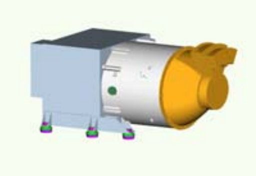
9:15 AM   3/1/2004

STEREO / PLASTIC



Logic Reference Document

University of New Hampshire

Revision 1.13

STEREO / PLASTIC Logic Reference Document

Prepared By:

Steven P. Longworth, UNH Date
Electrical System Engineer

Reviewed By:

Eberhard Mobius, UNH Date
Co-I

Lynn Kistler, UNH Date
Co-I

John Gaidos, UNH Date
Electrical Engineer

Approved By:

Toni Galvin, UNH Date
PI

STEREO / PLASTIC Logic Reference Document

Revision History

No.	Description	Date	Author
1.00	First release of Logic Reference Document	DEC 03	Steven P. Longworth
1.10	Changes to Instrument Controller Command & TM Structure	FEB 04	Steven P. Longworth
1.13	Minor corrections	FEB 04	Steven P. Longworth

STEREO / PLASTIC Logic Reference Document

Table of Contents

STEREO / PLASTIC Logic Reference Document

1 Introduction

The STEREO / PLASTIC instrument is designed to investigate the spectra and composition of solar wind and hemispheric ions. The primary (Level 1) STEREO Science Objectives are to:

1. Understand the causes and mechanisms of CME initiation.
2. Characterize the propagation of CMEs through the heliosphere.
3. Discover the mechanisms and sites of energetic particle acceleration in the low corona and the interplanetary medium.
4. Improve determination of the structure of the ambient solar wind.

To meet these objectives, there are four investigations on STEREO: SECCHI, IMPACT, PLASTIC and SWAVES.

The Plasma and Supra-Thermal Ion Composition (PLASTIC) Investigation on STEREO measures solar wind and suprathermal ions in the energy-per-charge range of ~0.3 (0.2) up to ~80 (100) keV/e. PLASTIC is the primary sensor on STEREO for studying coronal ⇔ solar wind and solar wind ⇔ heliospheric processes.

PLASTIC incorporates three science investigations:

The PLASTIC *Solar Wind Sector* (SWS) measures the distribution functions of solar wind protons (H^+) and alphas (He^{+2}), providing proton density (n) , velocity (V_{sw}), kinetic temperature (T_k) and its anisotropy ($T_\parallel$, $T_\perp$), and alpha to proton (He^{+2} / H^+) ratios with a time resolution up to about one minute (64 seconds).

The SWS measures the elemental composition, charge state distribution, kinetic temperature, and speed of the more abundant solar wind heavy ions (e.g., C, O, Mg, Si, and Fe). Typical time resolution for selected ions will be ~ 5 x 64 = 320 seconds (time resolution depends on telemetry allocation). The SWS will measure the distribution functions of high fluence suprathermal ions.

The PLASTIC *Wide-Angle Partition* (WAP) measures distribution functions of suprathermal ions, including interplanetary shock-accelerated (IPS) particles, recurrent particle events associated with Co-rotating Interaction Regions (CIRs), and heliospheric pickup ions. Typical time resolution for selected ions will be ~ 5 x 64 = 320 seconds (time resolution depends on telemetry allocation).

1.1 Instrument Specifications

1. Ability to measure solar wind protons and alphas over a broad range of speeds and intensities
2. Ability to handle high fluxes and narrow angular distributions of solar wind protons and solar wind alphas
3. Ability to make solar wind proton angular direction measurements on a 3-axes stabilized spacecraft
4. Ability to observe solar wind heavy ions with sufficient statistics
5. Sufficient mass and mass/charge resolution to identify H, He, C, O, Mg, Si, Fe and other key ion species in the solar wind (for some ions, this will involve fitting routines)
6. Ability to observe suprathermal ions with sufficient statistics
7. Wide angular and extensive energy coverage to determine pickup ion and suprathermal distribution functions for key species
8. Sufficient mass/charge resolution to identify suprathermal H+, He+2, and M/Q>2 categories

1.2 Scope

The intent of this document is to describe the digital logic system of the STEREO / PLASTIC instrument.

This document defines memory maps, tables, functions and registers as well as data structure and format.

1.3 Reference Documents

Interface Control Document (ICD) for the PLASTIC Investigation, 7381-9014, Applied Physics Laboratory (APL)

2 Instrument Controller Command and Telemetry Structure

PLASTIC commands from the IDPU are 24 bit words, which consist of an 8-bit command header and a 16-bit data field. Command signaling is described in detail in the STEREO IMPACT Intra Instrument Interface document. Commands are 24 bits long, preceded by a start bit, and followed by a parity bit and a stop bit. The 24 bits are sent MSB first. The parity is odd and includes the 24 command bits but not the start bit. A command with all 24 bits zero would have the parity bit set. Commands can start on any rising edge of the serial clock, and any number of idle bit periods can occur between commands. The data is transferred Most Significant Bit (MSB) first. The parity bit shall be set such that the sum of the number of set bits in the 24 command data bits plus the parity bit is odd. If the PLASTIC instrument receives a command with a parity error, or framing error (no stop bit), it will not attempt to execute the command. A count of received commands, executed commands and commands received with errors (not executed) will be kept. The detectable command errors are frame errors, parity errors and unknown / illegal command errors. Received and executed commands are stored in 16-bit counters. Each of the three error type counters is 4-bits. In addition to these three error counters, there is a fourth error counter called Command Errors, which is incremented by the logical ‘or’ of the three error types. Command counters can be cleared via a control register bit. The PLASTIC instrument does not expect an automatic command retry.

MSB LSB	MSB LSB
8 bit command	16 bit data field

The command word is broken up into two sections. The first nibble of the command indicates the PLASTIC module address. The second nibble of the command contains the actual command to be executed by the module. This allows for 14 different command module destinations . Some modules may have more than one module address. The 4-bit command field allows for 16 commands, per module address.

2.1 Module Definitions

0101	Unused
0110	Unused
0111	Unused
1000	Unused
1001	Unused
1010	Unused
1011	Unused
1100	Unused
1101	Unused
1110	Unused
1111	IDPU Commands

0000	Not Used
0001	Instrument Cont. Command
0010	Instrument Cont. Memory Commands
0011	Logic Board Immediate Commands
0100	Logic Board Memory Commands

2.2 Device Codes

Device Code	Module	Description
00000001	Classifier EEPROM A1	First 64K block of classifier EEPROM A
00000010	Classifier EEPROM A2	Last 64K block of classifier EEPROM A
00000011	Classifier EEPROM B1	First 64K block of classifier EEPROM B
00000100	Classifier EEPROM B2	Last 64K block of classifier EEPROM B
00000101	Classifier EEPROM C1	First 64K block of classifier EEPROM C
00000110	Classifier EEPROM C2	Last 64K block of classifier EEPROM C
00000111	Classifier EEPROM D1	First 64K block of classifier EEPROM D
00001000	Classifier EEPROM D2	Last 64K block of classifier EEPROM D
00001001	Classifier SRAM A	32K block of classifier SRAM A
00001010	Classifier SRAM B	32K block of classifier SRAM B
00001011	Logic Board EEPROM A1	First 64K block of EEPROM A
00001100	Logic Board EEPROM A2	Last 64K block of EEPROM A
00001101	Logic Board SRAM	32K block of SRAM

2.3 Module Commands

2.3.1 Instrument Controller Commands

MOD	CMD	Name	Data Field (16 bits)
0001	0001	Control Reg Write	Control Register Contents 0x00_ _ 8-bits of "0" followed by 8-bits data
0001	0010	Control Reg Read	0x0000
0001	0011	Status Reg Read	0x0000
0001	0100	Read Commands Received Counter	0x0000
0001	0101	Read commands Executed Counter	0x0000
0001	0110	Read Error Counters	0x0000
1111	0000	Sample Clk message From IDPU	See Impact Intra-Instrument Serial Interface Document
1111	1111	Reset cmd from IDPU	TBD_Impact

Note: The three command counter commands, when received, do not increment either the Commands Received or Command Executed counters. Also, the IDPU Time commands do not increment either the Commands Received or Command Executed counters. Lastly, an illegal or undefined MOD=1111 command will not increment the error counters.

2.3.2 Instrument Controller Memory Commands

MOD	CMD	Name	Data Field (16 bits)
0010	0001	Block Read	8 bit Device Code + 8 bit block ID
0010	0010	Block Write	Block Data, 2 Bytes per word
0010	0011	Set Device Code and block ID	8 bit Device Code + 8 bit block ID
0010	0100	Write Byte	Byte to write & Lower 8 Address Bits
0010	0101	Read Word	00000000 & Lower 8 Address Bits of Lower Byte
0010	1000	TC Block Write	Command Sent via serial channel. Block Data, 2 Bytes per word via Test Connector.
0010	1001	TC Block Read	Command Sent via serial channel. 8 bit Device Code + 8 bit block ID Data out via Test Connector.

2.3.3 Logic Board Immediate Command

MOD	CMD	Name	Data Field (16 bits)
0011	0001	Immediate Command	8 Bit Register Address & 8 Bits Data

2.3.4 Logic Board Memory Commands

MOD	CMD	Name	Data Field (16 bits)
0100	0001	Block Read	8 bit Device Code + 8 bit block ID
0100	0010	Block Write	Block Data, 2 Bytes per word
0100	0011	Set Device Code and block ID	8 bit Device Code + 8 bit block ID
0100	0100	Write Byte	Byte to write & Lower 8 Address Bits
0100	0101	Read Word	00000000 & Lower 8 Address Bits of Lower Byte
0100	1000	TC Block Write (Block Write cmd sent to Logic Board)	Command Sent via serial channel. Block Data, 2 Bytes per word via Test Connector.
0100	1001	TC Block Read	Command Sent via serial channel. 8 bit Device Code + 8 bit block ID Data out via Test Connector.

Note: a “block” is defined as 256-bytes. An EEPROM block write shall be performed using (2) 128-byte page writes. 10ms shall be the “wait time” between the end of the first page write and the start of the second page write. The msb of the EEPROM Device Code shall =0 for the lower page write (bytes 0 to 127) and =1 for the upper page write (bytes128 to 255).

Suggested command sequence to write one block of EEPROM:

1. Send “Set Device Code & Block ID” with msb of Device Code =0
2. Send 64 “Block Write” commands to write bytes 0 to 127 with <30us between commands
3. Wait 10ms
4. Send “Set Device Code & Block ID” with msb of Device Code =1
5. Send 64 “Block Write” commands to write bytes 128 to 255 with <30us between commands
6. Wait 10ms

2.4 Telemetry Format

The format of the telemetry stream consists of 2 to 1024 16-bit words. Each telemetry packet will begin with a 16-bit telemetry type header. For all messages the first word, called the MESSAGE_ID shall indicate the type of message and its length, and the 6 MSBs shall be a message type code. The message length shall be coded as the number of words in the message, including the MESSAGE_ID word, minus 2. The shortest possible message of two words shall have a length code zero, while the longest possible message will consist of the MESSAGE_ID word plus 1024 data words, with a length code of 1023.

16 bit type/length header	Data 0	Data 1…	To Data 1024

The telemetry header is formatted as follows:

MESSAGE_ID 6 Bits	Word Count 10 Bits

2.5 Telemetry Type Definitions

MESSAGE ID	Word Count	Name	Description
000001	0000000000	IC CMD I/F Error Counters	Header + 1 Word = 4-bit unknown cmd + 4-bit frame error + 4-bit parity error + 4-bit command error
000010	0000000000	IC Control Reg Read	Header + 1 Word = Control Reg Contents
000011	0000000000	IC Status Reg Read	Header + 1 Word = Status Reg Contents
000100	0010000001	Classifier Readout	Header + Device Code + Block ID + 8 Bit ESA Step + 0x00 + 128 Words of Classifier Memory
000101	0000000001	Memory Read Word	Header + Device Code + Block ID + Byte 1+ Byte 0
000110	0010000000	Memory Read Block	Header + Device Code + Block ID + Byte 1 + Byte 0 + Byte 255 + Byte 254
000111	0010000000	Logic Board Housekeeping	Header + Device Code + Block ID + 128 Words of HK Memory
001000	0000100000	Logic Board Monitor Rate Readout	Header + (8 Bit ESA Step) + (000 + 5 Bit Deflection Step) + 32 Monitor Rate Words
001001	0000000000	Instrument Controller I/F Commands Received Counter	Header + 1 Word = Counter Contents
001010	0000000000	Instrument Controller I/F Commands Executed Counter	Header + 1 Word = Counter Contents

2.6 Instrument Controller Status & Control Registers

Instrument Controller STATUS register

15	14	13	12	11	10	9	8
1	0	sysid1	sysid0	unknown cmd error	frame error	parity error	command error

7	6	5	4	3	2	1	0
1	0	sysid1	sysid0	unknown cmd error	frame error	parity error	command error

NOTES:
the same 8-bits are intentionally repeated in the upper & lower bytes
"command error" is the logical 'or' of "unknown cmd error", "frame error" and "parity error"
error flags in the Status Register are cleared when the Status Register is read

Instrument Controller Command register **default = 00000000**
All are active high

7	6	5	4	3	2	1	0
clr cmd counters	block mem r/w source/dest 0=idpu 1=test connector	pha_mode1	pha_mode0	tc_mode2	tc_mode1	tc_mode0	cmd_reset

pha_mode	00 = Logic Board to Classifier (Test Connector not used)
	01 = Logic Board to Classifier – Logic board to Test Connector
	10 = Test Connector to Classifier
	11 = PHA Off
tc_mode	000 = Test Connector disabled
	001 = PHA data from Test Connector to instrument
	010 = PHA data from instrument to Test Connector
	011 = memory block write from Test Connector to instrument
	100 = memory block read from instrument to Test Connector
	101 = (not used)
	110 = (not used)
	111 = (not used)

