Nuclear Physics Division
Fast Electronics Group

Description and Technical Information
for the
VXS Signal Distribution Module

Summer/Fall 2010

Fast Electronics Group
Jefferson Lab

Table of Contents

1.0 Introduction

The VME Switched Serial (VXS) base standard defines the physical arrangement of the VXS payload slots and the switch slot as a redundant star (see figure 1). Each payload slots has 8 differential pairs and two single-ended connections to the "A" and "B" switch slots. The Signal Distribution Board (SD) module occupies the "B" switch card slot as specified in VITA 41.The main purpose of this module is to distribute signals received from payload slot 18 (Trigger Interface board) of a VXS crate to 16 other payload slots (ADC boards).

The Signal Distribution (SD) module distributes the 4 differential pair LVPECL signals from payload slot 18 to 16 VXS payload slots within the crate. This is done using the high speed point to point connections from the switch slot to each payload slot. The four signals distributed are length matched to minimize output jitter seen on all the payload slots. Three of the four remaining pairs are LVDS signals routed from the each payload module to the FPGA on the SD module. The last pair is an LVDS signal routed from the FPGA on the SD module to each payload module. Each of the 16 payload modules has a single-ended signal to the SD module and one from the SD module back to the payload module.

A Jefferson Lab VXS crate will be configured to use up to eighteen (18) payload slots. Payload slot 17 is reserved for the CPU and slot 18 is used for the Trigger interface board.
The distribution of the remaining 16 slots can be such that:

a) Each half of the crate can hold eight: FADC-250, ADC125 or F1TDC boards.
b) All 16 payload slots can be populated with boards of the same type.

The switch slots are connected to each payload slot in a VXS crate. There are no inter-slot connections between the payload slots, therefore any inter-slot communication is handled by the Switch module. Therefore, in addition to using Switch B for signal distribution, other secondary functions of the Signal Distribution module are:

a) Managing payload module access to the VME bus (using the Token Passing scheme)]
b) Managing Intra-crate generated signals (i.e. collecting trigger and busy signals from the FADCs and passing them to the Trigger Interface board)
c) Providing a high speed link (approx 200Mbps) for data flow from the ADC module slots to the TI slot.

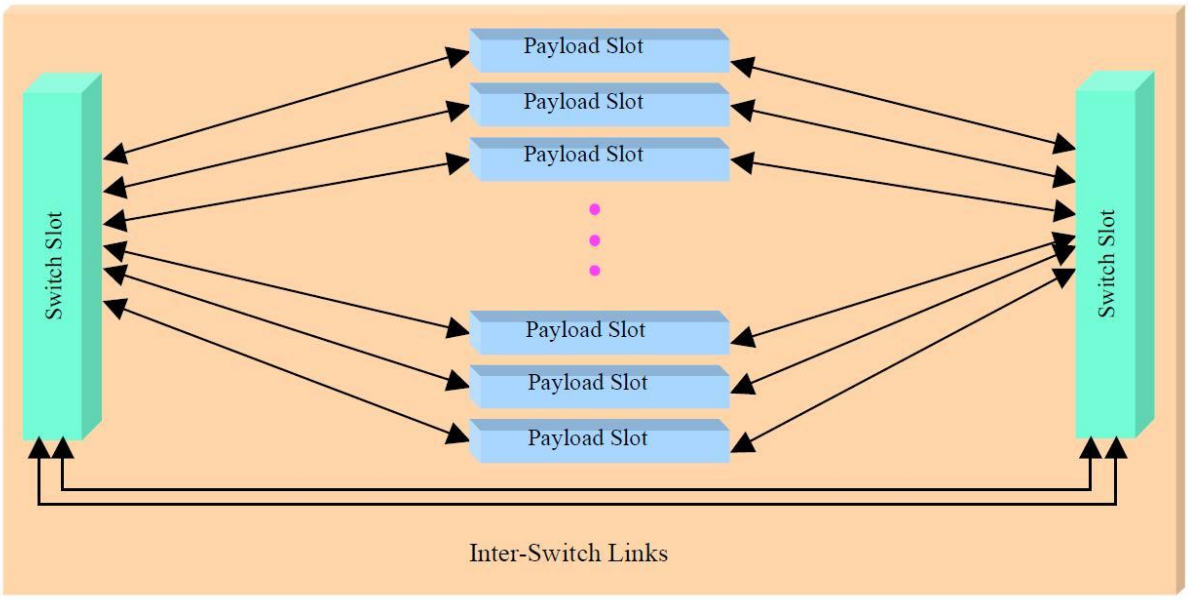


Figure 1-1 Redundant Star

Figure 1: Shows the Redundant Star configuration of the VXS crate.

2.0 Specifications Summary

MECHANICAL

- Single width VITA 41 "B" Switch Module

Front Panel:

- 5 – differential output ECL Signals
- 3 – Single-ended output LVPECL signals
- 2 – differential output LVPECL signals
- 2 – differential input LVPECL signals
- 1 - JTAG connector
- 1 - Active Serial connector
- 6 - Fault Indicator LEDs
- 1 - Reset Switch (momentary connect)

VXS OUTPUT SIGNALS

LVPECL level Signals (PP18 → PP[1 .. 16]):

- CLOCK
- SYNC
- TRIG1
- TRIG2

LVDS level Signals (PP[1 .. 16] → SD):

- SD_LINK
- TOKEN_IN
- TRIG_OUT

LVDS level Signals (PP18 → SD):

- TOKEN _OUT

LVDS level Signals (PP18 → SD):

- TOKEN _OUT

PROGRAMMING:

- Front Panel JTAG Port
- Front Panel Active Serial Port
- I^2C from Payload Port 18 (Trigger Interface)

POWER REQUIREMENTS:

- +5 V @ 8 amps Fused on board:
- local regulators for other required voltages

ENVIRONMENT:

- Forced air cooling.
- Commercial grade components (Celsius)

- Clock Jitter: (*to be tested)*
- Clock Signal Skew: (*to be tested)*

Power Section:

The switch slot in a VXS crate is powered by 5V (up to 40A). On the Signal Distribution board, most of the components are powered at 3.3V. The regulator used to provide the 3.3V is Linear Technology LTM4608 which can provide 8A of current at 3.3V. Other voltages required by the circuit are 2.5V and 1.2V, to provide power to the FPGA, these voltages are achieved by use of 1A linear regulators.
To terminate most of the LVPECL signals, 1.3V is provided by use of a linear regulator which can sink up to 2A. Addition of ECL signals required the use of an isolated power supply that provides -3.3V.

Total theoretical maximum current draw of the board is about 7A. The board is fused with 10A at the input and a reverse polarity diode protects the circuit from being damaged.

3.0 Functional Description [Block Diagram]

A VITA41 switch slot module is shown in Figure 2. Note the high speed multi-gigabit connectors (P1→P5) used for the point to point signals from payload modules to the switch card. The switch slot module only receives +5V power from the VXS backplane. Other voltages are generated using onboard voltage regulators. There are two mechanical alignment keys (K1 and K2) that are required to facilitate the proper mating between the board connectors and the backplane connectors. The switch board is a standard form factor of dimensions, 6U x 160mm x 4HP, as defined by IEEE 1101.1 standard.

