_RCD	ALCT LCT0 Received by TMB
3C	ADR_ALCT1_RCD	ALCT LCT1 Received by TMB
3E	ADR_ALCT_FIFO	ALCT FIFO RAM Status
40	ADR_DMB_MON	DMB Monitored signals

42	ADR_CFEB_INJ	CFEB Injector Control
44	ADR_CFEB_INJ_ADR	CFEB Injector RAM address
46	ADR_CFEB_INJ_WDATA	CFEB Injector Write Data
48	ADR_CFEB_INJ_RDATA	CFEB Injector Read Data
4A	ADR_HCM001	CFEB0 Ly0,Ly1 Hot Channel Mask
4C	ADR_HCM023	CFEB0 Ly2,Ly3 Hot Channel Mask
4E	ADR_HCM045	CFEB0 Ly4,Ly5 Hot Channel Mask
50	ADR_HCM101	CFEB1 Ly0,Ly1 Hot Channel Mask
52	ADR_HCM123	CFEB1 Ly2,Ly3 Hot Channel Mask
54	ADR_HCM145	CFEB1 Ly4,Ly5 Hot Channel Mask
56	ADR_HCM201	CFEB2 Ly0,Ly1 Hot Channel Mask
58	ADR_HCM223	CFEB2 Ly2,Ly3 Hot Channel Mask
5A	ADR_HCM245	CFEB2 Ly4,Ly5 Hot Channel Mask
5C	ADR_HCM301	CFEB3 Ly0,Ly1 Hot Channel Mask
5E	ADR_HCM323	CFEB3 Ly2,Ly3 Hot Channel Mask
60	ADR_HCM345	CFEB3 Ly4,Ly5 Hot Channel Mask
62	ADR_HCM401	CFEB4 Ly0,Ly1 Hot Channel Mask
64	ADR_HCM423	CFEB4 Ly2,Ly3 Hot Channel Mask
66	ADR_HCM445	CFEB4 Ly4,Ly5 Hot Channel Mask
68	ADR_SEQ_TRIG_EN	Sequencer Trigger Source Enables
6A	ADR_SEQ_TRIG_DLY0	Sequencer Trigger Source Delays
6C	ADR_SEQ_TRIG_DLY1	Sequencer Trigger Source Delays
6E	ADR_SEQ_ID	Sequencer Board + CSC ID
70	ADR_SEQ_CLCT	Sequencer CLCT Configuration
72	ADR_SEQ_FIFO	Sequencer FIFO Configuration
74	ADR_SEQ_L1A	Sequencer L1A Configuration
76	ADR_SEQ_OFFSET0	Sequencer Counter Offsets
78	ADR_SEQ_CLCT0	Sequencer Latched CLCT0
7A	ADR_SEQ_CLCT1	Sequencer Latched CLCT1
7C	ADR_SEQ_TRIG_SRC	Sequencer Trigger Source Read-back
7E	ADR_DMB_RAM_ADR	Sequencer RAM Address
80	ADR_DMB_RAM_WDATA	Sequencer RAM Write Data
82	ADR_DMB_RAM_WDCNT	Sequencer RAM Word Count
84	ADR_DMB_RAM_RDATA	Sequencer RAM Read Data
86	ADR_TMB_TRIG	TMB Trigger Configuration / MPC Accept
88	ADR_MPC0_FRAME0	MPC0 Frame 0 Data sent to MPC
8A	ADR_MPC0_FRAME1	MPC0 Frame 1 Data sent to MPC
8C	ADR_MPC1_FRAME0	MPC1 Frame 0 Data sent to MPC
8E	ADR_MPC1_FRAME1	MPC1 Frame 1 Data sent to MPC

90	ADR_MPC_INJ	MPC Injector Control
92	ADR_MPC_RAM_ADR	MPC Injector RAM address
94	ADR_MPC_RAM_WDATA	MPC Injector RAM Write Data
96	ADR_MPC_RAM_RDATA	MPC Injector RAM Read Data
98	ADR_SCP_CTRL	Scope control
9A	ADR_SCP_RDATA	Scope read data
9C	ADR_CCB_CMD	CCB TTC Command Generator
9E	ADR_BUF_STAT0	Buffer Status
A0	ADR_BUF_STAT1	Buffer Status
A2	ADR_BUF_STAT2	Buffer Status
A4	ADR_BUF_STAT3	Buffer Status
A6	ADR_BUF_STAT4	Buffer Status
A8	ADR_ALCT_FIFO1	ALCT Raw hits RAM Control
AA	ADR_ALCT_FIFO2	ALCT Raw hits RAM data
AC	ADR_SEQMOD	Sequencer Trigger Modifiers
AE	ADR_SEQSM	Sequencer Machine State
B0	ADR_SEQCLCTM	Sequencer CLCT msbs
B2	ADR_TMBTIM	TMB Timing for ALCT*CLCT coincidence
B4	ADR_LHC_CYCLE	LHC Cycle period, Maximum BXN+1
B6	ADR_RPC_CFG	RPC Configuration
B8	ADR_RPC_RDATA	RPC Sync Mode Read Data
BA	ADR_RPC_RAW_DELAY	RPC Raw Hits Delay + RPC BXN Differences
BC	ADR_RPC_INJ	RPC Injector Control
BE	ADR_RPC_INJ_ADR	RPC Injector RAM Addresses
C0	ADR_RPC_INJ_WDATA	RPC Injector Write Data
C2	ADR_RPC_INJ_RDATA	RPC Injector Read Data
C4	ADR_RPC_TBINS	RPC FIFO Time Bins
C6	ADR_RPC0_HCM	RPC0 Hot Channel Mask
C8	ADR_RPC1_HCM	RPC1 Hot Channel Mask
CA	ADR_BX0_DELAY	BX0 to MPC Delays
CC	ADR_NON_TRIG_RO	Non-triggering Event Enables
CE	ADR_SCP_TRIG	Scope Trigger Source Channel
D0	ADR_CNT_CTRL	Status Counter Control
D2	ADR_CNT_RDATA	Status Counter Data
D4	ADR_JTAGSM0	JTAG State Machine Control (reads JTAG PROM)
D6	ADR_JTAGSM1	JTAG State Machine Word Count
D8	ADR_JTAGSM2	JTAG State Machine Checksum

DA	ADR_VMESM0	VME State Machine Control (reads VME PROM)
DC	ADR_VMESM1	VME State Machine Word Count
DE	ADR_VMESM2	VME State Machine Checksum
E0	ADR_VMESM3	Number of VME Addresses Written by VMESM
E2	ADR_VMESM4	VME State Machine Write-Data Check
E4	ADR_DDDRSM	RAT 3D3444 State Machine Control
E6	ADR_DDDR0	RAT 3D3444 RPC Delays
E8	ADR_UPTIME	Uptime Counter
EA	ADR_BDSTATUS	Board Status Summary
EC	ADR_BXN_CLCT	CLCT BXN At CLCT-Pretrigger
EE	ADR_BXN_ALCT	ALCT BXN At ALCT-Valid-Pattern-Flag
F0	ADR_LAYER_TRIG	Layer-Trigger Mode
F2	ADR_ISE_VERSION	ISE Version + Service Pack
F4	ADR_TEMP0	Pattern Finder Pre-Trigger
F6	ADR_TEMP1	CLCT Separation
F8	ADR_TEMP2	CLCT Separation RAM Data
FA	ADR_PARITY	
FC	ADR_CCB_STAT1	
FE	ADR_BXN_L1A	CLCT BXN at last L1A arrival
100	ADR_L1A_LOOKBACK	L1A Lookback distance
102	ADR_SEQ_DEBUG	Sequencer debug signals
104	ADR_ALCT_SYNC_CTRL	ALCT sync mode control
106	ADR_ALCT_SYNC_TXDATA_1ST	ALCT sync mode transmit data 1st
108	ADR_ALCT_SYNC_TXDATA_2ND	ALCT sync mode transmit data 2nd
10A	ADR_SEQ_OFFSET1	Sequencer Counter Offsets Continued
10C	ADR_MINISCOPE	Internal 16 Channel Digital Miniscope
10E	ADR_PHASER0	ALCT rxd delay digital phase shifter
110	ADR_PHASER1	ALCT txd delay digital phase shifter
112	ADR_PHASER2	CFEB0 rxd delay digital phase shifter
114	ADR_PHASER3	CFEB1 rxd delay digital phase shifter
116	ADR_PHASER4	CFEB2 rxd delay digital phase shifter
118	ADR_PHASER5	CFEB3 rxd delay digital phase shifter
11A	ADR_PHASER6	CFEB4 rxd delay digital phase shifter

11C	ADR_DELAY0_INT	CFEB0-3 DDR RxD Interstage delays
11E	ADR_DELAY1_INT	CFEB4 DDR RxD Interstage delays Continued
120	ADR_SYNC_ERR_CTRL	Synchronization Error Control
122	ADR_CFEB_BADBITS_CTRL	CFEB Bad Bit Control/Status
124	ADR_CFEB_BADBITS_TIMER	CFEB Bad Bit Check Interval
126	ADR_CFEB0_BADBITS_LY01	CFEB0 Bad Bits Array
128	ADR_CFEB0_BADBITS_LY23	CFEB0 Bad Bits Array
12A	ADR_CFEB0_BADBITS_LY45	CFEB0 Bad Bits Array
12C	ADR_CFEB1_BADBITS_LY01	CFEB1 Bad Bits Array
12E	ADR_CFEB1_BADBITS_LY23	CFEB1 Bad Bits Array
130	ADR_CFEB1_BADBITS_LY45	CFEB1 Bad Bits Array
132	ADR_CFEB2_BADBITS_LY01	CFEB2 Bad Bits Array
134	ADR_CFEB2_BADBITS_LY23	CFEB2 Bad Bits Array
136	ADR_CFEB2_BADBITS_LY45	CFEB2 Bad Bits Array
138	ADR_CFEB3_BADBITS_LY01	CFEB3 Bad Bits Array
13A	ADR_CFEB3_BADBITS_LY23	CFEB3 Bad Bits Array
13C	ADR_CFEB3_BADBITS_LY45	CFEB3 Bad Bits Array
13E	ADR_CFEB4_BADBITS_LY01	CFEB4 Bad Bits Array
140	ADR_CFEB4_BADBITS_LY23	CFEB4 Bad Bits Array
142	ADR_CFEB4_BADBITS_LY45	CFEB4 Bad Bits Array
144	ADR_ALCT_STARTUP_DELAY	ALCT startup delay milliseconds for Spartan-6
146	ADR_ALCT_STARTUP_STATUS	ALCT startup delay machine status
	Virtex-6 VME Registers	
148	ADR_V6_SNAP12_QPLL	Virtex-6 SNAP12 Serial interface + QPLL status
14A	ADR_V6_GTX_RX_ALL	Virtex-6 GTX common control and status
14C	ADR_V6_GTX_RX0	Virtex-6 GTX0 control and status
14E	ADR_V6_GTX_RX1	Virtex-6 GTX1 control and status
150	ADR_V6_GTX_RX2	Virtex-6 GTX2 control and status
152	ADR_V6_GTX_RX3	Virtex-6 GTX3 control and status
154	ADR_V6_GTX_RX4	Virtex-6 GTX4 control and status
156	ADR_V6_GTX_RX5	Virtex-6 GTX5 control and status
158	ADR_V6_GTX_RX6	Virtex-6 GTX6 control and status
15A	ADR_V6_SYSMON	Virtex-6 Sysmon ADC
15C	ADR_V6_CFEB_BADBITS_CTRL	CFEB Bad Bit Control/Status extends Adr 122
15E	ADR_V6_CFEB5_BADBITS_LY01	CFEB5 Bad Bit Array

160	ADR_V6_CFEB5_BADBITS_LY23	CFEB5 Bad Bit Array
162	ADR_V6_CFEB5_BADBITS_LY45	CFEB5 Bad Bit Array
164	ADR_V6_CFEB6_BADBITS_LY01	CFEB6 Bad Bit Array
166	ADR_V6_CFEB6_BADBITS_LY23	CFEB6 Bad Bit Array
168	ADR_V6_CFEB6_BADBITS_LY45	CFEB6 Bad Bit Array
16A	ADR_V6_PHASER7	Phaser 7 cfeb5_rxd phase
16C	ADR_V6_PHASER8	Phaser 8 cfeb6_rxd phase
16E	ADR_V6_HCM501	CFEB5 Ly0,Ly1 Hot Channel Mask
170	ADR_V6_HCM523	CFEB5 Ly2,Ly3 Hot Channel Mask
172	ADR_V6_HCM545	CFEB5 Ly4,Ly5 Hot Channel Mask
174	ADR_V6_HCM601	CFEB6 Ly0,Ly1 Hot Channel Mask
176	ADR_V6_HCM623	CFEB6 Ly2,Ly3 Hot Channel Mask
178	ADR_V6_HCM645	CFEB6 Ly4,Ly5 Hot Channel Mask
17A	ADR_V6_EXTEND	DCFEB 7-bit extensions to 5 bit fields in 0x42,68

Register Definitions

Adr 70000_{16} ADR_BOOT Hardware Bootstrap Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R=tdo	R=ready	hard reset RPC/RAT	/mez clock enable	/fpga vme_en	/en_fpga reset_alct	hard reset TMB	hard reset ALCT	JTAG source vme/fpga	sel3	sel2	sel1	sel0	tck	tms	Tdi

Bit	Dir	Signal	Default	Description
[0]	RW	jtag_vme1 (tdi)	0	vme tdi
[1]	RW	jtag_vme2 (tms)	0	vme tms
[2]	RW	jtag_vme3 (tck)	0	vme tck
[3]	RW	sel_vme0	0	00XX ALCT JTAG Chain
[4]	RW	sel_vme1	0	01XX TMB Mezzanine FPGA + FPGA PROMs Chain
[5]	RW	sel_vme2	0	10XX TMB User PROMs JTAG chain
[6]	RW	sel_vme3	0	11XX TMB FPGA User JTAG chain
[7]	RW	vme/usr_en	0	1=JTAG sourced by Bootstrap Register, 0= from FPGA
[8]	RW	hard_reset_alct_vme	0	1=Hard reset to ALCT FPGA
[9]	RW	hard_reset_tmb_vme	0	1=Hard reset to TMB FPGA
[10]	RW	/en_fpga_reset_alct	0	0=Allow TMB FPGA to hard reset ALCT
[11]	RW	/fpga_vme_en	0	0=Allow TMB FPGA to issue VME commands
[12]	RW	/mez_clock_en	0	0=Enable TMB FPGA mezzanine clock
[13]	RW	hard_reset_rpc	0	1=Hard reset to RPC (RAT) FPGA
[14]	R	vme_ready	x	1=FPGA vme logic indicates ready
[15]	R	jtag_vme0 (tdo)	0	vme tdo
[14]	W	unassigned	-	No connection on PCB
[15]	W	unassigned	-	No connection on PCB

Adr 00 ADR_IDREG0 ID Register 0

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	ga4	ga3	ga2	ga1	ga0	fvers3	fvers2	fvers1	fvers0	ftype3	ftype2	ftype1	ftype0

Bits	Dir	Typical	Description
[03:00]	R	C	Firmware type, C=Normal CLCT/TMB, D=Debug loopback
[07:04]	R	D	Firmware version code
[12:08]	R	15	Geographic address for this board
[15:13]	R	0	Unassigned

Adr 02 ADR_IDREG1 ID Register 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
month msd3	month msd2	month msd1	month msd0	month lsd3	month lsd2	month lsd1	month lsd0	day msd3	day msd2	day msd1	day msd0	day lsd3	day lsd2	day lsd1	day lsd0

Bits	Dir	Typical	Description
[07:00]	R	09	DD Firmware Version Day (BCD)
[15:08]	R	04	MM Firmware Version Month (BCD)

Adr 04 ADR_IDREG2 ID Register 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
year digit33	year digit32	year digit31	year digit30	year digit23	year digit22	year digit21	year digit20	year digit13	year digit12	year digit11	year digit10	year digit03	year digit02	year digit01	year digit00

Bits	Dir	Typical	Description
[15:00]	R	2007	YYYY Firmware Version Year (BCD)

Adr 06 ADR_IDREG3 ID Register 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rev code 15	rev code 14	rev code 13	rev code 12	rev code 11	rev code 10	rev code 9	rev code 8	rev code 7	rev code 6	rev code 5	rev code 4	rev code 3	rev code 2	rev code 1	rev code 0

Bits	Dir	Typical	Description
[15:00]	R		Firmware Revcode (as stored in raw hits header)

Adr 08 ADR_VME_STATUS VME Status Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMB ready	local/ geo	iack	acfail	sysreset	sysfail	sysclk	ds1	as	lword	gap	ga4	ga3	ga2	ga1	ga0

Bits	Dir	Typical	Description
[04:00]	R		Crate slot Geographic Address
[05]	R		Crate slot Geographic Address Parity
[06]	R		VME signal lword
[07]	R		VME signal as
[08]	R		VME signal ds1
[09]	R		VME signal sysclk
[10]	R		VME signal sysfail
[11]	R		VME signal sysreset
[12]	R		VME signal acfail
[13]	R		VME signal iack
[14]	R		1=Address mode set to local, 0=Geographic
[15]	R		1=TMB reports ready to boot register

Adr 0A ADR_VME_ADR0 VME Address Read-Back

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
a15	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	lword

Bits	Dir	Typical	Description
[15:00]	R	a[15:0]	VME Address captured at last write cycle {a[15:1},lword}

Adr 0C ADR_VME_ADR1 VME Address Read-Back

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TMB ready	local/ geo	iack	acfail	sysreset	sysfail	sysclk	ds1	as	lword	gap	ga4	ga3	ga2	ga1	ga0

Bits	Dir	Typical	Description
[07:00]	R	a[23:16]	VME Address captured at last write cycle
[13:08]	R	am[5:0]	VME Address modifier
[15:14]	R	0	Unassigned

Adr 0E ADR_LOOPBK Loop-Back Control Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	dmb tx res 2	dmb tx res 1	dmb tx res 0	gtl_ oe	gtl_ loop	dmb_ oe	dmb_ loop	rpc loop bdtest	rpc_ loop_ tmb	rpc_ loop_ rat	alct_ txoe	alct_ rxoe	alct_ loop	cfeb_ oe

Bits	Dir	Signal	Default	Description
[00]	R	cfeb_oe	1	1=CFEB output enable
[01]	R	alct_loop	0	0=No ALCT loop-back
[02]	RW	alct_rxoe	1	1=Enable RAT ALCT LVDS receiver, 0=power down
[03]	RW	alct_txoe	1	1=Enable RAT ALCT LVDS transmitter, 0=power down
[04]	R	rpc_loop_rat	0	1=RAT FPGA enters loop-back mode
[05]	R	rpc_loop_bdtest	0	1=En RPC Loop-back (no RAT), used only in bdtest firmware
[06]	R	rpc_loop_tmb	0	1=TMBs RAT backplane ICs loop-back mode
[07]	R	dmb_loop	0	0=No DMB loop-back
[08]	R	dmb_oe	0	0=DMB driver enable
[09]	R	gtl_loop	0	0=No GTL loop-back
[10]	R	gtl_oe	0	0=Enable GTL outputs
[13:11]	RW	dmb_tx_reserved[2:0]	0	dmb_tx[48:46] unused, set to 0
[15:14]	RW	--		Unassigned

Adr 10 ADR_USR_JTAG User JTAG Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
tdo usr	0	0	0	0	0	0	0		sel3 usr	sel2 usr	sel1 usr	sel0 usr	tck usr	tms usr	tdi usr

Bits	Dir	Signal	Description
[00]	RW	tdi_usr	User JTAG Chain TDI (output from FPGA)
[01]	RW	tms_usr	User JTAG Chain TMS
[02]	RW	tck_usr	User JTAG Chain TCK
[06:03]	RW	sel_usr[3:0]	User JTAG Chain Select, 0=ALCT,1=Mez,2=UserPROMs,3=UserChain
[13:07]	RW	--	Unassigned
[14]	RW	wr_usr_jtag_dis	1=disable write access to ADR_USR_JTAG, set in Adr D4[11]
[15]	R	tdo_usr	User JTAG Chain TDO (input to FPGA)

Adr 12 ADR_PROM User PROMs Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	prom_src	prom1ce	prom1oe	prom1clk	prom0ce	prom0oe	prom0clk	prom_led7	prom_led6	prom_led5	prom_led4	prom_led3	prom_led2	prom_led1	prom_led0

Bits	Dir	Signal	Default	Description
[07:00]	RW	prom_led[7:0]	CD	PROM data bus shared with On-Board LEDs
[08]	RW	prom0_clk	0	PROM 0 clock
[09]	RW	prom0_oe	0	PROM 0 output enable
[10]	RW	prom0_ce	1	PROM 0 /chip_enable
[11]	RW	prom1_clk	0	PROM 1 clock
[12]	RW	prom1_oe	0	PROM 1 output enable
[13]	RW	prom1_ce	1	PROM 1 /chip_enable
[14]	RW	prom_src	0	Data bus 0=on-board LEDs, 1=enabled PROM
[15]	RW	--	0	Unassigned

Adr 14 ADR_DDDSM 3D3444 State Machine Control + DCM Lock Status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rpc lock	dcc lock	mpc lock	alctd lock	alct rxclk lock	tmb clock1 lock	tmb0d lock	tmb clock0 lock	ddd verify	ddds m busy	auto start	serial from	serial to ddd	adr latch	ddd clock	ddd start

Bits	Dir	Signal	Default	Description
[00]	RW	ddd_start_vme	0	Start DDD State Machine
[01]	RW	ddd_clock	0	DDD manual-mode clock
[02]	RW	ddd_adr_latch	1	DDD manual-mode address latch, active low
[03]	RW	ddd_serial_in	0	Serial data to DDD chain
[04]	RW	ddd_serial_out	0	Serial data from DDD chain
[05]	RW	ddd_auto_start	1	DDD State Machine autostart state
[06]	R	ddd_busy	0	DDD State Machine busy
[07]	R	ddd_verify_ok	1	DDD data read back verified OK
[08]	R	lock_tmb_clock0	1	TMB clock 0 DCM locked
[09]	R	lock_tmb_clock0d	1	TMB clock 0d DCM locked
[10]	R	lock_tmb_clock1	1	TMB clock 1 DCM locked
[11]	R	lock_alct_rxclock	1	ALCT rxclock DCM locked
[12]	R	lock_alct_clockd	1	ALCT rxclockd DCM locked
[13]	R	lock_mpc_clock	1	CFEB rxd clock DCM locked (was mpc)
[14]	R	lock_dcc_clock	1	DCC clock DCM locked
[15]	R	lock_rpc_clock	1	RPC clock DCM locked

Adr 16 ADR_DDD0 3D3444 Chip 0 Delays, 1 step = 2ns

Bits	Dir	Signal	Default	Description
[03:00]	RW	delay_ch0[3:0]	0	alct_tof_delay, shift entire ALCT in clockspace
[07:04]	RW	delay_ch1[3:0]	1	alct_rxclock delay, not used in muonic firmware
[11:08]	RW	delay_ch2[3:0]	6	DMB tx clock
[15:12]	RW	delay_ch3[3:0]	9	RPC tx clock

Adr 18 ADR_DDD1 3D3444 Chip 1 Delays, 1 step = 2ns

Bits	Dir	Signal	Default	Description
[03:00]	RW	delay_ch4[3:0]	0	tmb_clock1, not used
[07:04]	RW	delay_ch5[3:0]	0	mpc_clock not used
[11:08]	RW	delay_ch6[3:0]	0	cfeb_tof_delay, shift all cfebs in clockspace
[15:12]	RW	delay_ch7[3:0]	7	CFEB 0 clock

Adr 1A ADR_DDD2 3D3444 Chip 2 Delays, 1 step = 2ns

Bits	Dir	Signal	Default	Description
[03:00]	RW	delay_ch8[3:0]	7	CFEB 1 clock
[07:04]	RW	delay_ch9[3:0]	7	CFEB 2 clock
[11:08]	RW	delay_ch10[3:0]	7	CFEB 3 clock
[15:12]	RW	delay_ch11[3:0]	7	CFEB 4 clock

Adr 1C ADR_DDDOE 3D3444 Chip Output Enables

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	cfeb 4	cfeb 3	cfeb 2	cfeb 1	cfeb 0	dcc	mpc	tmb1	rpc tx	dmb tx	alct rx	alct tx

Bits	Dir	Signal	Default	Description
[11:00]	RW	ddd_oe[11:0]	FFF	Bit(n)=1=Enable DDD output channel n
[15:12]	RW	Unassigned	0	Unassigned

Adr 1E ADR_RATCTRL RAT Module Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	rpc dsn en	rpc free	rpc lptm	rpc posneg	rpc sync

Bits	Dir	Signal	Default	Description
[0]	RW	rpc_sync	0	1=RPC 80MHz sync pattern mode
[1]	RW	rpc_posneg	0	1=shift RPC data ½ cycle in RAT FPGA + dsn
[2]	RW	rpc_lptmb	0	Not used (for matching rpc_tx array)
[3]	RW	rpc_free_tx[0]	0	Unassigned
[4]	RW	rat_dsn_en	0	1=Enable RAT dsn readout
[15:5]	RW	--	0	Unassigned

Adr 20 ADR_STEP Clock Single-Step + Hard Resets

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	/tmb hard	/alct hard	alct clken	cfeb4 clken	cfeb3 clken	cfeb2 clken	cfeb1 clken	cfeb0 clken	step run	step cfeb	step rpc	step dmb	step alct

Bits	Dir	Signal	Default	Description
[00]	RW	step_alct	0	Step ALCT clock
[01]	RW	step_dmb	0	Step DMB clock
[02]	RW	step_rpc	0	Step RPC clock
[03]	RW	step_cfeb	0	Step CFEB clock
[04]	RW	step_run	0	0=run mode, 1=step clocks
[05]	RW	cfeb_clock_en0	1	1=enable CFEB0 clock
[06]	RW	cfeb_clock_en1	1	1=enable CFEB1 clock
[07]	RW	cfeb_clock_en2	1	1=enable CFEB2 clock
[08]	RW	cfeb_clock_en3	1	1=enable CFEB3 clock
[09]	RW	cfeb_clock_en4	1	1=enable CFEB4 clock
[10]	RW	alct_clock_en	1	1=enable ALCT clock
[11]	RW	/alct_hard_reset_en	1	1=disable ALCT hard reset
[12]	RW	/tmb_hard_reset_en	1	1=disable TMB hard reset
[15:13]	RW	--	0	Unassigned

Adr 22 ADR_LED Front Panel + On-Board LED Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
led bd7	led bd6	led bd5	led bd4	led bd3	led bd2	led bd1	led bd0	VME	NL1A	NMAT	INVP	L1A	CLCT	ALCT	LCT

Bits	Dir	Signal	Color	Description
[00]	RW	led_fp_lct	Blue	LCT TMB matched ALCT+CLCT
[01]	RW	led_fp_lct	Green	ALCT found a muon
[02]	RW	led_fp_clct	Green	CLCT found a muon
[03]	RW	led_fp_l1a	Green	L1A level 1 accept
[04]	RW	led_fp_invp	Amber	INVP invalid pattern after CSC drift
[05]	RW	led_fp_nmat	Amber	NMAT no match after ALCT or CLCT triggered
[06]	RW	led_fp_nl1a	Red	NL1A no L1A after trigger
[07]	RW	led_fp_vme	Green	VME power-up = on, off=vme access flash
[08]	RW	led_bd0	Blue	Buffer busy[0]
[09]	RW	led_bd1	Green	Buffer busy[1]
[10]	RW	led_bd2	Green	Buffer busy[2]
[11]	RW	led_bd3	Green	Buffer busy[3]
[12]	RW	led_bd4	Green	Buffer busy[4]
[13]	RW	led_bd5	Green	Buffer busy[5]
[14]	RW	led_bd6	Green	Buffer busy[6]
[15]	RW	led_bd7	Red	Buffer busy[7]

Adr 24 ADR_ADC ADC + Power Comparator Register

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	smb data	smb clk	ADC /cs	ADC din	ADC sclock	ADC dout	/tcrit	V1.5	V1.8	V3.3	V5.0

Bits	Dir	Signal	Typical	Description
[00]	R	vstat_5p0v	1	1 = 5.0V power supply OK
[01]	R	vstat_3p3v	1	1 = 3.3V power supply OK
[02]	R	vstat_1p8v	1	1 = 1.8V power supply OK
[03]	R	vstat_1p5v	1	1 = 1.5V power supply OK
[04]	R	/t_crit	1	1 = FPGA and Board Temperature OK
[05]	R	adc_dout	0	Voltage monitor ADC serial data receive
[06]	RW	adc_sclock	0	Voltage monitor ADC serial clock
[07]	RW	adc_din	0	Voltage monitor ADC serial data transmit
[08]	RW	/adc_cs	1	Voltage monitor ADC chip select
[09]	RW	smb_clk	0	Temperature monitor ADC serial clock
[10]	RW	smb_data	1	Temperature monitor ADC serial data, open drain
[15:11]	RW	--	0	Unassigned

Adr 26 ADR_DSN Digital Serial Numbers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	RAT DSN Data	RAT DSN Busy	RAT DSN Init	RAT DSN Write	RAT DSN Start	Mez DSN Data	Mez DSN Busy	Mez DSN Init	Mez DSN Write	Mez DSN Start	TMB DSN Data	TMB DSN Busy	TMB DSN Init	TMB DSN Write	TMB DSN Start

Bits	Dir	Signal	Default	Description
[00]	RW	tmb_sn_start	0	TMB Digital serial SM start
[01]	RW	tmb_sn_write	0	TMB Digital serial write pulse
[02]	RW	tmb_sn_init	0	TMB Digital serial Init pulse
[03]	R	tmb_sn_busy	-	TMB State DSN State Machine busy
[04]	R	tmb_sn_data	-	TMB State DSN read data
[05]	RW	mez_sn_start	0	Mez Digital Serial State Machine start
[06]	RW	mez_sn_write	0	Mez Digital Serial Write pulse
[07]	RW	mez_sn_init	0	Mez Digital Serial Init pulse
[08]	R	mez_sn_busy	-	Mez State DSN State Machine busy
[09]	R	mez_sn_data	-	Mez State DSN read data
[10]	RW	rat_sn_start	0	RAT Digital Serial State Machine start
[11]	RW	rat_sn_write	0	RAT Digital Serial Write pulse
[12]	RW	rat_sn_init	0	RAT Digital Serial Init pulse
[13]	R	rat_sn_busy	-	RAT State DSN State Machine busy
[14]	R	rat_sn_data	-	RAT State DSN read data
[15]	RW	-	0	Unassigned

Adr 28 ADR_MOD_CFG TMB Module Configuration

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mez done	ddd auto	power up	global reset enable	cfeb6 exists	cfeb5 exists	cfeb4 exists	cfeb3 exists	cfeb2 exists	cfeb1 exists	cfeb0 exists	bdled cylon	bdled vme	fpled flash	fpled cylon	fpled vme

Bits	Dir	Signal	Default	Description
[00]	RW	led_fp_src_vme	0	1=Front Panel LEDs sourced from VME register
[01]	RW	led_fp_cylon	0	1=FP LED Cylon mode, cool
[02]	RW	led_flash_on_stop	1	1=Flash Front Panel LEDs on TTT stop_trigger
[03]	RW	led_bd_src_vme	0	1=On-Board LEDs sourced from VME register
[04]	RW	led_bd_cylon	0	1=BD LED Cylon mode, cool
[11:5]	R	cfeb_exists[6:0]	7F	CFEB(n) instantiated in this firmware version
[12]	RW	global_reset_en	1	1=fire global reset if main DLL loses lock
[13]	R	power_up		Power-up FF
[14]	R	ddd_autostart		1=3D3444 auto-start enabled, copy of Adr14[05]
[15]	R	mez_done		1=Mezzanine FPGA loaded from PROM

Adr 2A ADR_CCB_CFG CCB Configuration

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
adb pulse async	adb pulse sync	alct hard reset	tmb hard reset	tmb resout 2	tmb resout 1	tmb resout 0	tmb res1	tmb res0	l1a vme	clct status en	alct status en	ccb status oe	int l1aen	disabl tx	ignore rx

Bits	Dir	Signal	Default	Description
[00]	RW	ccb_ignore_rx	0	1=Ignore Received CCB backplane inputs
[01]	RW	ccb_disable_tx	0	1=Disble transmitted CCB backplane outputs
[02]	RW	ccb_int_l1a_en	0	1=Enable internal L1A emulator
[03]	RW	ccb_status_oe_vme	0	1=Enable ALCT+CLCT status to CCB front panel
[04]	RW	alct_status_en	0	1=Enable ALCT status GTL outputs (req [03]=1)
[05]	RW	clct_status_en	0	1=Enable CLCT status GTL outputs (req [03]=1)
[06]	RW	l1accept_vme	0	1=fire ccb_l1accept oneshot
[08:07]	R	tmb_reserved[1:0]		Future use
[11:09]	R	tmb_reserved_out[2:0]		Future use
[12]	R	tmb_hard_reset		Reload TMB FPGA
[13]	R	alct_hard_reset		Reload ALCT FPGA
[14]	R	alct_adb_pulse_sync		ALCT synchronous test pulse from CCB
[15]	R	alct_adb_pulse_async		ALCT asynchronous test pulse from CCB
[12]	W	vme_evcntres	0	Event counter reset || ccb_evcntres
[13]	W	vme_bcntres	0	Bunch crossing reset || ccb_bcntres
[14]	W	vme_bx0	0	Bx0 signal || ccb_bx0
[15]	W	vme_bx0_emu_en	0	Bx0 Emulator enable

Adr 2C ADR_CCB_TRIG CCB Trigger Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l1a delay vme7	l1a delay vme6	l1a delay vme5	l1a delay vme4	l1a delay vme3	l1a delay vme2	l1a delay vme1	l1a delay vme0	Ignore start/ stop	ccb exttrig bypas	ext trig both	clct ext trg vme	alct ext trg vme	seq trig l1aen	clct ext trg l1aen	alct ext trg l1aen

Bits	Dir	Signal	Default	Description
[00]	RW	alct_ext_trig_l1aen	0	1=Request ccb l1a on alct ext_trig
[01]	RW	clct_ext_trig_l1aen	0	1=Request ccb l1a on clct ext_trig
[02]	RW	seq_trig_l1aen	1	1=Request ccb l1a on sequencer trigger
[03]	RW	alct_ext_trig_vme	0	1=Fire alct_ext_trig oneshot
[04]	RW	clct_ext_trig_vme	0	1=Fire clct_ext_trig oneshot
[05]	RW	ext_trig_both	0	1=clct_ext_trig fires alct + alct fires clct_trig, DC
[06]	RW	ccb_allow_extbypass	0	1=Allow clct_exttrig_ccb when ccb_ignore_rx=1
[07]	RW	ccb_ignore_startstop	0	1=Ignore ttc_trig_start, ttc_trig_stop
[15:08]	RW	l1a_delay_vme	72_{16}	Internal L1A delay (not same as sequencer L1A)

Adr 2E ADR_CCB_STAT0 CCB Status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ccb bx0	ccb bcntrs	ccb res4	ccb res3	ccb res2	ccb qpll locked	ccb ttcrx ready	ccb clock en	ccb cmd7	ccb cmd6	ccb cmd5	ccb cmd4	ccb cmd3	ccb cmd2	ccb cmd1	ccb cmd0

Bits	Dir	Signal	Default	Description
[07:00]	R	ccb_cmd[7:0]		CCB Command word from TTC
[08]	R	ccb_clock40_enable	1	1=TMB 40MHz clock from CCB enabled
[09]	R	ccb_reserved[0]	1	ccb_ttcrx_ready TTC ready signal from CCB
[10]	R	ccb_reserved[1]	1	ccb_qpll_locked PLL locked signal from CCB
[13:09]	R	ccb_reserved[4:2]		Future use
[14]	R	ccb_bcntres		Bunch counter reset from CCB (backplane)
[15]	R	ccb_bx0		Bunch crossing 0 from CCB (backplane)

Adr 30 ADR_ALCT_CFG ALCT Configuration

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	cfeb muon ic	alct muon ic	alct clk vme	alct clk ccb	alct seq cmd3	alct seq cmd2	alct seq cmd1	alct seq cmd0	assert alct ext inj	assert alct ext trg	alct ext inj en	alct ext trg en

Bits	Dir	Signal	Default	Description
[00]	RW	cfg_alct_ext_trig_en	1	1=Enable alct_ext_trig from CCB
[01]	RW	cfg_alct_ext_inject_en	0	1=Enable alct_ext_inject from CCB
[02]	RW	cfg_alct_ext_trig	0	1=Assert alct_ext_trig
[03]	RW	cfg_alct_ext_inject	0	1=Assert alct_ext_inject
[07:04]	RW	alct_seq_cmd[3:0]	0	ALCT Sequencer command
[08]	RW	alct_clock_en_use_ccb		1=alct_clock_en_vme = ccb_clock40_enable
[09]	RW	alct_clock_en_use_vme		sets alct_clock_en cable signal if [8]=0
[10]	R	alct_muonic	1	ALCT board has independent time-of-flight delay
[11]	R	cfeb_muonic	0	CFEBs have independent time-of-flight delay
[15:12]	RW	--	0	Unassigned

Adr 32 ADR_ALCT_INJ ALCT Injector Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	inject delay 4	inject delay 3	inject delay 2	inject delay 1	inject delay 0	l1a inj ram	alct inj ram	link inject w clct	start inject	clear alct

Bits	Dir	Signal	Default	Description
[00]	RW	alct_clear	0	1=Blank ALCT received data
[01]	RW	alct_inject_mux	0	1=Start ALCT injector State Machine
[02]	RW	alct_sync_clct	0	1=Link ALCT injector with CLCT inject command
[03]	RW	alct_inj_ram_en	0	1=Link ALCT injector to CFEB injector RAM
[04]	RW	l1a_inj_ram_en	0	1=Link L1A injector to CFEB injector RAM
[09:05]	RW	alct_inj_delay[4:0]	13	Injector delay
[15:10]	RW	--	0	Unassigned

Adr 34 ADR_ALCT0_INJ ALCT0 1st Muon To Inject

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1st bxn 1	1st bxn 0	1st key 6	1st key 5	1st key 4	1st key 3	1st key 2	1st key 1	1st key 0	1st amu	1st qualty 1	1st qualty 0	1st vpf

Bits	Dir	Signal	Default	Description
[00]	RW	alct_first_valid	1	Valid pattern flag
[02:01]	RW	alct_first_quality[1:0]	3	Pattern quality
[03]	RW	alct_first_amu	0	Accelerator muon flag
[10:04]	RW	alct_first_key[6:0]	7	Injected ALCT0 key wire-group
[12:11]	RW	alct_first_bxn[1:0]	1	Injected ALCT0 bunch crossing number
[15:13]	RW	--	0	Unassigned

Adr 36 ADR_ALCT1_INJ ALCT1 2nd Muon To Inject

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	2nd bxn 1	2nd bxn 0	2nd key 6	2nd key 5	2nd key 4	2nd key 3	2nd key 2	2nd key 1	2nd key 0	2nd amu	2nd qualty 1	2nd qualty 0	2nd vpf

Bits	Dir	Signal	Default	Description
[00]	RW	alct_second_valid	1	Valid pattern flag
[02:01]	RW	alct_second_quality[1:0]	2	Pattern quality
[03]	RW	alct_second_amu	0	Accelerator muon flag
[10:04]	RW	alct_second_key[6:0]	61_{10}	Injected ALCT1 key wire-group
[12:11]	RW	alct_second_bxn[1:0]	1	Injected ALCT1 bunch crossing number
[15:13]	RW	--	0	Unassigned

Adr 38 ADR_ALCT_STAT ALCT Sequencer Control/Status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
alct txdint delay 3	alct txdint delay 2	alct txdint delay 1	alct txdint delay 0	0	0	0	0	0	0	0	alct sync eccerr 1	alct sync eccerr 0	alct ecc blank	alct ecc en	alct cfg done

Bits	Dir	Signal	Default	Description
[00]	R	alct_cfg_done	1	ALCT FPGA loaded from PROM
[01]	RW	alct_ecc_en	1	ALCT ECC trigger data correction enable
[02]	RW	alct_ecc_err_blank	1	Blank alcts with uncorrected ecc errors
[04:03]	R	alct_sync_ecc_err[1:0]	0	ALCT sync-mode ECC error code
[11:05]	RW	--	0	Unassigned
[15:12]	RW	alct_txd_int_delay[3:0]	0	Delay data transmitted to ALCT by integer bx

Adr 3A ADR_ALCT0_RCD ALCT 1st Muon Received by TMB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1st bxn 1	1st bxn 0	1st key 6	1st key 5	1st key 4	1st key 3	1st key 2	1st key 1	1st key 0	1st amu	1st qualty 1	1st qualty 0	1st vpf

Bits	Dir	Signal	Typical	Description
[00]	R	alct_first_valid	1	Valid pattern flag
[02:01]	R	alct_first_quality[1:0]	0-3	Pattern quality
[03]	R	alct_first_amu	0	Accelerator muon flag
[10:04]	R	alct_first_key[6:0]	0-111	ALCT0 key wire-group
[12:11]	R	alct_first_bxn[1:0]	0-3	ALCT0 bunch crossing number
[15:13]	R	--	0	Unassigned

Adr 3C ADR_ALCT1_RCD ALCT 2nd Muon Received by TMB

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	2nd bxn 1	2nd bxn 0	2nd key 6	2nd key 5	2nd key 4	2nd key 3	2nd key 2	2nd key 1	2nd key 0	2nd amu	2nd qualty 1	2nd qualty 0	2nd vpf

Bits	Dir	Signal	Typical	Description
[00]	R	alct_second_valid	1	Valid pattern flag
[02:01]	R	alct_second_quality[1:0]	0-3	Pattern quality
[03]	R	alct_second_amu	0	Accelerator muon flag
[10:04]	R	alct_second_key[6:0]	0-111	ALCT1 key wire-group
[12:11]	R	alct_second_bxn[1:0]	0-3	ALCT1 bunch crossing number
[15:13]	R	--	0	Unassigned

Adr 3E ADR_ALCT_FIFO ALCT FIFO RAM Status

(Split with Adr A2 ADR_ALCT_FIFO1 and A4 ADR_ALCT_FIFO2)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	alct data17	alct data16	alct wdcnt10	alct wdcnt9	alct wdcnt8	alct wdcnt7	alct wdcnt6	alct wdcnt5	alct wdcnt4	alct wdcnt3	alct wdcnt2	alct wdcnt1	alct wdcnt0	alct RAM done	alct RAM busy

Bits	Dir	Signal		Description
[00]	R	alct_raw_busy		ALCT raw hits FIFO busy writing ALCT data
[01]	R	alct_raw_done		ALCT raw hits ready for VME readout
[12:02]	R	alct_raw_wdcnt[10:0]		ALCT raw hits word count stored in RAM
[14:13]	R	alct_raw_rdata[17:16]		ALCT raw hits data MSBs
[15]	R	--	0	Unassigned

Adr 40 ADR_DMB_MON DMB Monitored Signals

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	2nd bxn 1	2nd bxn 0	2nd key 6	2nd key 5	2nd key 4	2nd key 3	2nd key 2	2nd key 1	2nd key 0	2nd amu	2nd qualty 1	2nd qualty 0	2nd vpf

Bits	Dir	Signal	Typical	Description
[02:00]	R	dmb_cfeb_calibrate[2:0]	0	DMB calibration
[03]	R	dmb_l1a_release	0	DMB test
[08:04]	R	dmb_reserved_out[4:0]	0	DMB future use
[11:09]	R	dmb_reserved_in[2:0]	0	DMB future use
[15:12]	R	dmb_rx_ff[3:0]	0	DMB received

Adr 42 ADR_CFEB_INJ CFEB Injector Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
inj start	inj mask 4	inj mask 3	inj mask 2	inj mask 1	inj mask 0	inj febsel 4	inj febsel 3	inj febsel 2	inj febsel 1	inj febsel 0	mask all cfeb4	mask all cfeb3	mask all cfeb2	mask all cfeb1	mask all cfeb0

Bits	Dir	Signal	Default	Description
[04:00]	RW	mask_all[4:0]	11111_2	1=Enable, 0=Turn off CFEBn inputs See Adr68 p40
[09:05]	RW	inj_febsel[4:0]	0	1=Select CFEBn for RAM read/write
[14:10]	RW	injector_mask[4:0]	11111_2	Enable CFEBn for injector trigger
[15]	RW	inj_trig_vme	0	Start pattern injector

Adr 44 ADR_CFEB_INJ_ADR CFEB Injector RAM Address

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
inj adr 9	inj adr 8	inj adr 7	inj adr 6	inj adr 5	inj adr 4	inj adr 3	inj adr 2	inj adr 1	inj adr 0	inj ren 2	inj ren 1	inj ren 0	inj wen 2	inj wen 1	inj wen 0

Bits	Dir	Signal	Default	Description
[02:00]	RW	inj_wen[2:0]	0	1=Write enable injector RAMn (Ly01,23,45)
[05:03]	RW	inj_ren[2:0]	0	1=Read enable Injector RAMn
[15:06]	RW	inj_rwadr[9:0]	0	Injector RAM read/write address

Adr 46 ADR_CFEB_INJ_WDATA CFEB Injector Write Data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
inj wdata 15	inj wdata 14	inj wdata 13	inj wdata 12	inj wdata 11	inj wdata 10	inj wdata 9	inj wdata 8	inj wdata 7	inj wdata 6	inj wdata 5	inj wdata 4	inj wdata 3	inj wdata 2	inj wdata 1	inj wdata 0

Bits	Dir	Signal	Default	Description
[07:00]	RW	inj_wdata[7:0]	0	Triad bit for addressed Tbin Ly0 (or 2,4)
[15:08]	RW	inj_wdata[15:8]	0	Triad bit for addressed Tbin Ly1 (or 3,5)

Adr 48 ADR_CFEB_INJ_RDATA CFEB Injector Read Data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
inj rdata 15	inj rdata 14	inj rdata 13	inj rdata 12	inj rdata 11	inj rdata 10	inj rdata 9	inj rdata 8	inj rdata 7	inj rdata 6	inj rdata 5	inj rdata 4	inj rdata 3	inj rdata 2	inj rdata 1	inj rdata 0

Bits	Dir	Signal	Default	Description
[07:00]	R	inj_rdata[7:0]	0	Triad bit for addressed Tbin Ly0 (or 2,4)
[15:08]	R	inj_rdata[15:8]	0	Triad bit for addressed Tbin Ly1 (or 3,5)

Adr 4A ADR_HCM001 CFEB0 Ly0,Ly1 Hot Channel Mask

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ly1 distrip 7	ly1 distrip 6	ly1 distrip 5	ly1 distrip 4	ly1 distrip 3	ly1 distrip 2	ly1 distrip 1	ly1 distrip 0	ly0 distrip 7	ly0 distrip 6	ly0 distrip 5	ly0 distrip 4	ly0 distrip 3	ly0 distrip 2	ly0 distrip 1	ly0 distrip 0

Bits	Dir	Signal	Default	Description
[07:00]	RW	cfeb0_ly0_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 0
[15:08]	RW	cfeb0_ly1_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 1

Adr 4C ADR_HCM023 CFEB0 Ly2,Ly3 Hot Channel Mask

[07:00]	RW	cfeb0_ly2_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 2
[15:08]	RW	cfeb0_ly3_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 3

Adr 4E ADR_HCM045 CFEB0 Ly4,Ly5 Hot Channel Mask

[07:00]	RW	cfeb0_ly4_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 4
[15:08]	RW	cfeb0_ly5_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 5

Adr 50 ADR_HCM101 CFEB1 Ly0,Ly1 Hot Channel Mask

[07:00]	RW	cfeb1_ly0_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 0
[15:08]	RW	cfeb1_ly1_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 1

Adr 52 ADR_HCM123 CFEB1 Ly2,Ly3 Hot Channel Mask

[07:00]	RW	cfeb1_ly2_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 2
[15:08]	RW	cfeb1_ly3_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 3

Adr 54 ADR_HCM145 CFEB1 Ly4,Ly5 Hot Channel Mask

[07:00]	RW	cfeb1_ly4_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 4
[15:08]	RW	cfeb1_ly5_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 5

Adr 56 ADR_HCM201 CFEB2 Ly0,Ly1 Hot Channel Mask

[07:00]	RW	cfeb2_ly0_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 0
[15:08]	RW	cfeb2_ly1_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 1

Adr 58 ADR_HCM223 CFEB2 Ly2,Ly3 Hot Channel Mask

[07:00]	RW	cfeb2_ly2_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 2
[15:08]	RW	cfeb2_ly3_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 3

Adr 5A ADR_HCM245 CFEB2 Ly4,Ly5 Hot Channel Mask

[07:00]	RW	cfeb2_ly4_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 4
[15:08]	RW	cfeb2_ly5_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 5

Adr 5C ADR_HCM301 CFEB3 Ly0,Ly1 Hot Channel Mask

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ly1 distrip 7	ly1 distrip 6	ly1 distrip 5	ly1 distrip 4	ly1 distrip 3	ly1 distrip 2	ly1 distrip 1	ly1 distrip 0	ly0 distrip 7	ly0 distrip 6	ly0 distrip 5	ly0 distrip 4	ly0 distrip 3	ly0 distrip 2	ly0 distrip 1	ly0 distrip 0

Bits	Dir	Signal	Default	Description
[07:00]	RW	cfeb3_ly0_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 0
[15:08]	RW	cfeb3_ly1_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 1

Adr 5E ADR_HCM323 CFEB3 Ly2,Ly3 Hot Channel Mask

[07:00]	RW	cfeb3_ly2_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 2
[15:08]	RW	cfeb3_ly3_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 3

Adr 60 ADR_HCM345 CFEB3 Ly4,Ly5 Hot Channel Mask

[07:00]	RW	cfeb3_ly4_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 4
[15:08]	RW	cfeb3_ly5_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 5

Adr 62 ADR_HCM401 CFEB4 Ly0,Ly1 Hot Channel Mask

[07:00]	RW	cfeb4_ly0_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 0
[15:08]	RW	cfeb4_ly1_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 1

Adr 64 ADR_HCM423 CFEB4 Ly2,Ly3 Hot Channel Mask

[07:00]	RW	cfeb4_ly2_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 2
[15:08]	RW	cfeb4_ly3_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 3

Adr 66 ADR_HCM445 CFEB4 Ly4,Ly5 Hot Channel Mask

[07:00]	RW	cfeb4_ly4_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 4
[15:08]	RW	cfeb4_ly5_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 5

Adr 68 ADR_SEQ_TRIG_EN Sequencer Trigger Source Enables **

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeben source	cfeben 4	cfeben 3	cfeben 2	cfeben 1	cfeben 0	all cfebs active	ext trig inject	vme trig	alct ext trig en	clct ext trig en	adb ext trig en	adb ext trig en	alct*clct match trig en	alct pat trig en	clct pat trig en

Bits	Dir	Signal	Default	Description
[00]	RW	clct_pat_trig_en	1	1=Allow CLCT pattern triggers (CLCT Active FEB)
[01]	RW	alct_pat_trig_en	0	1=Allow ALCT pattern triggers (ALCT Active FEB)
[02]	RW	alct_match_trig_en	0	1=ALCT*CLCT pattern triggers
[03]	RW	adb_ext_trig_en	0	1=Allow ADB external triggers from CCB
[04]	RW	dmb_ext_trig_en	0	1=Allow DMB external triggers
[05]	RW	clct_ext_trig_en	0	1=Allow CLCT external triggers (scintillator) from CCB
[06]	RW	alct_ext_trig_en	0	1=Allow ALCT external triggers from CCB
[07]	RW	vme_ext_trig	0	1=Initiate Sequencer trigger (write 0 to recover)
[08]	RW	ext_trig_inject	0	1=Change clct_ext_trig to fire pattern injector
[09]	RW	all_cfebs_active	0	1=Make all CFEBs active when triggered
[14:10]	RW*	cfeb_en	11111_2	1=Enable CFEB[n] to trigger and send active_feb_flag
[15]	RW	cfeb_en_source	1	1=cfeb_en set by mask_all[4:0] in Adr 42, 0=set by 68

* normally, cfeb_en is copied from mask_all in Adr42 so that masked-off cfebs do not trigger TMB or send active feb to DMB. That prevents the CFEB pattern injector from triggering, so setting_cfeb_en_source=0, allows cfeb_en to be written independently via Adr68[14:10].