3 Logic Board Registers

3.1 Control ACTEL 1 Registers

00h **RESET_CTL** **default =** XXXXX000

7	6	5	4	3	2	1	0
x	x	x	x	x	reset_hv	reset3	reset2

01h **MODE_CTL** **default =** XXXXX000

7	6	5	4	3	2	1	0
x	x	x	x	x	d2	d1	d0

mode	**function**
000	halt mode
001	read DAC registers from RAM and write to DAC board one time
010	read ACTEL 2 Register Sequence from EEPROM and write to ACTEL 2 one time
011	normal mode (mode 001 & 010 on retrace interval with DAC tables active)
100	run RETRACE routine one time
101	read DAC registers from DAC board and store in RAM one time
110	halt mode (future use)
111	halt mode (future use)

02h **LOGIC_CTL_A** **default =** 00000011

watchdog tick (wd_tick) is checked when wd_en=1 *strobe register bits OR'd with mode generated strobes*

7	6	5	4	3	2	1	0
sweep_en	esa_strobe	defl_strobe	wd_en	adc_reset	mux_en	adc1_pd	adc0_pd

03h **LOGIC_CTL_B** **default =** 00XX0100

auto_lu_clr will enable ACTEL 1 to automatically clear an ADC latchup
retrace_rst_dis contols rst2_n, rst3_n, adc_reset & DAC board adc reset control register during retrace interval
(1 = don't do resets 0 = do resets)
retrace_ee_dis controls eeprom register sequences to ACTEL 2 (1 = don't run 0 = run sequences)

7	6	5	4	3	2	1	0
retrace_rst_dis	retrace_ee_dis	x	x	adc_lu_reset_reg	adc_ref_pd_reg	auto_lu_clr	clr_adc_lu_flag

04h **TAC_PWR_CTL** **default =** XXXXXX11

7	6	5	4	3	2	1	0
x	x	x	x	x	x	tac2_pwr_n	tac0_pwr_n

05h **REG_SEQ** **default =** XXXX0000

Active Event Selection (ACTEL 2) Register Sequence

7	6	5	4	3	2	1	0
x	x	x	x	d3	d2	d1	d0

06h EVENT_CTL default = XXXXXXX0

when s_ch_en=1 s-channel is enabled, main channel is disabled (and vice-versa)

7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	s_ch_en

07h RLIM_CH default = 0XX00000

Rate Limit Checking Channel

when rates_chk=1 rate limit checking is enabled (auto set s_ch_en=1 with rates – then =0 afterESA step)

7	6	5	4	3	2	1	0
rate_chk	x	x	d4	d3	d2	d1	d0

08h RLIM_HI default = 00000000

Rate Limit Checking Value (h-byte)

7	6	5	4	3	2	1	0
d15	d14	d13	d12	d11	d10	d9	d8

09h RLIM_LO default = 00000000

Rate Limit Checking Value (l-byte)

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

3Fh [no-operation]

7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x

3.2 Event ACTEL 2 Registers

41h POS_CTRL default = X0000000

pos_inhib, pos_d, pos_c, pos_b and pos_a must be ‘0’ when tac2 is powered down

7	6	5	4	3	2	1	0
x	pos_inhib	pos_pr2	pos_pr1	pos_d	pos_c	pos_b	pos_a

42h POS_DISABLE0 default = XXXXX000

7	6	5	4	3	2	1	0
x	x	x	x	x	pos1_1	pos1_0	ra_trig

43h POS_DISABLE2 default = 00000000

7	6	5	4	3	2	1	0
pos3_3	pos3_2	pos3_1	pos3_0	pos2_3	pos2_2	pos2_1	pos2_0

44h TAC0_CTRL default = XXXXXX00

MSB has highest priority and LSB lowest - set automatically by ACTEL 1 when mode_ctl = 3

7	6	5	4	3	2	1	0
x	x	x	x	x	x	force_rst	write_tac

45h TAC0_DAC default = 00000000

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

46h TAC2_CTRL default = XXXXXX00

MSB has highest priority and LSB lowest - set automatically by ACTEL 1 when mode_ctl = 3

7	6	5	4	3	2	1	0
x	x	x	x	x	x	force_rst	write_tac

47h TAC2_DAC default = 00000000

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

48h TAC0_UNDER_HI (h-byte) default = XXXXXX00

7	6	5	4	3	2	1	0
x	x	x	x	x	x	d9	d8

49h TAC0_UNDER_LO (l-byte) default = 00000000

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

4Ah TAC0_OVER_HI (h-byte) default = XXXXXX11

7	6	5	4	3	2	1	0
x	x	x	x	x	x	d9	d8

4Bh TAC0_OVER_LO (l-byte) default = 11111111

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

4Ch TAC2_UNDER_HI (h-byte) default = XXXXXX00

7	6	5	4	3	2	1	0
x	x	x	x	x	x	d9	d8

4Dh TAC2_UNDER_LO (l-byte) default = 00000000

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

4Eh TAC2_OVER_HI (h-byte) default = XXXXXX11

7	6	5	4	3	2	1	0
x	x	x	x	x	x	d9	d8

4Fh TAC2_OVER_LO (l-byte) default = 11111111

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

50h SSD_CTRL default = XXX0000X

MSB has highest priority and LSB lowest – ssd disable always enforced & must be '0' to read energy
set automatically by ACTEL 1 when mode_ctl = 3

7	6	5	4	3	2	1	0
x	x	x	force_reset	force_sync	send_cmd	send_hkc	x

51h SSD_CMD_H (h-byte) default = 00000000

7	6	5	4	3	2	1	0
d15	d14	d13	d12	d11	d10	d9	d8

52h SSD_CMD_L (l-byte) default = 00000000

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

53h SSD_DIS default = XXXXXXX0

7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	ssd_disable

54h SSD_UNDER_HI (h-byte) default = 00000000

7	6	5	4	3	2	1	0
x	x	x	x	x	x	d9	d8

55h SSD_UNDER_LO (l-byte) default = 00000000

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

56h SSD_OVER_HI (h-byte) default = 11111111

7	6	5	4	3	2	1	0
x	x	x	x	x	x	d9	d8

57h SSD_OVER_LO (l-byte) default = 11111111

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

60h SEL_CTRL default = XXXXXX11

sel2 controls quadrants 2 & 3 - sel0 controls quadrants 0 & 1
set automatically by ACTEL 1 when mode_ctl = 3

7	6	5	4	3	2	1	0
x	x	x	x	x	x	sel2_disable	sel0_disable

61h TRIG_MODE default = 0XXX0000

bits 7..4 for quadrants 2 & 3 - bits3..0 for quadrants 0 & 1

7	6	5	4	3	2	1	0
tmode2	x	x	x	tmode0(3)	tmode0(2)	tmode0(1)	tmode0(0)

62h ESA_STEP (for classification) default = 00000000

set automatically by ACTEL 1 when mode_ctl = 3

7	6	5	4	3	2	1	0
s_ch_en	d6	d5	d4	d3	d2	d1	d0

63h DEFL_STEP (for classification) default = XXX00000

set automatically by ACTEL 1 when mode_ctl = 3

7	6	5	4	3	2	1	0
x	x	x	d4	d3	d2	d1	d0

70h STIM_ENABLE default = XXX00000

tac0 must be ‘0’ when tac0 is powered down
pos & tac2 must be ‘0’ when tac2 is powered down

7	6	5	4	3	2	1	0
x	x	x	ra	ssd	tac0	tac2	pos

71h STIM_FREQ_HI (h-byte) default = 11111111

stim_freq = [25.6x10e+6 / (16 x 2 x actual stim frequency)] - 1
actual stim frequency = 25.6x10e+6 / [2 x (16 x stim_freq + 16)]
if stim_freq = 0 then actual stim frequency = 800kHz *stim_freq min. for ssd = 1 (400kHz)*
if stim_freq = 255 then actual stim frequency = 3.125kHz
if stim_freq=65535 then actual stim frequency = 12.2Hz (default)

7	6	5	4	3	2	1	0
d15	d14	d13	d12	d11	d10	d9	d8

72h STIM_FREQ_LO (l-byte) default = 11111111

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

75h SEL0_WINDOW default = 00000000

*window length [sec] = (register value+1) * 1/25.6x10e+6 plus uncertainty of 0.00 to 1/25.6x10e+6 [sec]*
1/25.6x10e+6 = ~39nsec

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

76h SEL2_WINDOW default = 00000000

7	6	5	4	3	2	1	0
d7	d6	d5	d4	d3	d2	d1	d0

80h POS_RA default = 00000000

only msb read therefore 80h – FFh are used
cannot be written via an immediate command – set automatically by ACTEL 1 when mode_ctl = 3

7	6	5	4	3	2	1	0
sat_b	sat_a	d5	d4	d3	d2	d1	d0

4 Logic Board Memory

4.1 RAM (32k x 8) – DAC Board Registers & HK

`0000-7fff` **`32768 bytes`**

To DAC Board Control Registers (lower 8-bits = DAC Board Address)

0000h PAC_MCP_SSD_CTL_MCP_LIMIT

7	6	5	4	3	2	1	0
limit(7)	limit(6)	limit(5)	limit(4)	limit(3)	pac_en	mcp_en	ssd_en

0001h PAC_LIMIT

7	6	5	4	3	2	1	0
limit(7)	limit(6)	limit(5)	limit(4)	limit(3)	limit(2)	limit(1)	limit(0)

0002h ESA_CTL

7	6	5	4	3	2	1	0
limit(7)	limit(6)	limit(5)	limit(4)	limit(3)	opto_en_neg	opto_en_pos	enable

0003h S_CH_CTL

7	6	5	4	3	2	1	0
limit(7)	limit(6)	limit(5)	limit(4)	limit(3)	limit(2)	opto_en	enable

0004h DFL_1_CTL

7	6	5	4	3	2	1	0
limit(7)	limit(6)	limit(5)	limit(4)	limit(3)	limit(2)	opto_en	enable

0005h DFL_2_CTL

7	6	5	4	3	2	1	0
limit(7)	limit(6)	limit(5)	limit(4)	limit(3)	limit(2)	opto_en	enable

0006h CONTROL

defl_sel = 0 for defl-1 supply *defl_sel = 1 for defl-2 supply*
auto_lu_recove = 0 for auto *auto_lu_recove = 1 for manual* man_lu_rc = manual lu recovery complete

7	6	5	4	3	2	1	0
dfine_static_n	man_lu_rc	lu_reset	reset_adc	pwr_dn_adc_ref	clr_lu_ctr	auto_lu_recove	defl_sel

0007h STATUS

7	6	5	4	3	2	1	0
x	x	mcp_cm_f_n	pac_cm_f_n	lu_ctr_ovfl	adc_hung	x	adc_lu_det

0008h ADC_LU_CTR

7	6	5	4	3	2	1	0
count(7)	count(6)	count(5)	count(4)	count(3)	count(2)	count(1)	count(0)

`[ not used ]` `0009-000f`

DAC Board HV Registers (lower 8-bits = DAC Board Address)

N_PAC_COARSE	0010
N_PAC_FINE	0011
N_PAC_CM_FS_COARSE	0012
N_PAC_CM_FS_FINE	0013
N_MCP_COARSE	0014
N_MCP_FINE	0015
N_MCP_CM_FS_COARSE	0016
N_MCP_CM_FS_FINE	0017
N_SSD_COARSE	0018
N_SSD_FINE	0019
N_ESA_COARSE	001a
N_ESA_FINE	001b
N_S_CH_COARSE	001c
N_S_CH_FINE	001d
N_S_CH_TAP_COARSE	001e
N_S_CH_TAP_FINE	001f
N_DFL_1_COARSE	0020
N_DFL_1_FINE	0021
N_DFL_1_DFINE	0022
N_DFL_2_COARSE	0023
N_DFL_2_FINE	0024
N_DFL_2_DFINE	0025
C_PAC_COARSE	0030
C_PAC_FINE	0031
C_PAC_CM_FS_COARSE	0032
C_PAC_CM_FS_FINE	0033
C_MCP_COARSE	0034
C_MCP_FINE	0035
C_MCP_CM_FS_COARSE	0036
C_MCP_CM_FS_FINE	0037
C_SSD_COARSE	0038
C_SSD_FINE	0039
C_ESA_COARSE	003a
C_ESA_FINE	003b
C_S_CH_COARSE	003c
C_S_CH_FINE	003d
C_S_CH_TAP_COARSE	003e
C_S_CH_TAP_FINE	003f
C_DFL_1_COARSE	0040
C_DFL_1_FINE	0041
C_DFL_1_DFINE	0042
C_DFL_2_COARSE	0043
C_DFL_2_FINE	0044
C_DFL_2_DFINE	0045
[future use]	0046-004f
[not used]	**0050-00ff**

DAC Board HK		*HK ADDR /*	*DAC ADDR*
LRNM_SPARE		0100-0101	50
LVC_-12V		0102-0103	51
LVC_-5V		0104-0105	52
ESA_VM_POS		0106-0107	53
DFL_1_VM		0108-0109	54
DFL_1_TAP		010a-010b	55
DFL_2_VM		010c-010d	56
DFL_2_TAP		010e-010f	57
LVC_+2.5V_B		0110-0111	58
PAC_CM_DC		0112-0113	59
LVC_+12V		0114-0115	5a
LRPM_SPARE		0116-0117	5b
LVC_+2.5V_A		0118-0119	5c
LVC_+5V		011a-011b	5d
ADC_AVDD		011c-011d	5e
ADC_DVDD		011e-011f	5f
PAC_VM		0120-0121	60
CAL_VREF		0122-0123	61
PAC_CM_AC	(not used)	0124-0125	62
MCP_VM		0126-0127	63
MCP_CM_DC		0128-0129	64
MCP_CM_AC	(not used)	012a-012b	65
SSD_CM_DC		012c-012d	66
S_CH_VM		012e-012f	67
S_CH_VM_TAP		0130-0131	68
S_CH_CM_DC		0132-0133	69
ESA_VM_NEG		0134-0135	6a
ESA_CM_DC		0136-0137	6b
DFL_1_CM_DC		0138-0139	6c
DFL_2_CM_DC		013a-013b	6d
LVC_+2.5V_B_IMON		013c-013d	6e
LVC_+12V_IMON		013e-013f	6f
LVC_-12V_IMON		0140-0141	70
LVC_+5V_IMON		0142-0143	71
LVC_-5V_IMON		0144-0145	72
LVC_+2.5V_A_IMON		0146-0147	73
SSD_VM		0148-0149	74

