# ATLAS Silicon ReadOut Driver (ROD) Users Manual

| Revision History | |
| --- | --- |
| October 31, 2007 | First draft |
| March 2008 | (V1.63) Second draft, released but in need of more work |
| July – October 2008 | (V1.64) Added details on EFB EV ID Checks, section of Formatter simulators, ROD Event Type Table.  Started indexing the manual. |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

# DRAFT COPY

John Joseph
Richard Jared
Douglas P. Ferguson
Andreas Korn
Trevor Vickey

**Table of Contents**

# 1   ATLAS Silicon Read Out Driver

The ATLAS Silicon Read Out Driver (SiROD) electronics board, used in the ATLAS Off-Detector electronics sub-system, is designed specifically to interface with Silicon Tracker (SCT) and Pixel Front End Detector modules. Each SiROD interfaces directly with one Back-of-Crate (BOC) electronics board that connects to the Inner Detector modules using Fiber Optic links. One Timing Interface Module (TIM) electronics board distributes the global timing and trigger information for up to 16 SiROD modules. The SiROD communicates with a system host through VMEBus, and formatted event data is transmitted off of the SiROD through a serial S-Link card.
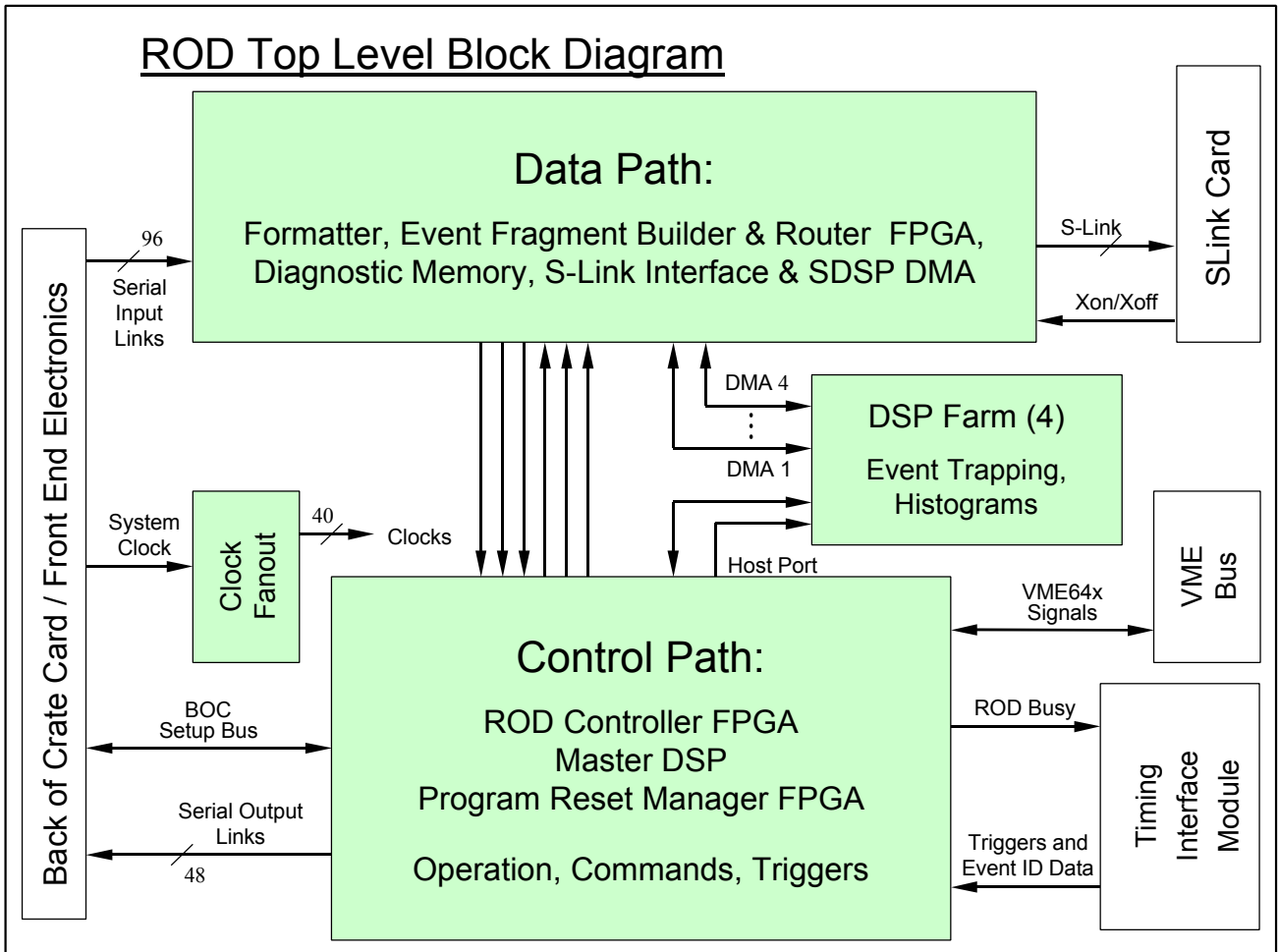
A simplified overview of the SiROD is shown in Figure 1.

## ROD Top Level Block Diagram

**Data Path:**

Formatter, Event Fragment Builder & Router  FPGA,
Diagnostic Memory, S-Link Interface & SDSP DMA

SLink Card

S-Link

Xon/Xoff

Back of Crate Card / Front End Electronics

96
Serial Input Links

System Clock

Clock Fanout

40

Clocks

DMA 4

DMA 1

**DSP Farm (4)**

Event Trapping, Histograms

Host Port

VME Bus

VME64x Signals

BOC Setup Bus

**Control Path:**

ROD Controller FPGA
Master DSP
Program Reset Manager FPGA

Operation, Commands, Triggers

ROD Busy

Triggers and Event ID Data

Timing Interface Module

Serial Output Links

48

**Figure 1: ROD Top Level Block Diagram**

One basic function of the SiROD is to transmit control commands and configuration data through the BOC to the SCT and Pixel FE Detector modules. The transmit data is sent to the modules as serial data streams and can be Level 1 triggers, Event Counter resets, Bunch-Counter resets, calibration commands, or module register data. A second basic function of the SiROD is to receive serial data streams from the FE Modules. This receive data can be event data or register data. The Field Programmable Gate Arrays (FPGA) on the SiROD are designed to detect and mark format errors in the event data, build event fragments, and transmit the event fragments to the Read Out Link (ROL) system and to any combination of 4 Slave Digital Signal Processors (DSP) set to trap event specific data on a non-interfering basis. Module calibration and event data monitoring are some of the functions that are performed in the Slave DSP farm.

In normal running the TIM supplies the trigger and event description data to the SiROD. The trigger is detected by the RCF and expanded into the module trigger commands required by the SCT or Pixel FE modules.   The FE

Trigger code is sent to a 48 wide mask gate that sends the trigger to the active modules. At the start of the run, the Master DSP sets the active links for the default mask. After the SiROD sends the Trigger code to the FE modules, a set of dynamic mode bits are sent to all of the Formatters. After sending the L1 pulse to the SiROD, the TIM transmits serial data that describes the rest of the ATLAS trigger information. When the TIM data for a specific trigger has been transmitted, that data, a dynamic mask and the number of event counter resets are sent to the Event Fragment Builder (EFB) FPGA. After a delay of a few clock ticks to allow the event description data to propagate to the EFB, a token is sent to the Formatters to start the readout of the link FIFOs.

## 1.1  Control Path Description

The primary elements of the SiROD control path are two custom FPGA designs and a fixed point Digital Signal Processor.
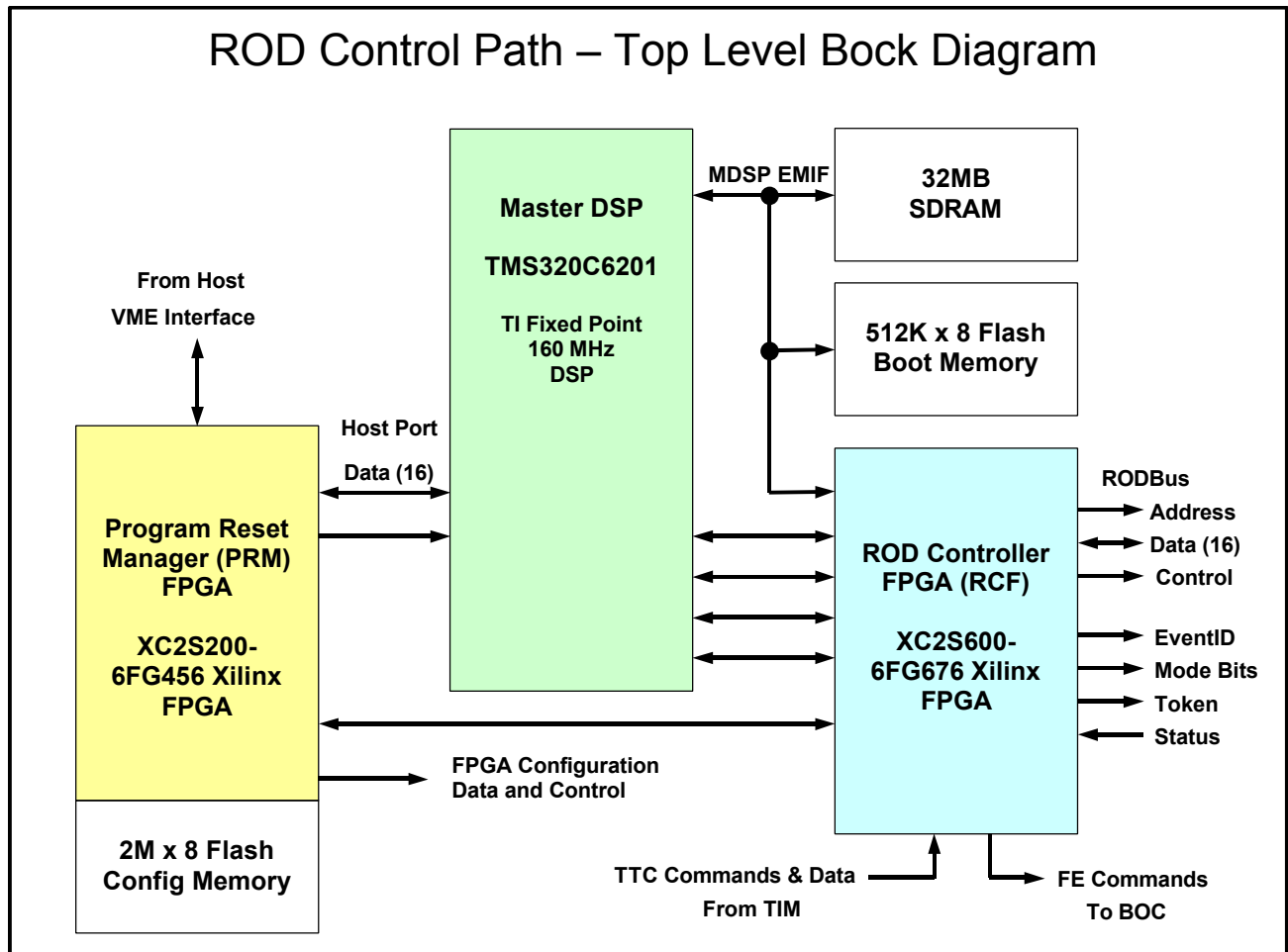


**Figure 2: Control Path Top Level Block Diagram**

### 1.1.1  Program Reset Manager FPGA and VME Interface

The Program Reset Manager (PRM) FPGA controls configuration and reset of the ROD and the data interface with the VME host. When the SiROD is powered on or the front panel reset switch is depressed, the PRM initializes all of the FPGAs and reboots the Master DSP. The ROD Busy signal is asserted to the TIM during reset and configuration.

The VME host communicates with the ROD via an A32/D32 interface using programmed I/O or Direct Memory (Block Transfer) access. The VME host accesses the MDSP and all of the SiROD peripheral devices through a VME to HPI Bridge in the PRM FPGA. The VME host has read/write access to the FPGA Configuration and MDSP Boot Flash memories allowing remote installation of new FPGA and MDSP code images for feature upgrades and repairs. The PRM register map includes FPGA Configuration, FPGA and DSP Reset, and other miscellaneous control and status registers. A complete description of all PRM registers is located in Appendix B. There is a 2-wire interface between the MDSP and the PRM that is routed through the ROD Controller FPGA. It is currently used by the MDSP to check the status of ROD Busy and to send SDSP reset commands if required.

Although the ROD supports VME interrupts, they are not expected to be used. There are no plans at this time to support the full ATLAS CRS space and logical addressing.

### 1.1.1.1  FPGA Configuration and Resets

The primary function of the ROD Program Reset Manager is the configuration and reset management of the ROD electronics board. The PRM allows a flexible combination of reset and reconfiguration possibilities by allowing separate controls for each FPGA and DSP device in the design. The FPGAs can be reconfigured or reset using different commands. A FPGA configuration command will completely erase the device and then transmit the serial configuration data stream at a rate of 20Mbits/second. A FPGA reset command is a very short, 100ns, pulse that resets all of the internal processes to an initial state that is defined in each design. A FPGA reset command will not flush any of the internal FPGA RAM blocks on the ROD.

At power on, the stored FPGA configuration images are loaded into all of the FPGA devices automatically by the PRM. After a new configuration image is programmed into the Configuration Flash memory, the user must then send a configuration command to either reconfigure an individual FPGA or all devices on the board. The reconfiguration process may cause some unexpected register results in some of the FPGAs depending on the order that devices are configured, and for that reason, it is recommended that the user follow a reconfiguration command with FPGA reset command. The FPGA configuration and FPGA reset commands and processes do not have any destructive effect on MDSP or SDSP memory. A detailed description of the configuration and reset command address and bit definitions are in Appendix B

The following process example is for the case that the user is required to clear a fault condition on the ROD without causing any destructive effect on DSP memory.

1. Reconfigure all ROD data path FPGA devices (completely erases current configuration):
    a. VME write to address location 0xC000000, value = 0x20. This will set the "Reconfigure all FPGAs" bit in the PRM FPGA
    b. Process completed check: VME read from address location 0xC00020, when value = 0x1F the operation is complete

2. Reset all ROD data path FPGA devices  (configuration intact, resets all internal registers):
    a. VME write to address location 0xC000004, value = 0x20. This will set the "Reset all FPGAs" bit in the PRM FPGA
    b. Process completed check: VME read from address location 0xC00024, when value = 0x1F the operation is complete

### 1.1.1.2  DSP Reset Commands

The PRM manages the reset signals that are connected to all of the DSPs on the ROD. The PRM allows the user to send reset commands at any time during operation, if a DSP requires a clean reboot of the internal and external program memory. Because the PRM is connected directly to the VME bus, a corrupted DSP that is blocking data access to the data path FPGA devices cannot prevent the host from issuing a reset. This register also allows the user to send a soft reset to all devices on the ROD that are connected to the PRM. The details of the Board and DSP resets are in Appendix B.

### 1.1.1.3  FPGA Flash RD/WR Access

The FPGA Configuration Flash memory is accessed by VME through a set of registers in the PRM FPGA allowing the user to upgrade to new design versions without removing the RODs from service.. Detailed information on all of the PRM Flash registers is located in Appendix B.  Read access is perform by writing the desired memory address to read to the "Flash Address and WR Data Register" and then to set the read control bit in the "Flash Control Register".  The PRM handles all of the read cycle control signals, places the return data into the Flash Data Read Register, and clears the read Flash control bit when the data is ready.

There are two modes for write access to the Flash.  One requires that the VME host send all of the Flash command sequences for erase and programming cycles.  Before the Flash memory can be programmed with configuration data, the appropriate sectors in the device must be erased.  A detailed description of the erase process is in Appendix B.  The other is a fully automated mode that only requires input of the write address location and data from the VME host, and assertion of the PRM write Flash control bit.  This mode is only useful on sectors that have been erased first.  To write a byte to the Flash memory, the address and data is stored in the "Flash Address and WR Data Register" and then the write Flash with PRM control bit is set to value 1.  After the control bit is cleared (self clearing when the cycle is complete) the data write can be verified by polling the read data.  Software programming details can be found in the manufacturer data sheet.

**Configuration Flash Part Number:**
AMD29LV017D, 2Mx8 Flash Memory, 90ns, 10K min cycle, 40-pin TTSOP
Manufacturer: AMD, Spansion

### 1.1.1.4  ROD Busy Module: Busy Duration Buffers

The ROD Busy Module is used to histogram the integrated history of the BUSY duration for each ROD.  The histogram data is stored in the ROD Busy (RB) duration buffers, a circular dual port RAM, in the PRM FPGA.  The details of the registers associated with the ROD Busy Module are located in Appendix B.

The algorithm uses 2 counters to measure the duration of the ROD Busy (RB) signal.  The first is a free running bin counter that counts up to 65636 (~6.5ms), and the second is a gated counter, and only increments if RB is asserted.  If the duration trapping engine is enabled and Trigger Mode is selected, when the "next" trigger is detected on the ROD, the bin counter starts counting.  While the bin counter is active, the RB counter is only incremented by one for each clock that ROD Busy is asserted.  When the bin counter reaches its terminal count, the value in the ROD Busy counter is latched to a holding register and both counters are reset to 0.  The bin counter continues to count, start from 0, and when it reaches its terminal count for the second time, the value in the ROD Busy counter plus the holding register are written the dual port RAM (DPRAM) in the half word order shown in Table 1.  At this time both counters are again set to 0, the DPRAM write address is incremented by 1, and the process continues until the engine is disabled.  The last write address is stored in a PRM register that also tracks the number of times that the DPRAM address has cycled through the buffer start point to accurately point to the current data record.  If the process is set to run in Auto Mode, then the counters starts as soon as the enable bit is set to value 1.

The minimum requirements for the ROD Busy Modules are as follows:

1. 512x16 FIFO
2. Increment a counter for every clock that ROD Busy is asserted
3. Write value every 6.5 ms
4. FIFOs are full after 3.3 sec
5. FIFOs are reset by global command
6. Each of the 16 FIFOs are accessible from VME

The specifications for the ROD Busy Module are shown in Table 1.