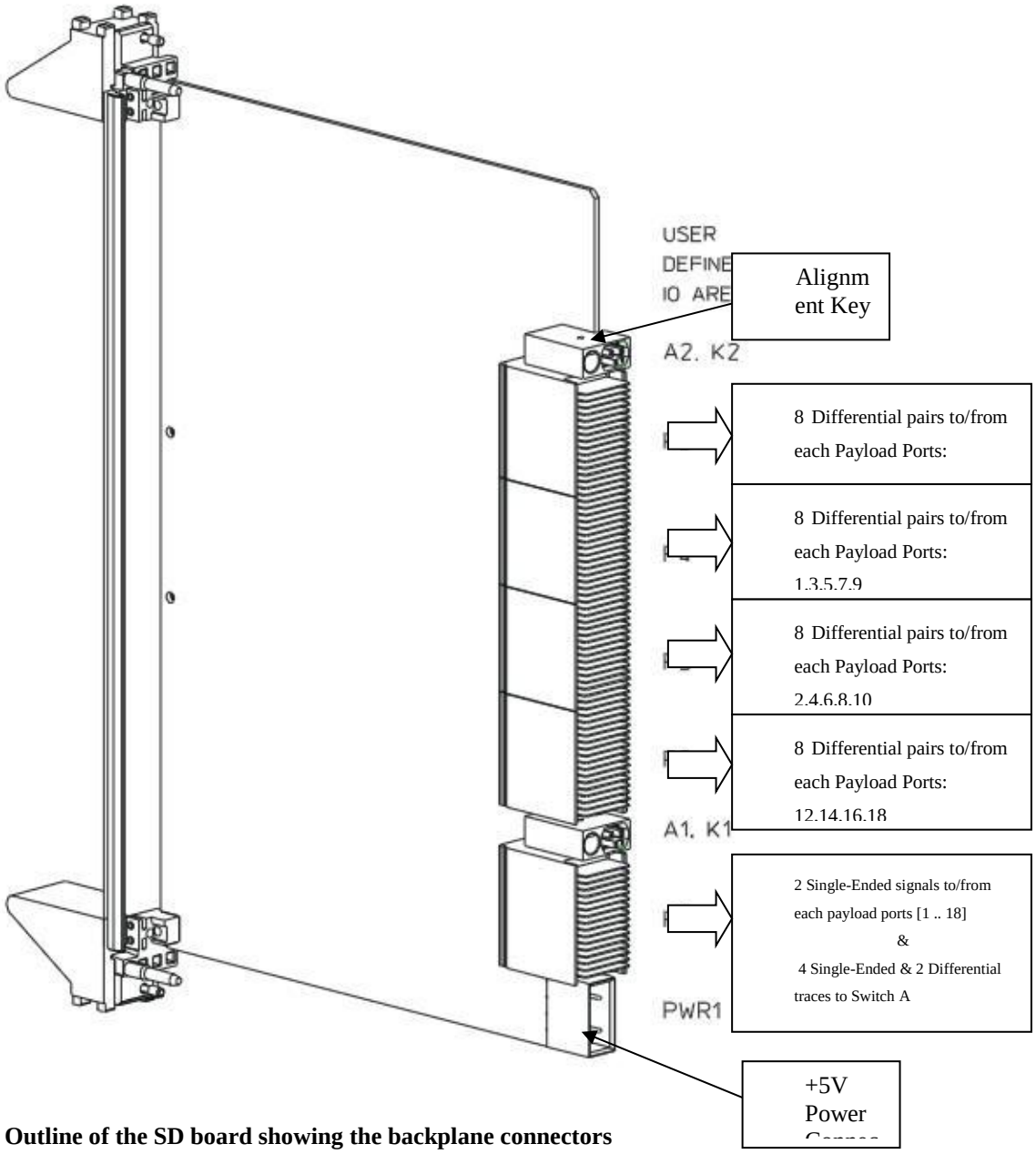


Figure 2. Outline of the SD board showing the backplane connectors

3.1: All Signals to/from the SD board.
Figure 3 below shows how the payload slots are logically connected to the "B" switch slot. The Signal Distribution module has been designed to accept all eight (8) differential pairs and two (2) single-ended from each payload port. The signal direction is defined by the user-defined function and the type of signal buffer (driver/receiver).

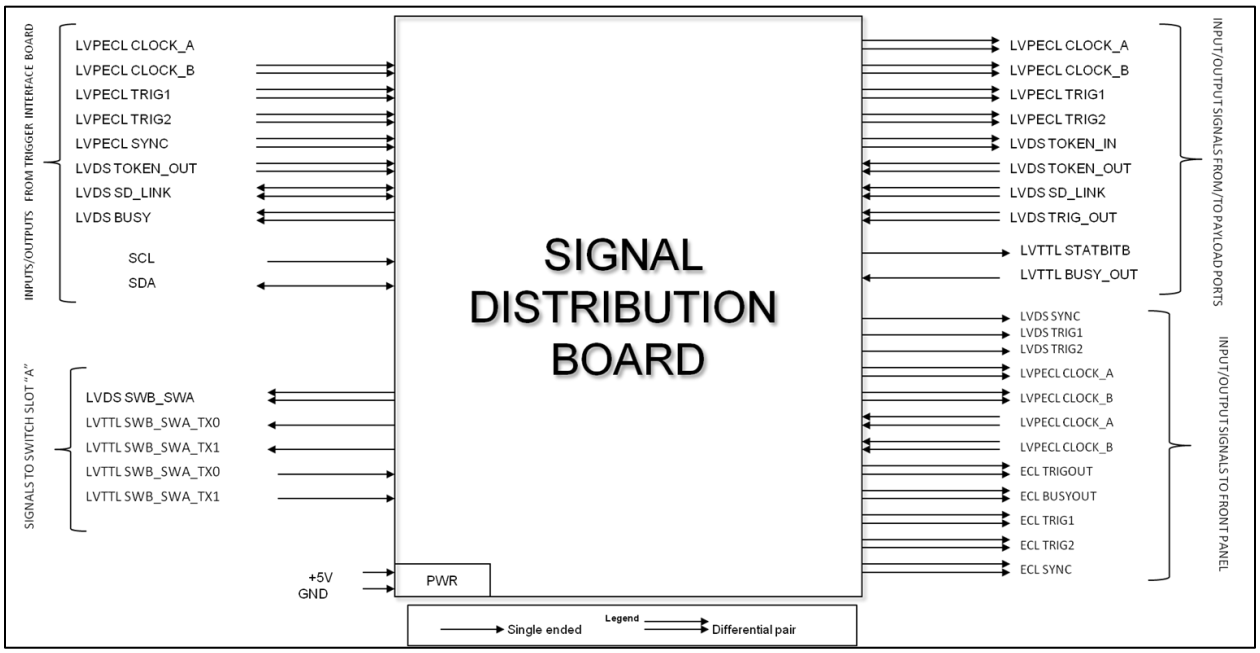


Figure 3 Shows the names and types of signals to/from the SD board

3.2: LVPECL Signals from the TI to Payload Boards via the SD board.
As shown in Figure 4 below the clock signals (ClockA and ClockB) from Trigger Interface board (PP18) are fanned out to drive the sixteen payload modules and Switch A (Crate Trigger Processor) module. To further reduce jitter on the clocks, a jitter attenuating PLL circuit is used before fanning out the signals to the payload slots. All the signals from the fanout buffer chip to the payload slot connectors are length matched to minimize device-to-device skew. CLOCKA and CLOCKB signals are 3000mils ± 10mils

