SWEA/STE Interface FPGA (SIF)
Description/Operation

Revision 0.5
5 July 2002

D. Gordon
ELF Electronics

References.

1. SWEA/STE Interface FPGA (SIF) Specification, Version D, 2002-Apr-11
2. SWEA UCB to CESR ICD, Version E, 2002-Mar-29
3. STEREO IMPACT Intra-Instrument Serial Interface, Version 4, 2002-Jun-19

Revision History

Revision Number	Date	Change Summary
0.1	29 March 2002	Initial Draft
0.2	9 April 2002	Design Evolution & Corrections
0.3	22 April 2002	Design Evolution & Addition of Rate Counters
0.4	1 May 2002	Addition of STE Cover Programmable Delay & Misc Corrections
0.5	5 July 2002	Monitor Rates Sampled at 5.8ms; Design further detailed; Misc Corrections

Table of Contents

1.0 Introduction

The SIF (SWEA/STE Interface FPGA), logic contained in an Actel 54SX32S, handles the overall control and communication requirements for two types of instruments: the SWEA/STE-D and the STE-U. It incorporates components required by both instruments; however certain of the modules can be disabled via command, enabling the same design to be used for both systems.

An external reset signal (active low) is controlled by an RC with a time constant of approximately TBD seconds.

Figure 1 on page 4 shows an overview of the main subsystems comprising the SIF.

NOTE: Figure 1 is the actual FPGA schematic, which shows the general subsystem breakdown. I have left the internal connects unhooked until the subsystems have been designed. However, the external I/O are shown as in/out arrows. The current I/O estimated pin usage is 155 pins (the limit is 173 pins for the 54SX32S in the 208 pin package).