**Table 1: ROD Busy Module Specification**

| Requirement | Parameter | |
|---|---|---|
| Memory Size | 1024 x 32 Dual Port RAM | |
| Host Data Access | VME A32/D32 | |
| Full Capture Time | 13 seconds | |
| Full Memory Readout Time | < 275us | |
| Memory Map Address Locations | 0xC01000 to 0xC01FFC | |
| Memory Organization | 2 16-bit bins per address | |
| **Bit Definitions** | | |
| **Location** | **Bits[31:16]** | **Bits[15:0]** |
| 0xC01000 | ROD Busy Count 1 | ROD Busy Count 0 |
| 0xC01004 | ROD Busy Count 3 | ROD Busy Count 2 |
| … | … | … |
| 0xC01FF8 | ROD Busy Count 2045 | ROD Busy Count 2044 |
| 0xC01FFC | ROD Busy Count 2047 | ROD Busy Count 2046 |

**ROD Busy Duration Histogram Setup:**

1. Set Triggering Mode at address 0xC0001C, bit 1
   a. Value = 0, start counters when Histogram control enabled
   b. Value = 1, start counters when next L1 trigger is detected
2. Set Histogram Control Enable at address 0xC0001, bit 0
   a. Value = 0, disable counters
   b. Value = 1, enable counters
3. Monitor Rod Busy Histogram address status, if required, at location 0xC0005C
   a. Bits [9:0] indicate the current data write location in the DPRAM
   b. Bits [15:10] indicate the number of times the address counter has wrapped around 0

## 1.1.1.5  MDSP to PRM Reset Command Interface

The Master DSP can send a very simple serial stream command to the PRM to force a reset on any SDSP. The signal is connected to the FSR_IN(1) port of the MDSP, and is set up to operate by sending a defined number of wide pulses to the PRM to request a SDSP reset. The intention was to allow the MDSP to reset a corrupted SDSP during calibrations if required. The specific details are noted in the MDSP code comments.

## 1.1.2  ROD Controller

The ROD Controller is composed of the Master DSP and ROD Controller FPGA (RCF). The Master DSP runs software to perform system control and non real-time functions and the RCF performs all of the real-time functions. The VME host communicates to the internal ROD and registers and DSP memory locations by using the Host Port Interface (HPI) of the Master DSP.

## 1.1.3  Master Digital Signal Processor (MDSP)

The Master DSP (MDSP) controls and coordinates operation of the entire ROD, although it is not required to run any specialized functions when the ROD is configured in Physics data-taking mode. It provides access to the controller and data path FPGAs, BOC, and all four Slave DSPs (SDSP) through a RCF port that is connected to one of its external memory interfaces. The DSP has two internal 64 KB memory blocks, which are used to store program instructions and data, and an additional external 32 MB of SDRAM, for additional program and general data storage.

While it is idle, the MDSP runs a polling loop that checks the MDSP's communication registers for requests from the host/user. The communication registers are a set of words located in the beginning of the MDSP's internal data memory, which communicate requests from the host (via command registers), and indicate the status of the MDSP (via status registers). To control the MDSP, the user downloads "primitive lists" into a reserved space in slow

memory known as the primitive buffer, and then sets a bit inside the main command register. Primitive lists consist of a header & trailer (used for list coordination and error checking), together with primitive data.

The list contains a series of primitives (subroutines, encoded using identification numbers) together with data for each. Primitives are typically executed once during list execution (though a primitive can repeat while a long operation is pending which requires access to the main polling loop, to use a handshaking process), and are executed serially.  If a primitive is sent to the ROD, the MDSP sets its list busy bit inside the main status register and executes the primitives in the list in series; one primitive is executed for each pass through the main loop.  At the end of the list the MDSP removes the busy bit and sets an acknowledgement bit, signaling to the host that the list is finished. The host then resets the list-ready bit in the command register, and the MDSP finishes the handshake by resetting the acknowledge bit.

A primitive's reply data, if there is any, is stored inside a "reply" primitive list, located in a separate reply buffer. After list execution has been completed, the host retrieves the reply buffer as part of the handshake process with the MDSP. To control operation of the four slave DSPs, there are additionally two private "inter-DSP" primitive and reply buffers.  The MDSP has four text buffers (error, information, diagnostic and transfer buffers); the host using a separate handshake process reads the text buffers. Primitive list execution can be aborted or paused, if needed, using bits within the command register.

The MDSP can also run a separate class of "task" routines.  Tasks are started using a primitive (START_TASK). Once started, they continue executing, once per iteration of the main polling loop, until they complete (they can also be halted manually).  Tasks can be run in "parallel", in that multiple tasks will all be executed during each polling loop iteration. Data output from tasks is retrieved through another primitive (TASK_OPERATION).

The MDSP performs FE Module configuration and calibration using the multi channel buffered serial ports 0 (default) and 1 (corrective) to send data through the 48 bit wide RCF FE Command Mask to any selected FE module from either serial port.

## 1.1.4  ROD Controller FPGA (RCF)

# ROD Controller FPGA

MDSP Serial Port 0 →
MDSP Serial Port 1 →
TTC Cmds & Data →

**Event ID & Trigger Processor** → **FE Command Processor** → FE CMD (48)

**Internal Scan Processor**

**Event Processor** → EFB EVT (16) → FMT MB (12)

**Diagnostics Generator**

**Control & Status Registers**

MDSP EMIF Data (32) ↔
MDSP EMIF Address →
MDSP EMIF Control →

**RCF  RODBus Interface** ↔ RODBus Data (16) → RODBus Address → RODBus Control

MDSP/SDSP Interrupts →
MDSP/SDSP/FPGA GPIO →

**RCF/MDSP Interface** → MDSP/SDSP Interrupts → MDSP/SDSP/FPGA GPIO
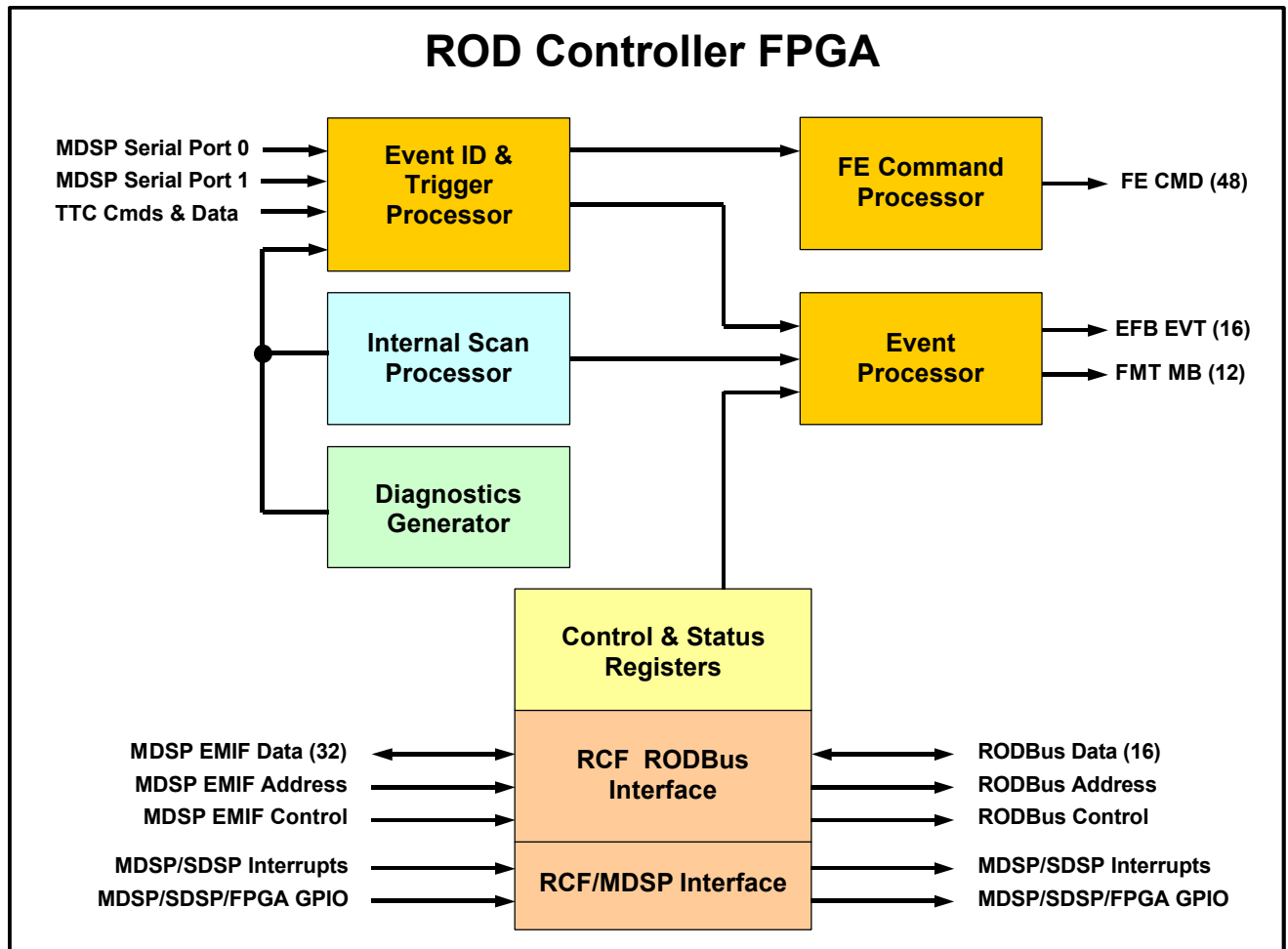
**Figure 3: ROD Controller FPGA Simplified Block Diagram**

The ROD Controller FPGA (RCF) coordinates all of the real-time control operations that are required for normal data taking, module calibration, and on-board diagnostics, connects the data path FPGAs, Slave DSPs, and BOC module to the MDSP, and interfaces with the TIM module for access to the TTC data and commands.  The RCF is composed of Event ID and Trigger Processor, FE Command Processor, Internal Scan Processor, Event Processor, Diagnostics Generator, Control and Status Registers, RODBus Interface, MDSP Interface, direct access to the PRM FPGA.

### 1.1.4.1  Event ID and Trigger Processor

The Event ID and Trigger processor is the primary collection of real-time event building operations in the ROD.  Its primary function is to gather triggers and event description data from all of the trigger sources, format the data, and transmit triggers to the FE Command Processor and data to the Event Processor functions in the RCF FPGA. Trigger sources available to and on the ROD include the TIM module, the Internal Scan Engine, the MDSP, and the Diagnostics Generator.

**TRIGGER AND ID SOURCE: TIM/TTC**
When the TIM is used as the source of trigger and timing signals, process blocks in the RCF, the Trigger Signal Decoder and the FE Command Processor, are used to de-serialize the Event ID information and detect trigger and timing commands that are transmitted to the ROD over the back plane TTC bus.  The list of defined Trigger, Timing

and Control signals and descriptions is shown in Table 2.  The L1ID, BCID, Atlas Event Type and TIM Event Type values are captured in the Trigger Signal Decoder (TSD) functional block.  Trigger and timing commands are captured in the FE Command Processor.

The TIM transmits the L1ID and BCID as a single 36 bit serial data packet on the bussed line /TTC(4), and the event types as a single 10 bit serial data packet on the bussed line /TTC(5).  The serial data packets include a wake-up bit to start the serial parsing state machines in the TSD.  The L1ID/BCID decoder and the Event Type ID decoder are independent blocks that can de-serialize input data from the TIM simultaneously.  When the data packets have been completely shifted into the RCF, the values are stored in a FIFO queue in the order that they are received, and the decoders return to the ready for data state. **If the ROD does not receive the Event ID data from the TIM in Normal Data Taking Mode, the tokens will not be issued to the Formatters, and the ROD will not transfer the Event Data to the S-Link or the Slave DSPs.**

**Table 2: TTC Bus Signal Definitions**

| TTC Bus Signals | Description/Function | **NOTES** |
|---|---|---|
| /TTC(0) | /L1A - L1 Accept Trigger | Active Low, 25ns pulse |
| /TTC(1) | /ECR – Event Counter Reset | Active Low, 25ns pulse |
| /TTC(2) | /BCR – Bunch Counter Reset | Active Low, 25ns pulse |
| /TTC(3) | /CAL – Calibration Strobe | Active Low, 25ns pulse |
| /TTC(4) | Event ID - L1ID and BCID Value | Active Low, Serial Stream |
| /TTC(5) | Event Type – Atlas and TIM Trigger Type | Active Low, Serial Stream |
| /TTC(6) | Not Used, Connected to the Controller | Active Low |
| /TTC(7) | Not Used, Connected to the Controller | Active Low |
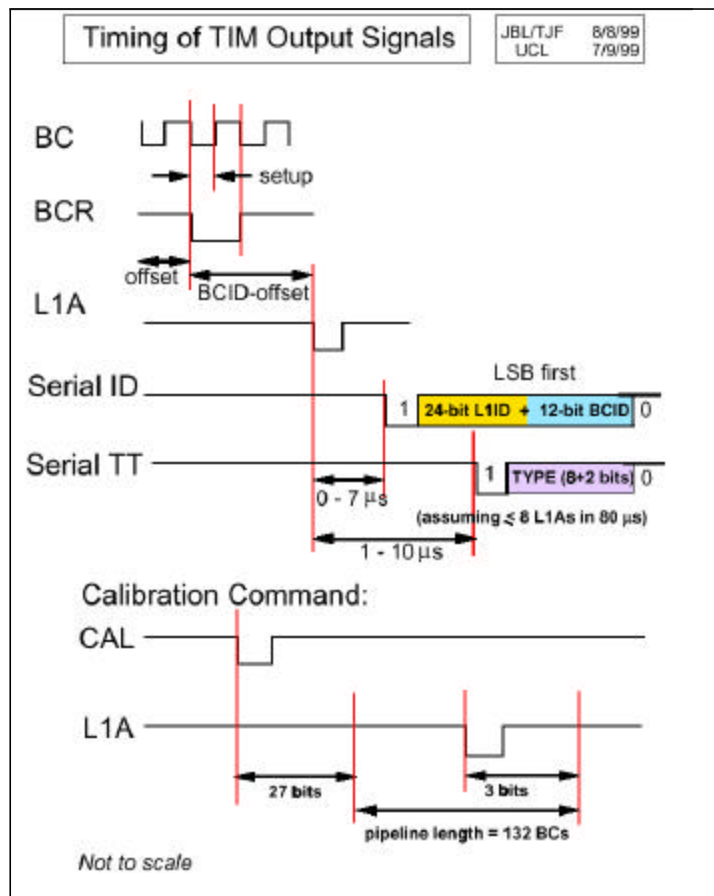


**Figure 4: TIM to ROD Timing**

When triggers are sent using MDSP Serial Port, SP0 or SP1, the FE command types are detected in the Serial Port Trigger Signal Decoder (SPTSD) functional block of the RCF, and the Event ID values stored in the FIFO queue are located in the RCF Memory Map. The MDSP always sends the actual timing command required, so the SPTSD must look at all of the signals that are transmitted by the MDSP. It is important to disable this block when it is not required to detect trigger and timing commands or the SiROD could respond to configuration signals as if they were. When a L1A is detected on one of the serial port inputs, the ID values are loaded into the queue FIFO on the rising edge of the next system clock. Each serial port input has a L1 ID counter. The L1ID counters are reset to 0 when an ECR command is detected, and incremented by 1 when and L1 trigger is detected. The block also contains 2 free running Bunch Counter ID (BCID) counters. When a Bunch Counter Reset (BCR) is detected, the BCID counters are reset to 0. When a L1A is detected, the value of the BCID counter at that time is latched into register and can be used in the Event Header as the event BCID. The L1ID and BCID can also be selected from static registers if the "Static L1ID Enable" bit (RRIF_CMND_1, 0x00404410, Bit 22) or "Static BCID Enable" bit (RRIF_CMND_1, 0x00404410, it 21) is set while in the Module Calibration Mode.

### 1.1.4.2  FE Command Processor

The FE Command Processor accepts trigger and timing commands from the TIM module, MDSP, Internal Scan Engine, Diagnostics Generator and distributes these signals to the BOC module and internally to specific ROD functions. The primary operation of the FE Command Processor block is to detect all input signals that correspond to L1 Accept (L1A) triggers, Event Counter Reset commands, Bunch Counter Reset commands, and Calibration Strobe commands as they arrive on the SiROD

The following table shows the FE Commands generated by the ROD when fast commands are received from the TIM over the TTC bus.

**Table 3: FE Module Trigger and Timing Commands**

| TTC Command | Output to SCT FE Chip | Output to Pixel FE Chip |
|---|---|---|
| **L1A** | "110" | "11010" |
| BCR | "1010010" | "101100001" |
| ECR | "1010100" | "101100010" |
| CAL | Contents from the Cal Command Data Register (0x00404444) in MSB first order, followed by a L1A packet after a preset delay equal to the value stored in the Cal Strobe Delay Register (0x00404440). The SCT Cal Command always begins with the pattern "1010111" | "101100100" followed by a L1A packet after a preset delay equal to the value stored in the Cal Strobe Delay Register (0x00404440). |

The serial communication ports of the Master DSP can be configured to send any pattern out to a FE Module, so these signals are routed directly through the FE Command Processor to the Command Output Masks.

The ROD uses two separate Command Signal Output Masks to send command signals to the FE Modules. A bit in the RCF Main Control Register selects the source used for Output Mask 0. Output Mask 1 is always connected to MDSP SP1. The values set in the FE Command Link Mask Registers determine which FE Modules will receive command streams. At power-on or reset, all of the outputs of both masks are shut off. The bits in the mask registers enable and disable the data outputs to the individual Command Links, and the ROD does not prevent the same mask bit from being set in both Mask Register sets. If this happens, the Command Data Stream could be corrupted by 2 sources OR'd together driving 1 output in the FPGA. The FE Command Link Masks can be set to point to different FE Modules as required by the user. In Normal Data Taking Mode, after writing an appropriate value to the Mask Register, the "New Mask Ready" Bit must be set in the RCF Main Control Register. When the next L1 Trigger is sent to the ROD, the Mask will be updated to the new value, and the "New Mask Ready" bit will clear itself. In modes that do not require L1 Triggers, after the Mask Values have been set, the "Config/Cal Mask Load Enable" and the "New Mask Ready" bits must be set to load the new mask. A look up table (LUT) in the RCF allows the

user to preset up to 8 sets of FE Command Masks for use during module calibration.  The appropriate masks will be loaded into the default and corrective mask registers when the FE_MASK_LUT_SELECT value is set to point at the desired preset masks.

