Fermi National Accelerator Lab

BLM Upgrade Control Card Program Design

Randy Thurman-Keup

AD / Instrumentation Deptartment

July 10, 2008

BEAMS-DOC-2433-V9

CVS Code Version v15

Abstract

This document details the structure and behavior of the program that runs in the eZ80 on the Controller Card. The main functionality of this program is to start and stop data acquisition and to maintain data buffers that are accessible from VME at any time without disrupting the aborting capability. These data buffers consist of three circular buffers containing summed data at short, medium, and long integration times, two linear buffers housing profile and flash frames, and a single entry buffer for the most recent display frame.

1 Introduction

The Controller Card (CC) is an embedded processor (eZ80) board residing in the VME crate of the Beam Loss Monitor (BLM) system [1]. The program's job is to transfer information between the crate processor and the digitizer cards (DC), timing card (TC), and abort card (AC), in a deadtimeless fashion, *i.e.* no interruption in the aborting capability of the system. The program operates with a combination of polling and interrupts.

The Controller Card Program (CCP) responds to 5 types of events.

- **TCLK**
  The TC puts relevant TCLK events into its FIFO. The CCP polls the TCLK FIFO status register to determine if there is data in the FIFO. If there is data, it reads it from the FIFO and handles it.
- **MDAT**
  When the TC receives the relevant MDAT frame, it writes the state information to the MDAT FIFO. The CCP polls the MDAT FIFO status register to determine if there is data in the FIFO. If there is data, it reads if from the FIFO and handles it.
- **DATA_LATCH (*Interrupt*)**
  This is an interrupt generated by the Timing Card when it is time to latch some flavor of Digitizer Card data (fast, slow, or very slow sums). The CCP must then read the status bytes in the Timing Card to determine which data to latch.
- **Abort_Service**
  The abort card generates an interrupt when one of three user selectable states occurs.
  - A digitizer card indicates that a channel is not OK
  - A digitizer channel is over one of the thresholds but the multiplicity requirement is not met for an abort
  - An abort has occurred

  The CCP does not respond to the interrupt. Instead it periodically polls the AC to determine the above information.
- **Crate Processor**
  When settings need to be updated, the CP writes the settings to the CC and sets an appropriate register in the CC which then responds by loading the settings into the appropriate cards at the appropriate time.

The CCP might need to do intelligent checking of the state of cards and possibly issue a Reset.

2 Initialization

At boot time, the CP and CC must handshake to properly bring up the system. This handshaking between CC and CP is documented in Figure 1 below. The initialization procedure includes stopping the CC at the beginning of the procedure. Stopping the CC at boot time is necessary for a number of reasons: first, at crate power up time, the other cards will not have gone through their FPGA programming sequence if the CC immediately starts to access them; second, the waiting gives the CP time to download settings to the CC; and third, if the CC reboots by, e.g. hitting an invalid instruction, it would probably be good to notify the outside world, and save a

snapshot of the CC's debug memory contents for later analysis. The settings that need to be downloaded to the CC are listed in Table 1.

- **Timing Card**
  The timing card must be downloaded with the appropriate TCLK events (see Table 3). In addition to the events listed in Table 3, there are several other TCLK and BSCLK events that the CCP does not respond to but which are responded to by the other cards in the system.
    - TCLK $8F – 1 Hz event for updating the TC clock
    - TCLK $5B – TeV Pbar Injection TBT trigger
    - TCLK $5C – TeV Proton Injection TBT trigger
    - BSCLK $AA – Revolution marker used to generate Make_Meas clock
    - BSCLK $DA – MI TBT trigger
- **Digitizer Card**
  The digitizer card must be setup for the correct memory map. The correct map is enabled by setting bit 7 of address 0xFF to 1. All other settings are part of the CP download.
- **Abort Card**
  The abort card settings are all downloaded as part of the CP download.

After successfully initializing, the CCP continuously polls for events, periodically interrupted by the Data Latch interrupt.

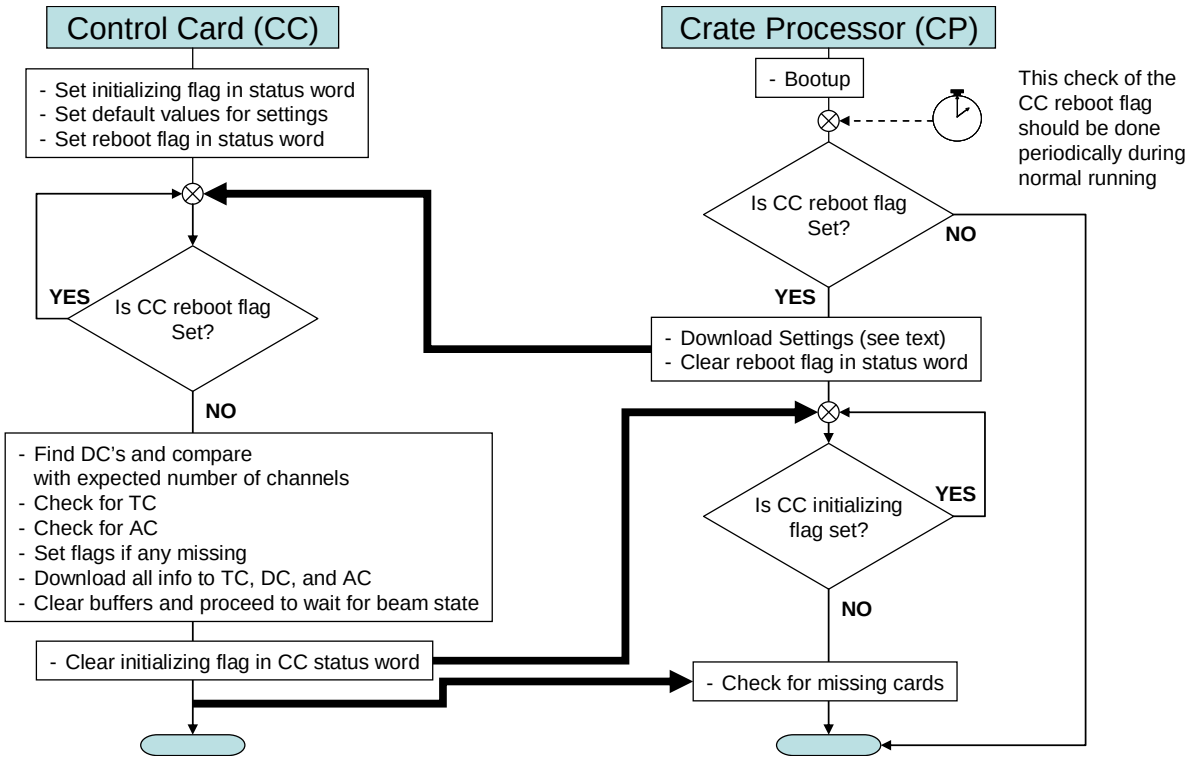


Figure 1: Flowchart of handshaking that occurs at boot time. The alarm clock symbol indicates that the CP should periodically check the reboot bit in the CC and if it finds it set, it should probably notify the outside world somehow. See Section 3 for a description of the flags in the status word.

