ROD Operations Manual

Revision History	
July 10, 2002	First Draft
July 19, 2002	Added Test Calibration Procedure
April 30, 2003	Added ROD Description and Updated all section
October 8, 2003	Added ROD Description and Updated all section
October 22, 2003	Added new Formatter Sections
November 1, 2005	First update in 2 years
November 2, 2005	Inserted Error Flag and Status Definitions on pg 31
May 24, 2006	Added ROD Functions: ROL Test Block, ROD Busy Module
June 5, 2006	Added ROL Test Block Data Format Table

DRAFT COPY

John Joseph
Richard Jared
Douglas P. Ferguson

Table of Contents

1.0 ROD Description

The Atlas Silicon Read Out Driver (SiROD) is designed to control and receive data from Silicon Tracker (SCT) or Pixel modules. Control commands are sent to the modules as serial data streams and can be Level 1 triggers, Event Counter resets, Bunch Crossing resets, calibration commands, or module register data. The serial return data can be event data or register data. The SiROD is designed to detect and mark format errors, build events, and transmit the event fragments to an S-Link card (interface to the Read Out System) and to any combination of 4 Slave Digital Signal Processors (DSP) set to trap event specific data on a non-interfering basis.

The SiROD is the interface used for configuration and read out of the front-end modules and the Back Of Crate (BOC) optical interface. Module calibration and event data monitoring are some of the function that are performed in the Slave DSP farm.

An Overview of the ROD is shown in the figure below:

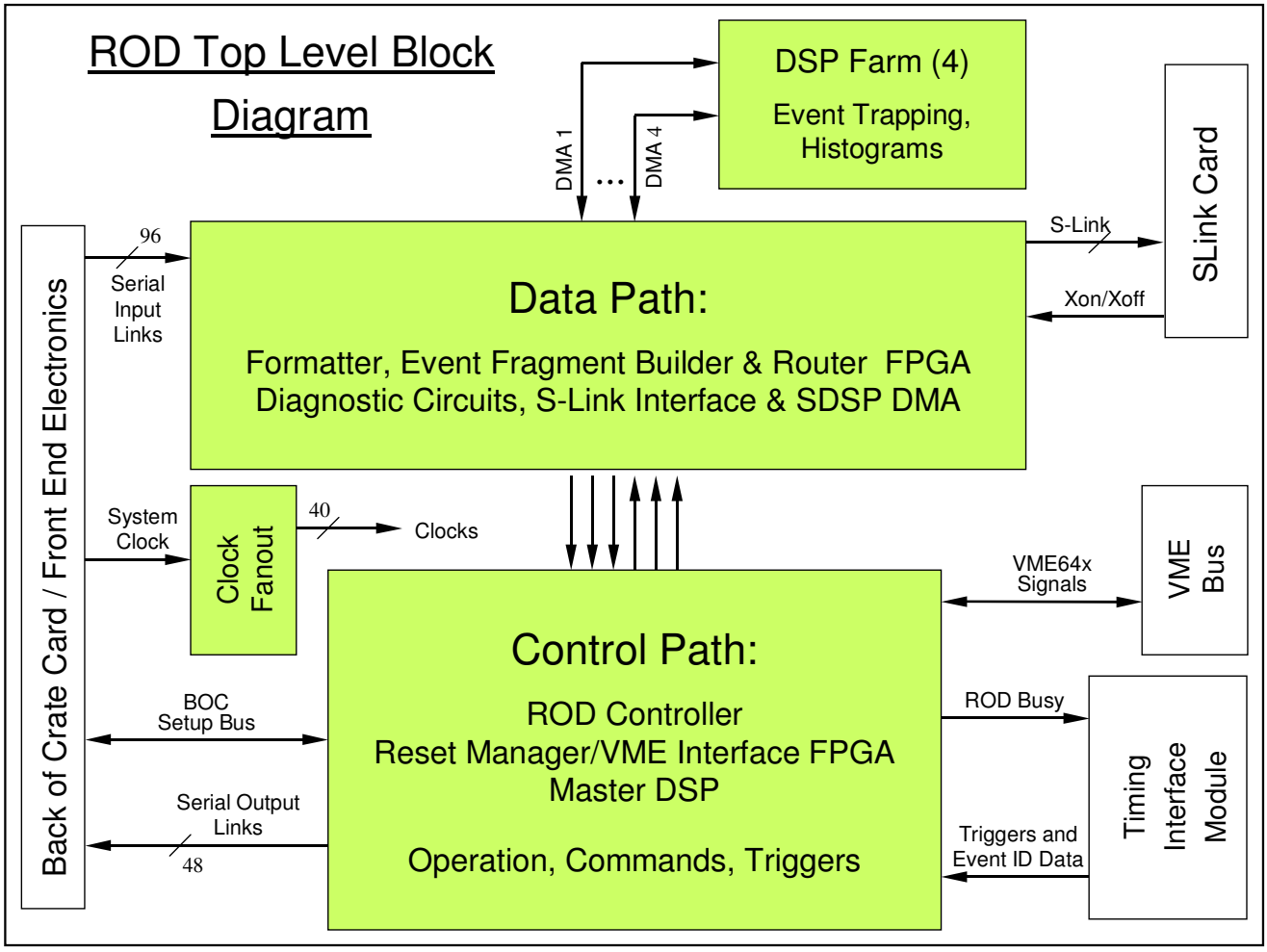


Figure 1: ROD Top Level Block Diagram

After power-on or hardware reset, the Program Reset Manager (PRM) fetches its configuration code from a ROM, and allows the host to ROD VME interface (A32/D32) to become active. Next, the PRM releases the Master DSP (MDSP) reset control, allowing the DSP to fetch boot code, and transmits configuration data from a bank of flash memory chips to initialize the FPGAs. The flash memories can be read or written by the VME host if changes to the ROD FPGA or MDSP boot code are required. The slave DSP boot code is loaded from the host via the Master DSP at a later time.

The PRM also operates as a bridge between the custom slave VME chipset and the Master DSP Host Port Interface (HPI) address space. Normally the VME interface is used to set the ROD mode and initialize the ROD and BOC registers.

Primitives are used to configure and initialize the ROD and BOC. Primitives are software entities that perform functions such as read/write a register. The software is in the ROD DSPs and the data in a defined format is sent from the ROD Crate Controller (RCC) over the VME bus.

Tasks are entities that perform operations such as calibration control in the Master DSP and event data histograms in slave DSPs. Tasks are initialized by primitives and are activated by the main control loop as required. They return to the main control loop when the operation is performed such as a histogram of an event or a trigger issued from the master DSP.

Triggering of the FE modules can be performed in three different ways. The first and primary mode is that a trigger can be issued from ATLAS via the Timing Interface Module (TIM) electronics board. This is the normal trigger during data taking. The RCC can also instruct the TIM electronics board to trigger the ROD for calibration or testing. The second way to trigger is to use an internal FIFO in the ROD that simulates the TIM interface. This mode is used in testing of the ROD and can issue triggers at a rate slightly greater than 100kHz. The third way is to use the histogram control task in the master DSP that uses the serial port to send triggers to the FE modules. This is the trigger mode used in calibration. In all modes, the ROD will issue a busy signal to the TIM, as required, to throttle triggers (e.g. if all ROD buffers are full).

Figure 1 shows the normal event data flows through the ROD. Serial input data always passes through the data receivers unchanged. The data receivers can route the serial data stream to diagnostic FIFOs located at the data inputs of the ROD (can be used in module configuration or to trap an event for diagnostics). As a diagnostic tool, the FIFOs can be loaded with events to play through the ROD.

The data receivers route the Event Data to 8 identical Formatter FPGAs. The Formatters convert the serial data streams to parallel format, and detect and mark data errors. De-randomizing buffers are located at the output of the formatters and used to queue events for transmission to the EFB. An event is transmitted from the Formatters to the EFB when the controller FPGA sends the dynamic Mode bits used to control the readout of each link and a readout token. These signals are generated when a trigger is detected.

The EFB receives ATLAS event ID data from the controller FPGA. In normal data taking, the source of the data is the TIM with additional information provided by the ROD (dynamic mode bits and event counter resets). After the header and mode information is sent to the EFB the ROD Controller FPGA (RCF) issues one token to the formatters. The formatters push data to the EFB. The EFB checks L1ID and BCID values and records errors. It also records any errors that were flagged by the formatters. The event data is then stored in 16K de-randomizing FIFOs (two each). There are two identical engines in the EFB transferring data at up to 40MHz (total bandwidth 80 MHz). When and event is ready (header, data body and trailer) it is transmitted to the Router.

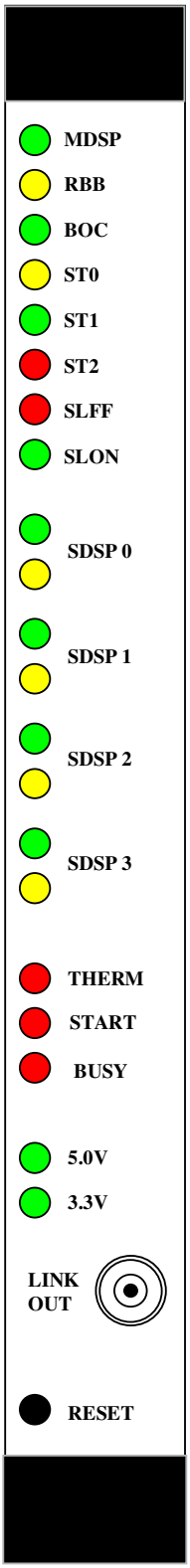
The Router has two main functions. The first, which is the main physics data path, is to transmit 32 bit data words to the S-Link at 40 MHz. If the S-Link is receiving data faster than it can transfer to the ROS, the S-Link can assert Xoff to apply backpressure to the ROD data path. When back pressure is applied, read out of data from the EFB FIFO is stopped. When the EFB memories are almost full backpressure is applied to the Formatters. This will stop event data transmission from the formatter link FIFOs. The Formatters have a programmable threshold that sends a flag (Header Trailer Limit) to the serial decoder block that allows only link headers and trailers to be written to the link FIFOs, and a programmable threshold that raises a busy signal that will be sent to the TIM via the Controller and PRM FPGAs. The probability of having the ROD go into ROD Busy as a result of the full input FIFOs is very low. (Simulation with 10,000 events has shown that there are no header trailer limits with a 95KHz trigger rate at 2 times occupancy. At 2.8 times occupancy the fraction of header trailer events was 2x10-4). A ROD Busy signal to the TIM should stop triggers from ATLAS until there is more space in the formatter FIFOs. The busy signal will preserve that there is one event for each ATLAS trigger.

The second function of the router is to trap data for the DSPs. This is performed with no effect on the S-Link data during normal running. When the ROD is in calibration mode the DSPs can assert backpressure to pause the ROD data flow. The Router traps event data by looking at the header event types and buffering the data in a FIFO (1024 words per DSP). A normal event at 1 times maximum occupancy is about 150 words. The data is transferred by DMA (256 word blocks) to the DSPs at 80MHz so that events larger than 1024 words can be handled with no backpressure (Up to the DSP fast memory limit of 5632 words).

Slave DSPs perform functions such as histograms of event data, monitoring (not coded), and error processing (not coded). Fitting of data (not coded) is also performed. The master DSP usually controls the operations in the slave DSPs.

The controller FPGA controls the actions that configuration ROD, BOC and Front End (FE) modules and the processing of triggers. Using primitives and tasks in the Master DSP performs configuration on the ROD and all of its peripherals. The Master DSP has read/write access to the ROD bus (general read write bus) to perform configuration of the ROD and BOC. Serial streams used to configure modules are connected to the controller FPGA for transmission to the modules. Primitives in the master DSP are used by the RCC to monitor the operation of the ROD in normal data acquisition. The controller FPGA (in the real time path) receives triggers and ATLAS trigger information from the TIM. An L1 Trigger from the TIM results in the ROD Controller issuing a trigger to the FE modules (selected by a mask in the controller FPGA) and transmission of the dynamic read out mode bits to the Formatters. When the event description is received from ATLAS, the Event ID information, mode bits that control incrementing of or decrementing of L1 number and event counter reset numbers are sent to the EFB for processing of the event. After a short delay (time of the EFB to make ready the data), a readout token is sent to the formatters.

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

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

ST0: MDSP control using CLKR0
ST1: MDSP control using CLKR1, RCF function to indicate ROL Test Block Mode
ST2: MDSP control using FSR0, RCF function to indicate ROD standalone clock enabled

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

SLON : Status of the LDOWN# signal from the S-Link Card. This indicator will flash or dim if there is activity on the S-Link.

SDSPn/GRN: Slave DSP Heartbeat. Indicates that the SDSP has booted and is running. The rate is controlled by Timer 0 using DSP pin TOUT0.

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. The LED is controlled using DSP pin EXT_INT7 as a GPIO output.

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

START : On when the Program Reset Manager FPGA is configuring and the ROD is coming out of 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 multiplexor 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 sourced through a 510 ohm resistor for short circuit protection.

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.

1.1 Program Reset Manager FPGA

The Program Reset Manager (PRM) FPGA controls configuration and reset of the ROD and the interface with the VME host. When the ROD is powered on or the front panel reset switch is depressed, the PRM initializes the board by holding the DSPs in reset and transmitting all of the serial configuration data to the FPGA chips. The Master DSP is allowed to boot with the FPGAs. Master DSP tests the ROD Busy signal from the PRM to determine when the FPGAs have booted.