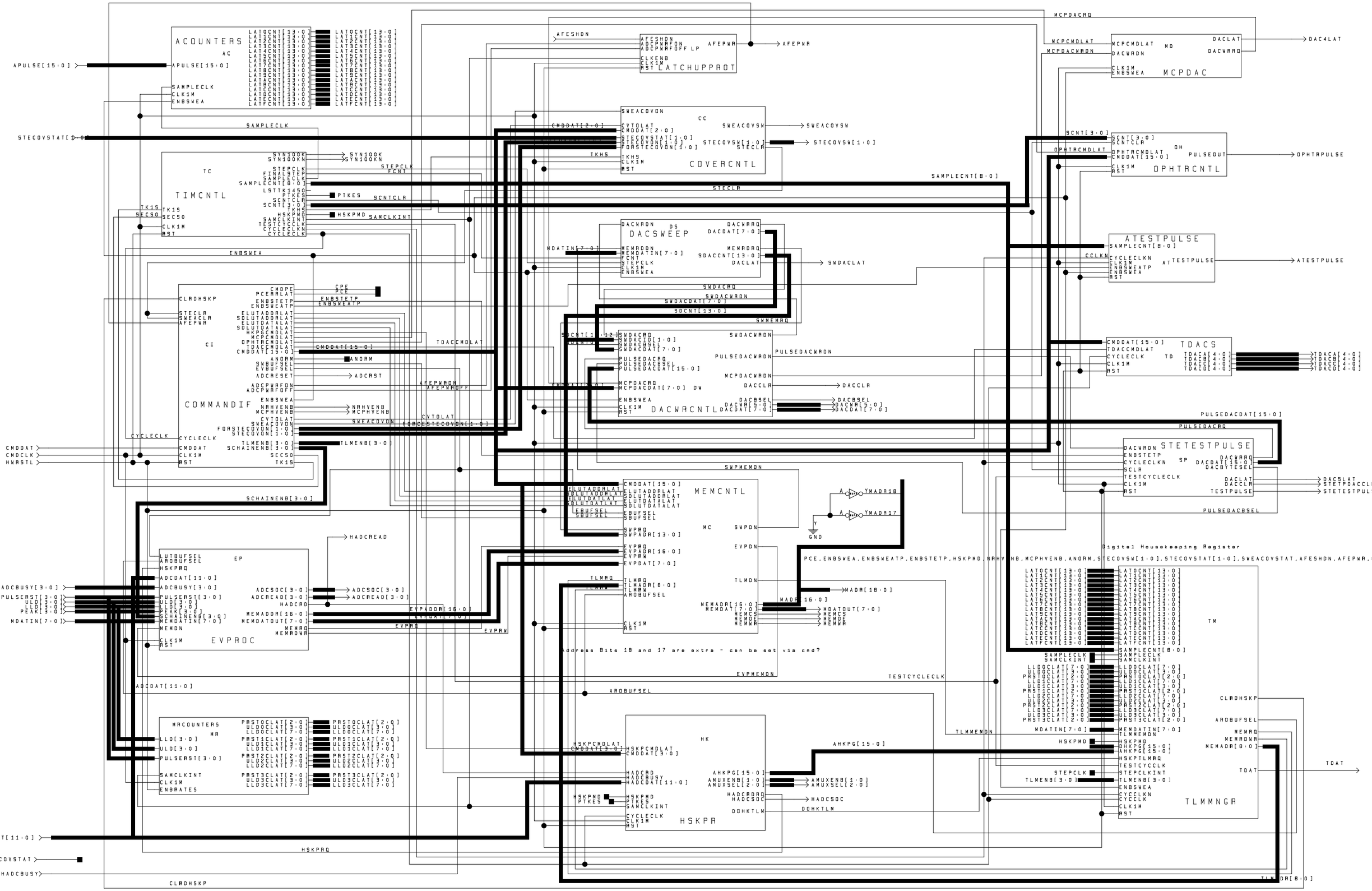


Figure 1: SIF FPGA – Overall Block Diagram

2.0 Subsystem Descriptions

2.1 Timing Control

The timing (TIMCNTL) module receives the 1MHz Spacecraft clock (CLK1M), the 1 Second Tick (TK1S) and the seconds count bit 0. It creates the following signals/strobes and fowards them to the various SIF subsystems:

CYCLECLK: Occurs every 2 seconds, as parsed from the "F0" command data (every "even" instance of TK1S). Only SECS[0] is used by the TIMCNTL module.

STEPCLK: STEPCLK divides the 2 second cycle clock into 1345 intervals as follows: at the beginning of a CYCLECLK period 1344 1.45 ms intervals are generated (totally 1948.8ms). The last interval is 51.2 ms.

SAMPLECLK and SAMPLECNT[8:0]: SAMPLECLK occurs every 4th STEPCLK during the 1344 1.45ms interval period, restarting at the next CYCLECLK. SAMPLECNT, a 9-bit down-counter, is loaded with the maximum count (337 decimal, 151 hex) at the occurrence of each CYCLECLK. The counter decrements every SAMPLECLK, reaching and holding the final count of 1 during the 51.2ms interval.

TESTCYCLECLK: Occurs every ten seconds. The TESTCYCLECLK hardware does not use the F0 data directly to determine timing, but counts CYCLECLKs, "ticking" every 5th CYCLECLK (different than as described in Reference 1).

HSKPMD: Toggles between 0 and 1 every CYCLECLK.

TK8HZ: This tick, which occurs 8 times per second (or 16 times per CYCLECLK), is used to time housekeeping data acquisition and telemetry.

TK2HZ: This tick, which occurs 2 times per second, is used to time the cover actuator "ON" durations.

SYN100K and SYN100KN: 100KHz Reference Supply (and an inverse version) which is forwarded to the SWEA-HVPS. The outputs are disabled (held at zero) when ENBSWEA is deasserted.

2.2 Command Interface

The command interface module performs serial to parallel conversion, receiving commands in the 24-bit format describes in Reference 3. If a parity error is detected, the command is ignored and an error indicator set (CMDPE). CMDPE is sent down as part of the digital housekeeping word (see Section 2.15, "Housekeeping Control," on page 15) and is reset following transmission.

The command interface module directly drives the simple control signals, either to internal SIF subsystems or off-chip to the various SWEA/STE-D or STE-U subsystems. For these outputs, any required staging is performed in the command interface module itself Other commands are forwarded to the SIF subsystems using the latched command data bus and subsystem select strobes.

The Command ID start at E0 hex.

The commands are:

COMMAND NAME	ID (Hex)	DESCRIPTION	Staged - When Effective
SRAM Buffer Select	E0	Sets Buffer Use D[1]: SWEEP Table Buffer Select - the value of this bit determines which buffer is the read buffer. D[0]: Energy Table Buffer Select - the value of this bit determines which buffer is the read buffer.	SWEEP LUT Updates at the beginning of the Next "GAP" Energy LUT updates at the Next CYCLECLK
MCP DAC Load	E1	Loads MCP DAC (8 bit value) D[7:0]: DACValue[15:8] (DACValue[7:0] remain cleared)	Immediate
SWEA/STE Controls/Enables	E2	Sets/Clears Various Controls D[15:12]: Telemetry Subsystem Enbs D15: Enable ACOUNTER Messages D14: Enable HSKPG Messages D13: Enable STE-PHA Messages D12: Enable RATES Messages D[11]: AFEPWR Force On D[10]: AFEPWR Force Off D[9]: STE Test Pulser Enable D[8]: SWEA Test Pulser Enable D[7:4]: Shaper Chain Enable[3:0] D[3]: ADCs Reset D[2]: Enable SWEA (enables the following SWEA/STE Subsystems: DAC SWEEP and SWEA Counter Readout) All bits default to disabled at reset: This results in all bits except the ADCs Reset defaulting to zero. The ADCs Reset line, which is active high, defaults to one; clearing it enables ADC operation.	Test Pulser, Telemetry Subsystem and Shaper Chain Enables take effect at the next CYCLK-CLK. All others are effective immediately.