Table 1: Settings that need to be set on the CC by the CP. The default value is the value that is used by the CCP if not overwritten by the CP and is a compile time setting in the CCP.

Address	Size	Setting	Description	Default Value
000004	2	Derippled or Fast sum	Controls whether the Flash/Profile/Display frames contain Fast sums (0) or Derippled sums (1) (for those that use Fast as opposed to Slow)	0x0000
000006	2	Flash, Display, Profile source	This determines the source of the first half of the frame (0=fast or 1=slow). Bit 0 is for Flash, bit 1 is for Profile, and bit 2 is for Display.	0x0006 (uses fast for flash, slow for others)
000014	4	System Time	Unix time in seconds since ??? Little Endian word order	????
00001C	2	Machine	Which machine is this (1=TeV, 2=MI, 3=Booster)	0x0002
00001E	2	Initial state	Initial MDAT machine state	0x0000
000100	2	# of Channels	Expected number of channels	Whatever is present
000102	2	Make Measure Div	Amount to divide down the clock by (int. osc.)	0x0001 for TeV 0x0002 for MI

Address	Size	Setting	Description	Default Value
000104	2	Fast Sum Length	# of measurements to accumulate in DC fast sum.	64
000106	2	Slow Sum Length	# of measurements to accumulate in DC slow sum.	1590 for TeV 1504 for MI
000108	2	Very Slow Sum Length	# of measurements to accumulate in DC very slow sum. (In MI this is the integral and must be 1/16 the pedestal length)	47710 in TeV 47 in MI
00010A	2	DC FPGA Control Register	Funny name for something which contains the number of make_meas to skip before doing pedestals divided by 16 (N_{skip} / 16) and the derippling down conversion offset divided by 32 (see the User's Guide for details)	0x10CC in TeV 0x1000 in MI
00010E	2	TC Operation mode	Value to be written to TC control bus CSR. Controls whether TC uses AA marker or internal oscillator.	0x0004 in TeV 0x0000 in MI
000112	2	IRQ3 Enable Bits	3 bits that determine what generates an IRQ3 interrupt on the AC	0x0007
000114	2	Abort Enable	Bit 0 Enables the functioning of the AC, Bit 4 requires two consecutive make_meas cycles with an abort before actually pulling the abort	0x0011
000116	2	Pedestal Length	Must be 16 * Very Slow Sum Length in MI (doesn't matter in TeV)	795 in TeV 752 in MI
000118	2	End of beam delay	Delay (in Fast latch periods) after end-of-beam event before asserting AIP	0x0012
00011A	2	Flash Delay	Delay from flash frame clock event until data is grabbed from buffer	0x0000
00011C	2	Profile Delay	Delay from profile frame clock event until data is grabbed from buffer	0x0000
00011E	2	Display Delay	Delay from display frame clock event until data is grabbed from buffer	0x0000
000120	2	Input Switch state for peds	Whether (1) or not (0) to open the input switch while taking pedestals	0x0001

Address	Size	Setting	Description	Default Value
000126	2	Delay after F sector end TCLK event	Delay (slow sum units) after the end TCLK event before re-enabling normal F sector aborts	0x0002
000128	2	Max DY	Maximum difference between successive CIC sums to replace baseline waveform (Deripple parameter)	256
00012A	2	CIC Sum Length	# of measurements to accumulate in the CIC sum (Deripple parameter)	133 in TeV 128 in MI
000200	2 per chan	DC Mode Selects	Digitizer channel mode select word (one word / channel). Controls integration mode and whether squelch is enabled among other things.	0x0002 for TeV 0x000A for MI (no squelch)
000202	2 per chan	DC Manual Set Value	The manual setting value if manual setting mode is chosen	0x0000
000400	120	Squelch Values	This must be in the appropriate units which is $X(\sigma_{raw}\sqrt{16\times17\times n_{vs}})$ where X is the # of sigma to place the squelch	0x0000 for now since we are not squelching
0E0000	256	MDAT to Abort State map	Maps MDAT state to abort state	0 → 0, 1 → 1, etc…
0E0100	256	F sector Start TCLKs	Starting MI TCLKs for F sector aborts	0x00
0E0200	256	F sector End TCLKs	Ending MI TCLKs for F sector aborts	0x00
0F0000	64K	F sector actions	See Text about F sector aborts	0x00
100000	256K	Abort Info	Thresholds, Masks, and Multiplicities for all States	Thresh 0xFF, Masks 0x00, Mult 0xFF, Crate Mask should be 0xFF

3 Program Components

3.1 CC VME Status Register

The status register consists of 16 bits in VME memory. Table 2 shows the bit definitions of the status word.

Table 2: Bit definitions in the CC Status word.

Bit	Description
15	Running
14	ERROR line on backplane has been asserted (latched until CC reboot)
13	DeRippled Buffer has wrapped

Bit	Description
12	Mismatched raw data pointers in TC, DC, or AC
11	Pedestals Valid; Don't read pedestals until this is set
10	Very Slow Buffer has wrapped
9	Slow Buffer has wrapped
8	Fast Buffer has wrapped
7	Wrong number of Digitizer Card channels found
6	Abort Card not found
5	Timing Card not found
4	Crate has triggered an Abort
3	Some channels are indicating abort
2	Some channels are not OK
1	CCP is in the process of initializing
0	The CC has rebooted; clearing this kicks off the CC

3.2 VME Accessible Circular Data Buffers

The data from all the digitizers is stored in VME memory in the form of circular buffers. There are index counters which indicate which frame is the current frame, and flags in the status register to indicate when each buffer has wrapped around. Each data frame header contains a flag byte which indicates a number of things. At end of beam, the last frame contains a 1 in the flag byte. This is nominally to allow the CP to ignore this frame at the beginning of the next beam cycle. The first frame of a cycle has a 2 in the flag byte to allow the CP to correct for a bug in the system whereby the DC waits to start summing, but the TC sends out latches immediately. The result is that the slower sums are incomplete when the first latch is received. The CP can divide by the actual sum length if it knows a frame is the first one of the cycle. The next frames contain a 3 in the flag byte until the input switch is closed. All other frames contain 0 in the flag byte.

3.3 State Machine

Figure 2 illustrates the CCP state machine.