**FE Command Pulse Counter / Command Cycle Generator**

The FE Command Pulse Counter stores the L1 Triggers as they arrive from the TIM, MDSP serial ports, or other trigger sources.   When a L1A is received, an up/down counter is incremented, a pulse is generated that initiates the output of the Formatter Mode Bits, the EFB Dynamic Mask, and an interrupt can be sent to the to the Master DSP to indicate if default or corrective masks have been issued.  This DSP interrupt is disabled by default in the current version of the ROD, but can be enabled in the RCF memory map by setting the appropriate bits in the Spare Control Register (See Appendix A for the bit definitions).  After the Formatters have received the Dynamic Readout Mode Bits and the EFB has received the Dynamic Mask, a token is sent to each bank of Formatters, a pulse is generated to indicate that the Command Cycle has been completed, and the up/down counter is decremented.  Because multiple L1 Triggers can arrive while a command cycle is in progress, the state machine will continue initiating Command Cycles as long as the number of cycles is less than the number of triggers received.  The value of the up/down counter can be monitored in the RCF Main Status Register (0x00404420 Bits[29:22]).  For Diagnostic purposes, any count value can be loaded into the Pulse Counter.

## 1.1.4.3   Event Processor

The main function of the Event Processor in the RCF is to gather and build the event information for each trigger, and distribute the data to the required data path FPGAs dynamically.  The VHDL modules in the RCF used in the Event Processor are the EFB Header ID and Dynamic Mask Encoder, Formatter Read Out Mode Bits Encoder, and the Front End Occupancy Counters.  The EFB Header ID and Dynamic Mask Encoder is used to transmit the trigger information that is sent from the TTC system or generated internally on the ROD to the EFB FPGA prior to the arrival of event data from the FE Modules.  The Formatter Read Out Mode Bits Encoder sends the appropriate Mode Bit mask to the Formatters when the ROD detects a L1 trigger from any source.  The Front End Occupancy Counters are used to monitor the status of the event data as it arrives on the ROD.

**EFB Header/Dynamic Mask Encoder**

The EFB Header/Dynamic Mask Encoder is implemented in the ROD Controller FPGA and consists of 5 FIFOs, 2 Look-Up-Tables (LUT) and a Finite State Machine to control the Event Header building task.  The primary function is to collect all of the trigger description data, format and send the Event ID Header and the default or corrective Dynamic Mask information to the EFB.
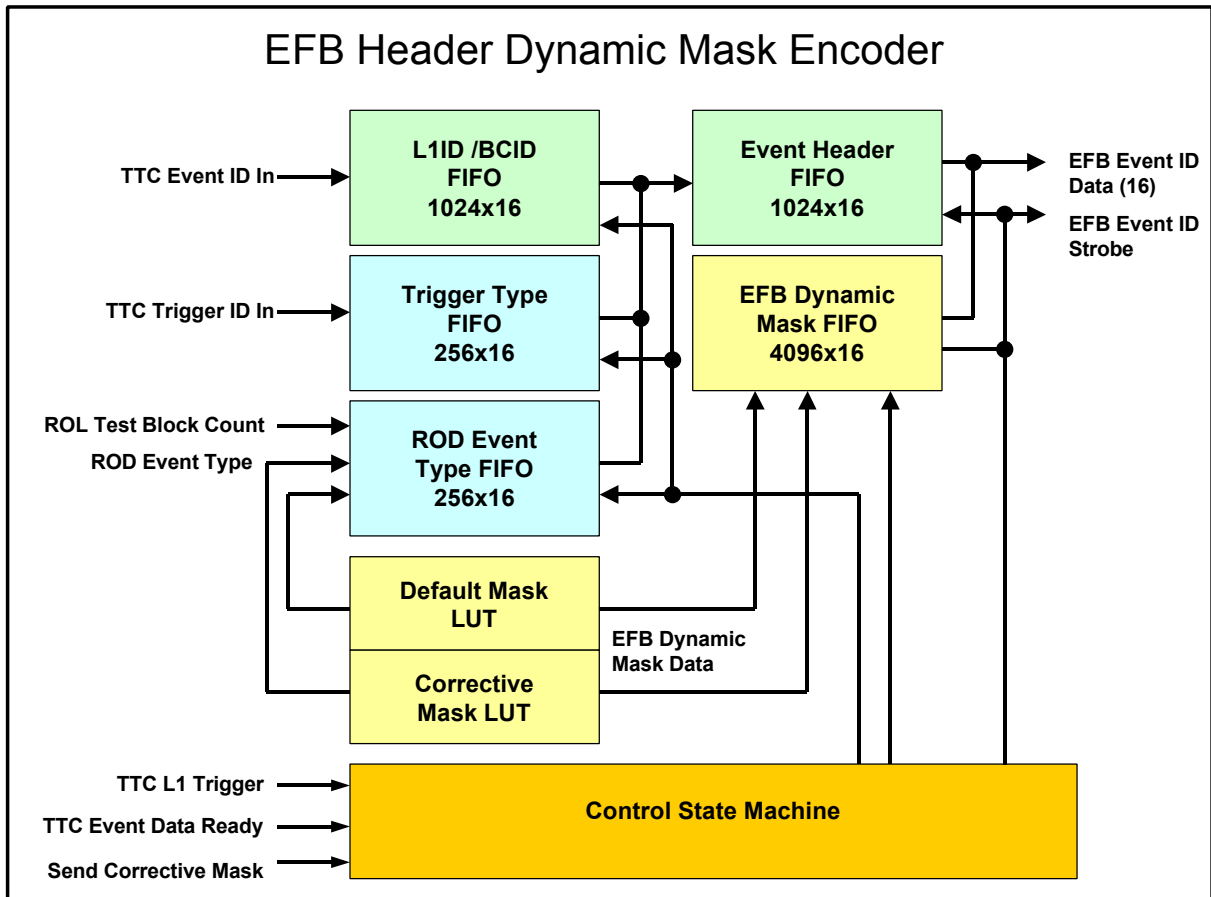
**Figure 5: EFB Header/Dynamic Mask Encoder**

The Default and Corrective Mask LUTs are used as a static source for the mask bits required by the EFB for event processing. The LUTs are addressed as internal memory from either the MDSP or the VME interface.

When a L1 trigger is issued to the ROD by the TIM or the selected calibration trigger source, the Control State Machine starts a

**EFB Header ID and Dynamic Mask:**
The EFB Header ID and Dynamic Mask are stored in internal Block RAM in the ROD Controller FPGA. All values in the LUT are initialized to '0' when the RCF is configured (power-on, hardware reset, or configuration command from the PRM FPGA), but not when the ROD is reset using the "Soft Reset" command. After a "Soft Reset", the values in the LUTs will not change.

The Header ID values are used to generate the Event Fragment Header, and consists of the event specific L1ID, ECRID, BCID, Atlas Trigger Type, TIM Trigger Type, and the ROD Event Type. The source for the Header ID values depends on the current mode of the ROD. If the TTC Trigger Signal Decoder is enabled, the L1ID, BCID, and Atlas and TIM Trigger Types will be sent to the ROD on the TIM TTC Bus inputs and when received, loaded to the Header/Dynamic Mask Output FIFO. A L1A from the TIM will load the ROD Trigger Type from the internal memory location to the Header/Dynamic Mask Output FIFO. If the ROD is using the MDSP Serial Port Outputs (SP0 & SP1) for triggers, the Header ID information is generated using internal counters and registers in the ROD Controller FPGA for each input. The L1ID can be stored in a register or implemented as counters that increment each time a L1A is detected. The BCID value can be a static value, or the value of a dynamic counter latched when a L1A is detected. The Atlas and TIM Trigger Types are stored in 2 registers, one for each trigger source. The Default ROD Event Type Register is used as the ROD Trigger Type when a trigger is detected on SP0, and the Corrective ROD Event Type Register is used as the ROD Trigger Type when a trigger is detected on SP1.

The Event Header ID information, as shown in section 2.2.1, is the first sub-packet of data in the Header/Dynamic Mask data packet, and consists of 4 16-bit words

Each ECR command issued to the FE modules is tracked using a counter in the RCF.  When the ROD receives an ECR from the TIM, the value of the ECR Counter (ECRID) is incremented by one.  The ECRID value is stored in the RCF, and can be monitored by accessing the RRIF_STATUS_0 register.  The ECR counter is reset by writing value '1' to bit 12 of the RRIF_CMND0 register, and loaded with the value stored in the ECR_COUNTER_VAL register (0x00404448) in the RCF.   The default value for the ECR_COUNTER_VAL register is 0 following a ROD reset.  The ECR ID value is transmitted to the Event Header FIFO simultaneously with the L1 ID and BC ID values for each all trigger received or generated by the ROD.

The Dynamic Mask Bits, 2 bits per link, enable dynamic synchronization error correction in the EFB on a link basis. A corrective mask, used for error correction, can be written to the Corrective Dynamic Mask Bits Register, and sent to the EFB when the ROD receives an L1A from the TIM and the "Corrective Mode Bits and Dynamic Mask Ready" bit is set in the RCF Main Control Register (0x00404410, Bit 23).  When a corrective mask is sent to the EFB, the Corrective Dynamic Mask bits are written into the Default Mask LUT.

When the ROD receives a "normal" (no correction required) L1A, the Default Dynamic Mask Registers are sent as the Dynamic Mask.  If the ROD is using the Serial Port Inputs (SP0 & SP1), the Default Dynamic Mask Bits are sent when a trigger is detected on SP0, and the Corrective Dynamic Mask Bits are sent when a trigger is detected on SP1.  The following table shows the definitions of the Dynamic Mask Bits.

**Table 4: EFB Dynamic Mask Bit Definitions**

| DMB[1:0] | EFB Dynamic Mask Bits Definitions |
|---|---|
| 00 | No Change to L1 ID |
| 01 | Increment L1 ID by 1 |
| 01 | Decrement L1 ID by 1 |

The Dynamic Mask Bits information, as shown in section 2.2.1, is the second sub-packet of data in the Header/Dynamic Mask data packet, and consists of 12 16-bit words.  They are shown in the table below.

The status of the internal FIFOs used to queue the Header ID and Dynamic Mask Bits are located in the RCF Register Map at address locations 0x00404458 (1D1-EFB_DM_FIFO_FLAG_STA) and 0x00404458C (1D2-EFB_DM_WC_STA_REG).  The FIFO Status Flags register consists of empty and full flags, for each FIFO used in the functional block.  The FIFO Word Count register consists of occupancy counters for each FIFO. See Appendix A for bit mapping definitions.

**Formatter Read Out Mode Bits:**
The Formatter Read Out Mode Bits are stored in a LUT in internal Block RAM in the RCF.  All values in the LUT are initialized to '0' when the RCF is configured (power-on, hardware reset, or configuration command from the PRM FPGA), but not when the ROD is reset using the "Soft Reset" command.  After a "Soft Reset", the values in the LUTs will not change.

The ROD has 2 sets of Read Out Mode Bit.  The default set, and the Corrective set.  The Read Out Mode Bits are grouped in the memory map in the same order as they are sent to the Formatters.  Mode Bit 0 for Links 0-11 are sent first, then Mode Bit 1 for Links 0-11, then Mode Bit 0 for links 12-23, then Mode Bit 1 for links 12-23, and so on until Mode Bit 1 for links 84-95 is sent.  See Appendix A for bit mapping details.

The Mode Bits enable different Read Out states for the Formatter Link FIFO outputs.  The following table shows the definitions of each different Read Out Mode.

**Table 5: Formatter Dynamic Readout Mode Bits Definitions**

| MB1 | MB0 | Formatter Read Out Mode Bits Definitions |
|---|---|---|
| 0 | 0 | Play 1st event from Link FIFO.  Normal data flow from Formatter to EFB. |
| 0 | 1 | Mask this link for 1 event.  In this mode, the Formatter will dump one event then move on to the next active link. |
| 1 | 0 | Skip read out of this link for 1 event for re-synchronization. |
| 1 | 1 | Read out 1st event from FIFO and dump it on the floor, play the 2nd event to the EFB. |

The corrective mode bits, used for error correction, can be written to the Corrective Formatter Mode Bits LUT, and sent to the Formatters when the ROD receives an L1A from the TIM and the "Corrective Mode Bits and Dynamic Mask Ready" bit is set in the RCF Main Control Register (0x00404410 Bit 23).

The default mode bits are sent to the Formatters for all "normal" (no correction required) triggers received from the TIM.  If the ROD is using the MDSP Serial Port Outputs (SP0 & SP1) for trigger generation, the Default Mode Bits are sent when a trigger is detected on SP0, and the Corrective Mode Bits are sent when a trigger is detected on SP1.

The status of the internal FIFOs used to queue the mode bits are located in the RCF Register Map at address location 0x00404450 (1D0-FRMT_RMB_STATUS).  The register consists of empty and full flags, and an occupancy counter for each FIFO (1 per Formatter Bank). See Appendix A for bit mapping definitions.

**Front End Occupancy Counters**
In the RCF, a counter for each link is use to indicate the occupancy of the FE modules.  The expectation is that the Front End Occupancy Counters are to be used with periodic resets to determine when the ROD can start configuring FE modules that are not acting properly.  The counter can also be used as a diagnostic on the performance of the system, and they are used by the Internal Scan Engine to indicate when a new trigger can be issued to the FE Modules during a calibration run.

The counter increments when a L1 trigger is issued to the FE chips, decrements when a trailer is written in to the Link Formatter FIFO.  The FE Occupancy Counters can be used as a diagnostic tool to monitor the occupancy of the Formatters and the status of data flow from a FE module.  When a L1 Trigger is sent to an active FE Module a 4-bit counter, one for each active link, is incremented by 1.  When the Formatter Link Data Decoder writes a trailer into the Output FIFO, a signal is sent to the ROD Controller FPGA to decrement the Pending Trigger Count for the specific link by 1.  The polling loop for this function is 8 system clocks, and is short enough to insure that a trailer write to the FIFO, which takes 16 clocks, is never missed.  The "FE Occupancy Counters All Zero" bit in the RCF Main Status Register is set if all of the FE Occupancy Counters have the value 0.  The "All Zero" bit can be used to indicate that all modules have finished sending event data to the ROD.  It will have a value of 0 if any Counter has a value greater than 0.  Each link counter can be reset independently of all others by setting the bits in the FE Occupancy Counter Reset Register that correspond to the inactive link.  The counters can be disabled individually by setting the bits in the FE Dead Data Link Mask Register that correspond to links that are turned off at the Formatter.  For diagnostic purposes, the FE Occupancy Counters can be loaded with a pre-set value, the same for all links.

## 1.1.4.4   Internal Scan Processor

The Internal Scan Processor in the RCF provides a speed-optimized system for running calibration scans on the ROD.  The primary structure of the process allows only L1 triggers and CAL commands to be sent on the FE command links, so all ROD setup, FE module configuration data, BCR and ECR commands must be controlled by the MDSP.  The configuration of the Scan Processor is accomplished using VME accessible static registers that allow the user to set the specific scan variables.  The input variables are as follows:
1) **NEvent**: The number of command sequences to issue per mask stage
2) **TrigDelay0**: The time delay between CAL and L1 commands in 25ns increment
3) **TrigDelay1**: The time delay between CAL and L1 commands in 25ns increment
4) **PreCalDelay**: The time delay between first L1 and second L1 commands in 25ns increment
5) **Interval**: The time delay between successive command sequences in 25ns increment
6) **TrigMode**: The type of trigger mode to be used for the scan

There are currently 6 supported types of scan triggering schemes that can be selected to run by the Internal Scan Processor.

1) **Self-Trigger Mode:** Command Sequence – L1A

    If the scan engine is configured in self-trigger mode, the RCF sends an L1A command to arm the FE Module, and data is returned when the module meets its trigger requirement. In this mode, the Timeout limit in the Formatters should be set to a high value to prevent the ROD from returning error events. Self-trigger mode can support up to 4 groups.

2) **Group synchronous scan:** Command sequence - CAL/TD0/L1A

    If the scan engine is configured for group synchronous mode, the RCF will send the first command sequence to all groups in rapid succession. The state machine then waits for the groups to finish returning data by monitoring the FE Occupancy Counters. When the data transmission for a group is complete, the next command sequence is issued to that group.

3) **Fixed interval scan:** Command sequence - CAL/TD0/L1A

    If the scan engine is configured for fixed interval mode, the RCF will send the first command sequence to all groups in rapid succession, and then wait for the time interval counter to finish. When the time interval has expired, the next command sequence is sent.

4) **Fully synchronous scan:** Command sequence - CAL/TD0/L1A

    If the scan engine is configured for fully synchronous mode, the RCF will send the first command sequence to all groups in rapid succession. The state machine then waits for all of the groups to finish returning data by monitoring the FE Occupancy Counters. When the data transmission for all groups is complete, the next command sequence is issued to all groups

5) **Cross-talk Scan:** Command sequence - CAL/CAL

6) **Cross-talk Scan:** Command sequence - CAL/NOCAL

    The cross-talk scan mode must use the default and corrective trigger functions on the ROD to operate correctly. The default/corrective trigger algorithm pairs Group0/Group 8, Group1/Group9, etc and utilizes the FE command mask dynamically to provide 2 potentially different command streams transmitted to different groups with no MDSP intervention.

    > GROUP 0 ➔ CAL/TD0/TRIG on Default path
    > GROUP 8 ➔ CAL/TD1/TRIG on Corrective path

Potential future additions

- **Simultaneous:** Only one trigger group would be defined, and each command sequence would be issued simultaneously to all modules in the group. An additional concept of a readout group would be needed to allow spreading the analysis over all SDSP (each SDSP would analyze only a sub-set of the complete event).

- **Delayed-mode:** Issue command sequences to a series of groups. Each subsequent group in the list has the command sequence issued **n** microseconds after the previous group

## 1.1.4.5  Diagnostics Generator

**Diagnostic Memory FIFO Interface**

The Diagnostic Memory FIFO interface generates control signals that allow the MDSP to access the Diagnostic FIFO Memories that are used in test modes of operation on the ROD. The FIFO interface block is required to translate the 48 bits wide FIFO bus to the 32 bits wide MDSP external memory interface bus, and because of the width mismatch, all data access to the diagnostic FIFOs requires 2 asynchronous memory cycles to fill one memory location.

**Test Bench Interface**

The Test Bench Interface is used to run internal, self-test diagnostics on the ROD that was used primarily in production to verify the function and performance of all of the fabricated RODs. To simulate the normal running modes without the use of FE modules, … n used in testing is to use the TIM FIFO in the controller FPGA as the trigger source and the date receiver FIFOs as the data source. Up to 10 trigger and corresponding data can be loaded in the FIFOs. There is also a counter that determines how many times the FIFOs will be transmitted. This allow for testing of the ROD with trigger rates of 100kHz. Selected data is recorded in the slave DSPs. There is a second variant where triggers come from the serial data stream from the Master DSP and the data from the Data Receiver FIFOs. This variant is slower than the first variant.