** See adr F0 p67 for layer-trigger mode

Adr 6A ADR_SEQ_TRIG_DLY0 Sequencer Trigger Source Delays

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
adb delay 3	adb delay 2	adb delay 1	adb delay 0	alct aff delay 3	alct aff delay 2	alct aff delay 1	alct aff delay 0	alct pretrig delay	alct pretrig delay 2	alct pretrig delay 1	alct pretrig delay 0	alct width 3	alct width 2	alct width 1	alct width 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	alct_trig_width[3:0]	3	ALCTCLCT Pre-trigger window width
[07:04]	RW	alct_pre_trig_dly	0 (2)	ALCT Pre-trigger delay for ALCT*CLCT
[11:08]	RW	alct_pat_trig_dly[3:0]	0	Delay alct_pat_trig (active feb flag from ALCT)
[15:12]	RW	adb_ext_trig_dly[3:0]	1	Delay adb_ext_trig from CCB

Adr 6C **ADR_SEQ_TRIG_DLY1** **Sequencer Trigger Source Delays**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	alct ext delay 3	alct ext delay 2	alct ext delay 1	alct ext delay 0	clct ext delay 3	clct ext delay 2	clct ext delay 1	clct ext delay 0	dmb ext delay 3	dmb ext delay 2	dmb ext delay 1	dmb ext delay 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	dmb_ext_trig_dly[3:0]	1	Delay dmb_ext_trig from DMB
[07:04]	RW	clct_ext_trig_dly[3:0]	7	Delay clct_ext_trig (scintillator) from CCB
[11:08]	RW	alct_ext_trig_dly[3:0]	7	Delay alct_ext_trig from CCB
[15:12]	RW	-	0	Unused

Adr 6E **ADR_SEQ_ID** **Sequencer Board + CSC Ids**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	run id 3	run id 2	run id 1	run id 0	csd id 3	csd id 2	csd id 1	csd id 0	board id 4	board id 3	board id 2	board id 1	board id 0

Bits	Dir	Signal	Default	Description
[04:00]	RW	board_id[4:0]	21	Board ID = VME Slot Geographic Adr
[08:05]	RW	csc_id[3:0]	5	CSC Chamber ID number
[12:09]	RW	run_id[3:0]	0	Run ID number
[15:13]	RW	--	0	Unassigned

Adr 70 **ADR_SEQ_CLCT** **Sequencer CLCT Configuration**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
pretrig halt	drft delay 0	drft delay 0	hit thresh post 2	hit thresh post 1	hit thresh post 0	dmb thresh 2	dmb thresh 1	dmb thresh 0	hs thresh 2	hs thresh 1	hs thresh 0	triad persist 3	triad persist 2	triad persist 1	triad persist 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	triad_persist	6	Triad One-Shot Persistence (6=150ns)
[06:04]	RW	hit_thresh_pretrig[2:0]	4	Pattern hits pre-trigger threshold
[09:07]	RW	dmb_thresh_pretrig[2:0]	4	Minimum pattern hits 0-6 for DMB active-febs
[12:10]	RW	hit_thresh_postdrift[2:0]	4	Minimum pattern hits allowed after drift
[14:13]	RW	drift_delay[1:0]	2	CSC Drift delay, number 25ns clock periods
[15]	RW	pretrig_halt	0	Pretrigger and halt until unhalt arrives

Adr 72 ADR_SEQ_FIFO Sequencer FIFO Configuration

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bcb read enable	0	fifo no raw hits	fifo pretrig 4	fifo pretrig 3	fifo pretrig 2	fifo pretrig 1	fifo pretrig 0	fifo tbins 4	fifo tbins 3	fifo tbins 2	fifo tbins 1	fifo tbins 0	fifo mode 2	fifo mode 1	fifo mode 0

Bits	Dir	Signal	Default	Description
[02:00]	RW	fifo_mode[2:0]	1	FIFO Mode: 0=no CFEB raw hits full header 1=all CFEB raw hits full header 2=local CFEB raw hits full header 3=no CFEB raw hits short header 4=no CFEB raw hits no header
[07:03]	RW	fifo_tbins[4:0]	7	Number FIFO time bins to read out
[12:08]	RW	fifo_pretrig[4:0]	2	Number FIFO time bins before pretrigger
[13]	RW	fifo_no_raw_hits	0	1=do not wait to store raw hits [a no_daq mode]
[14]	RW	--	0	Unassigned
[15]	RW	bcb_read_enable	0	1=enable cfeb blocked distrip bits in dmb readout

Adr 74 ADR_SEQ_L1A Sequencer L1A Configuration

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l1a intern dly 2	l1a intern dly 1	l1a intern dly 0	l1a intern	l1a windo 3	l1a windo 2	l1a windo 1	l1a windo 0	l1a delay 7	l1a delay 6	l1a delay 5	l1a delay 4	l1a delay 3	l1a delay 2	l1a delay 1	l1a delay 0

Bits	Dir	Signal	Default	Description
[07:00]	RW	l1a_delay[7:0]	128_{10}	Level1 Accept delay from pretrig status output
[11:08]	RW	l1a_window[3:0]	3	Level1 Accept window width after delay
[12]	RW	l1a_internal	0	Generate internal Level 1, overrides external
[15:13]	RW	l1a_internal_dly[2:0]	0	Window position for internal L1A

Adr 76 ADR_SEQ_OFFSET0 Sequencer Counter Offsets [see Adr10A p70 for L1A bxn offset]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bxn offset 11	bxn offset 10	bxn offset 9	bxn offset 8	bxn offset 7	bxn offset 6	bxn offset 5	bxn offset 4	bxn offset 3	bxn offset 2	bxn offset 1	bxn offset 0	l1a offset 3	l1a offset 2	l1a offset 1	l1a offset 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	l1a_offset[3:0]	0	L1A counter preset value
[15:04]	RW	bxn_offset_pretrig[11:0]	0	BXN offset at reset for pretrigger bxn

Adr 78 **ADR_SEQ_CLCT0** **Sequencer Latched CLCT0 (LSBs)**

(Split with Adr B0 ADR_SEQCLCTM)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
hs key 7	hs key 6	hs key 5	hs key 4	hs^{t} key 3	hs key 2	hs key 1	hs key 0	hs pid 3	hs pid 2	hs pid 1	hs pid 0	hs hits 2	hs hits 1	hs hits 0	vpf

Bits	Dir	Signal	Typical	Description
[00]	R	clct0[0] clct_1st_valid	1	Valid pattern flag
[03:01]	R	clct0[3:1] hs_hit_1st[2:0]	4-6	Hits on pattern: 0 to 6
[07:04]	R	clct0[7:4] hs_pid_1st[3:0]	0-10	Pattern shape 0 to 10
[15:08]	R	clct0[15:8] hs_key_1st[7:0]	0-159$_{10}$	Key ½-strip

Adr 7A **ADR_SEQ_CLCT1** **Sequencer Latched CLCT1 (LSBs)**

(Split with Adr B0 ADR_SEQCLCTM)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
hs key 7	hs key 6	hs key 5	hs key 4	hs^{t} key 3	hs key 2	hs key 1	hs key 0	hs pid 3	hs pid 2	hs pid 1	hs pid 0	hs hits 2	hs hits 1	hs hits 0	vpf

Bits	Dir	Signal	Typical	Description
[00]	R	clct1[0] clct_2nd_valid	1	Valid pattern flag
[03:01]	R	clct1[3:1] hs_hit_2nd[2:0]	4-6	Hits on pattern: 0 to 6
[07:04]	R	clct1[7:4] hs_pid_2nd[3:0]	0-10	Pattern shape 0 to 10
[15:08]	R	clct1[15:8] hs_key_2nd[7:0]	0-159$_{10}$	Key ½-strip

Adr 7C **ADR_SEQ_TRIG_SRC** **Sequencer Trigger Source Read-back**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	me1b pretrig	me1a pre trig	layer trig	vme trig	alct ext trig en	clct ext trig en	adb ext trig en	adb ext trig en	alct*clct pat trig en	alct pat trig en	clct pat trig en

Bits	Dir	Signal	Typical	Description
[00]	R	clct_pat_trig_en	1	CLCT pattern triggered sequencer
[01]	R	alct_pat_trig_en	0	ALCT pattern triggered sequencer
[02]	R	alct_match_trig_en	0	ALCT*CLCT pattern triggered sequencer
[03]	R	adb_ext_trig_en	0	ADB external triggered sequencer
[04]	R	dmb_ext_trig_en	0	DMB external triggered sequencer
[05]	R	clct_ext_trig_en	0	CLCT (CCB scintillator) external triggered sequencer
[06]	R	alct_ext_trig_en	0	ALCT (CCB) external triggered sequencer
[07]	R	vme_ext_trig	0	VME triggered sequencer
[08]	R	layer_trig	0	Layer trigger
[09]	R	me1a_only_pretrig	0	CLCT pattern trigger was on ME1A only
[10]	R	me1b_only_pretrig	0	CLCT pattern trigger was on ME1B only
[13:11]	R	--	0	Unassigned

Adr 7E ADR_DMB_RAM_ADR Sequencer RAM Address

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dmb wdata 17	dmb wdata 16	dmb reset	dmb wr	dmb adr 11	dmb adr 10	dmb adr 9	dmb adr 8	dmb adr 7	dmb adr 6	dmb adr 5	dmb adr 4	dmb adr 3	dmb adr 2	dmb adr 1	dmb adr 0

Bits	Dir	Signal	Default	Description
[11:00]	RW	dmb_adr[11:0]	0	Raw hits RAM VME read/write address
[12]	RW	dmb_wr	0	Raw hits RAM VME write enable
[13]	RW	dmb_reset	0	Raw hits RAM VME address reset
[15:14]	RW	dmb_wdata[17:16]	0	Raw hits RAM VME write data MSBs

Adr 80 ADR_DMB_RAM_WDATA Sequencer RAM Write Data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dmb wdata 15	dmb wdata 14	dmb wdata 13	dmb wdata 12	dmb wdata 11	dmb wdata 10	dmb wdata 9	dmb wdata 8	dmb wdata 7	dmb wdata 6	dmb wdata 5	dmb wdata 4	dmb wdata 3	dmb wdata 2	dmb wdata 1	dmb wdata 16

Bits	Dir	Signal	Default	Description
[15:00]	RW	dmb_wdata[15:0]	0	Raw hits RAM VME write data (msb in adr 76)

Adr 82 ADR_DMB_RAM_WDCNT Sequencer RAM Word Count

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	dmb busy	dmb rdata 17	dmb wdata 16	dmb wdcnt 11	dmb wdcnt 10	dmb wdcnt 9	dmb wdcnt 8	dmb wdcnt 7	dmb wdcnt 6	dmb wdcnt 5	dmb wdcnt 4	dmb wdcnt 3	dmb wdcnt 2	dmb wdcnt 1	dmb wdcnt 0

Bits	Dir	Signal	Default	Description
[11:00]	R	dmb_wdcnt[11: 0]	0	Raw hits RAM VME word count
[13:12]	R	dmb_rdata[17:16];	0	Raw hits RAM VME read data MSBs
[14]	R	dmb_busy	0	Raw hits RAM VME
[15]	R	--	0	Unassigned

Adr 84 ADR_DMB_RAM_RDATA Sequencer RAM Read Data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
dmb rdata 15	dmb rdata 14	dmb rdata 13	dmb rdata 12	dmb rdata 11	dmb rdata 10	dmb rdata 9	dmb rdata 8	dmb rdata 7	dmb rdata 6	dmb rdata 5	dmb rdata 4	dmb rdata 3	dmb rdata 2	dmb rdata 1	dmb rdata 16

Bits	Dir	Signal	Default	Description
[15:00]	R	dmb_rdata[15:0]	0	Raw hits RAM VME read data (msb in adr 7A)

Adr 86 ADR_TMB_TRIG TMB Trigger Configuration / MPC Accept

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mpc oe	mpc idle blank	mpc sel_ttc bx0	mpc reserved 1	mpc reserved 0	mpc accept 1	mpc accept 0	mpc delay 3	mpc delay 2	mpc delay 1	mpc delay 0	allow clct+alct match	allow clct only	allow alct only	sync err en 1	sync err en 0

Bits	Dir	Signal	Default	Description
[01:00]	RW	tmb_sync_err_en[1:0]	11_2	Allow sync_err to MPC for either muon
[02]	RW	tmb_allow_alct	0	Allow ALCT-only L1A (not used in current version)
[03]	RW	tmb_allow_clct	1	Allow CLCT0-only L1A
[04]	RW	tmb_allow_match	1	Allow ALCT+CLCT match pre-trigger
[08:05]	RW	mpc_rx_delay[3:0]	7	MPC accept response delay
[10:09]	R	mpc_accept[1:0]	-	MPC accept latched after delay
[12:11]	R	mpc_reserved[1:0]	-	MPC reserved latched after delay
[13]	RW	mpc_sel_ttc_bx0	1	1=MPC gets ttc_bx0, 0=bx0_local
[14]	RW	mpc_idle_blank	0	1=blank mpc data & bx0 except when triggered
[15]	RW	mpc_oe	1	1=enable outputs to MPC, 0=aset FFs to 1s

Adr 88 ADR_MPC0_FRAME0 MPC0 Frame0 Data Sent to MPC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1st vpf	lct 1st q 3	lct 1st q 2	lct 1st q 1	lct 1st q 0	clct 1st pat 3	clct 1st pat 2	clct 1st pat 1	clct 1st pat 0	alct 1st wg 6	alct 1st wg 5	alct 1st wg 4	alct 1st wg 3	alct 1st wg 2	alct 1st wg 1	alct 1st wg 0

Bits	Dir	Signal	Typical	Description
[06:00]	R	alct_first_key[6:0]	$0\text{-}111_{10}$	ALCT first key wire-group
[10:07]	R	clct_first_pat[3:0]	0-10	CLCT first pattern number
[14:11]	R	lct_first_quality[3:0]	8	LCT first muon quality
[15]	R	first_vpf	1	First valid pattern flag

Adr 8A ADR_MPC0_FRAME1 MPC0 Frame1 Data Sent to MPC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
csc id 3	csc id 2	csc id 1	csc id 0	tmb bx0 local	alct 1st bxn 0	sync err	clct 1st bend	clct 1st key 7	clct 1st key 6	clct 1st key 5	clct 1st key 4	clct 1st key 3	clct 1st key 2	clct 1st key 1	clct 1st key 0

Bits	Dir	Signal	Typical	Description
[07:00]	R	clct_first_key[7:0]	$0\text{-}159_{10}$	CLCT first muon key ½-strip
[08]	R	clct_first_bend	0	CLCT first muon bend direction
[09]	R	sync_err	0	BXN does not match at BX0
[10]	R	alct_first_bxn[0]	0-1	ALCT first muon bunch crossing number
[11]	R	clct_first_bx0_local	0-1	1=TMBs bxn[11:0]==0
[15:12]	R	csc_id[3:0]	1-9	CSC chamber ID

Adr 8C ADR_MPC1_FRAME0 MPC1 Frame0 Data Sent to MPC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
2nd vpf	lct 2nd q 3	lct 2nd q 2	lct 2nd q 1	lct 2nd q 0	clct 2nd pat 3	clct 2nd pat 2	clct 2nd pat 1	clct 2nd pat 0	alct 2nd wg 6	alct 2nd wg 5	alct 2nd wg 4	alct 2nd wg 3	alct 2nd wg 2	alct 2nd wg 1	alct 2nd wg 0

Bits	Dir	Signal	Typical	Description
[06:00]	R	alct_second_key[6:0]	$0\text{-}111_{10}$	ALCT second key wire-group
[10:07]	R	clct_second_pat[3:0]	0-10	CLCT second pattern number
[14:11]	R	lct_second_quality[3:0]	8	LCT second muon quality
[15]	R	second_vpf	1	Second valid pattern flag

Adr 8E ADR_MPC1_FRAME1 MPC1 Frame1 Data Sent to MPC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
csc id 3	csc id 2	csc id 1	csc id 0	tmb bx0 local	alct 2nd bxn 0	sync err	clct 2nd bend	clct 2nd key 7	clct 2nd key 6	clct 2nd key 5	clct 2nd key 4	clct 2nd key 3	clct 2nd key 2	clct 2nd key 1	clct 2nd key 0

Bits	Dir	Signal	Typical	Description
[07:00]	R	clct_second_key[7:0]	$0\text{-}159_{10}$	CLCT second muon key ½-strip
[08]	R	clct_second_bend	0	CLCT second muon bend direction
[09]	R	sync_err	0	BXN does not match at BX0
[10]	R	alct_second_bxn[0]	0-1	ALCT second muon bunch crossing number
[11]	R	clct_second_bx0_local	0-1	1=TMBs bxn[11:0]==0
[15:12]	R	csc_id[3:0]	1-9	CSC chamber ID

Adr 90 ADR_MPC_INJ MPC Injector Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mpc inj clct bx0	mpc inj alct bx0	mpc reserv 1	mpc reserv 0	mpc accep t 1	mpc accep t 0	ttc inj enabl e	mpc inject	mpc nfram 7	mpc nfram 6	mpc nfram 5	mpc nfram 4	mpc nfram 3	mpc nfram 2	mpc nfram 1	mpc nfram 0

Bits	Dir	Signal	Default	Description
[07:00]	RW	mpc_nframes[7:0]	5	Number frames to inject
[08]	RW	mpc_inject	0	1=Start MPC test pattern injector
[09]	RW	ttc_mpc_inj_en	1	1=Enable injector start by TTC command
[11:10]	R	mpc_accept[1:0]	-	MPC accept stored at injector RAM address
[13:12]	R	mpc_reserved[1:0]	-	MPC reserved stored at injector RAM address
[14]	RW	mpc_inj_alct_bx0	0	1=Fire alct_bx0 one-shot
[15]	RW	mpc_inj_clct_bx0	0	1=Fire clct_bx0 one-shot

Adr 92 ADR_MPC_RAM_ADR MPC Injector RAM Address

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mpc adr 7	mpc adr 6	mpc adr 5	mpc adr 4	mpc adr 3	mpc adr 2	mpc adr 1	mpc adr 0	mpc ren 3	mpc ren 2	mpc ren 1	mpc ren 0	mpc wen 3	mpc wen 2	mpc wen 1	mpc wen 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	mpc_wen[3:0]	0	Select RAM to write
[07:04]	RW	mpc_ren[3:0]	0	Select RAM to read
[15:08]	RW	mpc_adr[7:0]	0	Injector RAM read/write address

Adr 94 ADR_MPC_RAM_WDATA MPC Injector RAM Write Data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mpc wdata 7	mpc wdata 6	mpc wdata 5	mpc wdata 4	mpc wdata 3	mpc wdata 2	mpc wdata 1	mpc wdata 0	mpc wdata 3	mpc wdata 2	mpc wdata 1	mpc wdata 0	mpc wdata 3	mpc wdata 2	mpc wdata 1	mpc wdata 0

Bits	Dir	Signal	Default	Description
[15:00]	RW	mpc_wdata[15:0]	0	MPC Injector RAM write data

Adr 96 ADR_MPC_RAM_RDATA MPC Injector RAM Read Data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
mpc rdata 7	mpc rdata 6	mpc rdata 5	mpc rdata 4	mpc rdata 3	mpc rdata 2	mpc rdata 1	mpc rdata 0	mpc rdata 3	mpc rdata 2	mpc rdata 1	mpc rdata 0	mpc rdata 3	mpc rdata 2	mpc rdata 1	mpc rdata 0

Bits	Dir	Signal	Default	Description
[15:00]	R	mpc_rdata[15:0]	0	MPC Injector RAM read data

Adr 98 ADR_SCP_CTRL Scope Control [see Adr9A p48 SCP_RDATA and adr CE p59 SCP_TRIG]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		trig done	waitin for trigge r	ram sel 3	ram sel 2	ram sel 1	ram sel 0	tbins 2	tbins 1	tbins 0	no write	auto	force trig	run stop	trig en

Bits	Dir	Signal	Default	Description
[00]	RW	scp_ch_trig_en	1	1=Enable channel triggers, see AdrCE p59 for ch
[01]	RW	scp_runstop	0	1=Run, 0=Stop
[02]	RW	scp_force_trig	0	1=Force a trigger (to trig: set 0,1,0 in 3 writes)
[03]	RW	scp_auto	0	Sequencer readout mode 1=insert in DMB data
[04]	RW	scp_nowrite	0	1=Preserve initial RAM test pattern for debug
[07:05]	RW	scp_tbins[2:0]	4	Auto mode tbins per channel code, actual tbins/ch=(scp_tbins+1)*64, spans 64-512
[11:08]	RW	scp_ram_sel[3:0]	0	RAM bank address 0-9 for VME readout
[12]	R	scp_waiting	-	Scope waiting for trigger
[13]	R	scp_trig_done	-	Scope triggered, ready for readout
[15:14]	RW	-		Unassigned

Adr 9A ADR_SCP_RDATA Scope Read data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rdata 15	rdata 14	rdata 13	rdata 12	rdata 11	rdata 10	rdata 9	radr8 rdata 8	radr7 rdata 7	radr6 rdata 6	radr5 rdata 5	radr4 rdata 4	radr3 rdata 3	radr2 rdata 2	radr1 rdata 1	radr0 rdata 0

Bits	Dir	Signal		Description
[08:00]	W	scp_radr[8:0]		Scope data read address=tbin for selected RAM
[15:00]	R	scp_rdata[15:0]		See channel assignments below

Scope Channel Assignments

```
// Pre-trigger to DMB
        assign scp_ch[0]        = (clct_sm == pretrig);     // Trigger alignment marker, scope triggers on this ch usually
        assign scp_ch[1]        = triad_tp[0];              // Triad test point at input to raw hits RAM
        assign scp_ch[2]        = any_cfeb_hit;             // Any CFEB over threshold
        assign scp_ch[3]        = active_feb_flag;          // Active feb flag to DMB
        assign scp_ch[8:4]      = active_feb_list[4:0];     // Active feb list to DMB

// Pre-trigger CLCT*ALCT matching
        assign scp_ch[9]        = alct_active_feb;          // ALCT active feb flag, should precede alct0_vpf
        assign scp_ch[10]       = alct_pretrig_window;      // ALCT*CLCT pretrigger matching window

// Pre-trigger Processing
        assign scp_ch[13:11]    = clct_sm_vec[2:0];         // Pre-trigger state machine
        assign scp_ch[14]       = wr_buf_ready;             // Write buffer ready
        assign scp_ch[15]       = (clct_sm == pretrig);     // Skip channels 15,31,47,63,79,95,111,127,143,159
        assign scp_ch[27:16]    = bxn_counter[11:0];        // BXN counter
        assign scp_ch[28]       = discard_nowrbuf;          // Event discard, no write buffer

// CLCT Pattern Finder Output
        assign scp_ch[29]       = 0;
```

```
        assign scp_ch[30]       = 0;
        assign scp_ch[31]       = (clct_sm == pretrig);     // Skip channels 15,31,47,63,79,95,111,127,143,159

        assign scp_ch[34:32]    = hs_hit_1st[2:0];          // CLCT0 number hits after drift
        assign scp_ch[38:35]    = hs_pid_1st[3:0]; // CLCT0 Pattern number
        assign scp_ch[46:39]    = hs_key_1st[7:0];// CLCT0 ½-strip ID number

        assign scp_ch[47]       = (clct_sm == pretrig);     // Skip channels 15,31,47,63,79,95,111,127,143,159

        assign scp_ch[50:48]    = hs_hit_2nd[2:0]; // CLCT1 number hits after drift
        assign scp_ch[54:51]    = hs_pid_2nd[3:0];          // CLCT1 Pattern number
        assign scp_ch[62:55]    = hs_key_2nd[7:0];          // CLCT1 ½-strip ID number

        assign scp_ch[63]       = (clct_sm == pretrig);     // Skip channels 15,31,47,63,79,95,111,127,143,159

// CLCT Builder
        assign scp_ch[64]       = clct0_really_valid;       // CLCT0 is over threshold, not forced by an external trigger
        assign scp_ch[65]       = clct0_vpf;                // CLCT0 vpf
        assign scp_ch[66]       = clct1_vpf;                // CLCT1 vpf
        assign scp_ch[67]       = clct_push_xtmb;           // CLCT sent to TMB matching
        assign scp_ch[68]       = discard_invp;             // CLCT discarded, below threshold after drift

// TMB Matching
        assign scp_ch[69]       = alct0_valid;              // ALCT0 vpf direct from 80MHz receiver, before alct_delay
        assign scp_ch[70]       = alct1_valid;              // ALCT1 vpf direct from 80MHz receiver, before alct_delay

        assign scp_ch[71]       = alct_vpf_tprt;            // ALCT vpf in TMB after pipe delay, unbuffered real time
        assign scp_ch[72]       = clct_vpf_tprt;            // CLCT vpf in TMB
        assign scp_ch[73]       = clct_window_tprt;         // CLCT matching window in TMB
        assign scp_ch[77:74]    = tmb_match_win[3:0];       // Location of alct in clct window
        assign scp_ch[78]       = tmb_alct_discard;         // ALCT pair was not used for LCT

        assign scp_ch[79]       = (clct_sm == pretrig);     // Skip channels 15,31,47,63,79,95,111,127,143,15

        assign scp_ch[80]       = tmb_clct_discard;         // CLCT pair was not used for LCT

// TMB Match Results
        assign scp_ch[81]       = tmb_trig_pulse;           // TMB Triggered on ALCT or CLCT or both
        assign scp_ch[82]       = tmb_trig_keep;            // ALCT or CLCT or both triggered, and trigger is allowed
        assign scp_ch[83]       = tmb_match;                // ALCT and CLCT matched in time
        assign scp_ch[84]       = tmb_alct_only;            // Only ALCT triggered
        assign scp_ch[85]       = tmb_clct_only;            // Only CLCT triggered
        assign scp_ch[86]       = discard_tmbreject;        // TMB discarded event

// MPC
        assign scp_ch[87]       = mpc_xmit_lct0;            // MPC LCT0 sent
        assign scp_ch[88]       = mpc_xmit_lct1;            // MPC LCT1 sent
        assign scp_ch[89]       = mpc_response_ff;          // MPC accept is ready
        assign scp_ch[91:90]    = mpc_accept_ff[1:0];       // MPC muon accept response

// L1A
        assign scp_ch[92]       = l1a_pulse;                // L1A strobe from ccb or internal
        assign scp_ch[93]       = l1a_window_open;          // L1A window open duh
        assign scp_ch[94]       = l1a_match;                // L1A strobe match in window

        assign scp_ch[95]       = (clct_sm == pretrig);     // Skip channels 15,31,47,63,79,95,111,127,143,159
```

```
// Buffer push at L1A
        assign scp_ch[96]       = buf_push;             // Allocate write buffer space for this event
        assign scp_ch[103:97]   = buf_push_adr[6:0];    // Address of write buffer to allocate

// DMB Readout
        assign scp_ch[104]      = dmb_dav;              // DAV to DMB
        assign scp_ch[105]      = dmb_busy;             // Readout in progress
        assign scp_ch[110:106]  = read_sm_vec[4:0];     // Readout state machine

        assign scp_ch[111]      = (clct_sm == pretrig); // Skip channels 15,31,47,63,79,95,111,127,143,159

        assign scp_ch[126:112]  = seq_wdata[14:0];      // DMB dump image, very cool
        assign scp_ch[127]      = (clct_sm == pretrig); // Skip channels 15,31,47,63,79,95,111,127,143,159
        assign scp_ch[128]      = seq_wdata[15];        // DMB dump image, very cool

// CLCT+TMB Pipelines
        assign scp_ch[132:129]  = wr_buf_adr[3:0];      // Event address counter

        assign scp_ch[133]      = wr_push_xtmb;         // Buffer write strobe after drift time
        assign scp_ch[137:134]  = wr_adr_xtmb[3:0];     // Buffer write address after drift time

        assign scp_ch[138]      = wr_push_rtmb;         // Buffer write strobe at TMB matching time
        assign scp_ch[142:139]  = wr_adr_rtmb[3:0];     // Buffer write address at TMB matching time

        assign scp_ch[143]      = (clct_sm == pretrig); // Skip channels 15,31,47,63,79,95,111,127,143,159

        assign scp_ch[144]      = wr_push_xmpc;// Buffer write strobe at MPC xmit to sequencer
        assign scp_ch[148:145]  = wr_adr_xmpc[3:0];     // Buffer write address at MPC xmit to sequencer

        assign scp_ch[149]      = wr_push_rmpc;         // Buffer write strobe at MPC received
        assign scp_ch[153:150]  = wr_adr_rmpc[3:0];     // Buffer write address at MPC received

// Buffer pop at readout completion
        assign scp_ch[154]      = buf_pop;              // Specified buffer is to be released
        assign scp_ch[158:155]  = buf_pop_adr[3:0];     // Address of read buffer to release

        assign scp_ch[159]      = (clct_sm == pretrig); // Skip channels 15,31,47,63,79,95,111,127,143,159
```

Adr 9C ADR_CCB_CMD CCB TTC Command Generator (Internal)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ccb cmd 7	ccb cmd 6	ccb cmd 5	ccb cmd 4	ccb cmd 3	ccb cmd 2	ccb cmd 1	ccb cmd 0	0	fmm state 2	fmm state 1	fmm state 0	subadr strobe	data strobe	brcst strobe	discon ccb

Bits	Dir	Signal	Default	Description
[00]	RW	vme_ccb_cmd_enable	0	1=Disconnect CCB backplane ccb_cmd[7:0]
[01]	RW	vme_ccb_cmd_strobe	0	1=Assert internal ccb_cmd brcst strobe
[02]	RW	vme_ccb_data_strobe	0	1=Assert internal ccb_cmd data strobe
[03]	RW	vme_ccb_subaddr_strobe	0	1=Assert internal ccb_cmd sub-adr strobe
[06:04]	R	fmm_state[2:0]	-	FMM machine states: 0: fmm_startup 1: fmm_resync 2: fmm_stop 3: fmm_wait_bx0 4 : fmm_run
[07]	RW	-		Unassigned
[15:08]	RW	vme_ccb_cmd[7:0]	0	TTC command to generate

Adr 9E ADR_BUF_STAT0 Buffer Status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
buffer disp 7	buffer disp 6	buffer disp 5	buffer disp 4	buffer disp 3	buffer disp 2	buffer disp 1	buffer disp 0	buf stalled once	buf q adrerr	buf q udferr	buf q ovferr	buf q empty	buf q full	buf stalled	wr buf ready

Bits	Dir	Signal	Typical	Description
[00]	R	wr_buf_ready	1	Write buffer is ready
[01]	R	buf_stalled	0	Buffer write pointer hit a fence and stalled
[02]	R	buf_q_full	0	All raw hits ram in use, ram writing must stop
[03]	R	buf_q_empty	0	No fences remain in buffer queue
[04]	R	buf_q_ovf_err	0	Tried to push new event when queue full
[05]	R	buf_q_udf_err	0	Tried to pop event when queue empty
[06]	R	buf_q_adr_err	0	Fence adr popped from queue doesn't match expected adr
[07]	R	buf_stalled_once	0	Buffer stalled at least once since last resync
[15:08]	R	buf_display[7:0]	0	Buffer fraction in use, for in-board LED display

Adr A0 ADR_BUF_STAT1 Buffer Status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	buf adr 10	buf adr 9	buf adr 8	buf adr 7	buf adr 6	buf adr 5	buf adr 4	buf adr 3	buf adr 2	buf adr 1	buf adr 0

Bits	Dir	Signal	Typical	Description
[10:00]	R	wr_buf_adr[10:0]	-	Current address of event & header write buffer
[15:11]	R	-	0	Unassigned

Adr A2 **ADR_BUF_STAT2** **Buffer Status**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	buf fence dist 10	buf fence dist 9	buf fence dist 8	buf fence dist 7	buf fence dist 6	buf fence dist 5	buf fence dist 4	buf fence dist 3	buf fence dist 2	buf fence dist 1	buf fence dist 0

Bits	Dir	Signal	Typical	Description
[10:00]	R	buf_fence_dist[10:0]	-	Distance to 1st fence address
[15:11]	R	-	0	Unassigned

Adr A4 **ADR_BUF_STAT3** **Buffer Status**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	buf fence cntr 11	buf fence cnt 10	buf fence cnt 9	buf fence cnt 8	buf fence cnt 7	buf fence cnt 6	buf fence cnt 5	buf fence cnt 4	buf fence cnt 3	buf fence cnt 2	buf fence cnt 1	buf fence cnt 0

Bits	Dir	Signal	Typical	Description
[11:00]	R	buf_fence_cnt[11:0]	-	Number of fences in fence RAM currently
[15:12]	R	-	0	Unassigned

Adr A6 **ADR_BUF_STAT4** **Buffer Status**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	buf fence peak 11	buf fence peak 10	buf fence peak 9	buf fence peak 8	buf fence peak 7	buf fence peak 6	buf fence peak 5	buf fence peak 4	buf fence peak 3	buf fence peak 2	buf fence peak 1	buf fence peak 0

Bits	Dir	Signal	Typical	Description
[11:00]	R	buf_fence_cnt_peak [11:0]	-	Peak number of fences in fence RAM
[15:12]	R	-	0	Unassigned

Adr A8 ADR_ALCTFIFO1 ALCT Raw Hits RAM Control

(Split with Adr 3E ADR_ALCT_FIFO0 and Adr A4 ADR_ALCT_FIFO2)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	alct demux mode	0	alct radr 10	alct radr 9	alct radr 8	alct radr 7	alct radr 6	alct radr 5	alct radr 4	alct radr 3	alct radr 2	alct radr 1	alct radr 0	alct raw reset

Bits	Dir	Signal	Default	Description
[00]	RW	alct_raw_reset	0	Reset ALCT raw hits FIFO controller
[11:01]	RW	alct_raw_radr[10:0]	0	ALCT raw hits RAM read address or demux wd
[12]	RW	--	0	Unassigned
[13]	RW	alct_demux_mode	0	0=alctfifo2 has RAM data, 1=fifo2=demux data
[15:14]	RW	--	0	Unassigned

Adr AA ADR_ALCTFIFO2 ALCT Raw Hits RAM data (LSBs)

(Split with Adr 3E ADR_ALCT_FIFO0 and Adr A2 ADR_ALCT_FIFO1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
alct fifo 15	alct fifo 14	alct fifo 13	alct fifo 12	alct fifo 11	alct fifo 10	alct fifo 9	alct fifo 8	alct fifo 7	alct fifo 6	alct fifo 5	alct fifo 4	alct fifo 3	alct fifo 2	alct fifo 1	alct fifo 0

Bits	Dir	Signal	Default	Description
[15:00]	R	alct_raw_rdata[15:0] OR alct_1st_vme[14:1] alct_1st_vme[28:15] alct_2nd_vme[14:1] alct_2nd_vme[28:15]		ALCT FIFO data (msbs in adr_alct_fifo) alct_raw_radr=0 and alct_demux_mode=1 alct_raw_radr=1 alct_raw_radr=2 alct_raw_radr=3

Adr AC ADR_SEQMOD Sequencer Trigger Modifiers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
event clear vme	active feb src	Scint veto state	Clear scint veto	L1A alct only	L1A tmb nol1a	L1A no tmb	L1A tmb trig	valid clct requir	wr buf requir	hdr wr cont	wrbuf auto clr	flush timer 3	flush timer 2	flush timer 1	flush timer 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	clct_flush_delay[3:0]	1	Trigger sequencer flush state timer
[04]	RW	wr_buf_autoclr_en	1	1=Enable frozen buffer auto clear
[05]	RW	hdr_wr_continuous	0	1=allow continuous header buffer writing for invalid triggers
[06]	RW	wr_buf_required	1	Require wr_buffer available to pre-trigger
[07]	RW	valid_clct_required	1	Require valid CLCT after drift delay
[08]	RW	l1a_allow_match	1	Readout allows tmb trig pulse in L1A window
[09]	RW	l1a_allow_notmb	0	Readout allows notmb trig pulse in L1A window
[10}	RW	l1a_allow_nol1a	0	Readout allows tmb trig pulse outside L1A wind
[11]	RW	l1a_allow_alct_only	0	Allow ALCT-only events to readout at L1A
[12]	RW	scint_veto_clr	0	Clear scintillator veto FF
[13]	R	scint_veto_vme	0	Scintillator veto FF state
[14]	RW	active_feb_src	0	Active feb flag source, 0=pretrig, 1=tmb match
[15]	RW	event_clear_vme	0	Event clear for aff,clct,mpc VME diagnostic registers

Adr AE **ADR_SEQSM** **Sequencer Machine State**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	buf_q adr err	buf_q ovf	buf_q empty	buf_q full	read sm 2	read sm 4	read sm 3	read sm 2	read sm 1	read sm 0	clct sm 2	clct sm 1	clct sm 0

Bits	Dir	Signal		Description
[02:00]	R	clct_sm[2:0]		CLCT Trigger machine state
[07:03]	R	read_sm[4:0]		Readout machine state
[08]	R	buf_q_full		All raw hits ram in use, ram writing must stop
[09]	R	buf_q_empty		No fences remain in buffer queue
[10]	R	buf_q_ovf_err		Tried to push new event when queue full
[11]	R	buf_q_adr_err		Tried to pop event when queue empty
[15:12]	R	--	0	Unassigned

Adr B0 **ADR_SEQCLCTM** **Sequencer CLCT (MSBs)**

(Split with Adr 78 ADR_SEQCLCT0 and Adr 7A ADR_SEQCLCT1)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sync err	clock lock lost	0	0	0	0	active cfeb 6	active cfeb 5	active cfeb 4	active cfeb 3	active cfeb 2	active cfeb 1	active cfeb 0	sync err	clct bxn 1	clct bxn 0

Bits	Dir	Signal	Typ	Description
[01:00]	R	clctc[1:0] bxn_counter	-	Bunch crossing number at pretrigger, common to clct0/1
[02]	R	clctc[2] sync_err	0	BX0 disagrees with BXN counter, common to clct0/1
[9:3]	R	clctf[6:0]	-	Active feb list latched at TMB alct*clct matching time
[13:10]	R	--	0	Unassigned
[14]	R	clock_lock_lost FF	0	40MHz main clock lost lock, global_reset asserted
[15]	R	sync_err (direct)	0	Sync error: bxn counter==0 does not match bx0

Adr B2 **ADR_TMBTIM** **TMB Timing for ALCT*CLCT Coincidence**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	mpctx delay 3	mpctx delay 2	mpctx delay 1	mpctx delay 0	clct wind 3	clct wind 2	clct wind 1	clct wind 0	alct delay 3	alct delay 2	alct delay 1	alct delay 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	alct_delay[3:0]	1	Delay ALCT for CLCT match window
[07:04]	RW	clct_window[3:0]	3	CLCT match window width
[11:08]	RW	mpc_tx_delay[3:0]	0	MPC transmit delay
[15:12]	RW	--	0	Unassigned

Adr B4 **ADR_LHC_CYCLE** **LHC Cycle Period, Maximum BXN Count**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	lhc cycle 11	lhc cycle 10	lhc cycle 9	lhc cycle 8	lhc cycle 7	lhc cycle 6	lhc cycle 5	lhc cycle 4	lhc cycle 3	lhc cycle 2	lhc cycle 1	lhc cycle 0

Bits	Dir	Signal	Default	Description
[11:00]	RW	lhc_cycle[11:0]	3564	Maximum bxn+1 3564(hDEC) for LHC 924(h39C) for beam test
[15:12]	RW	--	0	Unassigned

Adr B6 **ADR_RPC_CFG** **RPC Configuration**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	rpc done	read bxn 2	read bxn 1	read bxn 0	read bank 1	read bank 0	bxn offset 3	bxn offset 2	bxn offset 1	bxn offset 0	read enable	0	0	rpc1 exists	rpc0 exists

Bits	Dir	Signal	Default	Description
[01:00]	RW	rpc_exists[1:0]	3	Bit (n) = 1 = RPC(n) Exists
[03:02]	RW	--	0	Unused
[04]	RW	rpc_read_enable	1	1=Include Existing RPCs in DMB Readout
[08:05]	RW	rpc_bxn_offset[3:0]	0	RPC BXN offset
[10:09]	RW	rpc_bank[1:0]	0	RPC bank address, for reading rdata sync mode
[13:11]	R	rpc_rbxn[2:0]	-	RPC rdata[18:16] msbs for sync mode, adr 1E
[14]	R	rpc_done	1	RPC FPGA reports configuration done
[15]	RW	--	0	Unassigned