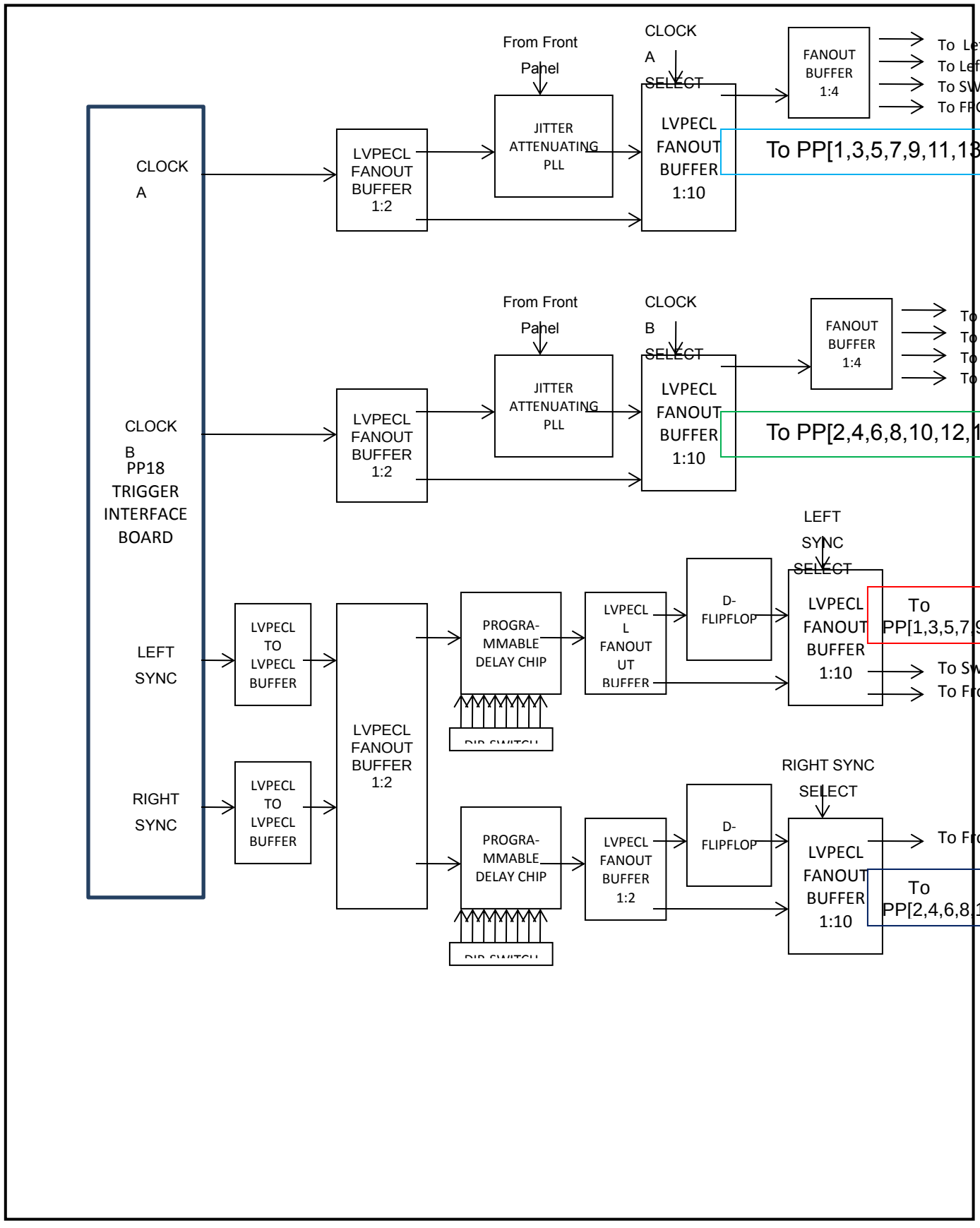
ClockA and ClockB are independent and can therefore distribute different frequencies in the same crate. This makes it possible to populate the same front-end crate with different payload module types (e.g. FADC and TDC). Note also that it is possible to get a clock input from the front panel to be fanned out.

Trig1, Trig2 and Sync signals are received from the Trigger Interface board (PP18) and are fanned out to the sixteen payload modules. The signals can be distributed in two ways:

a) They can be registered to reduce skew between the clock and these signals. That means the signals can be delayed to be in sync with the clock edge before being fanned out.
b) They can also be unregistered which means that they can be fanned out as received from the payload port 18 (TI).

All the signal traces from the fanout buffers to the backplane are length-matched to minimize propagation delay and device to device jitter between the fanout buffer chip and the backplane connectors.

- TRIG1 signals are 4000mils ± 0.5%
- TRIG1 signals are 5000mils ± 0.5%
- TRIG2 signals are 6000mils ± 0.5%


CLOCK
A
CLOCK
B
PP18
TRIGGER
INTERFACE
BOARD
LEFT
SYNC
RIGHT
SYNC
LVPECL
FANOUT
BUFFER
1:2
From Front
Panel
JITTER
ATTENUATING
PLL
CLOCK
A
SELECT
LVPECL
FANOUT
BUFFER
1:10
FANOUT
BUFFER
1:4
To Left SYNC CLK
To Left TRIG1 CLK
To SWA CLK
To FPGA
To PP[1,3,5,7,9,11,13,15]
From Front
Panel
JITTER
ATTENUATING
PLL
CLOCK
B
SELECT
LVPECL
FANOUT
BUFFER
1:10
FANOUT
BUFFER
1:4
To Right SYNC C
To Right TRIG1 C
To Right TRIG2 C
To FPGA
To PP[2,4,6,8,10,12,14,16]
LVPECL
TO
LVPECL
BUFFER
LVPECL
FANOUT
BUFFER
1:2
PROGRA-
MMABLE
DELAY CHIP
LVPECL
L
FANOUT
UT
BUFFER
D-
FLIPFLOP
LEFT
SYNC
SELECT
LVPECL
FANOUT
BUFFER
1:10
To
PP[1,3,5,7,9,11,13,15]
To Switch A
To Front Panel
RIGHT SYNC
SELECT
LVPECL
FANOUT
BUFFER
1:2
To Front Panel
To
PP[2,4,6,8,10,12,14,16]

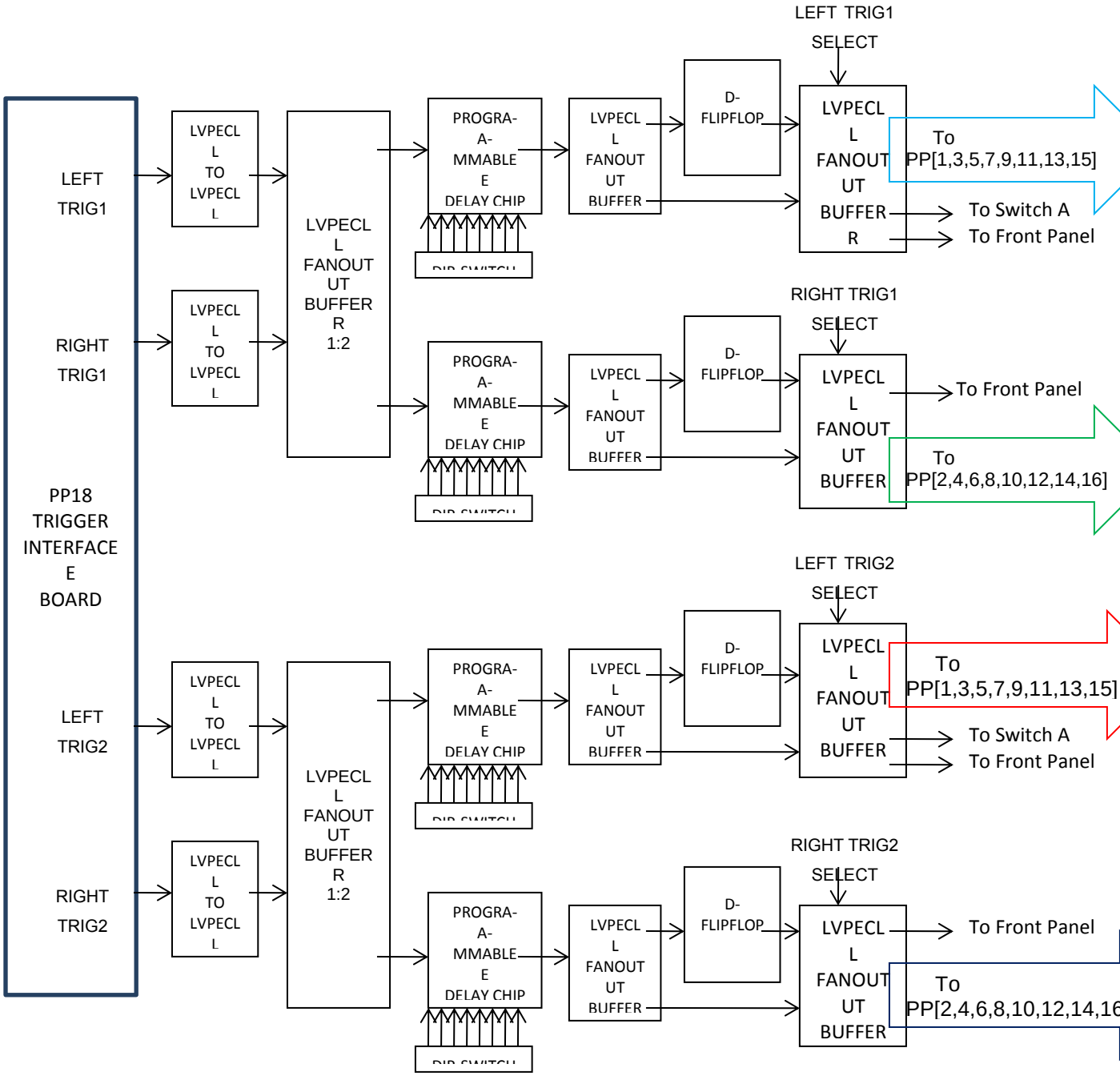


Figure 4. Shows a detailed block diagram of the LVPECL signals from the TI to the ADC payloads

3.3: LVDS Signals to/from the Payload Boards from/to the SD board.