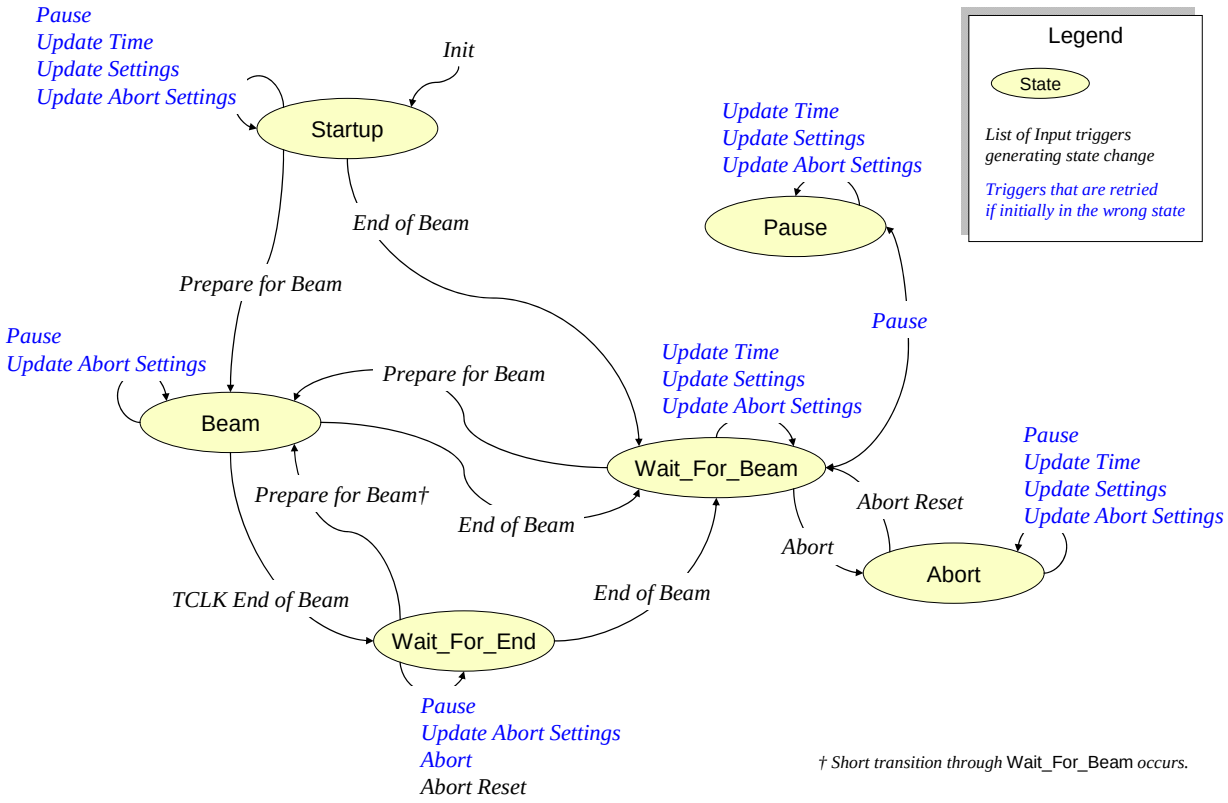


Figure 2: State Machine diagram showing states and state change triggers. The triggers in blue are persistent triggers in that they hang around if they occur during an invalid state. When a valid state is reached, they are replayed.

Figure 3 illustrates a typical beam cycle.

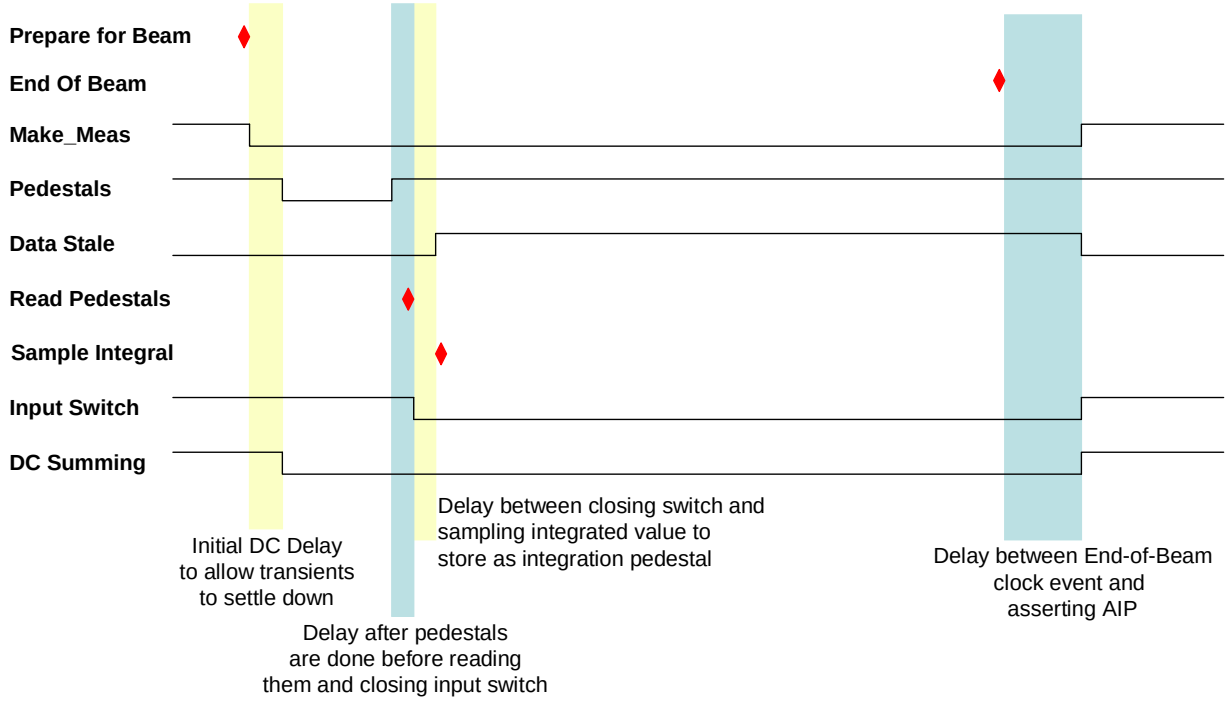


Figure 3: Beam cycle showing the relative timing of various events. The shaded areas are delays between various key events. The integrated value must be sampled after closing the input switch, since there is a glitch upon closing the switch causing the integral to go negative by some amount. The lines are true low.

3.4 Abort Information

Abort information is specified as a function of abort state. The CCP receives MDAT frames containing a state which is then mapped to abort state. This mapping is created by the console application user and sent to the CP which then copies the map to the CC as part of update abort settings.