Adr B8 **ADR_RPC_RDATA** **RPC Raw Hits Sync Mode Read Data,** See Adr 1E p29

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rdata 15	rdata 14	rdata 13	rdata 12	rdata 11	rdata 10	rdata 9	rdata 8	rdata 7	rdata 6	rdata 5	rdata 4	rdata 3	rdata 2	rdata 1	rdata 0

Bits	Dir	Signal	Default	Description
[15:00]	R	rpc_rdata[15:0]	-	RPC RAM read data for sync mode

Adr BA **ADR_RPC_RAW_DELAY** **RPC Raw Hits Data Delay + RPC BXN Differences**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rpc1 bxn diff3	rpc1 bxn diff2	rpc1 bxn diff1	rpc1 bxn diff0	rpc0 bxn diff3	rpc0 bxn diff2	rpc0 bxn diff1	rpc0 bxn diff0	rpc1 delay 3	rpc1 delay 2	rpc1 delay 1	rpc1 delay 0	rpc0 delay 3	rpc0 delay 2	rpc0 delay 1	rpc0 delay 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	rpc0_delay[3:0]	1	RPC0 Raw hits data delay
[07:04]	RW	rpc1_delay[3:0]	1	RPC1 Raw hits data delay
[11:08]	R	rpc0_bxn_diff[3:0];	-	RPC bxn – Offset (in adr B6)
[15:12]	R	rpc1_bxn_diff[3:0];	-	RPC bxn – Offset (in adr B6)

Adr BC ADR_RPC_INJ RPC Injector Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	rpc tbins test	inj rdata 18	inj rdata 17	inj rdata 16	inj wdata 18	inj wdata 17	inj wdata 16	inj sel	delay rat 3	delay rat 2	delay rat 1	delay rat 0	mask rpc	mask rat	mask all

Bits	Dir	Signal	Default	Description
[00]	RW	rpc_mask_all	1	1=Enable RPC Inputs from RAT, 0=disable all
[01]	RW	injector_mask_rat	0	1=Enable RAT for injector fire
[02]	RW	injector_mask_rpc	1	1=Enable RPC injector RAM for injector fire
[06:03]	RW	inj_delay_rat[3:0]	0	CFEB/RPC injectors wait for RAT
[07]	RW	rpc_inj_sel	0	1=Enable injector RAM write
[10:08]	RW	rpc_inj_wdata[18:16]	0	RPC injector write data MSBs, see adr C0 p56
[13:11]	R	rpc_inj_rdata[18:16]	-	RPC injector read data MSBs, see adr C0 p56
[14]	RW	rpc_tbins_test	0	Set write_data=address test mode
[15]	RW	--	0	Unassigned

Adr BE ADR_RPC_INJ_ADR RPC Injector RAM Addresses

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
inj adr 7	inj adr 6	inj adr 5	inj adr 4	inj adr 3	inj adr 2	inj adr 1	inj adr 0	0	0	inj ren 1	inj ren 0	0	0	inj wen 1	inj wen 0

Bits	Dir	Signal	Default	Description
[01:00]	RW	rpc_inj_wen[1:0]	0	1=Write enable injector RAMn
[03:02]	RW	--	0	Unused
[05:04]	RW	rpc_inj_ren[1:0]	0	1=Read enable Injector RAMn
[15:06]	RW	inj_rwadr[9:0]	0	Injector RAM read/write address

Adr C0 ADR_RPC_INJ_WDATA RPC Injector Write Data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
inj wdata 15	inj wdata 14	inj wdata 13	inj wdata 12	inj wdata 11	inj wdata 10	inj wdata 9	inj wdata 8	inj wdata 7	inj wdata 6	inj wdata 5	inj wdata 4	inj wdata 3	inj wdata 2	inj wdata 1	inj wdata 0

Bits	Dir	Signal	Default	Description
[15:00]	RW	rpc_inj_wdata[15:0]	0	RPC RAM write data LSBs (see Adr BC msbs p56)

Adr C2 ADR_RPC_INJ_RDATA RPC Injector Read Data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
inj rdata 15	inj rdata 14	inj rdata 13	inj rdata 12	inj rdata 11	inj rdata 10	inj rdata 9	inj rdata 8	inj rdata 7	inj rdata 6	inj rdata 5	inj rdata 4	inj rdata 3	inj rdata 2	inj rdata 1	inj rdata 0

Bits	Dir	Signal	Default	Description
[15:00]	R	rpc_inj_rdata[15:0]	-	RPC RAM read data LSBs (see Adr BC msbs p56)

Adr C4 ADR_RPC_TBINS RPC FIFO Time Bins

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	rpc decouple	rpc pre 4	rpc pre 3	rpc pre 2	rpc pre 1	rpc pre 0	rpc tbins 4	rpc tbins 3	rpc tbins 2	rpc tbins 1	rpc tbins 0

Bits	Dir	Signal	Default	Description
[04:00]	RW	fifo_tbins_rpc[4:0]	7	Number RPC FIFO time bins to read out
[09:05]	RW	fifo_pretrig_rpc[4:0]	2	Number RPC FIFO time bins before pretrigger
[10]	RW	rpc_decouple	0	1=Independent rpc tbins, 0=copy cfeb tbins
[15:11]	RW	--	0	Unused

Adr C6 ADR_RPC0_HCM RPC0 Hot Channel Mask

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
enable pad15	enable pad14	enable pad13	enable pad12	enable pad11	enable pad10	enable pad9	enable pad8	enable pad7	enable pad6	enable pad5	enable pad4	enable pad3	enable pad2	enable pad1	enable pad0

Bits	Dir	Signal	Default	Description
[15:00]	RW	rpc0_hcm[15:0]	FFFF	Bit(n)=1=Enable RPC Pad(n), FFFF=enable all

Adr C8 ADR_RPC1_HCM RPC1 Hot Channel Mask

Bits	Dir	Signal	Default	Description
[15:00]	RW	rpc1_hcm[15:0]	FFFF	Bit(n)=1=Enable RPC Pad(n), FFFF=enable all

Adr CA ADR_BX0_DELAY BX0 to MPC Delays

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	bx0 match	bx0 vpf test	alct bx0 enable	clct bx dly 3	clct bx dly 2	clct bx dly 1	clct bx dly 0	alct bx dly 3	alct bx dly 2	alct bx dly 1	alct bx dly 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	alct_bx0_delay[3:0]	0	ALCT bx0 delay to mpc transmitter
[07:04]	RW	clct_bx0_delay[3:0]	0	CLCT bx0 delay to mpc transmitter
[08]	RW	alct_bx0_enable	1	1=Enable using alct bx0, else copy clct bx0
[09]	RW	bx0_vpf_test	0	Sets clct_bx0=lct0_vpf for bx0 alignment tests
[10]	R	bx0_match	1	1=alct_bx0==clct_bx0, latched at clct_bx0
[15:11]	RW	--	-	Unused

Adr CC ADR_NON_TRIG_RO Non-Triggering Event Enables + ME1A/B Reversal

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
csc type 3	csc type 2	csc type 1	csc type 0	0	0	reverse me1b	reverse me1a	reverse csc	stagger csc	csc me1a b	cnt non me1a b	mpc me1a block	allow match ro	allow clct ro	allow alct ro

Bits	Dir	Signal	Default Type A	Description
[00]	RW	tmb_allow_alct_ro	0	1=Allow ALCT-only non-triggering readout
[01]	RW	tmb_allow_clct_ro	0	1=Allow CLCT-only non-triggering readout
[02]	RW	tmb_allow_match_ro	1	1=Allow ALCT*CLCT non-triggering readout
[03]	RW	mpc_me1a_block	1	Block ME1A LCTs from MPC, still queue for readout
[04]	RW	cnt_non_me1ab_en	1	Allow clct pretrig counters count non me1ab
[05]	R	csc_me1ab	0	1= CSC is ME1A or ME1B. 0=normal CSC
[06]	R	stagger_hs_csc	1	1=Staggered CSC, 0=non-staggered
[07]	R	reverse_hs_csc	0	1=Reversed staggered CSC, non-me1
[08]	R	reverse_hs_me1a	0	1=reversed me1a hstrips
[09]	R	reverse_hs_me1b	0	1=reversed me1b hstrips
[11:10]	RW	--	0	Free 2
[15:12]	R	csc_type[3:0]	A	Firmware compile type A=Normal CSC B=Reversed CSC C=Normal ME1B, Reversed ME1A D=Reversed ME1B, Normal ME1A

Firmware Compile Type Codes A,B,C,D:

CSC_type	stagger_hs_csc	reverse_hs_csc	reverse_hs_me1a	reverse_hs_me1b	csc_me1ab
A	1	0	0	0	0
B	0	1	0	0	0
C	0	0	1	0	1
D	0	0	0	1	1

Adr CE **ADR_SCP_TRIG** **Scope Trigger Source Channel**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	ch6	ch5	ch4	ch3	ch2	ch1	ch0

Bits	Dir	Signal	Default	Description
[06:00]	RW	trigger_ch[6:0]	0	ch0=trigger on sequencer→ pretrig
[14:07]	RW	--	0	Unassigned
[15]	RW	scp_ch_overlay	0	0=normal ch assignments, 1=debug assigments

Adr D0 **ADR_CNT_CTRL** **Status Counter Control [See Adr D2 p62]**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cnt select 6	cnt select 5	cnt select 4	cnt select 3	cnt select 2	cnt select 1	cnt select 0	cnt adr lsb	hdr clr resync	cnt clr resync	en alct debug	seq cnt ovf	alct cnt ovf	stop on ovf	snap shot	clear all

Bits	Dir	Signal	Default	Description
[00]	RW	cnt_all_clear	-	Clear VME counters (also clr on ccb_evcntres)
[01]	RW	cnt_snapshot	-	Take snapshot of current counter state
[02]	RW	cnt_stop_on_ovf	0	Stop all counters if any overflows
[03]	R	cnt_any_ovf_alct	-	At least 1 alct counter overflowed
[04]	R	cnt_any_ovf_seq	-	At least 1 sequencer counter overflowed
[05]	RW	cnt_alct_debug	1	1=enable alct_lct_err counter
[06]	RW	cnt_clear_on_resync	0	1=Clear VME counters on ttc_resync
[07]	RW	hdr_clear_on_resync	1	1=Clear header counters on ttc_resync
[08]	RW	cnt_adr_lsb	-	Counter half select, 0=bits[15:0], 1=bits[29:16]
[15:09]	RW	cnt_select[6:0]	-	Counter select address

Counter Select Addresses (even cnt_ad_lsb=LSBs, odd cnt_adr_lsb=MSBs)

$Address_{10}$	Counter Description	Bits	VME Clears
00	ALCT: alct0 vpf received	30	Y
01	ALCT: alct1 vpf received	30	Y
02	ALCT: alct data structure error	30	Y
03	ALCT: trigger path ECC 1-bit error corrected	30	Y
04	ALCT: trigger path ECC 2-bit error not corrected	30	Y
05	ALCT: trigger path ECC >2-bit error not corrected	30	Y
06	ALCT: trigger path ECC ≥2-bit error not corrected, ALCT data blanked	30	Y
07	ALCT: alct replied ECC 1-bit error corrected	30	Y
08	ALCT: alct replied ECC 2-bit error not corrected	30	Y
09	ALCT: alct replied ECC >2-bit error not corrected	30	Y
10	ALCT: raw hits readout	30	Y
11	ALCT: raw hits readout CRC error	30	Y
12		30	Y
13	CLCT: Pre-trigger was on any CFEB	30	Y
14	CLCT: Pre-trigger includes a CLCT on CFEB0	30	Y
15	CLCT: Pre-trigger includes a CLCT on CFEB1	30	Y
16	CLCT: Pre-trigger includes a CLCT on CFEB2	30	Y

17	CLCT: Pre-trigger includes a CLCT on CFEB3	30	Y
18	CLCT: Pre-trigger includes a CLCT on CFEB4	30	Y
19	CLCT: Pre-trigger includes a CLCT on CFEB5	30	Y
20	CLCT: Pre-trigger includes a CLCT on CFEB6	30	Y
21	CLCT: Pre-trigger was on ME1A cfeb4 only	30	Y
22	CLCT: Pre-trigger was on ME1B cfebs0-3 only	30	Y
23	CLCT: Pretrig discarded, no wrbuf available, buffer stalled	30	Y
24	CLCT: Pretrig discarded, no alct in window	30	
25	CLCT: CLCT discarded, clct0 had invalid pattern after drift	30	Y
26	CLCT: CLCT0 passed hit thresh but failed pid thresh after drift	30	Y
27	CLCT: CLCT1 passed hit thresh but failed pid thresh after drift	30	Y
28	CLCT: Bx pre-trigger machine waited for triads to dissipate before rearm	30	Y
29	CLCT: clct0 sent to TMB matching section	30	Y
30	CLCT: clct1 sent to TMB matching section	30	Y
31	TMB: TMB matching accepted a match, alct-only, or clct-only event	30	Y
32	TMB: CLCT*ALCT matched trigger	30	Y
33	TMB: ALCT-only trigger	30	Y
34	TMB: CLCT-only trigger	30	Y
35	TMB: TMB matching rejected event	30	Y
36	TMB: TMB matching rejected event, but queued for non-trigger readout	30	Y
37	TMB: TMB matching discarded an ALCT pair	30	Y
38	TMB: TMB matching discarded a CLCT pair	30	Y
39	TMB: TMB matching discarded CLCT0 from ME1A	30	Y
40	TMB: TMB matching discarded CLCT1 from ME1A	30	Y
41	TMB: Matching found no ALCT	30	Y
42	TMB: Matching found no CLCT	30	Y
43	TMB: Matching found One ALCT	30	Y
44	TMB: Matching found One CLCT	30	Y
45	TMB: Matching found Two ALCTs	30	Y
46	TMB: Matching found Two CLCTs	30	Y
47	TMB: ALCT0 copied into ALCT1 to make 2nd LCT	30	Y
48	TMB: CLCT0 copied into CLCT1 to make 2nd LCT	30	Y
49	TMB: LCT1 has higher quality than LCT0, error	30	Y
50	TMB: Transmitted LCT0 to MPC	30	Y
51	TMB: Transmitted LCT1 to MPC	30	Y
52	TMB: MPC accepted LCT0	30	Y
53	TMB: MPC accepted LCT1	30	Y
54	TMB: MPC rejected both LCT0 & LCT1	30	Y
55	L1A: L1A received	30	Y
56	L1A: L1A received, TMB in L1A window	30	Y
57	L1A: L1A received, no TMB in window	30	Y
58	L1A: TMB triggered, no L1A in window	30	Y

59	L1A: TMB readouts completed	30	Y
60	L1A: TMB readouts lost by 1-event-per-L1A limit	30	Y
61	STAT: CLCT Triads skipped	30	Y
62	STAT: Raw hits buffer had to be reset due to ovf, error	30	Y
63	STAT: TTC Resyncs received	30	Y
64	STAT: Sync error, bxn!=offset at bx0 arrival or no bx0 at bxn==offset	30	Y
65	STAT: Parity error in CFEB or RPC raw hits RAM, possible SEU	30	Y
	Event counters that follow are in the TMB header: They are cleared via TTC commands, such as event counter reset, and are not via direct VME command to Adr D0.		
66	HDR: Pre-trigger counter	30	N
67	HDR: CLCT counter	30	N
68	HDR: TMB trigger counter	30	N
69	HDR: ALCTs received counter	30	N
70	HDR: L1As received from ccb counter, 12 bits	12	N
71	HDR: Readout counter, 12 bits	12	N
72	HDR: Orbit counter	30	N
73	ALCT: Structure error, Expected alct0[10:1]=0 when alct0_vpf=0	8	Y
74	ALCT: Structure error, Expected alct1[10:1]=0 when alct1_vpf=0	8	Y
75	ALCT: Structure error, Expected alct0_vpf=1 when alct1_vpf=1	8	Y
76	ALCT: Structure error, Expected alct0[10:1]>0 when alct0_vpf=1	8	Y
77	ALCT: Structure error, Expected alct1[10:1]>0 when alct1_vpf=1	8	Y
78	ALCT: Structure error, Expected alct1!=alct0 when alct0_vpf=1	8	Y
79	CCB: TTCrx lock lost	8	Y
80	CCB: qPLL lock lost	8	Y
81	GTX: Optical receiver error gtx_rx_err_count0	16	N
82	GTX: Optical receiver error gtx_rx_err_count1	16	N
83	GTX: Optical receiver error gtx_rx_err_count2	16	N
84	GTX: Optical receiver error gtx_rx_err_count3	16	N
85	GTX: Optical receiver error gtx_rx_err_count4	16	N
86	GTX: Optical receiver error gtx_rx_err_count5	16	N
87	GTX: Optical receiver error gtx_rx_err_count6	16	N

Adr D2 ADR_CNT_RDATA Status Counter Data [See Adr D0 p59]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
rdata 15	rdata 14	rdata 13	rdata 12	rdata 11	rdata 10	rdata 9	rdata 8	rdata 7	rdata 6	rdata 5	rdata 4	rdata 3	rdata 2	rdata 1	rdata 0

Bits	Dir	Signal	Default	Description
[15:00]	R	cnt_rdata[15:0]	-	Data for selected counter (see adr D0 p59)

Adr D4 ADR_JTAGSM0 JTAG State Machine Control (reads JTAG PROM)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
throt 3	throt 2	throt 1	throt 0	dis	jsm jtag oe	jsm ok	vme ready	jsm tck ok	jsm wdcnt ok	jsm cksum ok	jsm abort	jsm busy	jsmsel / vsm jtag auto	jsm reset	jsm start

Bits	Dir	Signal	Typical	Description
[00]	RW	jsm_start	0	Manual cycle start command
[01]	RW	jsm_sreset	0	Status signal reset
[02]	RW	jsm_sel	0	1=select new JTAG format, 0=select old format
[03]	R	jsm_busy	0	State machine busy writing
[04]	R	jsm_aborted	0	State machine aborted reading PROM
[05]	R	jsm_cksum_ok	1	Check-sum matches PROM contents
[06]	R	jsm_wdcnt_ok	1	Word count matches PROM contents
[07]	R	jsm_tck_fpga_ok	1	FPGA jtag tck detected
[08]	R	vme_ready	1	TMB VME registers done loading from PROM
[09]	R	jsm_ok	1	JTAG state machine completed without errors
[10]	R	jsm_jtag_oe	0	Enable jtag drivers else tri-state
[11]	RW	wr_usr_jtag_dis	0	1=disable write access to ADR_USR_JTAG adr10
[15:12]	RW	jsm_throttle[3:0]	0	JTAGspeed, 0=fastest,20MHz read,10MHz TCK

Adr D6 ADR_JTAGSM1 JTAG State Machine Word Count

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
wdcnt 15	wdcnt 14	wdcnt 13	wdcnt 12	wdcnt 11	wdcnt 10	wdcnt 9	wdcnt 98	wdcnt 7	wdcnt 6	wdcnt 5	wdcnt 4	wdcnt 3	wdcnt 2	wdcnt 1	wdcnt 0

Bits	Dir	Signal	Typical	Description
[15:00]	R	jsm_wdcnt[15:0];	-	JTAG PROM word-count bits [15:0]

Adr D8 ADR_JTAGSM2 JTAG State Machine Checksum

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
chain ok	header ok	end ok	tck cnt OK	tck cnt 3	tck cnt 2	tck cnt 1	tck cnt 0	cksum 7	cksum 6	cksum 5	cksum 4	cksum 3	cksum 2	cksum 1	cksum 0

Bits	Dir	Signal	Typical	Description
[07:00]	R	jsm_cksum[7:0];	-	jtag state machine checksum
[11:08]	R	tck_fpga_cnt[3:0]	-	fpga jtag chain tck counter
[12]	R	jsm_tckcnt_ok	1	Total TCKs sent maches PROM trailer tck_cnt
[13]	R	jsm_end_ok	1	jtag PROM end marker detected
[14]	R	jsm_header_ok	1	jtag PROM header marker detected
[15]	R	jsm_chain_ok	1	jtag PROM chain marker detected

Adr DA ADR_VMEM0 VME State Machine Control (reads VME PROM)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
throt 3	throt 2	throt 1	throt 0	vsm phaser auto	vsm path ok	vsm ok	vme ready	vsm jtag auto	vsm wdcnt ok	vsm cksum ok	vsm abort	vsm busy	vsm auto start	vsm sreset	vsm start

Bits	Dir	Signal	Typical	Description
[00]	RW	vsm_start	0	Manual cycle start command
[01]	RW	vsm_sreset	0	Status signal reset
[02]	R	vsm_autostart	1	Auto-start after hard-reset
[03]	R	vsm_busy	0	State machine busy writing
[04]	R	vsm_aborted	0	State machine aborted reading PROM
[05]	R	vsm_cksum_ok	1	Check-sum matches PROM contents
[06]	R	vsm_wdcnt_ok	1	Word count matches PROM contents
[07]	RW*	vsm_jtag_auto	1	JTAG SM autostart after vmesm completes
[08]	R	vme_ready	1	TMB VME registers done loading from PROM
[09]	R	vsm_ok	1	State machine completed without errors
[10]	R	vsm_path_ok	1	State machine wrote 55AAh to ADR_VMESM4
[11]	RW	vsm_phaser_auto	1	Digital phase shifter autostart after vmesm done
[15:12]	RW	vsm_throttle[3:0]	0	VME PROM-read speed control, 0=fastest

* vsm_jtag_auto should be set to 0 to enable U76 testing, otherwise jtagsm will run, and erase U76 data.

Adr DC ADR_VMESM1 VME State Machine Word Count

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
wdcnt 15	wdcnt 14	wdcnt 13	wdcnt 12	wdcnt 11	wdcnt 10	wdcnt 9	wdcnt 98	wdcnt 7	wdcnt 6	wdcnt 5	wdcnt 4	wdcnt 3	wdcnt 2	wdcnt 1	wdcnt 0

Bits	Dir	Signal	Typical	Description
[15:00]	R	vsm_wdcnt[15:0];	-	VME PROM word-count bits [15:0]

Adr DE **ADR_VMESM2** **VME State Machine Checksum**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	jtag vec 1	jtag vec 0	fmt err 4	fmt err 3	fmt err 2	fmt err 1	fmt err 0	cksum 7	cksum 6	cksum 5	cksum 4	cksum 3	cksum 2	cksum 1	cksum 0

Bits	Dir	Signal	Typical	Description
[07:00]	R	vsm_cksum[7:0];	-	VME state machine checksum
[08]	R	vsm_fmt_err[0]	0	Missing BC header-begin marker
[09]	R	vsm_fmt_err[1]	0	Missing EC header-end marker
[10]	R	vsm_fmt_err[2]	0	Missing FC data-end marker
[11]	R	vsm_fmt_err[3]	0	Missing FF prom-end marker
[12]	R	vsm_fmt_err[4]	0	Word counter overflow
[14:13]	R	jtag_sm_vec[1:0]	0	JSM JTAG signal State Machine vector
[15]	R	--	-	unassigned

Adr E0 **ADR_VMESM3** **Number of VME Addresses Written by VMESM**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	format sm 2	format sm 1	format sm 0	prom sm 3	prom sm 2	prom sm 1	prom sm 0	nvme writes 7	nvme writes 6	nvme writes 5	nvme writes 4	nvme writes 3	nvme writes 2	nvme writes 1	nvme writes 0

Bits	Dir	Signal	Typical	Description
[07:00]	R	vsm_nvme_writes[7:0]	-	Number of vme addresses written
[11:8]	R	jsm_prom_sm_vec[3:0]	-	JSM PROM State Machine state vector
[14:12]	R	jsm_format_sm_vec[2:0]	-	JSM Data Format Machine state vector
[15]	R	--	0	unassigned

Adr E2 **ADR_VMESM4** **VME State Machine Write-Data Check**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
data 15	data 14	data 13	data 12	data 11	data 10	data 9	data 8	data 7	data 6	data 5	data 4	data 3	data 2	data 1	data 0

Bits	Dir	Signal	Typical	Description
[15:00]	R	vmesm4_rd[15:0]	55AAh	vsm_path_ok=1 if vsm writes 55aa to this adr

Adr E4 ADR_DDDSM RAT 3D3444 State Machine Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
verify dly1	verify dly0	rx phase	link tmb	oe 3	oe 2	oe 1	oe o	verify ok	sm busy	auto start	serial from	serial to	adr latch	dddr clock	dddr start

Bits	Dir	Signal	Default	Description
[00]	RW	dddr_start_vme	0	Start DDDR State Machine
[01]	RW	dddr_clock	0	DDDR manual-mode clock
[02]	RW	dddr_adr_latch	1	DDDR manual-mode address latch, active low
[03]	RW	dddr_serial_in	0	Serial data to DDDR chip
[04]	R	dddr_serial_out	0	Serial data from DDDR chip
[05]	RW	dddr_auto_start	1	DDDR State Machine autostart state
[06]	R	dddr_busy	0	DDDR State Machine busy
[07]	R	dddr_verify_ok	1	DDDR data read back verified OK
[11:08]	RW	dddr_oe[3:0]	0011_2	3D3444 output enables, 1=enable
[12]	RW	dddr_linktmb	1	1=start RAT machine when starting TMB machine
[13]	RW	dddr_rxphase	1	1=use negative clock edge to latch verify data, 0=posedge
[15:14]	RW	dddr_verify_dly[1:0]	3	Delay before latching verify data

Adr E6 ADR_DDDR0 RAT 3D3444 RPC Delays, 1 step = 2ns

Bits	Dir	Signal	Default	Description
[03:00]	RW	delay_ch0[3:0]	3	RPC0 rx clock
[07:04]	RW	delay_ch1[3:0]	3	RPC1 rx clock
[11:08]	RW	delay_ch2[3:0]	0	RAT2 rx clock, not used
[15:12]	RW	delay_ch3[3:0]	0	RAT2 rx clock, not used

Adr E8 ADR_UPTIME Uptime Counter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	up time 14	up time 13	up time 12	up time 11	up time 10	up time 9	up time 8	up time 7	up time 6	up time 5	up time 4	up time 3	up time 2	up time 1	up time 0

Bits	Dir	Signal	Typical	Description
[04:00]	R	uptime[14:0]	-	Seconds since last hard-reset
[15]	R	--	0	unassigned

Adr EA **ADR_BDSTATUS** **Board Status Summary (copy of raw-hits header)**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
status 15	status 14	status 13	status 12	status 11	status 10	status 9	status 8	status 7	status 6	status 5	status 4	status 3	status 2	status 1	status 0

Bits	Dir	Signal	Typical	Description
[00]	R	bdstatus_ok	1	Voltages OK, temperature OK, prom-load OK
[01]	R	vstat_5p0V	1	Voltage Comparator +5.0V, 1=OK
[02]	R	vstat_3p3v	1	Voltage Comparator +3.3V, 1=OK
[03]	R	vstat_1p8v	1	Voltage Comparator +1.8V, 1=OK
[04]	R	vstat_1p5v	1	Voltage Comparator +1.5V, 1=OK
[05]	R	/t_crit	1	Temperature ADC Tcritical, 1=OK
[06]	R	vsm_ok	1	VME Machine ran without errors
[07]	R	vsm_aborted	0	VME State machine aborted reading PROM
[08]	R	vsm_cksum_ok	1	VME Check-sum matches PROM contents
[09]	R	vsm_wdcnt_ok	1	VME Word count matches PROM contents
[10]	R	jsm_ok	1	JTAG State machine completed without errors
[11]	R	jsm_aborted	0	JTAG State machine aborted reading PROM
[12]	R	jsm_cksum_ok	1	JTAG Check-sum matches PROM contents
[13]	R	jsm_wdcnt_ok	1	JTAG Word count matches PROM contents
[14]	R	jsm_tck_fpga_ok	1	JTAG tck loopback detected on chain adr C
[15]	R	jsm_tckcnt_ok	1	JTAG state machine TCK count matches PROM

Adr EC **ADR_BXN_CLCT** **CLCT BXN At CLCT-Pretrigger**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	clct bxn 11	clct bxn 10	clct bxn 9	clct bxn 8	clct bxn 7	clct bxn 6	clct bxn 5	clct bxn 4	clct bxn 3	clct bxn 2	clct bxn 1	clct bxn 0

Bits	Dir	Signal	Typical	Description
[11:00]	R	bxn_clct_vme[11:0]	-	CLCT BXN latched at last CLCT pretrigger
[15:12]	R	--	0	unassigned

Adr EE **ADR_BXN_ALCT** **ALCT BXN At ALCT-Valid-Pattern-Flag**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	alct bxn 4	alct bxn 3	alct bxn 2	alct bxn 1	alct bxn 0

Bits	Dir	Signal	Typical	Description
[4:00]	R	bxn_alct_vme[4:0]	-	ALCT BXN latched at last ALCT vpf
[15:5]	R	--	0	unassigned

Adr F0 ADR_LAYER_TRIG Layer-Trigger Mode

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
clct throt 7	clct throt 6	clct throt 5	clct throt 4	clct throt 3	clct throt 2	clct throt 1	clct throt 0	0	nlayrs hit 2	nlayrs hit 1	nlayrs hit 0	layer thresh 2	layer thresh 1	layer thresh 0	layer trig en

Bits	Dir	Signal	Typical	Description
[00]	RW	layer_trigger_en	0	1=Enable layer trigger mode(see adr 68 p40)
[03:01]	RW	lyr_thresh_pretrig[2:0]	4	layer-trigger threshold
[06:04]	R	nlayers_hit_vme[2:0]	--	number layers hit on last layer-trigger
[07]	RW	--	--	Unassigned
[15:08]	RW	clct_throttle[7:0]	0	CLCT Pre-trigger rate throttle

Adr F2 ADR_ISE_VERSION ISE Version

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ver 7	ver 6	ver 5	ver 4	ver 3	ver 2	ver 1	ver 0	minor 3	minor 2	minor 1	minor 0	spack 3	spack 2	spack 1	spack 0

Bits	Dir	Signal	Typical	Description
[03:00]	R	ise_version[3:0]	03h	ISE Service Pack
[07:04]	R	ise_version[7:4]	01h	ISE Minor Version
[15:08]	R	ise_version[15:8]	10h	ISE Major Version

Adr F4 ADR_TEMP0 Pattern Finder Pre-Trigger

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
adj cfeb dist 5	adj cfeb dist 4	adj cfeb dist 3	adj cfeb dist 2	adj cfeb dist 1	adj cfeb dist 0	pid thresh post 3	pid thresh post 2	pid thresh post 1	pid thresh post 0	pid thresh pretrig 3	pid thresh pretrig 2	pid thresh pretrig 1	pid thresh pretrig 0	0	clct blank

Bits	Dir	Signal	Typical	Description
[00]	RW	clct_blanking	1	1=blank non-vpf clct output [requires alct-only or l1a-notmb mode enabled to change this to 0]
[01]	RW	--	0	Unassigned
[05:02]	RW	pid_thresh_pretrig[3:0]	0	Minimum pattern ID 0x0-0xA for pre-trigger
[09:06]	RW	pid_thresh_postdrift[3:0]	0	Minimum pattern ID 0x0-0xA after drift delay
[15:10]	RW	adjcfeb_dist[5:0]	5	Distance from key on CFEBn to CFEBn+1 to set active feb flag on CFEBn+1 for DMB Setting to 5 enables hs0,1,2,3,4 and hs31,30,29,29,28,27.

Adr F6 ADR_TEMP1 CLCT Separation

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
clct sep 7	clct sep 6	clct sep 5	clct sep 4	clct sep 3	clct sep 2	clct sep 1	clct sep 0	0	clct sep ram sel_ab	clct sep ram adr 3	clct sep ram adr 2	clct sep ram adr 1	clct sep ram adr 0	clct sep ram we	clct sep src

Bits	Dir	Signal	Typical	Description
[00]	RW	clct_sep_src	1	CLCT separation source 1=vme, 0=ram
[01]	RW	clct_sep_ram_we	0	1=enable CLCT separation ram for writing
[05:02]	RW	clct_sep_ram_adr[3:0]	0	CLCT separation RAM rw address 0-F
[06]	RW	clct_sep_ram_sel_ab	0	1=read me1a RAM, 0=me1b RAM or std RAM
[07]	RW	--	--	Unassigned
[15:08]	RW	clct_sep_vme[7:0]	10	Minimum CLCT separation in key ½-strips

Adr F8 ADR_TEMP2 CLCT Separation RAM Data

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
pspan 7	pspan 6	pspan 5	pspan 4	pspan 3	pspan 2	pspan 1	pspan 0	nspan 7	nspan 6	nspan 5	nspan 4	nspan 3	nspan 2	nspan 1	nspan 0

Bits	Dir	Signal	Typical	Description
[07:00]	RW/R	clct_sep_ram_wr[7:0]	10	nspan CLCT separation RAM data, blanks keys from 1st key to keys>=key-nspan
[15:08]	RW/R	clct_sep_ram_wr[15:8]	10	pspan CLCT separation RAM data, blanks keys from 1st key to keys<=key+pspan

Adr FA ADR_PARITY SEU Parity Errors

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
perr mux 5	perr mux 4	perr mux 3	perr mux 2	perr mux 1	perr mux 0	0	0	perr reset	perr ff	perr	perr en	perr adr 3	perr adr 2	perr adr 1	perr adr 0

Bits	Dir	Signal	Typical	Description
[03:00]	RW	perr_adr[3:0]	0	Parity data bank select address
[4]	R	perr_en	1	Parity error latch enabled
[5]	R	perr	0	Parity error summary
[6]	R	perr_ff	0	Parity error summary, latched
[7]	RW	perr_reset	0	Parity error reset
[9:8]	RW	--	0	Unassigned
[15;10]	R	parity_rd_mux[5:0]	0	Parity data multiplexer, selected by perr_adr[]

```
adr 0:  parity_rd_mux <= perr_ram_ff[ 5: 0];              // R   cfeb0 rams
adr 1:  parity_rd_mux <= perr_ram_ff[11: 6];              // R   cfeb1 rams
adr 2:  parity_rd_mux <= perr_ram_ff[17:12];              // R   cfeb2 rams
adr 3:  parity_rd_mux <= perr_ram_ff[23:18];              // R   cfeb3 rams
adr 4:  parity_rd_mux <= perr_ram_ff[29:24];              // R   cfeb4 rams
adr 5:  parity_rd_mux <= perr_ram_ff[35:30];              // R   cfeb5 rams
adr 6:  parity_rd_mux <= perr_ram_ff[41:36];              // R   cfeb6 rams
adr 7:  parity_rd_mux <= {1'b0,perr_ram_ff[46:42]};       // R   rpc   rams
adr 8:  parity_rd_mux <= {4'h0,perr_ram_ff[48:47]};       // R   mini  rams
adr 9:  parity_rd_mux <= {1'b0,perr_cfeb[6:0]};           // R   cfeb parity errors
adr 10: parity_rd_mux <= {1'b0,perr_cfeb_ff[6:0]};        // R   cfeb parity errors,latched
adr 11: parity_rd_mux <= {4'h0,perr_rpc_ff,perr_rpc};     // R   rpc  parity errors,latched
adr12:  parity_rd_mux <= {4'h0,perr_mini_ff, perr_mini};  // R   mini parity errors,latched
```

Adr FC ADR_CCB_STAT1 CCB Status Register Continued from Adr 2E

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits	Dir	Signal	Typical	Description
[00]	R	ccb_ttcrx_lock_never	0	TTCrx lock never achieved
[01]	R	ccb_ttcrx_lost_ever	0	TTCrx lock was lost at least once
[02]	R	ccb_qpll_lock_never	0	QPLL lock never achieved
[03]	R	ccb_qpll_lost_ever	0	QPLL lock was lost at least once
[15:04]	R	--	0	Unassigned

Adr FE ADR_BXN_L1A CLCT BXN at L1A Arrival

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	l1a bxn 11	l1a bxn 10	l1a bxn 9	l1a bxn 8	l1a bxn 7	l1a bxn 6	l1a bxn 5	l1a bxn 4	l1a bxn 3	l1a bxn 2	l1a bxn 1	l1a bxn 0

Bits	Dir	Signal	Typical	Description
[11:00]	R	bxn_l1a_vme[11:0]	-	CLCT BXN latched at last L1A arrival
[15:12]	R	--	0	unassigned

Adr 100 ADR_L1A_LOOKBACK L1A Lookback Distance

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
l1a win pri en	inj rdata 17	inj rdata 16	inj wdata 17	inj wdata 16	l1a look back 10	l1a look back 9	l1a look back 8	l1a look back 7	l1a look back 6	l1a look back 5	l1a look back 4	l1a look back 3	l1a look back 2	l1a look back 1	l1a look back 0

Bits	Dir	Signal	Typical	Description
[10:00]	RW	l1a_lookback[10:0]	128	bx to look back from L1As wr_buf_adr for L1A-only readouts
[12:11]	RW	inj_wdata[17:16]	0	Injector RAM write data MSBs
[14:13]	R	inj_rdata[17:16]	0	Injector RAM read data MSBs
[15]	RW	l1a_win_pri_en	1	1=Limit TMB to 1 event readout per L1A

Adr 102 ADR_SEQ_DEBUG Sequencer Debug Signals

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	l1a bxn 11	l1a bxn 10	l1a bxn 9	l1a bxn 8	l1a bxn 7	l1a bxn 6	l1a bxn 5	l1a bxn 4	l1a bxn 3	l1a bxn 2	l1a bxn 1	l1a bxn 0

Bits	Dir	Signal	Typical	Description
[03:00]	RW	seqdeb_adr[3:0]	-	Sequencer signal address 0-15
[15:04]	R	seqdeb_rd_mux[11:0]	-	Multiplexed sequencer data for debugging

```
adr 0:  deb_wr_buf_adr[10:0]        Buffer write address at last pretrig
adr 1:  deb_buf_push_adr[10:0]      Queue push address at last push
adr 2:  deb_buf_pop_adr[10:0]       Queue pop  address at last pop
adr 3:  deb_buf_push_data[11:0]     Queue push data at last push
adr 4:  deb_buf_push_data[23:12]      +Queue push data at last push
adr 5:  deb_buf_push_data[31:24]      +Queue push data at last push
adr 6:  deb_buf_pop_data[11:0]      Queue pop data at last pop
adr 7:  deb_buf_pop_data[23:12]       +Queue pop data at last pop
adr 8:  deb_buf_pop_data[31:24]       +Queue pop data at last pop
```

Adr 104 ADR_ALCT_SYNC_CTRL ALCT Sync Mode Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sync pre 3	sync pre 2	sync pre 1	sync pre 0	0	0	sync 2nd err ff	sync 1st err ff	sync 2nd err	sync 1st err	sync clr	sync rand	sync dly 3	sync dly 2	sync dly 1	sync dly 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	alct_sync_rxdata_dly [3:0]	0	Sync mode delay pointer to valid data
[04]	RW	alct_sync_tx_random	0	Sync mode tmb transmits random data to alct
[05]	RW	alct_sync_clr_err	0	ALCT sync mode clear rng error FFs
[06]	R	alct_sync_1st_err	0	1st-in-time match ok, alct-to-tmb
[07]	R	alct_sync_2nd_err	0	2nd-in-time match ok, alct-to-tmb
[08]	R	alct_sync_1st_err_ff	0	1st-in-time match ok, alct-to-tmb, latched
[09]	R	alct_sync_2nd_err_ff	0	2nd-in-time match ok, alct-to-tmb, latched
[11:10]	RW	--	0	Unassigned
[15:12]	RW	alct_sync_rxdata_pre[3:0]	9	Sync mode pre-delay pointer to valid data

Adr 106 ADR_ALCT_SYNC_TXDATA_1ST ALCT Sync Mode Transmit Data 1st

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	sync 1st 9	sync 1st 8	sync 1st 7	sync 1st 6	sync 1st 5	sync 1st 4	sync 1st 3	sync 1st 2	sync 1st 1	sync 1st 0

Bits	Dir	Signal	Typical	Description
[09:00]	RW	alct_sync_txdata_1st[9:0]	-	Sync mode data to send for loopback 1st in time
[15:10]	RW	--	0	Unassigned

Adr 108 ADR_ALCT_SYNC_TXDATA_2ND ALCT Sync Mode Transmit Data 2nd

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	sync 2nd 9	sync 2nd 8	sync 2nd 7	sync 2nd 6	sync 2nd 5	sync 2nd 4	sync 2nd 3	sync 2nd 2	sync 2nd 1	sync 2nd 0

Bits	Dir	Signal	Typical	Description
[09:00]	RW	alct_sync_txdata_1st[9:0]	-	Sync mode data to send for loopback 1st in time
[15:10]	RW	--	0	Unassigned

Adr 10A ADR_SEQ_OFFSET1 Sequencer Counter Offsets Continued [from Adr076]

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	bxn offset 11	bxn offset 10	bxn offset 9	bxn offset 8	bxn offset 7	bxn offset 6	bxn offset 5	bxn offset 4	bxn offset 3	bxn offset 2	bxn offset 1	bxn offset 0

Bits	Dir	Signal	Typical	Description
[11:00]	RW	bxn_offset_l1a[11:0]	-	L1A bxn offset preset value
[15:12]	RW	--	0	Unassigned

Adr 10C ADR_MINISCOPE Internal 16 Channel Digital Miniscope

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	tbins pretrig 4	tbins pretrig 3	tbins pretrig 2	tbins pretrig 1	tbins pretrig 0	tbins 4	tbins 3	tbins 2	tbins 1	tbins 0	tbins word nsert	tbins test mode	read enable

Bits	Dir	Signal	Default	Description
[00]	RW	mini_read_enable	1	Enable Miniscope readout to DMB
[01]	RW	mini_tbins_test	0	Miniscope data=write_address, for testing
[02]	RW	mini_tbins_word	1	Insert tbins and pretrig tbins in 1st word
[07:03]	RW	fifo_tbins_mini[4:0]	22	Number Mini FIFO time bins to read out, must multiple of 2 but not of 4
[12:08]	RW	fifo_pretrig_mini[4:0]	4	Number Mini FIFO time bins before pre-trigger
[15:13]	RW	--	0	Unassigned

Adr 10E ADR_PHASER0 ALCT rxd Digital Phase Shifter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
alct rxd delay 7	alct rxd delay 6	alct rxd delay 5	alct rxd delay 4	alct rxd delay 3	alct rxd delay 2	alct rxd delay 1	alct rxd delay 0	pos neg	sm 2	sm 1	sm 0	lock	busy	reset	fire

Bits	Dir	Signal	Default	Description
[00]	RW	fire_alct_rxd	0	Set new phase, software sets then unsets
[01]	RW	reset_alct_rxd	0	Reset current phase to 32
[02]	R	phaser_busy_alct_rxd	0	Phase shifter busy
[03]	R	lock_alct_rxd	1	DCM lock status
[06:04]	R	phaser_sm_alct_rxd[2:0]	0	Phase shifter machine state vector
[07]	RW	alct_rxd_posneg	0	0=latch inter-stage on falling main clock edge 1=latch inter-stage on rising main clock edge
[15:08]	RW	alct_rxd_delay[7:0]	32	Phase delay to latch data received from ALCT approximately 0.1ns steps (clock period/256)

Adr 110 ADR_PHASER1 ALCT txd Digital Phase Shifter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
alct txd delay 7	alct txd delay 6	alct txd delay 5	alct txd delay 4	alct txd delay 3	alct txd delay 2	alct txd delay 1	alct txd delay 0	pos neg	sm 2	sm 1	sm 0	lock	busy	reset	fire

Bits	Dir	Signal	Default	Description
[00]	RW	fire_alct_txd	0	Set new phase, software sets then unsets
[01]	RW	reset_alct_txd	0	Reset current phase to 32
[02]	R	phaser_busy_alct_txd	0	Phase shifter busy
[03]	R	lock_alct_txd	1	DCM lock status
[06:04]	R	phaser_sm_alct_txd[2:0]	0	Phase shifter machine state vector
[07]	RW	alct_txd_posneg	0	0=latch inter-stage on falling main clock edge 1=latch inter-stage on rising main clock edge
[15:08]	RW	alct_txd_delay[7:0]	32	Phase delay for data transmitted to ALCT approximately 0.1ns steps (clock period/256)

Adr 112 ADR_PHASER2 CFEB0 rxd Digital Phase Shifter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeb rxd delay 7	cfeb rxd delay 6	cfeb rxd delay 5	cfeb rxd delay 4	cfeb rxd delay 3	cfeb rxd delay 2	cfeb rxd delay 1	cfeb rxd delay 0	pos neg	sm 2	sm 1	sm 0	lock	busy	reset	fire

Bits	Dir	Signal	Default	Description
[00]	RW	fire_cfeb0_rxd	0	Set new phase, software sets then unsets
[01]	RW	reset_cfeb0_rxd	0	Reset current phase to 32
[02]	R	phaser_busy_cfeb0_rxd	0	Phase shifter busy
[03]	R	lock_cfeb0_rxd	1	DCM lock status
[06:04]	R	phaser_sm_cfeb0_rxd[2:0]	0	Phase shifter machine state vector
[07]	RW	cfeb0_rxd_posneg	0	0=latch inter-stage on falling main clock edge 1=latch inter-stage on rising main clock edge
[15:08]	RW	cfeb0_rxd_delay[7:0]	32	Phase delay to latch data received from CFEB approximately 0.1ns steps (clock period/256)

Adr 114 ADR_PHASER3 CFEB1 rxd Digital Phase Shifter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeb rxd delay 7	cfeb rxd delay 6	cfeb rxd delay 5	cfeb rxd delay 4	cfeb rxd delay 3	cfeb rxd delay 2	cfeb rxd delay 1	cfeb rxd delay 0	pos neg	sm 2	sm 1	sm 0	lock	busy	reset	fire

Bits	Dir	Signal	Default	Description
[00]	RW	fire_cfeb1_rxd	0	Set new phase, software sets then unsets
[01]	RW	reset_cfeb1_rxd	0	Reset current phase to 32
[02]	R	phaser_busy_cfeb1_rxd	0	Phase shifter busy
[03]	R	lock_cfeb1_rxd	1	DCM lock status
[06:04]	R	phaser_sm_cfeb1_rxd[2:0]	0	Phase shifter machine state vector
[07]	RW	cfeb1_rxd_posneg	0	0=latch inter-stage on falling main clock edge 1=latch inter-stage on rising main clock edge
[15:08]	RW	cfeb1_rxd_delay[7:0]	32	Phase delay to latch data received from CFEB approximately 0.1ns steps (clock period/256)

Adr 116 ADR_PHASER4 CFEB2 rxd Digital Phase Shifter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeb rxd delay 7	cfeb rxd delay 6	cfeb rxd delay 5	cfeb rxd delay 4	cfeb rxd delay 3	cfeb rxd delay 2	cfeb rxd delay 1	cfeb rxd delay 0	pos neg	sm 2	sm 1	sm 0	lock	busy	reset	fire