Figure 5 shows other backplane signal pairs that the Signal Distribution module must manage. 3 LVDS inputs and 1 LVTTL input signal from each payload module and 1 LVDS and 1 LVTTL output signals to each payload module are connected to the FPGA.

The (3) LVDS input signals are:

i) SD_LINK: This connection is for high speed data transfer (200Mbps peak) from the PP[1 → 16] to the SD board.

ii) Token_IN: This connection receives the token from a payload board and sends the token to the next payload board in the token passing sequence. The token is used to prevent bus conflict therefore only the payload board that holds the token has access to read/write on the VME bus.

iii) Trig_OUT: This connection is for receiving FADC generated triggers.

The LVDS output signal is Token_OUT which sends out the token to the next payload in the token passing sequence.
The LVTTL input signal is Busy_Out which is asserted by a payload module when its data buffer is full. The signal is passed on to the TI module via the SD board.
The LVTTL output signal is StatBitB which can simultaneously send data to all 16 payload module or to individual payload modules.
All the LVDS signals are translated to LVTTL signals before connection to the FPGA; this was done to reduce the FPGA I/O pin requirement.

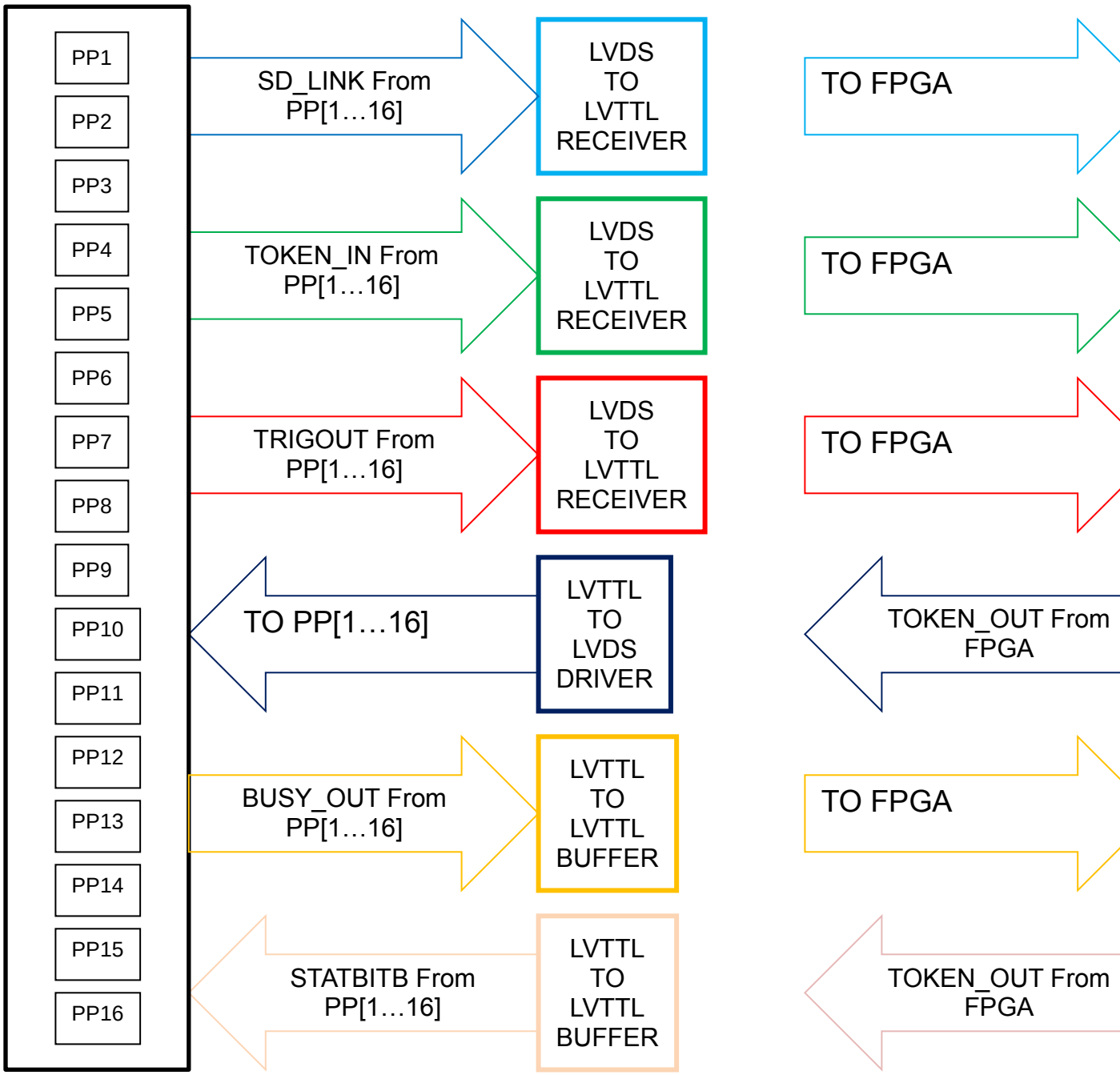
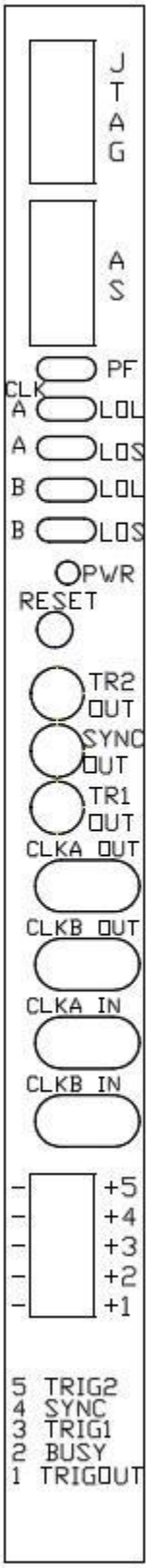


Figure 5: Shows the LVDS to LVTTL signal connections from the Payload Modules to the FPGA

Front Panel Specification Sheet



MECHANICAL

- Single width VITA 41 "B" Switch Module

INPUT SIGNALS

- CLOCKA (LVPECL)
- CLOCKB (LVPECL)

OUTPUT SIGNALS

- CLOCKA (LVPECL)
- CLOCKA (LVPECL)
- SYNC (LVPECL)
- TRIG1 (LVPECL)
- TRIG2 (LVPECL)
- BUSY (ECL)
- SYNC (ECL)
- TRIG1 (ECL)
- TRIG2 (ECL)
- TRIGOUT (ECL)

PROGRAMMING:

- JTAG Port
- Active Serial Port

LEDS:

- Power LED: Illuminates green when the board is powered up.
- PF: Power Fault (illuminates Red if the switching regulator LTM4608 malfunctions, green otherwise.
- CLOCKA loss of signal (LOS) illuminates red if the jitter attenuating PLL loses input signal, green otherwise.
- CLOCKB loss of signal (LOS) illuminates red if the jitter attenuating PLL loses input signal, green otherwise.
- CLOCKA loss of lock (LOL) illuminates red if the jitter attenuating PLL is unlocked, green otherwise.
- CLOCKB loss of lock (LOL) illuminates red if the jitter attenuating PLL is unlocked, green otherwise.

RESET:

- Momentary connect switch that puts the SD board in a known initial state when it is enabled.