3.4.1 Handling of Tevatron F Sector Aborts

The abort handling in the Tevatron is more complicated than in the Main Injector. Since there is routinely beam traversing the p2 and p3 beam lines in F sector located directly above the Tevatron beamline, there may be false aborts caused by losses from those lines hitting the Tevatron BLMs. To reject these false aborts, the CCP uses a table of MI states and corresponding begin and end TCLK events during which to change the behavior of the aborts. The default behavior is to mask off the aborts at the AC. The actual implementation on the CCP is in the form of the list of begin / end TCLKs indexed by MI MDAT state, and a table of actions indexed by TEV abort state and MI MDAT state. Both the list of TCLKs and the action table are byte arrays. A TCLK entry of 0x00 means do nothing. The 2-D action table is arranged such that the MI MDAT state index increments the fastest. The actions are defined in the following table

Action #	What it does
0x00	Globally mask off the aborts at the AC
0xFF	Do nothing
0xNN	Switch to TEV abort state 0xNN

As stated above, the default is set to be 0x00 which masks off the crate aborts. The memory maps for these lists are given in Appendix B.

3.5 Debugging Information

There is a section in VME accessible memory that contains possibly useful information for debugging and error trapping. There are counts of various quantities like the number of TCLKs, the number of each type of data latch, etc… This list is in Appendix A and in the BLM Users' Guide [1]. In addition, there is a CP trigger to dump Control Bus memory to a section of VME memory.

There are debugging features built in to the code that are worth calling out here: a collection of buffers to record the CPU time used for a variety of events, and a buffer to record the time of arrival of TCLK's, MDAT's, end-of-beam, aborts, settings updates, and abort settings updates. These two data collections can be read out through VME. Each of these methods uses a separate programmable reload timer running off the system clock and set to a time resolution of ~5-6 μs. The latter is referred to as an oscilloscope in the code and makes use of an interrupt generated from the timer which then increments a counter to provide a longer period timestamp.

4 External Triggers

4.1 TCLK Event

The code responds to TCLK events in the form of polling the TC. A supported TCLK event causes the TC to put the TCLK number in the FIFO. The CCP polls the FIFO by checking the TCLK FIFO status register and reading the TCLK event. Table 3 documents the supported TCLK events.

Table 3: List of CCP supported TCLK events, state inputs resulting from them, and a description of any action taken.

TCLK	State Input	Action
Tevatron		
$77	*None*	Create a flash frame. The current chosen sum and very slow sum buffer frames are appended to the flash frame buffer which contains a maximum of 256 flash frames and is cleared at Reset_DC.
$75	*None*	Create a profile frame. The current chosen sum and very slow sum buffer frames are appended to the profile frame buffer which contains a maximum of 256 profile frames. Cleared at Reset_DC.
$76 $78	*None*	Create a display frame. The current chosen sum and very slow sum buffer frames are placed in the display frame. Only one display frame is allowed at a time.
$71	*Prepare for Beam*	Prepare for beam by issuing a Reset_DC, resetting the circular buffers, and clearing the abort in progress line.

TCLK	State Input	Action
$4B	*End of Beam*	End of beam. Start timer for delay before telling TC to assert Abort In Progress line.
$47	*Abort*	End of beam, then goto abort state and wait for reset.
$48	*Abort Reset*	Get out of abort state and wait for beam.
$70	*None*	Reset linear buffers for flash and profile frames
Main Injector		
$7C	*None*	Create a flash frame.
$7A	*None*	Create a profile frame.
$7B	*None*	Create a display frame.
$79	*Prepare for Beam*	Prepare for beam by issuing a Reset_DC, resetting the circular buffers, and clearing the abort in progress line.
$26	*End of Beam*	End of beam.
$27	*Abort*	End of beam, then goto abort state and wait for reset.
$24	*Abort Reset*	Get out of abort state and wait for beam.

4.2 MDAT Frame

An MDAT frame forces a change of abort thresholds if the machine state has changed. The TC receives MDAT frames and places the state number in the FIFO. The CCP polls the FIFO status and reads the MDAT frame from the FIFO. It then switches to the proper thresholds page in the DC, copies the abort mask information corresponding to the MDAT machine state from VME memory to the AC, and tells the TC to generate an update abort settings signal. MDAT frame $12 contains the Tevatron state, and MDAT frame $56 contains the Main Injector state. The state number received by the CC is in the range 0-255 with 0-127 being the TeV state, and 128-255 being 128 + the Main Injector state.

4.3 Data Latch Interrupt

Periodically, at each sum period, a corresponding latch is generated by the TC by setting IRQ2 on the control bus. The CC is interrupted and the CCP reads the latch status registers in the TC to determine which sum data to read from the Digitizer Card. It then reads the data and stores it in the circular buffers in VME accessible memory. Finally it clears the corresponding status register which in turn clears IRQ2. Most of the actual memory copies from DC's to circular buffers are handled by assembly language instructions that are modularized in C language #defines. The assembly code makes use of specialized instructions to do multi-byte copies from one memory location to another, and is optimized to do most address manipulations in registers.

4.4 Abort Card Service

The AC is periodically polled by the CCP and if the current status indicates over threshold, or abort, or channel not OK, the CCP status register is modified to reflect the state of the AC. If the AC is indicating an abort was requested, the snapshot of the last frame of the abort card before the abort is copied to VME memory for access by the CP. The cumulative OR of the snapshots is also copied to VME memory.

4.5 Crate Processor

The CCP must poll various VME registers to respond to new data and/or requests from the front end. When the CP wants to initiate one of these events, it loads the proper settings and then

writes a non-zero value to the appropriate register. The italicized entries are triggers that are located in the debug section of VME memory.

- **Update Time**
  The Time value must be copied to the TC and the TC must be told to update its time value.
- **Update Settings**
  The DC and TC global settings (e.g. Fast Sum Length) must be copied to the DC and TC boards. This is only done between beam cycles. So in the Tevatron, if there is a necessity to do this during a store, the CCP must be given a fake end-of-beam event (see below) and then restarted after.
- **Update Abort Settings**
  The DC thresholds, and the AC abort settings must be written to the DC and AC boards. Additionally, the TC must be told to assert the update settings line in order for the changes to take effect in the AC. In the DC, changes are immediate. Since they are immediate, the CCP switches the DC to threshold page zero which contains pseudo-infinite thresholds, and then updates the threshold pages and sets the threshold page back to the proper one.
- **Read Pedestals**
  Read the pedestals from the digitizer cards and place them in VME memory.
- **Clear Abort Information**
  Bit 0 tells the CCP to reset the bits in the Status Word that indicate the state of the AC. This also clears the local abort snapshot buffer and resets the abort snapshot buffer on the AC.
  Bit 1 tells the CCP to clear the abort snapshot OR, both locally and on the AC.
  Bit 2 tells the CCP to reset the channel OK list, both locally and on the AC.
- **Read / Write Flash Memory**
  Reading copies the flash memory to the VME memory section reserved for flash downloads/uploads. Writing copies the VME memory section to flash memory. Writing to flash memory involves a protocol and is more elaborate than reading. To initiate a write, the CP must write a 0xA596 to the register.
- **Reboot CC**
  Reboot the CC which asserts RESET_SM on the control bus. For when you really would like to reset the crate but not the CP.
- ***Read/Write Control Bus Memory***
  Read copies 256 bytes of control bus memory to VME memory starting at the specified 16 bit address. The write copies the data byte to the specified 16 bit control bus address. To do this requires writing 0xA596 to the register to avoid accidentally doing this. These are used for debugging purposes and are not part of normal operation.
- ***Run User Function***
  This trigger runs a user specified chunk of assembly code that was downloaded to VME memory. It gets copied to normal RAM and the entry point is called.
- ***Fake Prepare for Beam***
  Initiates a prepare_for_beam state input. Exactly the same as if a prepare_for_beam clock event was received.