Bits	Dir	Signal	Default	Description
[00]	RW	fire_cfeb2_rxd	0	Set new phase, software sets then unsets
[01]	RW	reset_cfeb2_rxd	0	Reset current phase to 32
[02]	R	phaser_busy_cfeb2_rxd	0	Phase shifter busy
[03]	R	lock_cfeb2_rxd	1	DCM lock status
[06:04]	R	phaser_sm_cfeb2_rxd[2:0]	0	Phase shifter machine state vector
[07]	RW	cfeb2_rxd_posneg	0	0=latch inter-stage on falling main clock edge 1=latch inter-stage on rising main clock edge
[15:08]	RW	cfeb2_rxd_delay[7:0]	32	Phase delay to latch data received from CFEB approximately 0.1ns steps (clock period/256)

Adr 118 ADR_PHASER5 CFEB3 rxd Digital Phase Shifter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeb rxd delay 7	cfeb rxd delay 6	cfeb rxd delay 5	cfeb rxd delay 4	cfeb rxd delay 3	cfeb rxd delay 2	cfeb rxd delay 1	cfeb rxd delay 0	pos neg	sm 2	sm 1	sm 0	lock	busy	reset	fire

Bits	Dir	Signal	Default	Description
[00]	RW	fire_cfeb3_rxd	0	Set new phase, software sets then unsets
[01]	RW	reset_cfeb3_rxd	0	Reset current phase to 32
[02]	R	phaser_busy_cfeb3_rxd	0	Phase shifter busy
[03]	R	lock_cfeb3_rxd	1	DCM lock status
[06:04]	R	phaser_sm_cfeb3_rxd[2:0]	0	Phase shifter machine state
[07]	RW	cfeb3_rxd_posneg	0	0=latch inter-stage on falling main clock edge 1=latch inter-stage on rising main clock edge
[15:08]	RW	cfeb3_rxd_delay[7:0]	32	Phase delay to latch data received from CFEB approximately 0.1ns steps (clock period/256)

Adr 11A ADR_PHASER6 CFEB4 rxd Digital Phase Shifter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeb rxd delay 7	cfeb rxd delay 6	cfeb rxd delay 5	cfeb rxd delay 4	cfeb rxd delay 3	cfeb rxd delay 2	cfeb rxd delay 1	cfeb rxd delay 0	pos neg	sm 2	sm 1	sm 0	lock	busy	reset	fire

Bits	Dir	Signal	Default	Description
[00]	RW	fire_cfeb4_rxd	0	Set new phase, software sets then unsets
[01]	RW	reset_cfeb4_rxd	0	Reset current phase to 32
[02]	R	phaser_busy_cfeb4_rxd	0	Phase shifter busy
[03]	R	lock_cfeb4_rxd	1	DCM lock status
[06:04]	R	phaser_sm_cfeb4_rxd[2:0]	0	Phase shifter machine state vector
[07]	RW	cfeb4_rxd_posneg	0	0=latch inter-stage on falling main clock edge 1=latch inter-stage on rising main clock edge
[15:08]	RW	cfeb4_rxd_delay[7:0]	32	Phase delay to latch data received from CFEB approximately 0.1ns steps (clock period/256)

Adr 11C ADR_DELAY0_INT CFEB DDR RxD Interstage Delays

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeb3 delay 3	cfeb3 delay 2	cfeb3 delay 1	cfeb3 delay 0	cfeb2 delay 3	cfeb2 delay 2	cfeb2 delay 1	cfeb2 delay 0	cfeb1 delay 3	cfeb1 delay 2	cfeb1 delay 1	cfeb1 delay 0	cfeb0 delay 3	cfeb0 delay 2	cfeb0 delay 1	cfeb0 delay 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	cfeb0_rxd_int_delay[3:0]	0	Delay data received from CFEB0 by integer bx
[07:04]	RW	cfeb1_rxd_int_delay [3:0]	0	Delay data received from CFEB1 by integer bx
[11:08]	RW	cfeb2_rxd_int_delay [3:0]	0	Delay data received from CFEB2 by integer bx
[15:12]	RW	cfeb3_rxd_int_delay [3:0]	0	Delay data received from CFEB3 by integer bx

Adr 11E ADR_DELAY1_INT CFEB DDR RxD Interstage Delays Continued

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	cfeb6 delay 3	cfeb6 delay 2	cfeb6 delay 1	cfeb6 delay 0	cfeb5 delay 3	cfeb5 delay 2	cfeb5 delay 1	cfeb5 delay 0	cfeb4 delay 3	cfeb4 delay 2	cfeb4 delay 1	cfeb4 delay 0

Bits	Dir	Signal	Default	Description
[03:00]	RW	cfeb4_rxd_int_delay 3:0]	0	Delay data received from CFEB4 by integer bx
[07:04]	RW	cfeb5_rxd_int_delay 3:0]	0	Delay data received from CFEB5 by integer bx
[11:08]	RW	cfeb6_rxd_int_delay 3:0]	0	Delay data received from CFEB6 by integer bx
[15:12]	RW	--	0	Unassigned

Adr 120 ADR_SYNC_ERR_CTRL Synchronization Error Control

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sync err forced	clock lock err	bx0 match err	alct ecc tx err	alct ecc rx err	clct bx0 err	sync err	err stops L1As	err stops pretrig	err blanks LCTs	clock lock lost en	bx0 match en	alct tx ecc en	alct rx ecc en	clct bx0 en	sync err reset

Bits	Dir	Signal	Default	Description
[00]	RW	sync_err_reset	0	VME sync error reset
Sync error source enables:				
[01]	RW	clct_bx0_sync_err_en	1	TMB clock pulse count err: bxn!=0+offset at ttc_bx0 arrival
[02]	RW	alct_ecc_rx_err_en	0	ALCT uncorrected ECC error in data ALCT received from TMB
[03]	RW	alct_ecc_tx_err_en	0	ALCT uncorrected ECC error in data ALCT transmitted to TMB
[04]	RW	bx0_match_err_en	0	ALCT alct_bx0 != clct_bx0 in LCT to MPC
[05]	RW	clock_lock_lost_err_en	0	40MHz main clock lost lock
Sync error action enables:				
[06]	RW	sync_err_blanks_mpc_en	0	Sync error blanks LCTs to MPC
[07]	RW	sync_err_stops_pretrig_en	0	Sync error stops CLCT pre-triggers
[08]	RW	sync_err_stops_readout_en	0	Sync error stops L1A readouts
Sync error types latched for VME readout:				
[09]	R	sync_err	0	Sync error OR of enabled types of error
[10]	R	clct_bx0_sync_err_ff	0	TMB clock pulse count err: bxn!=0+offset at ttc_bx0 arrival
[11]	R	alct_ecc_rx_err_ff	0	ALCT uncorrected ECC error in data ALCT received from TMB
[12]	R	alct_ecc_tx_err_ff	0	ALCT uncorrected ECC error in data ALCT transmitted to TMB
[13]	R	bx0_match_err_ff	0	ALCT alct_bx0 != clct_bx0 in LCT to MPC
[14]		clock_lock_lost_err_ff	0	40MHz main clock lost lock
[15]	RW	sync_err_forced	0	Force sync_err=1

See p45 Adr86[1:0] for tmb_sync_err_en[1:0] Allow sync_err to MPC for either muon
See p35 Adr38[2] for alct_ecc_err_blank Blank alct muons having uncorrected ecc errors

Adr 122 ADR_CFEB_BADBITS_CTRL CFEB Bad Bits Control/Status (see 0x15C for V6)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeb blocked	found cfeb4	found cfeb3	found cfeb2	found cfeb1	found cfeb0	block cfeb4	block cfeb3	block cfeb2	block cfeb1	block cfeb0	reset cfeb4	reset cfeb3	reset cfeb2	reset cfeb1	reset cfeb0

Bits	Dir	Signal	Default	Description
[04:00]	RW	cfeb_badbits_reset[4:0]	0	0x1F=Reset bad cfeb bits FFs for cfeb[n]
[09:05]	RW	cfeb_badbits_block[4:0]	0	0x1F=Block bad cfeb bits in cfeb[n]
[14:10]	R	cfeb_badbits_found[4:0]	0	CFEB[n] has at least 1 bad bit
[15]	R	cfeb_badbits_blocked	0	At least one CFEB has a bad bit that was blocked

Adr 124 ADR_CFEB_BADBITS_TIMER CFEB Bad Bits Check Interval

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nbx 15	nbx 14	nbx 13	nbx 12	nbx 11	nbx 10	nbx 9	nbx 8	nbx 7	nbx 6	nbx 5	nbx 4	nbx 3	nbx 2	nbx 1	nbx 0

Bits	Dir	Signal	Default	Description
[15:00]	RW	cfeb_badbits_nbx [15:0]	3564	Check Interval for CFEB bad bits, bx units

Adr 126 ADR_CFEB0_BADBITS_LY01 CFEB0 Ly0,Ly1 Bad Bits List

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ly1 distrip 7	ly1 distrip 6	ly1 distrip 5	ly1 distrip 4	ly1 distrip 3	ly1 distrip 2	ly1 distrip 1	ly1 distrip 0	ly0 distrip 7	ly0 distrip 6	ly0 distrip 5	ly0 distrip 4	ly0 distrip 3	ly0 distrip 2	ly0 distrip 1	ly0 distrip 0

Bits	Dir	Signal	Default	Description
[07:00]	R	cfeb0_ly0_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb0_ly1_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 128 ADR_CFEB0_BADBITS_LY23 CFEB0 Ly2,Ly3 Bad Bits List

[07:00]	R	cfeb0_ly2_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb0_ly3_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 12A ADR_CFEB0_BADBITS_LY45 CFEB0 Ly4,Ly5 Bad Bits List

[07:00]	R	cfeb0_ly4_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb0_ly5_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 12C ADR_CFEB1_BADBITS_LY01 CFEB1 Ly0,Ly1 Bad Bits List

[07:00]	R	cfeb1_ly0_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb1_ly1_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 12E ADR_CFEB1_BADBITS_LY23 CFEB1 Ly2,Ly3 Bad Bits List

[07:00]	R	cfeb1_ly2_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb1_ly3_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 130 ADR_CFEB1_BADBITS_LY45 CFEB1 Ly4,Ly5 Bad Bits List

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ly1 distrip 7	ly1 distrip 6	ly1 distrip 5	ly1 distrip 4	ly1 distrip 3	ly1 distrip 2	ly1 distrip 1	ly1 distrip 0	ly0 distrip 7	ly0 distrip 6	ly0 distrip 5	ly0 distrip 4	ly0 distrip 3	ly0 distrip 2	ly0 distrip 1	ly0 distrip 0

Bits	Dir	Signal	Default	Description
[07:00]	R	cfeb1_ly4_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb1_ly5_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 132 ADR_CFEB2_BADBITS_LY01 CFEB2 Ly0,Ly1 Bad Bits List

[07:00]	R	cfeb2_ly0_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb2_ly1_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 134 ADR_CFEB2_BADBITS_LY23 CFEB2 Ly2,Ly3 Bad Bits List

[07:00]	R	cfeb2_ly2_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb2_ly3_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 136 ADR_CFEB2_BADBITS_LY45 CFEB2 Ly4,Ly5 Bad Bits List

[07:00]	R	cfeb2_ly4_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb2_ly5_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 138 ADR_CFEB3_BADBITS_LY01 CFEB3 Ly0,Ly1 Bad Bits List

[07:00]	R	cfeb3_ly0_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb3_ly1_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 13A ADR_CFEB3_BADBITS_LY23 CFEB3 Ly2,Ly3 Bad Bits List

[07:00]	R	cfeb3_ly2_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb3_ly3_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 13C ADR_CFEB3_BADBITS_LY45 CFEB3 Ly4,Ly5 Bad Bits List

[07:00]	R	cfeb3_ly4_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb3_ly5_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 13E ADR_CFEB4_BADBITS_LY01 CFEB4 Ly0,Ly1 Bad Bits List

[07:00]	R	cfeb4_ly0_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb4_ly1_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 140 ADR_CFEB4_BADBITS_LY23 CFEB4 Ly2,Ly3 Bad Bits List

[07:00]	R	cfeb4_ly2_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb4_ly3_badbits [7:0]	00000000_2	1=1=CFEB rx bit[n] went bad

Adr 142 ADR_CFEB4_BADBITS_LY45 CFEB4 Ly4,Ly5 Bad Bits List

[07:00]	R	cfeb4_ly4_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb4_ly5_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 144 ADR_ALCT_STARTUP_DELAY ALCT startup delay milliseconds for Spartan-6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
d[15]	d[14]	d[13]	d[12]	d[11]	d[10]	d[9]	d[8]	d[7]	d[6]	d[5]	d[4]	d[3]	d[2]	d[1]	d[0]

Bits	Dir	Signal	Default	Description
[15:00]	RW	alct_startup_delay [15:0]	116	Msec to wait after TMB powers up before Initializing DDD delays and ALCT JTAG. This setting is only used after TMB first powers up or has a hard reset, so changes need to be stored in VME PROM. ALCT Spartan-6 takes 212msec to configure. TMB Virtex-2 takes 100msec. This register holds the number of msec to wait after TMB configures, so a value of 116 corresponds to 100ms + 116ms =216ms after a TMB hard reset.

Adr 146 ADR_ALCT_STARTUP_STATUS ALCT startup delay machine status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	alct start done	alct wait cfg	alct wait vme	alct wait dll	alct start msec	vsm ready	power up	global reset

Bits	Dir	Signal	Typical	Description
[00]	R	global_reset	0	Global reset
[01]	R	power_up	1	DLL clock locked, we wait for it
[02]	R	vsm_ready	1	Injector RAM read data MSBs
[03]	R	alct_startup_msec	0	Startup machine millisecond pulse, width=25ns
[04]	R	alct_wait_dll	0	Startup machine waiting for TMB DLL lock
[05]	R	alct_wait_vme	0	Startup machine waiting for TMB VME user PROM
[06]	R	alct_wait_cfg	0	Startup machine waiting for ALCT FPGA to config
[07]	R	alct_startup_done	1	Startup machine done ALCT FPGA assumed configured
[15:08]	R	--	0	Unassigned

Adr 148 ADR_V6_SNAP12_QPLL Virtex-6 SNAP12 Serial interface + QPLL status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	r12 fok	r12 sdat	r12 sclk	0	qpll err	qpll lock	qpll nrst

Bits	Dir	Signal	Typical	Description
[00]	RW	qpll_nrst	1	nReset QPLL, 0=reset
[01]	R	qpll_lock	0	QPLL locked status
[02]	R	qpll_err	0	QPLL error status
[03]	RW	--	0	Unassigned
[04]	R	r12_sclk	1	SNAP12 Serial interface clock, drive high
[05]	R	r12_sdat	0	SNAP12 Serial interface data
[06]	R	r12_fok	0	SNAP12 Serial interface status
[15:07]	R	--	0	Unassigned

Adr 14A ADR_V6_GTX_RX_ALL Virtex-6 master GTX control and status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
gtx err sum bit 7	gtx err sum bit 6	gtx err sum bit 5	gtx err sum bit 4	gtx err sum bit 3	gtx err sum bit 2	gtx err sum bit 1	gtx err sum bit 0	gtx pol swap	gtx link bad	gtx linkha d err	gtx link good	gtx sync done	gtx en prbs	gtx reset	gtx en

Bits	Dir	Signal	Typical	Description
[00]	RW	gtx_rx_enable_all	1	Enable all GTX optical inputs, disables copper CFEBs
[01]	RW	gtx_rx_reset_all	0	Reset all GTX
[02]	RW	gtx_rx_en_prbs_test_all	0	Select all GTX for PRBS test input mode
[03]	R	&gtx_rx_sync_done[6:0]	1	All GTX are ready
[04]	R	&gtx_link_good[6:0]	1	All GTX links are locked (over 15 BX with clean structure)
[05]	R	|gtx_link_had_err[6:0]	0	At least one GTX link had an error since last reset
[06]	R	|gtx_link_bad[6:0]	0	At least one GTX link had over 100 errors
[07]	R	|gtx_rx_pol_swap[6:0]	0	GTX 5,6 [ie dcfeb 4,5] have swapped rx board routes
[15:8]	R	gtx_rx_err_count_sum_all	0	Sum of GTX link error counts (full scale count is hex FE)

Adr 14C-158 ADR_V6_GTX_RX0-6 Virtex-6 individual GTX (idcfeb[6:0]) control and status

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
gtx err count bit 7	gtx err count bit 6	gtx err count bit 5	gtx err count bit 4	gtx err count bit 3	gtx err count bit 2	gtx err count bit 1	gtx err count bit 0	gtx pol swap	gtx link bad	gtx linkha d err	gtx link good	gtx sync done	gtx en prbs	gtx reset	gtx en

Bits	Dir	Signal	Typical	Description
[00]	RW	gtx_rx_enable[idcfeb]	1	Enable this GTX optical input, disables copper input
[01]	RW	gtx_rx_reset[idcfeb]	0	Reset this GTX
[02]	RW	gtx_rx_en_prbs_test[idcfeb]	0	Select this GTX for PRBS test input mode
[03]	R	gtx_rx_sync_done[idcfeb]	1	GTX ready
[04]	R	gtx_link_good	1	GTX link is locked (over 15 BX with clean data frames)
[05]	R	gtx_link_had_err	0	GTX link had an error (bad data frame) since last reset
[06]	R	gtx_link_bad	0	GTX link had over 100 errors since last reset
[07]	R	gtx_rx_pol_swap[idcfeb]	0	GTX 5,6 [ie dcfeb 4,5] have swapped rx board routes
[15:8]	R	gtx_rx_err_count[idcfeb]	0	GTX link error count (full scale count is hex E0)

Adr 15A ADR_V6_SYSMON Virtex-6 Sysmon ADC

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	gtx err	gtx pol swal	gtx sync done	gtx match	gtx valid	gtx fc	gtx start	gtx en prbs	gtx reset err cnt	gtx reset	gtx en

Bits	Dir	Signal	Typical	Description
[04:00]	RW	adc_adr[4:0]	0	ADC channel
[05]	R	adc_valid	0	ADC RAM has valid data for this adc_adr, readonly
[05]	W	adc_reset	0	Reset Sysmon module, writeonly
[15:6]	R	adc_data[15:6]	0	ADC counts for this adc_adr

Virtex-6 Sysmon ADC Channel Assignments

adr	Source	Units	Conversion Factor
0	Temperature	Degrees C	= (ADC code × 503.975)/1024 – 273.15
1	VccINT	Volts	= (ADC Code / 1024) x 3V
2	VccAUX	Volts	= (ADC Code / 1024) x 3V
4	Vref 1.25V	Volts	= (ADC Code / 1024) x 3V
5	Vzero 0.00V	Volts	= (ADC Code / 1024) x 3V

Adr 15C ADR_V6_CFEB_BADBITS_CTRL CFEB Bad Bits Control/Status (See Adr 0x122)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	found cfeb6	found cfeb5	block cfeb6	block cfeb5	reset cfeb6	reset cfeb5

Bits	Dir	Signal	Default	Description
[01:00]	RW	cfeb_badbits_reset[6:5]	0	0x1F=Reset bad cfeb bits FFs for cfeb[n]
[03:02]	RW	cfeb_badbits_block[6:5]	0	0x1F=Block bad cfeb bits in cfeb[n]
[05:04]	R	cfeb_badbits_found[6:5]	0	CFEB[n] has at least 1 bad bit
[15:06]	RW	--	0	Unassigned

Adr 15E ADR_ V6_CFEB5_BADBITS_LY01 CFEB5 Ly0,Ly1 Bad Bits List

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ly1 distrip 7	ly1 distrip 6	ly1 distrip 5	ly1 distrip 4	ly1 distrip 3	ly1 distrip 2	ly1 distrip 1	ly1 distrip 0	ly0 distrip 7	ly0 distrip 6	ly0 distrip 5	ly0 distrip 4	ly0 distrip 3	ly0 distrip 2	ly0 distrip 1	ly0 distrip 0

Bits	Dir	Signal	Default	Description
[07:00]	R	cfeb5_ly0_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb5_ly1_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 160 ADR_ V6_CFEB5_BADBITS_LY23 CFEB5 Ly2,Ly3 Bad Bits List

[07:00]	R	cfeb5_ly2_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb5_ly3_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 162 ADR_ V6_CFEB5_BADBITS_LY45 CFEB5 Ly4,Ly5 Bad Bits List

[07:00]	R	cfeb5_ly4_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb5_ly5_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 164 ADR_ V6_CFEB6_BADBITS_LY01 CFEB6 Ly0,Ly1 Bad Bits List

[07:00]	R	cfeb6_ly0_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb6_ly1_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 166 ADR_ V6_CFEB6_BADBITS_LY23 CFEB6 Ly2,Ly3 Bad Bits List

[07:00]	R	cfeb6_ly2_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb6_ly3_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 168 ADR_ V6_CFEB6_BADBITS_LY45 CFEB6 Ly4,Ly5 Bad Bits List

[07:00]	R	cfeb6_ly4_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad
[15:08]	R	cfeb6_ly5_badbits [7:0]	00000000_2	1=CFEB rx bit[n] went bad

Adr 16A ADR_V6_PHASER7 CFEB5 rxd Digital Phase Shifter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeb rxd delay 7	cfeb rxd delay 6	cfeb rxd delay 5	cfeb rxd delay 4	cfeb rxd delay 3	cfeb rxd delay 2	cfeb rxd delay 1	cfeb rxd delay 0	pos neg	sm 2	sm 1	sm 0	lock	busy	reset	fire

Bits	Dir	Signal	Default	Description
[00]	RW	fire_cfeb0_rxd	0	Set new phase, software sets then unsets
[01]	RW	reset_cfeb0_rxd	0	Reset current phase to 32
[02]	R	phaser_busy_cfeb0_rxd	0	Phase shifter busy
[03]	R	lock_cfeb0_rxd	1	DCM lock status
[06:04]	R	phaser_sm_cfeb0_rxd[2:0]	0	Phase shifter machine state vector
[07]	RW	cfeb0_rxd_posneg	0	0=latch inter-stage on falling main clock edge 1=latch inter-stage on rising main clock edge
[15:08]	RW	cfeb0_rxd_delay[7:0]	32	Phase delay to latch data received from CFEB approximately 0.1ns steps (clock period/256)

Adr 16C ADR_V6_PHASER8 CFEB6 rxd Digital Phase Shifter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
cfeb rxd delay 7	cfeb rxd delay 6	cfeb rxd delay 5	cfeb rxd delay 4	cfeb rxd delay 3	cfeb rxd delay 2	cfeb rxd delay 1	cfeb rxd delay 0	pos neg	sm 2	sm 1	sm 0	lock	busy	reset	fire

Bits	Dir	Signal	Default	Description
[00]	RW	fire_cfeb0_rxd	0	Set new phase, software sets then unsets
[01]	RW	reset_cfeb0_rxd	0	Reset current phase to 32
[02]	R	phaser_busy_cfeb0_rxd	0	Phase shifter busy
[03]	R	lock_cfeb0_rxd	1	DCM lock status
[06:04]	R	phaser_sm_cfeb0_rxd[2:0]	0	Phase shifter machine state vector
[07]	RW	cfeb0_rxd_posneg	0	0=latch inter-stage on falling main clock edge 1=latch inter-stage on rising main clock edge
[15:08]	RW	cfeb0_rxd_delay[7:0]	32	Phase delay to latch data received from CFEB approximately 0.1ns steps (clock period/256)

Adr 16E ADR_V6_HCM501 CFEB5 Ly0,Ly1 Hot Channel Mask

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ly1 distrip 7	ly1 distrip 6	ly1 distrip 5	ly1 distrip 4	ly1 distrip 3	ly1 distrip 2	ly1 distrip 1	ly1 distrip 0	ly0 distrip 7	ly0 distrip 6	ly0 distrip 5	ly0 distrip 4	ly0 distrip 3	ly0 distrip 2	ly0 distrip 1	ly0 distrip 0

Bits	Dir	Signal	Default	Description
[07:00]	RW	cfeb5_ly0_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 0
[15:08]	RW	cfeb5_ly1_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 1

Adr 170 ADR_V6_HCM523 CFEB5 Ly2,Ly3 Hot Channel Mask

[07:00]	RW	cfeb5_ly2_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 2
[15:08]	RW	cfeb5_ly3_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 3

Adr 172 ADR_V6_HCM545 CFEB5 Ly4,Ly5 Hot Channel Mask

[07:00]	RW	cfeb5_ly4_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 4
[15:08]	RW	cfeb5_ly5_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 5

Adr 174 ADR_V6_HCM601 CFEB6 Ly0,Ly1 Hot Channel Mask

[07:00]	RW	cfeb6_ly0_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 0
[15:08]	RW	cfeb6_ly1_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 1

Adr 176 ADR_V6_HCM623 CFEB6 Ly2,Ly3 Hot Channel Mask

[07:00]	RW	cfeb6_ly2_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 2
[15:08]	RW	cfeb6_ly3_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 3

Adr 178 ADR_V6_HCM645 CFEB6 Ly4,Ly5 Hot Channel Mask

[07:00]	RW	cfeb6_ly4_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 4
[15:08]	RW	cfeb6_ly5_hcm[7:0]	11111111_2	1=Enable DiStrip[7:0] Layer 5

Adr 17A ADR_V6_EXTEND DCFEB 7-bit extensions to 5 bit fields in 0x42, 0x68

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	cfebe n 6 read	cfebe n 5 read	cfebe n 6 vme	cfebe n 5 vme	inj mask 6	inj mask 5	inj febsel 6	inj febsel 5	mask all cfeb6	mask all cfeb5

Bits	Dir	Signal	Default	Description
[01:00]	RW	mask_all[6:5]	11_2	Extend 0x42[4:0] = mask_all[4:0]
[03:02]	RW	inj_febsel[6:5]	0	Extend 0x42[9:5] = inj_febsel[4:0]
[05:04]	RW	injector_mask[6:5]	11_2	Extend 0x42[14:10] = injector_mask_cfeb[4:0]
[07:06]	RW	cfeb_en_vme[6:5]	11_2	Extend 0x68[14:10] = cfeb_en_vme[4:0]
[09:08]	R	cfeb_en[6:5]	11_2	Extend 0x68[14:10] = cfeb_en[4:0] readback
[15:10]	RW	--	0	Unassigned

TTC Commands:

Fast Control Bus ccb_cmd[5..0] Decoding Scheme

Signal	Code (hex)	Decoded by TMB	Description
BX0 (*)	01	Y	Bunch Crossing Zero
L1 Reset (*)	03	Y	Reset L1 readout buffers and resynchronize optical links
Hard_reset (*)	04		Reload all FPGAs from EPROMs
Start Trigger	06	Y	Go to trigger run , wait for bx0, may be disabled by Adr 2C[7]
Stop Trigger	07	Y	Go to stop state, wait for bx0, may be disabled by Adr 2C[7]
Test Enable	08		
Private Gap	09		
Private Orbit	0A		
Tmb_hard_reset (*)	10		Reload TMB FPGAs from EPROM
Alct_hard_reset (*)	11		Reload ALCT FPGAs from EPROM
Dmb_hard_reset (*)	12		Reload DMB FPGAs from EPROM
Mpc_hard_reset (*)	13		Reload MPC FPGAs from EPROM
Dmb_cfeb_calibrate0 (*)	14		CFEB Calibrate Pre-Amp Gain
Dmb_cfeb_calibrate1 (*)	15		CFEB Trigger Pattern Calibration
Dmb_cfeb_calibrate2 (*)	16		CFEB Pedestal Calibration
Dmb_cfeb_initiate (*)	17		Initiate CFEB calibration (Hold next L1ACC and Pretriggers)
Alct_adb_pulse_sync (*)	18		Pulse Anode Discriminator, synchronous
Alct_adb_pulse_async (*)	19		Pulse Anode Discriminator, asynchronous
Clct_external_trigger (*)	1A		External Trigger All CLCTs
Alct_external_trigger (*)	1B		External Trigger All ALCTs
Soft_reset (*)	1C		Initializes the FPGA on DMB, TMB and MPC boards
DMB_soft_reset (*)	1D		Initializes the FPGA on a DMB
TMB_soft_reset (*)	1E		Initializes the FPGA on a TMB
MPC_soft_reset (*)	1F		Initializes the FPGA on a MPC
Send_bcnt[7..0] (*)	20		Send Bunch_Counter[7..0] to ccb_data[7..0] bus
Send_evcnt[7..0] (*)	21		Send Event_Counter[7..0] to ccb_data[7..0] bus
Send_evcnt[15..8] (*)	22		Send Event_Counter[15..8] to ccb_data[7..0] bus
Send_evcnt[23..16] (*)	23		Send Event_Counter[23..16] to ccb_data[7..0] bus
Inject patterns from TMBs	24	Y	Injects patterns from TMB's internal RAM to MPC
Alct_adb_pulse (*)	25		Generate sync and async anode discriminator pulses
Inject patterns from MPCs	30		Injects patterns from MPC's input FIFO to SP
Inject patterns from MS	31		Injects patterns from MS input FIFO to Global Muon Trigger
tmb_bxreset	32	Y	Reset TMB/ALCT BXN, do not reset L1A counters

(*) – decoded by CCB

TMB Board Status Operations

()

ID Registers

id_slot=15	VME Slot
id_rev =D	Firmware version
id_type=C	Firmware Type
id_date=06/08/2004	Firmware Compile Date
id_rev =38CA=06/10/04 xc2v3000	Firmware Revcode

Digital Serial Numbers

```
Digital Serial for TMB CRC=DC DSN=00000A237E7F MFG=01 OK
Digital Serial for Mez CRC=BF DSN=000007E06194 MFG=01 OK
Digital Serial for RAT CRC=52 DSN=00000AB39AAD MFG=01 OK
```

Power Supply ADC

```
TMB2005E Comparators
5.0V  status=OK
3.3V  status=OK
1.8V  status=OK
1.5V  status=OK
Tcrit status=OK

TMB2005E ADC
+5.0 TMB      5.004 V     0.305 A
+3.3 TMB      3.221 V     1.160 A
+1.5 TMBcore  1.488 V     0.795 A
+1.5 GTLtt    1.492 V     0.230 A
+1.0 GTLref   1.004 V     0.000 A
+3.3 RAT      3.221 V     0.250 A
+1.8 RATcore  1.797 V     8.985 A
+vref/2       2.047 V     0.000 A
+vzero        0.000 V     0.000 A
+vref         4.095 V     0.000 A

TMB2005E Temperature IC
T tmb pcb    73.4   F    23.    C  Tcrit=261./127.
T tmb fpga   95.0   F    35.    C  Tcrit=261./127.

RAT2005E Temperature IC
T rat pcb    68.0   F    20.    C  Tcrit=261./127.
T rat xstr   69.8   F    21.    C  Tcrit=261./127.
```

Clock Delays

```
Current 3D3444 Delay Settings 02/27/2006
     Ch0   8steps  16ns  ALCT  tx clock   alct_tof_delay in muonic firmware versions
     Ch1   1steps   2ns  ALCT  rx clock   not used in muonic firmware versions
     Ch2   2steps   4ns  DMB   tx clock
     Ch3   9steps  20ns  RPC   tx clock
     Ch4   0steps   0ns  TMB1  rx clock   not used in muonic firmware versions
     Ch5   0steps   0ns  MPC   rx clock
     Ch6   0steps   0ns  DCC   tx clock   cfeb_tof_delay in muonic firmware versions
     Ch7   7steps  14ns  CFEB0 tx clock
     Ch8   7steps  14ns  CFEB1 tx clock
     Ch9   7steps  14ns  CFEB2 tx clock
     ChA   7steps  14ns  CFEB3 tx clock
     ChB   7steps  14ns  CFEB4 tx clock
```

JTAG Chains

```
Chain Select Address (X=don't care)
3210  Base  Function
00SS  0     ALCT: SS=00(Slow user) SS=01(Slow prom) SS=10(Mez user) SS=11(Mez prom)
01XX  4     TMB Mezzanine FPGA+PROMs
10XX  8     TMB User PROMs
1100  C     FPGA Monitor (for TMB self-test)
1101  D     RAT Module FPGA+PROM
```

RAT Module Status Register USER1

```
RAT FPGA device 0 Idcode=  20A10093
RAT PROM device 1 Idcode=  05024093
RAT FPGA device 0 USERcode=02232006
RAT PROM device 1 USERcode=02232006
RAT USER1=E00000007FFFFFFFE02300336205650565E400CCC989C20060223EB
rs_begin          B
rs_version        E
rs_monthday       0223
rs_year           2006
rs_syncmode       0
rs_posneg         0
rs_loop           1
rs_rpc_en         3
rs_clk_active     0
rs_locked_tmb     1
rs_locked_rpc0    0
rs_locked_rpc1    0
rs_locklost_tmb   0
rs_locklost_rpc0  1
rs_locklost_rpc1  1
rs_txok           0
rs_rxok           0
rs_ntcrit         1
rs_rpc_free       0
rs_dsn            0
rs_dddoe_wr       3
rs_ddd_wr         0033
rs_ddd_auto       1
rs_ddd_start      0
rs_ddd_busy       0
rs_ddd_verify_ok  1
rs_rpc0_parity_ok 1
rs_rpc1_parity_ok 1
rs_rpc0_cnt_perr  0565
rs_rpc1_cnt_perr  0565
rs_last_opcode    02
rw_rpc_en         3
rw_ddd_start      0
rw_ddd_wr         0033
rw_dddoe_wr       3
rw_perr_reset     0
rw_parity_odd     1
rw_perr_ignore    0
rw_rpc_future     00
rs_rpc0_pdata     7FFFF
rs_rpc1_pdata     7FFFF
rs_unused         0000000
rs_end            E
```

RAT Module Control Register USER2

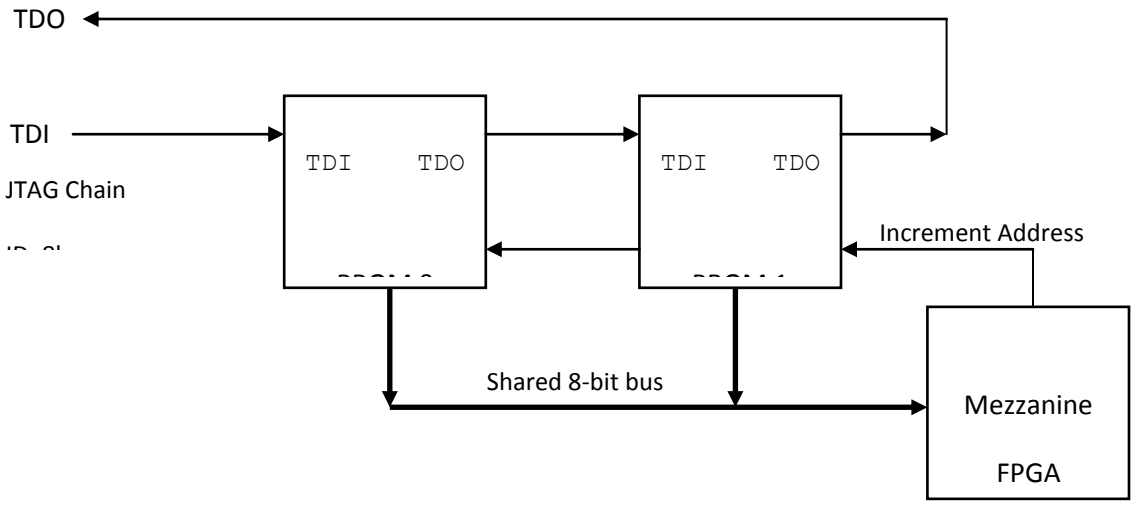
```
RAT USER2=0118019B
ws_rpc_en      3
ws_ddd_start   0
ws_ddd_wr      0033
ws_dddoe_wr    3
ws_perr_reset  0
ws_parity_odd  1
ws_perr_ignore 0
ws_rpc_future  00
```

User PROM Programming

TMB has two on-board erasable PROMs that contain non-volatile data for modifying VME registers and for initializing JTAG chains. The PROMs are read automatically by firmware state machines after power-up or hard-reset.

Both PROMs are Xilinx XC18V256 devices, which have a capacity of 256K bits stored 1-byte wide in 32K addresses. Some late-production TMBs may have one or two XC18V512 devices, because the 256K product was discontinued by the manufacturer.

PROM contents are programmed and verified either via the front-panel JTAG connector or from the VME backplane using TMBs boot register. The data may be read out 8-wide though TMBs PROM-port register ADR_PROM (12h) or it may be read 1-bit serial via JTAG.



Register Initialization

Initialization after power-up or hard reset proceeds as follows:

1) FPGA loads its firmware from mezzanine PROMs (100msec)

2) Delay Locked Loops (DLLs) acquire lock on the 40MHz TTC clock (lock time not yet tested)

3) VME registers load their default values (100ns)

4) The VMEsm state machine reads PROM 0 and writes any new data to the specified VME registers.

5) JTAGsm state machine reads PROM 1 and write JTAG data to the specified chains.

VME PROM-0

After TMB loads default values into its VME registers, the VMEsm state machine reads PROM 0 and updates the specified VME registers with PROM data. The PROM contains a 16-word header, followed by an arbitrary number of 24-bit VME-addresses and their associated 16-bit data word, then followed by a 3-byte trailer sequence that contains the checksum and end-of-PROM marker.

The VMEsm state machine expects PROM-0 data to be organized in a specific format and it may either terminate the read-process or indicate an error condition if the format is incorrect. If the BCh header-begin marker is missing, the state machine terminates reading immediately.

The 16-byte header consists of the BC "begin CLCT data" marker, a 2-byte word count, and the EC end marker. There are no restrictions on the values of the other 12 header bytes. VMEsm assembles the two word-count bytes into a 16-bit number that it uses to recognize when to stop reading the PROM.

The word-count includes every PROM address between the BC and FF markers, inclusive. If there are n VME addresses to be written, the word count would be 16 header + 5n vme data + 3 trailer = 19_{10} +5n. A counter in ADR_VMESM3 records the number of VME addresses what were actually written. A value of 55Aah should be written to VME address ADR_VMESM4 to test the PROM-to-FPGA data path.

VME PROM-0 data format:

```
Adr    7654 3210    Hex    Description
---    ---- ----   --      -----------
  0    1011 1100    BC     Begin CLCT Header Marker, if missing state machine stops
  1    tttt oooo    LL     Word count [7:0] kkkk hhhh tttt oooo from BC to FF
  2    kkkk hhhh    HH     Word count [15:8]
  3    tttt oooo    12     Month (month/day in "hex-ascii" December 31, 2006)
  4    tttt oooo    31     Day
  5    tttt oooo    06     Year   2006   = kkkk hhhh tttt oooo
  6    kkkk hhhh    20     Year
  7    vvvv vvvv    XX     Version number = vvvv vvvv
  8    xxxx xxxx    AA     Option (suggest AA to test even bits)
  9    xxxx xxxx    55     Option (suggest 55 to test odd  bits)
  A    xxxx xxxx    XX     Option
  B    xxxx xxxx    XX     Option
  C    xxxx xxxx    XX     Option
  D    xxxx xxxx    XX     Option
  E    xxxx xxxx    XX     Option
  F    1110 1100    EC     End Header Marker

 10    aaaa aaaa    LL     VME adr[7:0]        This 5-byte structure is repeated
 11    aaaa aaaa    MM     VME adr[15:8]       for every VME address that is
 12    aaaa aaaa    HH     VME adr[23:16]      to be modified
 13    dddd dddd    LL     VME data[7:0]
 14    dddd dddd    HH     VME data[15:8]

 L-2   1111 1100    FC     End of CLCT VME data Marker
 L-1   cccc cccc    cc     Check sum  [7:0] includes addresses 0 to L-2
 L     1111 1111    FF     End of PROM data Marker
```

If VMEsm detects an error condition, status information can be read from VME address ADR_VMESM2 (0xDE):

fmt_err[0] = Missing BC header-begin marker
fmt_err[1] = Missing EC header-end marker
fmt_err[2] = Missing FC data-end marker
fmt_err[3] = Missing FF prom-end marker
fmt_err[4] = Word counter overflow

JTAG PROM-1

After the VMEsm state machine completes successful, the JTAGsm machine is started. It reads PROM-1 and writes to the JTAG chains specified by the PROM data. The PROM contains a 16-word header, followed by an arbitrary number of 3-byte Chain Blocks, then followed by a 8-byte trailer sequence that contains the tck-count, word-count, checksum and end-of-PROM marker.

Header

The JTAGsm state machine expects PROM-1 data to be organized in a specific format and it may either terminate the read-process or indicate an error condition if the format is incorrect. If the Bah header-begin marker is missing, the state machine terminates reading immediately.

The 16-byte header consists of the BA "begin ALCT data" marker and the EA end marker. The state machine skips over the next 15 bytes, so are no restrictions on the header contents.

Adr	7654 3210	Hex	Description
0	10111010	BA	Begin ALCT Marker, if "BA" missing state machine stops
1	0000aaaa	03	ALCT MSD 3 Type (288,384,672)
2	0000aaaa	08	ALCT 8
3	0000aaaa	04	ALCT LSD 4
4	0000mmmm	00	Month MSD 0 in "hex-ascii" June 9, 2008
5	0000mmmm	06	Month LSD 6
6	0000dddd	00	Day MSD 0
7	0000dddd	09	Day LSD 9
8	0000yyyy	02	Year MSD 2
9	0000yyyy	00	Year 0
A	0000yyyy	00	Year 0
B	0000yyyy	08	Year LSD 8
C	0000vvvv	01	Version number [3:0]
D	0000xxxx	00	Future use
E	0000xxxx	00	Future use
F	00101010	EA	End ALCT Header Marker

Chain Block

A Chain Block marker "Cs" signals the start of a new sequence to generate TCK,TMS, and TDI for a JTAG chain. The "s" in the "Cs" marker is the chain address SEL[3:0]. Normally, TCK will be held high after the last TMS/TDI pair is sent, unless the "C" marker is replaced by a "D" maker. After Cs, there are two more bytes specifying the number of TCK clock pulses to send to the chain.

Following the TCK count are JTAG data bytes that contain packed TMS and TDI bits for up to 4 TCKs. The number of data bytes is tck_count/4+1.

Chain Block Format [7:0]:

76543210		
1100ssss	Cs	Chain Block begin marker chain address ssss =SEL[3:0]. Cs=hold tck high, Ds= do not
tttttttt	ww	TCK count [15:8]
tttttttt	ww	TCK count [7:0]
1sisisisi	si	JTAG data [7:0], I=TDI bit, s=TMS bit, defined below

4 TCKs packed per byte format, data[7:0]:

Bit	Signal
[0]	TDI[0]
[1]	TMS[0]
[2]	TDI[1]
[3]	TMS[1]
[4]	TDI[2]
[5]	TMS[2]
[6]	TDI[3]
[7]	TMS[3]

SEL[3:0] Selects the active JTAG chain:

SEL[3:0] Hex	3210	Function
0	0000	ALCT Slow Control FPGA user registers
1	0001	ALCT Slow Control FPGA PROM)
2	0010	ALCT Mezzanine FPGA user registers
3	0011	ALCT Mezzanine FPGA PROMs
4	01XX	TMB Mezzanine FPGA+PROMs
8	10XX	TMB User PROMs
C	1100	TMB FPGA Monitor (for TMB self-test)
D	1101	RAT Module FPGA+PROM

Chain Block, Continued

The state machine copies the signals TMS, and TDI to the JTAG chain selected by SEL[3:0]. It automatically generates the JTAG clock TCK and also arranges for asserting the next TMS and TDI values while TCK is low.

TMB hardware pulls TCK, TMS, and TDI high for chains that are not currently selected. When changing to a new chain ID, it is recommend that the last PROM word for that chain sets the signals to a logic high.

The number of Chain Blocks is limited only by the memory capacity of the PROM. Each block begins with a Cs marker, and the last block is indicated by an "FA" marker.

Trailer

An 8-byte trailer sequence contains the total-tck-count, PROM word count, data checksum and end-of-PROM marker. The tck-count is the number of TCKs the state machine sent to all chains. The word-count includes every PROM address between the BA and FA markers, inclusive, as well as the 3 TCK-count bytes. Checksum includes every PROM address from the BA marker to the last word-count byte.

Adr	7654 3210	Hex	Description	
T-1	11111010	FC	End of JTAG data Marker	Also set chain address to C
T+0	tttttttt	tt	TCK Count Total [17:16]	Includes tcks sent for all chain blocks
T+1	tttttttt	tt	TCK Count Total [15:8]	
T+2	tttttttt	tt	TCK Count Total [7:0]	
T+3	wwwwwwww	ww	Word Count [15:8]	Includes Adr 0 and end JTAG marker at T+2
T+4	wwwwwwww	ww	Word Count [7:0]	
T+5	cccccccc	cc	Check sum [7:0]	Includes addresses 0 and T+5
T+6	11111111	FF	End of PROM data Marker	

If the tck-count, word-count or checksum are incorrect, or if the last-word marker is missing, the state machine indicates the an error by setting jsm_tck_fpga_ok=0.

TCK Throttle

The JTAGsm state machine reads header and trailer bytes from the PROM at 20MHz, then slows to 2.5MHz for processing JTAG data bytes. This results in TCK being pulsed at 10MHz (because there are 4 TCKs per byte, each 50ns high + 50ns low).

If this rate is too high for ALCT JTAG chains, the state machine can be throttled to a lower speed by setting jsm_throttle[3:0] in ADR_VMESM0 (D4h). A jsm_throttle value of 0 corresponds firing TCK at 10MHz full speed. Increasing jsm_throttle by 'n' increases the TCK period by 25ns*n, and maintains a 50% duty cycle.

State Machine Status

Automatic operation of the state machine can be verified by including TCK writes to chain address C. That chain effectively loops-back TCK, TMS and TDI to the FPGA. If at least 1 TCK transition is seen on chain C, then signal jsm_tck_fpga_ok will be a logic 1 in VME register ADR_JTAGSM0, and tck_fpga_cnt[3:0] in ADR_JTAGSM2 counts

the number of TCKs. Demo software exists that converts old PROM data files to this new format and automatically inserts a C-chain Block if there is not one already.