Digital HK		*HK ADDR*	*(not used)*
PLUG_ID		014a	75
ADC_LU_FLAG		014b	--
STATUS	(DAC Board)	014c	76
ADC_LU_CTR	(DAC Board)	014d	--
[not used]		014e-014f	77

LOGIC Board HK			*HK ADDR*	*MUX ADDR*
ADC0_AGND	ADC1_AGND	0150-0151	78	0
RA_E0	RA_E1	0152-0153	79	1
TAC0_TSP	TAC2_TSP	0154-0155	7a	2
SB0_TSP	SB1_TSP	0156-0157	7b	3
TOF_HV0_TSP	TOF_HV1_TSP	0158-0159	7c	4
S_C_0_TSP	S_C_1_TSP	015a-015b	7d	5
LVC0_TSP	LVC1_TSP	015c-015d	7e	6
ADC0_VREF	ADC1_VREF	015e-015f	7f	7

SSD Board HK		*HK ADDR (not used)*	*REG SEQ*
SSD_STATUS	0160-0161	80	8
SSD_V_POS_ANALOG	0162-0163	81	9
SSD_V_NEG_ANALOG	0164-0165	82	a
SSD_HV_BIAS	0166-0167	83	b
SSD_TC0	0168-0169	84	c
SSD_TC1	016a-016b	85	d
SSD_TC2	016c-016d	86	e
SSD_V_POS_DIG	016e-016f	87	f

[future hk] 0170-01ff

From DAC Board Control Registers (00-4f) 0200-024f

Logic Board Register Write Log (00-7f) 0300-037f
(all registers except mode_ctl and event_ctl)

[not used] 0400-7fff

4.2 EEPROM (128k x 8) – Sequences & Tables

00000-1ffff 131072 bytes

4.2.1 ACTEL 2 Register Sequences

ACTEL 2 Register Sequences Only (ACTEL 1 Registers cannot be accessed)

address stored in odd EEPROM address locations
data stored in even EEPROM address locations

Sequence 0	00000-000ff	retrace interval reset sequence
Sequence 1	00100-001ff	[future use]
Sequence 2	00200-002ff	[future use]
Sequence 3	00300-003ff	[future use]
Sequence 4	00400-004ff	[future use]
Sequence 5	00500-005ff	[future use]
Sequence 6	00600-006ff	[future use]
Sequence 7	00700-007ff	[future use]

4.2.2 SSD Register Sequences

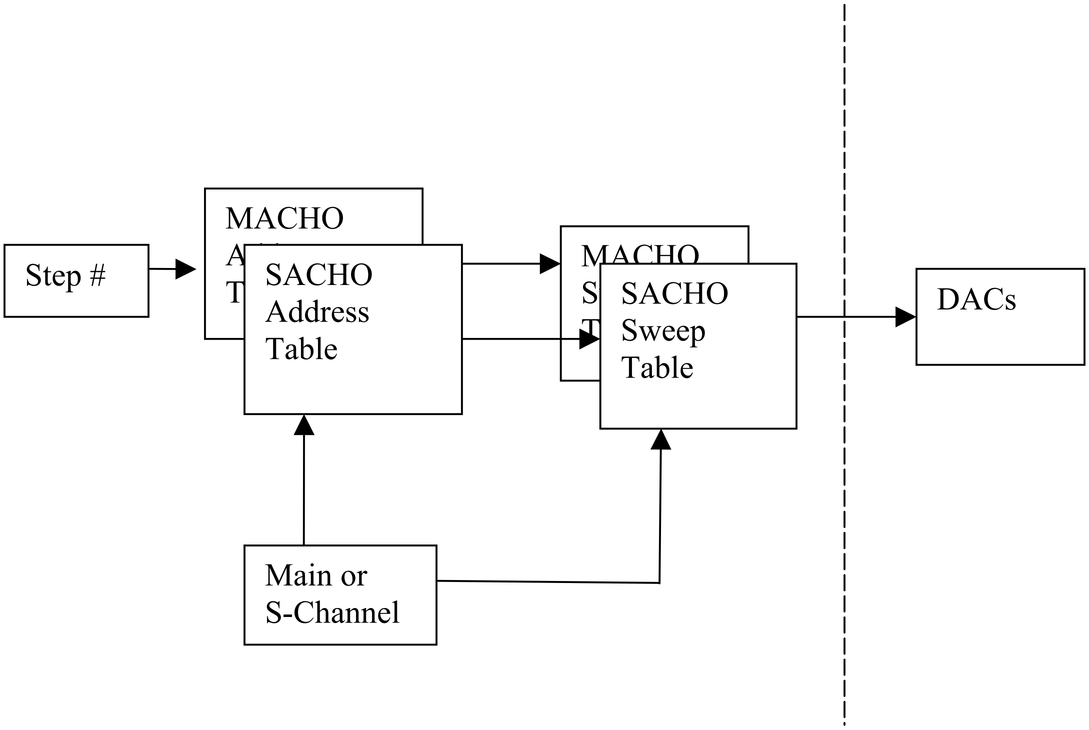
HK0	00800-008ff
HK1	00900-009ff
HK2	00a00-00aff
HK3	00b00-00bff
HK4	00c00-00cff
HK5	00d00-00dff
HK6	00e00-00eff
HK7	00f00-00fff

4.2.3 Sweep Tables

MACHO = Instrument **Ma**in **Ch**annel Aperture **O**peration
SACHO = Instruemnt **S**-**Ch**annel Aperture **O**peration

Each **address table** has 128 7-bit entries, corresponding to the steps in an ESA sweep. Each entry in the address table points to an entry in the MACHO or SACHO sweep table. There are 5 sweep tables, ESA, S-CH, S-CH TAP, DEFL-1 and DEFL-2.

Each **sweep table** is broken in two (coarse and fine... dfine is for possible future use), but can be taken as a single table with 128 16-bit entries. Each sweep table entry is input to a set of DACs, producing a voltage on the corresponding structure in the entry system (which is to be calibrated).



MACHO Address Table	01000-0107f	
SACHO Address Table	01080-010ff	Same as MACHO Address
MACHO ESA Course Table	01a00-01a7f	
SACHO ESA Course Table	01a80-01aff	Same as MACHO ESA Course
MACHO ESA Fine Table	01b00-01b7f	
SACHO ESA Fine Table	01b80-01bff	Same as MACHO ESA Fine
MACHO S-CH Course Table	01c00-01c7f	Set For S-CH Supply = 0V
SACHO S-CH Course Table	01c80-01cff	
MACHO S-CH Fine Table	01d00-01d7f	Set For S-CH Supply = 0V
SACHO S-CH Fine Table	01d80-01dff	
MACHO S-CH TAP Course Table	01e00-01e7f	Set For S-CH Supply = 0V
SACHO S-CH TAP Course Table	01e80-01eff	
MACHO S-CH TAP Fine Table	01f00-01f7f	Set For S-CH Supply = 0V
SACHO S-CH TAP Fine Table	01f80-01fff	
MACHO DEFL-1 Course Table	02000-0207f	
SACHO DEFL-1 Course Table	02080-020ff	
MACHO DEFL-1 Fine Table	02100-0217f	
SACHO DEFL-1 Fine Table	02180-021ff	
MACHO DEFL-1 DFine Table	02200-0227f	
SACHO DEFL-1 DFine Table	02280-022ff	
MACHO DEFL-2 Course Table	02300-0237f	Same as MACHO DEFL-1 Course
SACHO DEFL-2 Course Table	02380-023ff	Same as SACHO DEFL-1 Course
MACHO DEFL-2 Fine Table	02400-0247f	Same as MACHO DEFL-1 Fine
SACHO DEFL-2 Fine Table	02480-024ff	Same as SACHO DEFL-1 Fine
MACHO DEFL-2 DFine Table	02500-0257f	Same as MACHO DEFL-1 DFine
SACHO DEFL-1 DFine Table	02580-025ff	Same as SACHO DELF-1 DFine
[not used]	02600-0ffff	

4.2.4 Resistive Anode Table

Resistive Anode Table	10000-1ffff

```
if pos_ra = ff    then  ra_sat_a          is true
if pos_ra = fe    then  ra_sat_b          is true
if pos_ra = fd    then  ra_sat_both       is true
```

5 Operating Modes

NOTE: See "Event Selection" for trigger mode details.

5.1 Mode 0 – Halt Mode

Use this mode to exit from mode 3.

5.2 Mode 1 – Set DAC Registers

Use this mode to read DAC registers from Logic Board RAM and write to DAC board one time.

5.3 Mode 2 – Run Register Sequences

Use this mode to read ACTEL 2 Register Sequence from Logic Board EEPROM and write to ACTEL 2 one time.

5.4 Mode 3 – Normal Mode

Use this mode to begin automatic operation of instrument. Run mode 0, before any other command, to ensure proper exit from mode 3. Exception: the event_ctrl register may be written without exiting mode 3 (to set the s_ch_en bit).

After a time_sync occurs:

1. send tac0 reset, tac2 reset & ssd sync commands
2. send rst3_n to reset rate counters
3. set ACTEL 2 registers sel0_disable & sel2_disable to false
4. send **e_stp_stb** & **d_stp_stb** concurrently
5. send 256 d_stp_stb & 16 deflection_trig*
6. send **e_step_stb**
7. send 256 d_stp_stb & 16 deflection_trig*
8. send **e_stp_stb**
9. send **e_step_trig**
10. set ACTEL 2 registers sel0_disable & sel2_disable to true
11. read EEPROM DAC tables
12. send DAC board next state values... if s_ch_en = '1' then also set s-channel voltage
13. wait the remaining 0.026sec for hv supply settling
14. send instrument controller ESA step for Classifier telemetry
15. increment esa_step and send to ACTEL 2 (bit 7=s_ch_en)
16. perform above steps 128 times
17. run mode 4 (perform retrace interval steps)

* during the deflection_trig intervals, if ra_clr_n_b='1' AND ra_trig_b='1', perform an ADC on the resistive anode and send to ACTEL 2 in the RA Position register... if ra_clr_n_b='0' then set reg_addr(7) to '0'

After each deflection_trig:

1. increment defl_step and send along with esa_step (bit 7=s_ch_en) to ACTEL 2
2. send esa_step (bit 7=s_ch_en) & defl_step-1 instrument controller via util interface
3. send rates to instrument controller via util interface... if rate_chk='1' AND rlim > the rate for the rlim_ch selected, then set s_ch_en to '1' else set s_ch_en to '0'

5.5 Mode 4 – Retrace Interval

1. send **retrace_trig**
2. collect analog hk
    a. DAC hk
    b. Logic Board ADC hk
    c. SSD hk (SSD register sequences HK0..HK7)
3. collect digital hk
    a. read DAC registers and store in RAM one time
    b. read DAC status & LU count from RAM write hk RAM
    c. write plug_id & logic board adc_lu_flag to hk RAM
4. if retrace_rst_dis register = '0' then send logic board reset2_n & reset3_n & logic board adc_reset
5. send DAC board adc reset
6. run mode 1 to read DAC registers from RAM and write to DAC board one time
7. if retrace_ee_dis register = '0' then run mode 2 to read ACTEL 2 Register Sequence 0 from EEPROM and write to ACTEL 2 one time
8. send hk RAM block out util interface
9. run mode 3 to enter Normal Mode, release **retrace_trig** & wait for next time_sync

5.6 Mode 5 – Normal Mode

Use this mode to read DAC registers from DAC board and store in Logic Board RAM one time.

5.7 Mode 6 – Future Use 0

Same action as Mode 0.

5.8 Mode 7 – Future Use 1

Same action as Mode 0.