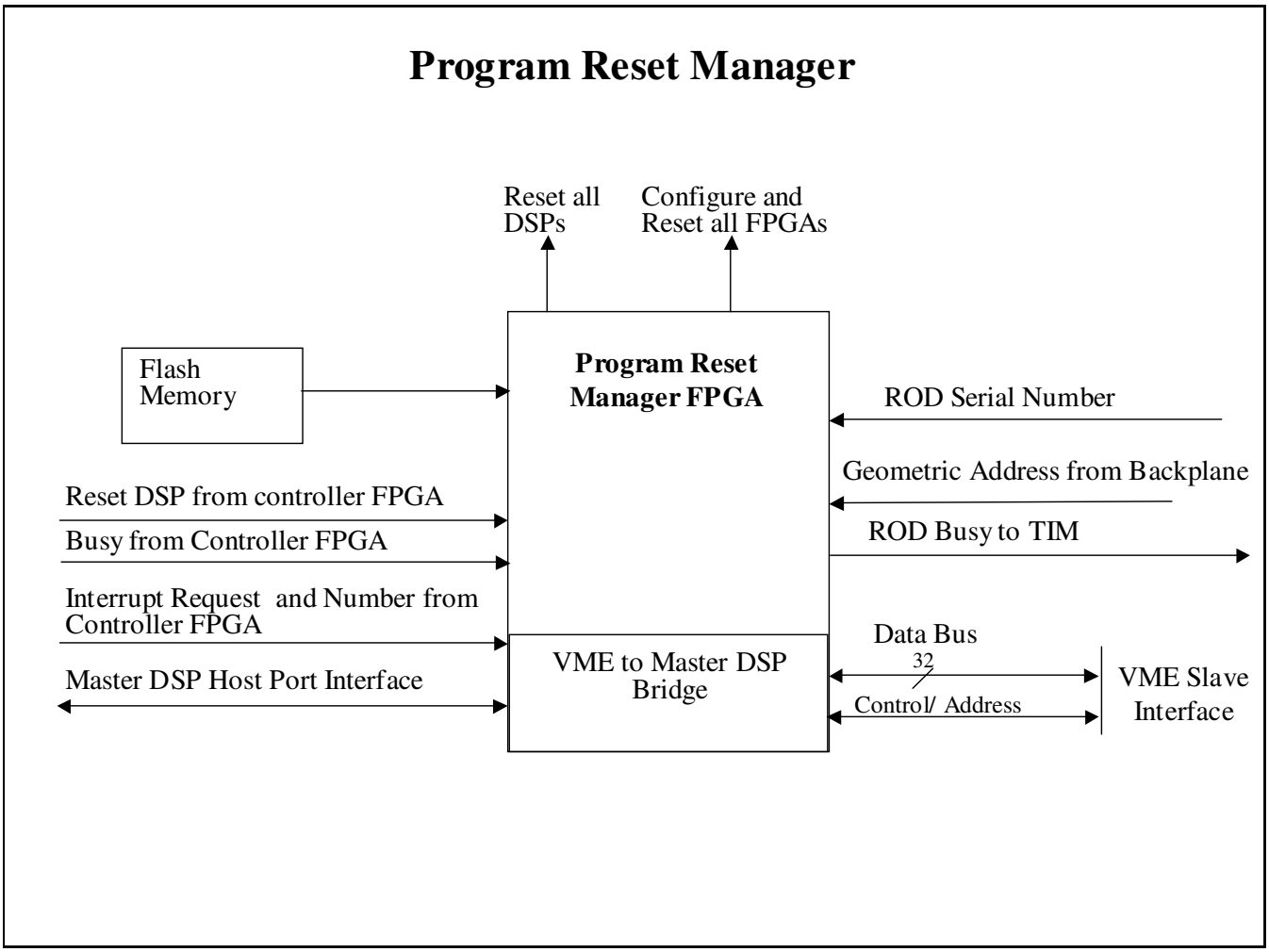


Figure 2: Program Reset Manager

The VME interface is then able to communicate to the ROD master DSP via the VME to HPI Bridge in the PRM, and the system can be initialized. Initialization includes loading operation specific Slave DSP (SDSP) code from the VME host. ROD Busy is asserted to the TIM during configuration. Busy will also be asserted in normal running if there is no buffer space available in the ROD. VME access is A32/D32 with programmed I/O or DMA. 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 (The PRM can be programmed to support this if a need develops).
Loading of new FPGA and MDSP code to the flash memories from a VME host is supported.

1.2 Controller

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

The Master DSP controls and coordinates operation of the entire ROD. It has access to the FPGAs and BOC through a register bank connected to one of its external memory interfaces; this also gives it access to the full memory range of the four slave DSPs through four separate registers attached to their host port interfaces. The DSP has two internal 64 KB memory blocks, which are used to store program instructions and data (separately), and an additional 256 MB of slow memory, for additional program storage and general data storage, attached to another of the external memory interfaces.

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

As an example, the primitive RW_REG_FIELD (read & write to a register field in the ROD's FPGA register interface) has five input fields including the register's identification number, the field's offset inside the register, the field width, a toggle indicating whether we are reading from or writing to the register and some input data (for reading, there is an output data field). If writing to a field inside the register, the master DSP will read the register, modify the desired field, and write the new value back into it.

A primitive's reply data, if there is any, is stored inside a separate "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 DSP. To control operation of the four slave DSPs, there are additionally two private "inter-DSP" primitive and reply buffers. The DSP 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 DSP can also run a separate class of "task" routines. 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. 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).

A major task on the master DSP is the histogram-control task, which controls the operation of the ROD during a calibration.

The FPGA controller is shown below:

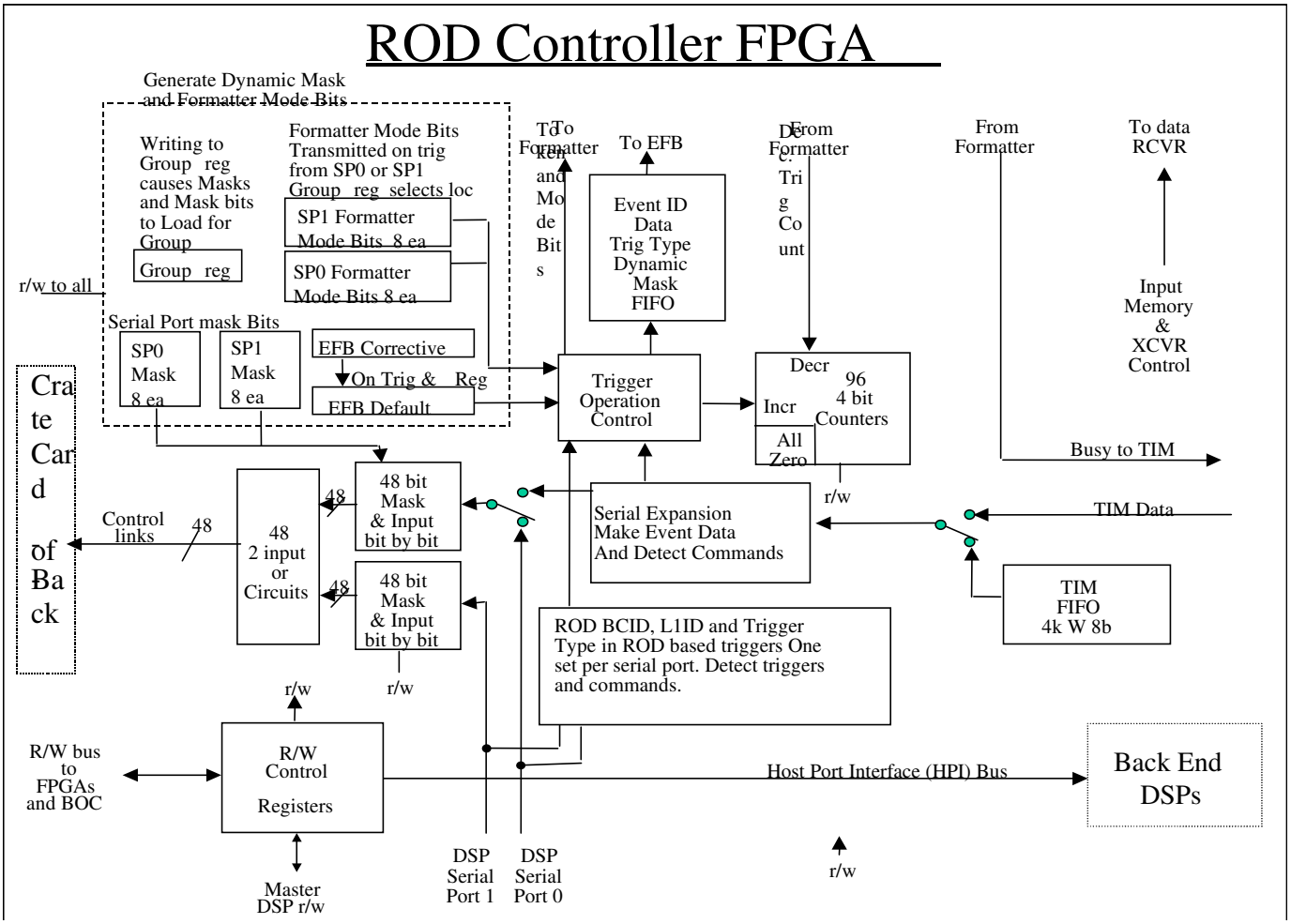


Figure 3: ROD Controller FPGA

Read/Write from the master DSP input is shown in the lower left of Figure 3. The R/W control bridges the MDSP EMIF port to the ROD R/W bus. The ROD R/W bus is connected to the Slave DSP HPI ports, the diagnostic FIFO memories (data receiver), and to all of the registers on the BOC and data path FPGAs. This path (programmed I/O or DMA) to the slave DSPs is used to load the slave code and control slave monitoring and operations.

In normal running the TIM supplies the trigger (middle right of figure 3) and event description (BCID, L1ID and Trigger Type). The trigger (one 25ns wide pulse) is detected in the Controller FPGA and expanded into the module trigger codes required by the SCT or Pixel FE modules. Calibration strobes and FE module reset commands are also detected and expanded as per the requirements. 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 ROD sends the Trigger code to the FE modules, a set of dynamic mode bits (2 bits per link 00b in normal running) are sent to all of the Formatters. After sending the L1 pulse to the ROD, 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 EFB. After a delay of a few clock ticks a token to start the readout of the link FIFOs is sent to the formatter. The delay is to allow the EFB to be ready for the data being pushed out of the formatters.

A variant of normal running 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.

The controller contains a counter for each link that is incremented for each trigger sent to a FE module and decremented when a trailer for each link is stored in the formatter de-randomizing buffers. The counter when all zero indicates that there is no data in 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.

The master DSP performs configuration of modules. It uses the master DSP multi channel buffered serial ports 0 (default) and 1 (corrective) to send data to the controller FPGA. The controller FPGA has 2 separate 48 bit wide masks that can be configured to send data to any selected FE module from either serial port. The ROD Data Receivers allow return configuration data to be trapped and inspected for data integrity.

The master DSP normally performs calibration of modules sending module configuration data and triggers on the command output serial links. In addition it uses BCID, L1ID and trigger type registers in the ROD Controller (on the lower middle of Figure 3) to build the event fragments. There is a separate set of registers for each serial port. The corrective (SP1) and default (SP0) formatter and EFB masks are used to determine what modules are active for calibration. At the start of a calibration sequence the resets are issued that zero the BCID and L1ID counters on the modules and in the controller FPGA. ATLAS trigger types are set with a write to the register associated with the serial ports. The upper left part of Figure 3 shows the serial port bit masks and the formatter mode bits. There are eight sets of these mode bits and mask bits for each serial port. These mode bits and bit masks are loaded at the beginning of a scan. Before serial data is sent the group register is set. Setting the FE mask register to a group number causes the group information to be loaded into mask bit register and the formatter mode bits to be loaded into a register for use. Module variables are set via the serial streams (calibration mode is off) as needed. Serial streams are then sent that contain calibration strobes and triggers to the previously selected module groups (up to 8 module groups that may require different masks). Masks are changed by the Master DSP based on the trigger stream needs. A trigger pattern in the serial stream causes BCID, L1ID, dynamic masks and trigger type to be processed in the normal manner. Routing of data to a particular DSP is controlled by the trigger type under the Master DSP control.

In the simplest (normal) form, each serial stream goes to one or more distinct modules. The calibration strobes and triggers for each scan point is repeated the required number of times. The scan variable is changed (not in calibration mode) and the sequence of triggers is repeated to build the histograms in slave DSP assigned. It is slightly more complex when 4 modules group (one trigger type for each slave DSP) are being triggered. In this case the master DSP must change the group number, trigger type and L1ID in the controller FPGA as needed.

In the case of cross talk or interference studies, the controller can issue calibration strobes and triggers to a module and a second strobe during readout of the first trigger data then trigger and read the second data set. The ROD is capable of routing the data to different or the same slave DSP. The ROD can also issue strobes and triggers to multiple modules while other modules receive time correlated strobes and triggers and route the data to the same or different DSPs. These correlation studies are available because the two serial streams from the master DSP are bit for bit matched in time. In all cases, the triggers in the two serial streams must be separated by a minimum of 5 40MHz clocks.

The SCT modules are expected to lose synchronization with the ATLAS L1ID due to upsets. The design provides for a correction of +1 or -1 to the L1ID as data is being taken on a non-interference basis. Synchronization problems are detected in the slave DSPs. The correction indicates what link L1ID needs to incremented or decremented. This value will be put in the EFB corrective dynamic mask. The formatter's corrective mask also requires mode bits that determine how each link is to be processed (01b indicates to skip the link and 11b indicates dump current event and read second). After the corrective masks are set the next trigger will execute the action of the masks once for the formatters and continuously for the EFB. The event fragments from each link are then synchronized with the ATLAS triggers. When a periodic reset is received the corrective action is removed from the EFB mask.

1.3 Data Receivers

The Controller FPGA controls all input memory functions. The basic operations include, passing normal data through with no trapping, trapping data from the input links, and playing of test data to the formatters.

The layout of the input memories (Data receivers) is shown below:

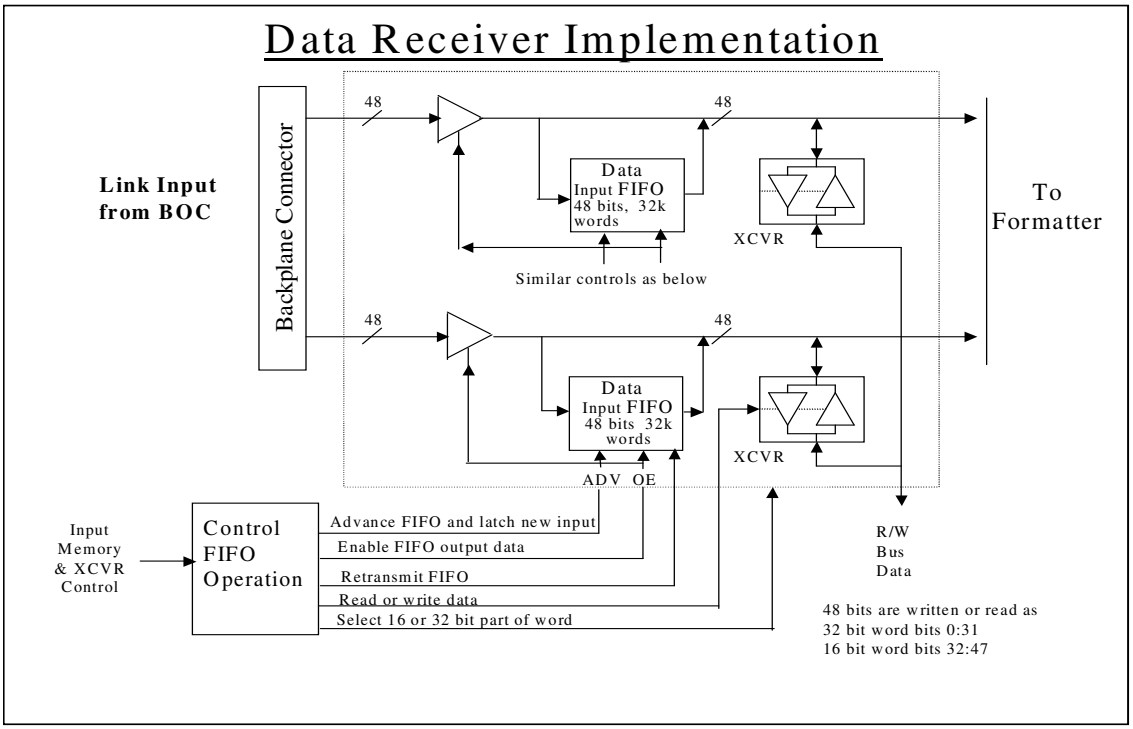


Figure 4: Data Receivers

Because the FIFOs are 48 bits wide and connected to a 32 bit wide bus, data must be written in and read out in 2 cycles. The first word is 32 bits wide, and the second is 16 bits wide. The 2 cycles represent 1 FIFO location. The control signal to perform these actions is provided by the controller FPGA. Data can be trapped in normal data taking but cannot be read out until the formatters have processed all of the event data, and triggers have stopped. When the ROD is not in normal running mode, the FIFOs can trap and read as required. This can be used during calibration of the timing of the BOC. The FIFOs can also be used to read configuration data back from the FE modules.