JTAG PROM-1 data format Example:

```
Adr   7654 3210   Hex   Description
---   ---- ----   --    -----------
  0   10111010    BA    Begin ALCT Marker, if "BA" missing state machine stops
  1   0000aaaa    03    ALCT  MSD 3 Type (288,384,672)
  2   0000aaaa    08    ALCT      8
  3   0000aaaa    04    ALCT  LSD 4
  4   0000mmmm    00    Month MSD 0  in "hex-ascii" June 9, 2008
  5   0000mmmm    06    Month LSD 6
  6   0000dddd    00    Day   MSD 0
  7   0000dddd    09    Day   LSD 9
  8   0000yyyy    02    Year  MSD 2
  9   0000yyyy    00    Year      0
  A   0000yyyy    00    Year      0
  B   0000yyyy    08    Year  LSD 8
  C   0000vvvv    01    Version number [3:0]
  D   0000xxxx    00    Future use
  E   0000xxxx    00    Future use
  F   00101010    EA    End ALCT Header Marker

  10  1100ssss    C3    Chain Block Markder for chain adr 3 (or Ds to set TCK low)
  11  tttttttt    ww    TCK count [15:8]
  12  tttttttt    ww    TCK count [7:0]
  13  sisisisi    si    JTAG data
  14  sisisisi    si    JTAG data
  15  sisisisi    si    JTAG data

  T-1 11111010    FC    End of JTAG data Marker
  T+0 tttttttt    tt    TCK Count Total [17:16] Includes tcks sent for all chain blocks
  T+1 tttttttt    tt    TCK Count Total [15:8]
  T+2 tttttttt    tt    TCK Count Total [7:0]
  T+3 wwwwwwww    ww    Word Count [15:8]       Includes Adr 0 and end JTAG marker at T+2
  T+4 wwwwwwww    ww    Word Count [7:0]
  T+5 cccccccc    cc    Check sum  [7:0]        Includes addresses 0 and T+5
  T+6 11111111    FF    End of PROM data Marker
```

DMB Readout

Full-Readout and Local-Readout Format (Long Header):

1	DB0C	header	Beginning Of Cathode Data	
7	event	header	Non-buffered event data	
e	clct	header	Cathode LCTs	
e	tmb	header	TMB match result	
e	mpc	header	MPC frames	
e	rpc	header	RPC status	
e	buf	header	Buffer status	
e	6E0B	header	End of header block	
n	hits		CFEB0 raw hits	
n	hits		CFEB1 raw hits	
n	hits		CFEB2 raw hits	
n	hits		CFEB3 raw hits	
n	hits		CFEB4 raw hits	
1	6B04		Start of RPC raw hits marker	(optional)
m	hits		RPC0 raw hits	(optional)
m	hits		RPC1 raw hits	(optional)
1	6E04		End of RPC raw hits marker	(optional)
1	6B05		Beginning scope data	(optional)
s	scope data		Scope data	(optional)
1	6E05		End of scope data	(optional)
1	6B07		Beginning miniscope data	(optional)
22	miniscope data		Miniscope data 22 words	(optional)
1	6E07		End of miniscope data	(optional)
1	6BCB		Beginning blocked CFEB list	(optional)
20	blocked cfebs list		Blocked CFEBs list 20 words	(optional)
1	6ECB		End blocked CFEB list	(optional)
1	6E0C		End of raw hits	
1	2AAA		Make word count x4	(inserted only if needed)
1	5555		Make word count x4	(inserted only if needed)
1	DE0F		End of Frame	
1	Dcrc0		CRC22[10:0]	
1	Dcrc1		CRC22[21:11]	
1	Dwordcount		Total words in transmission	(inclusive)

Word Count = 42(nheaders)
+ 1(E0B)
+ ncfebs*(6*ntbins)
+1(B04) (if RPC readout enabled)
+ nrpcs*(2*ntbins) (if RPC readout enabled)
+1(E04) (if RPC readout enabled)
+1(B05) (if Scope readout enabled)
+nch(128)/16*256 (if Scope readout enabled)
+1(E05) (if Scope readout enabled)
+1(B07) (if miniscope readout enabled)
+22 (if miniscope readout enabled)
+1(E07) (if miniscope readout enabled)
+1(BCB) (if blocked cfebs list readout enabled)
+ncfebs*4 (if blocked cfebs list readout enabled)
+1(ECB) (if blocked cfebs list readout enabled)
+1(EOC)
+2(2AAA 5555) (if needed to make word count multiple of 4)
+1(E0F)
+2(crc)
+1(wordcount)

Long Header-only Format:

1	DB0C	header	Beginning of Cathode Data
7	event	header	Non-buffered event data
e	clct	header	Cathode LCTs
e	tmb	header	TMB match result
e	mpc	header	MPC frames
e	buf	header	Buffer status
e	rpc	header	RPC status
1	6E0B		End of header block
1	6E0C		End of raw hits
1	DE0F		End of Frame
1	Dcrc0		CRC22
1	Dcrc1		CRC22
1	Dwordcount		Total words in transmission (inclusive)

40 words = nheaders(34)+E0B+E0C+2crc+E0F+wdcnt

Short Header-only Format:

1	DB0C	header	Beginning of Cathode Data
7	event	header	Non-buffered event data
1	DEEF		End of Frame
1	Dcrc0		CRC22
1	Dcrc1		CRC22
1	Dframe wordcount		Total words in transmission (inclusive)

12 words = 8headers+2crc+EEF+wdcnt

Miniscope:

TMB contains a smaller version of the main digital scope that is intended to be included in the DMB readout stream by default. The minscope displays a time history of ALCTs, CLCTs, and L1As that helps reproduce TMB behavior offline.

Miniscope data is written continuously to Block RAM, in the same fashion as CFEB raw hits are stored.
When there is a CLCT pre-trigger, miniscope data starts at `fifo_pretrig_mini` time bins before the pre-trigger occurred. If there is no CLCT pre-trigger, mininscope data can still be read out in an L1A-only event, using the L1A look-back mode.

The miniscope is enabled by setting:

```
0x10C [00]    mini_read_enable = 1   Turn miniscope on
0x10C [01]    mini_tbins_test  = 1   Turn debug mode off
0x10C [02]    mini_tbins_word  = 1   Insert tbins and pre-trig tbins settings in 1st word
0x10C [07:03] fifo_tbins_mini  = 22  Number of time bins to read out, must be a multiple of 2, but not of 4
0x10C [12:08] fifo_pretrig_mini = 4  Number of time bins before pre-trigger
```

Once enabled, miniscope data is included in every event readout that contains a full header.
Events the contain miniscope data will have

1) header19[14] mini_read_enable = 1 to indicate the event includes miniscope data.
2) 06B07 begin minscope marker in the data stream, located after RPC and main-scope data
3) Typically 22 miniscope time-bin data words
4) 06E07 end miniscope marker

The number of minscope time-bin data words depends on the VME register 0x10C[7:3].
Because the total number of words in the DMB readout is required to be a multiple of 4, and 2 words are used for 06E07|06E07 markers, the number of miniscope time-bin words must be a multiple of 2, but not a multiple of 4.

The first word after the 06B07 marker contains the number of minscope time-bins, which along with the mini_read_enable bit in header19[14] allows event-unpacking software to locate the miniscope data markers and predict the total event word count.

Miniscope data is inserted between RPC and Blocked-bits list

```
Adr=  282 Data= 06E04
Adr=  283 Data= 06B07   ← begin miniscope marker
Adr=  284 Data= 00416   ← tbin count 0x16 and tbins before pre-trigger 0x04
Adr=  285 Data= 00002   ← 14 minicope channels at tbin=1
Adr=  286 Data= 00002
Adr=  287 Data= 00002
Adr=  288 Data= 00403
Adr=  289 Data= 01805
Adr=  290 Data= 01009
Adr=  291 Data= 01009
Adr=  292 Data= 02009
Adr=  293 Data= 00009
Adr=  294 Data= 00158
Adr=  295 Data= 00302
Adr=  296 Data= 00102
Adr=  297 Data= 00002
Adr=  298 Data= 00002
Adr=  299 Data= 00002
Adr=  300 Data= 00002
Adr=  301 Data= 00002
Adr=  302 Data= 00002
```

```
Adr=  303 Data= 00002
Adr=  304 Data= 00002
Adr=  305 Data= 00002
Adr=  306 Data= 06E07   ← end miniscope marker
Adr=  307 Data= 06BCB   ← blocked bits marker, if blocked bits are in the readout
```

Miniscope Channel Assignments:

ch[00]	any_cfeb_hit	At least 1 CFEB meets pre-trigger layer-hit threshold
ch[03:01]	clct_sm_vec[2:0]	CLCT pre-trigger state machine vector
ch[04]	clct0_vpf	Valid pattern flag for 1st best CLCT
ch[05]	clct1_vpf	Valid pattern flag for 2nd best CLCT
ch[06]	alct0_vpf_tprt	Valid pattern flag for 1st best ALCT, after pipe delay
ch[07]	alct1_vpf_tprt	Valid pattern flag for 2nd best ALCT, after pipe delay
ch[08]	clct_window	CLCT match window
ch[09]	wr_push_rtmb	ALCT*CLCT match signal
ch[10]	tmb_push_dly	L1A signals are l1a_delay+2bx later
ch[11]	l1a_pulse	L1A from CCB
ch[12]	l1a_window_open	L1A window
ch[13]	l1a_push_me	L1A queued for readout signal
ch[14]	tmb_special	Always 0 in DMB readout, required by unpacker
ch[15]	ddu_special	Always 0 in DMB readout, required by DDU

CLCT Pre-trigger State Machine Vector:

0 s: Startup wait after hard-reset
1 I: Idle, waiting for pre-trigger
2 p: Pre-triggered
3 f: Flushing triad one-shots [checks any_cfeb_hit, waits n-bx, returns to idle, n may be 0, n=1 here]
4 t: Trigger-rate throttle [optional fixed delay before returning to idle]
5 h: Halted

Decoded Readout Example:

```
                          0000000000000000111111          ← Time bins
                          123456789ABCDEF012345
ch 00  any_cfeb_hit       ___------_____________          ← CLCT signals are in pre-trigger time domain
ch 03  clct_state_machine 111124444411111111111
ch 03  clct_state_machine iiiipfffffiiiiiiiiii            ← CLCT Pre-trigger machine state
ch 04  clct0_vpf          _________-__________
ch 05  clct1_vpf          ____________________
ch 06  alct0_vpf          _________-__________            ← ALCT arrival is needed to check TMB offline
ch 07  alct1_vpf          ____________________
ch 08  clct_window        _________---_________
ch 09  wr_push_rtmb       __________-__________           ← ALCT*CLCT match signal
ch 10  tmb_push_dly       ___-________________            ← L1A signals are l1a_delay+2bx later
ch 11  l1a_pulse          ____-_______________            ← L1A from CCB merged with other L1A
ch 12  l1a_window_open    ____---_____________
ch 13  l1a_push_me        _______-_____________           ← L1A queued for readout signal
ch 14  tmb_special        ____________________            ← Unpacker prevents TMB from using 15th bit
ch 15  ddu_special        ____________________            ← DDU prevents TMB from using 16th bit
```

Blocked CFEB DiStrips List Format:

1	6BCB marker	Beginning of blocked cfeb distrip list
1	CFEB0 word 0	[14:12]=cfebid[2:0]=0 [11:0]=cfeb0 distrips[11:0]
1	CFEB0 word 1	[14:12]=cfebid[2:0]=0 [11:0]=cfeb0 distrips[23:12]
1	CFEB0 word 2	[14:12]=cfebid[2:0]=0 [11:0]=cfeb0 distrips[35:24]
1	CFEB0 word 3	[14:12]=cfebid[2:0]=0 [11:0]=cfeb0 distrips[47:36]
1	CFEB1 word 0	[14:12]=cfebid[2:0]=1 [11:0]=cfeb1 distrips[11:0]
1	CFEB1 word 1	[14:12]=cfebid[2:0]=1 [11:0]=cfeb1 distrips[23:12]
1	CFEB1 word 2	[14:12]=cfebid[2:0]=1 [11:0]=cfeb1 distrips[35:24]
1	CFEB1 word 3	[14:12]=cfebid[2:0]=1 [11:0]=cfeb1 distrips[47:36]
1	CFEB2 word 0	[14:12]=cfebid[2:0]=2 [11:0]=cfeb2 distrips[11:0]
1	CFEB2 word 1	[14:12]=cfebid[2:0]=2 [11:0]=cfeb2 distrips[23:12]
1	CFEB2 word 2	[14:12]=cfebid[2:0]=2 [11:0]=cfeb2 distrips[35:24]
1	CFEB2 word 3	[14:12]=cfebid[2:0]=2 [11:0]=cfeb2 distrips[47:36]
1	CFEB3 word 0	[14:12]=cfebid[2:0]=3 [11:0]=cfeb3 distrips[11:0]
1	CFEB3 word 1	[14:12]=cfebid[2:0]=3 [11:0]=cfeb3 distrips[23:12]
1	CFEB3 word 2	[14:12]=cfebid[2:0]=3 [11:0]=cfeb3 distrips[35:24]
1	CFEB3 word 3	[14:12]=cfebid[2:0]=3 [11:0]=cfeb3 distrips[47:36]
1	CFEB4 word 0	[14:12]=cfebid[2:0]=4 [11:0]=cfeb4 distrips[11:0]
1	CFEB4 word 1	[14:12]=cfebid[2:0]=4 [11:0]=cfeb4 distrips[23:12]
1	CFEB4 word 2	[14:12]=cfebid[2:0]=4 [11:0]=cfeb4 distrips[35:24]
1	CFEB4 word 3	[14:12]=cfebid[2:0]=4 [11:0]=cfeb4 distrips[47:36]
1	6ECB marker	End of blocked cfeb distrip list

22 words

Blocked DiStrips list includes:

DiStrips turned off via VME Hot Channel Mask
DiStrips turned off via mask_all applied to the entire CFEB
DisStrips marked as bad by automatic bad-bits detection

CFEB DiStrip Bit Packing:

Each CFEB has 6 layers of 8 DiStrips = 48 bits

```
block_distrip_list[ 7: 0] = Layer0 Ds[7:0]
block_distrip_list[15: 8] = Layer1 Ds[7:0]
block_distrip_list[23:16] = Layer2 Ds[7:0]
block_distrip_list[31:24] = Layer3 Ds[7:0]
block_distrip_list[39:32] = Layer4 Ds[7:0]
block_distrip_list[47:40] = Layer5 Ds[7:0]
```

Which are packed into 4 readout words, 12 bits per word = 48 bits

```
CFEBn word 0  = block_distrip_list[11: 0]
CFEBn word 1  = block_distrip_list[23:12]
CFEBn word 2  = block_distrip_list[35:24]
CFEBn word 3  = block_distrip_list[47:36]
```

Header Word Descriptions:

First 4 header words must conform to DDU format specification:

header00_[11:0]	12'hB0C	Beginning of Cathode record marker
header00_[14:12]	3'b101	DDU code for TMB/ALCT
header00_[15]	1	DDU special-word flag
header01_[11:0]	pop_bxn_counter[11:0]	BXN pushed into L1A queue at L1A arrival
header01_[14:12]	3'b101	DDU code for TMB/ALCT
header01_[15]	1	DDU special-word flag

header01 notes:

(4) bxn_counter contains the value of the 12-bit BXN counter at the time L1A arrives, and is typically 128bx later than the pre-trigger BXN (see header08). Readouts will always have bxn_counter, but may or may not have pre-trigger data.

header02_[11:0]	pop_l1a_rx_counter[11:0]	L1As received by TMB
header02_[14:12]	3'b101	DDU code for TMB/ALCT
header02_[15]	1	DDU special-word flag
header03_[11:0]	readout_counter[11:0]	Counts L1A readouts
header03_[14:12]	3'b101	DDU code for TMB/ALCT
header03_[15]	1	DDU special-word flag

Next 4 words for short-header mode or full-header:

header04_[4:0]	board_id[4:0]	TMB module ID number = VME slot number 1-20
header04_[8:5]	csc_id[3:0]	Chamber ID number, set by VME register
header04_[12:9]	run_id[3:0]	Run info, set by VME register
header04_[13] =	buf_q_ovf_err	Tried to push new event when queue full
header04_[14] =	r_sync_err	BXN sync error

header04 notes:

(4) board_id defaults to the VME crate slot number, unless overridden via VME.

(2) csc_id is a user-set value to identify the CSC connected to this TMB

(3) run_id is a user-set value to identify the current data run.

(4) buf_q_ovf indicates that more L1As arrived than TMB was able to push into its readout processing queue. In this case the average trigger rate is probably higher than the readout data path can tolerate.

(5) sync_err indicates that bx0 did not arrive when the BXN counter turned over to the bxn-preset value. Either bx0 is not functioning, or the 40 MHz clock gained or lost counts. A sync_err is latched-on until ttc_resync or ttc_bxreset.

header05_[5:0]	r_nheaders[5:0]	Number of header words
header05_[8:6]	fifo_mode[2:0]	Raw hits fifo readout mode set via VME
header05_[10:9]	r_type[1:0]	Record type: dump, nodump, full header, short header
header05_[12:11]	l1a_type[1:0]	L1A Pop type code: buffers, no buffers, clct/alct_only
header05_[13]	r_has_buf	Event has clct and rpc buffer data
header05_[14]	r_buf_stalled	Buffer write pointer hit a fence and stalled

header05 notes:

(4) nheaders indicates the length of the current header block, including the BOC marker to 1 frame before the EOB marker.
In current firmware it will be 8 for short headers and 42 for full.

(2) FIFO Modes:

mode	raw hits	header	
0	no	full	(if buffer was available at pre-trigger)
1	all 5 CFEBs	full	(if buffer was available at pre-trigger)
2	local	full	(if buffer was available at pre-trigger), local=sparsified cfebs
3	no	short	
4	no	no	

(3) Record Type Codes:

r-type	raw hits	header	
0	no	full	
1	full	full	
2	local	full	
3	no	short	(no buffer was available at pre-trigger)

(4) L1A Type Codes:

l1a-type		
0	Normal CLCT trigger with buffer data and L1A window match	
1	ALCT-only trigger, no data buffers	(not usually read out)
2	L1A-only, no matching TMB trigger, no buffer data	(not usually read out)
3	TMB triggered, no L1A-window match, event has buffer data	(not usually read out)

header06_[14:0]	bd_status[14:0]	Board status summary

header06 notes:

(1)	bd_status[0]	bd_status_ok	Board all-OK: voltages OK, temperature OK, prom-load OK
	bd_status[1]	vstat_5p0v	Voltage Comparator +5.0V, 1=OK
	bd_status[2]	vstat_3p3v	Voltage Comparator +3.3V, 1=OK
	bd_status[3]	vstat_1p8v	Voltage Comparator +1.8V, 1=OK
	bd_status[4]	vstat_1p5v	Voltage Comparator +1.5V, 1=OK
	bd_status[5]	*t_crit	Temperature ADC Tcritical 1=OK
	bd_status[6]	vsm_ok	VME Machine ran without errors
	bd_status[7]	vsm_aborted	VME State machine aborted reading PROM
	bd_status[8]	vsm_cksum_ok	VME Check-sum matches PROM contents
	bd_status[9]	vsm_wdcnt_ok	VME Word count matches PROM contents
	bd_status[10]	jsm_ok	JTAG state machine completed without errors
	bd_status[11]	jsm_aborted	JTAG State machine aborted reading PROM
	bd_status[12]	jsm_cksum_ok	JTAG Check-sum matches PROM contents
	bd_status[13]	jsm_wdcnt_ok	JTAG Word count matches PROM contents
	bd_status[14]	jsm_tck_fpga_ok	FPGA jtag tck detected correctly

header07_[14:0]	revcode[14:0]	Firmware version date code

header07 notes:

(1)

revcode[04:00]	day 1-31
revcode[08:05]	month 1-12
revcode[12:09]	years after 2000
revcode[14:13]	fpga type, 1 for xc2v3000, 2 for xc2v4000

Full Header-mode words 8-EOB: Event Counters

header08_[11:0]	r_bxn_counter_ff[11:0]	CLCT Bunch Crossing number at pre-trigger, 0-3563
header08_[12]	r_tmb_clct0_discard;	TMB discarded clct0 from ME1A
header08_[13]	r_tmb_clct1_discard;	TMB discarded clct1 from ME1A
header08_[14]	clock_lock_lost	Main DLL clock lost lock
header09_[14:0]	r_pretrig_counter[14:0]	Counts CLCT pre-triggers [stops on ovf]
header10_[14:0]	r_pretrig_counter[29:15]	
header11_[14:0]	r_clct_counter[14:0]	Counts CLCTs post-drift [stops on ovf]
header12_[14:0]	r_clct_counter[29:15]	
header13_[14:0]	r_trig_counter[14:0]	Counts TMB triggers to MPC, L1A request to CCB,
header14_[14:0]	r_trig_counter[29:15]	[stops on ovf]
header15_[14:0]	r_alct_counter[14:0]	Counts ALCTs received from ALCT board [stops on ovf]
header16_[14:0]	r_alct_counter[29:15]	
header17_[14:0]	r_orbit_counter[14:0]	BX0s since last hard reset [stops on ovf]
header18_[14:0]	r_orbit_counter[29:15]	

CLCT Raw Hits Size:

header19_[2:0]	r_ncfebs[2:0]	Number of CFEBs read out
header19_[7:3]	r_fifo_tbins[4:0]	Number of time bins per CFEB in dump
header19_[12:8]	fifo_pretrig[4:0]	Number of time bins before pre-trigger
header19_[13]	scope_data_exists	Readout includes logic analyzer scope data
header19_[14]	mini_read_enable	Readout includes miniscope data, 22wds+2markers

CLCT Configuration:

header20_[2:0]	hit_thresh_pretrig[2:0]	Hits on pattern template pre-trigger threshold
header20_[6:3]	pid_thresh_pretrig[3:0]	Pattern shape ID pre-trigger threshold
header20_[9:7]	hit_thresh_postdrift[2:0]	Hits on pattern post-drift threshold
header20_[13:10]	pid_thresh_postdrift[3:0]	Pattern shape ID post-drift threshold
header20_[14]	stagger_hs_csc	CSC Staggering ON
header21_[3:0]	triad_persist[3:0]	CLCT Triad persistence
header21_[6:4]	dmb_thresh_pretrig[2:0]	DMB pre-trigger threshold for active-cfeb list
header21_[10:7]	alct_delay[3:0]	Delay ALCT for CLCT match window
header21_[14:11]	clct_window[3:0]	CLCT match window width

CLCT Trigger Status:

header22_[8:0] r_trig_source_vec[8:0] Pre-trigger source vector
header22_[14:9] r_layers_hit_vec[5:0] CSC layers hit on layer trigger

header22 notes:

(1)
- trig_source [0] CLCT pattern triggered sequencer
- trig_source [1] ALCT pattern triggered sequencer
- trig_source [2] ALCT*CLCT pattern triggered sequencer
- trig_source [3] ADB external triggered sequencer
- trig_source [4] DMB external triggered sequencer
- trig_source [5] CLCT (CCB scintillator) external triggered sequencer
- trig_source [6] ALCT (CCB) external triggered sequencer
- trig_source [7] VME triggered sequencer
- trig_source [8] Layer-mode trigger

header23_[4:0] r_active_feb_mux[4:0] Active CFEB list sent to DMB
header23_[9:5] r_cfebs_read[4:0] CFEBs read out for this event
header23_[13:10] pop_l1a_match_win[3:0] Position of l1a in window
header23_[14] active_feb_src Active CFEB list source, 0=pretrig, 1=at TMB match

CLCT+ALCT Match Status:

header24_[0] r_tmb_match ALCT and CLCT matched in time, pushed into L1A queue
header24_[1] r_tmb_alct_only Only ALCT triggered, pushed into L1A queue
header24_[2] r_tmb_clct_only Only CLCT triggered, pushed into L1A queue
header24_[6:3] r_tmb_match_win[3:0] Location of alct in clct window, pushed into L1A queue
header24_[7] r_no_alct_tmb; No ALCT
header24_[8] r_one_alct_tmb; One ALCT
header24_[9] r_one_clct_tmb; One CLCT
header24_[10] r_two_alct_tmb; Two ALCTs
header24_[11] r_two_clct_tmb; Two CLCTs
header24_[12] r_dupe_alct_tmb; ALCT0 copied into ALCT1 to make 2nd LCT
header24_[13] r_dupe_clct_tmb; CLCT0 copied into CLCT1 to make 2nd LCT
header24_[14] r_rank_err_tmb; LCT1 has higher quality than LCT0, error

CLCT Trigger Data:

header25_[14:0]	r_clct0_tmb[14:0]	CLCT0 after drift lsbs
header26_[14:0]	r_clct1_tmb[14:0]	CLCT1 after drift lsbs
header27_[0]	r_clct0_tmb[15]	CLCT0 after drift msbs
header27_[1]	r_clct1_tmb[15]	CLCT1 after drift msbs
header27_[4:2]	r_clctc_tmb[2:0]	CLCT0/1 common after drift msbs
header27_[5]	r_clct0_invp	CLCT0 had invalid pattern after drift delay
header27_[6]	r_clct1_invp	CLCT1 had invalid pattern after drift delay
header27_[7]	r_clct1_busy	2nd CLCT busy, logic error indicator
header27_[12:8]	perr_cfeb_ff[4:0]	CFEB raw hits RAM parity error, latched, SEU detection
header27_[13] 0	perr_rpc_ff | perr_mini_ff	RPC raw hits RAM parity error, latched, SEU detection
header27_[14] 0	perr_ff	Raw hits RAM parity error summary, latched, SEU

header25-27 notes:

(4) clct0, clct1, clctc packing format:

clct0[0]	clct_1st_valid	Valid pattern flag
clct0[3:1]	hs_hit_1st[2:0]	Hits on pattern 0-6
clct0[7:4]	hs_pid_1st[3:0]	Pattern shape 0-A
clct0[15:8]	hs_key_1st[7:0]	½-strip ID number
clct1[0]	clct_2nd_valid	Valid pattern flag
clct1[3:1]	hs_hit_2nd[2:0]	Hits on pattern 0-6
clct1[7:4]	hs_pid_2nd[3:0]	Pattern shape 0-A
clct1[15:8]	hs_key_2nd[7:0]	½-strip ID number
clctc[1:0]	bxn_counter_ff[1:0]	Bunch crossing number at pretrigger, common to clct0/1
clctc[2]	sync_err	BX0 disagrees with BXN count, common to clct0/1

ALCT Trigger Data:

header28_[0]	alct_1st_valid	ALCT0 valid pattern flag
header28_[2:1]	alct_1st_quality[1:0]	ALCT0 quality
header28_[3]	alct_1st_amu	ALCT0 accelerator muon flag
header28_[10:4]	alct_1st_key[6:0]	ALCT0 key wire group
header28_[14:11]	alct_pretrig_win[3:0]	ALCT active_feb_flag position in pretrig window
header29_[0]	alct_2nd_valid	ALCT1 valid pattern flag
header29_[2:1]	alct_2nd_quality[1:0]	ALCT1 quality
header29_[3]	alct_2nd_amu	ALCT1 accelerator muon flag
header29_[10:4]	alct_2nd_key[6:0]	ALCT1 key wire group
header29_[12:11]	drift_delay[1:0]	CLCT drift delay
header29_[13]	bcb_read_enable	CFEB blocked DiStrip bits list included in readout
header29_[14]	hs_layer_trig	Layer-mode trigger
header30_[4:0]	alct_bxn[4:0]	ALCT0/1 bxn
header30_[6:5]	alct_ecc_err[1:0]	ALCT trigger path ECC error code
header30_[11:7]	cfeb_badbits_found[4:0]	Bad distrip bits detected in cfeb[n]
header30_[12]	cfeb_badbits_blocked	At least one CFEB has a bad bit that was blocked
header30_[13]	alct_cfg_done	ALCT FPGA configuration done
header30_[14]	bx0_match	alct_bx0==clct_bx0, latched at clct_bx0 time

MPC Frames:

header31_[14:0] r_mpc0_frame0_ff[14:0] MPC muon 0 frame 0 LSBs
header32_[14:0] r_mpc0_frame1_ff[14:0] MPC muon 0 frame 1 LSBs
header33_[14:0] r_mpc1_frame0_ff[14:0] MPC muon 1 frame 0 LSBs
header34_[14:0] r_mpc1_frame1_ff[14:0] MPC muon 1 frame 1 LSBs

header35_[0] = r_mpc0_frame0_ff[15] MPC muon 0 frame 0 MSB
header35_[1] = r_mpc0_frame1_ff[15] MPC muon 0 frame 1 MSB
header35_[2] = r_mpc1_frame0_ff[15] MPC muon 1 frame 0 MSB
header35_[3] = r_mpc1_frame1_ff[15] MPC muon 1 frame 1 MSB
header35_[7:4] mpc_tx_delay[3:0] MPC transmit delay
header35_[9:8] r_mpc_accept[1:0] MPC muon accept response
header35_[14:10] cfeb_en[4:0] CFEBs enabled for triggering (didn't fit elsewhere)

header31-35 notes:

(4) MPCframe packing format:

```
mpc0_frame0[6:0]    =   alct0_key[6:0];
mpc0_frame0[10:7]   =   clct0_pat[3:0];
mpc0_frame0[14:11]  =   lct0_quality[3:0];
mpc0_frame0[15]     =   lct0_vpf;

mpc0_frame1[7:0]    =   {clct0_cfeb[2:0],clct0_key[4:0]};
mpc0_frame1[8]      =   clct0_bend;
mpc0_frame1[9]      =   clct_sync_err & tmb_sync_err_en[0];
mpc0_frame1[10]     =   alct0_bxn[0];
mpc0_frame1[11]     =   clct_bx0;   // bx0 gets replaced after mpc_tx_delay, keep here to mollify xst
mpc0_frame1[15:12]  =   csc_id[3:0];

mpc1_frame0[6:0]    =   alct1_key[6:0];
mpc1_frame0[10:7]   =   clct1_pat[3:0];
mpc1_frame0[14:11]  =   lct1_quality[3:0];
mpc1_frame0[15]     =   lct1_vpf;

mpc1_frame1[7:0]    =   {clct1_cfeb[2:0],clct1_key[4:0]};
mpc1_frame1[8]      =   clct1_bend;
mpc1_frame1[9]      =   clct_sync_err & tmb_sync_err_en[1];
mpc1_frame1[10]     =   alct1_bxn[0];
mpc1_frame1[11]     =   alct_bx0;   // bx0 gets replaced after mpc_tx_delay, keep here to mollify xst
mpc1_frame1[15:12]  =   csc_id[3:0];
```

RPC Configuration:

header36_[1:0] rd_list_rpc[1:0] RPCs included in read out
header36_[3:2] r_nrpcs_read[1:0] Number of RPCs in readout, 0,1,2, 0 if header-only
header36_[4] = rpc_read_enable RPC readout enabled
header36_[9:5] fifo_tbins_rpc[4:0] Number RPC FIFO time bins to read out
header36_[14:10] fifo_pretrig_rpc[4:0] Number RPC FIFO time bins before pretrigger

Buffer Status:

header37_[10:0] r_wr_buf_adr[10:0] Buffer RAM write address at pretrigger
header37_[11] r_wr_buf_ready Write buffer was ready at pretrig
header37_[12] wr_buf_ready Write buffer ready now
header37_[13] buf_q_full All raw hits ram in use, ram writing must stop
header37_[14] buf_q_empty No fences remain on buffer stack

header38_[10:0] r_buf_fence_dist[10:0] Distance to 1st fence address at pretrigger
header38_[11] buf_q_ovf_err Tried to push when stack full
header38_[12] buf_q_udf_err Tried to pop when stack empty
header38_[13] buf_q_adr_err Fence adr popped from stack doesn't match rls adr
header38_[14] buf_stalled_once Buffer stalled at least once since last resync

header39_[11:0] buf_fence_cnt[11:0] Number of fences in fence RAM currently
header39_[12] reverse_hs_csc 1=Reverse staggered CSC, non-me1
header39_[13] reverse_hs_me1a 1=ME1A hstrip order reversed
header39_[14] reverse_hs_me1b 1=ME1B hstrip order reversed

header40_[1:0] active_feb_mux[6:5]; Extend Hdr23[4:0] Active CFEB list sent to DMB
header40_[3:2] r_cfebs_read[6:5]; Extend Hdr23[9:5] CFEBs read out for this event
header40_[5:4] perr_cfeb_ff[6:5]; Extend Hdr27[12:8] CFEB RAM parity error, latched
header40_[7:6] cfeb_badbits_found[6:5]; Extend Hdr30[11:7] CFEB[n] has at least 1 bad bit
header40_[9:8] cfeb_en[6:5]; Extend Hdr35[14:10] CFEBs enabled for triggering
header40_[10] buf_fence_cnt_is_peak; Current fence is peak number of fences in RAM
header40_[11] (MXCFEB==7); TMB has 7 DCFEBs so hdr40_[10:0] are active
header40_[13:12] r_trig_source_vec[10:9] Pre-trigger source vector for ME1A/B
header40_[14] r_tmb_trig_pulse TMB trig_pulse signal matched rtmb_push

Spare Frame:

header41_[0] tmb_allow_alct Allow ALCT-only tmb-matching
header41_[1] tmb_allow_clct Allow CLCT-only tmb-matching
header41_[2] tmb_allow_match Allow Match-only tmb-matching
header41_[3] tmb_allow_alct_ro Allow ALCT-only tmb-matching, non-trigger readout
header41_[4] tmb_allow_clct_ro Allow CLCT-only tmb-matching, non-trigger readout
header41_[5] tmb_allow_match_ro Allow Match-only tmb-matching, non-trigger readout
header41_[6] r_tmb_alct_only_ro Only ALCT trig, pushed into L1A queue, non-triggering
header41_[7] r_tmb_clct_only_ro Only CLCT trig, pushed into L1A queue, non-triggering
header41_[8] r_tmb_match_ro ALCT*CLCT match, pushed into L1A queue, non-triggering
header41_[9] r_tmb_trig_keep This is a triggering readout event
header41_[10] r_tmb_non_trig_keep This is a non-triggering readout event
header41_[13:11] lyr_thresh_pretrig[2:0] Layer pre-trigger threshold
header41_[14] layer_trig_en Layer-trigger mode enabled

TMB Data Format: Short Header Mode

FIFO Control				DDU	TMB Data [14:0]														
Frame #	/write fifo	DAV Data Available	last word	d15 DDU special	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
No Write	**1**	**1**	0	0															
0	0	0	0	**1**	DDU Code 101_2			$B0C_{16}$											
1	0	0	0	**1**	DDU Code 101_2			BXN Counter at L1A arrival [11:0]											
2	0	0	0	**1**	DDU Code 101_2			L1A Rx Counter [11:0]											
3	0	0	0	**1**	DDU Code 101_2			Readout Counter[11:0]											
4	0	0	0	0	sync err	buf_q ovf	run_id[3:0]				csc_id[3:0]				board_id[4:0]				
5	0	0	0	0	buf stalled	has buf	l1a_type[1:0]		rec_type[1:0]		fifo_mode[2:0]			nheader_words[5:0]					
6	0	0	0	0	board_status[14:0]														
7	0	0	0	0	firmware_revcode[14:0]														
8	0	0	0	**1**	DDU Code 101_2			EEF_{16} [11:0] (=E0F for full header events, EEF for short header)											
9	0	0	0	**1**	DDU Code 101_2			1 TMB	CRC22[10:0]										
10	0	0	0	**1**	DDU Code 101_2			1 TMB	CRC22[21:11]										
11	0	0	**1**	**1**	DDU Code 101_2			1 TMB	Word Count [10:0]										
No Write	**1**	0	0	0															
Frame #	/write fifo	DAV Data Available	last word	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0

ALCT Header/Trailer Format:

FIFO Control				DDU	ALCT Data [14:0]														
Frame #	/write fifo	DAV Data Available	last word	d15 DDU special	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
No Write	**1**	**1**	0	0															
0	0	0	0	**1**	DDU Code 101_2			$B0A_{16}$											
1	0	0	0	**1**	DDU Code 101_2			BXN Counter at L1A arrival [11:0]											
2	0	0	0	**1**	DDU Code 101_2			L1A Rx Counter [11:0]											
3	0	0	0	**1**	DDU Code 101_2			Readout Counter[11:0]											
n-3	0	0	0	**1**	DDU Code 101_2			$E0D_{16}$ [11:0]											
n-2	0	0	0	**1**	DDU Code 101_2			0 ALCT	CRC22[10:0]										
n-1	0	0	0	**1**	DDU Code 101_2			0 ALCT	CRC22[21:11]										
n	0	0	**1**	**1**	DDU Code 101_2			0 CRC OK=1	Word Count [10:0]										
No Write	**1**	0	0	0															
Frame #	/write fifo	DAV Data Available	last word	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0

Notes:

[1] CRC OK=1 is inserted by TMB after it calculates the CRC for data received from ALCT, and compares it to the CRC words sent by ALCT

TMB Data Format: Long Header-Only Mode

FIFO Control				DDU	TMB Data [14:0]														
Frame #	/write fifo	DAV Data Available	last word	d15 DDU special	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
No Write	**1**	**1**	0	0															
0	0	0	0	**1**	DDU Code 101_2			$B0C_{16}$											
1	0	0	0	**1**	DDU Code 101_2			BXN Counter at L1A arrival [11:0]											
2	0	0	0	**1**	DDU Code 101_2			L1A Rx Counter [11:0]											
3	0	0	0	**1**	DDU Code 101_2			Readout Counter[11:0]											
4	0	0	0	0	sync err	buf_q ovf	run_id[3:0]				csc_id[3:0]				board_id[4:0]				
5	0	0	0	0	buf stalled	has buf	l1a_type[1:0]		rec_type[1:0]		fifo_mode[2:0]			nheader_words[5:0]					
6	0	0	0	0	board_status[14:0]														
7	0	0	0	0	firmware_revcode[14:0]														
8	0	0	0	0	lock lost	clct1 discard	clct0 discard	bxn_counter_ff[11:0]											
9	0	0	0	0	pretrig_counter[14:0]														
10	0	0	0	0	pretrig_counter[29:15]														
11	0	0	0	0	clct_counter[14:0]														
12	0	0	0	0	clct_counter[29:15]														
13	0	0	0	0	trig_counter[14:0]														
14	0	0	0	0	trig_counter[29:15]														
15	0	0	0	0	alct_counter[14:0]														
16	0	0	0	0	alct_counter[29:15]														
17	0	0	0	0	uptime_counter[14:0]														
18	0	0	0	0	uptime_counter[29:15]														
19	0	0	0	0	miniscope read ena	scope esixts	fifo_pretrig[4:0]					fifo_tbins[4:0]					ncfebs[2:0]		
20	0	0	0	0	stagger csc	pid_thresh_postdrift[3:0]			hit_thresh_postdrift[2:0]			pid_thresh_pretrig[3:0]					hit_thresh_pretrig[2:0]		
21	0	0	0	0	clct_window[3:0]				alct_delay[3:0]				dmb_thresh_pretrig[2:0]			triad_persist[3:0]			
22	0	0	0	0	layers_hit_vec[5:0]						trig_source_vec[8:0]								
23	0	0	0	0	aff source	l1a_match_win[3:0]				cfebs_read[4:0]					active_cfeb[4:0]				
24	0	0	0	0	lct rank err	dupe clct	dupe alct	two clct	two alct	one clct	one alct	no alct	match_win[3:0]				clct only	alct only	tmb match

25	0	0	0	0	clct0[14:0]														
26	0	0	0	0	clct1[14:0]														
27	0	0	0	0	perr summary	perr rpc+mini	parity error cfeb ram[4:0] SEU			clct1 busy	clct1 invp	clct0 invp	clctc[2:0]			clct1[15]	clct0[15]		
28	0	0	0	0	alct_pretrig_win[3:0]			alct0_key[6:0]					alct0 amu		alct0 quality[1:0]		alct0 valid		
29	0	0	0	0	layer triggerd	bcb readout	drift_delay[1:0]	alct1_key[6:0]					alct1 amu		alct1 quality[1:0]		alct1 valid		
30	0	0	0	0	bx0 match	alct cfg done	cfeb bits blocked	cfeb_badbits_found[4:0]			alct_ecc_err[1:0]		alct_bxn[4:0]						
31	0	0	0	0	mpc0_frame0[14:0]														
32	0	0	0	0	mpc0_frame1[14:0]														
33	0	0	0	0	mpc1_frame0[14:0]														
34	0	0	0	0	mpc1_frame1[14:0]														
35	0	0	0	0	cfeb_en[4:0]				mpc_accept[1:0]	mpc_tx_delay[3:0]			mpc1fr1 [15]	mpc1fr0 [15]	mpc0fr1 [15]	mpc0fr0 [15]			
36	0	0	0	0	fifo_pretrig_rpc[4:0]				fifo_tbins_rpc[4:0]			rpc read en	nrpcs[1:0]		rpc_list[1:0]				
37	0	0	0	0	buf_q empty	buf_q full	wr_buf ready	r_wr_buf ready	r_wr_buf_adr[10:0]										
38	0	0	0	0	buf stalled ff	buf_q adr err	buf_q udf err	buf_q ovf err	r_buf_fence_dist[10:0]										
39	0	0	0	0	reverse me1b	reverse me1a	reverse csc	buf_fence_cnt[11:0]											
40	0	0	0	0	tmb trig pulse	trig_src_vec[10:9]		mxcfeb=7	peak fence	cfeb_en[6:5]		perr_cfeb[6:5]	cfeb_badbits_found[6:5		cfebs_read[6:5]		active_cfeb[6:5]		
41	0	0	0	0	layer trig enabled	lyr_thresh_pretrig[2:0]			non-trig readout	triggered readout	non-trig match ro	non-trig clct ro	non-trig alct o	allow match ro	allow clct ro	allow alct ro	allow match	allow clct	allow alct
42	0	0	0	0	6			EOB End Header Block											
43	0	0	0	0	6			E0C End Cathode Block											
44	0	0	0	**1**	DDU Code 101_2			$E0F_{16}$[11:0]											
45	0	0	0	**1**	DDU Code 101_2			1 TMB	CRC22[10:0]										
46	0	0	0	**1**	DDU Code 101_2			1 TMB	CRC22[21:11]										
47	0	0	**1**	**1**	DDU Code 101_2			1 TMB	Word Count [10:0]										
No Write	**1**	0	0	0															
Frame #	/write fifo	DAV Data Available	last word	d15	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0

TMB Data Format: Full-Readout Mode

FIFO Control				DDU	TMB Data [14:0]														
Frame #	/write fifo	DAV Data Available	last word	d15 DDU special	d14	d13	d12	d11	d10	d9	d8	d7	d6	d5	d4	d3	d2	d1	d0
No Write	**1**	**1**	0	0															
0	0	0	0	**1**	DDU Code 101_2							$B0C_{16}$							
1	0	0	0	**1**	DDU Code 101_2							BXN Counter at L1A arrival [11:0]							
2	0	0	0	**1**	DDU Code 101_2							L1A Rx Counter [11:0]							
3	0	0	0	**1**	DDU Code 101_2							Readout Counter[11:0]							
4	0	0	0	0	sync err	buf_q ovf	run_id[3:0]				csc_id[3:0]				board_id[4:0]				
5	0	0	0	0	buf stalled	has buf	l1a_type[1:0]		rec_type[1:0]		fifo_mode[2:0]			nheader_words[5:0]					
6	0	0	0	0	board_status[14:0]														
7	0	0	0	0	firmware_revcode[14:0]														
8	0	0	0	0	lock lost	clct1 discard	clct0 discard	bxn_counter_ff[11:0]											
9	0	0	0	0	pretrig_counter[14:0]														
10	0	0	0	0	pretrig_counter[29:15]														
11	0	0	0	0	clct_counter[14:0]														
12	0	0	0	0	clct_counter[29:15]														
13	0	0	0	0	trig_counter[14:0]														
14	0	0	0	0	trig_counter[29:15]														
15	0	0	0	0	alct_counter[14:0]														
16	0	0	0	0	alct_counter[29:15]														
17	0	0	0	0	uptime_counter[14:0]														
18	0	0	0	0	uptime_counter[29:15]														
19	0	0	0	0	miniscopere ad ena	scope esixts	fifo_pretrig[4:0]				fifo_tbins[4:0]				ncfebs[2:0]				
20	0	0	0	0	stagger csc	pid_thresh_postdrift[3:0]			hit_thresh_postdrift[2:0]			pid_thresh_pretrig[3:0]				hit_thresh_pretrig[2:0]			
21	0	0	0	0	clct_window[3:0]				alct_delay[3:0]				dmb_thresh_pretrig[2:0]			triad_persist[3:0]			
22	0	0	0	0	layers_hit_vec[5:0]						trig_source_vec[8:0]								
23	0	0	0	0	aff source	l1a_match_win[3:0]				febs_read[4:0]				active_feb[4:0]					
24	0	0	0	0	lct rank err	dupe clct	dupe alct	two clct	two alct	one clct	one alct	no alct	match_win[3:0]				clct only	alct only	tmb match