COMMAND NAME	ID (Hex)	DESCRIPTION	Staged - When Effective
Protected Command Execute	E3	Sets/Clears Protected Signals D[15]: NR HV Off D[14]: MCP HV Off D[10]: SWEA Cover Actuator Off D[8]: FORCE STE Cover Actuator D[7]: NR HV On D[6]: MCP HV On D[2]: SWEA Cover Actuator ON D[1:0]: FORCE STE Cover Actuator ON D1: Set Force Cover In (Close) D0: Set Force Cover Out (Open) NOTE: Only one of the "On" Command bits should be set. This restriction does not apply to "Off" bits. All signals default to 0 (disabled) at reset	Immediate (Setting occurs only if armed; clearing does not require arming. Both setting and clearing occur immediately.) When STE Cover Actuator In or Out commanded without an arm, the command executes in "non-forced" mode.
Operational Heater Control	E4	Sets PWM Duty Cycle to OpHeater D[3:0]: Pulse Width Value 0-> 0% Duty Cycle (Pulse held at logic zero) 1-> 10% Duty Cycle (on 1µs, off 9µs) ... 9-> 90% Duty Cycle (on 9µs, off 1µs) 10-> 100% Duty Cycle (Pulse held at logic one)	Immediate
Threshold DAC Load	E5	Sets One of Four Threshold DACs D[7:6]: DAC Select D[4:0]: DAC Data DACs are initialized to all zeros at reset.	Next CYCLECLK (TBR: depends on how full FPGA becomes)
ARM Protected Command	E6	ARMs the following commands D7: 1-> ARM NR HV ON D6: 1-> ARM MCP HV ON D2: 1-> ARM SWEA Cover Actuator D1: 1-> ARM Force STE Cover In (Close) D0: 1-> ARM Force STE Cover Out (Open) NOTE: Only one of bits D7,D6,D2,D1 & D0 should be set; if more than one are set, the arm does not occur.	Immediate (Enables future commands): expires in 14-16 seconds
SWEEP Housekeeping Channel Select	E7	Selects Active Housekeeping Channel for Sweep Housekeeping Mode D[3:0]: SWEEP Housekeeping Select	Next CYCLECLK

COMMAND NAME	ID (Hex)	DESCRIPTION	Staged - When Effective
Sweep LUT Address Pointer Write	E8	D[12:1] - Word Address[12:1] Address pointer for IDPU writes to the Sweep Look-up Table. The address is incremented by one following each Sweep LUT Data Write	Immediate
Sweep LUT Data Word Write	E9	D[15:0] - Data to be written to current LUT location, determined by both the Sweep Table Buffer Select and the Sweep LUT Address Pointer	Immediate
Energy LUT Address Pointer Write	EA	D[14:1] - Word Address[14:1] Address pointer for IDPU writes to the Energy Look-up Table. The address is incremented by one following each Energy LUT Data Write	Immediate
Energy LUT Data Word Write	EB	D[15:0] - Data to be written to current LUT location, determined by both the Energy Table Buffer Select and the Energy LUT Address Pointer	Immediate
STE Cover Time-out Delay	EC	Sets STE Door Timeout D[2:0] - 8 settings, in 0.5 second increments from 0.5 seconds at the minimum setting of zero to 4 seconds at the maximum setting of seven.	Immediate

2.2.1 Protected Commands

The following commands are protected:

- Enabling the High Voltage Power Supplies (NR HV ON and MCP HV ON)
- Activating the SWEA Cover Actuator (SWEA Cover Actuator ON)
- Activating the FORCE STE Cover Actuators (Force STE Cover Actuator In ON & Force STE Cover Actuator Out ON)

A protected command is executed as follows:

1. Write the appropriate ARM to the “ARM Protected Commands (ID E6). The ARM command should only have one signal set; if more than one signal are set, it is ignored. The ARM expires in 14-16 seconds. During the period when a command is ARMed, all other ARM command are ignored.
2. Issue the corresponding execute command “Protected Command Execute (ID E3)” less than 14 seconds after the ARM command dispatch. The execute bit, must match the ARM bit, or the command is ignored, and the ARM is deactivated. Additionally, if more than one “ON” bit is set in the execute command, the command is ignored and the ARM is deactivated.

Note: Like “Radio Buttons”, the “Protected Command Execute” only sets or clears when the applicable bits in the command are set. Clearing signals does not require arming, and can be

issued with multiple clears in one execute. Clearing always overrides setting, if they occur simultaneously in the same execute command.