- ***Fake End of Beam***
  Initiates an end of beam state input. Exactly the same as if an end_of_beam clock event was received.
- ***Pause the system***
  When this is received, the system goes into a paused state as soon as it enters a non-beam state. So in the main injector, it would go here after the end of the current cycle (or immediately if currently between cycles). In this state, the control card effectively ignores clock events. This input acts like a toggle button, so successive triggers take the system in and out of the paused state.
- ***Change Machine State***
  Change state to the specified state. The specified state must first be written to the vme_MachineState register.
- ***Toggle TCLK polling***
  Toggles whether or not the ez80 is processing TCLKs.
- ***Toggle MDAT polling***
  Toggles whether or not the ez80 is processing MDATs.
- ***Toggle Data Latch Interrupts***
  Toggles whether or not the ez80 is processing data latches.
- ***Toggle Abort Card polling***
  Toggles whether or not the ez80 is processing AC information.
- ***Enable/Disable CPU Time measurements***
  Toggles the CPU time measurments.
- ***Update custom TCLK events***
  Triggers an update to the custom TCLK ←→ State Input map used during debugging sessions.
- ***Toggle Oscilloscope Readings***
  Toggles whether or not TCLKs, MDATs, and other such slow things are recorded in a timestamped buffer which can be dumped to VME memory.
- ***Dump Oscilloscope Readings***
  Copy the oscilloscope buffer to VME memory, starting from the oldest data.
- ***Freeze Pedestals***
  Put the system into a pseudo state where just before make_meas is started, the settings page in use on the DC is flipped to a temporary one. The pedestals are then taken and written to that page. Once pedestals are done, the settings page is restored. This allows the same pedestals to be used forever, rather than updating every cycle.
- ***Thaw Pedestals***
  This undoes the previous trigger. So pedestals are then updated every cycle.

5 Output Control Lines

The control bus Reset State Machine signal can be used to quickly reset the system on the fly in the event that a card gets confused. A Reset is wired to port B pin 1 on the eZ80 which is setup

as an output pin. This pin must be pulled low to assert the reset. This is done at boot time but currently at no other time. It would probably be initiated at other times from the CP.

6 Remote Programming

To remotely change the CC control program, one must first obtain a file containing the new code, and then write it to the flash memory of the remote board. The simplest way to do this is to flash a test board (which has to be done anyway in order to test changes to the code) using the normal Zilog tool and then copy the contents of the flash memory for use in the remote board. This avoids having to decipher where to install the various interrupt vector tables, not to mention *what* to install.

The exact procedure for reprogramming the flash memory on the CC is as follows:

1) Make whatever changes are needed to the code
2) Build the application under the Zilog IDE
3) Flash the memory of the test board using the Zilog IDE
4) Reboot the board and start it running
5) Write a 1 to address NN800008 to trigger a read of the flash memory
6) Copy the contents of 128Kb of memory starting at NN820000, which contains the 'just read' contents of the flash memory, to a file someplace
7) Copy the contents of the file to address NN820000 of the remote board
8) Write A596 to address NN800008 which triggers a write to flash memory
9) Reboot the board

From the BLM front end, one can use the ez80FlashRead("filename") and ez80FlashWrite("filename") functions to do steps 5…8.

One thing to be aware of is that the existing code is limited to 128kb of space. When the board is flashed using the Zilog tool, it tells you how many blocks of flash memory will be erased. This is the size of the program. If the program ever needs more than the first 4 blocks (more than 128k), then the eZ80 program and the front end code will need to be modified to handle the larger code package.