**RoL Test Block Trigger/Pattern Generator**
The Readout Link test block is used to send data up the S-Link and to verify the readout bandwidth of the ROD. It can also used to verify that the interface from the EFB and Router through to the Slave DSPs is working properly. The setup and control of the RoL block in the SiROD follows per the ECR specification issued by ATLAS requirements document (ATC-TD-ES-0002), but it can also be operated in enhanced modes. The size of the data word portion of the RoL fragment can be set from 32 to 2048 words long, and the TX state machine can be configured to send a fixed number of fragments in the range from 1 to 16777216 or set to transmit data forever. See section 2.4 to view the output data format of the RoL test block

## 1.1.4.6 Control and Status Register
**Register Block**
The Register Block defines, and allows read/write access to all of the internal registers and functional blocks in the RCF. The internal registers and bit definitions are listed in Appendix A.

## 1.1.4.7 MDSP/RODBus Interface
**Address Decoder / Data Bus Bridge**
The Address Decoder / Data Bus Bridge allows the host CPU and ROD MDSP read and write access to all ROD peripherals and to all BOC registers. The available ROD peripherals include the 4 slave DSPs, all data path FPGAs, all internal diagnostic FIFO memories. The Address Decoder looks at the RODBus address that has been sent to the RCF from the MDSP asynchronous memory address bus and enables the selected peripheral for read or write access. The data bus bridge is connected directly to the MDSP asynchronous memory data bus, a 32-bit structure, and automatically converts all transfers to the native bus width of each attached peripheral. The slave DSPs, Formatter and EFB FPGA require the 32-bit data word to be split into two 16-bit half words that are transferred during one access cycle. The Router FPGA has 16 bit registers, and the BOC has 8 bit registers. The diagnostic FIFOs have varying bus widths that handled by the DSP FIFO access primitive.
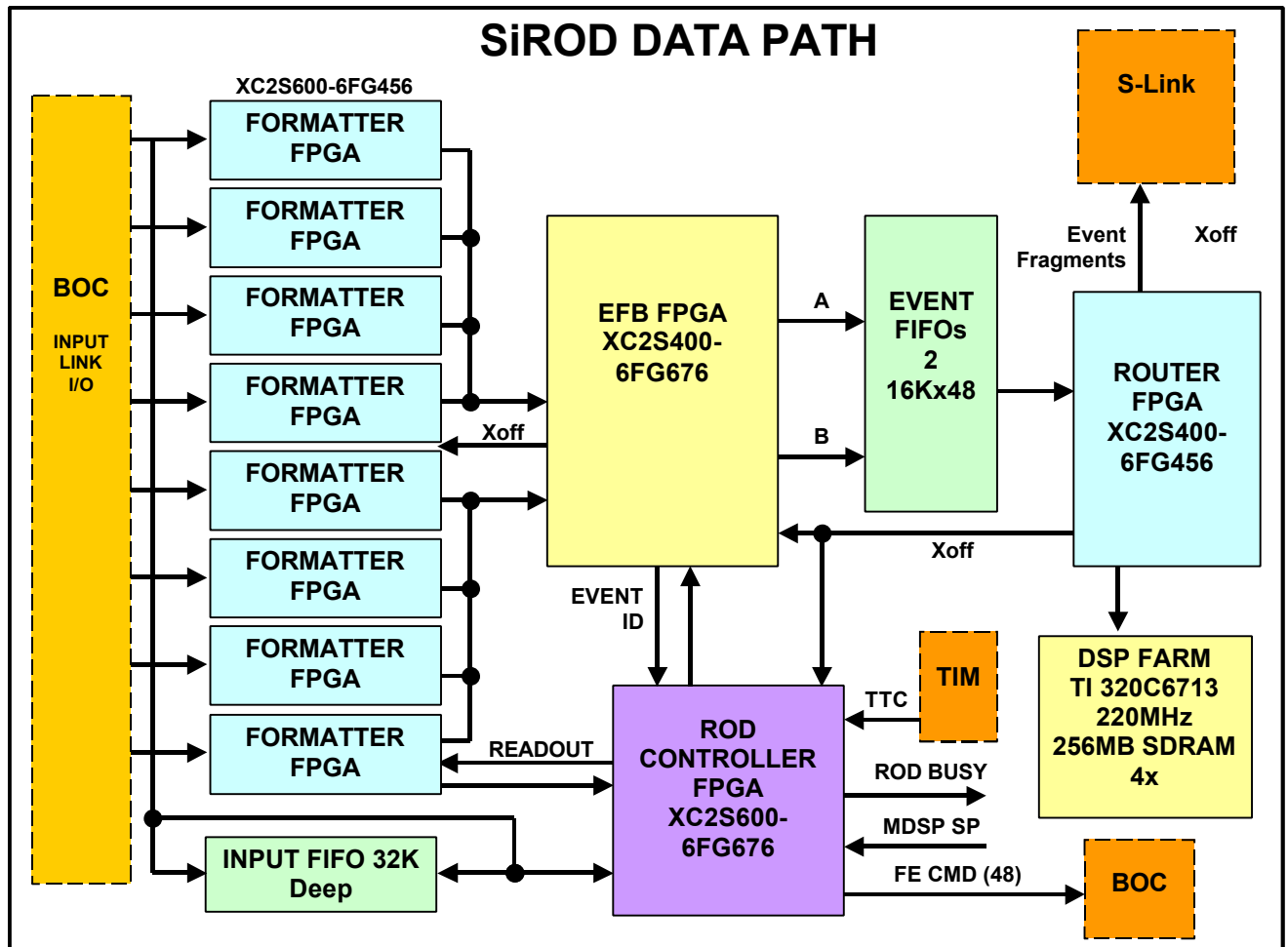
## 1.2   Data Path Description



**SiROD DATA PATH**

XC2S600-6FG456

FORMATTER FPGA (×9)

BOC
INPUT LINK I/O

EFB FPGA XC2S400-6FG676

Xoff

EVENT FIFOs 2 16Kx48

A

B

EVENT ID

ROUTER FPGA XC2S400-6FG456

Event Fragments    Xoff

S-Link

Xoff

DSP FARM TI 320C6713 220MHz 256MB SDRAM 4x

TIM

TTC

ROD CONTROLLER FPGA XC2S600-6FG676

READOUT

ROD BUSY

MDSP SP

FE CMD (48)

INPUT FIFO 32K Deep

BOC

**Figure 6: SiROD Data Path**

### 1.2.1   Input Link Interface (BOC)

The input link interface for each SiROD is the Back of Crate (BOC) optical electronics card in the off-detector electronics crate that transmits 96 links of the serial detector data through a custom back plane.  The input signal to the SiROD is buffered and multiplexed to allow the any incoming data to be mapped to the decoding Formatters and/or to First In/First Out (FIFO) diagnostics memories.  Data can be trapped in normal data taking mode, but cannot be read out until the formatters have processed all of the event data, and triggers have stopped.  When the ROD is in user defined diagnostic or calibration modes, the FIFO memories can be used to trap and read data as required.  The FIFO memories can also be used to read FE module configuration data or BOC calibration signals.

## 1.2.2   Formatter FPGA - Serial Data Decoding and Formatting

The SiROD Formatter FPGAs receive serial data from the input link interface circuit of the ROD.  Each Formatter has 12 serial link inputs, and can convert the serial data from the detector modules to 32 bit word data format and provide de-randomizing buffering for the event fragments for each link in parallel.  The Formatter serial decoder must detect a header pattern in the data stream before the any of the following data and trailers can be decoded.  The Formatters also detect and mark module flagged errors, header and trailer errors, and module data transmission time outs.  Direct connections to the ROD Controller FPGA allow real-time monitoring of link occupancy status, backpressure from the EFB, and ROD Busy status.

As 32 bits words are constructed they are written to an input link de-randomizing FIFO. The maximum depth of the link FIFO was chosen to allow room for the largest calibration data event from 1 trigger.  The FIFO occupancy for each link is stored dynamically in the FMT_LINK_OCC_COUNT() register.  Under normal running conditions with a 100kHz trigger rate, full luminosity and 2 times minimum bias occupancy, the buffers in the SCT Formatters will have an average occupancy of about 20 words and the Pixel Formatter will have an average occupancy of  **????**.  To guard against link FIFO overflow and excessive data loss, there are two user programmable registers that determine when special actions are required in the Formatters.  One is the Header Trailer Limit (HTL) Register that at startup is set at about 7/8 of the maximum link FIFO size.  When the link FIFO occupancy reaches the HTL, the link decoder will only store headers and trailers with a flag set in the data record.  The other is the ROD Busy Limit (RBL) Register that at startup is set at about 15/16 of the maximum link FIFO size. When the occupancy of 1 input link FIFO reaches the RBL, ROD busy is sent to the ATLAS via the TIM.

Differences in SCT and Pixel data formats and sizes require that two separate FPGA designs be maintained.  The SCT version supports 12 link inputs (6 modules) per Formatter, and the Pixel version supports up to 4 modules per Formatter.  The 8 formatter chips are arranged to readout data in two banks of 4 devices.  In this configuration, one of the Formatter is defined by a hardware setting to be the Master Formatter.  This architecture allows the formatters to push the data out at a maximum bandwidth approaching 80MHz, and guarantees that the bandwidth limit of the ROD will be determined by the bandwidth of the S-Link.
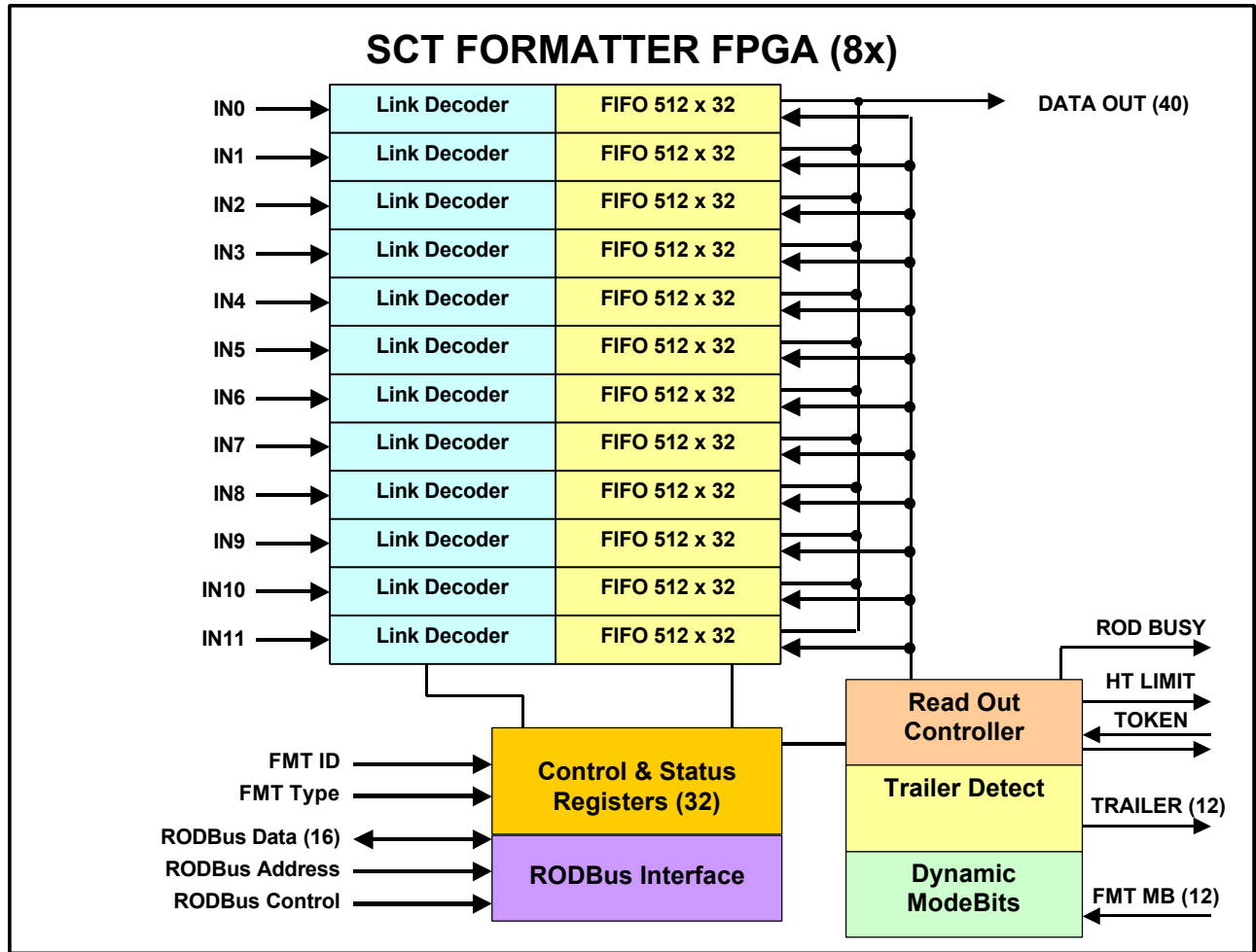
### 1.2.2.1 SCT Formatter



**Figure 7: SCT FORMATTER FPGA TOP LEVEL BLOCK DIAGRAM**

The SCT Formatter is composed of 12 identical serial data decoders, 12 512x32 input link data FIFOs, 12 half clock counters (not shown), a data readout controller, control and status registers, and a ROD Bus communications interface. The SCT formatter decodes and packs the serial data from 12 input links in one of 3 serial decoding modes: Condensed Data, Expanded Data and Configuration Data Mode. Two control bits in the internal Formatter register map, the Expanded Mode and Configuration Mode register bits, allow the user to select the operational serial data-decoding mode of the Formatters. Details of the data Formats for the SCT ROD are in section 2.1.2. Condensed Data mode is the default data-taking format, and one or two hits are always stored in each 16 bit half word of formatted data. In Expanded Data mode two hits can be stored in each 16 bit half word that is not the first hit of a cluster, where only 1 hit can be stored. In Raw or Configuration Data mode, up to 8 bits of data can be stored in each 16 bit half word. If there is a framing error in the serial data during an event, the formatter will record the error and change to raw data mode until a trailer is detected on the errant link. The 12 half clock counters can be used as a coarse timing check of the input signals sent by the BOC. The Read Out Controller is used to synchronize the readout of all of the internal FIFOs in each Formatter.
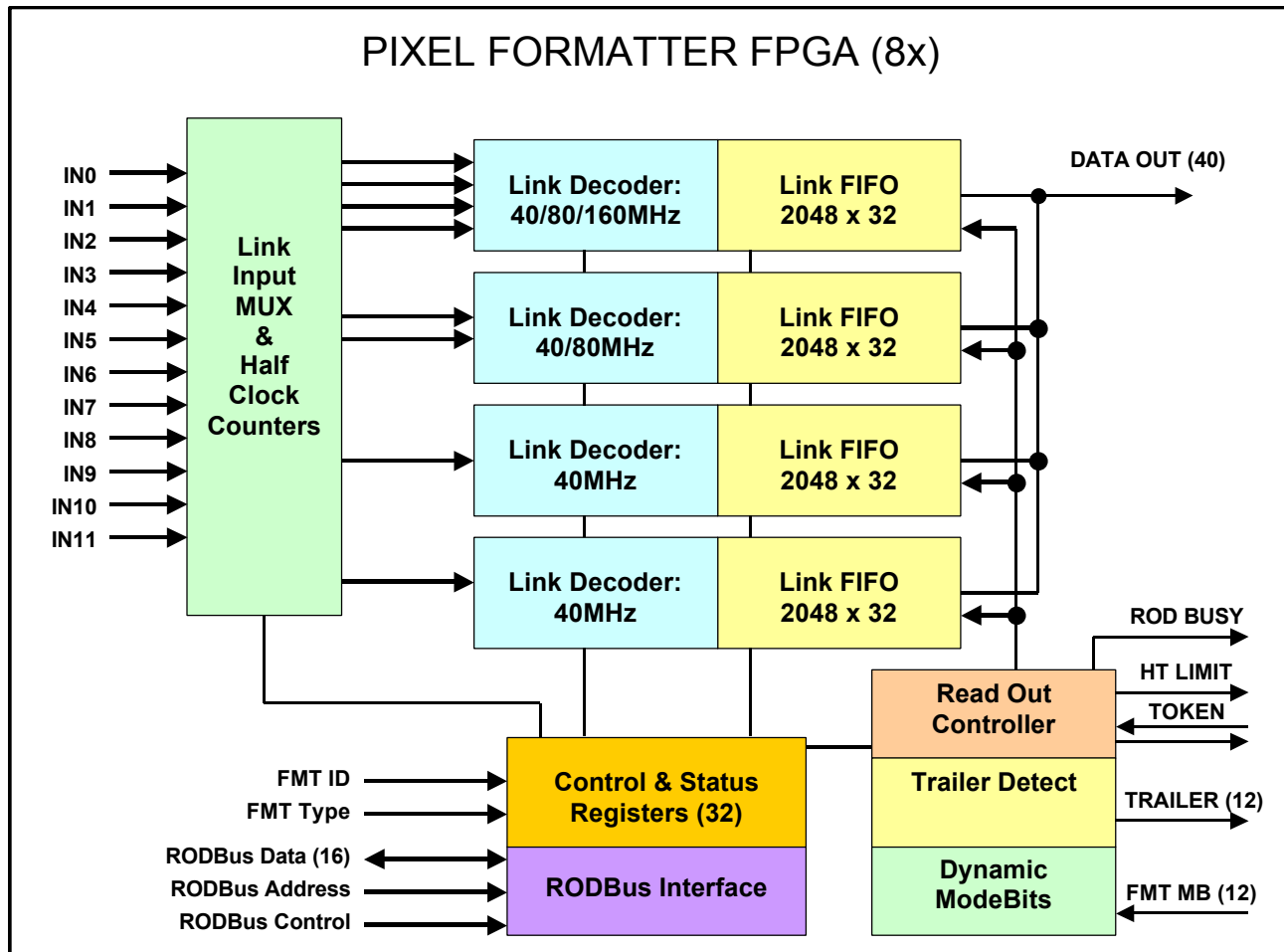
## 1.2.2.2 Pixel Formatter



**Figure 8: PIXEL FORMATTER FPGA TOP LEVEL BLOCK DIAGRAM**

The Pixel Formatter is composed of a 12 to 4 input signal multiplexer, 4 half clock counters, 4 configurable serial data decoders, 4 2048x32 input link data FIFOs, a data readout controller, control and status registers, and a ROD Bus communications interface. The Pixel formatter decodes and packs the serial data from up to 4 FE Modules in one of 3 serial decoding speeds: 40MHz, 80MHz, and 160MHz. The decoding speed can be set using hardware jumpers or by software control. The number of FE Modules serviced by a Formatter depends on the decoding speed selected. A Pixel Formatter is designed to service one 160MHz module, two 80MHz Modules, or 4 40MHz Modules. The Pixel Formatter always stores a full 32-bit header and 32-bit trailer for each module that returns data. Details of the data Formats for the Pixel ROD are in section 2.1.4. In addition to normal data capture, the Pixel Formatter also supports full raw data capture at 40MHz and 80 MHz. In Raw or Configuration Data mode, up to 24 bits of data can be stored in each 32 bit half word. If there is a framing error in the serial data during an event, the formatter will record the error and change to raw data mode until a trailer is detected on the errant link. The 4 half clock counters can be used as a coarse timing check of the input signals sent by the BOC. The Read Out Controller is used to synchronize the readout of all of the internal FIFOs in each Formatter.