A Protected Command Subsystem error is registered in the following cases:

- Timeout Error: the arming occurs and expires with no execute command received.
- Invalid Execute: the arming occurs and is followed by an invalid execute (more than one ON bit set, or the incorrect ON bit set).
- Illegal Arm: more than one signal armed, or an all zero arm received when the Protected Command Subsystem is in the IDLE state.
- Unarmed Execute: Execute occurs (Bits D7 or D6 or D2 set) when the Protected Command Subsystem is unarmed. (NOTE: If bits D0 or D1 are set in an execute command, the command executes, but in "nonForced" mode.)

The error, which is available as digital housekeeping, is cleared once it has been read-out.

A "disarm" capability is available. The user can undo an arm by sending another arm command with a data value of all zeros. (NOTE: if a "disarm" is received when the system is not currently armed, an error will be flagged.)

2.3 Latchup Protect

The Latchup Protect (LATCHUPPROT) module manages AFEPWR based on the states of an external overcurrent detect: AFESHDN and two commanded controls. The two commanded parameters, which both default to zero, are AFEPWR Force Off (AFEPWROFF) and AFEPWR Force On (AFEPWRON).

The following truth-table describes the AFEPWR control:

AFESHDN	AFEPWROFF	AFEPWRON	AFEPWR
0	0	0	Previous State
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

AFEPWR is active high: a logic 1 indicates that the supplies are turned on, and the system is active. AFEPWR is jammed low by reset, but is allowed to assert following reset if AFESHDN is low. The “Force” signals, AFEPWRON and AFEPWROFF, are set/cleared via the command interface.

An assertion of AFESHDN (when AFEPWRON is deasserted) clears AFEPWR. This clear is latched; AFEPWR remains off even when AFESHDN deasserts. It can be turned on again either by resetting the system or by commanding AFEPWRON.

In case of noise or slow transitions times, AFESHDN is sampled as a slower (TBR) frequency than the 1MHz system clock. SAMCLKINT is used (approximately 1.45ms) for this purpose.

NOTE: to exit “Force” mode, the AFEPWRON and AFEPWROFF signals should be explicitly commanded to zero directly following the force.

The following SIF modules are held in reset when AFEPWR is deasserted: ACOUNTERS, DACSWEEP, DACWRCNTL, EVPROC, MRCOUNTERS, TDACS, STETESTPULSE, ATESTPULSE, and MCPDAC. Most of the HSKPR module is reset. The Housekeeping ADC controller is held inactive, but the request for housekeeping telemetry is kept active, asserting every 1/8 second. This allows for transmission of digital housekeeping during AFESHDN periods.

Both AFEPWR and AFESHDN are available as status via the digital housekeeping register.

AFEPWR, which controls the external power switches for the analog supplies, is also fed to SIF subsystems so that the analog circuitry inputs can be suppressed when the power is off. The HV Enable and ADCRESET lines are cleared when AFEPWR goes off, and remain cleared until commanded on.

TBR: The SIF specification states that the signals to the analog circuitry should be held low. Would it be acceptable to just put these signals into high-impedance during AFESHDN? This would be simpler and use less modules.

The following outputs are forced into a high-impedance (TBR) state when AFEPWR is deasserted:

Signal/Bus Name	Description
ADCSOC[3:0]	Shaper Chain ADC Start of Conversion
ADCREAD[3:0]	Shaper Chain ADC Read
AMUXSEL[2:0]	Housekeeping Analog Mux Selects
AMUXENB[1:0]	Housekeeping Analog Mux Enables
HADCSOC	Housekeeping ADC Start of Conversion
HADREAD	Housekeeping ADC Read

TBR: What is the desired state of signals to the DACs when AFEPWR is low? It makes sense to force the DACCLR lines asserted (low), but what about the other signals? These include the DACLATCH, DACWRITE, DACBYTESEL and the DACDATA bus. Since the DACs take a digital supply, should these signals remain driven, with the strobes deasserted (high)? The DAC data bus is probably a don't care.

TBS: What should be done with the 100K Synch Pulses when AFEPWR is off? Are there any other signals which should be cleared or tristated? Threshold DACs, STETESTPULSE, etc. Some of the signals default to high when deasserted -- should they be cleared, tristated or driven during shutdown periods? It might be worthwhile to go through the entire pin list, and note how each output should behave when AFEPWR=0.

2.4 DAC Write Control

The DAC Write Control (DACWRCNTL) module receives write request (8-bit data format) to the six DACs that share the common DAC bus. It handles requests from the three client subsystems on a fixed (1=MCP, 2=TESTPULSER, 3=SWEEPDACS) priority basis, driving the appropriate address, byte-select, data and write strobe onto the external DAC control bus. The DAC control subsystems are responsible for generating their own DAC latch pulses and sequencing 16-bit writes if necessary. The SweepDAC subsystem (see below) controls/sequences the four sweep DACs, requesting the DAC Write bus eight times every STEPCLK.