1.4 Formatter

The main function of the formatters is to convert the serial data from the modules to word data and provide de-randomizing buffering for the event fragments. They also detects module packet errors, provide signals to the controller FPGA to indicate that all link trailers have been received, respond to back pressure from the EFB and provide busy to the controller FPGA to indicate fatal errors (buffers full). The FPGA has separate versions of the code for SCT and Pixels. The SCT version has 12 links (6 modules) per formatter. The pixel has 4 links per formatter. Block diagrams of the formatters are shown below:

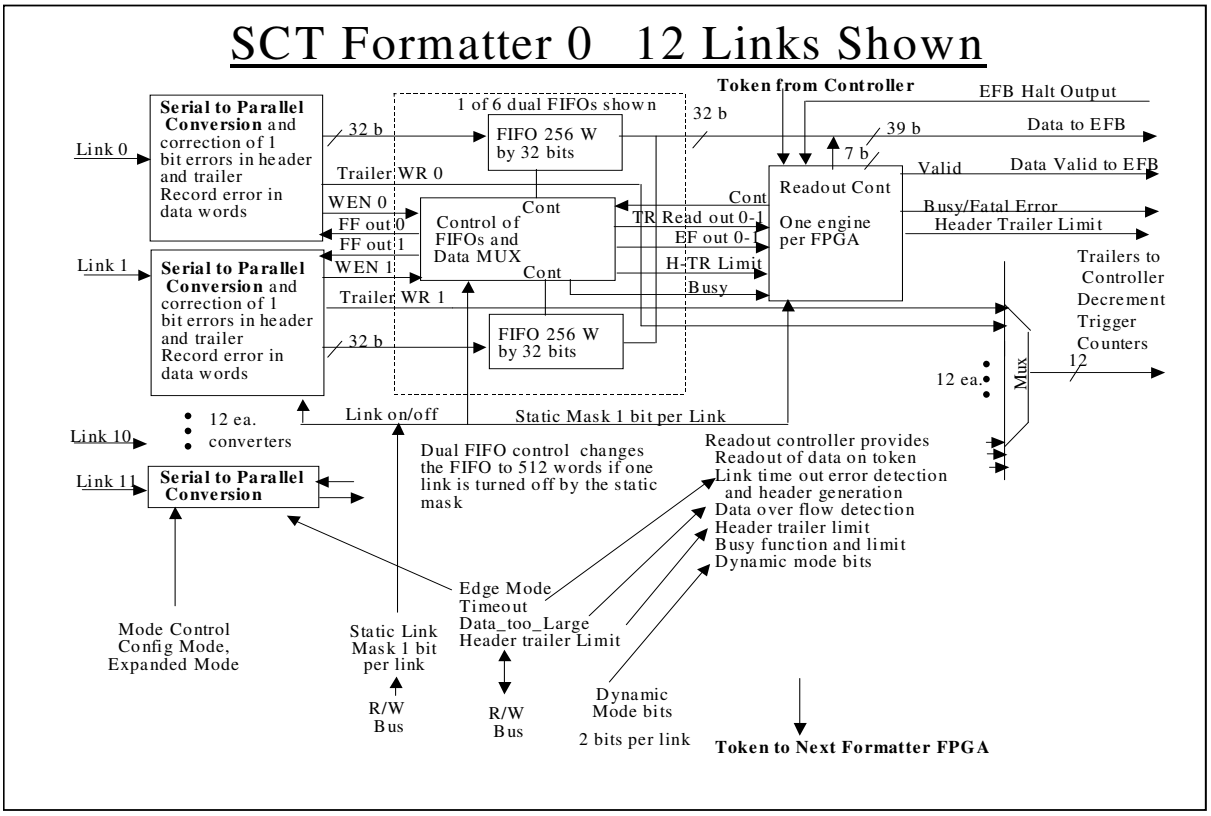


Figure 5A: ROD SCT Formatters

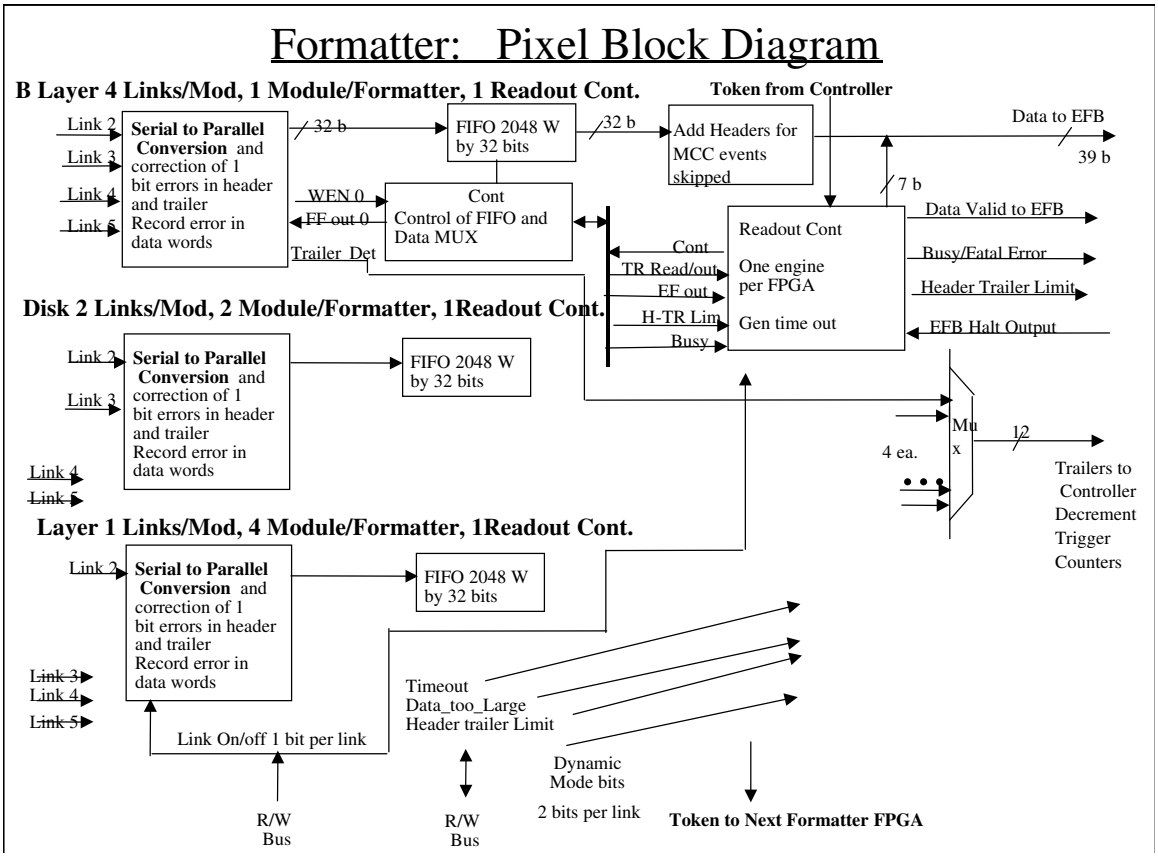


Figure 5B: ROD Pixel Formatters

The SCT formatter can operate in one of 3 modes. The Expanded Mode and Configuration Mode register bits control the formatter decoding modes. The modes are condensed (not Config & not Expanded), expanded (not Config & Expanded) or raw (Config mode bit) data. Details of the data Formats for the SCT and Pixel ROD are in section 3.1. Condensed mode is the normal data-taking format (one or two hits per 16 bits). If there is a framing error in the serial data the formatter will record the error and change to raw data mode for the remainder of the link data. Expanded mode uses 32 bits to store one or two hits. Raw mode places 8 bit of module data in 16 bits.

The Pixel Formatter with 4 data links per Formatter FPGA can support four 40MHz link inputs from the BOC.

One module of B-layer (2 80Mhz optical link per module) (1 module has 4 40 MHz ROD input per module) or
Two modules of forward disks (1 80MHz optical link per module) (1 modules has 2 40MHz inputs per module) or
Four modules of layer 1 (1 40MHz optical link per modules)(1 module has 1 40MHz input per module).

Selection of module type is done with jumpers or software.

The output of the Pixel Formatter, figure 5B, shows a box “Add header for MCC skipped event”. The MCC will write on the last good event how many events have been skipped. The ROD will remember this number and remove the error from the good event. It then will send the saved number of skipped event. Each event has an error flag. This performs the needed function of keeping event synchronization and informs offline of the number of skipped events for the module.

The SCT formatter formats and packs the link serial data bits 31:16 or 15:0 of the 32-bit data word. For example, an event (Expanded Mode) could be of the form,

Bits 31:16	Bits 15:0
Header L1 & BCID	Front End Number, Cluster Base Address and First Hit
One or Two Hits	Trailer

See section 3.1 for exact SCT and Pixel formats and examples. As 32 bits words are made they are written to the input link de-randomizing FIFO that is 512 words SCT and 2048 words pixels long. Under normal running condition with 95KHz trigger rate, full luminosity and 2 times min bias occupancy the buffers will have and mean occupancy of about 20 words. There are two programmable registers that determine special action. One is the header trailer limit that is expected to be set at about 4/5 of the FIFO size. When this limit is reached only header and trailers will be stored (one word per link hit SCT or twp words per link pixels). A bit in the data records this case. The other register is the busy limit expected to be set at about 9/10 of the FIFO size. When this limit is reach ROD busy is sent to the ATLAS via the TIM. If the events continue to come in a fatal error is generated and the ROD stops (busy asserted)(should never happen).

Formatter Error Detection:

The Formatter can detect and mark in the data header 1 bit errors, trailer 1 bit errors, condensed mode data errors, link readout time out errors, header trailer limit errors, and link data FIFO overflow errors. See section 3.1 for the location of errors in the data stream. 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. This also helps to prevent the mixing of data from time adjacent triggers.

Header Errors:

The SCT and Pixel 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 the header error bit is set in the link header.

SCT Headers Accepted:		
Data Pattern	Header Detected	Header Error
111010	Yes	No
1111010	Yes	Yes
011010	Yes	Yes
101010	Yes	Yes
110010	Yes	Yes
111110	Yes	Yes
111000	Yes	Yes
111011	Yes	Yes
1110100	Yes	Yes

Pixel Headers Accepted:		
Data Pattern	Header Detected	Header Error
11101	Yes	No
11100	Yes	Yes
11111	Yes	Yes
11001	Yes	Yes
10101	Yes	Yes
01101	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 as a 1 followed by 15 0's. The Pixel FE module trailer is defined as a 1 followed by 22 0's. 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.

Synchronization error detection:
If one of the field separator bits (sync bit) in the serial return data stream is incorrect or not detected, the link decoder will switch to raw data mode until a trailer 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.

Condensed mode error detection SCT:
The ROD variable “edge_mode” is used to determine if the input hit data is consistent with the state of this variable. If not a condensed mode error bit is set.

Link readout time out error detection:
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) error detection:
If the amount of returned link data causes the link FIFO occupancy to reach the header trailer limit, the Formatter will change to Header Trailer Limit mode. If the Formatter is currently receiving an event when the limit condition occurs, the decoder will stop writing data to the FIFO until a trailer is detected. The trailer will be written to the 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.

Data too large error detection (link data FIFO overflow errors):
This is a historic state that is not longer needed. If the event is too large (based on a register) a status bit is set.

In normal operation the MDSP via the controller FPGA sets the registers in the ROD that determine the following: (See the procedure section for exact details)

- Static masks (links that are active)
- Header Trailer Limit (link FIFO almost full)
- Busy Limit (link FIFO full)
- Link Readout Time Out Limit
- SCT Edge and Data Decoder Mode selection (Raw, Condensed or Expanded)

After the ROD Controller FPGA detects a trigger, dynamic mode bits, 2 per link, are sent to the formatters that determine how the data stored in the link FIFOs will be played to the EFB.

Link bits	Mode Bit Definitions
00	Normal data readout
01	Skip readout (used to resynchronize links to the ATLAS triggers)
10	Dump one event of link data (used in testing)
11	Dump first event, readout second (used to resynchronize links to the ATLAS triggers)

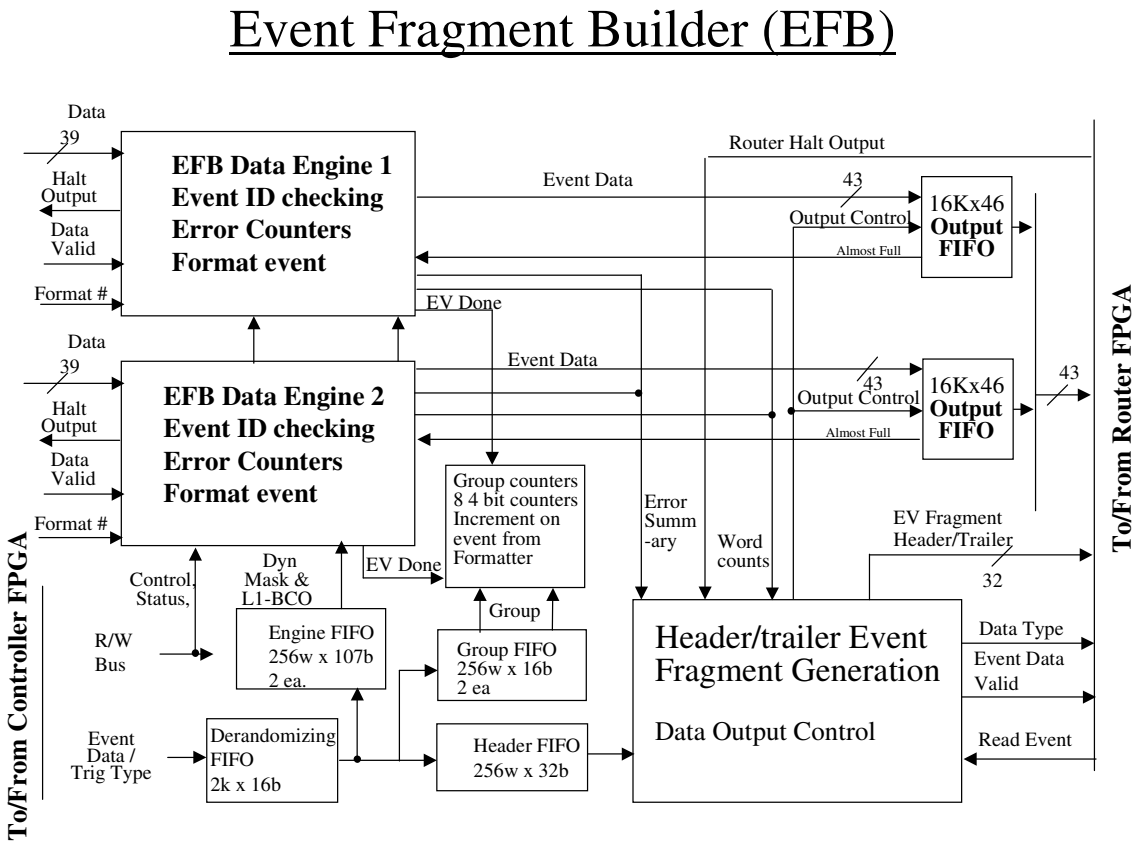
A trigger is then issued to the Formatters to start readout.

The 8 formatter chips are arranged in two banks of 4. This architecture allows the formatters to push the data out at bandwidths approaching 80MHz, twice the output bandwidth to the S-Link. With 2 banks of Formatters, the bandwidth limit of the ROD will be determined by the bandwidth of the S-Link.