4.0 Detailed Functional Description

4.1: Introduction

The Signal Distribution module is an integral part of the level 1 trigger architecture proposed for Jefferson Lab experimental halls. The Trigger architecture is composed of 3-unique crates:

- Trigger Distribution Crate
- Front End Crate
- Global Trigger Crate

The Signal Distribution module is present in all of the above crates which makes its function critical to the performance of the trigger architecture. The SD board is in the switch B position of the VME crate. Its purpose is to:

1. Distribute the ClockA, ClockB, Sync, Trig1 and Trig2 signals from the TI to all the payloads.
2. Collect TrigOUT and BusyOUT signals from all the payloads and send them to the TI.
3. Implement the token passing scheme for all payloads.
4. Communicate with the TI board (payload port 18) via I^2C.
5. Program the Jitter attenuating PLLs via SPI.
6. Collect high speed data (up to 200Mbps) from the payload boards and send the data to the TI.

1. Signal distribution

a) Clock Distribution

The main function of the SD board is signal distribution. Low skew, low jitter, low voltage PECL chips are used on the board for the distribution scheme. All the signals being fanned out from the TI to the payload modules is in LVPECL format.
The fanout scheme has been chosen such that the signals received from the Trigger Interface board are distributed to the right and left side of the crate using separate fanout chips.
Jitter on ClockA and ClockB is further attenuated using a jitter attenuating PLL. The PLLs are programmed via SPI protocol with the registers programmed according to the output frequency expected. The jitter attenuating PLLs were added to the SD board to enhance the clock jitter before distributing the clock to the FADCs. Clock jitter is prevalent mostly due to the fiber optic links used to connect the front end crate to the Trigger Distribution crate. The SD is the last board to receive the clock before distributing it to the FADCs, SSPs, TDs and other payload boards, therefore it makes sense that the jitter attenuation is done on the SD board.

b) Trigger and Sync signal Distribution

In order to compensate for the delay between clock, trigger and sync signals, a programmable delay is added to the trigger and the sync signal lines. In the front-end crate, these signals are registered (synchronized with the clock) before being fanned out to the payload modules**.** The delay is empirically calculated and adjusted using hardware mounted on the SD board.
It is also possible to fanout the trigger and sync signals without registering. This option is needed when the SD board is mounted in the Global Trigger crate. The delay chip has a resolution of 10ps selectable to a range of 10ns.

c) Inter switch Communication

Sync, Trigger (Trig1 and Trig2) and ClockA signals are shared by the Signal Distribution board and the Crate Trigger Processor board (Switch "A"). This is to maintain backwards compatibility between the new revision boards and the boards manufactured before the backplane mapping was changed. In addition, two spare single-ended signals and two differential pairs have also been assigned for inter switch communication.

2. Token passing

A token passing scheme has been implemented for the Signal Distribution board and the payload ports. The token passing scheme is necessary to manage payload module access to the VME bus to prevent bus conflict. Payload modules access the VME bus for write purposes and because they are slaves on the VME bus, they need a control signal to manage the VME bus write. The control signal is initialized by the TI board and is passed to each payload module in a predetermined sequence. Only the payload module that holds the control signal (token) is allowed to write to the VME bus.
After appropriate processing at a payload port, the token signal is handed back to the SD to be passed on to the next payload port. In this scheme bypassing of payload ports is also possible. The order of flow of the token signal is from the leftmost to rightmost slot in the VXS crate. The last board in the token passing scheme alerts the TI after it completes its write. The system is initialized when the TI provides a token.
Any change to the token passing scheme is done during the board initialization process and can only be changed if the TI reprograms the SD board registers.

3. Aggregate TrigOUT and BusyOut signals

The Signal Distribution board receives TrigOut and BusyOut signals from all the payload boards.

TrigOut is asserted by the FADCs when certain preset conditions are met to generate a trigger. TrigOut signals from all the payloads are OR'ed and passed to the front panel of the SD board. The front panel TrigOut signal is reserved for intra-crate triggering where the signal is then connected to the front panel of the TI board which manages the incoming trigger signal.
BusyOut is asserted by the FADCs when its data buffer reaches a preset threshold. The finite data buffer will become full if the payload board is collecting more data than it's writing to the readout bus. The BusyOut signals are OR'ed from all the payloads and sent to the TI which in turn passes the signal to the Trigger Distribution Crate to alert the trigger supervisor which will manage resources. The BusyOut signal is handled the same way in both the Front End Crate and the Trigger Distribution Crate.
The SD keeps track of the source of the BusyOut signal by keeping a count of the number of times each board has asserted the signal and which board last asserted the signal. This information is important when troubleshooting to find a faulty payload module.
There are three registers that are related to the BusyOut signal:

- BusyOut_State (Read Only) register 6. This is a 16-bit register that sets a flag for each slot that has asserted it's BusyOut signal. The flag is cleared after this register is read.
- BusyOut_Mask (Read/Write) register 3. This is a 16-bit register that is used to mask the slots that the user chooses to keep track of BusyOut signals. Write a '1' to keep track of the BusyOut signal of the corresponding slot or a '0' to mask out a particular slot. The default is to keep track of all Slots.
- BusyOut_Counter (read only) registers 8 (slot 2) to register 23 (slot 19). Each payload is assigned a 16-bit register that holds the number of times the BusyOut signal has been asserted for the particular payload. The counter in each register is cleared after the register is read.

4. I^2C Communication

The Signal Distribution board gets its initial register configuration from the Trigger Interface (TI) board (Payload Port 18) in each unique crate. Data transfer between the SD and the TI board is done through I^2C protocol. There are two dedicated single-ended lines that are reserved for this purpose. The proposed method of keeping the TI and the SD synchronized is to have a memory map on the TI that is a duplicate copy of the SD memory map.
The TI board is the always the master in the protocol and the SD is always the slave. Therefore, to keep the memory map up to date the TI will have to request data from the SD frequently. To reduce the amount of time it takes for this data exchange, the proposal is to run the I^2C communication protocol at a high data rate (the final value of which will be determined through testing).
The memory map and the programming requirements are detailed in section 5.0 below.

5. SPI Communication

The Signal Distribution board is responsible for configuring the jitter attenuating PLLs (Silicon Labs Si5326). Some of the register information needed to correctly configure the PLLs is written by the TI board. Data transfer between the SD and the PLLs board is done through SPI protocol. There are two jitter attenuating PLLs on the SD board one for ClockA and one for ClockB.

There are dedicated communication lines that are reserved for SPI communication purpose. Both PLLs share a common: Data In, Data Out, and Clock lines but each has a unique Slave Select line. The proposed method of keeping the SD and the PLLs synchronized is to have a memory map on the SD that is a duplicate copy of the PLL memory map. This will reduce the need to constantly poll the PLLs each time a data read is requested by the TI. Constant data reads can theoretically inject noise into the circuit.

The SD board is the always the master in the protocol and the PLLs are always the slave. Therefore, to keep the memory map up to date the SD will have to request data from the PLL as needed. To reduce the amount of time it takes for this data exchange, the proposal is to run the SPI communication protocol at a high data rate (the final value of which will be determined through testing). The minimum Clock period for the PLLs is 100ns which limits the data rate to a maximum of 10MHz.

The PLL memory map and the programming requirements are detailed in section 5.0 below.

6. High speed data

There is a dedicated differential pair between the all payload boards and the SD boards. It is a unidirectional link from the payloads [1 → 16] to the SD and a half-duplex link between SD and the TI. The proposed function is to have an alternative data path from the Payload modules to the TI board. Data speeds and frequency of use is yet to be tested and determined. However, hardware limits the data rate to a maximum of 200Mbps.

5.0 Programming Requirements

There are two main ways that the FPGA register configuration on the SD board can be programmed:

a) Using the JTAG or AS device connection on the front panel of the board.
b) Writing to the FPGA using I^2C.

Initial programming of the SD board is done using the Active Serial (AS) configuration and registers are initialized with specified Reset Values.

The VXS Signal Distribution module receives initial register status settings from the Trigger Interface (TI) board using I^2C bus protocol. The TI always plays the role of bus master and the SD board plays the role of bus slave.
The following register definitions will always be written with the specified Reset Values until programmed by the TI board through I^2C.

The register addresses are two bytes and the data is also two bytes long. The TI sends data to the SD in the following sequence:

i) First sends the SD address (one byte)
ii) Sends the MSB of the register address.
iii) Sends the LSB of the register address.
iv) Send the MSB of data_1.
v) Send the LSB of data_1.
vi) Send the MSB of data_2
vii) Send the LSB of data_2.