The five SWEA DACs share a common CLEAR signal, which is asserted during power-on reset, when the SWEA is disabled, and when AFEPWR is off.

The STE Test Pulser Module drives its own CLEAR signal, which is used by the STE Test Pulser Control Logic (and is also asserted during reset and AFEPWR off).

2.5 SWEA/STE DAC Sweep Control

The Sweep Control subsystem (DACSWEEP) is responsible for loading four DACs at the STEPCLK frequency, and cycling through a preloaded pattern every CYCLECLK. The IDPU loads the SWEEP LUT portion of memory and initiates LUT buffer swapping via the command interface.

For every 8-bit value loaded into the DACs, a memory read must be performed. DACSWEEP tracks the current address and requests SRAM cycles via the memory arbiter. When the read has been granted, DACSWEEP requests the DAC bus via the DAC controller/arbiter, continuing on until all the values for the next step have been written. The next values remain in the DACs' holding registers until DACSWEEP issues the SWDACLAT strobe, coincident with the next STEPCLK.

NOTE: The uppermost memory address bits [13:12] are driven by DACID during the memory reads, so that each DAC is assigned its own quadrant in the Sweep memory segment.

2.6 MCP DAC Control

The MCP DAC Control (MCPDAC) module receives a signal from the Command I/F following the reception of an MCP DAC Load Command. It then requests the DAC bus and writes the 8-bit value into the upper byte of the MCP DAC. The value is immediately written/latched into the DAC.

The lower byte, set to zero by the DACCLEAR signal (asserted at RESET), is never changed.

2.7 Operational Heater Control

The Operational Heater Control (OPHTRCNTL) module receives a signal from the Command I/F following the reception of an Operational Header Control Load Command. It then latches the control data as input to the Pulse Width Modulation Circuitry. The control data (PWM_VALUE) occupies the lower nibble of the 16-bit command word, and is defined as follows: the output pulse is high for a period of PWM_VALUE x 1μs. The period of the PWM is 10 μs.

The PWM_VALUE, which is cleared at reset, varies from 0-10 in order to cover all possibilities, all off to all on. If PWM_VALUE is set to an illegal number, OPHTRCNTL substitutes a value of zero.

2.8 SWEA/STE (Anode) Test Pulser

The Anode Test Pulser (ATESTPULSE) module generates a variable frequency test pulse, used to stimulate the SWEA anodes during testing. It is held in a reset state (at which time the output is a logic low) when the SWEA is disabled or when the SWEA Test Pulser is disabled. The test pulse is activated via the Controls/Enable command.

When enabled, the ATESTPULSE frequency is stepped up each SAMPLECLK and returns to the start frequency (approximately 3KHz) at each CYCLECLK.

The Anode Test pulser generates a pulse train based on CLK1M divided by SAMPLECNT[8:0]. (SAMPLECNT[8:0] is decremented each SAMPLECLK and set to the starting number of 337 each CYCLECLK). The pulse, which is active high, is one microsecond in duration for all intervals. During the 51.2ms "gap", the test pulse reaches its maximum frequency of 500KHz.

2.9 Threshold DAC Registers

The Threshold DAC module contains four 5-bit registers which directly drive four on-board DACs. They are programmable via the threshold DAC Load Command, with bits[7:6] selecting the DAC to be loaded, and bits[4:0] containing the DAC data to be loaded.

The DAC data lines are staged; any change registers at the next CYCLECLK.

TDAC registers all default to 10 hex at reset (the MSB is set and all the other bits are cleared).

2.10 STE Test Pulser

The STE Test Pulser (STETESTPULSE) module both programs an external DAC and generates a test pulse. It is enabled via command: (Controls/Enables Command ID E2), STE Test Pulser Enable.

When enabled, the DAC ramps up from zero to its maximum count (65536) incrementing by one every 100µs. The Ramp begins upon detection of a TESTCYCLECLK. The pulse, generated before each DAC increment, is active low and lasts 10µs. The DAC data, updated 1µs after the rising edge of the test pulse, is latched in via a dedicated DACLD pin. When the DAC count/ value reaches its maximum, the DACs are cleared and the pulses are stopped. The DACs remain at zero until the next TESTCYCLECLK, at which time the DAC ramping and test pulses are restarted. The logic uses the STE Test Pulser DAC CLR signal during the periods when the Pulser is IDLE. This signal, therefore, has been separated from the general DACCLR signal.

All DAC writes must be arbitrated via the DAC controller/arbiter, as for the Sweep and MCP DACs.

2.11 Cover Actuator Control

The control logic in this module (COVERCNTL) provides switch control for the cover actuators.
TBS: Polarity of actuator and status signals

2.11.1 SWEA Cover

There is one 1-time opening cover on the SWEA. This switch shall be activated via the IDPU command I/F (Actuator Enables - Command E3).