1.5 Event Fragment Builder

The Event fragment builder has two engines that collect the formatter output and place the reformatted data in large FIFOs (16k words) at the output of the EFB. These de-randomizing FIFOs can store about 200 events at full luminosity. In addition L1ID, BCID is checked and error counting is performed. The last function is to output the event header, trailer and data.

Figure 6: EFB



Processing of an event starts with receiving event descriptor data from the controller FPGA. This data consists of the EFB dynamic masks, and event header data (see section 3.2 for details). The event descriptor is stored in a de-randomizing FIFO. Then passed on as required to the engine FIFO and header FIFO. The secondary FIFOs are used in processing and event.

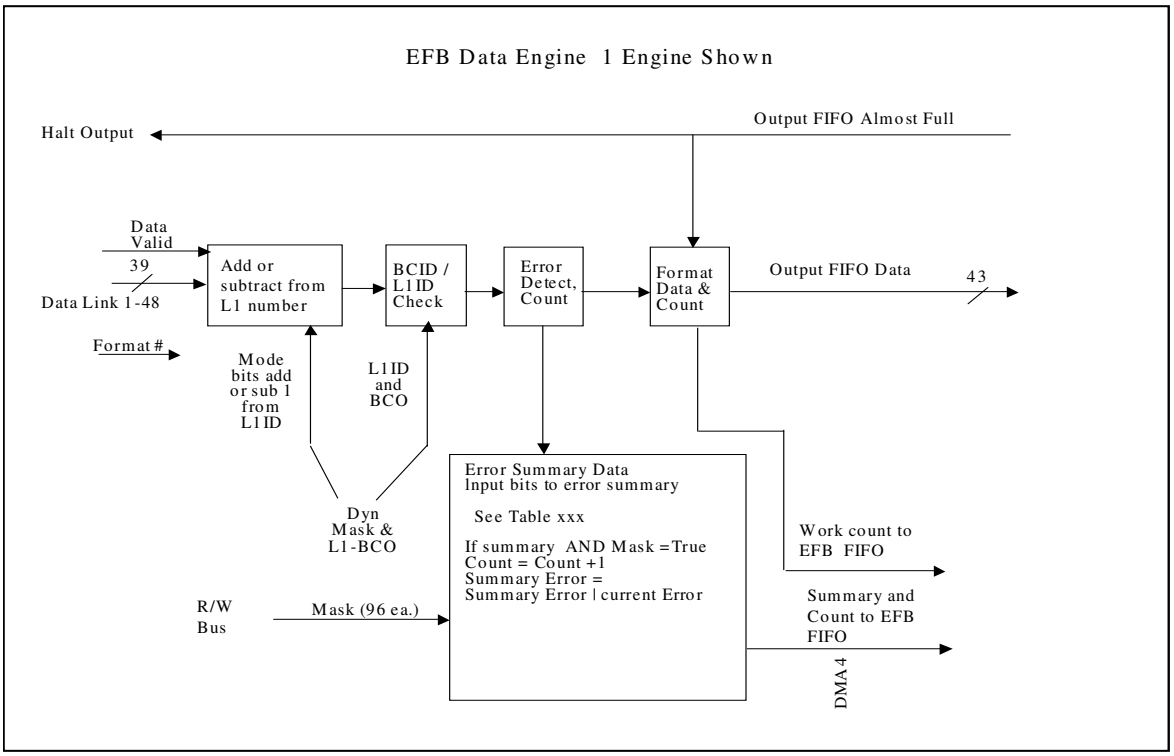


Figure 7: EFB data engine

Input data pushed from the formatters is processed with the EFB dynamic mask. This mask determines if the L1ID number is incremented, decremented or unchanged (used in link resynchronization). The BCID and L1ID are then checked against the ATLAS BCID and L1ID. If the values in the link headers do not match those supplied by the TTC System, errors will be marked and sent along in parallel with the link headers. The next step is to process all of the errors in the pipeline. A static mask determines if a link is to be processed. The errors processed for each link are shown below:

Mask Bit Location	Error Type
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	(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

For each link in the event the error bits above are counted, and AND'ed with the complement of the Error Mask register and OR'ed with errors from all of the links in a fragment. The result is that the number and type of errors in a fragment will be recorded and inserted into the Event Trailer.

After the error detection block the data is formatted and written to the output FIFO. EFB formats are shown in section 3.2. Trailers are removed if there is not an error in the link. If there are no errors in the link and no data the header and trailer are deleted. After unnecessary half words are removed, it re-packs all the data into 32bit words so that there are no gaps in the data. A word count of the data in the FIFO is then sent to the header section.

After error data and the data word counts are available from both EFB engines and trigger type data, an event is delivered to the router (data is pushed). An event consists of the header, data section and trailer.

The group counters used in calibration (4 bit counter per group, 8 groups) are increment when an event has left the Formatter for each EFB engine. The specific group counter is determined from the group FIFOs. This is used in calibration to know when and group has been read out of the formatter. A new trigger can then be issued subject to space available in the slave DSP DMA buffers.

If back pressure is supplied from the router, DSP asserts Stop (only calibration mode) or the S-Link asserts Xoff, data will not be transmitted, but the EFB can still process data and write to the output memories. If any of the internal EFB FPGA FIFOs are almost full or the output FIFOs are almost full, the FIFO controller will be told to pause and back pressure will be applied to the Formatters.

1.6 Router

The Router has two main functions. One is to pass date to the S-Link. The other is to trap and send data to the 4 slave DSPs. The router block diagram is shown below:

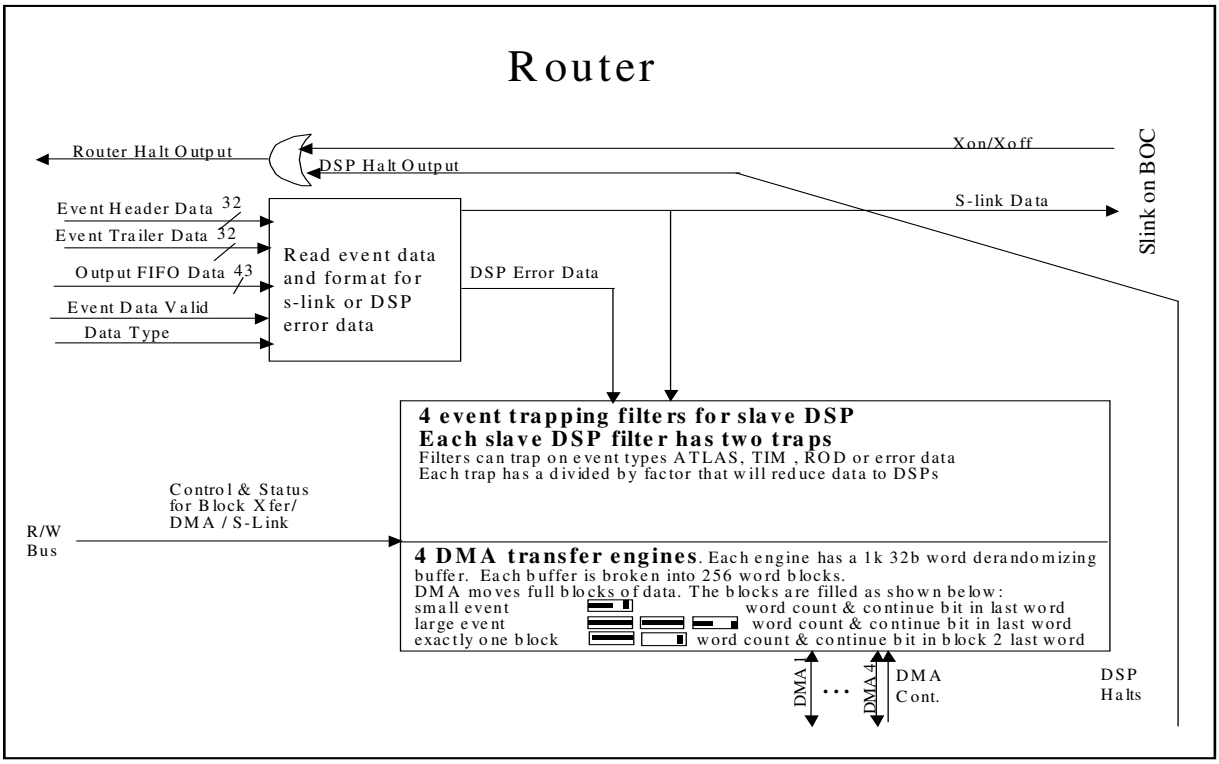


Figure 8: Router

Event data is passed through the router with only minor format changes. The interface to the S-Link is implemented. When test data is needed for the S-Link the ROD is taken out of normal running mode and the FIFOs in the data receiver or at the output of the EFB can be loaded. That data is then played through the ROD.

Trapping of data for the slave DSP can also be performed. There is a separate trap filter and 4 blocks of 256 words deep FIFO for each DSP. The trap can be set to look for ATLAS, TIM or ROD event types. The trap type data is found in the first word of the Event Header, and then removed so that the first event word to the S-Link is in the correct format. Traps also have a division factor to control the number of events that are sent to the DSPs. The Router can send the trap data to the DSPs in 2 formats. S-Link Data format is in the same format that is sent to the ROS, and is used in normal

running and calibration. Error Data format is the link data as is from the EFB routed to the DSPs. (See section 3.3 router formats). The main difference between Error ands-Link data format is a two bit per link error field in the trailer. The Error Data format is intended for link resynchronization in a slave DSP.

The 4 256 word data buffer for the DMA transfer per DSP store the data in the required format. The last word of an event block contains the data word count, trap identification and error bits. Blocks of 256 words are DMAed to the slave DSP at a 80MHz rate. Filling of the data blocks is at 40MHz. Since data is read out faster than it is read in events that are larger than 4 can be transferred. In the special case where slave buffer resources are consumed the event is terminated on the current block and the trailer is marked with and overflow bit.

1.7 Slave DSPs

As mentioned previously, 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. Like the master DSP, the slaves have two internal 64 KB memory blocks; they have 32 MB of external (slow) memory attached to two other external memory interfaces.

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 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 freed).

2.0 Modes of Operation

ROD Controller Modes

3.0 Data Formats

Data formats for Formatter, Event fragment Builder and Router are shown in sections 3.1 to 3.3.

3.1 Formatter Formats

SCT Serial Input Data

The module data format is shown below. For a more detailed description of the data format transmitted by a SCT Module please see the ABCD3T Chip Specification V1.2

Normal data

Header	DT	L1	BC ID						Trailer
<11101>	<0>	<nnnn>	<bbbb,bbbb>	<**1**>	<data block>	<**1**>	<data block n>		<1000,0000,0000,0000>

Data block

Leader	Chip Addr.	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>

Leader	Chip Addr.	Hit Channel Address		Hit Ch3		Hit Channel Address		Hit Ch5		Hit Ch6
<01>	<1101>	<0000011>	<**1**>	<ddd>	<**01**>	<0000101>	<**1**>	<ddd>	<**1**>	<ddd>

Hit Pattern

Detector Alignment 1xx or x1x or xx1; Level x1x; Edge 01x; Test xxx

Data block no hit <001>

Error data

Leader	Chip Addr.	Error Code	
<000>	<aaaa>	<eee>	<**1**>

In condensed or expanded data modes, if the link decoder misses a sync bit (bold bits note as shown), the Formatter will switch to raw data mode.

SCT Formatter Data Format

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

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)

Formatter Output (bits 31:0 bits 37:32 for all words)

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

Condensed example

```
W1 001pLLLLBBBBBBBB 1FFFFCCCCCCCxfx0   Header + 1 hit
W2 010zhvxxxxxxxxxx 0000000000000000   Trailer
W1 001pLLLLBBBBBBBB 1FFFFCCCCCCCsfx1   Header + 2 hit
W2 1FFFFCCCCCCCxfx0 010zhvxxxxxxxxxx   1 Hit + Trailer
```

Raw data example

```
W1 001pLLLLBBBBBBBB 011nnnxxBBBBBBBB   Header + 8 bit
W2 011nnnxxBBBBBBBB 010zhvxxxxxxxxxx   8 bit + trailer
```

Expanded example

```
W1 001pLLLLBBBBBBBB 1FFFFCCCCCCC0DDD   Header + 1 hit
W2 1xxxxxxx1DDD1DDD 1xxxxxxx0xxx1DDD   2 hit + 1 hit
W3 010zhvxxxxxxxxxx 0000000000000000   trailer
```

Condensed to Raw on Error Correction

```
W1 001pLLLLBBBBBBBB 1FFFFCCCCCCCsfx1   Header + 2 hit
W2 1FFFFCCCCCCCxfx0 010zhvxxxxxxxxxx   1 hit + Trailer
Error on 1 before 3rd hit (could have done it before 2nd hit)
W1 001pLLLLBBBBBBBB 1FFFFCCCCCCCsfx1   Header + 2 hit
W2 011nnnxxWWWWWWWW 010zhvxxxxxxxxxx   raw + trailer
```

Format will switch to raw on the following errors for the rest of the event.

Key :

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

Pixel Serial Input Data

The Pixel FE module data format is shown below. For a more detailed description of the data format transmitted by a Pixel FE Module please see the MCC-D2 Specifications, Output Data Format.

Normal data

Pixel Formatter Data Format

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	

Formatter Output (bits 31:0 for all words)

Name	Bits [31:0]
Header	001PxxxxxxxxAAAALLLLLLLLBBBBBBBB
Trailer	010ZHVxxxxxxxxxxxxxxxxxxxxxxxxxx
Hit	100xFFFFRRRRRRRRxxxCCCCCTTTTTTT
FE Flag Error (Old)	0000FFFFxxxxxxxxxx11110FFFFEEEE
FE Flag Error (New)	0001FFFFxxx11111eeeeeeeeEEEEEEEE
Raw Data	011DDDDDDDDDDDDDDDDDDDDDDDDDDD
Time Out Data	00100000000000000100000000000000

Key:
A = Number of Accepts per L1 Trigger
B = BCID
C = Pixel Column
D = Raw Data
E = FE Error Code
e = MCC Error Code
F = FE Number
H = Header Trailer Limit Error
L = L1ID
N = Count of raw data bits + 1
P = Preamble Error
R = Pixel Row
T = Time over Threshold value
V = Data Overflow Error
x = do not care (The ROD fills these with 0's)

3.2 EFB Formats

Event Data /Trigger type from Controller FPGA

This is the Event ID data sent from the controller to the EFB. The data is composed of 16 16-bit words. The data in the Event ID packet is used to form the event header and process events correctly in the data flow path.