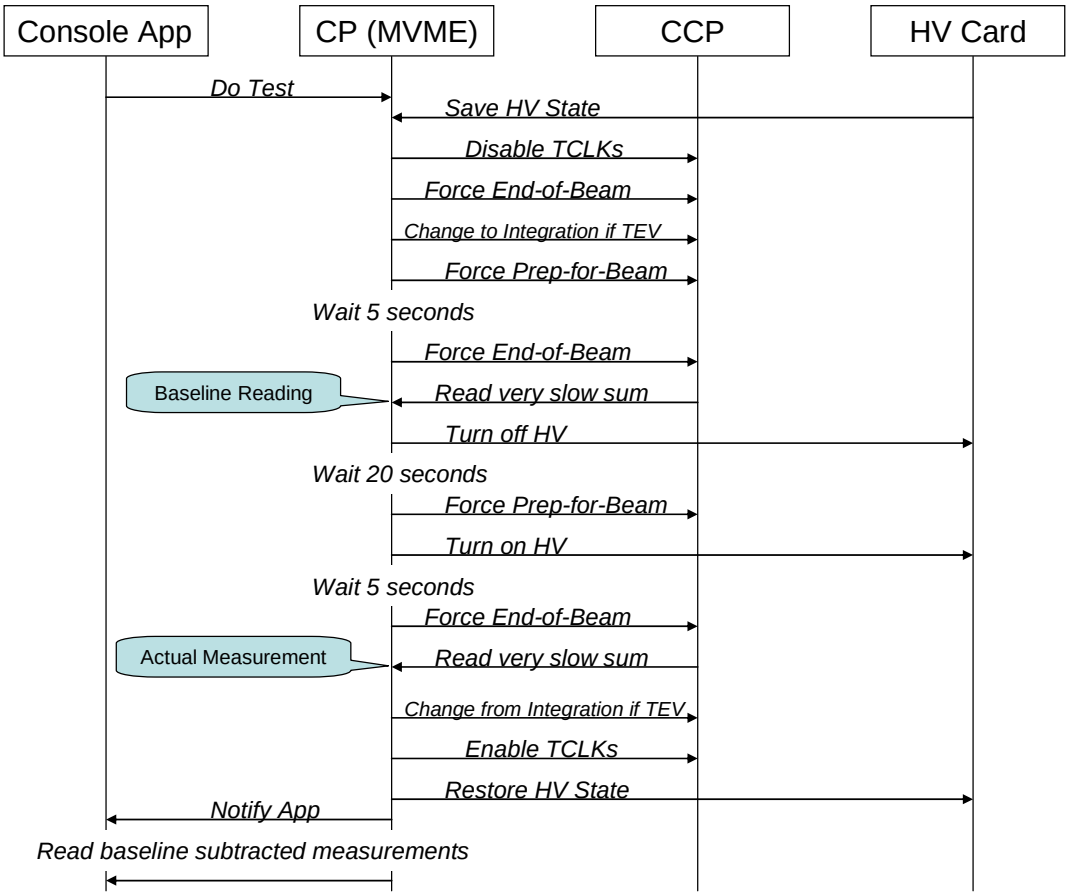
7 Miscellaneous Stuff

The ERROR line on the J2 backplane connector is wired up to port B pin 2 on the eZ80. It is periodically monitored by the CCP and if it is ever asserted (low), a bit is latched in the status word. Currently, the ERROR line is used for random purposes by at least the TC, so it should not be actively monitored by the CP.

The CCP also periodically copies the temperatures of the AC, TC, and DCs to VME memory for monitoring by the CP.

8 Test Mode

When testing the system hardware, it is advantageous to be able to pulse all the channels in a crate and see that they respond properly. One way of doing this with the BLM system, is to turn the high voltage off for a few minutes, and then turn it back on and record the signal coming from each channel. This is best accomplished by the following procedure on the CP.


Console App
CP (MVME)
CCP
HV Card
Do Test
Save HV State
Disable TCLKs
Force End-of-Beam
Change to Integration if TEV
Force Prep-for-Beam
Wait 5 seconds
Force End-of-Beam
Baseline Reading
Read very slow sum
Turn off HV
Wait 20 seconds
Force Prep-for-Beam
Turn on HV
Wait 5 seconds
Force End-of-Beam
Actual Measurement
Read very slow sum
Change from Integration if TEV
Enable TCLKs
Restore HV State
Notify App
Read baseline subtracted measurements

9 References

[1] **BEAMS-DOC-1410** Beam Loss Monitor Upgrade Users' Guide. Documents hardware and functionality of the system.

A Data Record

The fundamental unit stored in the various data buffers has the following format:

Byte Offset	Length	Description
0x00	1	Abort State
0x01	1	Measurement Divisor
0x02	2	Sum Divisor
0x04	1	Instantaneous Abort Status from AC
0x05	1	Channel Count
0x06	1	Data flag to indicate normal (0), last of cycle (1), new cycle (2), or waiting for stable data (3) frame
0x07	1	MDAT State
0x08	4	Time stamp: Microseconds since last TCLK 1 Hz Event
0x0C	4	Time stamp: seconds since T0 (*i.e.* 1 January 1970)
0x10 + n*4	240	Sliding Sum data for channel *n*

The data flag normally cycles through the following progression:

- at start of beam, wait for the digitizer delay, then write a 2 in the first frame collected (for each type of sum)
- write 3 in all the succeeding frames until after the input switch is closed for the specified amount of time
- write 0 in the rest of the frames of the cycle
- at end of beam, write a 1 in the last frame in the circular buffer

In the abort status word, bits 0-3 are the status of the abort from the Immediate, Fast, Slow, and Very Slow measurements, respectively. Bits 4-7 are not used. Loss data are stored as 32-bit long words with data in order from least significant to most significant byte.

Profile, Flash, and Display frames each contain 2 of these data frames. The first one is the user selectable Fast or Slow Sum frame and the second is the Very Slow Sum frame which in the Main Injector contains the integrated value.

B Abort Information Layout

Note that the maps below contain enough entries for 256 states to keep consistency with the abort threshold arrays. Presently the code only handles states 0-63 (zero cannot be changed from its default and is used internally when changing settings).

B.1 Channel Masks

For each species of abort, there are 7 bytes of channel masks. Byte 0 is unused. So, for instance, if there were 44 (0 to 43) channels in some crate, the channel assignment within the 8 bytes would be as follows (x means unused).

Bytes **Bits**	**7** **7… 0**	**6** **7… 0**	**5** **7… 0**	**4** **7… 0**	**3** **7… 0**	**2** **7… 0**	**1** **7… 0**	**0** **7… 0**
Channel	x…x	x…x 43…40	39…32	31…24	23…16	15…8	7…0	x…x

So to enable channel 17, one would write a 1 to byte 3, bit 1; and to enable channel 7, one would write a 1 to byte 1, bit 7.

B.2 MDAT → Abort State Map

The map to go from MDAT state to Abort state is a 256 byte array indexed by MDAT state. The following table illustrates how the information is stored in each word. There is room for 256 entries to cover all possible states, but only 64 are presently used.

VME address	Most Significant Byte	Least Significant Byte
0x098E0000	Abort state for MDAT state 1	Abort State for MDAT state 0
0x098E0002	Abort state for MDAT state 3	Abort State for MDAT state 2
0x098E0004	Abort state for MDAT state 5	Abort State for MDAT state 4
…	…	…

B.3 TEV F Sector

B.3.1 List of MI Start and End TCLK Events

This list is actually 2 lists of 256 bytes each: a list of starting TCLKs, and a list of ending TCLKs, each indexed by MI MDAT state. The format of each list is identical. The following table shows the format for either list. A starting TCLK value of 0x00 indicates there is no action to take for that MI MDAT state. There is room for 256 entries in each list to cover all possible states, but currently only the first 64 are used.

VME address		Most Significant Byte	Least Significant Byte
Starting TCLKs	Ending TCLKs		
0x098E0100	0x098E0200	TCLK for MDAT state 1	TCLK for MDAT state 0
0x098E0102	0x098E0202	TCLK for MDAT state 3	TCLK for MDAT state 2
0x098E0104	0x098E0204	TCLK for MDAT state 5	TCLK for MDAT state 4
…	…	…	…

B.3.2 Table of Actions vs. TEV Abort States / MI MDAT States

This is a 2-D byte array containing the action to take when a MI starting TCLK event occurs. It is indexed by both TEV abort state and MI MDAT state. A value of 0x00 indicates that the AC outputs should just be masked off. A value of 0xFF indicates that nothing should be done. Any

other value is interpreted as a TEV abort state to switch to between the starting and ending clock events. The default is 0x00, *i.e.* mask off the AC outputs. The indices are arranged with MI MDAT state incrementing the fastest. There is enough memory reserved for a 256 x 256 byte array, but currently, it is only 64 x 64. The following table illustrates the layout presently.