The SWEA cover control is a “protected command”, requiring a sequence of two commands in order to take effect. (See Section 2.2.1, “Protected Commands,” on page 8.)

The activator control bit automatically clears two seconds after it is set via the arm/execute sequence. It can also be cleared explicitly via the command interface. There is no arm required for clearing the activator control.

2.11.2 STE Cover

STE has a reclosable cover. There are two switch controls to the STE and two status signals from the STE. The controls are activated via the command I/F (Actuator Enables): one commands the cover open and the other commands the cover shut. When one of these commands is received, power is supplied to the appropriate actuator, until the status signal feedback indicates that the cover has reached its desired position. If at any time, the OPEN or CLOSE commands are received with the data bit set to zero, the actuator control will be deasserted immediately, regardless of the state of the status feedback. If both a set OPEN and CLOSE command are asserted simultaneously, neither command shall register.

There is also a “force” command available for each STE actuator. This allows power to be supplied to the appropriate actuator regardless of the state of the feedback switch. The force commands are protected via an “arm-execute” command sequence (one arm per actuator). (See Section 2.2.1, “Protected Commands,” on page 8.)

The activator enables, whether in “force” mode or not, automatically clear within a programmable delay period after assertion. The Programmable Delay is set via Command STE Cover Timeout Delay (ID EC) to a value ranging from 0.5 to 4 seconds.

2.12 Anode Counters

The Anode Counter (ACOUNTERS) module contains 16 counters, each 14-bits, which register pulses generated by the SWEA anodes.

Anode Pulses are 250-300ns, positive-going pulses; rising edges are counted. As the fastest clock used by the SIF is 1MHz, each anode pulses must be fed directly to the clock of its counter. (So if the input is noisy, the counter will increment. Also, runt pulses may be registered as counts.)

Each SAMPLECLK, the count values are transferred to holding registers, and the counters are cleared. The Telemetry Manager subsystem (see Section 2.17 on page 18) is responsible for reading out the holding registers (and the sweep housekeeping value) following each SAMPLECLK. There may be a few microseconds of downtime during the count latching/clearing every SAMPLECLK.

This module is held in a reset state when ENBSWEA is deasserted.

2.13 Event Processing

The event processing (EVPROC) subsystem handles the outputs from the STE shaper circuits. At initialization, the ADCs should be activated; each analog chain can be enabled/disabled (via the Controls/Enables command).

The four analog chains each drive a separate set of control/pulse signals, but share a common 12-bit ADC data bus. (This ADC data bus is also shared with the 12-bit analog housekeeping ADC data bus. Arbitration, between the four analog chains as well as the HSKPR module, is performed by the Event Processing subsystem.)

A combination of control signals (as described in Reference 1) determine whether an event should be processed. If the criteria are met, the SIF issues an ADCSOC pulse to the active chain. ADCSOC is initiated via combinational logic only, in order to insure a small time delay. The SIF then waits for the busy line to transition high (25ns MAX) and deasserts ADCSOC. When the busy line returns to low, it is queued up for readout and processing.

After conversion, a request is made to the Event Arbiter/Processor subsystem. Each chain is allowed to convert when another event from a different chain is being processed. However, any further events in the same chain that arrive while an event is still pending are dropped.

The arbiter is non-preferential. If all four detector chains are continuously receiving events, each will receive one-fourth of the event processing subsystem bandwidth. However, if only one detector is active, this chain will receive the total available bandwidth. An enable is provided for each analog chain, so that a noisy channel can be disabled.

Following readout, the EVPROC module uses the ADC Data along with the Detector ID as an index to a lookup table, located in external RAM. The 8-bit value returned is used as an index to a 16-bit counter, also located in external RAM. EVPROC reads the counter value, increments it, and writes it back into memory. Only the modified counter byte needs to be read/written back into memory. This eliminates two memory cycles for most cases.

2.14 Rate Counters

The Rate Counter (MRCOUNTERS) Module contains 12 counter/accumulators. Three signals from each PHA chain are used to clock the counters, which are latched into holding registers and cleared every occurrence of a "free-running" version of SAMPLECLK (does not stop during the "GAP"), approximately once every 5.8ms. Each accumulation interval, the Telemetry Manager reads the latched values and assembles them into a message. If the counters reach their maximum during any time during the accumulation, they lock this value in until cleared.

For PHA chain, the following signal assertions are counted:

Counter Clock	Number of Counter Bits
LLD	8
ULD	4
PULSERESET	3

The counting and latching is not synchronized to the 1MHz clock. Each signal rising edge clocks a ripple counter, which freezes at the final count. The counters are latched and reset every (~5.8ms) interval (during which time the subsystem will incur a few microseconds of dead-time).