## 1.2.2.3 Error Detection

The Formatter can detect errors in the serial data stream and mark the status in the data record in each event. The list of detectable errors includes Header errors, Trailer errors, Condensed Mode data errors, Serial Data Framing (Sync Bit) errors, Readout Timeout errors, Header Trailer Limit errors, ROD Busy Limit errors and Link Data FIFO Overflow errors. See section 2.1 for the location of error bits in the data stream.

**Header Errors:**

The SCT FE modules output a link data header at the start of each data packet transmitted to the ROD. One changed bit in any position is accepted as a header, but if an error condition is met, then the header error bit is set in the link header. Special care has been taken to insure the there is one set of link data per ATLAS trigger. A one-bit error is allowed for in the header or trailer so that data will be decoded in the formatters if there are transmission errors.

**Table 6: SCT Link Header Combinations**

| SCT Headers Accepted: | | |
|---|---|---|
| Data Pattern | Header Detected | Header Error |
| 0111010 | Yes | No |
| 1111010 | Yes | Yes |
| 0011010 | Yes | Yes |
| 0101010 | No | No |
| 0110010 | Yes | Yes |
| 0111110 | Yes | Yes |
| 0111000 | Yes | Yes |
| 0111011 | Yes | Yes |

**Table 7: Pixel Link Header Combinations**

| Pixel Headers Accepted: | | |
|---|---|---|
| Data Pattern | Header Detected | Header Error |
| 011101 | Yes | No |
| 011100 | Yes | Yes |
| 011111 | Yes | Yes |
| 011001 | Yes | Yes |
| 010101 | Yes | Yes |
| 001101 | Yes | Yes |
| 111101 | Yes | Yes |

**Trailer Errors**:

The FE modules output a trailer at the end of each event data packet to indicate that all data has been transmitted to the ROD. The SCT FE module link trailer is defined in the ABCD specification as 1000000000000000. If any one bit in the pattern of zeros is detected as a 1, the decoder will set the trailer error flag in the link trailer, and then reset to start looking for the next link header.

The Pixel FE module trailer is defined as a 100000000000000000000000. The Pixel ROD does not check for trailer errors because the case Row = 0, Column = 0 and TOT = 0 would correspond to a trailer with an error.

**Serial Data Framing (Sync Bit) Errors:**

If one of the field separator bits (sync bit) in the serial return data stream is not detected in its specified location, the link decoder will switch to raw data decoding mode until a trailer is detected. After a trailer is detected, the decoder resets to the selected decoding mode for the next set of hit data from the FE module. When a Sync Bit error is detected, the Formatter will set the "sync error" bit in the link header word and clear the "normal data mode" bit in the FMT_DATA_FMT_STATUS() register. The status register bit will remain set until the register is read, at that point, the link decoder status is checked and the bit will reset to 1 if in normal data mode, or latch a 0 if in raw data mode.

- **Note on excessive Optical Link noise:**

If the input optical links produce random noise at the input of the SCT Formatter while the ROD is configured to process Physics Data, the Formatter link decoders will respond to prevent the ROD from asserting ROD Busy during the run. If the Formatter Link Decoder (FLD) detects a Serial Data Framing Error while configured for Expanded or Condensed mode serial stream decoding, the FLD will move into the Raw Data decoding mode, but will automatically truncate the event after 2 to 3 raw data words have been written into the link FIFO. If this problem is detected on the ROD, the error flags in the Event Fragment will indicate that there is a link with a Sync Error (Bit 19) and a Data Overflow Error (Bit 17). If a trailer has been detected, the link decoder will reset to normal operation and start looking for the next link header to arrive. If there is no trailer, the link decoder will stop writing data to the link FIFO, and this link will produce a Timeout Error (Bit 2) for all triggers until a trailer is detected. For this mode to operate correctly, the Readout Timeout limit must be set to about 2 to 3 us maximum.

**Condensed Mode Errors:**
This error detection is only active when the Formatter is set in Condensed Data mode. A bit in the FMT_EDGE_MODE_EN() register enables this test for each input link. When the Edge Detect Mode bit is set for a link, the decoder will test the hit data to verify if the module is sending data in the Edge Mode or Level Mode format. If the Formatter is set to decode in Condensed Data mode and the data from the module is not consistent with the hit pattern expected, then the Condensed Mode Error bit will be set in the data record that is stored in the link FIFO for readout.

**Link Readout Timeout Errors:**
Link readout timeout errors indicate in the data record that a FE module is active and its Formatter link is enabled, but did not send data for a specific trigger. The value for the Link Readout Timeout Error limit is a user settable parameter that determines the amount of time that the formatter will wait to readout a link that has not received data from the FE module. If a link has not received data in the time set in the time timeout register, a header and trailer will be written to the link FIFO with a link time out error code.

**Header Trailer Limit (HTL) Errors:**
Header Trailer Limit (HTL) errors indicate in the data record that the amount of returned link data caused the link FIFO occupancy to reach the user-defined value for HTL. If the HTL condition is true, the Formatter will change to Header Trailer Limit mode operation. If the Formatter is currently receiving an event when the HTL condition occurs, the decoder will stop writing data to the FIFO until a trailer is detected and stored in the link FIFO with the HTL error bit set. When the next link header is detected, the link decoder will store it in a register, wait for the trailer to be detected, add the error bit, and write both half words to the link FIFO. The HTL Error Status bit for each link is latched whenever the HTL condition is met and held until the host reads from the FMT_HEADER_TRAILER_ERR() register.

**ROD Busy Limit (RBL) Errors:**
ROD Busy Limit (RBL) Errors indicate that the occupancy of the link FIFOs is equal to the user-defined value for RBL. There is no flag in the data record for RBL, instead, the condition is flagged and latched in the Formatter FMT_ROD_BUSY_ERR() register until read by the host. If the RBL condition is true, the Formatter will assert the Formatter ROD Busy signals that connect directly to the RCF and sent to the TIM to indicate to the TTC system that there is the potential for data loss on one or more RODs. If events continue to arrive on the Formatters, data loss will occur, a fatal error will be generated and the ROD will stop sending event data to the S-Link.

**Link Data FIFO Overflow errors:**
Link Data Overflow errors indicate that the amount of data received from a FE Module is larger than a user settable limit. If the event exceed the limit, then a bit is latched in the FMT_DATA_OVERFLOW_ERROR() register until read by the host. This is a historic state that is not longer used in the Formatters.

## 1.2.2.4  Data Readout

The Formatter Read Out Controller controls the data readout to the ROD data path. A trigger from the RCF that initiates the Token passing process in the Read Out Controller starts the data readout of the Formatters. When the Master Formatter in a block of 4 is finished sending all of the data that corresponds to a specific trigger, the Token is passed to the next Formatter. When the last Formatter of a block is finished with the data readout, the Token is returned to the Master Formatter and if a trigger is pending, a new readout cycle is initiated

Prior to the arrival of a readout Token, the Formatter receives the Dynamic Mode Bits (DMB) from the RCF to configure the readout state for each link. There are 2 bits per link, and they are transmitted to the Formatter each time the ROD receives an L1 trigger. The different readout modes are defined in Table 8.

**Table 8: Dynamic Mode Bit Definitions**

| Mode Bits | Mode Bit Definitions | Uses |
|---|---|---|
| 00 | Normal data readout | All modes |
| 01 | Skip link readout | Resynchronization to ATLAS Triggers Module Calibration |
| 10 | Dump one event of link data | |
| 11 | Dump first event, readout second | Resynchronization to ATLAS Triggers |

After receiving the Dynamic Mode Bits the Read Out Controller (ROC) waits for a trigger to arrive from the RCF. When the Master Formatter receives the trigger, it starts the Token passing process and sends formatted link data to the EFB per the configuration of the Mode Bits. The Token is only held and used by links that are enabled. If no data is present in a link FIFO that is holding the Token, the readout timeout counter will increment until either data is received from the FE Module or the timeout limit is met. If the timeout limit is met, then the ROC sends a timeout error data word to the EFB and passes the Token to the next active link. When the ROC detects that a trailer has been sent from a link, a bit is set in the Trailer Detect block that is sent to the RCF for processing and the Token is passed to next active link.

**PIXEL SPECIFIC CASES:**
Because the Pixel FE module can send from 1 to 16 events for each L1A trigger sent, the data link FIFO controller in the Pixel Formatter uses the value stored in the FMT_PXL_LINK_L1A_CNT() to readout the correct number of events from the data link FIFO. The data link FIFO controller keeps track of the number of trailers that exit the FIFO and compares the L1ID in each header with the current value obtained from the previous header. If the number of trailers matches the requested number of consecutive L1A triggers or if the L1ID changes before the number of events match is complete, then the controller will terminate the event readout to the FIFO Read Out Controller. In this mode, the Formatter also includes a BCID offset value in the data record that is used by the EFB when it checks the Event ID information from each link to guarantee that all BCID values are sequential for consecutive triggers..

There is also a Pixel specific mode that is implemented to handle the case where a Pixel MCC module receives more than 16 events, in a short time window, from the FE modules causing an overflow of the internal L1ID buffers. If a Pixel FE Module sends an MCC empty event indicator flag, the data link FIFO controller extracts and saves the number of lost events from the appropriate header and inserts the required number of empty events into the data record. The inserted empty events will contain the correct L1ID and sequential BCID and will contain a flag in the header that will indicate the number of events that were skipped for the current L1 trigger. This process performs the required function of keeping events in synchronization and informs the offline system that an error has occurred on a module.

## 1.2.2.5   Data Stream Simulator

The Data Stream Simulator is a VHDL code package that produces configurable Pixel MCC and SCT FE output data streams inside the ROD formatter FPGAs to test and debug the readout chain. For each detector, the implementation is intended to exercise the Readout chain and test analysis packages. The simulator engine is controlled via internal Formatter registers that can be set once during a run, and will produce module data for each L1 trigger as long as the formatters and the Simulator configuration register are not reset. To send a valid event into the formatters and through the entire ROD data path, including synchronous event information, ECR and BCR forwarding from the ROD Controller FPGA is required.

The configuration settings for the Pixel Data Stream Simulator are shown in Table 9 and Table 10. The Simulator acts the same as a FE module when receiving ECR, BCR and L1A trigger command. There are 3 counters in the Formatter register map that are used to count L1 triggers, ECR commands, and BCR commands. After reset, the L1A counter is initialized to –1 and the ECR and BCR counters to 0. Hit data, when requested, will always appear in the middle BC of any stretched trigger event, and the MCC skipped event configuration requires that the sum of the NA and MCC skipped event fields in the Simulator Configuration Register 1 is equal to the NA value in the Formatter FMT_PXL_LINK_L1A_CNT register. With the TIM as the trigger source for the modules and ROD, 0xFFh is the BCID offset value required in the EFB to prevent error flags during ID checks.

**Table 9: Pixel Formatter Simulator Configuration Register 1**

| Bits: | Value: | Meaning: |
|---|---|---|
| Config_in1[0] | '0'<br>'1' | Simulation disabled<br>Simulation enabled |
| Config_in1[2-1] | '00' & '11'<br>'01'<br>'10' | 40 Mbit/s mode<br>80 Mbit/s mode<br>160 Mbit/s mode |
| Config_in1[3] | '0'<br>'1' | Single simulator mode<br>Multiple simulator mode |
| Config_in1[7-4] | '0000'<br>'0001'<br>...<br>'1111' | LVL1A = 1, Width = 1 BC value<br>LVL1A = 2, Width = 2 BC values<br>...<br>LVL1A = 16, Width = 16 BC values |
| Config_in1[8] | | Not used |
| Config_in1[12-9] | | Number MCC Skipped Events (The sum of this and the LVL1A value must be equal to the value of the FMT_PXL_LINK_L1A_CNT register in the Formatter). The skipped events generated by the sim will always be in one L1 event. |
| Config_in1[15-13] | '000'<br>'001'<br>'010'<br>'011'<br>'100'<br>'101'<br>'110'<br>'111' | Nomal header: '11101'<br>Bit-flip header: '11100'<br>Bit-flip header: '11111'<br>Bit-flip header: '11001'<br>Bit-flip header: '10101'<br>Bit-flip header: '01101'<br>Bit-flip header: '111101' |
| Config_in1[23-16] | | 8-bit MCC flags |
| Config_in1[31-24] | | 8-bit FE flags |

**Table 10: Pixel Formatter Simulator Configuration Register 2**

| Bits: | Value: | Meaning: |
|---|---|---|
| Config_in2[2-0] | '000'<br>'001'<br>'010' | No bit-flip<br>Bit-flip all<br>Periodic (bit-flip setting) bit-flips |
| Config_in2[23-3] | | 21-bit bit-flip setting (cut down for different usages during M5) |
| Config_in2[31-24] | | Number of hits to be generated |

**Table 11: SCT Formatter Simulator Configuration Register**

| Bits: | Value: | Meaning: |
|---|---|---|
| Config_in1[0] | '0' | Simulation disabled |
| | '1' | Simulation enabled |
| Config_in1[1] | '0' | CLK/2 mode disabled |
| | '1' | CLK/2 mode enabled |
| Config_in1[2] | '0' | Debug mode disabled (Write BCID & L1ID to hit data) |
| | '1' | Debug mode enabled |
| Config_in1[7-3] | | Custom Header (field value translates to link header) |
| Config_in1[9-8] | | Hit Probability |
| | '00' | Always send a hit |
| | '01' | Half hit / half empty |
| | '10' | Mostly empty, some hits (1 per strip every 128 events) |
| | '11' | Always an empty event |
| Config_in1[11-10] | | Error Probability |
| | '00' | Always send hit/empty, so do nothing |
| | '01' | Mostly OK, but some errors |
| | '10' | Half OK / half error |
| | '11' | Always an error |
| Config_in1[15-13] | | Number of chips ("0000" = 1 chip) |
| Config_in1[22-16] | | Number of Hits/Chip |
| Config_in1[25-23] | | Hit Map |
| | '000' | Fixed at 011 |
| | '001' | Fixed at 010 |
| | '010' | Detector alignment 1xx |
| | '011' | Detector alignment x1x |
| | '100' | Detector alignment xx1 |
| | '101' | Level mode x1x |
| | '110' | Edge mode 01x |
| | '111' | Test mode xxx |
| Config_in1[27-26] | | Cluster Size |
| Config_in1[29-28] | | Bit Flip Probability in Link 0 |
| Config_in1[31-30] | | Bit Flip Probability in Link 1 |

## 1.2.2.6  Setup Configuration & Status

A complete and detailed listing of all ROD Formatter registers is located in Appendix A.  In normal operation the user should setup the following registers via the ROD Controller to control the Formatter as required.