VME address	Most Significant Byte	Least Significant Byte
0x098F0000	Action for MI MDAT state 1 and TEV abort state 0	Action for MI MDAT state 0 and TEV abort state 0
0x098F0002	Action for MI MDAT state 3 and TEV abort state 0	Action for MI MDAT state 2 and TEV abort state 0
0x098F0004	Action for MI MDAT state 5 and TEV abort state 0	Action for MI MDAT state 4 and TEV abort state 0
…	…	…
0x098F0040	Action for MI MDAT state 1 and TEV abort state 1	Action for MI MDAT state 0 and TEV abort state 1
0x098F0042	Action for MI MDAT state 3 and TEV abort state 1	Action for MI MDAT state 2 and TEV abort state 1
0x098F0044	Action for MI MDAT state 5 and TEV abort state 1	Action for MI MDAT state 4 and TEV abort state 1
…	…	…

C VME Dual Port Memory Map

VME dual port memory addresses start at 0x09800000. The eZ80 addressing of the dual port memory begins at 0x800000 (24 bit addressing). The following addresses are specified as offsets from the base addresses.

Address	Size	Description
System Status		
000000	2	Status Word
000002	2	Reboot the CC (requires 0xA596 to be written here)
000004	1	Choice for whether fast sum (0) or derippled (1) data is used for FPD frames
000006	1	Bit pattern for selecting Flash/Profile/Display source Bit 0 = flash, bit 1 = profile, bit 2 = display; 0 = fast, 1 = slow
000008	1	Read flash memory
00000A	2	Load flash memory (requires 0xA596 to be written here)
00000C	2	Flash memory load status
00000E	1	Clear Abort Info Bit 0 = abort info bits in status word and snapshot Bit 1 = snapshot OR Bit 2 = channel OK list
000010	1	Force Pedestal Read
000012	1	Update Time Setting
000014	2	Time Setting LSW
000016	2	Time Setting MSW
000018	1	Update Settings
00001A	1	Update Abort Settings
00001C	1	Machine Type
00001E	1	MDAT State
00001F	1	Abort State
Flash, Profile and Snapshot Indexes		
000020	2	Flash Frame Counter
000022	2	Profile Frame Counter
000024	2	Fast Sum Circular Buffer Frame Index 1
000026	2	Fast Sum Circular Buffer Frame Index 2
000028	2	Slow Sum Circular Buffer Frame Index 1
00002A	2	Slow Sum Circular Buffer Frame Index 2
00002C	2	Very Slow Sum Circular Buffer Frame Index 1
00002E	2	Very Slow Sum Circular Buffer Frame Index 2
000030	1	Fast Sum Frame Index Pointer
000032	1	Slow Sum Frame Index Pointer
000034	1	Very Slow Sum Frame Index Pointer
000036	2	TC Raw data pointer
000038	2	DC Raw data pointer
00003A	2	AC Raw data pointer
00003C	2	DeRippled Sum Circular Buffer Frame Index 1

Address	Size	Description
00003E	2	DeRippled Sum Circular Buffer Frame Index 2
000040	1	DeRippled Sum Frame Index Pointer
Temperatures		
000050	2	TC Temperature Value
000052	2	AC Temperature Value
000054	30	DC Temperature Values for all DCs (each 2 bytes)
Abort Info		
000080	7	Channel OK List from Abort Card (1 means OK, 0 means problem)
Settings		
000100	1	Channel Count
000102	2	Make_Meas Divisor
000104	2	Fast Sum Length
000106	2	Slow Sum Length
000108	2	Very Slow Sum Length
00010A	2	Digitizer FPGA Control Register (high byte is number of make_meas to skip before performing pedestal measurements / 16)
00010C	unused	Digitizer Test DAC
00010E	1	Timing Card Control Bus CSR Register
000110	unused	Timing Card Control Settings (reserved)
000112	1	What generates an IRQ3 interrupt from the Abort Card?
000114	1	Abort Card enable (Bit 0 is *global enable*, Bit 4 is *require 2 consecutive aborts*)
000116	2	Pedestal Sum Length
000118	1	End of beam delay before asserting AIP (in fast latch units)
00011A	1	Delay after Flash clock event before collecting flash frame (in fast latch units)
00011C	1	Delay after Profile clock event before collecting profile frame (in fast latch units)
00011E	1	Delay after Display clock event before collecting display frame (in fast latch units)
000120	1	Turn off inputs while doing pedestals
000122	1	Optional extra delay after pedestals before re-enabling DC inputs
000124	1	Delay after re-enabling DC inputs before reading data
000126	1	Delay (slow sum units) after the end TCLK event before re-enabling normal F sector aborts
000128	1	Max DY for CIC derippling procedure to allow new pedestal waveform
00012A	1	CIC sum length
000200	2	Channel 0 Mode Select
000202	2	Channel 0 MADC Manual Setting
		⇩
0002EC	2	Channel 59 Mode Select
0002EE	2	Channel 59 MADC Manual Setting

Address	Size	Description
Pedestal Storage		
000300	256	32-bit pedestals stored in standard data record
Squelch Storage		
000400	120	16-bit squelch values
Abort Snapshot Storage		
000500	32	Abort snapshot record from last frame before abort
000520	32	Abort snapshot OR of all abort snapshots since this was last cleared
000540	2	Abort circular buffer pointer at time of abort
000542	2	Fast Sum index at time of abort
000544	2	Slow Sum index at time of abort
000546	2	Very Slow Sum index at time of abort
000548	2	DeRippled Sum index at time of abort
Integration Pedestal Storage		
000600	256	32-bit Integration pedestals
Debug Information		
010000	48	Program name and build date
010030	4	Test Sequence (0x44332211)
010034	4	TCLK count
010038	4	MDAT frame count
01003C	2	Last TCLK received
01003E	2	Last MDAT state received
010040	4	Data Interrupt count
010044	4	Fast Sum Latch count
010048	4	Slow Sum Latch count
01004C	4	Very Slow Sum Latch count
010050	4	Profile request count
010054	4	Flash request count
010058	4	Display request count
01005C	4	Abort Interrupt count
010060	4	Channel not OK interrupt count
010064	4	Channel abort interrupt count
010068	4	Crate abort interrupt count
01006C	4	Missed Fast Latch count
010070	4	Missed Slow Latch count
010074	4	Missed Very Slow Latch count
010078	2	Program State
01007A	2	Number of fast data latches to wait until pedestals are done
01007C	4	Last Fast Latch time microsecs
010080	4	Last Fast Latch time seconds
010084	4	Last Slow Latch time microsecs
010088	4	Last Slow Latch time seconds
01008C	4	Last Very Slow Latch time microsecs
010090	4	Last Very Slow Latch time seconds
010094	4	Stack Pointer (Updated every time through polling loop)