Word	Description	Engine FIFO	Header FIFO
0	L1 ID [15:0]	[3:0]	Yes
1	ECR ID [7:0] & L1 ID [23:16]	No	Yes
2	BC ID [11:0] & 0 & ROL Test Block Enable & BOC Clock OK & TIM Clock OK	[7:0]	Yes
3	ROD Event Type [5:0] & TIM Event Type [1:0] & Atlas Trigger Type [7:0]	No	Yes
4	Dynamic Mask [7:0]	Yes	No
5	Dynamic Mask [15:8]	Yes	No
6	Dynamic Mask [23:16]	Yes	No
7	Dynamic Mask [31:24]	Yes	No
8	Dynamic Mask [39:32]	Yes	No
9	Dynamic Mask [47:40]	Yes	No
10	Dynamic Mask [55:48]	Yes	No
11	Dynamic Mask [63:56]	Yes	No
12	Dynamic Mask [71:64]	Yes	No
13	Dynamic Mask [79:72]	Yes	No
14	Dynamic Mask [87:80]	Yes	No
15	Dynamic Mask [95:88]	Yes	No

Dynamic Mask [95:0] is 2 bits per link, used to correct module synchronization.
DM Bits [1:0] “00” no change to L1 ID for this event
DM Bits [1:0] “01” Increment L1 ID for this event
DM Bits [1:0] “10” Decrement L1 ID for this event

Event Header

Word	Contents	Comment
0	0xB0FTTTTT	Beginning of fragment marker/ T = Router Trap Type Data
1	0xEE1234EE	Start of header
2	0x9	Header size
3	0x30000000	Format Version Number (Ver 3.0)
4	0x001XNNNN Pixel 0x002XNNNN SCT	Source Identifier N = Module ID, X = LS Nibble of Sub-detector ID
5	RN ID	Run Number
6	L1 ID	Level 1 ID
7	BC ID	Bunch crossing ID
8	L1 TT	ATLAS Level 1 trigger type
9	DET TT	Detector event type ROD or TIM

Event Trailer

Word	Contents	Comment
0	Error Flags	Status 1: Bit error see EFB errors [31:0]
1	Error Count and Static Flags	Status 2: Count of words with error [15:0] Static Error Flags and ROL Status
2	0x2	Number of status words
3	Ndata	Count of data words
4	0x1	Status block position: 0/1 = before/after data
5	0xE0F00000	End of fragment marker

EFB Output Link Format

SCT data is formatted in 16 bit packets packed into 32 bit words.

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

EFB Output (bits [31:0] bits for all words)

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

EFB Output examples for two SCT events

Bits [42:32]	Bits [31:16]	Bits [15:0]	Notes
	Event header 9 words		Normal event
blKtMMMMMMM	Link header	Condensed	
blKtMMMMMMM	Condensed	Link header	Trailers are removed if no trailer errors are present.
	Condensed	Condensed	
blKtMMMMMMM	Link header	Condensed	
	Condensed	Condensed	
	Condensed	Condensed	
	Event trailer 6 words		
	Event header 9 words		
blKtMMMMMMM	Link header	Condensed	
blKtMMMMMMM	Condensed	Link header	
	Condensed	Condensed	
blKtMMMMMMM	Link trailer	Link header	Error in trailer
	Condensed	Condensed	
blKtMMMMMMM	Link header	Condensed	
	Condensed	Link trailer	Special trailer fill
	Event trailer 6 words		

If there are **no errors** in the **trailer and header** and **no data** the l**ink header and trailer** are **not written**.

EFB Output Link Data Format – Pixel

All Pixel data is in 32 bit words. Data bits 42-32 are valid for header word only.

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

EFB Output (bits 31:0 bits for all words)

Name	Bits [31:0]
Header	001PxxxxxxxxAAAALLLLLLLLBBBBBBBB
Trailer	010ZHVxxxxxxxxxxxxxxxxxxxxxxxxxx
Hit	100xFFFFRRRRRRRRxxxCCCCCTTTTTTTT
FE Flag Error (Old)	0000FFFFxxxxxxxxxxx11110FFFFEEEE
FE Flag Error (New)	0001FFFFxxx11111eeeeeeeeEEEEEEEE
Raw Data	011DDDDDDDDDDDDDDDDDDDDDDDDDDDD
Time Out Data	00100000000000000100000000000000

EFB Output examples for two Pixel events

Bits [42:32]	Bits [31:0]
	Event header 9 words
blKtMMMMMMM	Link header
	Hit
	Hit
	Link trailer
blKtMMMMMMM	Link header
	Hit
	Hit
	Hit
	Link trailer
	Event trailer 6 words
	Event header 9 words
blKtMMMMMMM	Link header
	Hit
	Hit
	Link trailer
blKtMMMMMMM	Link header
	Hit
	Hit
	Hit
	Link trailer
blKtMMMMMMM	Link header
	Hit
	Hit
	Hit
	Hit
	Link trailer
	Event trailer 6 words

Key :
A = Number of Accepts per L1 Trigger
b = BCID error
B = BCID
C = Pixel Column
D = Raw Data
E = FE Error Code
e = MCC Error Code
F = FE Number
H = Header Trailer Limit Error
l = L1 error
L = L1ID
M = link number
N = Count of raw data bits
P = Preamble Error
R = Pixel Row
t = time out error
T = Time over Threshold value
V = Data Overflow Error
x = do not care
Z = trailer bit error

3.3 Router Formats

S-link Module Output Format (sent to DSP and S-link)

The Router modifies the output data format of the module data before it is sent out on the S-Link Bus. The L1ID and BCID is replaced with error codes and the link number as shown below:

Router Output – SCT Format (bits 31:0 bits for all words)

Name	Bits [15:0] or [31:16]	EFB Output
Header	001ptlbKxMMMMMMM	001pLLLLBBBBBBB
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 :
x= do not care (The ROD fills these with 0's)
B=BCID
W=raw data
C=cluster base address
D=3 bit hit data
E=ABC error code
F=FE number
h=header trailer limit error
L=L1ID
n=count of raw data
f=error in condensed mode data, 1st hit
s=error in condensed mode data, 2nd hit
p=preamble error
z=trailer bit error
v=data overflow error
t=time out error
K=condensed mode
l=L1 error
b=BCID error
M=link number

Router Output – Pixel (bits 31:0 bits for all words)

Name	Bits [31:0]	EFB Output
Header	001PtlbxxMMMMMMMLLLLLLLLBBBBBBBB	001PxxxxxxxxAAAALLLLLLLLBBBBBBBB
Trailer	010ZHVxxxxxxxxxxxxxxxxxxxxxxxxxx	
Hit	100xFFFFRRRRRRRxxxCCCCCTTTTTTT	
FE Flag Error	0000FFFFxxxxxxxxxx11110FFFFEEEE	
FE Flag Error	0001FFFFxxx11111eeeeeeeeEEEEEEEE	
Raw Data	011DDDDDDDDDDDDDDDDDDDDDDDDDDD	
Time Out Data	00100000000000000100000000000000	

Bits [42:32] ::= blxtMMMMMMM

Key :

A = Number of Accepts per L1 Trigger
b = BCID error
B = BCID
C = Pixel Column
D = Raw Data
E = FE Error Code
F = FE Number
H = Header Trailer Limit Error
l = L1 error
L = L1ID
M = Link number
N = Count of raw data bits
P = Preamble Error
R = Pixel Row
t = time out error
T = Time over Threshold value
V = Data Overflow Error
Z = Trailer bit error
e = MCC Error Code
x = do not care (The ROD fills these with 0's)

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:

Event Header

Word	Contents	Comment
0	0xB0F00000 + UCTRL	Beginning of fragment marker
1	0xEE1234EE	Start of header
2	0x9	Header size
3	0x30000000	Format Version Number (Ver 3.0)
4	0x001XMMMM Pixel 0x002XMMMM SCT	Source Identifier M = Module ID, X = LS Nibble of Sub-detector ID
5	0xTTSSSSSS	Run Number: T = Run Type >> 0x00 > Physics: 0x01 > Cal 0x02 > Cosmics: 0x0f > Test S = Sequence within Run Type
6	0xEELLLLLL	Extended Level 1 ID: E = ECR ID, L = L1ID
7	0x00000BBB	Bunch crossing ID
8	0x000000AA	ATLAS Level 1 trigger type
9	0x00RR000T	Detector event type R = ROD or T = TIM

Event Trailer

Word	Contents	Comment
0	Error Flags	Status 1: Flagged Data Errors
1	Error Count and Static Flags	Status 2: Count of errors in Event Fragment Static Error Flags and ROL Status
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

Bit Location	ERROR FLAGS: STATUS WORD 1
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	(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

Bit Location	ERROR STATUS: STATUS WORD 2
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

End of Block (sent to DSP)

The Router generates an end of block word that is located in 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:

Contents	Comment
01000000000t11dewwwwwwwwwwwwwwww	t = trap num, d = NOT data loss (FIFO full) e = NOT error (one or more links has an error) w = event word count

Error data Output Format for the DSP

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 changes as shown below:

Router error data output same as EFB Output (bits 31:0 bits for all words)

Name Bits 15:0 or 31:16
header 001pLLLLBBBBBBB
All other words are also the same.

Router Error Event Format (DSP only)

Event Header

Word	Contents	Comment
0	0xB0F00000	Beginning of fragment marker
1	0xEE1234EE	Start of header
2	0xTTSSSSSS	Run Number: T = Run Type >> 0x00 > Physics: 0x01 > Cal 0x02 > Cosmics: 0x0f > Test S = Sequence within Run Type
3	L1 ID	Level 1 ID
4	BC ID	Bunch crossing ID
5	L1 TT	ATLAS Level 1 trigger type
6	DET	Detector event type ROD or TIM

Event Trailer

Word	Contents	Comment
0	Error Flags	Status 1: Bit error see EFB errors [31:0]
1	Error Count and Static Flags	Status 2: Count of words with error [15:0] Static Error Flags and ROL Status
2	Value	Error status for links 0–15
3	Value	Error status for links 16–31
4	Value	Error status for links 32–47
5	Value	Error status for links 48–63
6	Value	Error status for links 64–79
7	Value	Error status for links 80–95
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:

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

ROL Test Output Format for the Slink

The Router supports a test mode that transmits a test sequence to the S-Link and SDSPs for diagnostics and test. All control bit are set in the RCF FPGA. The Router automatically identifies a ROL test block that has been received from the EFB and transmits it to the S-Link. The data format is as follows:

ROL Test Block Format – The ROL Test Block in the Silicon ROD is different from the block specified in Atlas Doc No. ATC-TD-ES-0002. The differences are shown below.	**Data**	**Offset**
Beginning of Fragment Control Word	0xB0F00000	
ROD Header Marker	0xEE1234EE	0
Header Length	0x00000009	1
Format Version	0x03000000	2
Source ID	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	

4.0 Operation Procedures

4.1 ROD Controller FPGA (RCF) Functions

Description

The ROD Controller FPGA is the real time controller for the ROD Electronics board.

4.1.1 Functional Blocks

A. Address Decoder / Data Bus Bridge
   The ROD Controller FPGA interface between the Master DSP External Memory Interface (EMIF) and all ROD peripherals. The main functions include the algorithm to allow single cycle access between the Master (D32 EMIF) and Slave DSP (D16 Host Port) chips, and read/write access to the BOC. This functional block also allows access to all of the Formatter, EFB, and Router Read/Write registers, and to all of the diagnostic FIFO memories.

B. Register Block
   This block defines, and allows read/write access to all of the internal registers in the RCF. This block controls the read/write access to the internal TIM FIFO in the ROD that is used to simulate operation with the TIM. The internal registers and bit definitions are listed in Appendix A.

C. Debug Memory FIFO Interface
   Generates control signal for the Diagnostic FIFO Memories.

D. Test Bench Interface
   Provides internal test algorithms to allow testing, diagnostics on the ROD.

E. Trigger Signal Decoder
   Receives data from the TIM and the Diagnostic Internal TIM FIFO. Recognizes and issues L1 Triggers, BCR commands, ECR Commands, and CAL Commands. Decodes the L1ID, the BCID, Atlas trigger type, and the TIM trigger type data sent by the TTC system on an Event basis.

F. EFB Header Dynamic Mask Encoder
   Generates and sends the Event ID and the default and corrective Dynamic Mask information to the EFB.

G. Formatter Dynamic Read Out Mode Bits Encoder
   Generates and sends the default and corrective Dynamic Read Out Mode Bits used by the Formatters to control proper read out of the Events stored in the Link FIFOs.

H. FE Command Pulse Counter
   Increments when and L1 Trigger is received, decrements when the token has been issued to the Formatters

I. FE Command Processor
   Provides the fan out of Command Streams from the TIM and MDSP serial ports. The command mask allows command generated by the TIM to be sent to one set of modules, and commands generated by the MDSP SP1 to be sent to a different module group simultaneously.

J. Front End Occupancy Counters
   Increments when a L1 trigger is issued to the FE chips, decrements when a trailer is written in to the Link Formatter FIFO.

K. Serial Port Trigger Command Detector
   Used in calibration mode to detect command types transmitted by the MDSP.

L. Internal Scan Engine/Trigger Generator

M. RoL Test Block Trigger/Pattern Generator

4.1.2 Initial ROD Controller FPGA states

When the RCF is configured at power up or when a configuration command is sent from the PRM, the state of the device is as follows:

a. All functional blocks turned off
b. FE Mask 0 all off
c. FE Mask 1 all off
d. All counters reset to 0
e. All internal BRAM blocks initialized to 0

When the RCF is reset by the PRM, the state of the device is as follows:

f. All functional blocks turned off
g. FE Mask 0 all off
h. FE Mask 1 all off
i. All counters reset to 0
j. All internal BRAM blocks initialized to retain current values

4.1.3 FE Command Data Streams and Output Masks

The ROD can output Fast, Slow and Calibration Commands to all or any combination of FE Modules. The block in the ROD Controller FPGA (RCF) that generates the FE Commands is called the FE Command Processor, and is enabled by setting the appropriate bit in the RCF Main Control Register. The sources for the FE Command Data Streams are the TIM Electronics Board, or the 2 Serial Communication Ports of the Master DSP (SP0 and SP1).

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

TIM 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 ROD Controller FPGA (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 the FE Modules that will receive command streams. At power-on or reset, all of the outputs of Output Mask 0 are on, and all of the outputs of Output Mask 1 are 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 changed 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.

The following Functional Procedures will define the steps required to enable the FE Command Processor, select between the TIM or MDSP SP0 for the Mask 0 outputs, set a new Output Mask in all ROD Modes, and send different types of FE Commands.

Functional Procedures:

Setup ROD to output FE Command Streams using the TIM as the source for Mask 0.

1. Enable the FE Command Processor and select the TIM as the Source for Mask 0. Using the MDSP primitive **RW_REG_FIELD**, set bits 0, and 1 of the RCF Main Control Register (RRIF_CMND_1: 0x00404410).

Setup ROD to Output FE Command Streams using the SP0 as the source for Mask 0.

1. Enable the FE Command Processor and select the SP0 as the Source for Mask 0. Using the MDSP primitive **RW_REG_FIELD**, set bit 0 and clear bit 1 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

SP1 is always the source for Mask 1

Change FE Command Mask in "Data Taking" mode

1. Set the FE Command Mask Registers to output commands to the selected FE Modules. Using the MDSP primitive **RW_REG_FIELD**, set the required values in the FE Command Link Mask0 LO (FE_CMND_MASK_0_LO: 0x00404430) and FE Command Link Mask0 HI (FE_CMND_MASK_0_HI: 0x00404434). Check that the values in FE Command Link Mask1 LO (FE_CMND_MASK_1_LO: 0x00404438) and FE Command Link Mask1 HI (FE_CMND_MASK_1_HI: 0x0040443C) do not conflict with the Mask 0 values.
2. Set the "New Mask Ready" Bit. Using the MDSP primitive **RW_REG_FIELD**, set bit 2 of the RCF Main Control Register (RRIF_CMND_1: 0x00404410).
3. The new mask will be used when the next L1A is sent from the TIM.