**COMMON CONFIGURATION**
- FMT_LINK_EN() – Enables each input link
  - Default – All links off
- FMT_CONFIG_MODE_EN() – Selects normal or raw serial data decoding mode
  - Default – Normal decoding mode
- FMT_READOUT_TIMEOUT() – Sets the Readout Timeout Limit Value
  - Default – SCT = 6.4us, Pixel = 1.6ms
- FMT_HEADER_TRAILER_LIMIT() – Sets the Header Trailer limit value
  - Default – SCT = 0x1C0, Pixel – 0x700 words
- FMT_ROD_BUSY_LIMIT() – Sets the ROD Busy limit value
  - Default – SCT = 0x1F0, Pixel = 0x600 words

**SCT SPECIFIC CONFIGURATION**
- FMT_EXP_MODE_EN() – Selects condensed or expanded serial data decoding mode
  - Default – Condensed Mode
- FMT_EDGE_MODE_EN() – Selects Condensed Mode Error checking mode
  - Default – Level Mode detect

**PIXEL SPECIFIC CONFIGURATION:**

- FMT_PXL_LINK_L1A_CNT() – Sets value of expect L1 accepts from Pixel FE module
  - Default – "0"
- FMT_PXL_BANDWIDTH() – Software selection of Pixel decoding mode: Bits[1:0]
  - "00" – Single Link 40MHz Decoder (default value)
  - "01" – Dual Link 80MHz input
  - "10" – Quad Link 160MHz input


**SPECIAL DIAGNOSTIC CONFIGURATIONS:**

**BOC Timing Fast Scan:**
When the BOC is configured to send a 20MHz clock at a defined phase into the ROD, the half clock counter block counts the number of 1s received during the active count time and that value can be used to determine the BOC timing settings. This is configured in the Formatters as follows:

1. Write the Global Counter Value to bits [15:0] in the FMT_GLB_COUNTER() register. This value is only loaded to the global counter if the BOC scan engine is in reset (Enable Bit low).
2. Set the BOC Scan Control Enable Bit to value 1 in the FMT_CNT_COUNTER() register. If the input multiplexer on the ROD is routing signals to the Formatters, then data will cause the data link counter to increment. When the value in the Current Count bits of the FMT_GLB_COUNTER() register is equal to 0, the link bit counters will stop looking at the data stream.
3. Read out the counters: FMT_STREAM_01(), FMT_STREAM_23(), FMT_STREAM_45(), FMT_STREAM_67(), FMT_STREAM_89(), FMT_STREAM_AB()
4. The clear the data link counters, clear the Enable Bit. This does not reset the Global Count Load value

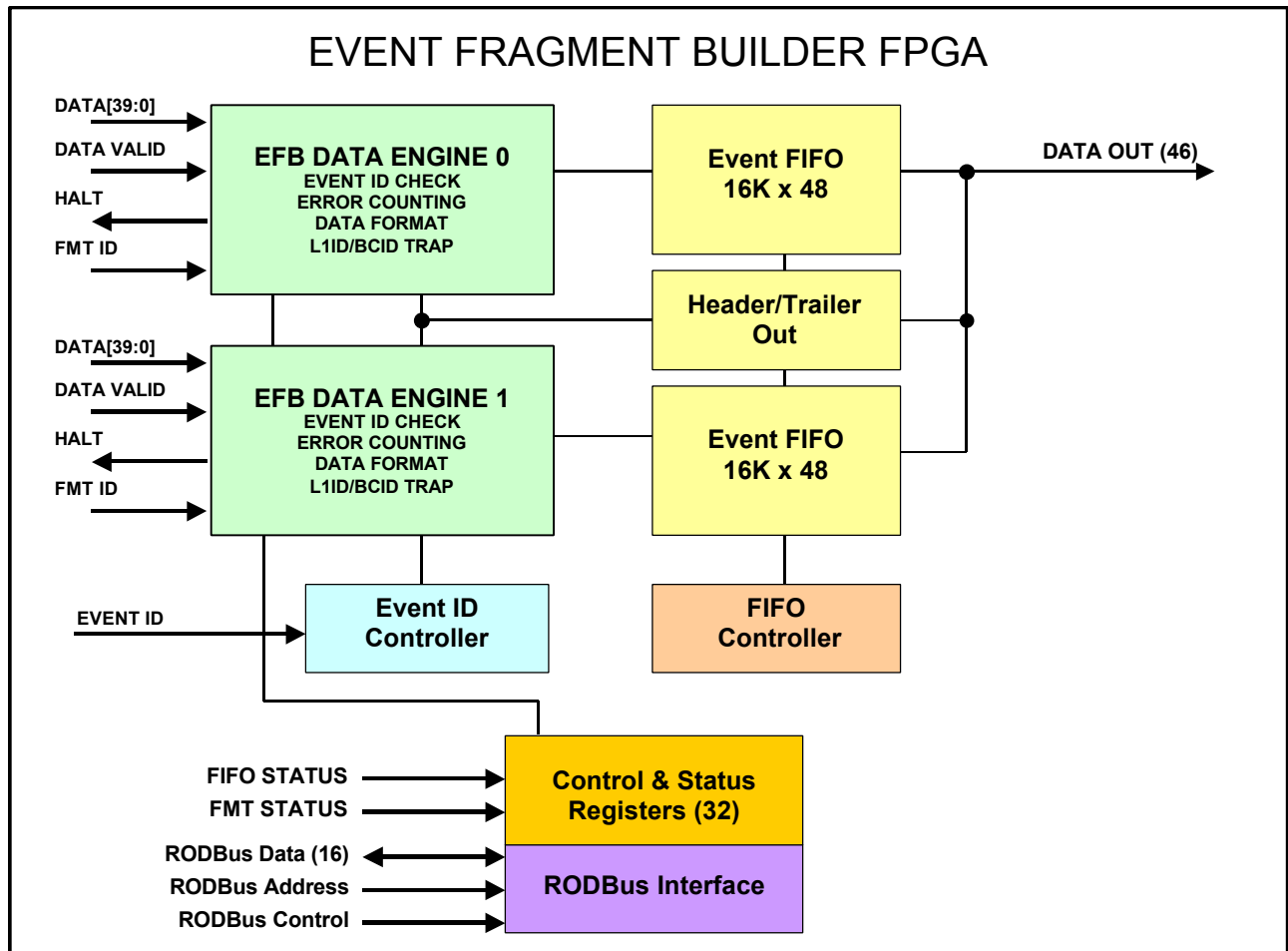## 1.2.3 Event Fragment Builder (EFB) FPGA and Event Memory



**Figure 9: EFB FPGA TOP LEVEL BLOCK DIAGRAM**

### 1.2.3.1 EFB FPGA Overview

The primary functions of the Event Fragment Builder (EFB) on the SiROD are to generate the ROD Event Fragments per Atlas requirements (reference the EV doc) and flag and count errors in the event data stream. The EFB has two processing engines that collect the data output simultaneously from each bank of Formatters to increase the efficiency of the ROD data path, and allows the processing bandwidth to approach 80MHz. The event data stream is monitored for errors, including L1ID abd BCID synchronization verification, then built into an Event Fragment that is stored in a 16K word deep FIFO until the complete fragment is ready to send to the S-Link. The EFB monitors a static bit that is set by the Formatter, and determined by the FPGA binary to identify the data structure, SCT or Pixel, on the data path.

The event processing cycle begins when the event descriptor data is received from the ROD Controller FPGA (RCF). This packet of data is the trigger identification information, and includes the L1ID, ECRID, BCID, all specified Trigger Types (Atlas and detector specific), and a set of link specific Dynamic Mask bits that initiate special functions in the data processing engine (see Section 2.2.1). The EFB has a set of dedicated internal de-randomizing FIFOs that can store enough Event ID data for 512 triggers.

## 1.2.3.2 Event Synchronization Verification and Error Detection

The input data is pushed from the Formatters to the EFB. The first block in the EFB checks the Dynamic Mask bits to determine if the module L1ID number is to be incremented, decremented or unchanged (this feature is used in the in link resynchronization task). The module L1ID and BCID values are then checked against the ATLAS L1ID and BCID. If the values in the link headers do not match those supplied by the TTC System, errors will be flagged and sent along in parallel with the link headers in the event data stream. The next block in the EFB processing engine looks through the data in real time to flag and count any errors in the byte stream pipeline. There is a 32-bit register for each link that allows specific errors to be masked of on a link-by-link basis. A table of the errors flagged and counted on the ROD is located in Table 31 in Section 2.3.3.1. The address location of the mask registers is located in Appendix A. Errors are counted for each link, but the link that produced the error is not noted in the Event Fragment. A flag for each error in the event is set and transmitted in ERROR STATUS WORD 0, and the total number of errors in the event is transmitted in ERROR STATUS WORD 1 of the Event Fragment Trailer.

After the data passes through the error detection block, the event is formatted per each detector group's requirements. The Pixel data stream is sent straight through without any alterations. The SCT data is packed to minimize the Event Fragment size using a zero suppression type of compression scheme. If a link did not record any data, and there are no errors, the link header and trailer are removed from the event. If a link produces data and there is no error in the trailer, then the trailer is removed from the event. While the unnecessary half words are removed from the data, the EFB packs the data to remove all of the gaps that have opened up due to headers and trailers that have been removed. The count of the number of 32-bit words that have been written to the Event FIFO and a count of all the data errors in the event are stored in internal EFB FIFOs and when both EFB engines make these counts available, an Event Fragment is transmitted to the Router FPGA. EFB data formats are shown in Section 2.2.

## 1.2.3.3 Data Flow Control

If backpressure is applied by the Router, either from the S-Link or from the SDSP during calibration (the SDSP can not assert back pressure when the ROD is configured for Physics data taking), data transmission to the Router will stop. The EFB will continue to process events and write to the Event FIFOs until either the FIFOs are almost full or the internal Event ID FIFOs are almost full. The EFB sends backpressure directly to the Formatters for flow control, and also to the RCF as status. If the Event ID FIFOs on the ROD are almost full, the RCF will assert ROD Busy to the TIM.

**CALIBRATION AND DIAGNOSTIC SPECIFIC FUNCTIONS**

**GROUP COUNTERS**: Group counters are used during calibration scans to monitor the event occupancy on the ROD. There is one 4-bit counter per group. When the event is transmitted from the Formatter, the group number is available with the TTC L1ID and BCID information that is used for the synchronization verification. When the Formatters finish sending all of the data from the group to the EFB, the group counter that matches the ROD Trigger Type value is incremented by one. The group counters only act on the ROD Trigger Type values 0 to 7 when they are enabled. The group counters are useful during calibrations by letting the MDSP know if there is room on the ROD for another.

**L1ID/BCID Trap:** The L1ID/BCID trap allows the user to take a snapshot of the synchronization status of the ROD at any time. The trap is enabled by setting bit 11 in EFB_CMND_0 to value 1 and is ready to readout when bit 11 of EFB_RUNTIME_STAT_REG set to value 1. The to trap more data, the user must clear the enable bit then set it to value 1. When the first word of a new event arrives on the EFB, the trap will capture data.
**Note:** The L1ID of both Pixel and SCT modules is reset to value 0x1 when the module receives an ECR. The TTC system expects the first L1ID to be ox0, and sends value 0x0 over the TTC bus for the first trigger. The Front End (FE) module L1ID data in the trap will always be 1 greater than the TTC L1ID value when the system is in synchronization.

**Write some text on re-sync**

**Diagnostic Access to the EVENT FIFO:** The Event FIFOs can be accessed by the MDSP over the ROD Bus for diagnostic purposes. The access id handled by a MDSP primitive. This access is not allowed during real time data taking modes.
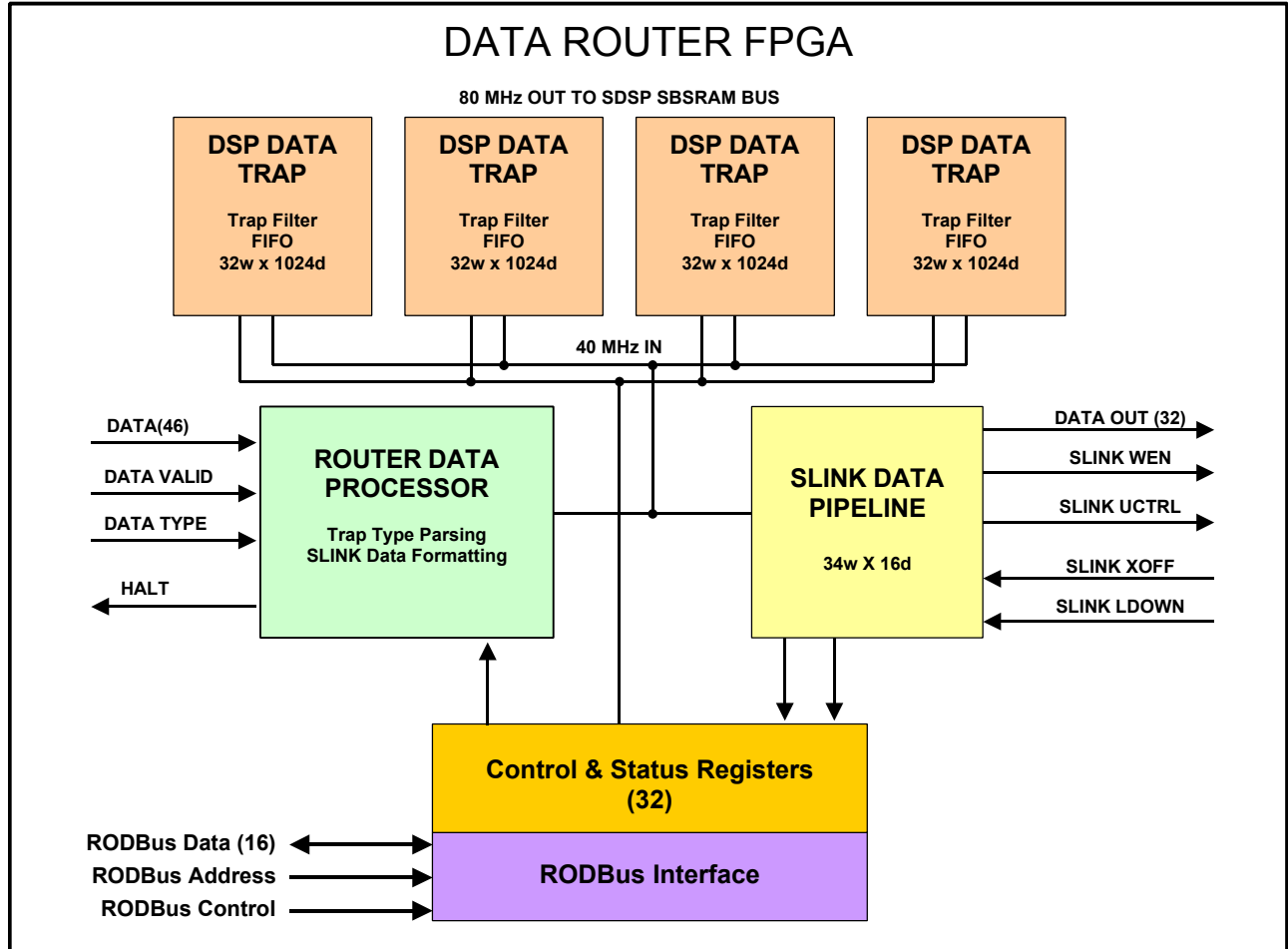
## 1.2.4 Data Router FPGA



**Figure 10: ROUTER FPGA TOP LEVEL BLOCK DIAGRAM**

## 1.2.4.1 Router FPGA Overview

The two main functions of the Data Router FPGA are to transmit Event Fragments to the S-Link, and to trap and send event data to the 4 slave DSPs on the SiROD. The last part of the data formatting is completed in the Router. As the event data passes through the ROD, link and error information is added on to the link header word, and increasing it to size that is larger than the width of the S-Link. When the data flows through the Router FPGA, the module ID, link ID and 4 common error flags (L1ID, BCID, timeout, and header error) are added into the link header. In the SCT byte stream, 16-bit half word format, the module L1ID and BCID are removed for the header and the link information is supplied instead. In the Pixel byte stream, 32-bit word format, the information is placed in a portion on the link header that was previously undefined. As with the EFB, the Router also monitors the static bit from the Formatter FPGA to identify the data format on the ROD. The interface to the S-Link is implemented per the Atlas specification, and when test data patterns are required to test the S-Link, the RoL test function can be used..

### 1.2.4.2 Data Trapping

The Router also has several data trapping algorithms implemented to allow flexible trapping of data to send to the Slave DSPs for module calibrations, error monitoring, and synchronization correction tasks.  For each SDSP on the ROD, there are 2 separate, configurable trap filters.  The filters can be set to trap everything, or to look for up to 2 specific ATLAS, TIM or ROD event types.  There are also configuration settings that allow the trap frequency to be divided to a slower rate and modulo to control the number of events that are sent to the DSPs.  Each trap filter is followed by a 1024x32 dual port RAM FIFO (internal to the FPGA) that is partitioned into 4 blocks that are 256 words deep and connected directly to the SBSRAM interface of one SDSP.  The trap criteria data is located in the first word of the Event Header when it arrives on the Router, and then removed to return the beginning of fragment marker to the proper format.

There are two selectable data formats used for the trapped data to be sent to the SDSPs.  The default format is the standard S-Link Data format that is used for data transmitted to the ROS and is used in normal running and calibration.  In the second, Error Data format, the Event Header and Trailers are modified to transmit specific link error information to the SDSP error task.  When the error data format is selected, the Router removes 3 words from the Event Header, adds 3 words and modifies 6 words in the Event Trailer.  The Error Data format details are shown in Section 2.3.4.  The modified error words in the Event Trailer show the status of the data from each FE Module using a 2-bit code per link.  The position of the bits in the words indicates which link and the value of the bits indicate if there is data and if there is an error from that link.  The Error Data format is intended for link resynchronization in a slave DSP and is useful for the SCT detector because they limit each data field to 16-bits and do not receive specific link L1ID or BCID information in each link.  The Pixel data format is 32-bits wide for all data words, and all L1ID and BCID information for each module can be included in the event data.

### 1.2.4.3 DMA Data Transfer from the Router to the SDSP

The data trapped in the Router must be transferred to the SDSP at a rate that will not reduce the processing bandwidth of the ROD data path during calibration and provide efficient data capture during Physics data taking runs.  The SDSP SBSRAM interface is used to allow the Router the burst the data into the internal SRAM of the DSP using direct memory access (DMA) at 80MHz, twice the speed that the data is written into the trap FIFO.  The trap FIFO will fill with 256 words in 6.4us and the transfer from Router to SDSP takes 3.2us.  This allows the DSP 3.2us between transfers to finish the interrupt service routine (ISR) and prepare for the next burst to internal memory.  The automatic transfer occurs when the Router has trapped 1 complete 256-word block of data, or has detected the end of an event in the one of the trap FIFOs.  When the data-trapping algorithm detects the end of a trapped event, an end of event word is written into the last location of the current 256-word block to give the DSP a quick status of the event that has just been transferred.  Because the SDSPs can not apply back pressure to the data path in normal Physics data taking mode, the end of event word will also indicate that there was data lost in the event if the Router to SDSP transfers are not able to keep up with the trigger rate of the system.

### 1.2.5 ROD Busy

#### 1.2.5.1 ROD Busy Overview
ROD Busy is used to notify the TTC system that a ROD can no process triggers in it's current state of operation.  The ROD Busy output of a ROD connects directly to the TIM module in each VME crate where all are combined and propagated to the TTC system.  The ROD also has an algorithm in place to provide a histogram of the status of ROD Busy.  The details for the ROD Busy Module Histogram can be found in section 1.1.1.4

#### 1.2.5.2 ROD Busy Sources
There are multiple sources of ROD Busy on the ROD that are defined within 2 separate groupings.  One grouping includes all Busy sources issued from a ROD that not configured for data taking.  For this case, ROD Busy is asserted on the ROD whenever the board is in the process of re-configuring any or all of the FPGA devices or if any device on the board is in reset.  The other grouping includes all Busy sources issued when a ROD is configured in data taking mode.  The 2 primary sources for ROD Busy during data taking are the EFB and Formatter FPGAs.

If a ROD is processing data, and one or more of the Formatter link FIFOs has an occupancy greater than or equal to the value in the ROD Busy Limit register, then the ROD will assert Busy to notify the TIM that there is no room to

store more event data.  If the Event ID Data FIFO in the EFB is almost full, the ROD will again assert ROD busy.  In both cases, if more triggers are sent to the ROD, events will be dropped from the byte stream.  In most cases, a Busy state on the ROD is preceded by prolonged back pressure (Xoff) from the S-Link or an extended period where the FE modules have transmitted large amounts of event data to the ROD.

## 1.3   Backend Digital Signal Processor Farm

The ROD has four "slave" DSPs attached to its data path (the router FPGA transfers events independently to the four DSPs). Each slave DSP is connected to a separate pipeline in the router FPGA, through one of its external memory interfaces. The slave DSPs are floating point devices that are configured to operate at 220MHz, with 264K Bytes of internal memory and 256M Bytes of external SDRAM memory that operates at 80MHz.