6 Event Selection

6.1 ssd (SW) Quadrants 0 & 1

NOTES ***tac0-ofw*** *is TAC0 adc overflow indication*
no energy-hk *indicates that the ssd ACTEL is NOT in house-keeping mode*
nomulti-energy & no energy-hk *are NOT used for validation if ssd_disable bit is set in SSD_CTRL register*
validation all modes require ***tac0_under ≤ tof ≤ tac0_over*** *&* ***ssd_under ≤ energy ≤ ssd_over***

6.1.1 SF & SFR (TOF) Mode Requirements

Mode 0 – Working Instrument (energy required)

for validation	sf, sfr, one position, energy, no multi-energy, no energy-hk, no tac0-ofw
s_e_not_rqd	sf, sfr, one position, no tac0-ofw
s_e_rqd	sf, sfr, one position, energy, no multi-energy, no energy-hk, no tac0-ofw

Mode 1 – Working Instrument (no energy required)

for validation	sf, sfr, one position, no multi-energy, no energy-hk, no tac0-ofw
s_e_not_rqd	sf, sfr, one position, no tac0-ofw
s_e_rqd	sf, sfr, one position, energy, no multi-energy, no energy-hk, no tac0-ofw

Mode 2 – Position Failure (energy required)

for validation	sf, sfr, energy, no multi-energy, no energy-hk, no tac0-ofw (0, 1 and 2 positions ok)
s_e_not_rqd	sf, sfr, no tac0-ofw
s_e_rqd	sf, sfr, energy, no multi-energy, no energy-hk, no tac0-ofw

Mode 3 – Position Failure (no energy required)

for validation	sf, sfr, no multi-energy, no energy-hk, no tac0-ofw (0, 1 and 2 positions ok)
s_e_not_rqd	sf, sfr, no tac0-ofw
s_e_rqd	sf, sfr, energy, no multi-energy, no energy-hk, no tac0-ofw

6.1.2 SF Mode Requirements

Mode 4 – TAC ADC Failure (energy required)

for validation	sf, one position, energy, no multi-energy, no energy-hk
s_e_not_rqd	sf, one position
s_e_rqd	sf, one position, energy, no multi-energy, no energy-hk

Mode 5 – TAC ADC Failure (no energy required)

for validation	sf, one position, no multi-energy, no energy-hk
s_e_not_rqd	sf, one position
s_e_rqd	sf, one position, energy, no multi-energy, no energy-hk

Mode 6 & Mode 7 – NOT USED

6.1.3 No SF and no SFR (TOF) Mode Requirements

Mode 8 – TAC Board Failure (energy required)

for validation	one position, energy, no multi-energy, no energy-hk
s_e_not_rqd	one position
s_e_rqd	one position, energy, no multi-energy, no energy-hk

<u>Mode 9 – TAC Board Failure (no energy required)</u>

for validation   one position, no multi-energy, no energy-hk
s_e_not_rqd   one position
s_e_rqd   one position, energy, no multi-energy, no energy-hk

<u>Mode 10 – TAC Board & Position Failure (energy required)</u>

for validation   energy, no multi-energy, no energy-hk (0, 1 and 2 positions ok)
s_e_not_rqd   [not used]
s_e_rqd   energy, no multi-energy, no energy-hk

Note that Mode 10 uses SSD information to determine position as follows:

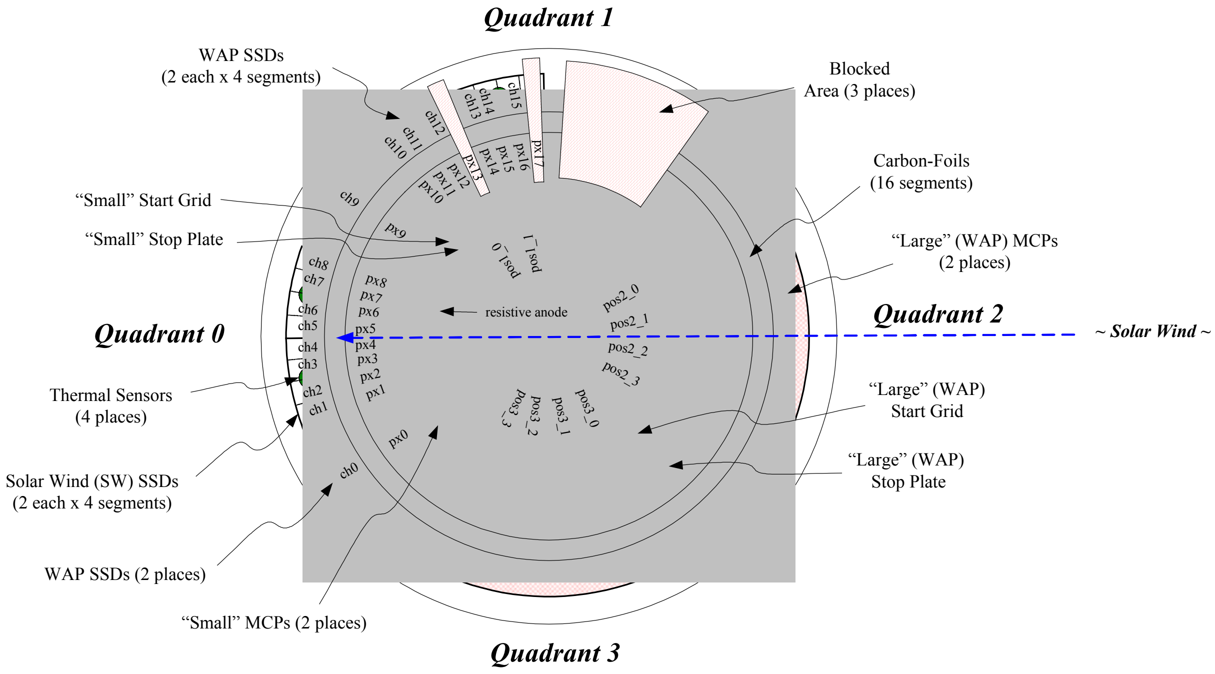
```
IF     ssd_id="0000" THEN     quadrant<="00"; position<="001000" (08);  section<="10";

ELSIF ssd_id="0001" THEN      quadrant<="00"; position<="010010" (18);  section<='0' & s_channel;
ELSIF ssd_id="0010" THEN      quadrant<="00"; position<="010110" (22);  section<='0' & s_channel;
ELSIF ssd_id="0011" THEN      quadrant<="00"; position<="011010" (26);  section<='0' & s_channel;
ELSIF ssd_id="0100" THEN      quadrant<="00"; position<="011110" (30);  section<='0' & s_channel;
ELSIF ssd_id="0101" THEN      quadrant<="00"; position<="100010" (34);  section<='0' & s_channel;
ELSIF ssd_id="0110" THEN      quadrant<="00"; position<="100110" (38);  section<='0' & s_channel;
ELSIF ssd_id="0111" THEN      quadrant<="00"; position<="101010" (42);  section<='0' & s_channel;
ELSIF ssd_id="1000" THEN      quadrant<="00"; position<="101110" (46);  section<='0' & s_channel;

ELSIF ssd_id="1001" THEN      quadrant<="00"; position<="111000" (56);  section<="10";

ELSIF ssd_id="1010" THEN      quadrant<="01"; position<="001000" (08);  section<="10";
ELSIF ssd_id="1011" THEN      quadrant<="01"; position<="001000" (08);  section<="10";
ELSIF ssd_id="1100" THEN      quadrant<="01"; position<="001000" (08);  section<="10";

ELSIF ssd_id="1101" THEN      quadrant<="01"; position<="011000" (24);  section<="10";
ELSIF ssd_id="1110" THEN      quadrant<="01"; position<="011000" (24);  section<="10";
ELSE                          quadrant<="01"; position<="011000" (24);  section<="10";
```



6.1.4 All Modes

1. Writing registers from ACTEL 1 (control actel) to ACTEL 2 has highest priority.
2. A force_reset_tac0 register command from ACTEL 1 is next lower priority. This command is normally used to reset the TAC0 board after a power-up sequence.
3. A write_tac0 register command is the next lower priority. This command is used with the tac0_stim for test pulser operation of the TAC0 board.

6.1.5 Modes 0 to 7 SF0 Starts Event

1. Assume the TAC0 and ssd boards have been reset and the tac0_sf_latch is reset and enabled. Note that the TAC0 board is reset after an event is processed. At this time the tac0_sfr, position (pos1_0 and pos1_1), resistive anode and ssd latches are all reset and disabled and will not capture pulses.

2. If tac0_sf_latch='1' AND sel0_disable=’0’ (used for event blanking during PAC HV recharge time), then enable the tac0_sfr, position, resistive anode and ssd latches.

Note that POS Disable Register 0 can be used to disable any or all of the 3 position channels. When a position channel is disabled, its associated rate-counter will not receive any pulses and will not count.

3. Wait until tac0_sfr_latch is set or sel0_window time, whichever comes first.
4. Disable inputs to all latches & get TOF information from TAC0.
5. Check ssd latch and get energy information if ssd latch is set.
6. Check position and ra latches and get resistive anode information if ra_latch is set.
7. Pulse the appropriate rate counters.
8. Check mode requirements.
9. If mode requirements are met, pulse the s_valid rate, build & send pha word to the Classifier Board.
10. Reset and disable tac0_sfr, position (pos1_0 and pos1_1), resistive anode and ssd latches. Reset and enable the tac0_sf_latch. Send a 1usec tac0_reset pulse to the TAC0 board & s_sync_n to ssd module.

6.1.6 Modes 8 and 9 Position Signal Starts event

1. Assume the ssd board has been reset and position latches are reset and enabled as well as the resistive anode. The ssd latch is reset and disabled. Note that the these actions occur after an event is processed.

2. If a position latch gets set AND sel0_disable=’0’ (used for event blanking during PAC HV recharge time), then enable the ssd latch.

Note that POS Disable Register 0 can be used to disable any or all of the 3 position channels. When a position channel is disabled, its associated rate-counter will not receive any pulses and will not count.

3. Wait until the ssd latch is set or sel0_window time, whichever comes first.
4. Disable inputs to all latches.
5. Check ssd latch and get energy information if ssd latch is set.
6. Check position and ra latches and get resistive anode information if ra latch is set.
7. Pulse the appropriate rate counters.
8. Check mode requirements.
9. If mode requirements are met, pulse the s_valid rate, build & send pha word to the Classifier Board.
10. Reset and enable position (pos1_0 and pos1_1), resistive anode and ssd latches.

6.1.7 Mode 10 Energy Signal Starts event

1. Assume the ssd board has been reset and the ssd latch enabled. Note that this action occurs after an event is processed.
2. When the ssd latch gets set, disable it and get energy information. Position determined by which ssd got triggered.
3. Pulse the appropriate rate counters.
4. Check mode requirements.
5. If mode requirements are met, pulse the s_valid rate, build & send pha word to the Classifier Board.
6. Reset and enable the ssd latch.

6.1.8 Building the 48-Bit PHA Word

esa_step	7 bits from sweep controller ACTEL 1
defl_step	5 bits from sweep controller ACTEL 1
quadrant	00 or 01 (00 if no position available, except mode 10)
ssd_id	4 bits from ssd board (0000 if none available)
ssd_energy	10 bits from ssd board (0000000000 if none available)
tof	10 bits from TAC0 board (1111111111 if none available or if modes > 3)
position*	000000..111111 (ra in quadrant 0) 001000, 011000 (8, 24 in quadrant 1 for bins 4, 5)
section **	00=SW_main channel (or no position available) 01=SW_S channel 10=WAP with SSD's
spare	00 (always)

* position = 000000 if no position information available (except mode 10)

** "Main channel" vs. "s-channel" information from sweep controller ACTEL 1
"SW" vs. "WAP with SSD's" determined from position information

6.2 non-ssd (WAP) Quadrants 2 & 3

NOTE validation all modes require ***tac2_under ≤ tof ≤ tac2_over***

6.2.1 Trigger Mode Register

tmode2='0' for normal mode
tmode2='1' do not invalidate events based on absence of a position pulse or for multiple position pulses. Classify all these events using quadrant 2, position 8.

6.2.2 tmode2 = '0' or '1'

1. Writing registers from ACTEL 1 (control actel) to ACTEL 2 has highest priority.
2. A force_reset_tac2 register command from ACTEL 1 is next lower priority. This command is normally used to reset the TAC2 board after a power-up sequence.
3. A write_tac2 register command is the next lower priority. This command is used with the tac2_stim for test pulser operation of the TAC2 board.
4. Assume the TAC2 board has been reset and is waiting for a start pulse. Note that the TAC2 board is reset after an event is processed (or aborted). Also, the tac2_sf is reset and enabled, the tac2_sfr latch & position latches are reset and disabled and will not capture pulses.
5. If tac2_sf_latch='1' AND sel2_disable='0' (used for event blanking during PAC HV recharge time), then begin looking for a tac2_sfr signal and enable the position latches for pos2_0, pos2_1, pos2_2, pos2_3, pos3_0, pos3_1, pos3_2, pos3_3.

Note that POS Disable Register 2 can be used to disable any or all of these position channels. When a position channel is disabled, its associated rate-counter will not receive any pulses and will not count.

6. After sel2_window time OR after tac2_sfr_latch is set (whichever comes first), disable the position latches and evaluate:
   a. if no position arrived, pulse the w_n_pos rate-counter
   b. if more than one position arrived, pulse the w_multi_pos rate-counter
   c. if tmode2='0' AND if no position arrived, the event is NOT valid – abort
   d. if tmode2='0' AND if more than one position arrived, the event is NOT valid – abort
   e. if tmode2='1' AND tac2_sfr_latch isn't set, the event is NOT valid – abort
   f. otherwise continue
7. Perform an A/D conversion on TAC2 to get TOF.
8. If tac2_ofw='1' then the event is NOT valid – abort (this is a TAC2 board a/d overflow).
9. If no abort, the event is valid. Pulse the w_val rate-counter, build & send pha word to the Classifier Board. Note that if tmode2='1', then the quadrant is set to 2 and the position in the quadrant is set to 0.
10. Reset and disable tac2_sfr_latch & position latches, reset and enable the tac2_sf_latch and send a 1usec tac2_reset pulse to the TAC2 board after an event is processed or after an abort.

6.2.3 Building the 48-Bit PHA Word

esa_step	7 bits from sweep controller ACTEL 1
defl_step	5 bits from sweep controller ACTEL 1
quadrant	10 or 11 (10 if no position available)
ssd_id	0000
ssd_energy	0000000000
tof	10 bits from TAC2 board
position*	001000, 011000, 101000, 111000 (8, 24, 40 or 56 in quadrant 2 for bins 0..3)
	001000, 011000, 101000, 111000 (8, 24, 40 or 56 in quadrant 3 for bins 4..7)
section	11 (WAP without SSDs)
spare	00 (always)

* position = 001000 if no position information is available or if multiple position pixels are triggered

7 Rates

7.1 Quadrants 0 & 1

rt(31)	s_valid	valid events (meet validation requirements… see mode details)
rt(30)	s_e_not_req	events with energy not required (see mode details)
rt(29)	s_e_req	events with energy required (see mode details)
rt(28)	s_no_pos	events with no position trigger
rt(27)	s_mult_pos	events with multiple position triggers
rt(26)	s_no_e	events with no energy trigger
rt(25)	s_mult_e	events with multiple energy triggers
rt(24)	ra_sat_a	resistive anode saturation counters (4)
rt(23)	ra_sat_b	|
rt(22)	ra_sat_both	|
rt(21)	ssd_sw	ssd solar-wind rate
rt(20)	ssd_st	ssd supra-thermal rate
rt(19)	sf0	TAC0 start-flag
rt(18)	sfr0	TAC0 stop coincidence (after start)
rt(17)	stp0	TAC0 “True” stop rate
rt(16)	ra_trig	resistive anode position trigger
rt(15)	pos1_0	position trigger
rt(14)	pos1_1	position trigger

7.2 Quadrants 2 & 3

rt(13)	w_no_pos	events not valid no position
rt(12)	w_mult_pos	events not valid multiple positions
rt(11)	w_valid	events valid
rt(10)	sf2	TAC2 start-flag
rt(9)	sfr2	TAC2 stop coincidence (after start)
rt(8)	stp2	TAC2 “True” stop rate
rt(7)	pos2_0	position triggers
rt(6)	pos2_1	|
rt(5)	pos2_2	|
rt(4)	pos2_3	|
rt(3)	pos3_0	|
rt(2)	pos3_1	|
rt(1)	pos3_2	|
rt(0)	pos3_3	|

Actual Rate [Hz] = rt(n) / 12.8msec (12.8msec = 1 deflection step)

Note that all rate counter, except rt(31) & rt(30), act independent of event validation.