Change FE Command Mask in all other modes

1. Set the FE Command Mask Registers to output commands to the selected FE Modules. Using the MDSP primitive **RW_REG_FIELD**, set the required values in the FE Command Link Mask0 LO (FE_CMND_MASK_0_LO: 0x00404430) and FE Command Link Mask0 HI (FE_CMND_MASK_0_HI: 0x00404434). Check that the values in FE Command Link Mask1 LO (FE_CMND_MASK_1_LO: 0x00404438) and FE Command Link Mask1 HI (FE_CMND_MASK_1_HI: 0x0040443C) do not conflict with the Mask 0 values.

2. Set the “Config/Cal Mask Load Enable” Bit. Using the MDSP primitive **RW_REG_FIELD**, set bit 20 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).
3. Set the “New Mask Ready” Bit. Using the MDSP primitive **RW_REG_FIELD**, set bit 2 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).
4. The new mask will be used when the next commands are sent from the Master DSP.

4.1.4 FE Occupancy Counters

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.

Functional Procedures:

Enable FE Occupancy Counters.

1. Using the MDSP primitive **RW_REG_FIELD**, set bit 3 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Reset or mask the FE Occupancy Counters.

1. Using the MDSP primitive **RW_REG_FIELD**, write a value to the FE Occupancy Counter Reset Registers (FE_OCC_CNTR_RESET(0): 0x004044A0, FE_OCC_CNTR_RESET(1): 0x004044A4, FE_OCC_CNTR_RESET(2): 0x004044A8) that corresponds to the link counter that require a reset. (See Appendix A for more details on bit mapping)
2. Clear all of the FE Occupancy Counter Reset Registers to enable counting

4.1.5 FE Command Pulse Counter / Command Cycle Generator

The FE Command Pulse Counter stores the L1 Triggers as they arrive from the TIM, the MDSP serial ports, or the internal diagnostic TIM FIFO. 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 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 Event Fragment Builder 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.

Functional Procedures:

Reset FE Command Pulse Counter / Command Cycle Generator

1. Using the MDSP primitive **RW_REG_FIELD**, set bit 4 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Enable FE Command Pulse Counter / Command Cycle Generator

1. Using the MDSP primitive **RW_REG_FIELD**, clear bit 4 and set bit 5 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Load Maximum FE Command Pulse Counter Value

1. With FE Command Pulse Counter Enabled, set then clear bit 6 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

4.1.6 Event ID Information (L1ID, BCID, ECR ID, Atlas and TIM Event Type)

The TTC Trigger Signal Decoder block de-serializes the Event ID information supplied to the ROD by the TIM over the TTC Bus in Normal Data Taking Mode. The L1ID, BCID, Atlas Event Type and TIM Event Type values are decoded in this functional block. The L1ID and BCID are transmitted as one 36 bit serial stream, and the event types are transmitted as one 10 bit serial stream.

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

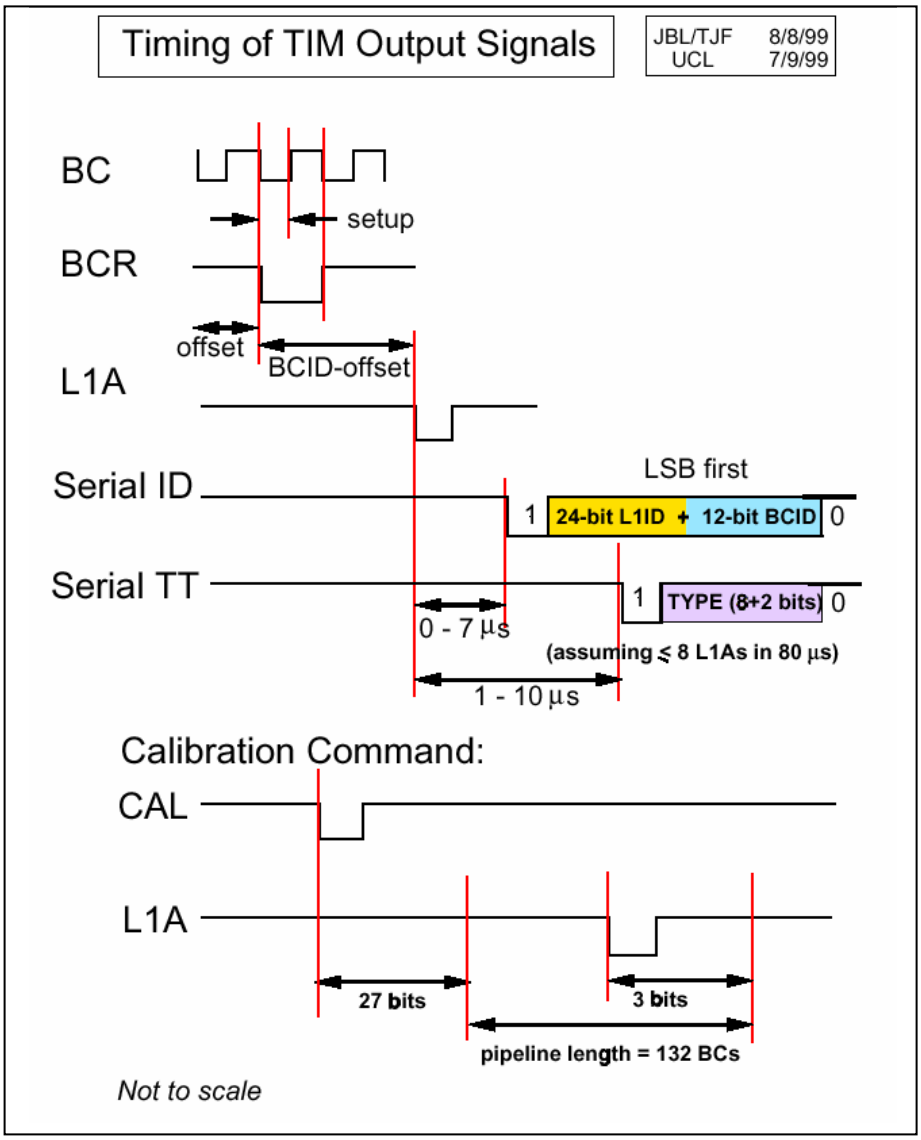


Figure 9: TIM to ROD Timing

The serial parsing of the ID data begins when a wake-up bit is received for each ID packet from the TIM on the respective TTC serial inputs. 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.

When triggers are sent using either MDSP Serial Port input, FE command types are detected in the Calibration Trigger Signal Decoder functional block of the ROD Controller FPGA (RCF), and the Event ID values stored in the FIFO queue are located in the ROD Controller FPGA Memory Map. 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 L1ID 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 BCO counters. When a BCR is detected, the BCO counters are reset to 0. When a L1A is detected, the value of the BCO counter at that time is latched into register and can be used in the Event Header as the BC ID. 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.

The ECR ID value is stored in an internal register in the RCF. When the ROD receives an ECR, the value of the ECR Counter in incremented by 1. Writing a value of 1 to bit 12 ("ECR Counter Reset Bit) in the RCF Spare Control Register resets this counter. The ECR ID value is written to the Event Header FIFO at the same time as the L1 ID and BC ID values for all trigger sources of the ROD.

Functional Procedures:

Reset TTC Trigger Signal Decoder

1. Using the MDSP primitive **RW_REG_FIELD**, set bit 7 and clear bit 8 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Enable TTC Trigger Signal Decoder

1. Using the MDSP primitive **RW_REG_FIELD**, clear bit 7, set bit 8, and clear bit 18 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Enable Calibration Trigger Signal Decoder

1. Using the MDSP primitive **RW_REG_FIELD**, set bit 7, clear bit 8, and set bit 18 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Set ROD Controller Calibration Event Type Registers

1. If the L1 Trigger source is SP0, the value of the Atlas or TIM Event Type is stored in the Calibration Event Type 0 Register (CAL_EVT_TYPE_0: 0x004044D0). Bits[7:0] = Atlas Trigger Type, Bits[9:8] = TIM Trigger Type.
2. If the L1 Trigger source is SP1, the value of the Atlas or TIM Event Type is stored in the Calibration Event Type 1 Register (CAL_EVT_TYPE_1: 0x004044D4). Bits[7:0] = Atlas Trigger Type, Bits[9:8] = TIM Trigger Type.
3. If the L1 Trigger source is SP0, the value of the ROD Specific Event Type is stored in the ROD Default Event Type Register (DFLT_ROD_EVT_TYPE: 0x00404700). Bits[7:0] = ROD Default Event Type.
4. If the L1 Trigger source is SP1, the value of the ROD Specific Event Type is stored in the ROD Corrective Event Type Register (CRTV_ROD_EVT_TYPE: 0x00404740). Bits[7:0] = ROD Corrective Event Type.

Set Static L1ID for Calibration Modes

1. Set bit 22 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410) after the Calibration Trigger Signal Decoder has been enabled.
2. Write the value of the Static L1ID to address location 0x004044E0, bits[23:0] for triggers issued on SP0.
3. Write the value of the Static L1ID to address location 0x004044E4, bits[23:0] for triggers issued on SP1.

Set Static BCID for Calibration Modes

1. Set bit 21 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410) after the Calibration Trigger Signal Decoder has been enabled.
2. Write the value of the Static BCID to address location 0x004044E8, bits[11:0] for triggers issued on SP0.
3. Write the value of the Static BCID to address location 0x004044E8, bits[27:16] for triggers issued on SP1.

Reset ECR ID Counter

1. Set bit 12 of the RCF Spare Control Register (RRIF_CMND_0, 0x00404414). This bit is self-clearing.

4.1.7 Formatter Mode Bits / EFB Header ID and Dynamic Mask

The Formatter Read Out Mode Bits and the EFB Header ID and Dynamic Mask are required to play events through the ROD. They are used to define the output states of the Formatters and EFB parameters that are used to perform link error correction.

Formatter Read Out Mode Bits:

The Formatter Read Out Mode Bits are stored as a Look-Up Table (LUT) 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 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.

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.

4.1.8 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 is the first sub-packet of data in the Header/Dynamic Mask data packet, and consists of 5 16-bit words.

They are shown in the table below.

Event Header ID Information Sub-packet (16 Bit)	
Word 0	L1 ID [15:0]
Word 1	ECR ID [7:0] & L1 ID [23:16]
Word 2	BC ID [11:0] & Dump Data Flag & 000
Word 3	000000 & Tim Event Type [1:0] & Atlas Trigger Type [7:0]
Word 4	00000000 & ROD Event Type [7:0]

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.

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 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.

Dynamic Mask Information Sub-packet (16 Bit)	
Word 5	Dynamic Mask for Links [7: 0]
Word 6	Dynamic Mask for Links [15: 8]
Word 7	Dynamic Mask for Links [23:16]
Word 8	Dynamic Mask for Links [31:24]
Word 9	Dynamic Mask for Links [39:32]
Word 10	Dynamic Mask for Links [47:40]
Word 11	Dynamic Mask for Links [55:48]
Word 12	Dynamic Mask for Links [63:56]
Word 13	Dynamic Mask for Links [71:64]
Word 14	Dynamic Mask for Links [79:72]
Word 15	Dynamic Mask for Links [87:80]
Word 16	Dynamic Mask for Links [95:88]

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.

Functional Procedures:

MDSP primitive **RW_REG_FIELD** should be used for Internal ROD Register access.

Reset Formatter Mode Bits Encoder

1. Set bit 9 and clear bit 10 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Enable Formatter Mode Bits Encoder

1. Clear bit 9 and set bit 10 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Reset EFB Header ID/Dynamic Mask Encoder

1. Set bit 11 and clear bit 12 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Enable EFB Header ID/Dynamic Mask Encoder

1. Clear bits 11, 13, and 14 and set bit 12 of the RCF Main Control Register (RRIF_CMND_1, 0x00404410).

Send Default Mode and Header ID/Dynamic Mask Bits, L1 Triggers from the TIM
Send Corrective Mode and Header ID/Dynamic Mask Bits, L1 Triggers from the TIM

1. Enable the Formatter Mode Bits Encoder, the EFB Header ID Dynamic Mask Encoder, the FE Command Processor, and the TTC Trigger Signal Decoder. Set the FE Command Source bit to route the TIM inputs to the active functional blocks in the ROD Controller FPGA. (RRIF_CMND_1: 0x00404410, Value = 0x1523, Width=13, Offset=0)
2. Write Mode Bit values to the Corrective Formatter Mode Bits Registers. (RMB0_CRTV_LUT(0) to RMB1_CRTV_LUT(7): 0x00404640 to 0x0040467C, Value, Width=12, Offset=0). See appendix A for register and bit mapping details.
3. Write ROD Event Type to the Corrective ROD Event Type Register (CRTV_ROD_EVT_TYPE: 0x00404740, Value, Width=8, Offset=0) if required.
4. Write Dynamic Mask values to the Corrective Dynamic Mask Bits Registers. (DM_CRTV_LUT(0) to DM_CRTV_LUT(12): 0x00404744 to 0x00404770, Value, Width=16, Offset=0)
5. Set the Corrective Mode Bits and Dynamic Mask Ready bit in the RCF Main Control Register (RRIF_CMND_1, 0x00404410, Value = 1, Width=1, Offset=22 {Bit 22}). This bit will clear when an L1 Trigger is detected.
6. Wait for an L1 Trigger from the TIM.

Send Mode and Header ID/Dynamic Mask Bits, L1 Triggers from DSP SP0 or SP1

1. Enable the Formatter Mode Bits Encoder, the EFB Header ID Dynamic Mask Encoder, the FE Command Processor, and set the FE Command Source bit to route the SP0 input to the active functional blocks in the ROD Controller FPGA. (RRIF_CMND_1: 0x00404410, Value = 0x14A1, Width=13, Offset=0)
2. Enable the Calibration Trigger Signal Decoder. (RRIF_CMND_1: 0x00404410, Value = 1, Width=1, Offset=18 {Bit 18})
3. Load the Event ID Registers. For triggers from SP0 write to the Calibration Event Type 0 Register (CAL_EVT_TYPE_0: 0x004044D0, Value, Width=10, Offset=0). For triggers from SP1 write to the Calibration Event Type 1 Register (CAL_EVT_TYPE_0: 0x004044D4, Value, Width=10, Offset=0). Bits[7:0] = Atlas Trigger Type, Bits[9:8] = TIM Trigger Type.
4. Select Static or Dynamic BCID. (RRIF_CMND_1: 0x00404410, Value=Static(1)/Dynamic(0), Width=1, Offset=21 {Bit 21})
5. For triggers from SP0 load the Default Mode Bits Registers. (RMB0_DFLT_LUT(0) to RMB1_DFLT_LUT(7): 0x00404600 to 0x0040463C, Value, Width=12, Offset=0)
6. For triggers from SP1 load the Corrective Mode Bits Registers. (RMB0_CRTV_LUT(0) to RMB1_CRTV_LUT(7): 0x00404640 to 0x0040467C, Value, Width=12, Offset=0).
7. For triggers from SP0 load the Default ROD Event Type. (DFLT_ROD_EVT_TYPE: 0x00404700, Value, Width=8, Offset=0)
8. For triggers from SP1 load the Corrective ROD Event Type. (CRTV_ROD_EVT_TYPE: 0x00404740, Value, Width=8, Offset=0)
9. For triggers from SP0 load the Default Dynamic Mask Registers. (DM_DFLT_LUT(0) to DM_DFLT_LUT(12): 0x00404704 to 0x00404730, Value, Width=16, Offset=0)
10. For triggers from SP1 load the Corrective Dynamic Mask Registers. (DM_CRTV_LUT(0) to DM_CRTV_LUT(12): 0x00404744 to 0x00404770, Value, Width=16, Offset=0)
11. Send an L1 Trigger on SP0 or SP1.