25	0	0	0	0	clct0[14:0]														
26	0	0	0	0	clct1[14:0]														
27	0	0	0	0	perr summary	perr rpc+mini	parity error cfeb ram[4:0] SEU					clct1 busy	clct1 invp	clct0 invp	clctc[2:0]			clct1[15]	clct0[15]
28	0	0	0	0	alct_pretrig_win[3:0]				alct0_key[6:0]						alct0 amu		alct0 quality[1:0]		alct0 valid
29	0	0	0	0	layer triggerd	bcb readout	drift_delay[1:0]		alct1_key[6:0]						alct1 amu		alct1 quality[1:0]		alct1 valid
30	0	0	0	0	bx0 match	alct cfg done	cfeb bits blocked	cfeb_badbits_found[4:0]					alct_ecc_err[1:0]		alct_bxn[4:0]				
31	0	0	0	0	mpc0_frame0[14:0]														
32	0	0	0	0	mpc0_frame1[14:0]														
33	0	0	0	0	mpc1_frame0[14:0]														
34	0	0	0	0	mpc1_frame1[14:0]														
35	0	0	0	0	cfeb_en[4:0]					mpc_accept[1:0]		mpc_tx_delay[3:0]				mpc1fr1 [15]	mpc1fr0 [15]	mpc0fr1 [15]	mpc0fr0 [15]
36	0	0	0	0	fifo_pretrig_rpc[4:0]					fifo_tbins_rpc[4:0]					rpc read en	nrpcs[1:0]		rpc_list[1:0]	
37	0	0	0	0	buf_q empty	buf_q full	wr_buf ready	r_wr_buf ready	r_wr_buf_adr[10:0]										
38	0	0	0	0	buf stalled ff	buf_q adr err	buf_q udf err	buf_q ovf err	r_buf_fence_dist[10:0]										
39	0	0	0	0	reverse me1b	reverse me1a	reverse csc	buf_fence_cnt[11:0]											
40	0	0	0	0	tmb trig pulse	trig_src_vec[10:9]		mxcfeb=7	peak fence	cfeb_en[6:5]		perr_cfeb[6:5]		cfeb_badbits_found[6:5		cfebs_read[6:5]		active_cfeb[6:5]	
41	0	0	0	0	layer trig enabled	lyr_thresh_pretrig[2:0]			non-trig readout	triggered readout	non-trig match ro	non-trig clct ro	non-trig alct o	allow match ro	allow clct ro	allow alct ro	allow match	allow clct	allow alct
42	0	0	0	0	6			EOB End Header Block											
43					CFEB 0			Tbin 0				Ly0[7:0] Triad bits							
44					CFEB 0			Tbin 0				Ly1[7:0]							
45					CFEB 0			Tbin 0				Ly2[7:0]							
46					CFEB 0			Tbin 0				Ly3[7:0]							
47					CFEB 0			Tbin 0				Ly4[7:0]							
48					CFEB 0			Tbin 0				Ly5[7:0]							
49					CFEB 0			Tbin 1				Ly0[7:0]							
50					CFEB 0			Tbin 1				Ly1[7:0]							
51					CFEB 0			Tbin 1				Ly2[7:0]							
52					CFEB 0			Tbin 1				Ly3[7:0]							
53					CFEB 0			Tbin 1				Ly4[7:0]							
54					CFEB 0			Tbin 1				Ly5[7:0]							
55-246					---			---				---							
247					CFEB 4			Tbin 6				Ly0[7:0]							
248					CFEB 4			Tbin 6				Ly1[7:0]							
249					CFEB 4			Tbin 6				Ly2[7:0]							

250					CFEB 4			Tbin 6										Ly3[7:0]		
251					CFEB 4			Tbin 6										Ly4[7:0]		
252					CFEB 4			Tbin 6										Ly5[7:0]		
253					6			$B04_{16}$ Begin RPC Raw Hits (if RPC readout enabled)												
254					RPC 0			Tbin 0										Pads[7:0] RPC0 Pads		
255					RPC 0	clct pretrigger		rpc_bxn[2:0]										Pads[15:8]		
256					RPC 0			Tbin 1										Pads[7:0]		
257					RPC 0	clct pretrigger		rpc_bxn[2:0]										Pads[15:8]		
258					RPC 0			Tbin 2										Pads[7:0]		
259					RPC 0	clct pretrigger		rpc_bxn[2:0]										Pads[15:8]		
260					RPC 0			Tbin 3										Pads[7:0]		
261					RPC 0	clct pretrigger		rpc_bxn[2:0]										Pads[15:8]		
262					RPC 0			Tbin 4										Pads[7:0]		
263					RPC 0	clct pretrigger		rpc_bxn[2:0]										Pads[15:8]		
264					RPC 0			Tbin 5										Pads[7:0]		
265					RPC 0	clct pretrigger		rpc_bxn[2:0]										Pads[15:8]		
266					RPC 0			Tbin 6										Pads[7:0]		
267					RPC 0	clct pretrigger		rpc_bxn[2:0]										Pads[15:8]		
268					RPC 1			Tbin 0										Pads[7:0] RPC1 Pads		
269					RPC 1	clct pretrigger		rpc_bxn[2:0]										Pads[15:8]		
270-279					--			--										--		
280					RPC 1			Tbin 6										Pads[7:0]		
281					RPC 1	clct pretrigger		rpc_bxn[2:0]										Pads[15:8]		
282					6			$E04_{16}$ End RPC Raw Hits												
283					6			$E0C_{16}$ End Cathode Data												
opt					$2AAA_{16}[14:0]$ (Optional to make word count multiple of 4)															
opt					$5555_{16}[14:0]$ (Optional to make word count multiple of 4)															
284	0	0	0	**1**	DDU Code 101_2			$E0F_{16}[11:0]$												
285	0	0	0	**1**	DDU Code 101_2			1 TMB	CRC22[10:0]											
286	0	0	0	**1**	DDU Code 101_2			1 TMB	CRC22[21:11]											
287	0	0	**1**	**1**	DDU Code 101_2			1 TMB	Word Count [10:0]											
No Write	**1**	0	0	0																

Sample TMB Raw Hits Dump

TMB internal pattern injector + RPC internal pattern injector

7-Time bins, full 5 CLCTs+ 2 RPCs raw hits readout (Blocked CFEB DiStrips list turned off)

```
Adr=   0 Data=2DB0C      Adr=  70 Data=00400      Adr= 140 Data=02200      Adr= 210 Data=03600
Adr=   1 Data=0DCC8      Adr=  71 Data=00402      Adr= 141 Data=02200      Adr= 211 Data=04000
Adr=   2 Data=0D001      Adr=  72 Data=00400      Adr= 142 Data=02200      Adr= 212 Data=04000
Adr=   3 Data=0D001      Adr=  73 Data=00500      Adr= 143 Data=02200      Adr= 213 Data=04000
Adr=   4 Data=04045      Adr=  74 Data=00500      Adr= 144 Data=02200      Adr= 214 Data=04000
Adr=   5 Data=0226A      Adr=  75 Data=00500      Adr= 145 Data=02300      Adr= 215 Data=04000
Adr=   6 Data=0777F      Adr=  76 Data=00500      Adr= 146 Data=02300      Adr= 216 Data=04000
Adr=   7 Data=0512C      Adr=  77 Data=00500      Adr= 147 Data=02300      Adr= 217 Data=04100
Adr=   8 Data=00C47      Adr=  78 Data=00500      Adr= 148 Data=02300      Adr= 218 Data=04100
Adr=   9 Data=00001      Adr=  79 Data=00600      Adr= 149 Data=02300      Adr= 219 Data=04100
Adr=  10 Data=00000      Adr=  80 Data=00600      Adr= 150 Data=02300      Adr= 220 Data=04100
Adr=  11 Data=00001      Adr=  81 Data=00600      Adr= 151 Data=02400      Adr= 221 Data=04100
Adr=  12 Data=00000      Adr=  82 Data=00600      Adr= 152 Data=02400      Adr= 222 Data=04100
Adr=  13 Data=00001      Adr=  83 Data=00600      Adr= 153 Data=02400      Adr= 223 Data=04200
Adr=  14 Data=00000      Adr=  84 Data=00600      Adr= 154 Data=02400      Adr= 224 Data=04200
Adr=  15 Data=00001      Adr=  85 Data=01000      Adr= 155 Data=02400      Adr= 225 Data=04200
Adr=  16 Data=00000      Adr=  86 Data=01000      Adr= 156 Data=02400      Adr= 226 Data=04200
Adr=  17 Data=002DD      Adr=  87 Data=01000      Adr= 157 Data=02500      Adr= 227 Data=04200
Adr=  18 Data=00000      Adr=  88 Data=01000      Adr= 158 Data=02500      Adr= 228 Data=04200
Adr=  19 Data=0023D      Adr=  89 Data=01000      Adr= 159 Data=02500      Adr= 229 Data=04300
Adr=  20 Data=04204      Adr=  90 Data=01000      Adr= 160 Data=02500      Adr= 230 Data=04300
Adr=  21 Data=01A46      Adr=  91 Data=01100      Adr= 161 Data=02500      Adr= 231 Data=04300
Adr=  22 Data=07E01      Adr=  92 Data=01100      Adr= 162 Data=02500      Adr= 232 Data=04300
Adr=  23 Data=003E1      Adr=  93 Data=01100      Adr= 163 Data=02600      Adr= 233 Data=04300
Adr=  24 Data=00301      Adr=  94 Data=01100      Adr= 164 Data=02600      Adr= 234 Data=04300
Adr=  25 Data=005AD      Adr=  95 Data=01100      Adr= 165 Data=02600      Adr= 235 Data=04400
Adr=  26 Data=00000      Adr=  96 Data=01100      Adr= 166 Data=02600      Adr= 236 Data=04400
Adr=  27 Data=0531C      Adr=  97 Data=01200      Adr= 167 Data=02600      Adr= 237 Data=04400
Adr=  28 Data=010A7      Adr=  98 Data=01200      Adr= 168 Data=02600      Adr= 238 Data=04400
Adr=  29 Data=01000      Adr=  99 Data=01200      Adr= 169 Data=03000      Adr= 239 Data=04400
Adr=  30 Data=00001      Adr= 100 Data=01200      Adr= 170 Data=03000      Adr= 240 Data=04400
Adr=  31 Data=07D0A      Adr= 101 Data=01200      Adr= 171 Data=03000      Adr= 241 Data=04500
Adr=  32 Data=02605      Adr= 102 Data=01200      Adr= 172 Data=03000      Adr= 242 Data=04500
Adr=  33 Data=00000      Adr= 103 Data=01300      Adr= 173 Data=03000      Adr= 243 Data=04500
Adr=  34 Data=00000      Adr= 104 Data=01300      Adr= 174 Data=03000      Adr= 244 Data=04500
Adr=  35 Data=07C01      Adr= 105 Data=01300      Adr= 175 Data=03100      Adr= 245 Data=04500
Adr=  36 Data=008FB      Adr= 106 Data=01300      Adr= 176 Data=03100      Adr= 246 Data=04500
Adr=  37 Data=01E25      Adr= 107 Data=01300      Adr= 177 Data=03100      Adr= 247 Data=04600
Adr=  38 Data=007FF      Adr= 108 Data=01300      Adr= 178 Data=03100      Adr= 248 Data=04600
Adr=  39 Data=00001      Adr= 109 Data=01400      Adr= 179 Data=03100      Adr= 249 Data=04600
Adr=  40 Data=06001      Adr= 110 Data=01400      Adr= 180 Data=03100      Adr= 250 Data=04600
Adr=  41 Data=02326      Adr= 111 Data=01400      Adr= 181 Data=03200      Adr= 251 Data=04600
Adr=  42 Data=06E0B      Adr= 112 Data=01400      Adr= 182 Data=03200      Adr= 252 Data=04600
Adr=  43 Data=00000      Adr= 113 Data=01400      Adr= 183 Data=03200      Adr= 253 Data=06B04
Adr=  44 Data=00000      Adr= 114 Data=01400      Adr= 184 Data=03200      Adr= 254 Data=00000
Adr=  45 Data=00000      Adr= 115 Data=01500      Adr= 185 Data=03200      Adr= 255 Data=00000
Adr=  46 Data=00000      Adr= 116 Data=01500      Adr= 186 Data=03200      Adr= 256 Data=00100
Adr=  47 Data=00000      Adr= 117 Data=01500      Adr= 187 Data=03300      Adr= 257 Data=00000
Adr=  48 Data=00000      Adr= 118 Data=01500      Adr= 188 Data=03300      Adr= 258 Data=00200
Adr=  49 Data=00100      Adr= 119 Data=01500      Adr= 189 Data=03300      Adr= 259 Data=007AB
Adr=  50 Data=00100      Adr= 120 Data=01500      Adr= 190 Data=03300      Adr= 260 Data=00301
Adr=  51 Data=00100      Adr= 121 Data=01600      Adr= 191 Data=03300      Adr= 261 Data=006AB
Adr=  52 Data=00100      Adr= 122 Data=01600      Adr= 192 Data=03300      Adr= 262 Data=00402
Adr=  53 Data=00100      Adr= 123 Data=01600      Adr= 193 Data=03400      Adr= 263 Data=005AB
Adr=  54 Data=00100      Adr= 124 Data=01600      Adr= 194 Data=03400      Adr= 264 Data=00503
Adr=  55 Data=00202      Adr= 125 Data=01600      Adr= 195 Data=03400      Adr= 265 Data=004AB
Adr=  56 Data=00202      Adr= 126 Data=01600      Adr= 196 Data=03400      Adr= 266 Data=00604
Adr=  57 Data=00202      Adr= 127 Data=02000      Adr= 197 Data=03400      Adr= 267 Data=003AB
Adr=  58 Data=00202      Adr= 128 Data=02000      Adr= 198 Data=03400      Adr= 268 Data=01000
Adr=  59 Data=00202      Adr= 129 Data=02000      Adr= 199 Data=03500      Adr= 269 Data=01000
Adr=  60 Data=00202      Adr= 130 Data=02000      Adr= 200 Data=03500      Adr= 270 Data=01100
Adr=  61 Data=00300      Adr= 131 Data=02000      Adr= 201 Data=03500      Adr= 271 Data=01000
Adr=  62 Data=00302      Adr= 132 Data=02000      Adr= 202 Data=03500      Adr= 272 Data=01200
Adr=  63 Data=00300      Adr= 133 Data=02100      Adr= 203 Data=03500      Adr= 273 Data=017CD
Adr=  64 Data=00302      Adr= 134 Data=02100      Adr= 204 Data=03500      Adr= 274 Data=01301
Adr=  65 Data=00300      Adr= 135 Data=02100      Adr= 205 Data=03600      Adr= 275 Data=016CD
Adr=  66 Data=00302      Adr= 136 Data=02100      Adr= 206 Data=03600      Adr= 276 Data=01402
Adr=  67 Data=00402      Adr= 137 Data=02100      Adr= 207 Data=03600      Adr= 277 Data=015CD
Adr=  68 Data=00400      Adr= 138 Data=02100      Adr= 208 Data=03600      Adr= 278 Data=01503
Adr=  69 Data=00402      Adr= 139 Data=02200      Adr= 209 Data=03600      Adr= 279 Data=014CD
```

```
Adr= 280 Data=01604         Adr= 283 Data=06E0C         Adr= 286 Data=0DDF2
Adr= 281 Data=013CD         Adr= 284 Data=0DE0F         Adr= 287 Data=1D920
Adr= 282 Data=06E04         Adr= 285 Data=0D94F
```

Configuration

Shunt Settings

Table 1: Shunts

#	Shunt	Default	Function
1	SH501 [CLK SRC]	1-2 [CCB]	1-2 [CCB] CCB sources TMB s 40MHz Main clock 2-3 [XTAL] Onboard crystal sources TMBs 40MHz Main clock
2	SH502 [CFEB CLK]	**2-3** [DCC]	1-2 [NRM] CFEB clocks sourced by 3D3444 64/36 duty-cycle 2-3 [DCC] CFEB clocks duty-cycle corrected
3	SH55 [GBL]	1-2 [ENA]	1-2 [ENA] Boot Register responds to TMB VME Global Address 2-3 [DIS] Boot Register ignores TMB VME Global Address
4	SH56 [GEO]	1-2 [ENA]	1-2 [ENA] Boot Register responds to VME Geographic Address 2-3 [DIS] Boot Register ignores VME Geographic Address
5	SH57 [ADM]	1-2 [REQ]	1-2 [REQ] Boot Register requires VME Address Mode 39h or 3Dh 2-3 [PAS] Boot Register accepts any VME Address Mode
6	SH97 [FPGA]	1-2 [ENA]	1-2 [ENA] Cleared Boot Register enables FPGA VME access 2-3 [DIS] Cleared Boot Register disables FPGA VME access
7	SH69-1 [REG IN]	**2-3** [REG]	1-2 [BPL] Disconnect +3.3V from U69 1.5V regulator input 2-3 [REG] Connect +3.3V to U69 1.5V regulator input
8	SH69-2 [REG OUT]	**2-3** [REG]	1-2 [BPL] Backplane sources +1.5Vtt 2-3 [REG] Onboard regulator U69 sources +1.5Vtt
9	SH95 [BOOT]	1-2 [EN]	1-2 [EN] CCB hard-reset clears Boot Register 2-3 [DIS] CCB hard-reset does not clear Boot Register
10	SH104 [VME]	1-2 [EN]	1-2 [EN] VME write to Boot Register can hard-reset TMB or ALCT 2-3 [DIS] Boot Register can not hard-reset TMB or ALCT
11	SH105 [CCB]	1-2 [EN]	1-2 [EN] CCB can hard-reset TMB or ALCT 2-3 [DIS] CCB can not hard-reset TMB or ALCT
12	SH106 [ALCT]	1-2 [EN]	1-2 [EN] TMB FPGA can hard-reset ALCT board if enabled by Boot 2-3 [DIS] TMB FPGA firmware can not hard-reset ALCT board
13	SH107 [SELF]	**2-3** [DIS]	1-2 [EN] TMB FPGA firmware can hard-reset TMB 2-3 [DIS] TMB FPGA firmware can not hard-reset TMB
14	SH1081 [ALCT]	1-2 [EN]	1-2 [EN] TMB can send hard-reset to ALCT board 2-3 [DIS] TMB can not send hard-reset to ALCT board
15	SH1082 [TMB]	1-2 [EN]	1-2 [EN] TMB can send hard-reset to TMB 2-3 [DIS] TMB can not send hard-reset to TMB
16	SH1083 [RPC]	1-2 [EN]	1-2 [EN] TMB can send hard-reset to RAT/RPC board 2-3 [DIS] TMB can not send hard-reset to RAT/RPC board
17	SH1084 [PUP]	1-2 [EN]	1-2 [EN] Issue hard-reset to TMB on power-up 2-3 [DIS] Do not issue hard-reset to TMB on power-up

18	SH62 [VMEADR]	1-2 [GEO]	1-2 [GEO]	Slot Address derived from VME backplane GEO pins
			2-3 [LCL]	Slot Address derived from SW2/SW1 hex switches
19	SH74 [JTAG SRC]	**2-3** [SW3]	1-2 [XBL]	X-blaster board sources JTAG chain address
			2-3 [SW3]	Onboard SW3 hex switch sources JTAG chain address
20	SH921 [RAT ADC]	1-2 [ACORE]	1-2 [ACORE]	ADC Ch9 measures RAT core current
			2-3 [+3.3VR]	ADC Ch9 measures RAT main +3.3V supply

Switch Settings

Table 2: Switch Settings

Switch	Module	Default	Function
SW1	VME	A	SW2/SW1 Form the VME slot address for Local Mode addressing and for the Boot Register Global Address. Normally, SH62 [VMEADR] will be set to 1-2 [GEO] and the VME Slot-Address is determined by the Slot-ID signals from the VME P1 backplane connector. SW2/SW1 should be set to 1A [26 decimal] to specify the Boot Register Global Address. SW2 is the most-significant hex digit. If SH62 [VMEADR] is set 2-3 [LCL], hex rotary switches SW2/SW1 determine the VME slot address for the module, and the P1 Geographic Address signals are ignored.
SW2	VME	1	
SW3	JTAG	4	Selects the JTAG Chain Address for the Xblaster when SH74 [JTAG ADR SRC] is set to 2-3 [SW3]. If SH74 is set 1-2 [XBL], the JTAG Chain Address is determined by the Xblaster. 0000 0 ALCT SLOW USER 0001 1 ALCT SLOW PROM 0010 2 ALCT FPGA USER 0011 3 ALCT FPGA PROMs 01XX 4 TMB FPGA PROMs 10XX 8 TMB User PROMs 1100 C TMB FPGA Loop (for TMB self-test) 1101 D RAT FPGA PROM

Fuses

Table 3: Fuses for TMB2005

Fuse	Circuit	Amps	LED	Function	Littelfuse Part #
F1	+5.0V	5A	D_F1 Red	VME-bus I/O	154005.DR
F2	+3.3V	10A	D_F2 Red	Main board logic, FPGA IOBs	154010.DR
F3	+1.5V	5A	None	GTLP termination	154005.DR
F4	+3.3VXB	1A	D_F4 Grn	Xblaster power	154001.DR

Table 4A: Fuses for TMB2013

Fuse	Circuit	Amps	LED	Function	Littelfuse Part #
F1	+5.0V	10A	D_F1 Red	VME-bus I/O	0451010.MRL
F2	+3.3V	15A	D_F2 Red	Main board logic, FPGA IOBs	0451015.MRL
F3	+1.5V	10A	None	GTLP termination	0451010.MRL
F4	+3.3VXB	1A	D_F4 Grn	Xblaster power	154001.DR

Signal Summary

CCB

1 Input LVDS 40MHz clock
46 Inputs GTLP at 40MHz
46 Outputs GTLP at 40MHz

Table 5: CCB Signal Summary

Signal	Bits	Dir	Logic	Function
Clock Bus				
ccb_clock40	1	In	LVDS	
Total	1			
Fast Control Bus				
ccb_clock40_enable	1	In	GTLP	
ccb_cmd[5..0]	6	In	GTLP	
ccb_evcntres	1	In	GTLP	
ccb_bcntres	1	In	GTLP	
ccb_cmd_strobe	1	In	GTLP	
ccb_bx0	1	In	GTLP	
ccb_l1accept	1	In	GTLP	
ccb_data[7..0]	8	In	GTLP	
ccb_data_strobe	1	In	GTLP	
ccb_reserved[4..0]	5	In	GTLP	
Total	26			
TMB Reload Bus: ALCT+CLCT+TMB FPGA Reload				
tmb_hard_reset	1	In	GTLP	
tmb_cfg_done[8..0]	9	Out	GTLP	
alct_hard_reset	1	In	GTLP	
alct_cfg_done[8..0]	9	Out	GTLP	
tmb_reserved[1..0]	2	In	GTLP	
Total	22			
DAQ Special Purpose Bus [Used by DMB and TMB]				
dmb_cfeb_calibrate[2..0]	3	In	GTLP	
dmb_l1a_release	1	(In)	GTLP	
dmb_reserved_out[4..0]	5	In	GTLP	
dmb_reserved_in[2..0]	3	(In)	GTLP	
Total	12			
Trigger Special Purpose Bus [Used by TMB only]				
alct_adb_pulse_sync	1	In	GTLP	
alct_adb_pulse_async	1	In	GTLP	
clct_external_trigger	1	In	GTLP	
alct_external_trigger	1	In	GTLP	
clct_status[8..0]	9	Out	GTLP	
alct_status[8..0]	9	Out	GTLP	
tmb_l1a_request	1	Out	GTLP	
tmb_l1a_release	1	Out	GTLP	
tmb_reserved_in[4..0]	5	Out	GTLP	
tmb_reserved_out[2..0]	3	In	GTLP	
Total	32			

ALCT

29 Inputs LVDS Multiplexed at 80 MHz
20 Outputs LVDS Multiplexed at 80 MHz
1 Output LVDS 40MHz clock

Table 6: ALCT Signal Summary

Signal	Bits	Pins	Mux	Dir	Logic	Function
first_valid	1	0.5	Yes	In	LVDS	Valid Pattern Flag, best muon
first_quality[1..0]	2	1	Yes	In	LVDS	Pattern Quality, best muon
first_amu	1	0.5	Yes	In	LVDS	Accelerator Muon Flag, best muon
first_key[6..0]	7	3.5	Yes	In	LVDS	Key Wire Group, best muon
second_valid	1	0.5	Yes	In	LVDS	Valid Pattern Flag, 2nd best muon
second_quality[1..0]	2	1	Yes	In	LVDS	Pattern Quality, 2nd best muon
second_amu	1	0.5	Yes	In	LVDS	Accelerator Muon Flag, 2nd best muon
second_key[6..0]	7	3.5	Yes	In	LVDS	Key Wire Group, 2nd best muon
bxn[4..0]	5	2.5	Yes	In	LVDS	Bunch Crossing Number
daq_data[13..0]	14	7	Yes	In	LVDS	DAQ data
/wr_fifo	1	0.5	Yes	In	LVDS	/Write_enable DAQ FIFO
lct_special	1	0.5	Yes	In	LVDS	LCT Special Word Flag
ddu_special	1	0.5	Yes	In	LVDS	DAQ Special Word Flag
first_frame	1	0.5	Yes	In	LVDS	First DAQ Frame
last_frame	1	0.5	Yes	In	LVDS	Last DAQ Frame
seq_status[1..0]	2	1	Yes	In	LVDS	Sequencer Status
seu_status[1..0]	2	1	Yes	In	LVDS	Radiation SEU Status
active_feb_flag	1	0.5	Yes	In	LVDS	Active FEB Flag (ALCT pre-triggered)
cfg_done	1	0.5	Yes	In	LVDS	FPGA configuration done
reserved_out[3..0]	4	2	Yes	In	LVDS	Future use
tdo	1	1	No	In	LVDS	JTAG tdo from ALCT
Total Inputs	57	29				

Signal	Bits	Pins	Mux	Dir	Logic	Function
ccb_brcst[7..0]	8	4	Yes	Out	LVDS	CCB broadcast command
brcst_str	1	0.5	Yes	Out	LVDS	ccb_brcst strobe
dout_str	1	0.5	Yes	Out	LVDS	ccb_dout strobe
subadr_str	1	0.5	Yes	Out	LVDS	ccb_subaddr strobe
bx0	1	0.5	Yes	Out	LVDS	Bunch Crossing Zero
ext_inject	1	0.5	Yes	Out	LVDS	External Test Pattern Inject Command
ext_trig	1	0.5	Yes	Out	LVDS	External Trigger Command
level1_accept	1	0.5	Yes	Out	LVDS	Level 1 Accept
sync_adb_pulse	1	0.5	Yes	Out	LVDS	Synchronous ADB Test Pulse
seq_cmd[2..0]	3	1.5	Yes	Out	LVDS	Sequencer Command
reserved_in[4..0]	5	2.5	Yes	Out	LVDS	Future use
clock	1	1	No	Out	LVDS	40MHz clock
clock_en	1	1	No	Out	LVDS	Clock enable
hard_reset	1	1	No	Out	LVDS	FPGA Reload Command
async_adb_pulse	1	1	No	Out	LVDS	Asynchronous ADB Test Pulse
jtag_select[1..0]	2	2	No	Out	LVDS	JTAG Chain Select
tck	1	1	No	Out	LVDS	JTAG tck to ALCT
tms	1	1	No	Out	LVDS	JTAG tms to ALCT

tdi	1	1	No	Out	LVDS	JTAG tdi to ALCT
Total Outputs	33	21				

DMB

3 Inputs LVTTL at 40 MHz
45 Outputs[1] LVTTL at 40 MHz (see note[1] below)
47 Outputs[2] LVTTL at 40 MHz (see note[2] below)

Table 7: DMB Signal Summary

Signal	Bits	Dir	Logic	Function
tmb_data[14:0]	15	Out	LVTTL	TMB data[14:0] to DMB FIFO
alct_data[14:0]	15	Out	LVTTL	ALCT data [14:0] to DMB FIFO
tmb_ddu_special	1	Out	LVTTL	TMB DDU Special Word Flag
tmb_last_frame	1	Out	LVTTL	TMB Last FIFO frame
tmb_data_available	1	Out	LVTTL	TMB Data Available
/tmb_write_enable_fifo	1	Out	LVTTL	TMB FIFO /write_enable
tmb_active_feb_flag	1	Out	LVTTL	TMB Active Front-End-Board Flag
tmb_active_feb[4…0] [1]	5	Out	LVTTL	TMB Active FEB indicators[4...0] [1]
tmb_active_feb[6…0] [2]	7	Out	LVTTL	TMB Active FEB indicators[6...0] [2]
fifo_clock	1	Out	LVTTL	40MHz FIFO storage clock [= tmb_clock]
alct_ddu_special	1	Out	LVTTL	ALCT DDU Special Word Flag
alct_last_frame	1	Out	LVTTL	ALCT Last FIFO frame
alct_first_frame(dav)	1	Out	LVTTL	ALCT First FIFO frame, data available
/alct_write_enable_fifo	1	Out	LVTTL	ALCT FIFO /write_enable
reserved_to_dmb	5	Out	LVTTL	Unassigned
Total Outputs	57			
dmb_request_lct	1	In	LVTTL	DMB requests active_feb_flag from TMB
dmb_ext_trig	1	In	LVTTL	DMB external trigger to TMB
dmb_fpga_pgm_done	1	In	LVTTL	DMB FPGA Program Done
reserved_from_dmb	3	In	LVTTL	Unassigned
Total Inputs	6			

[1] For TMB 2005 operation with up to 5 CFEBs.

[2] For TMB 2013 operation with 7 DCFEBs.

CFEB

120 Inputs LVDS data multiplexed at 80 MHz
5 Outputs LVDS 40MHz clock

Table 8: CFEB Signal Summary

Signal	Bits	Pins	Dir	Logic	Function
cfeb0_ly[5..0]_tr[7..0]	48	24	In	LVDS	CFEB0 6 layers x 8 triads, 80MHz
cfeb1_ly[5..0]_tr[7..0]	48	24	In	LVDS	CFEB1 6 layers x 8 triads, 80MHz
cfeb2_ly[5..0]_tr[7..0]	48	24	In	LVDS	CFEB2 6 layers x 8 triads, 80MHz
cfeb3_ly[5..0]_tr[7..0]	48	24	In	LVDS	CFEB3 6 layers x 8 triads, 80MHz
cfeb4_ly[5..0]_tr[7..0]	48	24	In	LVDS	CFEB4 6 layers x 8 triads, 80MHz
Total Inputs	240	120			

Signal	Bits	Pins	Dir	Logic	Function
cfeb0_lct_clock	1	1	Out	LVDS	CFEB 1 40MHz clock
cfeb1_lct_clock	1	1	Out	LVDS	CFEB 2 40MHz clock
cfeb2_lct_clock	1	1	Out	LVDS	CFEB 3 40MHz clock
cfeb3_lct_clock	1	1	Out	LVDS	CFEB 4 40MHz clock
cfeb4_lct_clock	1	1	Out	LVDS	CFEB 5 40MHz clock
Total Outputs	5	5			

MPC

1 Input GTLP at 80MHz
32 Outputs GTLP at 80MHz

Table 9: MPC Signal Summary

	Signal	Bits	Pins	Dir	Logic	Function
First In Time	alct_first_key[6:0]	7	3.5	Out	GTLP	LCT 0 ALCT key wire-group
	clct_first_pat[3:0]	4	2	Out	GTLP	LCT 0 CLCT pattern number
	lct_first_quality[3:0]	4	2	Out	GTLP	LCT 0 Muon quality
	first_vpf	1	0.5	Out	GTLP	LCT 0 Valid pattern flag
	alct_second_key[6:0]	7	3.5	Out	GTLP	LCT 1 ALCT key wire-group
	clct_second_pat[3:0]	4	2	Out	GTLP	LCT 1 CLCT pattern number
	lct_second_quality[3:0]	4	2	Out	GTLP	LCT 1 Muon quality
	second_vpf	1	0.5	Out	GTLP	LCT 1 Valid pattern flag
Second In Time	clct_first_key[7:0]	8	4	Out	GTLP	LCT 0 CLCT key ½-strip
	clct_first_bend	1	0.5	Out	GTLP	LCT 0 CLCT bend direction
	lct 0 sync_err	1	0.5	Out	GTLP	LCT 0 BXN does not match at BX0
	alct_first_bxn[0]	1	0.5	Out	GTLP	LCT 0 ALCT bunch crossing number
	clct_first_bx0_local	1	0.5	Out	GTLP	LCT 0 local BXN from CLCT
	csc_id[3:0]	4	2	Out	GTLP	CSC chamber ID
	clct_second_key[7:0]	8	4	Out	GTLP	LCT 1 CLCT key ½-strip
	clct_second_bend	1	0.5	Out	GTLP	LCT 1 CLCT bend direction
	lct 1 sync_err	1	0.5	Out	GTLP	LCT 1 BXN does not match at BX0
	alct_second_bxn[0]	1	0.5	Out	GTLP	LCT 1 ALCT bunch crossing number
	clct_second_bx0_local	1	0.5	Out	GTLP	LCT 1 local BXN from CLCT
	csc_id[3:0]	4	2	Out	GTLP	CSC chamber ID
	Total Output Signals	37	18.5			2:1 Multiplexing at 80 MHz

Time	Signal	Bits	Pins	Dir	Logic	Function
1st	lct0_accept	1	0.5	In	GTLP	LCT 0 Accepted by MPC best 3 of 18 sort
2nd	lct1_accept	1	0.5	In	GTLP	LCT 1 Accepted by MPC best 3 of 18 sort
	Total Input Signals	2	1			2:1 Multiplexing at 80 MHz

RPC

80 Inputs 40MHz LVTTL [from receivers on transition module]
0 Outputs

Table 10: RPC Signal Summary

Signal	Bits	Dir	Logic	Function
rpc0_seg[15..0]	16	In	LVTTL	RPC Segment[15..0]
rpc0_bxn[2..0]	3	In	LVTTL	Bunch Crossing Number [2..0]
rpc0_clock	1	In	LVTTL	Clock from RPC Link Board
rpc1_seg[15..0]	16	In	LVTTL	RPC Segment[15..0]
rpc1_bxn[2..0]	3	In	LVTTL	Bunch Crossing Number [2..0]
rpc1_clock	1	In	LVTTL	Clock from RPC Link Board
Total data bits	40			

VME

24 Inputs TTL Address
16 BiDir TTL Data

Table 11: VME Signal Summary

Signal	Bits	Dir	Logic	Function
address	24	In	TTL	VME Address[23..0]
data	16	BiDir	TTL	VME Data[15..0]
control in		In	TTL	VME Control Inputs
control out		Out	TTL	VME Control Outputs

JTAG

5 Inputs LVDS
1 Outputs LVDS

Table 12: JTAG Signal Summary

Signal	Bits	Dir	Logic	Function
tck	1	In	LVDS	JTAG TCK
tms	1	In	LVDS	JTAG TMS
tdi	1	In	LVDS	JTAG TDI
chain_select	2	In	LVDS	Chain Select Address
tdo	1	Out	LVDS	JTAG TDO

LEDs & Testpoints

Table 13: TMB Front Panel LEDs

LED Label	Color	Function
LCT	Blue	ALCT vpf and CLCT vpf match within the 75ns ALCT window (see note[3] below)
ALCT	Green	ALCT active FEB flag (may not always be the same as ALCT valid pattern flag)
CLCT	Green	CLCT active FEB flag Or any external trigger except ALCT
L1A	Green	Level 1 Accept arrived in L1A window
INVP	Yellow	Invalid CLCT pattern after drift delay Pattern dropped below threshold, probably triggered on noise
NMAT	Yellow	ALCT vpf or CLCT vpf arrived but did not match in ALCT window
NL1A	Red	No Level 1 Accept arrived in L1A window after TMB triggered Constant flash rate = buffers full
VME	Green	ON = TMB FPGA loaded successfully from PROM Flashes OFF when module addressed by VME

Table 12A: Mez-2013 SMT LEDs

LED Label	Color	Function
D1	Blue	When lit, indicates at least one fiber link had an error since last reset
D2	Green	When lit, indicates TMB clock0 MMCM has locked since the last reset
D3	Yellow	When lit, indicates TMB clock0 MMCM is not currently locked
D4	Red	When lit, indicates TMB clock0 MMCM lost lock at least once since the last reset
D5	Green	When lit, indicates the QPLL has locked since the last reset
D6	Yellow	When lit, indicates the QPLL is not currently locked
D7	Red	When lit, indicates the QPLL lost lock at least once since the last reset
D8	Green	When lit, indicates at least one fiber link has a stable input
D0	Yellow	When lit, indicated the FPGA is Not Programmed (i.e. DONE is False)

Table 12B: Mez-2013 Testpoints

Testpoint Label	Function
9	TRUE indicates at least one fiber link had over 100 errors since the last reset
8	TRUE indicates all fiber links have a stable input
7:1	TRUE signas indicate that links 7:1 have a stable input, respectively

[3] NB: All external triggers (including scintillator and ALCT active FEB) create a dummy CLCT to force the TMB to read out raw hits. ALCT triggers can produce an LCT match if the ALCT active FEB is followed by an ALCT valid pattern flag.

TMB Total I/O Count

Table 14: TMB Total I/O Count

Bits	Pins	Dir	Logic	Connect To	Function
1	2	In	LVDS	CCB	Clock Bus
26	26	In	GTLP	CCB	Fast Control Bus
4	4	In	GTLP	CCB	TMB Reload Bus
12	12	In	GTLP	CCB + 9 DMB + 9 TMB	DAQ Special Purpose Bus
7	7	In	GTLP	CCB + 9 TMB	Trigger Special Purpose Bus
240	120	In	LVDS	5 CFEBs	CFEB Comparators
57	29	In	LVDS	ALCT	ALCT Module
6	6	In	LVTTL	1 DMB	DMB commands
2	1	In	GTLP	MPC	MPC winner
80	40	In	LVTTL	RPC	RPC Inputs
24	24	In	TTL	VME	VME Address
16	16	BiDir	TTL	VME	VME Data
5	5	In	LVTTL	JTAG	JTAG
480	292				Totals

Bits	Pins	Dir	Logic	Connect To	Function
0	0	Out	LVDS	CCB	Clock Bus
0	0	Out	GTLP	CCB	Fast Control Bus
18	18	Out	GTLP	CCB	TMB Reload Bus
0	0	Out	GTLP	CCB + 9 DMB + 9 TMB	DAQ Special Purpose Bus
25	25	Out	GTLP	CCB + 9 TMB	Trigger Special Purpose Bus
5	5	Out	LVDS	5 CFEBs	CFEB Comparators
33	21	Out	LVDS	ALCT	ALCT Module
50	50	Out	LVTTL	1 DMB	DMB data
64	32	Out	GTLP	MPC	MPC winner
0	0	Out	LVTTL	RPC	RPC Inputs
0	0	Out	TTL	VME	VME data is BiDir
1	1	Out	LVTTL	JTAG	JTAG
196	152				Totals

FPGA I/O Estimate: 392 in – 7 clock + 152 out = 537

Connectors

TMB Connector Summary

Table 15: TMB2005 Connector Summary

ID	Pins	Type	Function
J0	50	SCSI-II	CFEB0 Inputs + Clock Out
J1	50	SCSI-II	CFEB1 Inputs + Clock Out
J2	50	SCSI-II	CFEB2 Inputs + Clock Out
J3	50	SCSI-II	CFEB3 Inputs + Clock Out
J4	50	SCSI-II	CFEB4 Inputs + Clock Out
J5	50	SCSI-II	ALCT Cable 1 Inputs
J6	50	SCSI-II	ALCT Cable 2 I/O
J7	10	Header	Xilinx LVDS X-Blaster I/O
P1	160	VME64x	VME J1/P1 Bus I/O
P2A	125	Z-Pack 25x5	CCB + DMB I/O
P2B	55	Z-Pack 11x5	DMB I/O
P3A	55	Z-Pack 11x5	MPC I/O
P3B	125	Z-Pack 25x5	RPC Inputs + ALCT alternate I/O
MTP Rx	-	aqua MTP with reduced flange & SC footprint	10 Gbps panel-mount adapter to receive up to 12 fiber links carrying DCFEB comparator data to Mez-2013 boards

J0-J4 CFEB0-CFEB4 Connectors

Function: Receives 80MHz data from CFEBs. Transmits 40MHz clock.

Connector Type:
PCB: AMP 787190-5
Cable: AMP 749111-4
Shell: AMP 749889-3 [with latches]

Table 16: J0-J4 CFEB0/4-to-TMB Connectors
(This table uses Layer numbers Ly0-to-Ly5, Triad numbers Tr0-to-Tr7)

Pair	Pin		Dir	Logic	Multiplexed Signals	
1	1+	2-	In	LVDS	Ly0Tr0	Ly3Tr0
2	3+	4-	In	LVDS	Ly0Tr2	Ly3Tr2
3	5+	6-	In	LVDS	Ly5Tr0	Ly4Tr0
4	7+	8-	In	LVDS	Ly5Tr2	Ly4Tr2
5	9+	10-	In	LVDS	Ly1Tr0	Ly2Tr0
6	11+	12-	In	LVDS	Ly1Tr2	Ly2Tr2
7	13+	14-	In	LVDS	Ly0Tr4	Ly3Tr4
8	15+	16-	In	LVDS	Ly0Tr6	Ly3Tr6
9	17+	18-	In	LVDS	Ly5Tr4	Ly4Tr4
10	19+	20-	In	LVDS	Ly5Tr6	Ly4Tr6
11	21+	22-	In	LVDS	Ly1Tr4	Ly2Tr4
12	23+	24-	In	LVDS	Ly1Tr6	Ly2Tr6
13	25+	26-	Out	LVDS	LCT_Clock	
14	27+	28-	In	LVDS	Ly1Tr7	Ly2Tr7
15	29+	30-	In	LVDS	Ly1Tr5	Ly2Tr5
16	31+	32-	In	LVDS	Ly5Tr7	Ly4Tr7
17	33+	34-	In	LVDS	Ly5Tr5	Ly4Tr5
18	35+	36-	In	LVDS	Ly0Tr7	Ly3Tr7
19	37+	38-	In	LVDS	Ly0Tr5	Ly3Tr5
20	39+	40-	In	LVDS	Ly1Tr3	Ly2Tr3
21	41+	42-	In	LVDS	Ly1Tr1	Ly2Tr1
22	43+	44-	In	LVDS	Ly5Tr3	Ly4Tr3
23	45+	46-	In	LVDS	Ly5Tr1	Ly4Tr1
24	47+	48-	In	LVDS	Ly0Tr3	Ly3Tr3
25	49+	50-	In	LVDS	Ly0Tr1	Ly3Tr1

J5 ALCT Cable1 Connector (Receiver)

Function: Receives 80MHz data from ALCT.

Connector Type: PCB: AMP 787190-5
Cable: AMP 749111-4
Shell: AMP 749889-3 [with latches]

Table 17: J5 ALCT Cable1 Connector [J10 on ALCT board]

Modified 4/12/01 to match ALCT2001 PCB. Stinking bad signal inversion = ☹

Pair	Inverted	Pin		Dir	Logic	Multiplexed Signals	
		+	-			First in Time	Second in Time
1	☹	1+	2-	In	LVDS	first_valid	second_valid
2		49+	50-	In	LVDS	first_amu	second_amu
3	☹	3+	4-	In	LVDS	first_quality0	second_quality0
4		47+	48-	In	LVDS	first_quality1	second_quality1
5	☹	5+	6-	In	LVDS	first_key0	second_key0
6		45+	46-	In	LVDS	first_key1	second_key1
7	☹	7+	8-	In	LVDS	first_key2	second_key2
8		43+	44-	In	LVDS	first_key3	second_key3
9	☹	9+	10-	In	LVDS	first_key4	second_key4
10		41+	42-	In	LVDS	first_key5	second_key5
11	☹	11+	12-	In	LVDS	first_key6	second_key6
12		39+	40-	In	LVDS	bxn0	bxn3
13	☹	13+	14-	In	LVDS	bxn1	bxn4
14		37+	38-	In	LVDS	bxn2	/wr_fifo
15	☹	15+	16-	In	LVDS	daq_data0	daq_data7
16		35+	36-	In	LVDS	daq_data1	daq_data8
17	☹	17+	18-	In	LVDS	daq_data2	daq_data9
18		33+	34-	In	LVDS	daq_data3	daq_data10
19	☹	19+	20-	In	LVDS	daq_data4	daq_data11
20		31+	32-	In	LVDS	daq_data5	daq_data12
21	☹	21+	22-	In	LVDS	daq_data6	daq_data13
22		29+	30-	In	LVDS	lct_special	first_frame
23	☹	23+	24-	In	LVDS	parity_out0 seq_status0	parity_out2 seu_status0
24		27+	28-	In	LVDS	parity_out1 seq_status1	parity_out3 seu_status1
25	☹	25+	26-	In	LVDS	ddu_special	last_frame

J6 ALCT Cable2 Connector (Transmitter)

Function: Sends/Receives 80MHz data to/from ALCT.

Connector Type:
PCB: AMP 787190-5
Cable: AMP 749111-4
Shell: AMP 749889-3 [with latches]

Table 18: J6 ALCT Cable2 Connector [J11 on ALCT board]

Modified 4/12/01 to match ALCT2001 PCB. Stinking bad signal inversion ☹

Pair	Inverted	Pin		Dir	Logic	Multiplexed Signals	
		+	-			First in Time	Second in Time
1		1+	2-	Out	LVDS	tdi	
2	☹	49+	50-	Out	LVDS	tms	
3		3+	4-	Out	LVDS	tck	
4	☹	47+	48-	Out	LVDS	jtag_select0	
5		5+	6-	Out	LVDS	jtag_select1	
6	☹	45+	46-	Out	LVDS	ccb_brcst0	ccb_brcst4
7		7+	8-	Out	LVDS	ccb_brcst1	ccb_brcst5
8	☹	43+	44-	Out	LVDS	ccb_brcst2	ccb_brcst6
9		9+	10-	Out	LVDS	ccb_brcst3	ccb_brcst7
10	☹	41+	42-	Out	LVDS	brcst_str1	subaddr_str
11		11+	12-	Out	LVDS	dout_str	bx0
12	☹	39+	40-	Out	LVDS	ext_inject	ext_trig
13		13+	14-	Out	LVDS	level1_accept	sync_adb_pulse
14	☹	37+	38-	Out	LVDS	seq_cmd0	seq_cmd2
15		15+	16-	Out	LVDS	seq_cmd1	seq_cmd3 reserved_in4
16	☹	35+	36-	Out	LVDS	parity_in0 reserved_in0[4]	parity_in2 reserved_in2
17		17+	18-	Out	LVDS	parity_in1 reserved_in1	parity_in3 reserved_in3
18	☹	33+	34-	Out	LVDS	async_adb_pulse	
19		19+	20-	Out	LVDS	/hard_reset	
20	☹	31+	32-	Out	LVDS	clock_en	
21		21+	22-	Out	LVDS	clock	
22		29+	30-	In	LVDS	tdo	
23	☹	23+	24-	In	LVDS	parity_out4 reserved_out0	parity_out6 reserved_out2
24		27+	28-	In	LVDS	parity_out5 reserved_out1	alct_bx0 reserved_out3
25	☹	25+	26-	In	LVDS	active_feb_flag	cfg_done

[4] Reserved cable input signals connect to ALCT FPGA user input pins

J1-J6 SCSI-II 50-Pin Connector Pin Convention

Figure 5: 50-Pin PCB Connector (Female)[5]

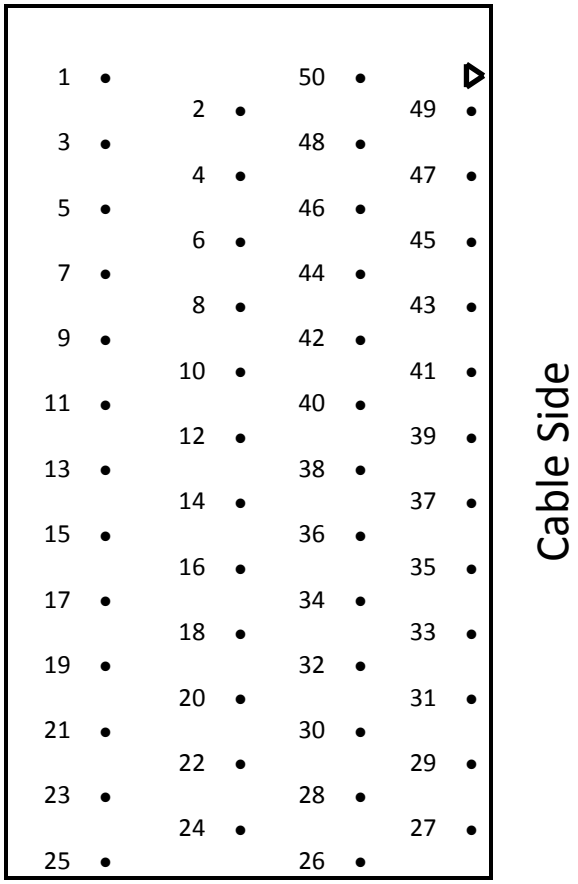
(Looking into PCB Connector)

Figure 6: 50 Pin Cable Connector (Male)

(Looking into Cable Connector)

Figure 7: 50 Pin PCB Connector Pin Convention

(Looking At Top of PCB)



[5] Copied from CFEB design: http://www.physics.ohio-state.edu/~gujh/works/cmpdata.html

J7 Xilinx LVDS Xilinx X-Blaster Connector

Function: Connects TMB2005 to LVDS x-Blaster for programming FPGAs and PROMs.
The LVDS signals and voltage sources on this connector are not directly compatible with the standard Xilinx programming cable.
JTAG chain select signals SEL[3:0] are TTL ☹.