The register address is automatically incremented as more data is received until a stop condition is encountered which resets the state machine.

Register Map

Physical Slot #	2	3	4	5	6	7	8	9	12	13	14	15	16	17	18	19
Payload Port#	15	13	11	9	7	5	3	1	2	4	6	8	10	12	14	16
Logical Slot #	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

The register map is defined according to the Physical slot location in the crate. The first slot is the left most slot in the crate and the last slot is the rightmost slot in the crate. In a 20 slot crate the designation is as shown in the table above.
Physical slot #2 is the leftmost slot occupied by an FADC module and is designated as the first slot (logical slot #0). Physical slots #10 and #11 are occupied by switch modules and are therefore not assigned corresponding logical slot numbers.

The logical slot designation described above is the one used in the following register map definition.

Register Address Map Summary:

i) SD system status: 16-bit wide register that holds the settings for the main SD functions.
ii) SD Main Status 16-bit wide register showing the registers updated since the last read by TI.
iii) Populated boards: 16-bit register that shows all the boards populated in the crate.
iv) TokenPassingReg 16-Bit register that shows payloads taking part in the token passing scheme.
v) BusyOutMask 16-bit register containing the payload(s) to take part in the masked OR.
vi) TrigOutMask 16-bit register containing the payload(s) to take part in the masked OR.
vii) BusyOutState 16-bit register showing the payload(s) which have asserted busy since last read.
viii) TrigOutState 16-bit register showing the payload(s) that have asserted TrigOut.
ix) BusyOut_Cnt[1 .. 16] Sixteen 16-bit registers keeping count of how many times each payload has asserted busy since the last read.
x) sdtest_busyout
xi) sdtest_sdlink
xii) sdtest_token_in
xiii) sdtest_trigout
xiv) sdtest_tokenout_reg
xv) sdtest_statbitb_reg
xvi) Version_Reg

Command, Status and Configuration Registers

Register 0. SYSTEM_REG Address = 0x00

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	SD_TEST RESET	RFU	RFU	RFU	RFU	RFU	ENABLE SD_TI_Link	RFU
Type	R/W	R	R	R	R	R	R/W	R

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	CLKB FREQ	CLKB FREQ	RFU	CLKB PLL BYPASS	CLKA FREQ	CLKA FREQ	RFU	CLKA PLL BYPASS
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 1000 0000 1100 1100

Bit	Name	Function
15	SD_TEST RESET	Reset the SD Test Code when set to 1, the bit should be set to 0 to operate normally. 0: Reset SD 1: Normal Operation
14:10	RFU	Reserved for Future Use
9	SD_TI_LINK ENABLE	Register Data synchronized between SD and TI. 0: Link inactive 1: Link active
8	RFU	Reserved for Future Use
7:6	USER SET CLKB FREQUENCY	ClockB Freq (MHz) – This user set value determines the frequency programmed into the PLL. 00: Undefined 01: 31.25 10: 125.00 11: 250.00
5	RFU	Reserved for Future Use
4	CLKB PLL BYPASS	This selects ClockA mode 0: Bypass Mode 1: PLL Mode
3:2	USER SET CLKA FREQUENCY	ClockA Freq (MHz) – This user set value determines the frequency programmed into the PLL. 00: Undefined 01: 31.25 10: 125.00 11: 250.00
1	RFU	Reserved for Future Use
0	CLKA PLL BYPASS	This selects ClockA mode 0: Bypass Mode 1: PLL Mode

Register 1. STATUS_REG Address = 0x01

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	RFU	RFU	RFU	RFU	CLKB FREQ	CLKB FREQ	CLKA FREQ	CLKA FREQ
Type	R	R	R	R	R	R	R	R

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	RFU	BUSYOUT STATUS	TRIGOUT STATUS	POWER FAULT	CLKA LOL	CLKA LOS	CLKB LOL	CLKB LOS
Type	R	R	R	R	R	R	R	R

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:12	RFU	Reserved for Future Use
11:10	DETECTED CLKB FREQUENCY	ClockB Freq (MHz) – The detected frequency received from the TI 00: Undefined 01: 31.25 10: 125.00 11: 250.00
9:8	DETECTED CLKA FREQUENCY	ClockA Freq (MHz) – The detected frequency received from the TI 00: Undefined 01: 31.25 10: 125.00 11: 250.00
7	RFU	Reserved for Future Use
6	BUSYOUT_ STATUS	This bit is set when any board has asserted its BusyOut. 0: No BusyOut. 1: At least one board has asserted BusyOut (Masked).
5	TRIGOUT_ STATUS	This bit is set when any board has asserted its TrigOut. 0: No TrigOut. 1: At least one board has asserted TrigOut (Masked).
4	POWER_FAULT	This bit is set when there is a power fault detected. 0: Normal Operation 1: Powerfault detected
3	CLKA_LOL	This bit is set after ClockA PLL Loss of Lock is detected and set to 0 after Loss of Lock is resolved 0: Normal Operation 1: Loss of Lock on ClockA PLL.
2	CLKA_LOS	This bit is set after ClockA PLL Loss of Signal is detected 0: Normal Operation 1: Loss of Signal
1	CLKB_LOL	This bit is set after ClockB PLL Loss of Lock is detected 0: Normal Operation 1: Loss of Lock
0	CLKB_LOS	This bit is set after ClockB PLL Loss of Signal 0: Normal Operation 1: Loss of Signal

Register 2. **POPULATED BOARDS REG** **Address = 0x02**

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	PAYLOAD_BOARDS[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	PAYLOAD_BOARDS[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:0	POPULATED_ BOARDS[15:0]	Sets the value for the payload boards that are populated in the crate. 0: Board not present in crate 1: Board present in crate *Note: D0→ Slot 1 … D15→ Slot 16*

Register 3 **TOKEN PASSING MASK** **Address = 0x03**

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	TOKEN_PASSING_MASK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	TOKEN_PASSING_MASK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:0	TOKEN_ PASSING _REG[15:0]	Sets the payload boards that are taking part in the Token Passing Scheme. 0: Board not participating in token passing. 1: Board participating. *Note: D0→ First Slot in Token Passing Sequence … D15→ last Slot in Token Passing Sequence*

Register 4. **BUSYOUT MASK** **Address = 0x04**

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	BUSYOUT_MASK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	BUSYOUT_MASK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:0	BUSYOUT _MASK[15:0]	Sets the value for the payload boards whose BusyOut needs to be sent to the TI board. 0: Board not participating in BusyOut to TI. 1: Board participating.

Register 5. TRIGOUT MASK Address = 0x05

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	TRIGOUT_MASK[15:8]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	TRIGOUT_MASK[7:0]							
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:0	TRIGOUT _MASK[15:0]	Sets the value for the payload boards whose TrigOut is participating in generating a trigger output. 0: Board **not** participating in generating a trigger. 1: Board participating.

Register 7. BUSYOUT STATE Address = 0x07

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	BUSYOUT_STATE[15:8]							
Type	R	R	R	R	R	R	R	R

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	BUSYOUT_STATE[7:0]							
Type	R	R	R	R	R	R	R	R

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:0	BUSYOUT_ STATE[15:0]	Shows the current BusyOut state of the payload boards. 0: Normal Operation 1: Busy Out asserted. *Note: D0 → Slot 1 … D15 → Slot 16*

Register 8. TRIGOUT STATE Address = 0x08

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	TRIGOUT_STATE[15:8]							
Type	R	R	R	R	R	R	R	R

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	TRIGOUT_STATE[7:0]							
Type	R	R	R	R	R	R	R	R

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:0	TRIGOUT_ _STATE[15:0]	Shows the most recent TrigOut state of the payload boards. 0: Normal Operation 1:TrigOut asserted.