4.1.9 ROD Internal Test Bench

The ROD Controller FPGA contains an internal Test Bench that allows different degrees of testing to be performed on the ROD without the need for the TIM or the BOC Electronic Boards. The internal registers used by the Test Bench functional block are the Input Memory A/B Count Register (INP_MEM_CTRL: 0x0040460), the Debug Memory A/B Count Register (DBG_MEM_CTRL: 0x0040464), the Configuration Read Back Count Register (CFG_READBACK_CNT: 0x0040468), the Debug Mode Configuration Register (IDE_MEM_CTRL: 0x0040470), and the Debug Mode Status Register (IDE_MEM_STAT: 0x0040474). Each count register contains 2 16-bit counters that are used to control different functions in each Diagnostic Mode. The Debug Mode Configuration Register contains control bits for loading and enabling the counters, and the bit field that selects the test type. The Diagnostic Mode is selected by setting a value to the Test Bench Mode Select Bits in the Debug Mode Configuration Register (IDE_MEM_CTRL: 0x0040470, Value=Test Modes, Width=6, Offset=12, Bits{17:12}). The Debug Mode Status Register is used to monitor the progress of the Test in progress. This register stores the done flags of the counters, the done flags of the test in progress, and Occupancy Status Flags for the Input Memory FIFOs and the Debug Memory FIFOs.

This block controls an internal FIFO that is used to issue TIM commands. The Internal TIM is loaded with data and read out using the MDSP primitive **RW_FIFO**, and can be played and re-transmitted into the ROD Controller FPGA to emulate the TIM TTC Bus inputs using the control signals from the Test Bench.

Setting bit 16 and clearing bit 15 in the ROD Controller FPGA (RCF) Main Control Register enables the Test Bench Interface.

Test Modes:

Mode 0x20: Play Simulated FE Data from the Input Memory FIFOs through the Formatters, EFB, and Router, to the S-Link and the Slave DSPs.

This mode is used as a diagnostic tool for the complete data path of the ROD. It tests the data flow from the input of the Formatters to the output of the Router, the ID and Dynamic Mask functional blocks of the ROD Controller FPGA, the Router trapping algorithms, the 80MHz DMA from the Router to the Slave DSPs, and many other functions of the ROD. The setup that will follow is a basic example that can be modified to suit other test requirements. This mode uses use the L1 trigger to start playing the INMEM FIFO to approximate when real data would be transmitted to the ROD.

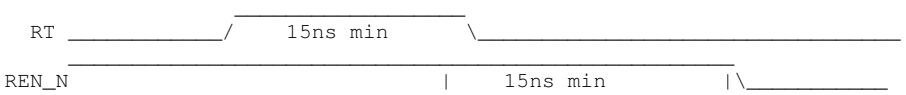
Function	Register Address	Primitive	REG ID	Value	Width	Off set
The ROD Setup Commands listed in the first portion of this Test Mode only need to be executed once.						
Load the Slave Code to all DSPs that will be used.		RW_SLAVE_ MEMORY				
Start the Slave Executable on all DSPs that will be used.		START_SLAVE_ EXECUTING				
Enable the FE Command Outputs w/TIM as the source, FE OCC Counters, FE Command Pulse Counter, TTC Trigger Signal Decoder, Formatter Mode Bits Encoder, EFB Header ID Dynamic Mask Encoder and the Test Bench I/O.	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1152B	17	0
Set value for Test Mode	0x00404470	RW_REG_FIELD	IDE_MEM_CTRL	0x20	6	12
Set the value for the length of time to play the Internal TIM and the INMEM FIFOs.	Input Memory A/B Count 0x00404460	RW_REG_FIELD	INP_MEM_CTRL	Play (15:0)	16	0
Load the Default Mode Bits	0x00404600 to 0x0040463C	RW_REG_FIELD	RMB0_DFLT_LUT(0) RMB1_DFLT_LUT(7)	See App A	12	0
Load the Default Dynamic Mask	0x00404704 to 0x00404730	RW_REG_FIELD	DM_DFLT_LUT(0) DM_DFLT_LUT(11)	See App A	16	0
Set all Router Data Trapping Parameters		EVENT_TRAP_ SETUP				
Send Start Event Trapping to the Slaves		SEND_SLAVE_ LIST				
Start the Event Trap Primitive		START_SLAVE_ LIST				
The commands that control the data flow through the ROD can be repeated as many times as required.						
Set value for Input Data Link Mux: RCF Debug FIFO Control	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	3	25
Mask all Formatter Link Inputs. Disables all inputs and flushes all FIFOs		RW_REG_FIELD		0x1	4	0
Reset the Input Memory, TIM, Mode Bits, and EFB ID FIFOs	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0xFF	8	0
Enable the Input Memory, TIM, Mode Bits, and EFB ID FIFOs	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0x30	8	0
Load INMEMs and TIM FIFO with Simulated Data		RW_FIFO				
Set value for Input Data Link Mux: Test Bench I/O control	0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x2	3	25
Enable the Formatter Link Inputs of Interest and parse in Condensed Data Mode.		RW_REG_FIELD		0x2	4	0
Set Test Bench Run Bit to start trapping data.	0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	1	17
Poll the Test Operation Done Bits for end of test.	0x00404474	POLL_REG_FIELD	IDE_MEM_STAT	0x3	2	6
Clear Test Bench Run Bit to set Test Mode State Machine to idle.	0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x0	1	17

Read Router to IDRAM DMA Buffer in the Slave DSP Memory and save to a file.		RW_SLAVE_ MEMORY	
The data files generated by the last command can be compared to the output files generated by the simulation to determine if the ROD is operating correctly			

Mode 0x24: Play and re-transmit simulated FE data from the Input Memory FIFOs through the Formatters, EFB, and Router, to the S-Link and the Slave DSPs.

This mode is used as a diagnostic tool for the complete data path of the ROD for the 100kHz Trigger Rate testing. The mode is very similar in set up to Test Mode 0x20. It tests the data flow from the input of the Formatters to the output of the Router, the ID and Dynamic Mask functional blocks of the ROD Controller FPGA, the Router trapping algorithms, the 80MHz DMA from the Router to the Slave DSPs, and many other functions of the ROD. A re-transmit counter allows 65536 times the Events per Simulation File to be sent through the ROD. The Word Count counter is used to set the period of the test. The time period can be adjusted in 25ns increments, and controls the amount of time that the TIM and INMEM FIFOs are playing data to the ROD. The setup that will follow is a basic example that can be modified to suit other test requirements. This mode uses use the L1 trigger to start playing the INMEM FIFOs to approximate when real data would be transmitted to the ROD.

Re-transmit Timing



Function	Register Address	Primitive	REG ID	Value	Width	Offset
The ROD Setup Commands listed in the first portion of this Test Mode only need to be executed once.						
Load and start all SDSPs		xxxx				
Enable the FE Command Outputs w/TIM as the source, FE OCC Counters, FE Command Pulse Counter, TTC Trigger Signal Decoder, Formatter Mode Bits Encoder, EFB Header ID Dynamic Mask Encoder and the Test Bench I/O.	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1152B	17	0
Set value for Test Mode	0x00404470	RW_REG_FIELD	IDE_MEM_CTRL	0x24	6	12
Set the value for the length of time to play the Internal TIM and the INMEM FIFOs (0xFFF = 102.5us), and the number of desired Re-Tx.	Input Memory A/B Count 0x00404460	RW_REG_FIELD	INP_MEM_CTRL	Re-Tx (31:16) Play (15:0)	32	0
Load the Default Mode Bits	0x00404600 to 0x0040463C	RW_REG_FIELD	RMB0_DFLT_LUT(0) RMB1_DFLT_LUT(7)	See App A	12	0
Load the Default Dynamic Mask	0x00404704 to 0x00404730	RW_REG_FIELD	DM_DFLT_LUT(0) DM_DFLT_LUT(11)	See App A	16	0
Set all Router Data Trapping Parameters and start Event Trapping on SDSP						
The commands that control the data flow through the ROD can be repeated as many times as required.						
Set value for Input Data Link Mux: RCF Debug FIFO Control	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	3	25
Mask all Formatter Link Inputs. Disables all inputs and flushes all FIFOs		RW_REG_FIELD		0x0	12	0
Reset the Input Memory, TIM, Mode Bits, and EFB ID FIFOs	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0xFF	8	0
Enable the Input Memory, TIM, Mode Bits, and EFB ID FIFOs	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0x30	8	0
Load INMEM A, INMEM B, and the internal TIM FIFO with Simulated Data		RW_FIFO	FifoID&Bank (INPUT_MEM-A) FifoID&Bank (INPUT_MEM-B) FifoID&Bank (TIM)			
Set value for Input Data Link Mux: Test Bench I/O control	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x2	3	25
Enable the Formatter Link Inputs of Interest and parse in Condensed Data Mode.		RW_REG_FIELD			12	0
Set Test Bench Run Bit to start trapping data.	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	1	17
Poll the Test Operation Done Bits for end of test.	Debug Mode Stat 0x00404474	POLL_REG_FIELD	IDE_MEM_STAT	0x3	2	6
Clear Test Bench Run Bit to set Test Mode State Machine to idle.	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x0	1	17
Read Router to IDRAM DMA Buffer in the Slave DSP Memory and save to a file.		RW_SLAVE_MEMORY				
The data files generated by the last command can be compared to the output files generated by the simulation to determine if the ROD is operating correctly						

4.1.10 Readout Link (RoL) Test Block

The Readout Link test block is used to send data up the S-Link and to verify the readout bandwidth of the ROD. It is also used to verify that the interface from the EFB 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 (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 65536 or set to transmit data forever.

RoL Function Modes:

Send 1 32 data word fragment, ROD triggered
- OP0 → Reset the ROD
- OP1 → Set IDE_MEM_CTRL register (0x00404470)::Bit 21 = 0x1
- OP2 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x1 (Start)
- OP3 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x0 (Reset)

Send 1 n data word fragment, ROD triggered
- OP0 → Reset the ROD
- OP1 → Set IDE_MEM_CTRL register (0x00404470)::Bit 21 = 0x1, Bits[29:24] = nData
- OP2 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x1 (Start)
- OP3 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x0 (Reset)

Repeat send 32 data word fragments, ROD triggered
- OP0 → Reset the ROD
- OP1 → Set IDE_MEM_CTRL register (0x00404470)::Bit 21 = 0x0
- OP2 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x1 (Start)
- OP3 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x0 (Stop)

Repeat send n data word fragments, ROD triggered
- OP0 → Reset the ROD
- OP1 → Set IDE_MEM_CTRL register (0x00404470)::Bit 21 = 0x0, Bits[29:24] = nData
- OP2 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x1 (Start)
- OP3 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x0 (Stop)

Send a fixed number of data word fragments, ROD triggered
- OP0 → Reset the ROD
- OP1 → Set DBG_MEM_CTRL register (0x00404464)::Bits[23: 0] = nEvents
- OP2 → Set IDE_MEM_CTRL register (0x00404470)::Bits[23:21] = 0x4, Bits[29:24] = nData
- OP3 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x1 (Start)
- OP4 → Poll IDE_MEM_STAT register (0x00404474)::Bit15 (1 = ROL Test Triggers Active)
- OP5 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x0 (Reset)

Send 1 32 data word fragment, TIM triggered
- OP0 → Reset the ROD
- OP1 → Set IDE_MEM_CTRL register (0x00404470)::Bits[22:21] = 0x3
- OP2 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x1 (Start)
- OP3 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x0 (Stop)

Send 1 n data word fragment, TIM triggered
- OP0 → Reset the ROD
- OP1 → Set IDE_MEM_CTRL register (0x00404470)::Bits[22:21] = 0x3, Bits[29:24] = 0xn
- OP2 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x1 (Start)
- OP3 → Set IDE_MEM_CTRL register (0x00404470)::Bit 20 = 0x0 (Stop)

RoL Test Block Data Format:

ROL Test Block Format – The ROL Test Block in the Silicon ROD is different from the block specified in Atlas Doc No. ATC-TD-ES-0002. The differences are shown below.	**Data**	**Offset**
Beginning of Fragment Control Word	0xB0F00000	
ROD Header Marker	0xEE1234EE	0
Header Length	0x00000009	1
Format Version	0x03000000	2
Source ID	0x00110000	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
	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	0x00000020	44
Status Block Position	0x00000001	45
End of Fragment Control Word	0xE0F00000	

4.1.12 Internal Scan Engine Block

```
 Input Definitions
   NEvent     : Number of command sequences to issue in this mask stage
   TrigDelay0 : Delay between CAL and L1 commands in 0.1us increment
   TrigDelay1 : Delay between first L1 and second L1 cmd in 25ns increment
   Interval   : Delay between successive cmd sequences in 25ns increment
   TrigMode   : Trigger Mode Select Register/Active groups
                Bits (3:0) Trigger Sequence/Trigger Algorithm
   VAL  SCAN DESCRIPTION
   0x1   Self Trigger up to 4 groups
   0x2   Group synchronous scan CAL/TD0/L1A
   0x3   Fixed interval scan CAL/TD0/L1A
   0x4   Fully synchronous scan CAL/TD0/L1A
```

Modes to control scanning:

TrigSequence = decide what sequence of commands to issue for one event:
* Self-trigger => Issue single TRIG command only
* Normal   => Issue CAL/TrigDelay1/TRIG sequence
* Dual   => Issue CAL/TrigDelay1/TRIG/TrigDelay2/TRIG seq.

TrigAlgorithm = decide between different algorithms for issuing sequences to different module groups, and for issuing successive sequences.

* 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 microsecs after the previous group.

* Next trigger mode => Issue the next command sequence based on one of three different algorithms:

Fixed interval ➔ Issue next sequence after a fixed delay, independent of the status of returned data

Group synchronous ➔ Issue the next sequence for a given module group as soon as all of the FE occupancy counters for the modules in that group go to zero (meaning all expected trailers have been seen).

Fully synchronous ➔ Issue the next sequence only when all modules for all active groups have their FE occupancy counters returned to zero.