Resistive Anode Note: *bit 7 from RA table shall = ‘1’ for saturation A*
bit 6 from RA table shall = ‘1’ for saturation B

8 Classifier

8.1 48-Bit PHA Word To Classifier

Description	Name	Range	Bits	Notes
ESA Step	SWPE	0-127	7	an ESA sequence will step through 128 steps
Deflection Step	SWPD	0-31	5	for +/- 20 degrees, only affects the SW sector (section=0)
Quadrant	QUADRANT	0-3	2	determined from which anode is triggered
Which SSD	SSD_ID	0-15	4	10 SSD signals in Q0, 6 SSD signals in Q1
SSD Energy	SSDE	0-1023	10	0 = no SSD measurement, SSDE<threshold also considered zero
Time of Flight	TOF	0-1023	10	1023 = no TOF
Position in Quadrant	POSITION	0-63	6	64 steps in 90 degrees is 1.4 degrees/step
Instrument Section	SECTION	0-3	2	which portion of the instrument is triggered: 0=SW_main channel 1=SW_S channel 2=WAP with SSD's 3=WAP w/o SSD's
Spare	SPARE	0-3	2	not used

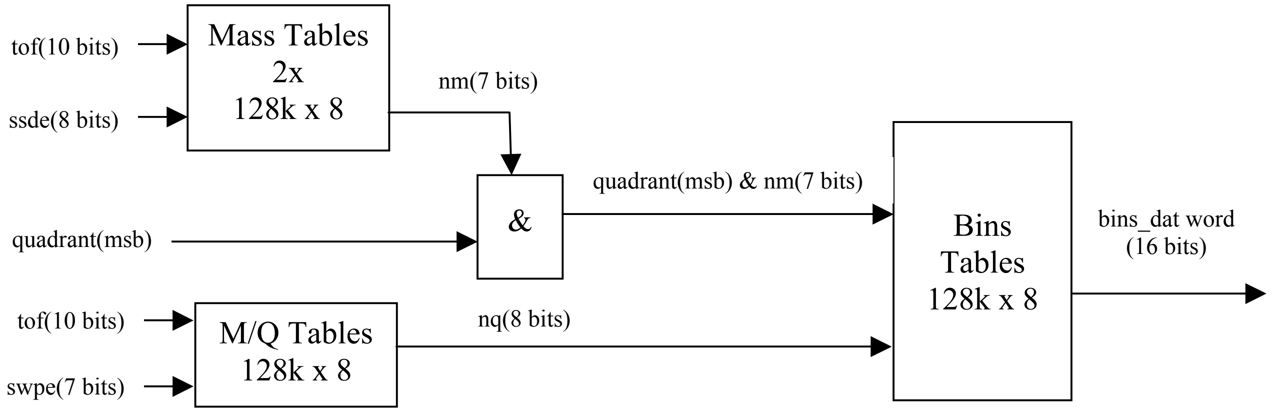
Bit Sequence Definition:
SWPE=47..41 SWPD=40..36 QUADRANT=35..34 SSD_ID=33..30 SSDE=29..20 TOF=19..10 POSITION=9..4 SECTION=3..2 SPARE=1..0

8.2 48-Bit Stored PHA Word From Classifier

Description	Name	Range	Bits	Notes
ESA Step	SWPE	0-127	7	an ESA sequence will step through 128 steps
Deflection Step	SWPD	0-31	5	for +/- 20 degrees, only affects the SW sector (section=0)
Quadrant	QUADRANT	0-3	2	determined from which anode is triggered
Which SSD	SSD_ID	0-15	4	10 SSD signals in Q0, 6 SSD signals in Q1
SSD Energy	SSDE	0-1023	10	0 = no SSD measurement, SSDE<threshold also considered zero
Time of Flight	TOF	0-1023	10	1023 = no TOF
Position in Quadrant	POSITION	0-63	6	64 steps in 90 degrees is 1.4 degrees/step
Instrument Section	SECTION	0-3	2	which portion of the instrument is triggered: 0=SW_main channel 1=SW_S channel 2=WAP with SSD's 3=WAP w/o SSD's
Priority	PRIORITY	0-3	2	PHA priority rates

Bit Sequence Definition:
SWPE=47..41 SWPD=40..36 QUADRANT=35..34 SSD_ID=33..30 SSDE=29..20 TOF=19..10 POSITION=9..4 SECTION=3..2 PRIORITY=1..0

8.3 EEPROM Tables



NOTES:
1. MSBit of "Mass Tables" data byte is ignored (remaining 7 bits are Nm)
2. quadrant(msb) = '0' for sections 0,1 & 2 else '1' for section 3
3. ssde(8 bits) is compressed from 10 bits

If $E_d < 96$, $E_{compressed} = E_{digital}$

Else $E_c = \text{INT}[\, E_d / 2^L \,] + 48{*}L$,
where $L = \text{INT}[\, \log_2(E_d / 48)]$

Measured Energy E_m [keV]	**10-BIT ENERGY** (Event Word E_d)		**8-BIT ENERGY** (Mass Classification E_c)			
	Linear Channel	**Number of Channels**	**Compressed Channel**	**Linear chn per Compr. chn**	**keV per Compr. chn**	**Chn Resolution at lower to upper measured energy**
0-191	0-95	96	0-95	1	2 keV/Chn	8% (25keV) to 1%
192-383	96-191	96	96-143	2	4 keV/Chn	2.1% to 1%
384-767	192-383	192	144-191	4	8 keV/Chn	2.1% to 1%
768-1535	384-767	384	192-239	8	16 keV/Chn	2.1% to 1%
1536-2047	768-1023	256	240-255	16	32 keV/Chn	2.1% to 1.6%

8.4 EEPROM Tables Address Logic

n_cs1(0..3) correspond to 4 each, 128k x 8 EEPROMs

mass_tables_addr (16..0) = tof (8..0) & Ecompressed (7..0) tof(9)=0 for n_cs1(0) tof(9)=1 for n_cs1(1)

m/q_tables_addr (16..0) = tof(9..0) & swpe (6..0) n_cs1(2)

bins_tables_addr (16..0) = quadrant(1) & Nm(6..0) & Nq(7..0) & 0 [for bins_dat word (7..0)] n_cs1(3)
bins_tables_addr (16..0) = quadrant(1) & Nm(6..0) & Nq(7..0) & 1 [for bins_dat word (15..8)] n_cs1(3)

8.5 Bins_Dat Word

one set for SECTONS 0, 1 & 2 & one set for SECTION 3

pha pri rates	*supra no E*	*supra wide*	*sw Z >2*	*sw all*	*sw H alpha*
0 0	0 0 0	0 0 0 0	0 0 0 0	0	0 0

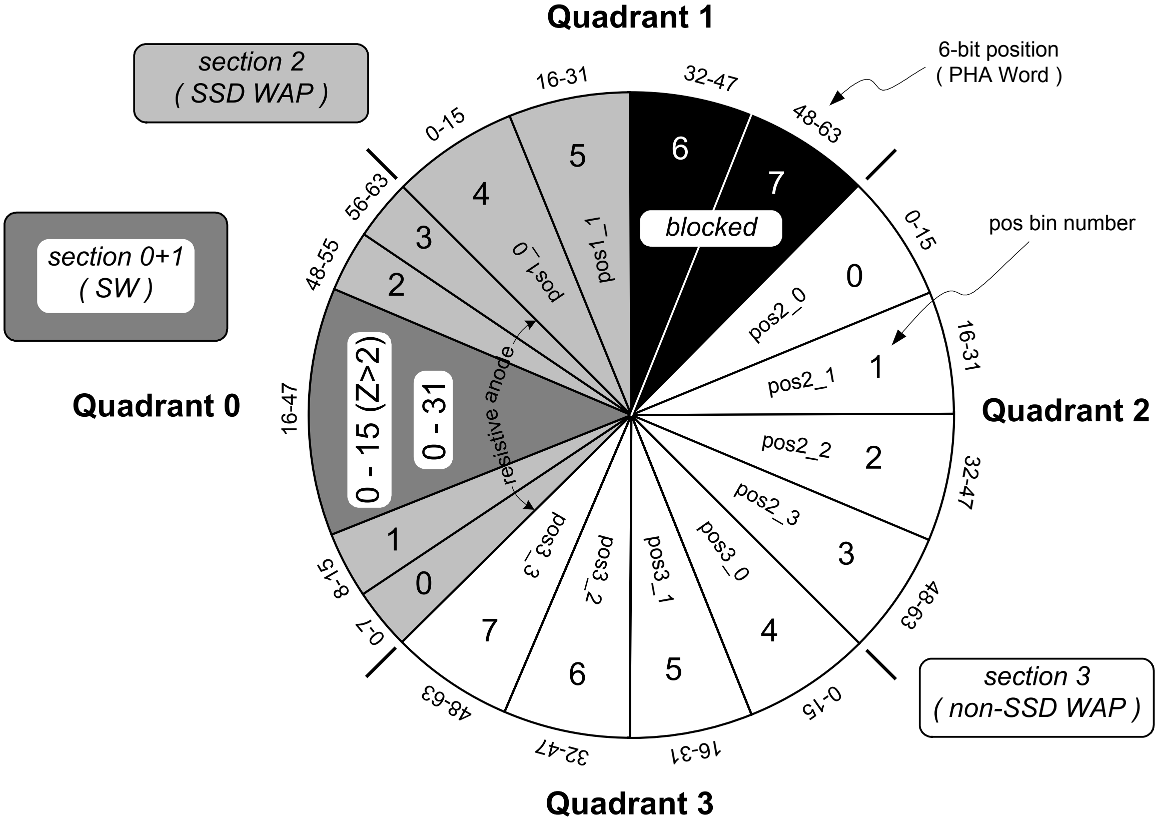
NOTE: only MSB of *pha pri rates* is used for WAP_PHA_Priority_rates

Type	***Total Bins***	***Range***	***Do Not Bin Code***
sw H alpha	3	00 - 10	11
sw all	1	0 only	1 only
sw Z > 2	15	0000 - 1110	1111
supra wide	15	0000 - 1110	1111
supra no E	7	000 - 110	111
pha pri rates	4	00 - 11	always bin

Name	*Class. Bins*	*Pos. Bins*	*Def. Bins*	*Energy Steps*	*No. of Sections*	*Section*	*Total Bins*	*Bits/Item*	*Total Bytes*
SW H / Alpha*	3	32	32	1	1	0 or 1	3072	16	6144
SW_All	1	32	32	1	1	0 or 1	1024	16	2048
SW Z>2 doubles & triples	15	16	8	1	1	0 or 1	1920	16	3840
Suprathermal – Wide triples	15	8	1	1	1	2	120	16	240
Suprathermal – noE doubles	7	8	1	1	2	2,3	112	16	224
SW_PHA_Priority_rates	4	1	32	1	1	0 or 1	128	16	256
WAP_PHA_Priority_rates	2	1	1	1	2	2,3	4	16	8
PHA Data							512	48	3072
Total									15832

* 2 bins for doubles and 1 bin for triples

8.6 Position Bin Information



A resistive position anode is used in Quadrant 0. There are also 10 discreet position anodes --- 2 in Quadrant 1 (#4 & #5) --- 4 in Quadrant 2 (#0..#3) --- and 4 in Quadrant 3 (#4..#7).

quadrant	*6-bit position*	*section*	*data type*	*pos bin no.*
0	16 - 47	0 or 1	sw H alpha / sw all	position-16
0	16 - 47	0 or 1	sw Z > 2	(position-16) / 2
0	0 - 7	2	supra wide / supra no E	0
0	8 - 15	2	supra wide / supra no E	1
0	48 - 55	2	supra wide / supra no E	2
0	56 - 63	2	supra wide / supra no E	3
1	0 - 15	2	supra wide / supra no E	4
1	16 - 31	2	supra wide / supra no E	5
1	32 - 47	2	blocked / not used	6
1	48 - 63	2	blocked / not used	7
2	0 - 63	3	supra no E	position / 16
3	0 - 63	3	supra no E	position / 16 + 4

Quadrant 0:

```
IF section = "00"  OR  section = "01" THEN
 pos = NOT [position(4)] & position(3..0)
```

	position						*pos*				
#	5	4	3	2	1	0	4	3	2	1	0
16-31	0	1	x	x	x	x	0	x	x	x	x
32-47	1	0	x	x	x	x	1	x	x	x	x

pos(0) not used for sw Z>2

```
ELSE IF quadrant = "00" THEN
pos = "000" & position(4..3)
```

	position						*pos*				
#	5	4	3	2	1	0	4	3	2	1	0
00-07	0	0	0	x	x	x	0	0	0	0	0
08-15	0	0	1	x	x	x	0	0	0	0	1
48-55	1	1	0	x	x	x	0	0	0	1	0
56-63	1	1	1	x	x	x	0	0	0	1	1