2.15 Housekeeping Control

At initialization, the Housekeeping ADC must be activated (via the Controls/Enables command). (The Housekeeping ADC shares a command bus and reset line with the Detector ADCs.)

The Housekeeper (HSKPR) module receive a status signal (HSKPMD) and a timing signal from the TIMCNTL subsystem. HSKPMD, which toggles every CYCLECLOCK, selects either CYCLING or SWEEP Mode. The timing signal occurs 8 times/second, or 16 times every CYCLECLK. (NOTE: The 8Hz Housekeeping frequency is slightly uneven: some intervals may be 122ms, while others might be 128ms.)

When in CYCLING mode: at CYCLECLK the AMUX channel is set to zero. At the first DOHSKP, HSKPR performs an ADC conversion and latches the result. (NOTE: All ADC Reads must be negotiated through the Event Processing subsystem ADC bus arbiter.) When the readout has completed, HSKPR informs TLMMNGR, which constructs a message containing the 12-bit ADC result in bits 11:0 and the 4-bit Analog Mux channel in bits 15:12, followed by the 16-bit digital housekeeping register. After latching the current result, HSKPR increments the AMUX channel in preparation for the next conversion, which occurs at the next DOHSKP tick. This process continues until the 16-values have been scanned through in one CYCLECLK period.

When in SWEEP mode: The Housekeeper maintains a 4-bit register, loaded via command, which sets the analog channel for ADC conversions. At the next CYCLECLK when HSKPMD=SWEEP, the AMUX channel is set to the commanded value. Conversions are then performed every SAMPLECLK. TLMMNGR still reads the result and constructs messages (the

last ADC conversion followed by the digital housekeeping register) eight times per second; however the AMUX channel is not incremented. TLMMNGR also appends the ADC result onto the SWEEP messages.

The digital housekeeping register provides status from the various digital and external subsystems. The register is defined as follows:

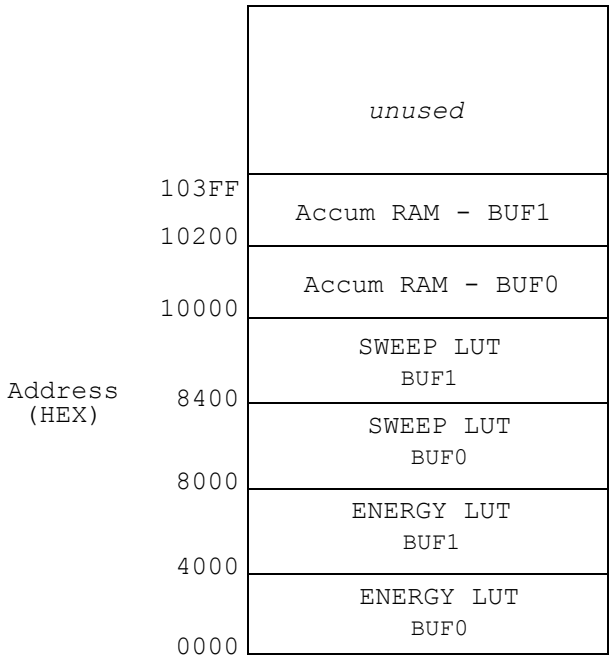
Bit Position(s)	Signal Name	Definition
15	PCE	Protected Command Subsystem Error Detect (occurred in previous 1/8 second interval)
14	ENBSWEA	SWEA Subsystem Enable
13	ENBSWEATP	SWEA Test Pulser Enable
12	ENBSTETP	STE Test Pulser Enable
11	HSKPMD	Current Housekeeping Mode
10	NRHVENB	NR High Voltage Enable
9	MCPHVENB	MCP High Voltage Enable
8	ANORM	OR of any unusual setting or conditions
7:6	STECOVSW[1:0]	STE Cover Switch Control Outputs
5:4	STECOVSTAT[1:0]	STE Cover Status
3	SWEACOVSTAT	SWEA Cover Status
2	AFESHDN	AFE Shutdown
1	AFEPWR	AFE Power
0	CPE	Command Parity Error in previous 1/8 second interval

The ANORM signal indicates a divergence from normal operational mode. It ORs the assertion of the following conditions or signals: AFEPWRON, AFEPWROFF, all of the Telemetry Subsystem Disables, Protected Command Subsystem Currently Armed, and all of the Shaper Chain Disables.

2.16 Memory Cycle Control

An external 512Kx8 SRAM holds the various data buffers used by the SIF: the Energy LUT, the Accumulator RAM, and the Sweep LUT. All of these SRAM Areas are double buffered. The LUT buffer select is determined by IDPU command; the Accumulator RAM buffer select is swapped regularly as messages are read-out.