Register 9. BUSYOUT COUNTER Address = 0x09

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	BUSYOUT_COUNTER[15:8]							
Type	R	R	R	R	R	R	R	R

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	BUSYOUT_COUNTER[7:0]							
Type	R	R	R	R	R	R	R	R

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:0	BUSYOUT_ COUNTER[15:0]	Shows the current count for the number of times a payload board has asserted its BusyOut since the last read. The value is reset to zero after being read. 0000 0000 0000 0000: (Zero) To 1111 1111 1111 1111: (65535)

Note: Addr 0x0009 → Payload board 1 ……. Addr 0x0018 → Payload board 16

SDTEST Registers: These registers are used when testing the SD board connectivity with payload boards. The test program reads/writes to the registers to validate the connection. The input registers (READ ONLY) show the value of the payload boards with the MSB being PP16 and the LSB is PP01.

Register 19: SDTEST_BUSYOUT Reset value = 0000 0000 0000 0000 (**READ ONLY**)
Register 1A: SDTEST_SDLINK Reset value = 0000 0000 0000 0000 (**READ ONLY**)
Register 1B: SDTEST_TOKEN_IN Reset value = 0000 0000 0000 0000 (**READ ONLY**)
Register 1C: SDTEST_TRIGOUT Reset value = 0000 0000 0000 0000 (**READ ONLY**)
Register 1D: SDTEST_TKNOUT Reset value = 0000 0000 0000 0000
Register 1E: SDTEST_STATBITB Reset value = 0000 0000 0000 0000
Register 1F: Version_Reg Reset value = 0000 0000 XXXX XXXX (**READ ONLY**)
This register holds the firmware version that is programmed in the SD board.

Register 21. TRIGOUT_COUNTER Address = 0x29

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	TRIGOUT_COUNTER[15:8]							
Type	R	R	R	R	R	R	R	R

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	TRIGOUT_COUNTER[7:0]							
Type	R	R	R	R	R	R	R	R

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:0	TRIGOUT_ COUNTER[15:0]	Shows the current count for the number of times a payload board has asserted its TrigOut since the last read. The value is reset to zero after being read. 0000 0000 0000 0000: (Zero) to 1111 1111 1111 1111: (65535)

Note: Addr 0x0021 → Payload board 1 …….. Addr 0x0030 → Payload board 16

Register 20. SDTEST_REG Address = 0x20

Bit	D15	D14	D13	D12	D11	D10	D9	D8
Name	RFU	RFU	TEST_FP_ TRIGOUT	TEST_FP_ BUSYOUT	RFU	TEST_TI_ BUSYOUT	TEST_TI_ SD_LINK	TEST_TI_ TKNOUT
Type	R	R	R/W	R/W	R	R/W	R/W	R/W

Bit	D7	D6	D5	D4	D3	D2	D1	D0
Name	RFU	SWB_TO_ SWA_SE1	SWB_TO_ SWA_SE2	SWB_TO_ SWA_DP	RFU	SWA_TO_ SWB_SE1	SWA_TO_ SWB_SE2	SWA_TO_ SWB_DP
Type	R	R	R	R	R	R/W	R/W	R/W

Reset value = 0000 0000 0000 0000

Bit	Name	Function
15:14	RFU	Reserved for Future Use
13	TEST_FP_ TRIGOUT	When in test mode, the bit value written here should appear at the front panel TRIGOUT output.
12	TEST_FP_ BUSYOUT	When in test mode, the bit value written here should appear at the front panel BUSYOUT output.
11	RFU	Reserved for Future Use
10	TEST_TI_ BUSYOUT	When in test mode, the bit value written here should appear at the TI's BUSYOUT input.
9	TEST_TI_ SD_LINK	When in test mode, the bit value written here should appear at the TI's SD_LINK input.
8	TEST_TI_ TKNOUT	When in test mode, the bit value written here should appear at the TI's TKNOUT input.
7	RFU	Reserved for Future Use
6	SWB_TO_ SWA_SE1	When in test mode, the bit value written here should appear at the SWB_TO_SWA_SE1 input.
5	SWB_TO_ SWA_SE2	When in test mode, the bit value written here should appear at the SWB_TO_SWA_SE2 input.
4	SWB_TO_ SWA_DP	When in test mode, the bit value written here should appear at the SWB_TO_SWA_DP input.
3	RFU	Reserved for Future Use
2	SWA_TO_ SWB_SE1	When in test mode, the bit value written here should appear at the SWA_TO_SWB_SE1 input.
1	SWA_TO_ SWB_SE2	When in test mode, the bit value written here should appear at the SWA_TO_SWB_SE2 input.
0	SWA_TO_ SWB_DP	When in test mode, the bit value written here should appear at the SWA_TO_SWB_DP input.

This register shows the number of clock cycles counted during a 0.1ms period. These values determine the CLOCK frequencies of CLOCKA and CLOCKB which are then updated in the status reg.

Register 31: CLOCKA_COUNT Reset value = 0000 0000 0000 0000 (**READ ONLY**)

Register 32: CLOCKB_COUNT Reset value = 0000 0000 0000 0000 (**READ ONLY**)

TRIGOUT Registers: The following registers are used when reading/writing the TRIGOUT lookup table.

Register 39: TRIGOUT_PW_STRETCH Reset value = 0000 0000 0000 0101
This is a 16-bit value to stretch the sampling window pulse width of the TrigOut table input address. This value defaults to 5 clock periods (100ns) and can be increased by writing to this register.

- **bit** (**15**): if '1' Update mode if '0' non-update mode
- **bits** (**14:0**): 15-bits representing the value of the TrigOUT window that allows the TrigOUT bits from the payload boards to be detected and decoded (each bit increment is equivalent to 4ns window width increment).

Register 40: TRIGOUT_ADDR Reset value = 0000 0000 0000 0000 (12-bit address for the TrigOut lookup table)
Register 41: TRIGOUT_DATA Reset value = 0000 0000 0000 0000 (16-bit data for the TrigOut lookup table)
To read a particular address in the lookup table, write the address to Reg 40 and read the data from Reg 41.

SD Configuration Registers: The following registers are used when remotely programming the SD firmware into the external SRAM. The instructions are found in the "Remote Programming of SD Board" document.

Register 45:	**Config_ADDR (15 : 0)**	16-bit value that is equivalent to the LSB of the Config memory address.
Register 46:	**Config_ADDR (23 : 16)**	16-bit value that is equivalent to the MSB of the Config memory address.
Register 47:	**Config_DATA (7 : 0)**	8-bit Config memory data.
	Config_DATA (8)	1-bit value executes the write to Config memory.
	Config_DATA (9)	1-bit value enables the write to Config memory.
	Config_DATA (10)	1-bit value shifts the data bytes to Config memory.
	Config_DATA (12)	1-bit value executes the sector erase operation on Config memory.
	Config_DATA (13)	1-bit value executes the sector erase operation on Config memory.
Register 48:	**Config_DATA (8)**	1-bit value executes the read from Config memory.
	Config_DATA (9)	1-bit value enables the read from Config memory.
	Config_DATA (10)	1-bit value shows illegal erase of Config memory.
	Config_DATA (11)	1-bit value shows illegal write of Config memory.

(**READ ONLY**)

Register 49:	**Config_STATUS (7 : 0)**	8-bit Config status data.
	Config_STATUS (8)	1-bit value shows Config memory is busy.
	Config_STATUS (9)	1-bit value shows Config memory data is valid.

Communication Summary

I^2C

The protocol used to store and read data from the SD is in the format of:

- A byte representing address.
- A word (two bytes) representing data.

This I^2C link was successfully tested at a speed of about 3Mbps. The TI is the master and the SD is the slave in this communication link with an address of 1.

SD TRIGOUT FUNCTION.

The TrigOut function on the SD board is used to generate a programmable trigger bit within the crate. The SD receives a bit from payload board(s) and outputs a trigger bit based on the preprogrammed lookup table. The TrigOUT table is 4096 words deep. There are three (3) registers involved in the TrigOUT function programming:

- **Reg 39:**
    - **bit** (**15**): if '1' Update mode if '0' non-update mode (explained below)
    - **bits** (**14:0**): 12-bits representing the value of the TrigOUT window that allows the TrigOUT bits from the payload boards to be detected and decoded (each bit increment is equivalent to 4ns window width increment).
- **Reg 40:** This register programs the 12-bit address of the TrigOUT lookup table. The most significant nibble is ignored and only the 12 LSB are used for the address.
- **Reg 41:** This register holds the 16-bit data to write to the TrigOUT lookup table. The data is the bit pattern used to detect a valid trigger.