Quadrants 1, 2 & 3:

```
ELSE pos = "00" & quadrant(0) & position(5..4)
```

	quad	*position*						*pos*				
#	(0)	5	4	3	2	1	0	4	3	2	1	0
00-15	0	0	0	x	x	x	x	0	0	0	0	0
16-31	0	0	1	x	x	x	x	0	0	0	0	1
32-47	0	1	0	x	x	x	x	0	0	0	1	0
48-63	0	1	1	x	x	x	x	0	0	0	1	1
00-15	1	0	0	x	x	x	x	0	0	1	0	0
16-31	1	0	1	x	x	x	x	0	0	1	0	1
32-47	1	1	0	x	x	x	x	0	0	1	1	0
48-63	1	1	1	x	x	x	x	0	0	1	1	1

8.7 RAM Bins Memory Map

0000 - 17ff	sw H alpha	6144 bytes
1800 - 1fff	sw all	2048 bytes
2000 - 2eff	sw Z > 2	3840 bytes
2f00 – 2fff	sw pha rates	256 bytes
3000 - 30ef	supra wide	240 bytes
30f0 - 30ff	SCRATCH RAM	16 bytes
3100 – 31df	supra no E	224 bytes
31e0 – 31ff	NOT USED	32 bytes
3200 – 3207	wap pha rates	8 bytes
3208 – 3fff	NOT USED	3576 bytes

8.8 RAM Bins Address Logic

```
addr(0) = addr_null WHEN (sw_halpha = “11” OR section(1) = ‘1’) ELSE
      “00”  &  sw_halpha(1..0)  &  pos(4..0)  &  swpd(4..0)  &  a_lsb

addr(1) = addr_null WHEN (sw_all = ‘1’ OR section(1) = ‘1’) ELSE
      “001 1”  &  pos(4..0)  &  swpd(4..0)  &  a_lsb

addr(2) = addr_null WHEN (sw_zgr2  = “1111” OR section(1) = ‘1’) ELSE
      “010”  &  sw_zgr2(3..0)  &  pos(4..1)  &  swpd(4..2)  &  a_lsb

addr(3) = addr_null WHEN (supra_wid = “1111” OR section ≠ “10”) ELSE
      “011 0000”  &  supra_wid(3..0)  &  pos(2..0)  &  a_lsb

addr(4) = addr_null WHEN (supra_noe = “111” OR section(1) = ‘0’) ELSE
      “011 0001” supra_noe(2..0)  &  pos(2..0)  &  section(0)  &  a_lsb

addr(5) =

“010 1111”  &  pha_rates(1..0)  &  swpd(4..0)  &  a_lsb  WHEN section(1) = ‘0’

ELSE   “011 0010 0000 0”  &  pha_rates(1)  &  section(0)  &  a_lsb
```

NOTES:

```
addr_null   =  “011 0000 1111 111”  &  a_lsb  (SCRATCH RAM memory)

a_lsb  =    address lsb       0 for data bits  7..0
                              1 for data bits 15..8
```

8.9 PHA Data Memory Map

```
4000 - 41ff     sw0         (64)        384  +     128 =  512 bytes
4200 - 47f7     NOT USED                                 1528 bytes
47f8 - 47ff     SCRATCH RAM                                 8 bytes
4800 - 49ff     sw1         (64)        384  +     128 =  512 bytes
4a00 - 4ff7     NOT USED                                 1528 bytes
4ff8 - 4fff     SCRATCH RAM                                 8 bytes
5000 - 54ff     sw2        (160)        960  +     320 = 1280 bytes
5500 - 57f7     NOT USED                                  760 bytes
57f8 - 57ff     SCRATCH RAM                                 8 bytes
5800 - 5cff     sw3        (160)        960  +     320 = 1280 bytes
5d00 - 5ff7     NOT USED                                  760 bytes
5ff8 - 5fff     SCRATCH RAM                                 8 bytes
6000 - 60ff     wap0        (32)        192  +      64 =  256 bytes
6100 - 67f7     NOT USED                                 1784 bytes
67f8 - 67ff     SCRATCH RAM                                 8 bytes
6800 - 68ff     wap1        (32)        192  +      64 =  256 bytes
6900 - 6ff7     NOT USED                                 1784 bytes
6ff8 - 6fff     SCRATCH RAM                                 8 bytes
7000 - 7fff     NOT USED                                 4096 bytes
                             PHA DATA        +  SPARE
```

8.10 PHA Word Address Logic

```
            pha_addr(0)   (7..0) = pha_base_addr  &  00  &  0
            pha_addr(1)  (15..8) = pha_base_addr  &  00  &  1
            pha_addr(2) (23..16) = pha_base_addr  &  01  &  0
            pha_addr(3) (31..24) = pha_base_addr  &  01  &  1
            pha_addr(4) (39..32) = pha_base_addr  &  10  &  0
            pha_addr(5) (47..40) = pha_base_addr  &  10  &  1
                         (SPARE) = pha_base_addr  &  11  &  0
                         (SPARE) = pha_base_addr  &  11  &  1

                       ↑
               48-bit pha word bits
```

```
pha_base_addr = “1”  &  section(1)  &  pha_rates(1..0)  &  pha_cnt(7..0)

IF    (section(1) & pha_rates)="000" THEN pha_cnt<= sw0_cnt;
ELSIF (section(1) & pha_rates)="001" THEN pha_cnt<= sw1_cnt;
ELSIF (section(1) & pha_rates)="010" THEN pha_cnt<= sw2_cnt;
ELSIF (section(1) & pha_rates)="011" THEN pha_cnt<= sw3_cnt;
ELSIF (section(1) & pha_rates)="100" THEN pha_cnt<=wap0_cnt;
ELSIF (section(1) & pha_rates)="101" THEN pha_cnt<=wap1_cnt;
ELSIF (section(1) & pha_rates)="110" THEN pha_cnt<=0;
ELSIF (section(1) & pha_rates)="111" THEN pha_cnt<=0;
```

NOTES:

a separate pha_cnt(7..0) is used for each priority rate (6 total)
pha_cnt(7..0) is incremented once per pha write cycle (2, 8-bit writes)
pha_addr_3 is SPARE and contains no data

8.11 Signal Functions & Phasing

ser_dat(5..0): Positive logic, serial data inputs, 8-bit wide, MSB first. Must be stable before falling edge of ser_clk.

ser_clk: 12.8MHz serial clock input. Falling edge captures ser_dat(n).

ser_gat: Gate input. Must be logic '1' to enable ser_dat(n) input. Falling edge latches the data. ser_gat must transition on rising edge of ser_clk.

busy: Output goes to logic '1' one clock cycle after 8 serial bits are shifted in and ser_gat returns to '0'. BUSY will stay '1' until the classifier has finished processing the event. A second will not be clocked in on the serial link until <u>after</u> BUSY goes back to '0'. EEPROMs may be reprogrammed when BUSY is '0'.

ram_sel: Input. Logic '0' selects RAM bins_a for classifier data storage. Logic '1' for RAM bins_b selection. Must toggle ram_sel when BUSY='0'. Data offload may occur on RAM part that is NOT selected.

tab_tri: Test output. Used for internal PGA buffers. '1' indicates EEPROM table buffers are tri-stated and not being accessed.

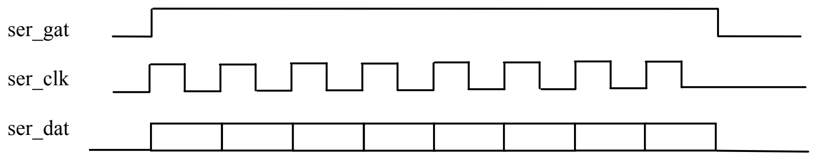
dat2_w: Test output. Used for internal PGA buffers. '1' indicates write operation to RAM.

bins_tri: Test output. Used for internal PGA buffers. '1' indicates RAM buffers are tri-stated and not being accessed.

spare: Test output. Connected to incoming pha_word(0) (spare bit).

n_rst: Reset to PGA for all memory elements. '0' for reset and '1' to operate.

clk: 25.6MHz clock input to PGA. Not sync'd to SER_CLK.



NOTES:

The 48-bit PHA word to classifier is made up of 6 [ser_dat(0..5)], 8-bit, words. The PHA MSByte is ser_dat(5) and the LSByte is ser_dat(0).

It takes 139 clk cycles to process an event beginning at the falling edge of ser_gat. At 25.6MHz, that is 5.43uS. The above does not include clocking in of ser_dat, which will take an additional 8 clk cycles at 12.8MHz, or 0.63uS. Therefore, the total processing time of one event is 6.06uS or 165kHz.

Regarding power-up: When busy drops to '0', about 8us after n_rst goes to '1', both 32k x 8 RAM parts must be cleared, presumably by doing a dummy read-and-clear.

8.12 Hardware & Layout

1. (4 ea) 128k x 8 EEPROM's must be 120ns or faster (Maxwell 28C011TRPFE-12).

2. (2 ea) 32k x 8 RAM must be 25ns or faster (Honeywell HX6356).

3. PGA is an ACTEL RT54SX32S-CQ208B (A54SX32A-PQ208 for commercial version).

4. Layout PCB to accommodate sockets, CQ208 and the PQ208 packages.

5. Pull-up resistors must be installed on all address, data, n_cs, n_rd and n_wr lines of EEPROM and RAM parts and also on ser_dat, ser_gat, ser_clk and ram_sel lines.

6. Allow for hard-wire grounding of ACTEL RTST pin (JTAG disable for flight).

7. Schmitt trigger may be required on 25.6MHz oscillator if it's slew rate is too slow (see "using Schmitt triggers" App. Note)

8. For ACTEL: input protection (and Schmitt triggers) may be required on ser_dat, ser_gat, ser_clk, ram_sel and n_rst. Output driver may be required on busy line.

9. Vcca must not be above 0.0V during power up. Vcca ramp rate must be 0.66V/ms or slower. ACTEL I/O's drive low during power up – be sure no inputs are applied during power up sequence (see "power-up / power-down App. Note).

10. Pull-down unused ACTEL I/O pins.

9 Logic Board Signals & Notes

9.1 Trig Signals

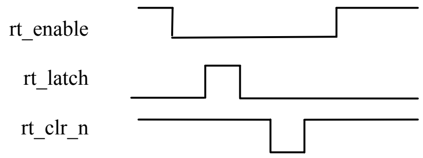
The following trig signals are all 400usec wide.

e_step_trig pulses are sent from ACTEL 1 to the instrument controller at a rate of 128 per minute. After the pulse is sent, the previous esa step (the step corresponding to the classifier data) is sent via the util_dat interface. Next, the instrument controller switches classifier RAM banks and send classifier data to the IDPU.

deflection_trig pulses are sent from ACTEL 1 to the instrument controller at a rate of 4096 per minute. After the pulse is sent, rates data is send to sent to the instrument controller via the util_dat interface.

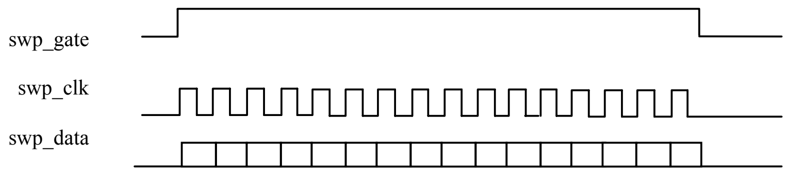
retrace_trig pulses are sent from ACTEL 1 to the instrument controller at a rate of 1 per minute. After the pulse is sent, house-keeping data (with headers) is sent to the instrument controller via the util_dat interface. Next, ACTEL 1 updates all ACTEL 2 registers and HV control registers.

9.2 Rate Control



9.3 HV Control

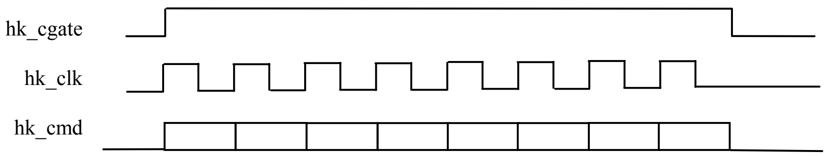
8-bits HV-address immediately followed by 8-bits HV-data are sent to the DAC ACTEL using swp_data at 3.2MHz as follows, MSB first:



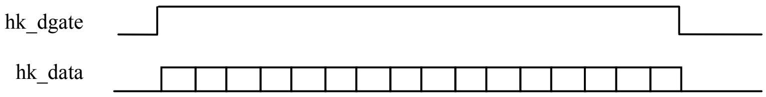
The d_stp_srb and e_stp_stb are both 400usec wide. They will be sent from ACTEL 1 to the DAC ACTEL after the “next state” DAC registers are sent. The e_stp_stb will initiate a transfer of data from “next state” to “present state” while the d_stp_srb simply increments the deflection ramp counter in the DAC. These registers may be read-back via the House-Keeping Control interface. d_stp_srb and e_stp_stb shall be sent concurrently for to indicate the end of a retrace interval.

9.4 House-Keeping Control

ACTEL 1 on the LOGIC board will send an 8-bit house-keeping address to the DAC board. The DAC board in response shall perform an a/d conversion. This address is sent to the DAC ACTEL using hk_cmd with hk_clk at 3.2MHz as follows, MSB first:

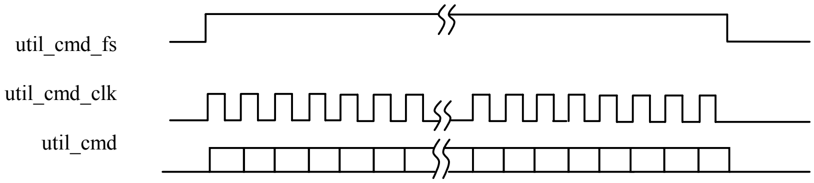


The DAC ACTEL shall respond with the house-keeping data within 72 clock cycles (of the 25.6MHz system clock) of the falling edge of hk_cgate and send the requested house-keeping data to ACTEL 1 at 3.2MHz (clk / 8… using hk_dat), MSB first. When reading an 8-bit control register, the 8-bit register is repeated twice to make up the 16-bit hk_data word.