Address	Size	Description
010098	4	Stack Integrity (0xA4A3A2A1; also updated every loop)
01009C	16	Digitizer Map (map of found digitizers; 16th byte is 0)
0100AC	4	Timing Card clock error count
0100B0	4	Count of times TCLK queue has more than 1 entry
0100B4	4	Count of times MDAT queue has more than 1 entry
0100B8	4	Number of times a state ≥ 64 was received
0100BC	2	Last state machine input
0100BE	2	Force prepare for beam event (requires 0xA596 to be written here)
0100C0	2	Force end of beam event (requires 0xA596 to be written here)
0100C2	2	Pause the system (requires 0xA596 to be written here). This happens now if waiting for beam, or after next end of beam event.
0100C4	2	Change the system state (requires 0xA596 to be written here).
0100C6	2	Toggle TCLK polling (requires 0xA596 to be written here)
0100C8	2	Toggle MDAT polling (requires 0xA596 to be written here)
0100CA	2	Toggle Data interrupts (requires 0xA596 to be written here)
0100CC	2	Toggle Abort polling (requires 0xA596 to be written here)
0100CE	2	Enable CPU time measurements (just write a 0x0001 here; 0x0000 turns off CPU time measurements)
0100D0	2	Number of species of CPU time measurements
0100D2	2	DC Debug output. (0=off, 1=pedestal waveform is put in very slow sum buffer, 2=CIC output is put in very slow sum buffer)
0100D4	2	Update custom TCLKs (requires 0xA596 to be written here)
0100D6	2	Toggle the event history recording (requires 0xA596 to be written there)
0100D8	2	Copy history data to buffer (requires 0xA596 to be written there)
0100DA	2	Freeze the next set of pedestals (requires 0xA596 to be written there)
0100DC	2	Unfreeze the pedestal collection (requires 0xA596 to be written there)
010100	1024	TCLK counts for each TCLK type (256 types)
010500	1024	MDAT counts for each MDAT state (256 states)
010900	2048	CPU time measurement buffer (each measurement type occupies 32 words)
011200	512	State inputs for each custom TCLK
012000	8192	History data buffer
018000	2	Request a Control Bus memory dump
018002	2	Request a Control Bus write (requires 0xA596 to be written here)
018004	2	Control Bus address to dump / write (dumps 256 bytes / write 1 byte)
018100	256	Data copied from / written to Control Bus (only 1 byte in case of write)
019000	2	Request User Code to be run (requires 0xA596 to be written here)
01A000	8192	Buffer for User Code to be placed
Flash Memory Download		
020000	128K	Buffer for new flash memory contents
Flash Frames		
080000	512	Flash Frame 0
080200	512	Flash Frame 1

Address	Size	Description
		⇩
09FE00	512	Flash Frame 255
		Profile Frames
0A0000	512	Profile Frame 0
0A0200	512	Profile Frame 1
		⇩
0BFE00	512	Profile Frame 255
		Display Frame
0C0000	512	Display Frame
		Abort Stuff
0E0000	256	MDAT state to Abort state map
0E0100	256	F sector start TCLKs (indexed by MI MDAT state; a value of 0 means not implemented) This is a 1-D byte array, one byte per MI state.
0E0200	256	F sector end TCLKs (also indexed by MI MDAT state) Also a byte array
0F0000	65536	Array of F sector actions to take, indexed by TEV MDAT state and MI MDAT state. (An action of 0 means mask off the Abort card; a value of 0xFF means don't do anything, any other value means switch to that TEV abort state) These actions are implemented between the start and end TCLKs given by the lists above. This is a 2-D byte array. The fastest changing index is the MI state, i.e. unsigned char array[TEV states][MI states]
		Abort Machine States
		State 0
100000	1	State Number
100002	2	Channel Masks 0 1 for Immediate Abort
100004	2	Channel Masks 2 3 for Immediate Abort
100006	2	Channel Masks 4 5 for Immediate Abort
100008	2	Channel Masks 6 7 for Immediate Abort
10000A	2	Channel Masks 0 1 for Fast Abort
10000C	2	Channel Masks 2 3 for Fast Abort
10000E	2	Channel Masks 4 5 for Fast Abort
100010	2	Channel Masks 6 7 for Fast Abort
100012	2	Channel Masks 0 1 for Slow Abort
100014	2	Channel Masks 2 3 for Slow Abort
100016	2	Channel Masks 4 5 for Slow Abort
100018	2	Channel Masks 6 7 for Slow Abort
10001A	2	Channel Masks 0 1 for Very Slow Abort
10001C	2	Channel Masks 2 3 for Very Slow Abort
10001E	2	Channel Masks 4 5 for Very Slow Abort
100020	2	Channel Masks 6 7 for Very Slow Abort
100022	2	Abort Multiplicity for Immediate and Fast Abort

Address	Size	Description
100024	2	Abort Multiplicity for Slow and Very Slow Abort
100026	2	Crate Abort Mask
100028	8	Unused
100030	2	Channel 0 Immediate Threshold
100032	2	Channel 1 Immediate Threshold
		⇓
1000A6	2	Channel 59 Immediate Threshold
1000A8	8	Unused
1000B0	2	Channel 0 Fast Threshold LSW
1000B2	2	Channel 0 Fast Threshold MSW
1000B4	2	Channel 1 Fast Threshold LSW
1000B6	2	Channel 1 Fast Threshold MSW
		⇓
10019C	2	Channel 59 Fast Threshold LSW
10019E	2	Channel 59 Fast Threshold MSW
1001A0	16	Unused
1001B0	2	Channel 0 Slow Threshold LSW
1001B2	2	Channel 0 Slow Threshold MSW
1001B4	2	Channel 1 Slow Threshold LSW
1001B6	2	Channel 1 Slow Threshold MSW
		⇓
10029C	2	Channel 59 Slow Threshold LSW
10029E	2	Channel 59 Slow Threshold MSW
1002A0	16	Unused
1002B0	2	Channel 0 Very Slow Threshold LSW
1002B2	2	Channel 0 Very Slow Threshold MSW
1002B4	2	Channel 1 Very Slow Threshold LSW
1002B6	2	Channel 1 Very Slow Threshold MSW
		⇓
10039C	2	Channel 59 Very Slow Threshold LSW
10039E	2	Channel 59 Very Slow Threshold MSW
1003A0	80	Unused
		States 1 – 255; each with same offsets as State 0
100400	1024	State 1 Settings
100800	1024	State 2 Settings
		⇓
13F800	1024	State 255 Settings

Address	Size	Description
In Use Abort Settings for States 0-255		
140000	256 K	Copy of settings that are in use. These are the settings that are copied to AC and DCs.
Circular Buffers		
180000	512 K	DeRippled Sum Buffer 2K frames deep
200000	4 Meg	Fast Sum Buffer 16K frames deep
600000	1 Meg	Slow Sum Buffer 4K frames deep
700000	1 Meg	Very Slow Sum Buffer 4K frames deep