Slave DSPs have a program structure similar to the master DSP, with communication registers, primitive, reply, and text buffers. The main differences between the different DSP types is that since the slave DSPs process event data they have larger amounts of memory, and that the master DSP is continuously communicating with the slave DSPs during normal ROD operation.  Since multiple accesses to the slaves' memories would confuse the ROD and cause bus errors, the host does not directly access the slave DSPs, but rather uses the master DSP as a waypoint. The master DSP contains some extra program handshaking processes which control primitive list transfer from the host to the slave DSPs, the host portion of the handshaking process, and retrieval of any text or reply buffers.

The primary functions of the slave DSPs are event analysis during data taking, and detector calibration. The DSP retrieves data frames (256 words) from the Router FPGA using a DMA transfer with an ISR upon completion. The Router keeps track of event trailers and indicates the end of an event using the last data word of the frame; once an event is complete the DSP ISR places it on the event queue. Events are flagged for processing by the different tasks using the EVENT_TRAP_SETUP primitive. On each pass through the DSP's polling loop, one event is transferred for processing to all tasks for which it was slated; the event is then removed from the queue (and the memory it used is released).

## 1.4   Configurations and Functions

### 1.4.1   Basic ROD Configuration for Normal Data Taking

The ROD is designed to process triggered event data in real time with minimal use of the Master DSP and completely independent of the SDSPs.  The MDSP is used to distribute the setup data to the ROD FPGA registers and to transmit the configuration data to the FE modules prior to the start of triggers.  Prior to the start of a new run, all of the ROD data path FPGAs should be reconfigured to guarantee that they are in a known default startup state.  The MDSP and SDSPs do not need to be reset before the start of every run, but if they are, then the first step is to load and start (boot) the SDSPs.  Next, if there is event trapping planned in the SDSP, then the Router trap registers should be set up appropriately.  If there is no trapping planned, there is no need to configure the Router because in its startup default state, it is ready to process data.  The EFB is ready to process data in its startup default state, so again no configuration is required unless error masking is required or L1ID/BCID trapping is planned during the run.

**ADD DETECTOR SPECIFIC TEXT**

### 1.4.2   L1ID and BCID Value Trap in the EFB

L1ID and BCID data can be trapped on the EFB to allow the user to view diagnostic data of the current state of the event synchronization during data taking.  The trap does not interfere with normal data processing, and can be used by a MDSP task to recover from loss of event synchronization in a ROD.  The trap operates on single events only and the global enable signal must be cleared and set prior to filling the internal memory used to store the Event ID data.  The trap engine operates on each EFB data processing path independently, and each path can process separate events simultaneously, so in a single trap run there may be data from 2 different L1 triggers in the data record.  The trap implementation includes a Global Enable signal, a Trap Full signal, a single register location to latch the current TTC values for L1ID and BCID, and a block of memory to store the Module L1ID and BCID and the calculated L1ID values.  The structure is shown in the tables below:

| Description | REG ID | Address | Access | Width |
|---|---|---|---|---|
| EFB: EFB Command 0 Register, EFB_CMND_0 | 1A4 | 00402210 | RW | 32 |
| | | | Bit Value | |
| | | | 1 | 0 |
| **Bit 11** : L1ID Trap Enable<br>When this bit is set, the EFB will trap L1ID and BCID values in a block RAM. | | | Enable | Idle |

| Description | REG ID | Address | Access | Width |
|---|---|---|---|---|
| EFB: Run-Time Status Register, EFB RUNTIME STAT REG | 1A6 | 00402218 | R | 32 |
| | | | Bit Value | |
| | | | 1 | 0 |
| **Bit 11:** L1ID Trap data ready for read out | | | Data Ready | Idle |

| Description | REG ID | Address | Access | Width |
|---|---|---|---|---|
| EFB: L1ID/BCID TTC Trap Value | | 0040227C | RW | 1 |
| The TTC values for BCID and L1ID when trapping Event ID data. | | | Bit Value | |
| | | | 1 | 0 |
| **Bits[ 3: 0]:** EFB Engine 0 – TTC L1ID | | | Value | |
| **Bits[15: 8]:** EFB Engine 0 – TTC BCID | | | Value | |
| **Bits[19:16]:** EFB Engine 1 – TTC L1ID | | | Value | |
| **Bits[31:24]:** EFB Engine 1 – TTC BCID | | | Value | |

| Description | REG ID | Address | Access | Width |
|---|---|---|---|---|
| EFB: Link/Module L1ID/BCID Trap Value | | | R | 32 |
| Link0/Link48 Trap Values | | 00402280 | | |
| Link1/Link49 Trap Values | | 00402284 | | |
| … | | | | |
| Link46/Link94 Trap Values | | 00402338 | | |
| Link47/Link95 Trap Values | | 0040233C | | |
| Event ID data value trap memory. The data field includes the L1ID and BCID values received from the module or chip and the expected L1ID value used in the error checking process. [n=0 to 47] | | | Bit Value | |
| | | | 1 | 0 |
| **Bits[ 3: 0]** : Module/Chip L1ID – Link(n) | | | Value | |
| **Bits[ 7: 4]** : Expected Link L1ID – Link(n) | | | Value | |
| **Bits[15: 8]** : Module/Chip BCID – Link(n) | | | Value | |
| **Bits[19:16]** : Module/Chip L1ID - Link(n+48) | | | Value | |
| **Bits[23:20]** : Expected Link L1ID – Link(n+48) | | | Value | |
| **Bits[31:24]** : Module/Chip BCID – Link(n+48) | | | Value | |

To operate the L1ID/BCID trap, at any time before or during data-taking, set the L1ID trap enable signal, Bit[11], in the EFB_CMND_0 register to value '1', then monitor the L1ID trap data ready bit in the EFB_RUNTIME_STAT_REG register. When the value of Bit[11] is '1', read out the data from address locations 0x0040227C to 0x0040223C. To restart the trap, clear the L1ID trap enable signal and check to see that the L1ID trap data ready signal value returns to '0', then set the L1ID trap enable signal to value '1'.

### 1.4.3 RoL Test Block Trigger/Pattern Generator

The Readout Link test block is used to send data up the S-Link and to verify the readout bandwidth of the ROD. It can also used to verify that the interface from the EFB and Router through to the Slave DSPs is working properly. The setup and control of the RoL block in the SiROD follows per the ECR specification issued by ATLAS requirements document (ATC-TD-ES-0002), but it can also be operated in enhanced modes. The size of the data word portion of the RoL fragment can be set from 32 to 2048 words long, and the TX state machine can be configured to send a fixed number of fragments in the range from 1 to 16777216 or set to transmit data forever. See section 2.4 to view the output data format of the RoL test block

**RoL Function Modes:**
**Standard functions as defined in ATLAS requirements document ATC-TD-ES-0002**
1.  Transmit one 32 data word fragment for each start command issued by the host.
    a.  Reset the ROD
    b.  Set IDE_MEM_CTRL register (0x00404470), Bits[22:21] = 0x1
    c.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x1 (Start)
    d.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x0 (Reset)

2.  Transmit 32 data word fragments forever or until the host sends a stop command.
    a.  Reset the ROD
    b.  Set IDE_MEM_CTRL register (0x00404470), Bits[22:21] = 0x0
    c.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x1 (Start)
    d.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x0 (Stop)

3.  Transmit one 32 data word fragment for each TIM generated L1A trigger.
    a.  Reset the ROD
    b.  Set IDE_MEM_CTRL register (0x00404470), Bits[22:21] = 0x2
    c.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x1 (Start)
    d.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x0 (Stop)

**Enhanced RoL test block functions not defined in the requirements document**
4.  Transmit 1 [(n+1)*32] word fragment for each start command issued by the host.
    a.  Reset the ROD
    b.  Set IDE_MEM_CTRL register (0x00404470), Bit 21 = 0x1, Bits[29:24] = nData
    c.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x1 (Start)
    d.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x0 (Reset)

5.  Transmit [(n+1)*32] data word fragments forever or until the host sends a stop command.
    a.  Reset the ROD
    b.  Set IDE_MEM_CTRL register (0x00404470), Bit 21 = 0x0, Bits[29:24] = nData
    c.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x1 (Start)
    d.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x0 (Stop)

6.  Transmit a fixed number of [(n+1)*32] data word fragments for each start command issued by the host.
    a.  Reset the ROD
    b.  Set DBG_MEM_CTRL register (0x00404464), Bits[23: 0] = nEvents
    c.  Set IDE_MEM_CTRL register (0x00404470), Bits[23:21] = 0x4, Bits[29:24] = nData
    d.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x1 (Start)
    e.  Poll IDE_MEM_STAT register (0x00404474), Bit15 (1 = ROL Test Triggers Active)
    f.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x0 (Reset)

7.  Transmit 1 [(n+1)*32] data word fragment for each TIM generated L1A trigger.
    a.  Reset the ROD
    b.  Set IDE_MEM_CTRL register (0x00404470), Bits[22:21] = 0x3, Bits[29:24] = 0xn
    c.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x1 (Start)
    d.  Set IDE_MEM_CTRL register (0x00404470), Bit 20 = 0x0 (Stop)

# 2    Data Formats

Input and output data formats for the Data Path FPGAs as defined by the SCT and Pixel detector groups are included in this section.

## 2.1    Formatter Data Formats

### 2.1.1    SCT Serial Input Data

**The SCT module output data format is shown in the tables below.  A more detailed description of the data format transmitted by a SCT Module is available in the ABCD3T Chip Specification V1.2.  The general data format packet is shown in Table 12 below.  The size and location relative to the Header of the L1ID, BCID, Sync Bits (S) and the Trailer fields are constant from event to event, but the Data block field varies in size dependent on the amount of hit data read out of the module as shown in**

Table 13.

**Table 12: SCT Data Packet Format**

| Header | DT | L1ID | BC ID | S | | S | | Trailer |
|--------|-----|-------|-------------|-----|--------------|-----|----------------|------------------------------|
| <**11101**> | <0> | <tttt> | <bbbb bbbb> | <**1**> | <data block> | <**1**> | <data block n> | <**1000 0000 0000 0000**> |

**Legend:**   L1ID  >>   **t** = L1ID value from FE Module
BCID  >>  **b** = BCID value from FE Module

**Table 13: SCT Data Block (Hit in Data)**

| Leader | Chip Address | Hit Channel Address | | First Hit | | Next Chan Hit | | | Nth Chan Hit | | N+1th Chan Hit |
|--------|--------------|---------------------|-----|-----------|-----|---------------|-----|-----|--------------|-----|----------------|
| <**01**> | <aaaa> | <ccc cccc> | <**1**> | <ddd> | <**1**> | <ddd> | <**1**> | --- | <ddd> | <**1**> | <ddd> |

**Hit Pattern <ddd> :**
1. Detector Alignment = <1xx> or <x1x> or <xx1>
2. Level = <x1x>
3. Edge = <01x>
4. Test = <xxx>

**Empty Data block (no hit):** Leader **=** <**001**> is the only content in the data packet

**Table 14: SCT Error Data Block**

| Leader | Chip Address | Error Code | |
|--------|--------------|------------|-----|
| <**000**> | <aaaa> | <eee> | <**1**> |

### 2.1.2    SCT Formatter Output Data

All event data is in 16 bit packets packed in 32 bit words.  The Formatter adds information to the event data by expanding it from 32 to 38 bits wide.

**Table 15: SCT Formatter Output Field Definitions**

| Bits | Definition | Notes |
|------|-----------|-------|
| [31:0] | Event Data | |
| [35:32] | Link Number | (Present for all 32 bit words) |
| [36] | Time Out Error Bit | (Present for all 32 bit words) |
| [37] | Condensed Mode Bit | (Present for all 32 bit words) |

**Table 16: SCT Formatter Output, Bits [31:0]**

| Name | Bits [15:0] or [31:16] |
|------|------------------------|
| Header | 001pLLLLBBBBBBBB |
| Trailer | 010zhvxxxxxxxxxx |
| 1 hit condensed | 1FFFFCCCCCCCxfx0 |
| 2 hits condensed | 1FFFFCCCCCCCsfx1 |
| 1st hit cluster expanded | 1FFFFCCCCCCC0DDD |
| 1 hit cluster expanded | 1xxxxxxx0xxx1DDD |
| 2 hit cluster expanded | 1xxxxxxx1DDD1DDD |
| Flagged error | 000xxxxxxFFFFEEE |
| Raw data | 011nnnxxWWWWWWWW |

```
Key:

  B = BCID                                    n = count of raw data bits + 1
  C = cluster base address                    p = preamble error
  D = 3 bit hit data                          s = error in condensed mode data, 2nd hit
  E = ABC error code                          v = data overflow error
  F = FE number                               W = raw data
  f = error in condensed mode data, 1st hit   x = Don't care (ROD fills these with 0's)
  h = header trailer limit error              z = trailer bit error
  L = L1ID
```

## 2.1.3   Pixel Serial Input Data

The Pixel FE Module output data format is shown in the tables below.  A more detailed description of the data format transmitted by a Pixel FE Module is available in the MCC-DSM Chip Specification. The general data format packet is shown in Table 17 below.  The size and location relative to the Header of the L1ID, BCID, Sync Bits (S) and the Trailer fields are constant from event to event, but the Data block field varies in size dependent on the amount of hit data read out of the module as shown in Table 18.

**Table 17: Pixel Data Packet Format**

| Header | L1ID | BC ID | S | | S | | Trailer |
|--------|------|-------|---|---|---|---|---------|
| <11101> | <ssss,tttt> | <bbbb bbbb> | <1> | <data block 0> | <1> | <data block n> | <1 0000 0000 0000 0000 0000 00> |

**Legend:**  L1ID  >>  **s** = Number of skipped event detected by the FE Module
L1ID  >>  **t** = L1ID value from FE Module
BCID  >>  **b** = BCID value from FE Module

**Table 18: Pixel Data Block 0 Combinations**

| TYPE | 1st Field | S | 2nd Field | 3rd Field | 4th Field |
|------|-----------|---|-----------|-----------|-----------|
| **HIT** | <1110 FFFF> | <1> | <RRRR RRRR> | <C CCCC> | <TTTT TTTT> |
| **FE-ERROR** | <1110 FFFF> | <1> | <1 1110 EEEE MMMM> | | |
| **MCC-ERROR** | <MMMM MMMM> | | | | |
| **TRAILER** | <1 0000 0000 0000 0000 0000 00> | | | | |

**Legend:**    **C** = Pixel Column      **E** = FE Chip Error Code    **F** = FE Chip Number
**M** = MCC Error Code    **R** = Pixel Row          **T** = Time Over Threshold

**Table 19:Pixel Data Block n Combinations**

| TYPE | 1st Field | S | 2nd Field | 3rd Field | 4th Field |
|---|---|---|---|---|---|
| **HIT SAME FE** | <RRRR RRRR> | | <C CCCC> | <TTTT TTTT> | |
| **HIT NEW FE** | <1110 FFFF> | <1> | <RRRR RRRR> | <C CCCC> | <TTTT TTTT> |
| **FE-ERROR SAME FE** | <1 1110 EEEE MMMM> | | | | |
| **FE-ERROR NEW FE** | <1110 FFFF> | <1> | <1 1110 EEEE MMMM> | | |
| **TRAILER** | <1 0000 0000 0000 0000 0000 00> | | | | |

Legend:  **C** = Pixel Column    **E** = FE Chip Error Code   **F** = FE Chip Number
**M** = MCC Error Code   **R** = Pixel Row          **T** = Time Over Threshold

## 2.1.4   Pixel Formatter Output Data
All Pixel data words are 32 bits wide, including the header and the Trailer

**Table 20: Pixel Formatter Output Field Definitions**

| Bits | Definition | Notes |
|---|---|---|
| [31:0] | Event Data | |
| [35:32] | Link Number | (Present for all 32 bit words) |
| [36] | Time Out Error Bit | (Present for all 32 bit words) |
| [37] | Not Used | |

**Table 21: Pixel Formatter Output Bits [31:0]**

| Name | Bits [31:0] |
|---|---|
| Header | 001PxxxxxxxxAAAAMMMMLLLLBBBBBBBB |
| Trailer | 010ZHVxxxxxxxxxxxxxxxxxxxxxxxxxx |
| Hit | 100xFFFFTTTTTTTTTxxxCCCCCRRRRRRRR |
| FE Flag Error (Old) | 0000FFFFxxxxxxxxxxx11110FFFFEEEE |
| FE Flag Error (New) | 0001FFFFxxx11111eeeeeeeeeEEEEEEEE |
| Raw Data | 011DDDDDDDDDDDDDDDDDDDDDDDDDDDDD |
| Time Out Data | 00100000000000000000000000000000 |

```
Key:

A = BCID Offset used in EFB error check   M = Number of MCC Skipped Events
B = BCID                                  N = Count of raw data bits + 1
C = Pixel Column                          P = Preamble Error
D = Raw Data                              R = Pixel Row
E = FE Error Code                         T = Time over Threshold value
e = MCC Error Code                        V = Data Overflow Error
F = FE Number                             x = don't care  (ROD fills these with 0's)
H = Header Trailer Limit Error            Z = trailer bit error
L = L1ID
```

## 2.2 EFB Data Formats

### 2.2.1 Event Data /Trigger Type from ROD Controller FPGA

The data in the Event ID packet is used to form the event header and process events correctly in the data flow path and is composed of 16 16-bit words that are transferred from the RCF to the EFB with each L1 Trigger.