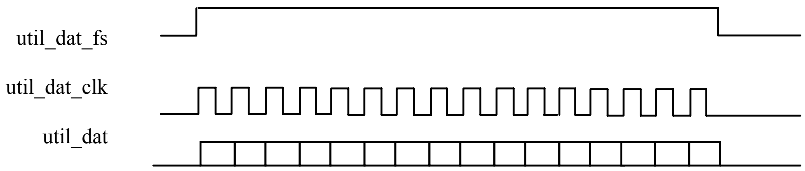


9.5 UTIL Interface

24-bit UTIL command is sent from the instrument controller to ACTEL 1 at 3.2MHz as follows, MSB first:



16-bit UTIL data is sent from ACTEL 1 to the instrument controller at 3.2MHz as follows, MSB first:



10 TAC Board Signals & Notes

1. Rising edge of TAC RESET pulse (from LOGIC board)
    a. START flip-flop is cleared
    b. STOP flip-flop is cleared via START flip-flop
    c. Integration capacitor begins discharging

2. During TAC RESET pulse
    a. START flip-flop is held in clear condition – start signals have no affect
    b. STOP flip-flop is held in clear condition via START flip-flop – stop signals have no affect
    c. Integration capacitor discharges

3. Falling edge of TAC RESET pulse
    a. START flip-flop ready – start signals will now affect START flip-flop
    b. STOP flip-flop held in clear condition via START flip-flop – stop signals have no affect
    c. Integration capacitor fully discharged

4. Awaiting Start Pulse

5. Start Pulse arrives
    a. START flip-flop sets – new start signals have no affect
    b. STOP flip-flop made ready via START flip-flop – stop signals will now affect STOP flip-flop
    c. Integration capacitor begins charging

6. If no Stop Pulse arrives
    a. Overflow Reset Circuit responds after voltage on integrating capacitor reaches about 6V (about 300nS)
    b. TAC board performs first 3 reset steps described above

7. If Stop Pulse arrives
    a. START flip-flop remains set – new start signals have no affect
    b. STOP flip-flop sets – new stop signals have no affect
    c. Integration capacitor stops charging

8. Perform A/D conversion to measure time-of-flight
    a. Begin at first step above

NOTE: "Stop" output signal for rates is always enabled

VHDL NOTE: tac0_tof<=NOT (tac0_ofw_n) & tac0_adc and tac2_tof<=NOT (tac2_ofw_n) & tac2_adc

11 SSD Commands, Signals & Notes

11.1 Command Format

The general command format is:

```
 MSB                                                         LSB
  15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|    Command    |                 Parameters                    |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Commands are transmitted bit 15 (MSB) first to bit 0 (LSB) last. Bits 15-12 contain the command and bits 11-0 contain command parameters, if any. Commands may be sent at any time, except when reading housekeeping or status information or when loading the ASIC configuration.

The relationship between the CMD_IN, CMD_CLK, and CMD_STROBE signals when sending a command should be:

```
              +---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
CMD_IN        | 15| 14| 13| 12| 11| 10| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
--------------+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+------

                +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+
CMD_CLK         | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
----------------+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +------

                                                                          +-----+
CMD_STROBE                                                                |     |
--------------------------------------------------------------------------+     +----
```

Note that the CMD_IN bits will be sampled by the SSDB Actel on the rising edge of MD_CLK. They should be changed coincident with the falling edge of CMD_CLK. Note also that CMD_STROBE should rise coincident with the leading edge of the last CMD_IN bit and be held high for one and one half clock times. The CMD_IN, CMD_CLK, and CMD_STROBE signals should be held in a 0 = low = no light state when inactive. For command bits on CMD_IN, 0 = low = no light and 1 = high = light.

11.2 NOOP Command

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

This command performs no action. It may be used as a time or space filler in a sequence of commands. (In this and in the following commands, "-" indicates a parameter bit which is currently ignored by the SSDB Actel and is, in effect, a "don't care". However, these bits should be set to "0" in commands to allow for future expansion.)

11.3 Mode Command

```
  15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 0 | 0 | 1 | - | - | - | - | - | - | - | - | - | - | - |h/e|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

h/e: 0 = energy mode, 1 = housekeeping mode.

This command is used to control whether energy data or housekeeping data is read back from the SSDB Actel.

11.4 Stimulus Command

```
  15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 0 | 1 | 0 | - | - | - | - | - | - | - | - |db3|db2|db1|db0|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

db3-db0: 0000 = low, 1111 high.

This command is used to set the stimulus level.

11.5 ADC Command

```
  15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 0 | 1 | 1 | - | - | - | - | - | - | - | - | sw|res| cs| rd|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

sw:	0 = normal operation	1 = reset ADC SEU power switch
res:	0 = normal operation	1 = reset ADC chip
cs:	0 = normal operation	1 = disable ADC chip
rd:	0 = normal operation	1 = disable ADC outputs

This command is used control the analog to digital converter.

11.6 MUX Command

```
  15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 1 | 0 | 0 | - | - | - | - | - | - | - | - |inh|ad2|ad1|ad0|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

inh: 0 = normal operation 1 = inhibit (disable) mux
ad2-ad0: mux address

This command is used control the analog mux.

11.7 ASIC Command

```
  15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 1 | 0 | 1 | - | - | - | - | - | - | - | - | - |pwr|mrs| tx|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

pwr:	0 = normal operation	1 = reset ASIC SEU power switch
mrs:	0 = normal operation	1 = reset ASIC chip
tx:	0 = normal operation	1 = pass following 11 commands words to the ADC chip

This command is used control the ASIC chip. An ASIC command, which sets the tx bit, should be followed immediately by eleven 16-bit words containing the configuration data for the ASIC chip. These eleven 16-bit words are ignored by the command logic and passed without change directly to the ASIC chip. After receipt of the eleventh word, the tx bit is automatically reset and any subsequent words are interpreted as commands. A detailed description of the eleven 16-bit words is shown in “Command Sequence for ASIC Register Loading” section below.

11.8 READ Command

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

This command is used to read housekeeping or status information. Before sending a READ command, an (normal) ADC command should sent to configure the ADC and a MUX command sent to select the signal source. Receipt of a READ command will cause a convert signal to be issued to the ADC. After a suitable amount of time, the value of the conversion may be read.

11.9 Response Data

Meaningful information may be read TBD microseconds after a READ command in house-keeping mode or a TRIGGER_VE signal in energy mode. The meaning of the data read will depend on the mode and, in housekeeping mode, on the setting of the MUX inh bit.

11.10 h/e = 0

```
  15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| ASIC channnel |h/e| mh|           ADC output                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

11.11 h/e = 1 and inh = 0

```
  15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|inh|ad2|ad1|ad0|h/e| mh|           ADC output                  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

11.12 h/e = 1 and inh = 1

```
  15  14  13  12  11  10  9   8   7   6   5   4   3   2   1   0
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|inh|ad2|ad1|ad0|h/e| mh| pe| sw|res| cs| rd|seu|pwr|mrs| tx| 0 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Note: mh is the MULTIHIT flag.
pe is the ADC power on flag.
seu is the ASIC power on flag.

The output data is self-describing. Bit 11 (h/e) identifies energy and housekeeping data. If bit 11 = 1, bit 15 = 1 identifies status data and bit 15 = 0 identifies ADC data. In ADC data, bits 14-12 identify the data source.

11.13 Reading Data

To read data, the CMD_CLK line should be pulsed ten times. CMD_IN should be held low and no CMD_STROBE should be issued. The response data will appear on the DATA_OUT_1 and DATA_OUT_0 lines.

```
CMD_IN
--------------------------------------------------------------------

                +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+
CMD_CLK         | | | | | | | | | | | | | | | | | | | |
----------------+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-+ +-------------


CMD_STROBE
--------------------------------------------------------------------

                   +---+---+---+---+---+---+---+---+---+
DATA_OUT1          |   | 15| 14| 13| 12| 11| 10| 9 | 8 |
-------------------+   +---+---+---+---+---+---+---+---+--------

                    +---+---+---+---+---+---+---+---+---+
DATA_OUT0           |   | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
--------------------+   +---+---+---+---+---+---+---+---+-------
```

10 command clocks are sent to the SSD ACTEL to clock out 8 bits of energy / hk data on s_dat0 and s_dat1 for a total of 16 bits of data. The first clock guarantees a 0-to-1 transition on ssd_dat0 and s_dat1 for the receiver to synchronize to. The final clock forces s_dat0 and s_dat1 back to 0. s_sync and s_m_reset both will force ssd_dat0 and ssd_dat1 to 0.

Each DATA_OUT signal consists of a high start bit, eight data bits, and a low stop bit. At the SSDB chip, DATA_OUT transitions occur on the falling edge of CMD_CLK. However, because of delays in the optical circuitry, at the receiver they may be skewed from CMD_CLK and from each other. Use the rising edge of each start bit to start the receivers' data recovery. Bit 15 is the MSB and bit 0 is the LSB of the 16-bit response. MSB is clocked out first.

Note that ssd_clk on the logic board is the same as s_cmd_clk (CMD_CLK) on the ssd module and ssd_dat0 / ssd_dat1 on the logic board are the same as s_dat0 /s_dat1 (DATA_OUT0 & DATA_OUT1) on the ssd module.

11.14 Other Optical Signals

M_RESET (SSDB Input): The master reset. Clears all flip-flops and registers in the SSDB chip. Most commands set bits in the SSDB chip, which directly drive other chips. M_RESET clears these bits and puts the chips into the corresponding mode. M_RESET does NOT reset the ASIC and the ADC chips and does NOT reset the ASIC and ADC power supplies. This signal is 1usec in width.

TRIG_ST (SSDB Output): One pulse for every suprathermal event detected by the ASIC.

TRIG_SW (SSDB Output): One pulse for every four (4) solar wind events detected by the ASIC.

TRIGGER_VE (SSDB Output): In energy mode, indicates an event detected by the ASIC. After TBD microseconds, the pixel, height, and multi-hit information can be retrieved by a read cycle.

SYNC (SSDB Input): In energy mode, generated in response to a TRIGGER_VE pulse, indicating the conversion and receipt of the event data. Primes the ASIC and the SSDB to accept another event. This signal is 1usec in width.

STIM (SSDB Input): In energy mode, and in conjunction with the DAC level, generates a simulated event.

11.15 Command Sequence for ASIC Register Loading

The 175 bit ASIC control register is loaded into the ASIC by 11 consecutive 16-bit commands. **No other command can be send in between these commands. They need to be in the correct order.** The command sequence must be preceded by the ASIC transmit command, setting the tx bit to 1 (50 01). These sequence of commands are stored as ACTEL 2 Register Sequences in the LOGIC Board EEPROM.

CMD_1,

Bias DACs: MSB sets the sign

TE# = Test enable channel # (1=enable, 0=disable)

15	14	13	12	11	10	9	8
0	Sha_bias (MSB)	Sha_bias	Sha_bias (LSB)	IFS (MSB)	IFS	IFS (LSB)	TE#15

7	6	5	4	3	2	1	0
TE#14	TE#13	TE#12	TE#11	TE#10	TE#09	TE#08	TE#07

CMD_2,

TE# = Test enable channel # (1=enable, 0=disable)

Thr# = Threshold DAC setting channel # (0-255)

15	14	13	12	11	10	9	8
TE#06	TE#05	TE#04	TE#03	TE#02	TE#01	TE#00	Thr#15 (MSB)

7	6	5	4	3	2	1	0
Thr#15	Thr#15	Thr#15	Thr#15	Thr#15	Thr#15	Thr#15 (LSB)	Thr#14 (MSB)

CMD_3,

Thr# = Threshold DAC setting channel # (0-255)

15	14	13	12	11	10	9	8
Thr#14	Thr#14	Thr#14	Thr#14	Thr#14	Thr#14	Thr#14 (LSB)	Thr#13 (MSB)

7	6	5	4	3	2	1	0
Thr#13	Thr#13	Thr#13	Thr#13	Thr#13	Thr#13	Thr#13 (LSB)	Thr#12 (MSB)

CMD_4,

Thr# = Threshold DAC setting channel # (0-255)

15	14	13	12	11	10	9	8
Thr#12	Thr#12	Thr#12	Thr#12	Thr#12	Thr#12	Thr#12 (LSB)	Thr#11 (MSB)

7	6	5	4	3	2	1	0
Thr#11	Thr#11	Thr#11	Thr#11	Thr#11	Thr#11	Thr#11 (LSB)	Thr#10 (MSB)

CMD_5,

Thr# = Threshold DAC setting channel # (0-255)

15	14	13	12	11	10	9	8
Thr#10	Thr#10	Thr#10	Thr#10	Thr#10	Thr#10	Thr#10 (LSB)	Thr#09 (MSB)

7	6	5	4	3	2	1	0
Thr#09	Thr#09	Thr#09	Thr#09	Thr#09	Thr#09	Thr#09 (LSB)	Thr#08 (MSB)

CMD_6,

Thr# = Threshold DAC setting channel # (0-255)

15	14	13	12	11	10	9	8
Thr#08	Thr#08	Thr#08	Thr#08	Thr#08	Thr#08	Thr#08 (LSB)	Thr#07 (MSB)

7	6	5	4	3	2	1	0
Thr#07	Thr#07	Thr#07	Thr#07	Thr#07	Thr#07	Thr#07 (LSB)	Thr#06 (MSB)

CMD_7,

Thr# = Threshold DAC setting channel # (0-255)

15	14	13	12	11	10	9	8
Thr#06	Thr#06	Thr#06	Thr#06	Thr#06	Thr#06	Thr#06 (LSB)	Thr#05 (MSB)

7	6	5	4	3	2	1	0
Thr#05	Thr#05	Thr#05	Thr#05	Thr#05	Thr#05	Thr#05 (LSB)	Thr#04 (MSB)

CMD_8,

Thr# = Threshold DAC setting channel # (0-255)

15	14	13	12	11	10	9	8
Thr#04	Thr#04	Thr#04	Thr#04	Thr#04	Thr#04	Thr#04 (LSB)	Thr#03 (MSB)

7	6	5	4	3	2	1	0
Thr#03	Thr#03	Thr#03	Thr#03	Thr#03	Thr#03	Thr#03 (LSB)	Thr#02 (MSB)

CMD_9,

Thr# = Threshold DAC setting channel # (0-255)

15	14	13	12	11	10	9	8
Thr#02	Thr#02	Thr#02	Thr#02	Thr#02	Thr#02	Thr#02 (LSB)	Thr#01 (MSB)

7	6	5	4	3	2	1	0
Thr#01	Thr#01	Thr#01	Thr#01	Thr#01	Thr#01	Thr#01 (LSB)	Thr#00 (MSB)

CMD_10,

Thr# = Threshold DAC setting channel # (0-255)
Dis# = Disable channel # (0=enable, 1=disable)

15	14	13	12	11	10	9	8
Thr#00	Thr#00	Thr#00	Thr#00	Thr#00	Thr#00	Thr#00 (LSB)	Dis#15

7	6	5	4	3	2	1	0
Dis#14	Dis#13	Dis#12	Dis#11	Dis#10	Dis#09	Dis#08	Dis#07

CMD_11,

Dis# = Disable channel # (0=enable, 1=disable)
Global control bits (0=off, 1=on)
Chip address: unused (0)
DLT mode:0=parallel triggers on

15	14	13	12	11	10	9	8
Dis#06	Dis#05	Dis#04	Dis#03	Dis#02	Dis#01	Dis#00	Shaper reset

7	6	5	4	3	2	1	0
Chip addr.	Chip addr.	Chip addr.	Chip addr.	Preamp reset	Test mode	Current comp.	DLT mode

12 Appendix A – Sample Command Sequences

12.1 Normal Power-Up Reset

1. Set registers in ACTEL 1 (use Logic Board immediate command)
    a. RESET_CTL to reset_hv, reset3 and reset2
    b. ACTEL_CTL
    c. LATCHUP_CTL
    d. TAC_PWR_CTL
2. Load DAC Board Control Registers to Logic Board RAM (for initial conditions)
3. Run mode 1 to read DAC registers from RAM and write to DAC Board one time
4. Clear Logic Board RAM
5. Clear Classifier RAM
6. Load Logic Board EEPROM (if not already done)
    a. ACTEL 2 Register Sequences 0..7
    b. SSD Register Sequences HK0..HK7
    c. DAC Tables
    d. Resistive Anode Table
7. Load Classifier EEPROM Tables (if not already done)
    a. Mass Tables
    b. M/Q Tables
    c. Bins Tables

12.2 Normal Mode

1. Set REG_SEQ to X”01” in ACTEL 1 (Register Sequence 1)
2. Run mode 2 (reads Register Sequence 1 from EEPROM and writes to ACTEL 2 one time)
3. Set immediate command registers
    a. REG_SEQ
    b. EVENT_CTL
    c. RLIM_CH
    d. RLIM_HI
    e. RLIM_LO
4. Run mode 3

12.3 ACTEL 2 Register Sequence 0 (retrace interval reset sequence)

1. force ssd_reset
2. program ssd ASIC
3. set SSD to energy mode
4. force tac0_reset
5. force tac2_reset
6. force send_sync to ssd

12.4 ACTEL 2 Register Sequence 1 (normal reset sequence)

1. sel0_disable to true
2. sel2_disable to true
3. force tac0_reset
4. force tac2_reset
5. send ssd setup commands
6. ssd_ctrl to force_reset
7. ssd_ctrl to force_sync
8. clear ssd_ctrl register

12.5 Classifier Tables

12.5.1 Format

n_cs1(0..3) correspond to 4 each, 128k x 8 EEPROMs

mass_tables_addr (16..0) = tof (8..0) & Ecompressed (7..0) tof(9)=0 for n_cs1(0) tof(9)=1 for n_cs1(1)

m/q_tables_addr (16..0) = tof(9..0) & swpe (6..0) n_cs1(2)

bins_tables_addr (16..0) = quadrant(1) & Nm(6..0) & Nq(7..0) & 0 [for bins_dat word (7..0)] n_cs1(3)
bins_tables_addr (16..0) = quadrant(1) & Nm(6..0) & Nq(7..0) & 1 [for bins_dat word (15..8)] n_cs1(3)

12.5.2 Sample Event #1

PHA Word