Description	REG ID	Address	Access	Width
ROD Controller: FE Occupancy Counter L1A Value (31:0) FE_OCC_CNTR_NUM_ACC(0)	**252**	**004044B0**	**RW**	**32**
ROD Controller: FE Occupancy Counter L1A Value (31:0) FE_OCC_CNTR_NUM_ACC(1)	**253**	**004044B4**	**RW**	**32**
ROD Controller: FE Occupancy Counter L1A Value (31:0) FE_OCC_CNTR_NUM_ACC(2)	**254**	**004044B8**	**RW**	**32**
ROD Controller: FE Occupancy Counter L1A Value (31:0) FE_OCC_CNTR_NUM_ACC(3)	**255**	**004044BC**	**RW**	**8**
This register stores the number of L1 Accepts to expect from each FE module and is used in the Pixel FE Occupancy Counters.			**Bit Value**	

4.8: ROD Controller Diagnostic Calibration Event ID Registers

Description	REG ID	Address	Access	Width
ROD Controller: Internal Scan NEvent Register: (23:0)	**256**	**004044C0**	**RW**	**15**
Number of events to issue to Pixel FE Modules. NEvent : Number of command sequences to issue in this mask stage			**Bit Value**	

Description	REG ID	Address	Access	Width
ROD Controller: Internal Scan TrigDelay Register: (31:0)	**257**	**004044C4**	**RW**	**32**
Trigger Delay for Internal scans. TrigDelay0 : Delay between CAL and L1 commands in 100ns inc. for even groups TrigDelay1 : Delay between CAL and L1 commands in 100ns inc. for odd groups			**Bit Value**	

Description	REG ID	Address	Access	Width
ROD Controller: Internal Scan Interval Register: (31:0)	**258**	**004044C8**	**RW**	**32**
Event Interval for Internal scans.			**Bit Value**	
Interval : Delay between successive cmd sequences in 25ns increment			**1**	**0**

Description	REG ID	Address	Access	Width
ROD Controller: Internal Scan Trigger Mode: (31:0)		**004044CC**	**RW**	**32**
Trigger Mode for Internal scans. VALUE SCAN DESCRIPTION 0x1 Self Trigger up to 8 groups L1A only 0x2 Group synchronous scan CAL/TD0/L1A 0x3 Fixed interval scan CAL/TD0/L1A 0x4 Fully synchronous scan CAL/TD0/L1A			**Bit Value** **1**	**0**

Description	REG ID	Address	Access	Width
ROD Controller: Internal Scan Status		**004044D8**	**R**	**1**
			Bit Value	
Bit [0]: Scan Done			Done	Idle

Description	REG ID	Address	Access	Width
FE Occupancy Counters: Group to Link Map		**00404530**	**RW**	**32**
FE Occupancy Counters: Group to Link Map		**00404534**	**RW**	**32**
FE Occupancy Counters: Group to Link Map		**00404538**	**RW**	**32**
FE Occupancy Counters: Group to Link Map		**0040453C**	**RW**	**32**
(12*n + 0, 1, 2, 3) ((48-12*n) + 0, 1, 2, 3))			**Bit Value**	

Description	REG ID	Address	Access	Width
FE Occupancy Counters: Internal Scan Group Enable		**00404540**	**RW**	**8**
			Bit Value	

4.2 ROD Program Reset Manager FPGA (PRM) Functions

4.2.1 FPGA Configuration and Resets
4.2.2 Board and DSP Resets
4.2.3 FPGA Flash RD/WR access
4.2.4 ROD Busy Module: Busy Duration Buffers

A requirement of the ROD-Busy module is to store a history of the integrated Busy duration for each ROD. This function is implemented on the ROD in the PRM FGPA.

Busy Duration Buffers Minimum Requirements:

- 512x16 FIFO
- Increment a counter for every clock that ROD Busy is asserted
- Write value every 6.5 ms
- FIFO's are full after 3.3 sec
- FIFO's are reset by global command
- Each of the 16 FIFO's are readable from VME

Busy Duration Buffers Specification:

- 1024x32 Dual Port RAM
- Full Capture Time: 13 seconds
- Memory Readout time: ~ 250us
- Address Map Location: 0xC01000 to 0xC01FFC
- Memory Organization: 2 16-bit bins per address

```
            Bit31---------Bit16 | Bit15----------Bit0
0xC01000 == ROD Busy Count 1    | ROD Busy Count 0
0xC01004 == ROD Busy Count 3    | ROD Busy Count 2
...
0xC01FFC == ROD Busy Count 2047 | ROD Busy Count 2046
```

Algorithm to enable histogram:

1. Set trigger type bit at location 0xC0001C, bit 1. Value = 0, start counters when Histogram control enabled. Value = 1, start counters when next L1 trigger is detected.
2. Set Histogram Control Enable Bit to value = 1. Location 0xC0001C, bit 0.
3. Monitor Rod Busy Histogram address status if required at location 0xC0005C. Bits [9:0] indicate the current data write location in the DPRAM, and bits [15:10] indicate the number of times the address counter has wrapped around 0.

Functional Algorithm:

1. Enable the histogram engine per the enable algorithm.
2. If “next trigger mode” is enabled, following the next detected trigger (to ensure all RODs are synchronized), the bin counter starts and counts to 6.5ms
3. If ROD Busy is asserted, a counter is incremented
4. If ROD Busy is not asserted, the RB counter does not increment
5. When the bin counter reaches 6.5ms, the value in the ROD Busy counter is latched to a holding register and the RB counter is reset to 0.
6. The bin counter starts again and counts to 6.5ms.
7. When the bin counter reaches 6.5ms for the second time, the value in the ROD Busy counter plus the holding register is written the dual port RAM, the RB counter is reset to 0, and the bin counter is reset to 0
8. The DPRAM address is incremented by 1 count. If the memory is full, the address pointer wraps to address 0 and the process repeats.
9. The DPRAM block can be read by the crate-controller at any time provided that the number of readout locations is less than the current write address.

4.2 ROD Configurations

Functional Procedure: Set ROD for Normal Data Taking

The following is a basic setup that can be used to set up the ROD for Normal Data Taking. The Slave DSPs do not need to be running at this time as data will be transmitted on the S-Link, but a simple monitoring setup allows data checking when the S-Link is not available.

4.3 Functional Procedure: Trap Configuration Data in the Input Memory FIFOs

The ROD supports a mode that allows the trapping and read back of configuration data in the INMEM FIFO. The largest configuration stream that can be trapped is 32K bits long for any input link. The ROD Controller FPGA enables the trapping of all input links simultaneously. The input data is trapped in the Input Memory (INMEM) FIFO as it arrives from the BOC and can be read and verified by the MDSP SDRAM or the VME host. The "Control Signal Mux" bits (RRIF_CMND_1, Bits[27:25]) must be set to the value 0x5h to enable the trapping function. After data has been trapped in the INMEM FIFO, the value of the "Control Signal Mux" bits must be set to 0x1h to allow the MDSP to read the Configuration data from the FIFO. The function requires 2 counters. The first, register location 0x00404468 Bits [31:16], sets a time delay to align the first returning bit with the first location in the Input Memory FIFO. The second counter, location 0x00404468 Bits [15:0], controls the number of words to trap. One method of transmitting a serial configuration data stream to the FE Module and trapping the return data is to use the MDSP primitive **SEND STREAM** with the "Data Capture Bit" set.

Function	Register Address	Primitive	REG ID	Value	Width	Offset
Reset the INMEM FIFOs.	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0x1	1	3
Enable the INMEM FIFOs.	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0x0	1	3
Disable the Formatter Link Inputs.	Formatter Channel Enable	RW_REG_FIELD	FMT_LNK_EN(0) FMT_LNK_EN(7)	0x0	12	0
Enable the FE Command Outputs with the SP0 as the source for Mask 0.	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	2	0
Load the FE Command Mask0 LO	FE Command Link Mask0 LO 0x00404430	RW_REG_FIELD	FE_CMND_MASK_0_LO	Value	32	0
Load the FE Command Mask0 HI	FE Command Link Mask0 HI 0x00404434	RW_REG_FIELD	FE_CMND_MASK_0_HI	Value	16	0
Load the FE Command Mask1 LO	FE Command Link Mask1 LO 0x00404438	RW_REG_FIELD	FE_CMND_MASK_1_LO	Value	32	0
Load the FE Command Mask1 HI	FE Command Link Mask1 HI 0x0040443C	RW_REG_FIELD	FE_CMND_MASK_0_HI	Value	16	0
Set Configuration/Cal Mask Load Enable Bit.	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	1	20
Set New Mask Ready Bit to load FE Command Mask	RCF Main Control	RW_REG_FIELD	RRIF_CMND_1	0x1	1	2
Set the trap time and delay time values for the INMEM FIFO. (0x1 = 25ns)	Configuration ReadBack Count 0x00404468	RW_REG_FIELD	CFG_READBACK_CNT	Delay (31:16) Play (15:0)	32	0
Set value for Control Mux: Configuration Read Back Mode	0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x5	3	25
Send Configuration Data, set Configuration Read Back Bit						
Poll the Configuration Read Back Done Bit	RCF Main Status 0x00404420	POLL_REG_FIELD	RRIF_STATUS_1	0x1	1	3
Set value for Control Mux: Use Debug FIFO Control	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	3	25
Read INMEM A FIFO and save data to a file		RW_FIFO	Read/Write Flag (Set to Read) FifoID&Bank (INPUT_MEM-A) Output File ()			
Read INMEM B FIFO and save data to a file		RW_FIFO	Read/Write Flag (Set to Read) FifoID&Bank (INPUT_MEM-B) Output File ()			

4.4 Functional Procedure: Play Link Data to the Formatters, and Trap the same in the Input FIFOs

The ROD supports a mode that allows the trapping of Link Data in the Input Memory FIFOs while routing through to the Formatters. The process includes 2 counters, the first, register location 0x00404468 Bits [31:16], sets a time delay to align the first returning bit with the first location in the Input Memory FIFO. The second counter, location 0x00404468 Bits [15:0], controls the number of words to trap. The MDSP primitive **SEND STREAM** sends the triggers to the ROD and the Modules. The L1 Trigger starts the data trap state machine.

Function	Register Address	Primitive	REG ID	Value	Width	Off set
Reset the INMEM FIFO.	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0x1	1	3
Enable the INMEM FIFO.	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0x0	1	3
Disable the Formatter Link Inputs.		RW_REG_FIELD		0x0		0
Enable the FE Command Outputs with the SP0 as the source for Mask 0.	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	2	0
Load the FE Command Mask0 LO	FE Command Link Mask0 LO 0x00404430	RW_REG_FIELD	FE_CMND_MASK_ 0_LO	Value	32	0
Load the FE Command Mask0 HI	FE Command Link Mask0 HI 0x00404434	RW_REG_FIELD	FE_CMND_MASK_ 0_HI	Value	16	0
Load the FE Command Mask1 LO	FE Command Link Mask1 LO 0x00404438	RW_REG_FIELD	FE_CMND_MASK_ 1_LO	Value	32	0
Load the FE Command Mask1 HI	FE Command Link Mask1 HI 0x0040443C	RW_REG_FIELD	FE_CMND_MASK_ 0_HI	Value	16	0
Set Config/Cal Mask Load Enable Bit.	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	1	20
Set New Mask Ready Bit to load FE Command Mask	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	1	2
Set the value for the length of time to trap data in the INMEM FIFOs (0x1 = 25ns), and the delay from L1A to data out from FIFOs.	Configuration Read Back Count 0x00404468	RW_REG_FIELD	CFG_READBACK_CNT	Delay (31:16) Play (15:0)	32	0
Set value for Input Data Link Mux: Use Trap Input Data Mode	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x7	3	25
Enable required Formatter Link Inputs.		RW_REG_FIELD		0x0		0
Send an L1 Trigger		BUILD_STREAM SEND STREAM	**DO NOT set the capture Bit**			
Poll the “Configuration Read Back/Link Data Trap Done” Bit	RCF Main Status 0x00404420	POLL_REG_FIELD	RRIF_STATUS_1	0x1	1	3
Set value for Input Data Link Mux: Use Diagnostic Memory Control Mode	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	3	25
Read INMEM A FIFO and save data to a file		RW_FIFO	Read/Write Flag (Set to Read) FifoID&Bank (INPUT_MEM-A) Output File ()			
Read INMEM B FIFO and save data to a file		RW_FIFO	Read/Write Flag (Set to Read) FifoID&Bank (INPUT_MEM-B) Output File ()			

4.5 Functional Procedure: Calibration Test Using Input Memory FIFOs as a data source, and SP0 and SP1 as the Trigger Sources

Function	Register Address	Primitive	REG ID	Value	Width	Off set
Load the Slave Code to all DSPs that will be used.		RW_SLAVE_ MEMORY				
Start the Slave Executable on all DSPs that will be used.		START_SLAVE_ EXECUTING				
Enable the FE Command Outputs w/SP0 as the Mask0 source, FE Occ. Counters, FE Cmd Pulse Counter, CAL Trigger Signal Decoder, Formatter Mode Bits Encoder, and the EFB HeaderID Dynamic Mask Encoder.	RCF Main Control 0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x494A9	19	0
Select Static or Dynamic BCID 1=Static 0=Dynamic	0x00404410	RW_REG_FIELD	RRIF_CMND_1	Value	1	21
Load Static BCID, if required	0x004044E8	RW_REG_FIELD		SP1 (27:16) SP0 (11:0)	32	0
Load Event Type for SP0 Load Event Type for SP1	0x004044D0 0x004044D4	RW_REG_FIELD	CAL_EVT_TYPE_0 CAL_EVT_TYPE_1	TIM (9:8) ATLAS (7:0)	10	0
Load the Default Mode Bits Used when SP0 L1A detected	0x00404600 to 0x0040463C	RW_REG_FIELD	RMB0_DFLT_LUT(0) RMB1_DFLT_LUT(7)	See AppA	12	0
Load the Default Dynamic Mask Used when SP0 L1A detected	0x00404704 to 0x00404730	RW_REG_FIELD	DM_DFLT_LUT 0 DM_DFLT_LUT 11	See AppA	16	0
Load Corrective Mode Bits Used when SP1 L1A detected	0x00404640 to 0x0040467C	RW_REG_FIELD	RMB0_CRTV_LUT(0) RMB1_CRTV_LUT(7)	See AppA	12	0
Load Corrective Dynamic Mask Used when SP1 L1A detected	0x00404744 to 0x00404770	RW_REG_FIELD	DM_CRTV_LUT 0 DM_CRTV_LUT 11	See AppA	16	0
Set the value for the length of time to play the INMEM FIFOs (0xFFF = 102.5us), and the delay from L1A to data out from FIFOs.	0x00404468	RW_REG_FIELD	CFG_READBACK_CNT	Delay (31:16) Play (15:0)	32	0
Set value for Input Data Link Mux: Test Bench I/O control	0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x1	3	25
Mask all Formatter Link Inputs. Disables all inputs and flushes all FIFOs		RW_REG_FIELD				0
Reset the Input Memory, TIM, Mode Bits, and EFB ID FIFOs	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0xFF	8	0
Enable the Input Memory, TIM, Mode Bits, and EFB ID FIFOs	RCF Spare Control	RW_REG_FIELD	RRIF_CMND_0	0x30	8	0
Load INMEM A, and INMEM B with simulated FE Data		RW_FIFO				
Set value for Input Data Link Mux: Calibration Test Mode.	0x00404410	RW_REG_FIELD	RRIF_CMND_1	0x6	3	25
Set all Router Data Trapping Parameters		EVENT_TRAP_ SETUP	Permit Back Pressure (Calibration Only)			
Send Start Event Trapping to the Slaves		SEND_SLAVE_ LIST	Slave Number (0,1,2,3) Primitive (START_EVENT_TRAPPING)			
Start the Event Trap Primitive		START_SLAVE_ LIST				
Enable the Formatter Link Inputs of Interest		RW_REG_FIELD				0
Poll the Calibration Test Ready Bit.	Main Status 0x00404420	POLL_REG_FIELD	RRIF_STATUS_1	0x1	1	4
Build the Serial Data Stream		BUILD_STREAM				
Any number of L1 Triggers can be sent at this point. The 1 restriction is that only one trigger can be active at a time in this mode. The Calibration Test Ready Bit indicates the status of the ROD. If the bit value is 1, a new trigger can be issued, if the bit value is 0, the Input Memory FIFO are still playing simulation data to the Formatters, and a re-transmit at this point would result in data corruption.						
Transmit a Trigger						
Poll the Calibration Test Ready Bit.	Main Status 0x00404420	POLL_REG_FIELD	RRIF_STATUS_1	0x1	1	4