Connector Type:

PCB:	3M 3316-5002	16-pin right angle center key
Cable:	3M 3452-6600	16-pin center bump

Table 19: J8 Xilinx LVDS X-Blaster Connector

+TCK	In	1		2	In	-TCK
+TDO	Out	3		4	Out	-TDO
+TMS	In	5		6	In	-TMS
+3.3V	Out	7		8	-	GND
+TDI	In	9		10	In	-TDI
+3.3V	In	11		12	In	JTAG_EN(TTL)
SEL0 (TTL)	Out	13		14	In	SEL1 (TTL)
SEL2 (TTL)	In	15		16	In	SEL3 (TTL)

P1 Backplane VME64x J1/P1 Connector

Function: VME interface.

Connector Type: PCB: Harting 02-02-160-2101 Male Right-Angle
Backplane: Harting 02-01-160-2201 Female
Address bits: 24
Data bits: 16
Geographic Address bits: 5

Table 20: P1 VME64x Connector

Pin	Row z	Row a	Row b	Row c	Row d
1	MPR	D00	BBSY*	D08	VPC
2	GND	D01	BCLR*	D09	GND
3	MCLK	D02	ACFAIL*	D10	+V1
4	GND	D03	BG0IN*	D11	+V2
5	MSD	D04	BG0OUT*	D12	RsvU
6	GND	D05	BG1IN*	D13	-V1
7	MMD	D06	BG1OUT*	D14	-V2
8	GND	D07	BG2IN*	D15	RsvU
9	MCTL	GND	BG2OUT*	GND	GAP*
10	GND	SYSCLK	BG3IN*	SYSFAIL*	GA0*
11	RESP*	GND	BG3OUT*	BERR*	GA1*
12	GND	DS1*	BR0*	SYSRESET*	+3.3V
13	RsvBus1	DS0*	BR1*	LWORD*	GA2*
14	GND	WRITE*	BR2*	AM5	+3.3V
15	RsvBus2	GND	BR3*	A23	GA3*
16	GND	DTACK*	AM0	A22	+3.3V
17	RsvBus3	GND	AM1	A21	GA4*
18	GND	AS*	AM2	A20	+3.3V
19	RsvBus4	GND	AM3	A19	RsvBus11
20	GND	IACK*	GND	A18	+3.3V
21	RsvBus5	IACKIN*	SERCLK	A17	RsvBus12
22	GND	IACKOUT*	SERDAT	A16	+3.3V
23	RsvBus6	AM4	GND	A15	RsvBus13
24	GND	A07	IRQ7*	A14	+3.3V
25	RsvBus7	A06	IRQ6*	A13	RsvBus14
26	GND	A05	IRQ5*	A12	+3.3V
27	RsvBus8	A04	IRQ4*	A11	LI/I*
28	GND	A03	IRQ3*	A10	+3.3V
29	RsvBus9	A02	IRQ2*	A09	LI/O*
30	GND	A01	IRQ1*	A08	+3.3V
31	RsvBus10	-12V	+5VSTDBY	+12V	GND
32	GND	+5V	+5V	+5V	VPC

P2A Backplane CCB+DMB Connector

Function: Sends and receives data to/from CCB, and carries some DMB signals.

Connector Type: PCB: AMP Z-Pack 125 (25 rows of 5 pins) female AMP 100145-1
Backplane: AMP Z-Pack 125 (25 rows of 5 pins) male AMP ?

Table 21: P2A Backplane CCB+DMB Connector

Pin	Dir	Logic	Signal
A1	In	LVDS	ccb_clock40+
A2	In	GTLP	ccb_clock40_enable
A3	In	GTLP	ccb_cmd0
A4	In	GTLP	ccb_cmb4
A5	In	GTLP	ccb_cmd_strobe
A6	In	GTLP	ccb_data0
A7	In	GTLP	ccb_data4
A8	In	GTLP	ccb_reserved0
A9	In	GTLP	tmb_hard_reset
A10	In	GTLP	alct_adb_pulse_sync
A11	Out	GTLP	clct_status0
A12	Out	GTLP	clct_status4
A13	Out	GTLP	clct_status6
A14	Out	GTLP	alct_status3
A15	Out	GTLP	alct_status7
A16	Out	GTLP	tmb_reserved_in0
A17	Out	GTLP	tmb_reserved_in4
A18			
A19	In	GTLP	dmb_cfeb_calibrate0
A20	In	GTLP	dmb_reserved_out0
A21	In	GTLP	dmb_reserved_out4
A22			
A23	Out	LVTTL	tmb_data0
A24	Out	LVTTL	tmb_data4
A25	Out	LVTTL	tmb_data8

Pin	Dir	Logic	Signal
B1	In	LVDS	ccb_clock40-
B2	In	GTLP	ccb_reserved4
B3	In	GTLP	ccb_cmd1
B4	In	GTLP	ccb_cmb5
B5	In	GTLP	ccb_bx0
B6	In	GTLP	ccb_data1
B7	In	GTLP	ccb_data5
B8	In	GTLP	ccb_reserved1
B9	In	GTLP	alct_hard_reset
B10	In	GTLP	alct_adb_pulse_async
B11	Out	GTLP	clct_status1
B12	Out	GTLP	clct_status5
B13	Out	GTLP	alct_status0
B14	Out	GTLP	alct_status4
B15	Out	GTLP	alct_status8
B16	Out	GTLP	tmb_reserved_in1
B17	In	GTLP	tmb_reserved_out0
B18			
B19	In	GTLP	dmb_cfeb_calibrate1
B20	In	GTLP	dmb_reserved_out1
B21	(In)[1]	GTLP	dmb_reserved_in0
B22			
B23	Out	LVTTL	tmb_data1
B24	Out	LVTTL	tmb_data5
B25	Out	LVTTL	tmb_data9

Pins C1 through C25 are connected to Backplane Ground

Notes:

4) TMB can monitor signals to/from DMB, but can not assert them.

P2A Backplane CCB+DMB Connector Continued

Table 20: P2A Backplane CCB Connector Continued

Pin	Dir	Logic	Signal
D1	Out	GTLP	tmb_config_done
D2			
D3	In	GTLP	ccb_cmd2
D4	In	GTLP	ccb_evcntres
D5	In	GTLP	ccb_l1accept
D6	In	GTLP	ccb_data2
D7	In	GTLP	ccb_data6
D8	In	GTLP	ccb_reserved2
D9	In	GTLP	tmb_reserved0
D10	In	GTLP	clct_external_trigger
D11	Out	GTLP	clct_status2
D12	Out	GTLP	clct_status6
D13	Out	GTLP	clct_status1
D14	Out	GTLP	alct_status5
D15	Out	GTLP	tmb_l1a_request
D16	Out	GTLP	tmb_reserved_in2
D17	In	GTLP	tmb_reserved_out1
D18			
D19	In	GTLP	dmb_cfeb_calibrate2
D20	In	GTLP	dmb_reserved_out2
D21	(In)[1]	GTLP	dmb_reserved_in1
D22			
D23	Out	LVTTL	tmb_data2
D24	Out	LVTTL	tmb_data7
D25	Out	LVTTL	tmb_data11

Pin	Dir	Logic	Signal
E1	Out	GTLP	alct_config_done
E2			
E3	In	GTLP	ccb_cmd3
E4	In	GTLP	ccb_bcntres
E5	In	GTLP	ccb_data_strobe
E6	In	GTLP	ccb_data3
E7	In	GTLP	ccb_data7
E8	In	GTLP	ccb_reserved3
E9	In	GTLP	tmb_reserved1
E10	In	GTLP	alct_external_trigger
E11	Out	GTLP	clct_status3
E12	Out	GTLP	clct_status7
E13	Out	GTLP	alct_status2
E14	Out	GTLP	alct_status6
E15	Out	GTLP	tmb_l1a_release
E16	Out	GTLP	tmb_reserved_in3
E17	In	GTLP	tmb_reserved_out2
E18			
E19	(In)[1]	GTLP	dmb_l1a_release
E20	In	GTLP	dmb_reserved_out3
E21	(In)[1]	GTLP	dmb_reserved_in2
E22			
E23	Out	LVTTL	tmb_data3
E24	Out	LVTTL	tmb_data7
E25	Out	LVTTL	tmb_data11

Notes:
1) TMB can monitor signals to/from DMB, but can not assert them.

P2B Backplane DMB Connector

Function: Sends and receives data to/from DMB.
Connector Type: PCB: AMP Z-Pack 55 (11 rows of 5 pins) female AMP 100161-1
Backplane: AMP Z-Pack 55 (11 rows of 5 pins) male AMP ?

Table 22: P2B Backplane DMB Connector

Pin	Dir	Logic	Signal
A1	Out	LVTTL	tmb_data12
A2	Out	LVTTL	alct_data1
A3	Out	LVTTL	alct_data5
A4	Out	LVTTL	alct_data9
A5	Out	LVTTL	alct_data13
A6	Out	LVTTL	tmb_data_available
A7	Out	LVTTL	tmb_active_feb1
A8	Out	LVTTL	fifo_clock
A9	Out	LVTTL	alct_last_frame
A10	In	LVTTL	res_from_dmb2
A11	Out	LVTTL	tmb_active_feb6 (v6)

Pin	Dir	Logic	Signal
B1	Out	LVTTL	tmb_data13
B2	Out	LVTTL	alct_data2
B3	Out	LVTTL	alct_data6
B4	Out	LVTTL	alct_data10
B5	Out	LVTTL	alct_data14
B6	Out	LVTTL	/tmb_write_enable_fifo
B7	Out	LVTTL	tmb_active_feb2
B8	In	LVTTL	dmb_request_lct
B9	In	LVTTL	dmb_ext_trig
B10	In	LVTTL	res_from_dmb3
B11	Out	LVTTL	res_to_dmb3

Pin	Dir	Logic	Signal
D1	Out	LVTTL	tmb_data14
D2	Out	LVTTL	alct_data3
D3	Out	LVTTL	alct_data7
D4	Out	LVTTL	alct_data11
D5	Out	LVTTL	tmb_ddu_special
D6	Out	LVTTL	tmb_active_feb_flag
D7	Out	LVTTL	tmb_active_feb3
D8	Out	LVTTL	/alct_write_enable_fifo
D9	In	LVTTL	res_from_dmb1
D10	Out	LVTTL	res_to_dmb5
D11	Out	LVTTL	res_to_dmb4

Pin	Dir	Logic	Signal
E1	Out	LVTTL	alct_data0
E2	Out	LVTTL	alct_data4
E3	Out	LVTTL	alct_data8
E4	Out	LVTTL	alct_data12
E5	Out	LVTTL	tmb_last_frame
E6	Out	LVTTL	tmb_active_feb0
E7	Out	LVTTL	tmb_active_feb4
E8	Out	LVTTL	alct_ddu_special
E9	Out	LVTTL	alct_data_available
E10	Out	LVTTL	tmb_active_feb5 (v6)
E11	In	LVTTL	dmb_fpga_pgm_done

Pin	Dir	Logic	Signal
C1	In	Pwr	Gnd
C2	In	Pwr	V_{TT} (+1.5V)
C3	In	Pwr	Gnd
C4	In	Pwr	V_{TT}
C5	In	Pwr	Gnd
C6	In	Pwr	V_{TT}
C7	In	Pwr	Gnd
C8	In	Pwr	V_{TT}
C9	In	Pwr	Gnd
C10	In	Pwr	V_{TT}
C11	In	Pwr	Gnd

P3A Backplane MPC Connector

Function: Sends and receives data to/from MPC.

Connector Type: PCB: AMP Z-Pack 55 (11 rows of 5 pins) female AMP 100161-1
Backplane: AMP Z-Pack 55 (11 rows of 5 pins) male AMP ?

Table 23: P3A Backplane MPC Connector

See ADR_MPCx_FRAMEx on p45 for signal assignments

Pin	Dir	Logic	Signal
A1	Out	GTLP	/mpc_out0
A2	Out	GTLP	/mpc_out4
A3	Out	GTLP	/mpc_out8
A4	Out	GTLP	/mpc_out12
A5	Out	GTLP	/mpc_out16
A6	Out	GTLP	/mpc_out20
A7	Out	GTLP	/mpc_out24
A8	Out	GTLP	/mpc_out28
A9	In	GTLP	lct_winner
A10			
A11			

Pin	Dir	Logic	Signal
B1	Out	GTLP	/mpc_out1
B2	Out	GTLP	/mpc_out5
B3	Out	GTLP	/mpc_out9
B4	Out	GTLP	/mpc_out13
B5	Out	GTLP	/mpc_out17
B6	Out	GTLP	/mpc_out21
B7	Out	GTLP	/mpc_out25
B8	Out	GTLP	/mpc_out29
B9			
B10			
B11			

Pin	Dir	Logic	Signal
D1	Out	GTLP	/mpc_out2
D2	Out	GTLP	/mpc_out6
D3	Out	GTLP	/mpc_out10
D4	Out	GTLP	/mpc_out14
D5	Out	GTLP	/mpc_out18
D6	Out	GTLP	/mpc_out22
D7	Out	GTLP	/mpc_out26
D8	Out	GTLP	/mpc_out30
D9			
D10			
D11			

Pin	Dir	Logic	Signal
E1	Out	GTLP	/mpc_out3
E2	Out	GTLP	/mpc_out7
E3	Out	GTLP	/mpc_out11
E4	Out	GTLP	/mpc_out15
E5	Out	GTLP	/mpc_out19
E6	Out	GTLP	/mpc_out23
E7	Out	GTLP	/mpc_out27
E8	Out	GTLP	/mpc_out31
E9			
E10			
E11			

Pins C1 through C11 are connected to Backplane Ground

P3B Backplane RPC+ALCT Connector

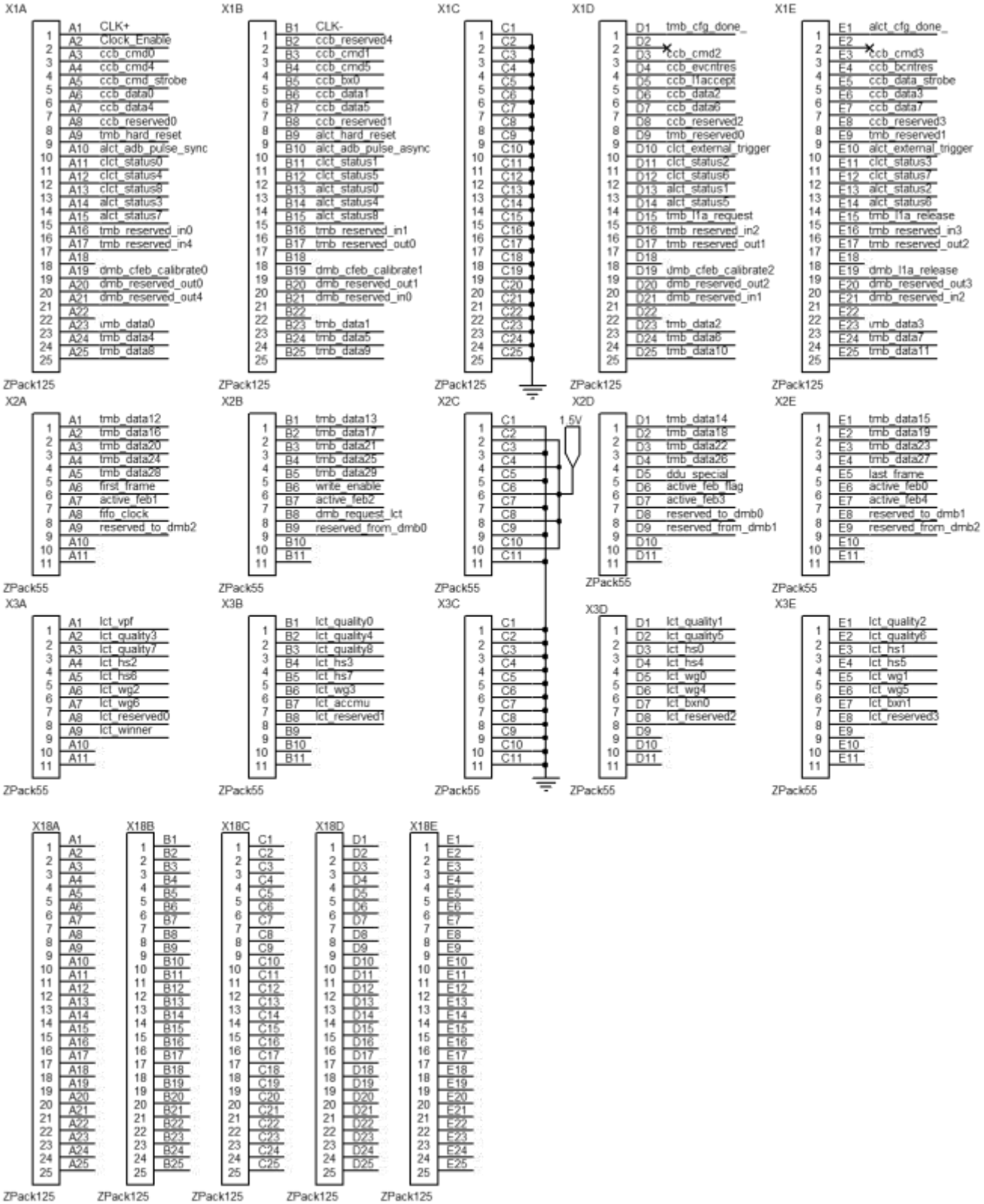
Function: Sends and receives data to/from ALCT, and receives from RPC.

Connector Type: PCB: AMP Z-Pack 125 (25 rows of 5 pins) female AMP 100145-1
Backplane: AMP Z-Pack 125 (25 rows of 5 pins) male AMP ?

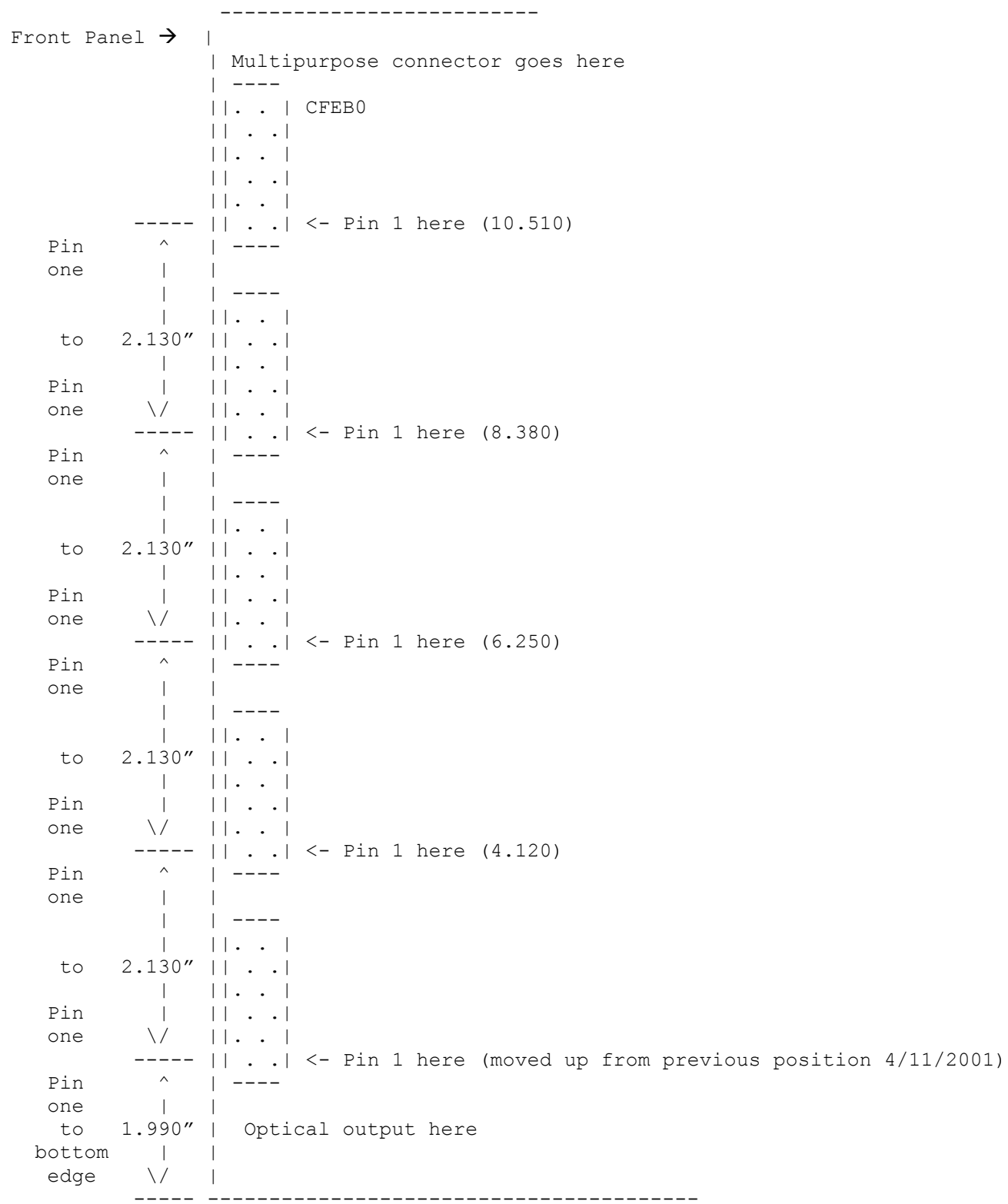
Table 24: P3B Backplane RPC+ALCT Connector

Pin	Signal		Pin	Signal		Pin	Signal		Pin	Signal		Pin	Signal
A1	rpc_rx31		B1	rpc_rx30		C1	rpc_rx29		D1	rpc_rx28		E1	rpc_rx27
A2	rpc_rx26		B2	rpc_rx25		C2	rpc_rx24		D2	rpc_rx23		E2	rpc_rx22
A3	rpc_rx21		B3	rpc_rx20		C3	rpc_rx19		D3	rpc_rx18		E3	rpc_rx17
A4	rpc_rx16		B4	rpc_rx15		C4	rpc_rx14		D4	rpc_rx13		E4	rpc_rx12
A5	rpc_rx11		B5	rpc_rx10		C5	rpc_rx9		D5	rpc_rx8		E5	rpc_rx7
A6	rpc_rx6		B6	rpc_rx5		C6	rpc_rx4		D6	rpc_rx3		E6	rpc_rx2
A7	rpc_rx1		B7	rpc_rx0		C7	rpc_clock		D7	smb_clk		E7	tck
A8	tms		B8	tdi		C8	rpc_loop		D8	posneg		E8	sync
A9	tdo		B9	smbrx		C9	rpc_in37		D9	rpc_in36		E9	rpc_in35
A10	rpc_in34		B10	rpc_in33		C10	rpc_in32		D10			E10	+3.3V
A11	+3.3V		B11	+3.3V		C11	GND		D11	GND		E11	GND
A12	alct_rx31		B12	alct_rx30		C12	alct_rx29		D12	alct_rx28		E12	alct_rx27
A13	alct_rx26		B13	alct_rx25		C13	alct_rx24		D13	alct_rx23		E13	alct_rx22
A14	alct_rx21		B14	alct_rx20		C14	alct_rx19		D14	alct_rx18		E14	alct_rx17
A15	alct_rx16		B15	alct_rx15		C15	alct_rx14		D15	alct_rx13		E15	alct_rx12
A16	alct_rx11		B16	alct_rx10		C16	alct_rx9		D16	alct_rx8		E16	alct_rx7
A17	alct_rx6		B17	alct_rx5		C17	alct_rx4		D17	alct_rx3		E17	alct_rx2
A18	alct_rx1		B18	alct_rx0		C18	smbtx		D18	hrst_rpc		E18	free_tx0
A19	alct_oe		B19	alct_clock		C19	alct_clk_en		D19	txoe		E19	alct_loop
A20	alct_tx23		B20	alct_tx22		C20	alct_tx21		D20	alct_tx20		E20	alct_tx19
A21	alct_tx18		B21	alct_tx17		C21	alct_tx16		D21	alct_tx15		E21	alct_tx14
A22	alct_tx13		B22	alct_tx12		C22	alct_tx11		D22	alct_tx10		E22	alct_tx9
A23	alct_tx8		B23	alct_tx7		C23	alct_tx6		D23	alct_tx5		E23	alct_tx4
A24	alct_tx3		B24	alct_tx2		C24	alct_tx1		D24	alct_tx0		E24	+1.8V
A25	+1.8V		B25	+1.8V		C25	GND		D25	GND		E25	GND

Backplane Pin Diagram



Front Panel Connector Locations



CCB Front Panel

CCB Input connector P10

Pin (+)	Pin (-)	Signal
1	2	external_clock40
3	4	external_clock40_enable
5	6	external_l1accept
7	8	dmb_cfeb_calibrate[0]
9	10	dmb_cfeb_calibrate[1]
11	12	dmb_cfeb_calibrate[2]
13	14	alct_adb_pulse_sync
15	16	alct_adb_pulse_async
17	18	clct_external_trigger
19	20	alct_external_trigger
21	22	tmb_l1a_request
23	24	ccb_fp_reserved_in[0]
25	26	ccb_fp_reserved_in[1]
27	28	
29	30	
31	32	
33	34	

CCB Output connector P11

Pin (+)	Pin (-)	Signal	Test Point	TMB Assignment	Description
1	2	clct_status[0]	391-1	pretrig	Sequencer pre-triggered
3	4	clct_status[1]	391-2	seq_busy	Sequencer busy
5	6	clct_status[2]	391-3	invpat	Invalid pattern after drift delay
7	8	clct_status[3]	391-4	daqmb	Dump to DMB in progress
9	10	clct_status[4]	391-5	l1a_window	L1A window
11	12	clct_status[5]	391-6	l1a	L1A (should be in L1A window)
13	14	clct_status[6]	391-7	tmb	CLCT sent for TMB match
15	16	clct_status[7]	391-8	tmb_flush	TMB found no match or rejected trigger
17	18	clct_status[8]	391-9	nol1a_flush	No L1A, Sequencer flushing event
19	20	ccb_clock40			
21	22	ccb_bx0			
23	24	ccb_l1accept			
25	26	ccb_cmdstr			
27	28	ccb_fp_reserved_out[0]			
29	30				
31	32				
33	34				

CCB Output connector P12

Pin (+)	Pin (-)	Signal	Test Point	TMB Assignment	Description
1	2	alct_status[0]	392-1	alct_active_feb	ALCT fast active AFEB
3	4	alct_status[1]	392-2	first_valid	First muon valid pattern flag
5	6	alct_status[2]	392-3	second_valid	Second muon valid pattern flag
7	8	alct_status[3]	392-4	first_amu	First accelerator muon flag
9	10	alct_status[4]	392-5	second_amu	Second accelerator muon flag
11	12	alct_status[5]	392-6	/wr_fifo(alct)	ALCT /write enable raw-hit FIFO
13	14	alct_status[6]	392-7	alct_vpf	ALCT 1st Valid Pattern (TMB pipe)
15	16	alct_status[7]	392-8	clct_vpf	CLCT 1st Valid Pattern (TMB pipe)
17	18	alct_status[8]	392-9	scint_veto	Scintillator Veto (clears via VME)
19	20				
21	22				
23	24				
25	26				
27	28				
29	30				
31	32				
33	34				

Documentation Revision History

Version	Date	Action
1.00	07/12/2003	Initial, copied from TMB2001
1.01	10/29/2003	Replaced PHOS4 registers with DDD, update shunts table, rat signals updated
1.02	03/15/2004	Update 3D3444 registers, add RAT/RPC registers
1.03	03/16/2004	Move RAT register addy to match bdtest.v
1.04	04/15/2004	Copy from TMB2003
2.01	05/19/2004	Add RPC Readout
2.01	06/09/2004	New registers B6-CC, 4 new header words, new raw hits format, started logic docs
2.02	06/10/2004	Add mpc_tx_delay, modify header 21,22
2.03	06/18/2004	Add register CE for new scope trigger source, add mpc data format
2.04	07/23/2004	Add nph_pattern to header word 21
2.05	08/03/2004	Add counter registers
2.06	10/01/2004	Add scp bit to header
2.07	10/04/2004	Add mpc_bx0 to reg ADR_TMB_TRIG
2.08	10/07/2004	Typos fixed
2.09	12/15/2004	Typos fixed
3.00	06/02/2005	Ported from TMB2004 v2.09 dated 12/15/04
3.00	06/09/2005	Add rpc_hard_reset to boot reg, update ADR_LOOPBK for RAT signals
3.01	06/28/2005	alct_rxoe, alct_txoe in 0E are now readonly to foil errant software
3.02	08/01/2005	Change default alct_status_en, tmb_status_en=0 for multi-crate use
3.03	08/08/2005	Change c_status_oe to ccb_status_oe
3.04	02/27/2006	Add configuration section
3.05	05/24/2006	Add jtag state machine registers, add alct scope channels
3.06	07/28/2006	Add vme state machine registers, update raw hits header, add prom info
3.07	08/08/2006	Add layer-or trigger mode
3.08	09/05/2006	Add rat delay registers
3.09	09/15/2006	Add CLCT processing steps
3.10	10/16/2006	Add ISE Version register, triad persistence default now 6 for 6 clocks
3.11	04/09/2007	Reduce RPCs from 4 to 2, add mpc_oe, affects 86,B6,BA,BE,C4,CA,CC,header22
3.12	04/11/2007	Remove duplicate header22 description, update header tables
3.13	04/27/2007	Add ignore ttc_start/stop to Adr 2C, expand fmm_state in Adr 9C
3.14	05/16/2007	New pattern finder registers
4.01	05/24/0207	New clct pattern finder algorithm
4.02	06/21/2007	New pattern ID numbers, new layer triggermods to Adrs 10,D4, and A0,A6,A8,AA,CA,CC,header16
4.03	07/05/2007	Key layer shifted from ly3 to ly2, adjcfeb_dist expanded to 6 bits, activefeb bugfix
4.04	07/10/2007	Firmware update, no change to doc
4.05	07/24/2007	Update Adr 78,88,8C descriptions to match current CLCT0/1 4-bit pattern IDs
4.06	07/26/2007	Fix text for adr 68, 6C, A6
4.07	08/06/2007	Increase injector read-write adr to 10 bins in Adr 44, Adr BE
4.08	08/30/2007	New header/trailer format
4.09	09/04/2007	Mods to header format, expand VME counters to 30 bits
4.10	09/10/2007	New fields in header 24, new VME counters
4.11	09/14/2007	Header bug fixes, new no-alct counter
4.12	09//21/2007	Push hsds bit into pat[3] in documentation
4.13	12/17/2007	Switch to big-buffer logic
4.14	01/28/2008	Replace entire L1A logic, replace lct_ quality

4.15	02/05/2008	Add raw hits RAM parity errors to header27
4.16	04/29/2008	Major update for new flow-through triggering
4.17	05/01/2008	All scope channels reassigned, header frame bug fixes
4.18	05/12/2008	Add clct + alct bx0 heartbeat to mpc_tx[11], mpc_tx[27]
4.19	05/23/2008	Adr B0[15:14] Add clock_lock_lost and sync_er
4.20	06/03/2008	Add counter clears on ttc_resync, add aclt debug to scope, new active_feb_flag
4.21	07/09/2008	Replace jtag state machine, rename thresh signals, add pid_thresh_postdrift
4.22	07/15/2008	non-trigger readout mods
4.23	08/12/2008	Add bx0_match, add me1a/b separation ram mux, add alct data tx delay
4.24	08/28/2008	Programmable stagger removed, csc firmware types created, new counters
4.25	09/12/2008	Add tmb logic signals to header40,41
4.26	09/16/2008	Update tables for header words 36(rpc),40,41. Updated sample raw hits dump
4.27	10/01/2008	Mod header08,36,40,41, replaced scope module and vme register, jtag sel
4.28	11/18/2008	Add reg FE, L1A and parity logic mods, add sync err and parity err counters
4.29	11/19/2008	Remove stagger_csc from Adr F4[1], staggering info is in Adr CC
4.30	12/10/2008	Add l1a_lookback mode, l1a_lookback reg + sequencer debug reg; add parity ram
4.31	02/05/2009	Add alct-loopback logic in alct.v module
4.32	03/16/2009	Add ecc to alct data, add ecc counters, add cfeb counters
4.33	04/15/2009	Updates for alct 80MHz output stages
4.34	05/15/2009	Miniscope added, RPC readout mods, bx0 test mode
4.35	05/29/2009	Add alct muonic timing
4.36	06/22/2009	More alct muonic mods
4.37	06/29/2009	Replace alct_txd, alct_rxd DDD delays with digital phase shifters
4.38	07/13/2009	Add cfeb muonic timing, 5 digital phase shifters, 2 vme delay registers
4.39	08/14/2009	CFEB muonic version, removed cfeb and alct_rx posnegs, still has alct_tx posneg
4.40	08/25/2009	ALCT+CFEB muonic with posneg sync stages, ISE 8.2sp3
4.41	09/02/2009	Correct adr 16
4.42	09/08/2009	Add digital phase shifter autostart
4.43	09/21/2009	Add sync register 0x120, limit bxn offsets to be < lhc_cycle
4.44	10/15/2009	Add alct0==alct1 error counter, change D0[5] default to 1, enables alct err counter
4.45	11/23/2009	Text corrections
4.46	12/09/2009	Add alct and cfeb muonic figures
4.47	12/16/2009	Add cfeb bad bit detection and registers
4.48	01/14/2010	Add cfeb bad bit list to header30, mod bad bit ctrl reg
4.49	02/10/2010	Add event clear for aff, clct, mpc VME diagnostic registers
4.50	03/12/2010	Add blocked bits readout, clct-only mode bug fix
4.51	04/09/1200	Fix header30 typo, add firmware log for 3/19/2010 version
4.52	05/14/2010	Mod Adr 0x2A and Adr 0x120, add firmware log for 5/14/10 version
4.53	07/02/2010	Mod Adr 0x32, 0x42, 0x9E, 0x100, and hdr38[14]
4.54	07/08/2010	Move injector ram msbs to lookback reg, revert injector wen,ren
4.55	07/15/2010	Add miniscope section to header chapter
4.56	09/03/2010	Convert to MS Word 2010
4.57	08/17/2012	Add startup delay for ALCT Spartan-6 mezzanine
4.58	09/16/2012	Add Virtex-6 GTX optical receivers
4.59	09/23/2012	Add Virtex-6 Sysmon
4.60	10/15/2012	Change pid_thresh descriptions in adr 0xF4
4.61	03/07/2013	Convert doc to docx, mod header + add registers for 7 DCFEB Virtex-6

4.62	2013/10/24	Change order of keys in LCT quality module, documentation updated accordingly
5.01	2013/12/19	Created assignments for Mez-2013 TPs & SMT LEDs: qpll & mmcm lock monitoring. Also minor corrections for VME GTX register table (Adr 14C-158)
5.02	2013/01/06	Changes to Reset for CFEB badbits, fiber link monitor logic & GTX VME registers 14A-158

Firmware Change Log

Version 01/06/2014

Improved control & monitoring for Virtex-6 SNAP12 GTX fiber optic receivers

First applied in firmware version 12302013

1) TTC_Resync now included in reset logic for CFEB badbits
   cfeb_badbits_reset[4:0] = (cfeb_badbits_ctrl_wr[4:0] | {5{ttc_resync}});
   cfeb_badbits_reset[6:5] = (cfeb_v6_badbits_ctrl_wr[1:0] | {2{ttc_resync}});

2) GTX VME registers 14A-158 changed significantly to reflect improved fiber link monitor & control logic
   Removed functions that were not useful (previously accessed in bits 10, 7:4, 2)
   Reassigned some other functions to different bits (previous bit 9 → bit 7, bit 3 → bit 2)
   Added new functions to bits 6:3 (gtx_link_bad, gtx_link_had_err, gtx_link_good, gtx_rx_sync_done)
   Changed the readout response for bits 2:0 in registers 14C – 158 when master GTX control is asserted
      When the master GTX control bits are set in register 14A (bits 2:0) they are now accurately reflected in the respective readout bits for the individual GTX links (bitwise OR in each GTX)
   The error count now counts link sync failures when the PRBS function is not enabled
      An occassional loss of sync for a link is expected due to SEUs, but large numbers of them or repeated occurrences may indicate a problem in the system
   Increased the error count size to 8 bits and assigned this to bits 15:8
      For registers 14C – 158 the maximum error count is hex E0 (n.b. a count larger than 1 is severe)
      Register 14A is the sum of the other 7 counters, with maximum allowed value of hex FE

Version 12/19/2013

Adds diagnostic features for Mez-2013

Applied in firmware versions 12162013 – 12172013

1) Assign Mez-2013 SMT LEDs to indicate qpll & mmcm lock conditions and status changes
   See Table 12A

2) Assign additional diagnostic signals to the Mez-2013 test points and SMT LEDs
   See Tables 12A and 12B

3) Minor corrections to the table for VME GTX registers (Adr 14C-158)
   Removed references to “all”

Version 03/08/2013

Adds support for 7 DCFEBs with Virtex-6

1) Modified ADR_SEQCLCTM=0xB0 Sequencer CLCT msbs Register, Readonly
Added seq_clctmsb_rd[9:8] = clctf_vme[6:5], bits were formerly unused
Now seq_clctmsb_rd[9:3] = clctf_vme[6:0] Active cfeb list at TMB match

2) ADR_PARITY=0xFA Parity errors
Changed sub-address assignments to accommodate DCFEB[6:5]

3) Modified ADR_CNT_CTRL=0xD0 Status Counter Control
Inserted 2 new counters after cnt[18], shifts subsequent counter addresses by 2
cnt_en[19] <= cfeb_hit_at_pretrig[5] CLCT pretrigger is on CFEB5
cnt_en[20] <= cfeb_hit_at_pretrig[6] CLCT pretrigger is on CFEB6

4) Modified ADR_DELAY1_INT=0x11E DDR Interstage delays for DCFEB[6:5]
Added:
cfeb5_rxd_int_delay[3:0] = delay1_int_wr[7:4];
cfeb6_rxd_int_delay[3:0] = delay1_int_wr[11:8];

5) Added ADR_V6_CFEB_BADBITS_CTRL=0x15C CFEB Bad Bits Control/Status
For DCFEB[6:5] badbits, extends 5 bit fields from ADR_CFEB_BADBITS_CTRL

6) Added 6 new bad bit VME registers
ADR_V6_CFEB5_BADBITS_LY01 = 0x15E CFEB5 Bad Bit Array
ADR_V6_CFEB5_BADBITS_LY23 = 0x160 CFEB5 Bad Bit Array
ADR_V6_CFEB5_BADBITS_LY45 = 0x162 CFEB5 Bad Bit Array

ADR_V6_CFEB6_BADBITS_LY01 = 0x164 CFEB6 Bad Bit Array
ADR_V6_CFEB6_BADBITS_LY23 = 0x166 CFEB6 Bad Bit Array
ADR_V6_CFEB6_BADBITS_LY45 = 0x168 CFEB6 Bad Bit Array

7) Added 2 new Digital Phase Shifter VME registers
ADR_V6_PHASER7=0x16A DCM Phase Shifter Register: CFEB5 rxd
ADR_V6_PHASER8=0x16C DCM Phase Shifter Register: CFEB6 rxd

8) Added 6 New hot channel mask registers
ADR_V6_HCM501 = 0x16E CFEB5 Ly0,Ly1 Hot Channel Mask
ADR_V6_HCM523 = 0x170 CFEB5 Ly2,Ly3 Hot Channel Mask
ADR_V6_HCM545 = 0x172 CFEB5 Ly4,Ly5 Hot Channel Mask

ADR_V6_HCM601 = 0x174 CFEB6 Ly0,Ly1 Hot Channel Mask
ADR_V6_HCM623 = 0x176 CFEB6 Ly2,Ly3 Hot Channel Mask
ADR_V6_HCM645 = 0x178 CFEB6 Ly4,Ly5 Hot Channel Mask

9) Modified ADR_MOD_CFG 0x28 TMB Module Configuration Register
Replaced led_flash_rate from mod_cfg_wr[11:10] with cfeb_exists[6:5]
Now: mod_cfg_rd[11:5] = cfeb_exists[6:0] = CFEBs instantiated in this firmware

10) Added new VME register ADR_V6_EXTEND = 0x17A: DCFEB 7-bit extensions
Extends 5-bit cfeb fields in Adr 0x42 and 0x68 to 7 bits

```
mask_all[6:5]          =[1:0] Extend 0x42[4:0]   = mask_all[4:0]1=Enable, 0=Turn off all CFEB inputs
```

```
inj_febsel[6:5]        =[3:2] Extend 0x42[9:5]   = inj_febsel[4:0]1=Select CFEBn for RAM read/write
injector_mask_cfeb[6:5]=[5:4] Extend 0x42[14:10] = injector_mask_cfeb[4:0] 1=Enable CFEB(n) for injector
cfeb_en_vme[6:5]       =[7:6] Extend 0x68[14:10] = cfeb_en_vme[4:0] 1=Enable CFEBs for triggering
```

11) Modified ADR_LOOPBK 0x0E Loop-Back Control Register 2 remove 2 spare DMB signals

Reassigned dmb_tx_reserved to remove the 2 new active feb bits from the unused list
Updated bits [10:5] to match current firmware

12) TMB-to-DMB Active CFEB List backplane signals extended 5 bits to 7

Assigned 2 new bits to the next available spare DMB signals.
dmb_tx[45:44] <= active_feb_list[6:5]

TMB schematic signals:
Signal res_to_dmb1 is now active_feb_list[5] on backplane pin E10
Signal res_to_dmb2 is now active_feb_list[6] on backplane pin A11

13) Modified header40_[11:0] for 7-bit extensions to various 5-bit header fields

Header40_[11:0] used to contain the peak RAM fence counter for raw hits storage debugging. The peak count has been removed, but can still be read out with VME, and its dynamic status is indicated by header40_[10].

The flag bit in header40_[11] indicates when bits [10:0] are valid for7-dcfeb firmware versions.

header40_[1:0] = active_feb_mux[6:5]; Extend Hdr23[4:0] Active CFEB list sent to DMB
header40_[3:2] = r_cfebs_read[6:5]; Extend Hdr23[9:5] CFEBs read out for this event
header40_[5:4] = perr_cfeb_ff[6:5]; Extend Hdr27[12:8] CFEB RAM parity error, latched
header40_[7:6] = cfeb_badbits_found[6:5]; Extend Hdr30[11:7] CFEB[n] has at least 1 bad bit
header40_[9:8] = cfeb_en[6:5]; Extend Hdr35[14:10] CFEBs enabled for triggering
header40_[10] = buf_fence_cnt_is_peak; Current fence is peak number of fences in RAM
header40_[11] = (MXCFEB==7); TMB has 7 DCFEBs so hdr40_[10:0] are active

14) Modified raw hits readout

Readout can now include raw hits from 0 to 7 CFEBs, as indicated by cfebs_read[6:0]. The 7-bit CFFEB list cfebs_read[6:0] is constructed from cfebs_read[4:0] in header23_[9:5] and cfebs_read[6:5] in header40_[3:2].

In 5-CFEB firmware versions, header40[11] is zero (unless the is a fault in the readout logic). A zero in header40_[11] indicates a 5-CFEB TMB, and the bits in header40_[10:0] should not be used to extend 5-bit fields (cfebs_read for instance).

15) Blocked bits readout

Readout can now include blocked hits from 0 to 7 CFEBs, as indicated by cfebs_read[6:0].

Version 09/23/2012

1) Add VME register 0x15A: ADR_VIRTEX6_SYSMON

Version 09/16/2012

Adds support for Virtex-6 SNAP12 GTX fiber optic receivers

2) Add VME register 0x148: ADR_VIRTEX6_SNAP12_QPLL
   Virtex-6 mezzanine QPLL reset and status
   Virtex-6 mezzanine SNAP12 receiver serial interface

3) Add VME register 0x14A: ADR_VIRTEX6_GTX_RX_ALL
   GTX control and status common to all 7 receivers

3) Add 7 VME registers 0x14C-0x158: ADR_VIRTEX6_GTX_RX0 - ADR_VIRTEX6_GTX_RX6
   GTX control and status for individual receivers

4) Add event counters 79-85:
   GTX receiver counters for fibers 0-6
   Clears on gtx_rx_reset_err_cnt

5) Add mezzanine test points for GTX receiver [0]

```
`ifdef VIRTEX6
      assign meztp20 = gtx_rx_start[0];
      assign meztp21 = gtx_rx_valid[0];
      assign meztp22 = gtx_rx_match[0];
      assign meztp23 = gtx_rx_fc[0];
      assign meztp24 = alct_wait_cfg;
      assign meztp25 = lock_tmb_clock0;
      assign meztp26 = 0;
      assign meztp27 = sump;
`else
      assign meztp20 = alct_startup_msec;
      assign meztp21 = alct_wait_dll;
      assign meztp22 = alct_startup_done;
      assign meztp23 = alct_wait_vme;
      assign meztp24 = alct_wait_cfg;
      assign meztp25 = lock_tmb_clock0;
      assign meztp26 = 0;
      assign meztp27 = sump;
`endif
```

Version 08/17/2012

1) Delay JTAG PROM data stream to ALCT by 116msec to allow Spartan-6 mezzanine to finish configuration.