**Table 22: EFB Event ID and Dynamic Mask Data**

| Word | Description Bits[15:0] |
|------|------------------------|
| 0 | L1ID [15:0] |
| 1 | ECRID [7:0] **&** L1ID [23:16] |
| 2 | BCID [11:0] **&** 0 **&** ROL Test Block Enable **&** BOC Clock OK **&** TIM Clock OK |
| 3 | ROD Event Type [5:0] **&** TIM Event Type [1:0] **&** Atlas Trigger Type [7:0] |
| 4 | Dynamic Mask [ 15:  0]   \|   Links [ 7, 6, 5, 4, 3, 2, 1, 0] |
| 5 | Dynamic Mask [ 31: 16]   \|   Links [15,14,13,12,11,10, 9, 8] |
| 6 | Dynamic Mask [ 47: 32]   \|   Links [23,22,21,20,19,18,17,16] |
| 7 | Dynamic Mask [ 63: 48]   \|   Links [31,30,29,28,27,26,25,24] |
| 8 | Dynamic Mask [ 79: 64]   \|   Links [39,38,37,36,35,34,33,32] |
| 9 | Dynamic Mask [ 95: 80]   \|   Links [47,46,45,44,43,42,41,40] |
| 10 | Dynamic Mask [111: 96]   \|   Links [55,54,53,52,51,50,49,48] |
| 11 | Dynamic Mask [127:112]   \|   Links [63,62,61,60,59,58,57,56] |
| 12 | Dynamic Mask [143:128]   \|   Links [71,70,69,68,67,66,65,64] |
| 13 | Dynamic Mask [159:144]   \|   Links [79,78,77,76,75,74,73,72] |
| 14 | Dynamic Mask [175:160]   \|   Links [87,86,85,84,83,82,81,80] |
| 15 | Dynamic Mask [191:176]   \|   Links [95,94,93,92,91,90,89,88] |

Dynamic Mask [95:0] ➔ 2 bits per link and is used only correct module L1ID offset errors.
   Field value >> "00" = No change to L1 ID for this event
   Field value >> "01" = Increment L1 ID for this event
   Field value >> "10" = Decrement L1 ID for this event

### 2.2.2 Event Header and Trailer Data

The Event Header information is defined by the ATLAS collaboration, and a detailed description of the data format specified for the Event Header and Trailer is available in the Event Format Specification document (Ref). At the output of the EFB, the data in the header is still internal to the ROD and extra fields used to improve Router functions can be removed before the data is transmitted to the S-Link.

**Table 23: Event Header Format at the Output of the EFB**

| Word | Contents | Comment |
|------|----------|---------|
| 0 | 0xB0FTTTTT | Beginning of fragment marker/ <br> T = Router Trap Type Data (Removed by Router) |
| 1 | 0xEE1234EE | Start of header |
| 2 | 0x9 | Header size |
| 3 | 0x30100000 | Format Version Number (Ver 3.1) |
| 4 | 0x001XNNNN Pixel <br> 0x002XNNNN SCT | Source Identifier <br> N = Module ID, X = LS Nibble of Sub-detector ID |
| 5 | 0xTTSSSSSS | Run Number: T = Run Type ➔ 0x00 > Physics <br>                                     0x01 > Calibration <br>                                     0x02 > Cosmics <br>                                     0x0F > Test <br>                     S = Sequence within Run Type |
| 6 | 0xEELLLLLL | Extended Level 1 ID: E = ECR ID, L = L1ID |
| 7 | 0x00000BBB | Bunch Counter ID |
| 8 | 0x000000AA | ATLAS Level 1 Trigger Type |
| 9 | 0x00RR000T | Detector Event Type R = ROD or T = TIM |

**Table 24: Event Trailer Format at the Output of the EFB**

| Word | Contents | Comment |
|---|---|---|
| 0 | Event Fragment Error Flags | Status 1: Bit error see EFB errors [31:0] (Table 31) |
| 1 | Error Count and<br>Static Error Flags | Status 2: Count of words with error [15:0]<br>       TIM OK, BOC OK and ROL Status |
| 2 | 0x2 | Number of status words |
| 3 | Ndata | Count of data words |
| 4 | 0x1 | Status block position: 0 = before, 1 = after data |
| 5 | 0xE0F00000 | End of fragment marker |

## 2.2.3   EFB Output Data

### 2.2.3.1   SCT Format

SCT data is formatted in 16 bit packets packed into 32 bit words.  The data word format for bits [31:0] is the same as shown in Table 16.

**Table 25: EFB Output Link Data Format - SCT**

| Bits | Definition | Notes |
|---|---|---|
| [31:0] | Event Data | |
| [38:32] | Link Number | Present for in header word only |
| [39] | Time Out Error Bit | Present for in header word only |
| [40] | Condensed Mode Bit | Present for in header word only |
| [41] | L1 ID Error Bit | Present for in header word only |
| [42] | BC ID Error Bit | Present for in header word only |

### 2.2.3.2   Pixel Format

All Pixel data is in 32 bit words.  Data bits 42-32 are valid for header word only.  The data word format for bits [31:0] is the same as shown in Table 21.

**Table 26: EFB Output Link Data Format - Pixel**

| Bits | Definition | Notes |
|---|---|---|
| [31:0] | Event Data | |
| [38:32] | Link Number | Present for in header word only |
| [39] | Time Out Error Bit | Present for in header word only |
| [40] | Not Used | Present for in header word only |
| [41] | L1 ID Error Bit | Present for in header word only |
| [42] | BC ID Error Bit | Present for in header word only |

## 2.3   Router Formats

The Router modifies the output data format of the module data before it is sent out on the S-Link Bus.  The only change in format is performed in all of the link headers where the L1ID and BCID are replaced with error codes and the link number as shown in Table 27 for SCT and Table 28 for Pixel.  All of the other data fields are the same as shown in Table 16.  The data fields used to over write values in the link headers are located in Bits [42:32] of the data word that output from the EFB.

### 2.3.1 SCT Format

**Table 27: Router Link Header Output – SCT Format Bits[31:0]**

| Name | Bits [15:0] or [31:16] – Output to the S-Link | EFB Output |
|------|----------------------------------------------|------------|
| Header | 001ptlbKdMMMMMMM | 001pLLLLBBBBBBBB |

```
Key:
b = BCID error             B = BCID
l = L1 error               L = L1ID
M = link number            p = Preamble Error
t = time out error         x = don't care
d = link masked by DSP
```

### 2.3.2 Pixel Format

**Table 28: Router Link Header Output – Pixel Format Bits[31:0]**

| Name | Bits [31:0] – Output to the S-Link | EFB Output |
|------|-----------------------------------|------------|
| Header | 001ptlbxdNNNNNNNMMMMLLLLBBBBBBBB | 001pxxxxxxxxAAAALLLLLLLLBBBBBBBB |

```
Key:
A = Number of Accepts per L1 Trigger
b = BCID error                        B = BCID
l = L1 error                          L = L1ID
N = Link number                       p = Preamble Error
t = time out error                    M = Number of MCC Skipped Events
d = link masked by DSP

x = don't care  (ROD fills these with 0's)
```

### 2.3.3 S-Link Event header Output Format (sent to S-link and DSP)

The Router adds the UCTRL bit in the Router output format to indicate to the ROS the location of the beginning and end of fragment markers.   The format is shown below:

### 2.3.3.1 S-Link Event Header and Trailer Output Format

**Table 29: Event Header Format at the Output of the Router**

| Word | Contents | Comment |
|---|---|---|
| 0 | 0xB0F00000 + UCTRL | Beginning of fragment marker |
| 1 | 0xEE1234EE | Start of header |
| 2 | 0x9 | Header size |
| 3 | 0x30100000 | Format Version Number (Ver 3.1) |
| 4 | 0x001XMMMM Pixel<br>0x002XMMMM SCT | Source Identifier<br>M = Module ID, X = LS Nibble of Sub-detector ID |
| 5 | 0xTTSSSSSS | Run Number: T = Run Type ➜ 0x00 > Physics<br>0x01 > Calibration<br>0x02 > Cosmics<br>0x0F > Test<br>S = Sequence within Run Type |
| 6 | 0xEELLLLLL | Extended Level 1 ID: E = ECR ID, L = L1ID |
| 7 | 0x00000BBB | Bunch Counter ID |
| 8 | 0x000000AA | ATLAS Level 1 Trigger Type |
| 9 | 0x00RR000T | Detector Event Type R = ROD or T = TIM |

**Table 30: Event Trailer Format at the Output of the Router**

| Word | Contents | Comment |
|---|---|---|
| 0 | See Table 31 | Status 1: Flagged Data Errors |
| 1 | See Table 32 | Status 2: Count of errors in Event Fragment (C)<br>Static Error Flags and ROL Status (S) |
| 2 | 0x2 | Number of status words |
| 3 | 0x0000Ndata | Count of data words |
| 4 | 0x1 | Status block position: 0/1 = before/after data |
| 5 | 0xE0F00000 | End of fragment marker |

**Table 31: ERROR FLAGS: STATUS WORD 0**

| Bit Location | ERROR FLAGS: STATUS WORD 0 |
|---|---|
| 0 | BC ID Error Count |
| 1 | L1 ID Error Count |
| 2 | FE Module Timeout Error Count |
| 3 | Data may be incorrect (see Bits 31:16) |
| 4 | Internal Buffer Overflow (see Bits 17:16) |
| 5 – 15 | Reserved for Atlas |
| 16 | Almost Full Error, Data Truncated Count |
| 17 | Data Overflow Count |
| 18 | Header Bit Error Count |
| 19 | Sync Error Count |
| 20 | (SCT)Flagged Error Count / (Pixel)Invalid Row(>159) OR Column(>17) |
| 21 | (SCT)Condensed Mode Hit Pattern Error / (Pixel)MCC Sent Empty Event |
| 22 | (SCT)Non-Sequential Chip Num / (Pixel)MCC Flagged Error - EOCOVERFLOW |
| 23 | (SCT)Invalid FE Chip / (Pixel)MCC Flagged Error - HAMMINGCODE |
| 24 | (SCT)Trailer Bit Error Count / (Pixel)MCC Flagged Error - REGPARITY |
| 25 | (SCT)Active Link Masked / (Pixel)MCC Flagged Error – HITPARITY |
| 26 | (Pixel)MCC Flagged Error – BITFLIP |
| 27 | (Pixel)MCC Flagged Error - HITOVERFLOW |
| 28 | (Pixel)MCC Flagged Error - EOEOVERFLOW |
| 29 | (Pixel)MCC Flagged Error - L1CHKFAILFE |
| 30 | (Pixel)MCC Flagged Error - BCIDCHKFAIL |
| 31 | (Pixel)MCC Flagged Error - L1CHKFAILGLOBAL |

**Table 32: ERROR STATUS: STATUS WORD 1**

| Bit Location | ERROR STATUS: STATUS WORD 1 |
|---|---|
| 0 - 15 | Count of Errors in Event Fragment |
| 16 | TIM Clock Error |
| 17 | BOC Clock Error |
| 18 | ROL Test Block Indicator |
| 19 - 31 | Reserved for future expansion |

## 2.3.4 Error Event Output Format (sent to SDSP only)

The Router can output a format used by the DSPs for error diagnostics. The format requires that the link headers remain unchanged as they are processed in the Router. All error flags, module numbers and L1ID and BCID fields are required by the DSP in this mode. The Pixel ROD link headers are the same with out regard to which format is selected. The event header format is changed as shown in table .

### 2.3.4.1 DSP Error Event Header and Trailer Output Format

**Table 33: Error Event Header Format at the Output of the Router**

| Word | Contents | Comment |
|---|---|---|
| 0 | 0xB0F00000 | Beginning of fragment marker |
| 1 | 0xEE1234EE | Start of header |
| 2 | 0xTTSSSSSS | Run Number: T = Run Type ➜ 0x00 > Physics<br>0x01 > Calibration<br>0x02 > Cosmics<br>0x0F > Test<br>S = Sequence within Run Type |
| 3 | 0xEELLLLLL | Extended Level 1 ID: E = ECR ID, L = L1ID |
| 4 | 0x00000BBB | Bunch Counter ID |
| 5 | 0x000000AA | ATLAS Level 1 Trigger Type |
| 6 | 0x00RR000T | Detector Event Type R = ROD or T = TIM |

**Table 34: Error Event Trailer Format at the Output of the Router**

| Word | Contents | Comment |
|---|---|---|
| 0 | See Table 31 | Status 1: Bit error see EFB errors [31:0] |
| 1 | See Table 32 | Status 2: Count of words with error [15:0]<br>Static Error Flags and ROL Status |
| 2 | Bit Value (Table 35) | Error status for links 15 - 0 |
| 3 | Value | Error status for links 31 - 16 |
| 4 | Value | Error status for links 47 - 32 |
| 5 | Value | Error status for links 63 - 48 |
| 6 | Value | Error status for links 79 - 64 |
| 7 | Value | Error status for links 95 - 80 |
| 8 | 0xE0F00000 | End of fragment marker |

As the data is processed in the Router, six 32-bit words are built allowing 2-bits of status per link. The two bits encode the following cases:

**Table 35: Link Error Status Codes**

| Bit Value | Status |
|---|---|
| 00 | No Data |
| 01 | Good Data |
| 10 | BCID or L1ID error |
| 11 | Time Out or Chip error |

The position of each pair of bits in the word indicates which link corresponds to the status shown. In the first word that shows the error status for links 0 to 15, bits 0 and 1 indicate status for link 0 and bits 30 and 31 indicate the status for link 15. This structure is consistent for all error words.

## 2.3.5    Router to SDSP Trap Format

The Router fills each trap FIFO with data in the selected trap format, and the way that the FIFO is filled and data transferred to the SDSPs can be selected using a register setting.  The default mode is to fill the FIFO and transfer data in full event mode.  In this mode, the Router generates a status word at the end of an event and writes it to the last location of the last frame of a trapped event.  The word contains trap information that is required by the DSP for signal processing.  The format is as follows:

**Table 36:  End of Frame Word Format used in SDSP Trapping**

| Contents (Binary format) | Comment |
|---|---|
| 01000000000t11dewwwwwwwwwwwwwwwwww | ```
T = trap number
d = NOT data loss (FIFO full)
e = NOT error (one or more links has  an error)
w = event word count
``` |

## 2.4 ROL Test Output Format for the Slink

The SiROD supports a test mode that transmits a test sequence to the Readout Link and SDSPs for diagnostics.  The data format is as follows:

**Table 37: ROL Test Readout Data Format**

| The ROL Test Block Format in the Silicon ROD is different from the block specified in Atlas Doc No. ATC-TD-ES-0002. | Data | Offset |
|---|---|---|
| Beginning of Fragment Control Word | 0xB0F00000 | |
| ROD Header Marker | 0xEE1234EE | 0 |
| Header Length | 0x00000009 | 1 |
| Format Version | 0x03000000 | 2 |
| Source ID (SS = 10 Pixel, 20 SCT) | 0x00SS0000 | 3 |
| Run Number | 0x00000000 | 4 |
| L1 ID | 0x00nnnnnn | 5 |
| Fixed BCID | 0x0000BCID | 6 |
| ATLAS Trigger Type (May vary for TIM Triggers) | 0x00000000 | 7 |
| TIM Trigger Type (May vary for TIM Triggers) / Repeat Count | 0x00rr0000 | 8 |
| First Data Word (Walking '1') | 0x00000001 | 9 |
| Walking word data pattern will repeat per the repeat count | 0x00000002 | 10 |
| | 0x00000004 | 11 |
| | 0x00000008 | 12 |
| | 0x00000010 | 13 |
| | 0x00000020 | 14 |
| | 0x00000040 | 15 |
| | 0x00000080 | 16 |
| | 0x00000100 | 17 |
| | 0x00000200 | 18 |
| | 0x00000400 | 19 |
| | 0x00000800 | 20 |
| | 0x00001000 | 21 |
| | 0x00002000 | 22 |
| | 0x00004000 | 23 |
| | 0x00008000 | 24 |
| | 0x00010000 | 25 |
| | 0x00020000 | 26 |
| | 0x00040000 | 27 |
| | 0x00080000 | 28 |
| | 0x00100000 | 29 |
| | 0x00200000 | 30 |
| | 0x00400000 | 31 |
| | 0x00800000 | 32 |
| | 0x01000000 | 33 |
| | 0x02000000 | 34 |
| | 0x04000000 | 35 |
| | 0x08000000 | 36 |
| | 0x10000000 | 37 |
| | 0x20000000 | 38 |
| | 0x40000000 | 39 |
| Last Data Word | 0x80000000 | 40 |
| Status Element 0 | 0x00000000 | 41 |
| Status Element 1 (Set Bit indicates ROL Test Block) | 0x00040000 | 42 |
| Number of Status Elements | 0x00000002 | 43 |
| Number of Data Words (== Repeat Count*32) | 0xnnnnnnnn | 44 |
| Status Block Position | 0x00000001 | 45 |
| End of Fragment Control Word | 0xE0F00000 | |

# 3 ROD Front Panel Indicators

**MDSP** : Master DSP Heartbeat. Indicates that the MDSP has booted and is running. The rate is controlled by Timer 0.

**RBB** : (ROD Bus Busy) Indicates communication on the ROD BUS

**TRP** : (Triggers Pending) Indicates that the ROD is processing L1 triggers. This signal is derived from the Front End Occupancy counters in the ROD Controller FPGA. The FE Occupancy counter is incremented by 1 for each L1 sent to the FE Modules and decremented by 1 each time a trailer is written into the link FIFO in the Formatter.

**ST0/ST1/ST2** : These lights are controlled using the Master DSP Serial Receiver ports as GPIO. They are currently defined in the Master DSP source code.

**XOFF** : The state of the Xoff signal from the S-Link. On if Xoff is active.

**BOC** : (BOC Busy) On if the BOC is installed and if the signal BOC_BUSY is inactive.

**SDSPn/GRN**: Slave DSP Heartbeat. Indicates that the SDSP has booted and is running. The rate is controlled by Timer 0.

**SDSPn/YEL** : This lights is controlled using one of the SDSP serial ports as GPIO. This indicates that a DMA block has been transferred from the Router to the DSP by toggling for each completed transfer. The definition for this indicator may change in the future.

**THERM** : (Thermal Overheat) On if the ROD temperature is greater than 70C

**START** : On when the Program Reset Manager FPGA is in configuration mode following a hardware reset

**BUSY** : (ROD Busy) On when the ROD is in any of the ROD Busy states defined in the specification.

**5.0V** : 5V power supply is operational

**3.3V** : 3.3V power supply is operational

**LINK OUT** : This front panel LEMO connector is connected to a signal mux that allows one link input at a time to be routed to the front of the ROD for examination. It is a LVTTL signal that is bandwidth and current limited by a series 510 ohm resistor

**RESET** : This front panel push switch allows a hardware reset of the ROD from the front of the crate. A hardware reset of the ROD will reset all of the DSPs and will force all of the FPGAs to be re-configured.

# 4 References

[1] MCC Design Group, ATLAS/Pixel Collaboration, edited by R. Beccherle and G. Darbo, "MCC-DSM Specifications - Output Data Format", September 2002

[2] ABCD3T Specifcation V1.2

[3] ATL-D-ES-0019v3.1, ATLAS Event Format

[4] ATC-TD-ES-0002 ROL Readout

[5] ROD Users Manual Appendix A

[6] ROD Users Manual Appendix B

[7] Spansion, AMD, Fujitsu, Am29LV017D Data Sheet, Doc # 21415, Rev E, Amendment +3, June 13, 2005

# 5   Index

# 6 List of Tables