Programming the lookup table

The lookup table is populated by writing data to the SD using I^2C protocol. Writing data to the table is a two step process:

- Step 1: Write the LookUp table 16-bit address to (SD Reg 40h).
- Step 2: Write the LookUp table 16-bit data to (SD Reg 41h). The data will be written to the lookup table starting at the address written in step 1 above. Note that to fill the 65K bit lookup table, subsequent addresses will need to be multiples of 16 to address the look up table.

To read from the lookup table is also a two step process:

- Step 1: Write the LookUp table 16-bit address to read from to (SD Reg 40h).
- Step 2: Read the Lookup table data from SD Reg 41h.

Generating a trigger bit

The TrigOut bits from the payload boards are used to address the lookup table with PP16 being the MSB and PP01 being the LSB. The preprogrammed output trigger bit has a programmable pulse width that can be either updating on non-updating mode.

Updating mode means that if a trigger bit is detected during the trigger window, then the pulse width of the trigger window is extended by the same time period starting at the last received trigger bit. In the updating mode, if another trigger bit is detected during the previous trigger window, then the trigger window stretches starting from the new pulse until the preset pulse width elapses, unless another pulse is detected.

In non-updating mode, the trigger window is static. That is, if another trigger bit is detected while the previous pulse is active, then the new pulse is ignored and the pulse finishes after the preset pulse width.

In both modes, the trigger bit pulse width is set by writing to I^2C address (39h) the number of clock cycles the pulse-width should be, the default is 5 clock cycles (ensures a 20ns pulse when using the 250MHz clock). The pulse width is the time window that allows for trigger bits from the payloads to be received and decoded. The pulse width is scalable from 5 clock cycles to 16K clock cycles.

The “Update” bit is set by writing a ‘1’ to bit 15 of SD Reg 39h.

Counting Trigger Bits.

The trigger bits generated by each payload port are counted using a 16bit register. The counter is updated everytime a rising edge is detected on the trigger line. The trigger counter register is assigned as follows: Reg 21h → PP01 …. sequentially to …. Reg 30h → PP16. Reading the TrigOUT count register resets the corresponding counter to zero.

REMOTE PROGRAMMING OF SD BOARD and SERIAL NUMBER PROGRAMMING

Ability to program Signal Distribution (SD) boards via I^2C has been implemented. This is to support programming of the module without use of a JTAG cable. This requirement was added due to the nature of the trigger board locations and quantities in the Halls enabling users to remotely update firmware on the SD module.

In addition to the SD board, the following boards are needed to make this happen:

1. TI (Trigger Interface) module: This will send data using I^2C to the SD module.
2. CPU – Used to send VME transactions to the TI board which translates from VME to I^2C.

The SD board uses an Altera Cyclone III FPGA which is connected to a serial memory device EPCS64. The memory device stores firmware that is uploaded to the FPGA whenever the SD board is powered up. Updating the firmware therefore requires the FPGA to program the memory device with the new firmware. Once the new firmware is programmed in the memory device and verified for correctness, then power cycling SD power will upload the new firmware.

The firmware sent to the serial memory device is a raw bit form (.rbf) file generated by Altera Quartus compiler or another compatible compiler. The firmware is checked for correctness by reading back the serial memory device and comparing the contents with the original file. If the comparison is successful then the new firmware is saved, if the comparison is unsuccessful, then the serial memory device should be erased and the programming process restarted.

PROCEDURE for REMOTE PROGRAMMING.

CPU parses .rbf file and sends the data starting from MSB to TI using VME. The TI will send the information to the SD via I^2C.

The data should be written starting at the first sector of the memory (Address 000000h). The data should then be read back at the end of the total data transfer to verify that the correct information is programmed.

If the memory is programmed correctly, power cycling the SD will load the new program on restart.

If the memory is incorrect, a bulk erase should be issued and the process restarted after the bulk erase is complete.

Writing/Reading from the serial memory device on the SD board EPCS64, is done in one byte sizes. There are 16MB sectors that are addressable using 23 address bytes. Therefore to write to the serial device, the user needs to furnish 24bit address and 8bits of data at the least. To read from the serial device the user needs to furnish the 24bit address to read from.

WRITE

Writing data to the serial device can be done in two ways:

1. Writing a byte at a time.
2. Writing a page (256) bytes a time.

(refer to serial configuration devices datasheet for details).

When writing a byte, write the memory address to Addr45h and Addr46h. Then write the data in Addr47h [7 to 0] and set Bits 8 and 9 to ‘1’.

To write a page, write the memory address to Addr45h and Addr46h. Then write the data in Addr47h [7 to 0] and set Bits 9 and 10 to ‘1’ for 255 bytes (this puts the data in a buffer before writing to the serial memory). When sending the 256^{th} byte, set bits 8 and 9 to ‘1’ to send the data from the buffer to the serial memory.

The page method should be used to write large data sizes to memory because of time savings.

To program the serial memory device, the following registers on the SD board need to be programmed:

SERIAL MEMORY DEVICE Address is 24 bits long.
Addr45h: Bits [15 to 0]→ Write the LSB of the serial memory device address (16 bits).
Addr46h: Bits [7 to 0] → Write the MSB of the serial memory device address (8 bits).

Addr47h: Bits[7 to 0] → Write the Data to be sent to the serial memory device MSB first (8 bits)
Addr47h: Bit [8]-WRITE → 1: Executes the write operation to serial memory device.
Addr47h: Bit [9]-WREN → 1: Allows write and erase operations to be performed.
Addr47h: Bit [10]-SHIFT_BYTES → 1: Shift data bytes during the write operation.
Addr47h: Bit [12]-SECTOR_ERASE → 1: Execute the sector erase operation.
Addr47h: Bit [13]-SECTOR_PROTECT → 1: Execute the sector protect operation.

Addr48h: Bit [8]-READ → 1: Executes the read operation from the serial memory device.
Addr48h: Bit [9]-RDEN → 1: Allows read operations to be performed.
Addr48h: Bit [10]-READ_STATUS → 1: Executes the read EPCS status register operation.
Addr48h: Bit [11]-NEGATE_BUSY → 1: Set this bit to ignore busy when reading.

WRITING DATA
Data is written in sectors to the serial memory device. The CPU needs to set bits accordingly.
To send the sector data to the serial memory device, the following bits must be set on Addr 47h:
Bit 9 and Bit 10, when writing intermediate data.
When sending the 256th byte or the last data byte (whichever comes first) the following bits need to be set: Bit 8 and Bit 9.
After each sector is full (256 bytes) then the starting address of the next sector needs to be sent before sending the data (Program Addr45h and Addr46h with the first address of the new sector). Send the sector data as described above.

Reading back data
To read back the data sent to the serial memory device, the CPU sets the address to read from by writing to Addr 46h and Addr47h first and then sets the following bits on Addr 48h.
Bit 8 and Bit 9.
The data is then read from the eight LSB of Addr 48h. Data is read one byte at a time.

SECTOR PROTECT
The serial number should be written in the last sector of the serial memory device (Address 0xFC0000). This section should then be protected to prevent overwriting when the rest of the memory is written. This is done by setting bits: 2, 9 and 13 on reg. 47:

1. Write the bits that protect the last sector (64) as 04h. Set this as the Data (LSB) of reg. 48.
2. Enable the WREN bit 9 and the Sector protect bit 13.

READ STATUS
Reading the status verifies that the sector has been protected. This is done by setting the read_status bit (bit 10) on reg .48h.
The status is read by reading the LSB of Reg 48. To verify the sector has been protected the 8 LSB of reg 48h should be 04h.

To Program the Serial Number in the SD board.

1. The serial number is programmed in sector 64 (starting at address 0xFC0000).
2. The serial number format is three bytes long.
    a. The first byte is the numerical serial number.
    b. The second byte is the Assigned Hall (A, B, C, D) (F starnds for FE group/prototype boards) and Board Version (1 – prototype, 2 – production)..
    c. The third byte is the code version (first nibble is the version: A- production, the second nibble is the version number) – Currently at A4.

After writing the serial number, the memory should be protected from overwrites.