   Spartan-6 takes 212msec to configure and TMB takes 100msec.
   TMB sends JTAG data to ALCT 100+116=216msec, after a
   simultaneous hard reset.
   The 4msec pad allows ALCTs DLL/PLL time to lock.

2) Add VME register Adr 0x144 ALCT Spartan-6 startup delay

3) Add VME register Adr 0x146 ALCT Spartan-6 startup state machine status

4) Change Adr 0xD4[2]=jsm_sel, was Write-only, is now Read/Write

Version 07/07/2010

1) Injector RAM mods:
   Move injector RAM data msbs [17:16] from 0x44 to 0x100
   Revert 0x44 wen and ren to independent RAM enables

Version 07/04/2010

1) Changed default Adr 0x100[15] l1a_win_pri_en = 1
   Enables window prioritizing mode to limit TMB to 1 readout per L1A

Version 07/01/2010

1) Bug fix in alct*clct matching for a rare case when 2 CLCTs are exactly clct_window bx apart in time.
Caused 1st CLCT to be replaced by 2nd CLCT, sending 2 identical events to L1A pipeline.

2) New algorithm for L1A matching prevents multiple events from reading out for 1 L1A.
Original algorithm allowed multiple events to read out if they were within the L1A window.
A system downstream of TMB apparently fails to tolerate multiple readouts per L1A.
Adr 0x100[15] l1a_win_pri_en = 0 enables original multiple event readouts per L1A mode
l1a_win_pri_en = 1 enables a prioritizing mode that limits TMB to 1 readout per L1A
Current default l1a_win_pri_en = 0 to allow checking alct*clct matching logic.
Next firmware release will set the default to 1.

3) New counter 58 is inserted after counter 57, all subsequent counters shift up 1channel number.
Counter 58 counts events lost from readout queue due to L1A window prioritizing that limits TMB to 1 event readout per L1A.

4) A buffer stalled-at-least-once bit has been added to Adr 0x9E[7] and Header38[14].
Indicates there was a least one buffer stall since the last resync.

5) Adr 0x44 rebuilt to expand CFEB pattern injectors to also assert L1A and ALCTs at an arbitrary time.
The 3 individual injector RAM select bits ren and wen have been replaced by 2-bit RAM addresses.
Injector RAM data width expanded from 16 to 18 bits to provide storage for 2 ALCTs and L1A.

6) Adr 0x32[3] alct_inj_ram_en = 1 enables ALCT pattern injector RAM
Adr 0x32[4] l1a_inj_ram_en = 1 enables L1A pattern injector RAM

7) Miniscope is now turned on by default: Adr 0x10C[0] mini_read_enable = 1.
Miniscope word count automatically inserted in readout stream by default Adr 0x10C[2]=1
New channel assignments, see Adr 0x10C section.

Version 05/14/2010

1) Add bx0 emulator enable to Adr 0x2A[15]. Power-up default is 0.
   Generates bx0 that is ORed with ttc_bx0. For use in systems lacking a CCB.

2) Add clock_lock_lost_err_en to Adr 0x120[5] (shifts other bits left by 1).
   Add clock_lock_lost_err_ff to Adr 0x120[14].
   Add force_sync_err to Adr 0x120[15].

3) Redesign vme.v to use initial blocks to specify power-up state for VME registers instead of load pulse
   Add explicit integer widths to VME address decoder case statement.
   Replace constants passed as signals from top level module with defparam mechanism.

4) Remove clock_lock_lost from OR with bx0_sync_err in sequencer.v
   Add clock_lock_lost term to sync_err_ff in sync_err_ctrl.v
   Remove bx0_sync_err from sync_err_ff, add it as an independent OR with sync_err signal.
   Fixes sync_err fails-to-clear bug:
      Sync_err_ff clears on ttc_resync but bx0_sync_err takes n bx to clear.
      TMB was synchronizing correctly on ttc_resync, but incorrectly latched sync_err=1.

5) Modify bx0 sync error counter[61] to count only when TMB is in trigger-run state.
   Was counting sync errors before ttc_resync arrived, now only counts errors that occur after a resync.
   A non-zero value in this counter indicates incorrect bxns in TMB readouts and LCTs.

6) Modify FMM state machine fmm_trig_stop signal to power up as a 1.
   Was powering up as 0, then set to 1 after the 1st clock cycle.
   Entered trigger-run state for 1bx but was ignored since pre-trigger logic is held off 5bx after power up

7) Fix tmb.v kill_clct logic for type C|D for 1 clct + 2 alct case where clct is on ME1A

Version 03/19/2010

1) Mod cfeb.v module busy hs delimiters for me1a me1b cscs to separate cfeb4 from cfebs0-3.
   Prevents pattern finder from discarding 2nd CLCT at the me1a|me1b boundary.

Version 03/07/2010

1) Added requested mod that blocks bad cfeb distrips from both trigger path and data path.

2) Add bcb_read_enable to Adr72[15] p42 to include blocked CFEB DiStrip list in the
   DMB readout stream:
   Blocked bits include:
   CFEB DiStrip bits turned off in the Hot Channel Mask,
   CFEB DiStrip bits turned off by automatic bad-bits detection
   Entire CFEBs turned off via mask_all

   Set Adr72[[15]=1 to enable blocked bits readout.
   Power up default is 0, which is backwards compatible with older versions of the readout stream.

3) Add bcb_read_enable to Header29[13]

4) Document CFEB blocked DiStrip readout format, p95

5) Bug fix for CLCT-only trigger mode
   The 2nd CLCT in previous firmware versions had bxn=0 in the LCT sent to MPC when using the clct-only mode, and there was also no ALCT coincidence.

Version 02/10/2010

1) Add event_clear_vme to AdrAC[15] to clear aff, clct, and mpc VME read-back registers

2) Add active_feb list reversal for TypeB CSCs

Version 01/14/2010 CFEB bad di-strip bit detection

1) Header30[11:7] = cfeb_badbits_found[4:0] Bad distrip bits detected in cfebn[n]
   Header30[12] = 0

2) New VME regisers Adr 0x122 to 0x142

Adr 122 ADR_CFEB_BADBITS_CTRL, CFEB Bad Bits Control/Status

```
[04:00]   RW  cfeb_badbits_reset[4:0]  Reset bad cfeb bits FFs for cfeb[n]
[09:05]   RW  cfeb_badbits_block[4:0]  Block bad cfeb bits in cfeb[n]
[14:10]   R   cfeb_badbits_found[4:0]  CFEB[n] has at least 1 bad bit
[15]      R   cfeb_badbits_blocked     At least one CFEB has a bad bit that was blocked
```

Adr 124 ADR_CFEB_BADBITS_TIMER CFEB Bad Bits Check Interval
Sets number of bx a bit must be continuously high before being marked as bad.

Adr 126	ADR_CFEB0_BADBITS_LY01	CFEB0 Ly0,Ly1 Bad Bits List
Adr 128	ADR_CFEB0_BADBITS_LY23	CFEB0 Ly2,Ly3 Bad Bits List
Adr 12A	ADR_CFEB0_BADBITS_LY45	CFEB0 Ly4,Ly5 Bad Bits List
Adr 12C	ADR_CFEB1_BADBITS_LY01	CFEB1 Ly0,Ly1 Bad Bits List
Adr 12E	ADR_CFEB1_BADBITS_LY23	CFEB1 Ly2,Ly3 Bad Bits List
Adr 130	ADR_CFEB1_BADBITS_LY45	CFEB1 Ly4,Ly5 Bad Bits List
Adr 132	ADR_CFEB2_BADBITS_LY01	CFEB2 Ly0,Ly1 Bad Bits List
Adr 134	ADR_CFEB2_BADBITS_LY23	CFEB2 Ly2,Ly3 Bad Bits List
Adr 136	ADR_CFEB2_BADBITS_LY45	CFEB2 Ly4,Ly5 Bad Bits List
Adr 138	ADR_CFEB3_BADBITS_LY01	CFEB3 Ly0,Ly1 Bad Bits List
Adr 13A	ADR_CFEB3_BADBITS_LY23	CFEB3 Ly2,Ly3 Bad Bits List
Adr 13C	ADR_CFEB3_BADBITS_LY45	CFEB3 Ly4,Ly5 Bad Bits List
Adr 13E	ADR_CFEB4_BADBITS_LY01	CFEB4 Ly0,Ly1 Bad Bits List
Adr 140	ADR_CFEB4_BADBITS_LY23	CFEB4 Ly2,Ly3 Bad Bits List
Adr 142	ADR_CFEB4_BADBITS_LY45	CFEB4 Ly4,Ly5 Bad Bits List

Usage Notes:

1) Dead channel detection:
   Detects CFEB channels that never fire
      Set Adr122[09:5]=0x00 to turn off badbit blocking
      Set Adr124[15:0]=0x0001 to set high-time threshold to 1 bx
      Read dead channel list from Adrs126-142

2) Noisy channel detection:
   Detects CFEB channels that have after-pulsing or frequent firing,
   for instance, 3 consecutive triad starts)
      Set Adr122[09:5]=0x00 to turn off badbit blocking
      Set Adr124[15:0]=0x0007 to set high-time threshold to 7 bx
      Read noisy channel list from Adrs126-142

3) Bad bit detection or blocking
   Detects CFEB channels that are always high or high for an unreasonable length of time
      Set Adr122[09:5]=0x1F to turn on badbit blocking (or set 0x00 for just monitoring)
      Set Adr124[15:0]=0x0DEC to set high-time threshold to 3564 bx or something similar
      Read bad channel list from Adrs126-142

Version 10/15/09 ALCT duplicate alct detection + Header r-type

1) Header05[10:9] r_type always == 1 on previous versions.
   It should equal fifo_mode unless the event buffer is full.

2) Added ALCT structure error counter cnt[75] to count events where alct0==alct1.

3) AdrD0[5] Changed default to cnt_alct_debug=1 to enable ALCT data structure error counters.
   N.B. The ALCT structure error counters are only 8 bits, and could reach full scale quickly.
   So, AdrD0[02] cnt_stop_on_ovf should be 0, when setting cnt_alct_debug=1, otherwise
   all event counters will stop counting if there are excessive ALCT errors.

Version 09/21/09 Synchronization Error Control Register + bxn offset limit

Add limits for bxn_offset_pretrig and bxn_offset_l1a
bxn_offset > lhc_cycle is converted to lhc_cycle-1

Add sync_err_ctrl register Adr 0x120

Version 09/08/09 Digital Phase Shifter Autostart

Add vsm_phaser_auto to AdrDA[11].
Default = 1, starts digital phase shifters after VME user PROM is read

Version 08/25/09 PosNeg sync FFs for ALCT and CFEBs

Same as 8/14/09 version but has ALCT and CFEB posneg sync stages enabled.
Switched to ISE 8.2sp3 because ISE 10.1sp3 could not complete PAR.

Documentation updates to conform firmware to c++ demo code:

Phaser register signal names now absorb hcycle and qcycle bits into 1 8-bit phase delay:

Adr10E:	alct_rxd_delay[7:0]	Delays latching data received from ALCT in 0.1ns steps
Adr110:	alct_txd_delay[7:0]	Delays data transmitted to ALCT in 0.1ns steps
Adr112-11A:	cfeb[n]_rxd_delay[7:0]	Delays latching data received from CFEB[n] in 0.1ns steps

Modify interstage delay signals to make it clear they are integer bx delays:

Adr38:	alct_txd_int_delay[3:0]	Delay data transmitted to ALCT by integer bx
Adr11C-11E:	cfeb[n]_rxd_int_delay[3:0]	Delay data received from CFEB[n] by integer bx

Version 08/14/09 Digital Phase shifters for CFEBs

Has both ALCT and CFEB muonic timing.
Disabled cfeb posnegs and alct_rxd_posneg else compile fails. Alct_txd_posneg is OK.

Notes on ALCT and CFEB timing adjustments:

ALCT:

1) Select a Time of Flight delay:
   - Using DDD 2ns steps, ranging from 0 to 12, spanning 0 to 24ns
   - Based on distance from IP to "some point" on the CSC
   - Also compensate for tmb-to-alct cable propagation delay differences between CSCs
   - Write alct_tof_delay to DDD chip in Adr16[3:0]

2) Tune alct_rxd_delay to the good-data window center
   - Using Digital Phase Shifter 0.1ns steps, ranging from 0 to 255, spanning 0 to 25ns
   - Put ALCT into loopback mode to send a test pattern to TMB
   - Scan alct_rxd_delay 0-255 using Phaser0 Adr10E[15:8]
   - Scan alct_rxd_posneg 0-1 using Phaser0 Adr10E[15:8] (disabled in 8/14/09 firmware)

3) Tune alct_txd_delay to the good-data window center
   - Using Digital Phase Shifter 0.1ns steps, ranging from 0 to 255, spanning 0 to 25ns
   - Put ALCT into loopback mode to send a test pattern to TMB
   - Scan alct_txd_delay 0-255 using Phaser1 Adr110[15:8]
   - Scan alct_rxd_posneg 0-1 using Phaser1 Adr110[15:8] (not disabled in 8/14/09 firmware)

CFEBs:

1) Select a Time of Flight delay:
   - Using DDD 2ns steps, ranging from 0 to 12, spanning 0 to 24ns
   - Based on distance from IP to "some point" on the CSC
   - Also compensate for tmb-to-cfeb cable propagation delay differences between CSCs
   - Write cfeb_tof_delay to DDD chip in Adr18[11:8]

2) Tune cfeb[n] clock delays for simultaneous arrival at all 5 cfebs
   - Using DDD 2ns steps, ranging from 0 to 12, spanning 0 to 24ns
   - Delays might be set according to known cable propagation delays
   - Delays might be determined empirically by setting high comparator thresholds to make the analog signal time-over-threshold less than 25ns, then scanning DDD delay vs 6-hit efficiency.
   - Write cfeb[n] clock delays to DDD channels in Adr18[15:12] and Adr1A[15:0]

3) Tune cfeb_rxd_delay for cfeb[n] to the good-data window center
   - Using Digital Phase Shifter 0.1ns steps, ranging from 0 to 255, spanning 0 to 25ns
   - Generate CFEB test pulses or use muon tracks
   - Scan cfeb_rxd_delay 0-255 using Phaser2-6 Adr112-Adr11A bits[15:8]
   - Scan cfeb_rxd_posneg 0-1 using Phaser2-6 Adr112-Adr11A

4) Tune cfeb inter-stage integer delay for cfeb[n]
   - Set a delay 0-15bx so that triad bits from all 5 CFEBs arrive at TMB on the same bxn
   - Might be done by pulsing all 5 CFEBs simultaneously, then checking the CFEB raw hits readout to see that triad start bits all appear in the same bxn.

Set cfeb[n] inter-stage delays in Adr11C-Adr11E

Version 07/13/09 Digital Phase shifters for CFEBs

Added 5 digital phase shifters for cfeb rxd delays: Adr112-Adr11A

Add 2 VME registers: Adr11C-Adr11E for CFEB interstage delays

Version 06/29/09 Digital Phase shifters for ALCT

Two digital phase shifters replace DDD delays for alct_txd_delay and alct_rxd_delay

Add VME registers Adr10E and Adr110 for digital phase shifters

Adr14[13,11,10] reverted to old format, removed phase shifter DCM locks

Adr30[15:13] removed posnegs, they now reside in Adr10E and Adr110, cfeb posneg is gone for now

Version 06/22/09 Muonic Timing for ALCT

Added muonic timing to float ALCT board in clock-space independently
of good-data rxd|txd windows.

Changes to DLL lock register:

Adr14[10] lock_alct_rxd [these get undone in 6/29/09 version]
Adr14[11] lock alct_txd
Adr 14[13] lock_cfeb_rxd

Changes to DDD delay and posneg registers:

Adr16[3:0] alct_tof_delay Shift entire ALCT in clock-space to compensate for muon time of flight
Adr16[7:4] alct_txd_delay Latches TMB-to-ALCT data in middle of transmit data window
Adr16[11:8] dmb_tx_delay Change default to 6

Adr18[3:0] alct_rxd_delay Latches ALCT-to-TMB data in middle of receive data window
Adr18[7:4] cfeb_rxd_delay Latches CFEB-to-TMB data in middle of receive data window
Adr18[11:8] cfeb_tof_delay Shift all CFEBs in clock-space to compensate for muon time of flight.

Adr30[8] alct_clock_en_use_ccb moved from [11]
Adr30[9] alct_clock_en_use_vme moved from [12]
Adr30[10] alct_muonic 1=ALCT muonic version instantiated, readonly
Adr30[11] cfeb_muonic 1=CFEB muonic version instantiated, readonly
Adr30[12] unassigned

[These changes get undone in 6/29/09 version:]
Adr30[13] cfeb_rxd_posneg Sets receive data posneg clock polarity (new signal)
Adr30[14] alct_txd_posneg Sets transmit data posneg clock polarity, (was alct_posneg)
Adr30[15] alct_rxd_posneg Sets receive data posneg clock polarity (new signal)

Changed 8bx constant delay in alct random number pipeline to be VME programmable

Added 2bx to compensate for muonic sync stages.
Default delay is now 8+2-1=9bx
Good spots for reference TMB+ALCT384 occur at pipedepth 4 when alct_sync_rxdata_pre=9
Adr104[15:12] = alct_sync_rxdata_pre[3:0], default=9

Changed Adr 0E[15:11] to connect dmb_tx_reserved[4:0] to dmb_tx[48:44]
Adr 0E[15:11] =dmb_tx_reserved[4:0], just set to 0 for now

Version 06/05/09

Re-structure dmb_tx[48:0] flip-flops to force IOB instantiation
No other changes

Version 05/15/09

Added miniscope to monitor clct pretrigger processing and alct*clct matching.
- Added miniscope VME register Adr 0x10C.
- Rebuilt parity register Adr 0xFA to accept miniscope RAM parity.
- Restructured DMB image RAM from 5 BRAMs down to 4 BRAMs to free up 1 RAM
- Reduced ALCT raw hits storage RAM from 2048bx down to 1024bx to free up 1 RAM

Replaced Virtex-E era RPC de-mux and pipeline stages to minimize latency.
- Added clct pre-trigger signal to RPC readout to DMB in a former always-zero bit.
- Added data=address test mode to RPC storage RAM

Added 8bx constant delay in alct random number pipeline. Good spots at depth 12 before are now at 4.
Removed legacy alct signals from Adr 0x38,0x30 and Hdr 30[12:7] that are now used for ECC parity.
Added alct_ecc_err_blank to Adr 038[2] to blank alct data that has uncorrected ecc errors.
Added counter[6] to count alct data blanked due to uncorrected ecc errors

Adr 0x30: Removed Adr30[11:8] alct_reserved_out[3:0], as these bits now carry ecc parity.

Adr 0x38: Adr38 has been reorganized to make room for the new alct_ecc_err_blank signal

Moved alct_ecc_en to Adr38[1]	**n.b. this affects loop-back test software**
Added alct_ecc_err_blank to Adr38[2]	blanks alcts with uncorrected ecc errors.
Moved alct_sync_ecc_err to Adr38[4:3]	**n.b. this affects loop-back test software**
Removed Adr38[2:1] seq_status[1:0]	
Removed Adr38[4:3] seu_status[1:0]	
Removed Adr38[8:5] reserved_out[3:0]	

```
alct_stat_rd[0]     = alct_cfg_done;         // R   ALCT FPGA loaded
alct_stat_rd[1]     = alct_ecc_en;           // RW  Enable ALCT ECC decoder, else do no ECC correction
alct_stat_rd[2]     = alct_ecc_err_blank;    // RW  Blank alcts with uncorrected ecc errors
alct_stat_rd[4:3]   = alct_sync_ecc_err[1:0];// R   ALCT sync mode ecc error syndrome
alct_stat_rd[11:5]  = alct_stat_wr[11:5];    // RW  Free
alct_stat_rd[15:12] = alct_txd_delay[3:0];   // RW  ALCT data transmit delay, integer bx
```

Adr 0xFA: AdrFA has been reorganized to make room for the new miniscope RAM parity
perr_adr[] expanded from 3 to 4 bits.

Adr 0x10C: New Miniscope control register.

Adr 0xBC[14] Added rpc_tbins_test for RPC RAM data=address test mode

Adr 0xCA[9] Moved bx0_match to [10]
Added bx0_vpf_test to [9]

Hdr27[13] now ORs miniscope RAM parity errors with RPC RAM parity errors

Hdr19[14] vme_exists replaced by mini_read_enable

RPC readout format: unused tbin bit[11] was always 0, now has clct-pretrigger flag
Counter[06]: Inserted scnt[06]="ALCT: trigger path ECC>=2-bit error, ALCT discarded"
Shifts subsequent counters up by 1 address.

TMB readout format changed to include miniscope data and markers when mini_read_enable=1.

By default, inserts B07 marker, 22 scope words, then E07 marker after RPC data.

ALCT legacy cable signals that are now ecc parity updated in
Table 25: J5 ALCT Cable1 Connector [J10 on ALCT board]
Table 17: J6 ALCT Cable2 Connector [J11 on ALCT board]

Versions 04/07/2009 - 04/14/09

Added TMB-to-ALCT sync-stage and inter-stage in alct.v module.
Improves alct_rx_clock windows, and allows a ½-cycle shift
in alct_rx_clock at inter-stage, but adds 2bx to output signals.

Add Adr 30[14] alct_posneg

Modified UCF to constrain ALCT inter-stage flip-flops to FPGA slice locations near ALCT IOBs

Compiled 1 version with ALCT 80 MHz IOBs set to Slew=Fast | Drive=12
Another version has ALCT 80 MHz IOBs set to LVDCI_33 (Digitally Controlled Impedance, 50Ω)

Added TMB-to-ALCT 80MHz diagram to this doc

Version 03/16/2009

New Features:

Error Correcting Code to ALCT-to-TMB trigger data path, using reserved rx signals
Error Correcting Code to TMB-to-ALCT TTC command path, using reserved tx signals
Separate bxn counter and offset for L1A

New event counters + re-numbered ALCT-counter-group 0-11 [see counter register adr 0x000]
(see AdrD0 p59 for details)

3 for ECC rx data
3 for ECC tx data
5 for individual CFEB pre-triggers
1 for alct_bx0

Register Changes:

Adr 016 change delay_ch0[3:0] alct_tx_clock default to 11 for use with reference ALCT

Adr 076 rename from adr_seq_offset to adr_seq_offset0
Adr 076 rename bxn_offset[11:0] to bxn_offset_pretrig[11:0]

Adr 0D0 increase cnt_select[6:0] by 1 bit to address more event counters
Adr 0D0 move perr_reset from AdrD0[15] to AdrFA[6]

Adr 10A new register adr_seq_offset1
Adr 10A add bxn_offset_pretrig[11:0], which is a separate bxn offset for the new L1A bxn counter

Adr038[10:9] new signal: alct_sync_ecc_err[1:0] is ALCT sync-mode ECC error code, readonly
Adr038[11] new signal: alct_ecc_en is ALCT ECC trigger data correction enable with default=1
N.B. setting alct_ecc_en =0 stops ALCT trigger data correction, but does not affect ECC counters

Header Changes:

Header30[6:5] now contains alct_ecc_err[1:0]

ALCT Cable Signal Changes for ECC

Old	New	Comment
reserved_in[0]	parity_in[0]	ECC parity [5:0] for TMB-to-ALCT
reserved_in[1]	parity_in[1]	“”
reserved_in[2]	parity_in[2]	“”
reserved_in[3]	parity_in[3]	“”
seq_cmd[0]	seq_cmd[0]	Activates ALCT sync mode
seq_cmd[1]	parity_in[4] seq_cmd[1]	Parity sent unless in sync mode
seq_cmd[2]	seq_cmd[2]	Activates ALCT sync mode
seq_cmd[3]	seq_cmd[3] | parity_in[5]	Parity sent unless in sync mode
seq_status[0]	parity_out[0]	ECC parity [6:0] for ALCT-to-TMB
seq_status[1]	parity_out[1]	“”
seu_status[0]	parity_out[2]	“”
seu_status[1]	parity_out[3]	“”
reserved_out[0]	parity_out[4]	“”
reserved_out[1]	parity_out[5]	“”
reserved_out[02]	parity_out[6]	“”

Version 02/05/2009

Added ALCT-TMB sync mode loop-back test logic to alct.v module.

Mod Adr F2: Change compiler ID field to accommodate extra digit for ISE 10.1I
Add Adr104, Adr106, Adr108 for ALCT sync-mode data

Version 01/13/2009

Rename ALCT cable 2 pair 15 from reserved_in4 to seq_cmd3
Adr 30: alct_cfg: rename Adr30[7] to seq_cmd3, replaces reserved_in4

Version 12/10/2008

Add L1A-only readout mode with full header and raw hits
Add L1A lookback offset in new VME register Adr 0x100
Add sequencer debug signals to new VME register Adr 0x102

Replaced parity errors Adr 0x0FA with sub-adr multiplexing
Add 35-bit RAM parity error array to Adr 0x0FA

Version 11/18/2008

Replaced L1A data storage logic:
Moved L1A data from header RAM to fence queue RAM
Allows L1A-only TMB readout mode to have valid data in short-header (i.e. bxn at L1A arrival)
To enable L1A-only TMB readout
[1] fifo_mode=3 sets short header
[2] l1a_allow_notmb=1 allows readout when tmb didn't trigger for that L1A
[3] turn off TMB pre-triggers (set mask_all=0 or halt pre-trigger machine)
[4] send L1A via TTC....all ~500 TMBs should send short header to DDU

Inverted raw hits RAM parity.
Now parity bit=1 if RAM data[7:0]=8'b00000000
Modified CFEB and RPC raw hits RAMB16s to be read-first instead of write-first.
Guarantees parity data on port B is valid before writing new data to port A.

AdrFE[2] BXN latched at last L1A

New Event Counters
Inserted 2 new counters after counter[47]
counter[48]= Sync error, bxn!=offset at bx0 arrival or bx0 did not arrive at bxn==offset
counter[49]= Raw hits RAM parity error, possible radiation SEU

Note on enabling internal scope readout to DMB/DDU
Adr98 = 0x108B
Adr9A = 0x0000
AdrCE = 0x0000

Version 09/30/2008

AdrD4[2] add jsm_sel to select old/new alct user prom format.
This is a write-only bit, it reads back the value of vsm_jtag_auto

New Internal Scope Logic:

Allows scope channel data to be inserted in DMB readout stream
All scope channel signal assignments have bee replaced

Adr98 Replaced with new internal scope signals
Adr9A Replaced with new internal scope signals

To enable scope data in DMB readout, set
scp_ch_trig_en =1
scp_runstop=1
scp_force_trig=0
scp_auto=1 (also appears in Header19[13]
scp_tbins=4 (may be 0 to 7, number of scope tbins = 64*(scp_tbins+1), thus spanning 64 to 512)
scp_nowrite=0

Rename Event Counters to better describe their functions
counter[22]= TMB matching discarded an ALCT pair (all alcts in the pair were discarded)
counter[23]= TMB matching discarded a CLCT pair (all clcts in the pair were discarded)

New Event Counters
Inserted 2 new counters after counter[23]
counter[24]= TMB matching discarded CLCT0 from ME1A
counter[25]= TMB matching discarded CLCT1 from ME1A
Shifts all subsequent counter addresses up by 2

Header Updates:

header08_[12]	r_tmb_clct0_discard;	TMB discarded clct0 from ME1A
header08_[13]	r_tmb_clct1_discard;	TMB discarded clct1 from ME1A

Version 09/12/2008

Added tmb_trig_pulse to header40[14]
Added tmb_trig_keep to header41[9]
Added tmb_non_trig_keep to header41[10]

Version 09/05/2008

Add blocking of LCTs to MPC for ME1A

Version 08/28/2008

Logic modifications:

1) Firmware compile type codes introduced, replaces programmable stagger and reversal
2) Added hs reversal for ME1A and ME1B and full hs reversal for non-ME1A/B CSCs
3) Added blocking for ME1A to MPC, but is not yet functional in this release
4) Added 2 state machines to detect and count TTC lock loss signals from CCB

Adr 2E[9]: Add ccb_ttcrx_ready TTC ready signal from CCB
Adr 2E[10]: Add ccb_qpll_locked Lock signal from CCB
Adr 2E changed register symbolic name from ADR_CCB_STAT to ADR_CCB_STAT0

Adr FA[15:0] Add new register ADR_PARITY contains parity SEU error status
Adr FC[15:0] Add new register ADR_CCB_STAT1 contains TTC lock status from lock state machines

Rename event counter [5] "Pre-trigger was on any cfeb"
Inserted 2 new event counters after counter[5], shifts all other counter addresses up 2

- [6] Pre-trigger was on ME1A cfeb4 only
- [7] Pre-trigger was on ME1B cfebs0-3 only

Add 2 new event counters after counter[57],

- [58] CCB: TTCrx lock lost
- [59] CCB: qPLL lock lost

Add new register AdrFC for CCB lock detection

AdrFC[00]	ccb_ttcrx_lock_never	TTCrx lock never achieved
AdrFC [01]	ccb_ttcrx_lost_ever	TTCrx lock was lost at least once
AdrFC [02]	ccb_qpll_lock_never	QPLL lock never achieved
AdrFC [03]	ccb_qpll_lost_ever	QPLL lock was lost at least once

Extended trigger source vector in VME and Header:

- Adr7C[9] me1a_only_pretrig
- Adr7C[10] me1b_only_pretrig
- Hdr40[12]=r_trig_source_vec[9]
- Hdr40[13]=r_trig_source_vec[10]

Introduced Firmware Compile Type Codes

- A=Normal CSC
- B=Reversed CSC
- C=Normal ME1B, Reversed ME1A
- D=Reversed ME1B, Normal ME1A

Extended ½-strip reversal and ME1A/B signals to Adr CC

Adr CC[05] csc_me1ab	1= CSC is ME1A or ME1B. 0=normal CSC
Adr CC[06] stagger_hs_csc	1=Staggered Adr CSC, 0=non-staggered
Adr CC[07] reverse_hs_csc	1=Reversed staggered CSC, non-me1
Adr CC[08] reverse_hs_me1a	1=reversed me1a hstrips
Adr CC[09] reverse_hs_me1b	1=reversed me1b hstrips
Adr CC[15:12] csc_type[3:0]	Firmware compile type A, B,C or D

Added ½-strip reversal signals to Header

Header does not contain csc_type explicitly, but csc_type can be inferred from reversal signals in Hdr39[14:12] and Hdr20[14]

Hdr20[14]	stagger_hs_csc	CSC Staggering ON
Hdr39[12]	reverse_hs_csc	1=Reverse staggered CSC, non-me1
Hdr39[13]	reverse_hs_me1a	1=ME1A hstrip order reversed
Hdr39[14]	reverse_hs_me1b	1=ME1B hstrip order reversed

Added ME1A LCT blocking to MPC [not yet functional in this release]

Adr CC[03] mpc_me1a_block	Block ME1A LCTs from MPC, still queue for readout
Adr CC[04] cnt_non_me1ab_en	Allow clct pretrig counters count non me1ab events

End of 8/28/2008 mods

Version 08/12/2008

Adr38[15:12] add alct_txd_delay[3:0] to delay alct tx data, delay=0 by default has same timing as previous firmware versions.

Adr CA[9]: add bx0_match
Hdr30[14]: add bx0_match

Version 08/04/2008

Adr F6[6]: add clct_sep_ram_sel_ab to select A or B separation RAM data readback

Version 08/01/2008

Adr F4[1]: stagger_csc is now read-only, set at firmware compile time

Header36: r_nrpcs_read in header now gated with rpc_read_enable.
now indicates 0 rpcs when rpc readout is disabled

Version 07/15/2008

Added ability to readout non-triggering events

Header41[0] = VME settings for tmb_allow_alct, for trigger and readout
Header41[1] = VME settings for tmb_allow_clct, for trigger and readout
Header41[2] = VME settings for tmb_allow_match, for trigger and readout

Header41[3] = VME settings for tmb_allow_alct_ro, for non-triggering readout
Header41[4] = VME settings for tmb_allow_clct_ro, for non-triggering readout
Header41[5] = VME settings for tmb_allow_match_ro, for non-triggering readout

Header41[6] = alct-only non-triggering event
Header41[7] = clct-only non-triggering event
Header41[8] = alct*clct match non-triggering event

Header41[9] = This event is a non-triggering readout

New Event counter at subadr[19] counts non-triggering events queued for readout
Shifts all subsequent counter addresses up 1

New VME register adrCC:
AdrCC[0] = tmb_allow_alct_ro allow alct-only non-triggering event readout
AdrCC[1] = tmb_allow_clct_ro allow clct-only non-triggering event readout
AdrCC[2] = tmb_allow_match_ro allow alct*clct-match non-triggering event readout

Version 07/09/2008

Replaced entire ALCT UserPROM JTAG State Machine
New compressed data format: see JTAG PROM-1 section p90

Adr 70: Move dmb_thresh[2:0] from adr F4[8:6] to adr70[9:7]
Adr 70: Rename dmb_thresh to dmb_thresh_pretrig
Adr 70: Rename hit_thresh to hit_thresh_pretrig
Adr 70: Rename nph_thresh to hit_thresh_postdrift

Adr F0: Rename lyr_thresh[2:0] to lyr_thresh_pretrig[2:0]

Adr F4: Remove dmb_thresh from F4[8:6]
Adr F4: Add new signal pid_thresh_postdrift[3:0] to F4[9:6]
Adr F4: Move adjcfeb_dist from F4[14:9] to F4[15:10]
Adr F4: Rename pid_thresh to pid_thresh_pretrig

Add adr D8[12] jsm_tckcnt_ok JTAG PROM TCKs sent matches TCKs in trailer frame
Add adr D8[13] jsm_end_ok JTAG PROM FF end marker found where expected
Add adr D8[14] jsm_header_ok JTAG PROM BA begin marker found where expected
Add adr D8[15] jsm_chain_ok JTAG PROM Chain Block marker found where expected

Add adr DE[14:13] jtag_sm_vec[1:0] JSM JTAG signal state machine vector

Add adr E0[11:8] jsm_prom_sm_vec[3:0] JSM PROM state machine vector
Add adr E0[14:12] jsm_format_sm_vec[3:0] JSM Data format state machine vector

Add adr EA[15] jsm_tckcnt_ok JSM tckcnt added to board status

Header20: remove header20_[12:10] lyr_thresh[2:0]
Header20 remove header20_[13] layer_trig_en
Header20 add header20_[13:10] pid_thresh_postdrift[3:0]

Header41: add header41_[13:11] lyr_thresh_pretrig[2:0]
Header41 add header41_[14] layer_trig_en

New Counters:
Inserted 2 counters after counter at SubAdr[8]
SubAdr[9]CLCT: CLCT0 passed hit thresh but failed pid thresh after drift
SubAdr[10]CLCT: CLCT0 passed hit thresh but failed pid thresh after drift
Shifts all other counter addresses up by 2, i.e. old counter at SubAdr[9] moved to [11]

Change CLCT Processing Algorithm at bx11 to also require pid $\geq$ pid_thresh_postdrift

Version 06/03/2008

Modifies global_reset and ttc_resync behavior
Adds ability to send active feb flag to DMB at tmb alct*clct matching, retains ability to send at pre-trig
Overlays ALCT rx data with normal scope channels to aid alct debugging

(1) Remove Adr A8[12] alct_raw_sync, wasn't being used
(2) Add temporary alct structure error counters [48]-[52]
(3) Add Adr D0[6] cnt_clear_on_resync clears VME counters [0]-[40] on ttc_resync, default=0
(4) Add Adr D0[7] hdr_clear_on_resync clears header counters [41]-[47] on ttc_resync, default=1
(5) Update MPC frame format doc (reflects changes to 5/12/08 firmware)
(6) Added a startup state to the readout state machine to wait 1bx for buf_q_empty to update after a reset (prevents machine from resuming a readout that was in progress at the time of a ttc_resync)
(7) Add perr_reset (one-shot) to Adr D0[15], removed ttc_resync perr reset logic
(8) Block ttc_resync from clearing resync event counter, requires vme-clear
(9) Adr2A[3] change ccb_status_oe default from 1 to 0, turns off backplane drivers to ccb
Added write-only bits to parallel non-decoded ccb commands:

Adr2A [12]	vme_evcntres	Event counter reset || ccb_evcntres
Adr2A[13]	vme_bcntres	Bunch crossing reset || ccb_bcntres
Adr2A14]vme_bx0		Bx0 signal || ccb_bx0

(10) Adr28[12] now contains global_reset_en=1 to enable resets on DLL lock-lost
(11) Header08[14] now contains clock_lock_lost
(12) AdrAC[14]=active_feb_src, 0=pretrig, 1=at tmb matching
(13) AdrB0[7:3]=clctf[4:0] active cfeb list at tmb matching
(14) Header23[4:0] active feb list is stored either at pretrig time or tmb match, depending active_feb_src
(15) Header23[14] now contains active_feb_src bit
(16) Header29[14] now contains hs_layer_trig (moved from header23[14])
(17) AdrCE[15]=scp_ch_overlay, 0=normal scope channels, 1=use debugging channel overlay
Current overlay assignments:
scp_ch[71:0] = normal
scp_ch[128:72] = scp_alct_rx[55:0]

```
assign      scp_alct_rx[0]          = alct_active_feb_flag;
assign      scp_alct_rx[1]          = alct_first_valid;
assign      scp_alct_rx[2]          = alct_first_amu;
assign      scp_alct_rx[4:3]        = alct_first_quality[1:0];
assign      scp_alct_rx[11:5]       = alct_first_key[6:0];
assign      scp_alct_rx[12]         = alct_second_valid;
assign      scp_alct_rx[13]         = alct_second_amu;
assign      scp_alct_rx[15:14]      = alct_second_quality[1:0];
assign      scp_alct_rx[22:16]      = alct_second_key[6:0];
assign      scp_alct_rx[27:23]      = alct_bxn[4:0];
assign      scp_alct_rx[28]         = #alct_wr_fifo;
assign      scp_alct_rx[29]         = alct_first_frame;
assign      scp_alct_rx[43:30]      = alct_daq_data[13:0];
assign      scp_alct_rx[44]         = alct_lct_special;
assign      scp_alct_rx[45]         = alct_ddu_special;
assign      scp_alct_rx[46]         = alct_last_frame;
assign      scp_alct_rx[48:47]      = alct_seq_status[1:0];
assign      scp_alct_rx[50:49]      = alct_seu_status[1:0];
assign      scp_alct_rx[54:51]      = alct_reserved_out[3:0];
assign      scp_alct_rx[55]         = alct_cfg_done;
```

Version 05/23/2008

(1) Adr B0[15:14] Add clock_lock_lost and sync_er

Version 05/12/2008

(1) mpc alct_bx0 and clct_bx0 signals now bypass mpc_tx_delay
(2) Note: Adr86[13] default=0 uses ttc_bx0, set it to 1 to use local bxn counter instead
Adr86[14] default=0 enables bx0 to mpc continuously, 1 blanks mpc frames unless triggering
(3) Adr CA[8] default changed to 1 to enable using reserved[3] signal from alct as alct_bx0
(4) Adr90[15:14] now contains bx0 injector one-shots

Version 05/01/2008

(1) All scope channels replaced, see Scope Channel Assignments p48
(2) Header11 CLCT counter was behind by 1 event, fixed.
(3) Header22 Trigger source for alct*clct matching is set even if match mode is off, fixed.
(4) Header28/29/30 ALCT data was latched n-bx early, leaving empty frames, fixed.

Version 04/29/2008

(1) ALCT signal reserved_out[3] (alct-to-tmb) is now alct_bx0

(2) Adr 68[02] renamed match_pat_trig_en to alct_match_trig_en

(3) Adr AC[5] hdr_wr_continuous should be set to 0 unless using l1a_allow_notmb=1 mode

(4) Adr AC[14] removed allow_pretrig_noflush bit

(5) Adr AE[11:0] sequencer state shortened from [14:0]

(6) Adr B2[7:4] Renamed clct_width to clct_window

(7) Adr BA[15:8] Now contains rpc bxn differences moved from Adr C4

(8) Adr C4 renamed to ADR_RPC_TBINS
Adr C4 added rpc tbins, tbins before pre-trigger, and rpc_decouple [=0 to copy cfeb tbins]

(9) Adr CA[] new name ADR_BX0_DELAY, all new signals for bx0_delay and bx0 source

(10) Adr D0[8] now contains counter lower-half / upper-half mux bit cnt_adr_lsb
Adr D0[14:9] now contains counter sub-address cnt_adr[5:0]
Adr D0 Event counters replaced with new names and new sub-address channel numbers

(11) Adr F0[15:8] = clct_throttle[7:0], default=0

(12) Adr F4[0]=clct_blanking=1 (new default), prevented from setting to 0unless l1a_allow_notmb=1 or tmb_alct_only=1

(13) Header21[14:11] Renamed clct_width to clct_window

(14) Header36 replaced to display rpc_tbins and rpc_pretrig, affects event-size calculation software, rpc_exists[1:0] deleted.

(15) Header41 now contains the enabled TMB matching modes

Version 02/5/2008

Added parity checking to cfeb and rpc raw hits RAMs for SEU detection
Added parity error bits to header27
Add drift_delay to header29
Add alct pretrig window position to header28

Version 01/24/2008

Replaced entire L1A logic.
Replaced 8-buffer system with 2048 buffers.
Replaced readout stack with event queue.
Replaced lct_quality.
Several VME addresses have changed:

(1) Adr AC[4] is now wr_buf_autoclr (formerly clct_turbo).

Adr AC[5] 1=allow continuous header buffer writing for invalid triggers
Previously an unused bit. Default remains 0 until new trigger logic is ready.

(2) New buffer status signals now occupy Adrs 9E,A0,A2,A4,A6.

ADR_BUF_STAT0=9E
ADR_BUF_STAT1=A0
ADR_BUF_STAT2=A2
ADR_BUF_STAT3=A4
ADR_BUF_STAT4=A6

Old Adr A2 (alctfifo1) moved to A8 (old A8 was empty)
Old Adr A4 (alctfifo2) moved to AA (old AA was empty)
Old Adr A6[5:0] adjcfeb_dist[5:0] moved to F4[14:9]

(3) Adr 72[13] is now fifo_no_raw_hits [1=do not wait to store raw hits. A no_daq mode.]

(4) Adr 74[15:13] is now l1a_internal_dly[2:0] (mostly for use by the simulator)

(5) Adr AE[14:11] signal names changed for buffer status

(6) Adrs B0, 78, 7A contain new CLCT internal format

(7) Header04[13] is now buf_q_ovf_err, formerly stack_ovf_latch

Header05[14] is now buf_stalled, formerly buf_full

Header37,38,39, 40 changed for buffer status [some assignments are probably temporary]

Header25,26,27 contain new CLCT internal format

(8) Scope channels replaced

(9) Counters 1A, 1C, 1E changed, now count debug signal presence instead of time-outs

Version 10/11/2007

(1) CLCT raw hits CRC now stops at the frame before the DE0F marker due to DDUs failure to include the marker.
(2) ALCT raw hits CRC check stops at the frame before the DE0D marker, same reason.

Version 9/14/2007

(1) Header bug fixes in readout counter and lct-duplication flags
(2) New no-alct VME counter, channel 32/33, shifts subsequent channel numbers up 2
(3) ALCT DDR transmitter constraints minimize routing delays between alct_rx_clock and main clock.

Version 9/10/2007

(1) New raw hits readout header+trailer format replaces all previous header field assignments.
(2) New VME counters, channels 32 to 4C.

Version 7/10/2007

(1) Increase tmb.v-sequencer.v handshake time-out from 8bx to 15bx to prevent late alct*clct matches from being counted as discarded events when using higher than normal clct_width values.

(2) Modify pre-trigger state machine to wait for alct*clct matching (or clct-only) before re-arming for the next pre-trigger event to prevent writing current event to wrong header buffer when using higher than normal clct_width values.

(3) Modify pre-trigger state machine to wait for active_feb signal to return to 0 before the next pre-trigger event to avoid re-triggering on same event when using longer than normal triad_persistence.

Version 7/05/2007

(1) Pattern-finder key layer shifted from ly3 back to ly2.
Ly3 pattern templates were flipped top-to-bottom to shift key layer to ly2, and flipped left-to-right to preserve bend direction.

(2) Adr A6 adjcfeb_dist[5:0] is now 6-bits instead of 5 to allow dist=32 to span a full cfeb, default value remains 5hs.

(3) Adr F4 dmb_thresh[3:0] default is now 4 to reduce spurious active feb signals to DMB

(4) CLCT Processing Algorithm description updated for current patterns

Version 6/21/2007

(1) Adr D4 bit 11 now contains wr_usr_jtag_dis.
When wr_usr_jtag_dis=1, write access to register adr_usr_jtag is blocked.
This allows parallel writes to jtag chains for selected alcts.
Adr 10 bit 14 indicates the state of wr_usr_jtag_dis.

(2) Adrs A0, A8, AA, CA, CC are now obsolete.
Their FFs have been removed, and they read back 0000h.

(3) Adr A6 now contains adjcfeb[4:0] with a default value of 5 hstrips.
This replaces the function of mask registers A6, A8, and AA.

If there is n hstrip key on hs 0,1,2,3,4 on CFEBn, with hits $\geq$ hit_thresh_pretrig, then CFEBn-1 will be marked in the active_feb list for DMB readout.

If there is a hstrip key on hs 31,30,29,28,27 on CFEBn, with hits $\geq$ hit_thresh_pretrig, then CFEBn+1 will be marked in the active_feb list for DMB readout.

(4) Adr 6C, layer_trig_dly has been removed.

(5) Header05[12] now contains trigger source vector bit [8] (layer trigger)
Header05[13] hsds bit removed.

(6) Header16[14:11] now contains pid_thresh[3:0] instead of ds_thresh[2:0].

(7) References to hs_thresh[2:0] have been changed to hit_thresh_pretrig[2:0].
References to ds_thresh[2:0] have been removed.

(8) Adr 70, ds_thresh[2:0] has been removed.

(9) Pattern ID numbers have been shifted from 0-to-8 to 2-to-10.
Pattern ID=1 now indicates a layer-trigger event.
Pattern ID=0 now indicates no pattern matches found.

(10) Adr F4, clct_blanking=1 is the new default value