```
ESA Step (SWPE)        0101011
Defl Step (SWPD)       11100
Quadrant               00
SSD ID                 0000
SSD Energy (SSDE)      0101010010  -  compressed = 10110100
TOF                    0010111001
Position               010011
Section                01
Spare                  00

PHA Word to Classifier = 57C01522E534
PHA Word stored in RAM = 57C01522E535
```

Tables Accessed

```
EEPROM n_cs1(0)   addr = 0B9B4      data = 36   (NM)
EEPROM n_cs1(2)   addr = 05CAB      data = A2   (NQ)
EEPROM n_cs1(3)   addr = 06D44      data = BF
EEPROM n_cs1(3)   addr = 06D45      data = 57
```

RAM Addresses Incremented (16-bit counters)

271E / 271F	inc. by 1
2F78 / 2F79	inc. by 1
30FE / 30FF	inc. by 4

PHA Word Storage (48-bits) RAM Addresses

4800..4805

12.5.3 Sample Event #2

PHA Word

ESA Step (SWPE)	0101011
Defl Step (SWPD)	10101
Quadrant	01
SSD ID	1111
SSD Energy (SSDE)	0000000001 - compressed = 00000001
TOF	0101001000
Position	101110
Section	01
Spare	01

PHA Word to Classifier = 5757C01522E5
PHA Word stored in RAM = 5757C01522E5

Tables Accessed

EEPROM n_cs1(0)	addr = 14801	data = 36	(NM)
EEPROM n_cs1(2)	addr = 0A42B	data = A2	(NQ)
EEPROM n_cs1(3)	addr = 06D44	data = BF	
EEPROM n_cs1(3)	addr = 06D45	data = 57	

RAM Addresses Incremented (16-bit counters)

27FA / 27FB	inc. by 1
2F6A / 2F6B	inc. by 1
30FE / 30FF	inc. by 4

PHA Word Storage (48-bits) RAM Addresses

4800..4805

13 Appendix B – Other Notes

13.1 IDPU Notes

1. IDPU issue to s/c an emergency power-down command if we are out of limit on currents / voltages
2. IDPU… reset adc latchups for logic & dac boards (when auto reset is disabled)… check SSD status housekeeping word for a) adc latchup and b) ASIC latchup… run a set of commands to recover
3. ACTEL 1 to accept IDPU commands to "safe" PLASTIC when a power-down is imminent due to a s/c emergency (such as setting of the "off point bit" in the s/c status message)
4. Prior to a block read / write from util interface, the IDPU shall send a "Set Device Code and Block ID" command.
5. To prevent eeprom corruption, sys_reset should be active DURING power-down

13.2 Disable Plug

pac_dis, mcp_dis, ssd_dis, esa_dis, s_ch_dis, dfl_1_dis & dfl_2_dis are **GROUNDED** to disable the associated HV supply. When **GROUNDED**, the swp_limit line enables limiting of the s-channel, esa, deflection_1 and deflection_2 voltages to the values contained in the DAC Board Control Registers. When **GROUNDED**, the pac_mcp_limit line enables limiting of the pac and mcp voltages to the values contained in the DAC Board Control Registers. plug_id(3..0) indicates the "plug type" installed as indicated below.

plug_id(3..0)	color	function
0000	red	full disable
0001	blue	register limit
0010	yellow	hard limit*
1111 (open)	green	full enable

* hard limits: 100V for all sweeps & MCP supply – 5kV for PAC supply – no limit for SSD supply

		RED PLUG	**BLUE PLUG**	**YELLOW PLUG**	**GREEN PLUG**
		FULL DISABLE	**REGISTER LIMIT**	**HARD LIMIT**	**FULL ENABLE**
PIN	**SIGNAL NAME**	**CONNECTIONS**	**CONNECTIONS**	**CONNECTIONS**	**CONNECTIONS**
1	SWP_LIMIT	GND	GND	GND	NO CONNECTION
2	PAC_MCP_LIMIT	GND	GND	GND	NO CONNECTION
3	PAC_DIS	GND	NO CONNECTION	NO CONNECTION	NO CONNECTION
4	MCP_DIS	GND	NO CONNECTION	NO CONNECTION	NO CONNECTION
5	SSD_DIS	GND	NO CONNECTION	NO CONNECTION	NO CONNECTION
6	ESA_DIS	GND	NO CONNECTION	NO CONNECTION	NO CONNECTION
7	DFL_1_DIS	GND	NO CONNECTION	NO CONNECTION	NO CONNECTION
8	DFL_2_DIS	GND	NO CONNECTION	NO CONNECTION	NO CONNECTION
9	S_CH_DIS	GND	NO CONNECTION	NO CONNECTION	NO CONNECTION
10	PLUG_ID0	GND	NO CONNECTION	GND	NO CONNECTION
11	PLUG_ID1	GND	GND	NO CONNECTION	NO CONNECTION
12	PLUG_ID2	GND	GND	GND	NO CONNECTION
13	PLUG_ID3	GND	GND	GND	NO CONNECTION
14	GND	GND	GND	GND	NO CONNECTION
15	GND	GND	GND	GND	NO CONNECTION

14 Appendix C – Acronyms

14.1 General

A/D	: Analog to Digital Converter
ADC	: Analog to Digital Converter
AMU	: Atomic Mass Unit
ARR	: Authorization Return Receipt
AU	: Astronomical Unit
CHNL	: Channel
CIR	: Co-rotating Interaction Region
CLK	: Clock
CMD	: Command
CME	: Coronal Mass Ejection
CSA	: Charge Sensitive Amplifier
CTL	: Control
CVS	: Concurrent Version System
DAC	: Digital to Analog converter
DEFL	: Deflection
DFL	: Deflection
DIR	: Directive
DSCM	: Digitally Signed Command Message
-e	: Electron
E	: Energy
E/m	: Energy per Mass
E/Q	: Energy per Charge
E-BOX	: Electronics Box
EEPROM	: Electrically Erasable Programmable Read-Only Memory
EM	: Engineering Model
EMI	: Electro-Magnetic Interference
ESA	: Electrostatic Analyzer
ESEA	: Entrance System Energy Analyzer
ESP	: Energetic Storm Particle
ETU	: Engineering Test Unit
FEE	: Front End Electronics
FM	: Flight Model
FOP	: Flight Operations Plan
FOT	: Flight Operations Team
FOV	: Field of View
FPGA	: Field Programmable Gate Array
GND	: Ground
GSE	: Ground Support Equipment
GSFC	: Goddard Space Flight Center
HK	: Housekeeping
HV	: High Voltage
I&T	: Integration and Test
I/O	: Input / Output
ICD	: Interface Control Document
ICME	: Interplanetary Coronal Mass Ejection
IDPU	: Instrument Data Processing Unit
IISI	: Inter-Instrument Serial Interface
IPS	: Interplanetary Shock-Accelerated Particle
keV	: Kilo-Electron Volts
LVC	: Low Voltage Converter

M	: Mass
M/Q	: Mass per Charge
MCP	: Microchannel Plate
MOC	: Mission Operations Center
PAC	: Post Acceleration Voltage
PHA	: Pulse Height Analysis
PHA_PRI	: Pulse Height Analysis Priority Rates
PLASTIC	: Plasma And Suprathermal Ion Composition Investigation on STEREO
POC	: Payload Operations Center
POS	: Position
PTP	: Payload Telemetry Packet
PUI	: Pickup Ion
Q	: Charge
RAM	: Random Access Memory
S/C	: Spacecraft
SCM	: Supplemented Command Message
SEE	: Secondary Electron Emission
SEP	: Solar Energetic Particle
SM	: Structural Model
SPE	: Solar Particle Event
SPRS	: Software Problem Reporting System
SSD	: Solid-State Detector
SSDE	: Solid-State Detector Energy
ST	: Suprathermal
STEREO	: Solar-Terrestrial Relations Observatory
STP	: Supplemental Telemetry Packet
SW	: Solar Wind
SWPD	: Deflection Step Voltage
SWPE	: Electrostatic Analyzer Step Voltage
SWS	: Solar Wind Sector
TAC	: Time-to-Amplitude Converter
TCP	: Transmission Control Protocol
TINI	: Tini InterNet Interface
TLM	: Telemetry
TOF	: Time-of-Flight
UDP	: User Datagram Protocol
UTC	: Universal Time Code
UV	: Ultra-Violet
VCD	: Version Control Document
VDD	: Version Description Document
WAP	: Wide Angle Partition

14.2 Entrance System Electrodes

ESA-I	: Electrostatic Analyzer Inner
ESA-O	: Electrostatic Analyzer Outer
ESA-TC	: Electrostatic Analyzer Top-Cap
MA-L	: Main Channel Aperture Lower Duckbill
MA-U	: Main Channel Aperture Upper Duckbill
MC	: Main Channel
MG-L	: Main Channel Gate Lower
MG-U	: Main Channel Gate Upper
SAD	: S-Channel Aperture Diaphragm
SA-L	: S-Channel Aperture Lower Duckbill
SA-U	: S-Channel Aperture Upper Duckbill
SCI-L	: S-Channel Inner Lower
SCI-U	: S-Channel Inner Upper
SCO-L	: S-Channel Outer Lower
SCO-U	: S-Channel Outer Upper
WA-L	: Wide Angle Partition Lower
WA-U	: Wide Angle Partition Upper