Memory allocation is shown below:



TBD: We could use bits 18:17 of the SRAM as a general RAM buffer select, which might be useful if a memory cell develops a problem during the life of the mission. The general select could be set as the LUT Buffers are, via the IDPU Command I/F. This feature would be most valuable if memory dump command is also offered, so that the different memory sections could be tested/verified via the command and telemetry interfaces. Addition of these features will be considered if FPGA space appears to be available. (It's very tight now, July 2002, but we may be able to spare one latch for at least allowing buffer swap.)

The Memory Controller (MEMCNTL) manages the cycles to the 8-bit SRAM, selecting between the following memory clients: DACSWEEP, EVPROC, TLMMNGR, and COMMANDIF.

The DACSWEEP module must read out of SRAM in order to update the Sweep DACs. It requires eight byte reads every STEPCLK.

EVPROC requires one byte read (counter address), followed by one byte read and one byte write (counter LSB increment), the possibly a second byte read/byte write (counter MSB increment).

TLMMNGR requires 512 byte reads followed by 512 byte writes (clears) every CYCLECLK, for the accumulator messages.

COMMANDIF requires one word write each time a LUT DATA WRITE command is received. Both LUT address registers and write control subsystems are included in the MEMCNTL module.

MEMCNTL coordinates all the SRAM buffer controls, based on input from the Command I/F and the various timing signals. It selects the client for the current cycle (delineated by CLK1M), drives the selected address, data and control strobes onto the memory bus, and informs the client that the cycle has been completed. The memory bus arbitration scheme is fixed with the following priority assigned: 1: TLM, 2: CMD, 3: SWP, 4: EVPROC. This assures that each client is

able to obtain the bus with a small enough latency to assure continuous operation. For example, when TLMMNGR is reading out the accumulator RAM, two byte reads must occur at least every 16μs in order to keep up with the message generation. Bursts are spaced out at the subsystem level so that none of the higher priority subsystems will ever monopolize the bus.

2.17 Telemetry Manager

The telemetry manager (TLMMNGR) module constructs, formats and serializes messages in accordance with the requirements outlined in Reference 3. The following message types are constructed by TLMMNGR:

Message Type	ID	Length	Frequency	Description
Anode Counters without HSKPG	30	18	336 msg/sec	The 16 latched counter values accumulated by the SWEA Anodes every SAMPLECLK, followed by the SAMPLECNT value (bits 8:0) for the accumulating interval.
Anode Counters with HSKPG	31	19	336 msg/sec	The 16 latched counter values accumulated by the SWEA Anodes every SAMPLECLK, followed by the SAMPLECNT value (bits 8:0) for the accumulating interval, followed by the SWEEP Housekeeping Value (bits 11:0) and analog mux channel (bits 15:12).
Energy Bin Counters TESTCYCLKCLK =0	32	257	one msg/ CYCLKCLK	The 256 event counters used for STE PHA.
Energy Bin Counters TESTCYCLKCLK =1	33			
Housekeeping	34	3	8 msg/sec	The most recently converted HSKPG-ADC value, followed by one Digital Housekeeping value
Rates	35	5	1 msg/ ~5.8msec	The 12 monitor rate counter values accumulated over the CYCLECLK interval There are four data words, transmitted from Chains 0, 1, 2, 3, arranged as follows: Bits[15:8]: LLDLatchedCount[7:0] Bit 7: Always Zero Bits[6:4]: PulseResetLatchedCount[2:0] Bits[3:0]: ULDLatchedCount[3:0]

The length is the number of telemetry message words, including the first (header) word. (The header word contains the message ID in the upper 6 bits and LENGTH-2 in the lower 10 bits, as specified in Reference 3.) TLMMNGR handles the timing for Anode Counter readout, starting the message construction following every SAMPLECLK, and extracting the necessary data from the ACOUNTER (and HSKPR if in SWEEP Housekeeping mode) modules.

TLMMNGR constructs and transmits the housekeeping message upon receipt of the signal HKPGDN from HSKPR, generated following each ADC conversion, 8 times/second.

The Energy Bin Counters: While TLMMNGR is constructing Energy Bin Counter messages, it requests RAM via MEMCNTL for two byte reads per counter readout. The Accumulator Buffer is cleared as a block action after the message is constructed, but preceding the Accumulator

Buffer Swap. During the accumulator clear phase, each memory request is spaced by an interval of one step clock. This prevents TLMMNGR from accessing memory too frequently, which could cause the other subsystems to slow down.

The accumulator RAM is not cleared until one message buffer has been read out. This invalidates the first accumulation following system start-up.

Each telemetry message type has its own enable, which is set via the command I/F. TBR: Telemetry enables just depend on the command I/F currently. It may be better to gate them with the subsystem resets as well. Since the messages are timed by the clock in most cases, rather than subsystem requests, we would get garbage messages when AFEPWR is off or ENBSWEA is off unless the user explicitly disables telemetry for each subsystem.