Endcap Muon Trigger System: Read-out Driver Design Overview

Lorne Levinson, Weizmann Institute of Science
ATLAS TGC group

2 January 2009

Revisions:	(changes from previous major version are marked by revision bars)
25 Feb 2004	First draft for Final Design Review
13 April 2005	Changes accumulated since Final Design Review
2 Jan 2009	Updated to reflect the final ROD board (all revision bars removed)

Please check the revision date on your copy. The latest version can be found at:
http://cern.ch/atlas-tgc/doc/TGCROD_design.pdf

1 Introduction and context

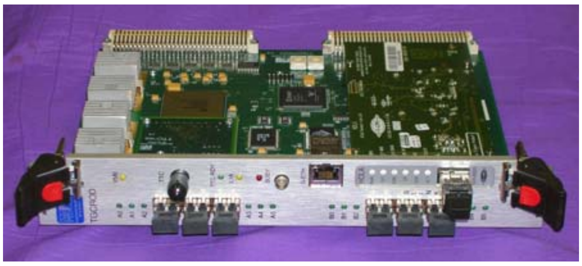

Figure 1 The TGC ROD VME module

This document describes the hardware and FPGA firmware of the Read Out Driver (ROD) for the ATLAS muon endcap trigger chambers, known as TGC or Thin Gap Chambers. This design fulfils all the requirements and specifications needed for ATLAS. The ROD must be able to handle up to 100kHz of Level-1 event accept triggers. Table 1 provides a summary of the major parameters of the TGC ROD context.

The ROD is a double-width 6U VME module that builds an event from up to 10 Front End links from the Star Switches (SSW) located around the periphery of the TGC wheels[1], checks the integrity of the data, decodes and re-formats the data, sends the data to the ATLAS Read Out System, and provides sampled data for online monitoring, to both the ROD crate processor via the VMEbus and to PC computers via the ROD's gigabit ethernet output.

Each TGC Read-out Driver reads out hit and coincidence data from a tower of the seven detector layers in one 1/12th TGC sector[2], and that sector's Sector Logic (the last stage of a sector's trigger electronics). The Inner Wheel of the TGC is physically constructed in octants. Consequently an Inner quadrant is read out by the ROD reading the middle 1/12th of each quadrant. There are four ROD VME crates; each contains one ROD crate processor and six RODs.

The requirements and specifications for this ROD can be found in the Preliminary Design Review document[ref. 1]. An overview of the TGC read-out can be found in the ATLAS Level-1 Trigger Technical Design Report [ref. 2] and [ref. 3]. The web page for the ROD can be found in [ref. 4]. Front End data rates and a simulation of the Front End data buffering can be found in [ref. 5]. ROD Prototype-0 has been described in [ref. 6]. A report on its performance can be found in [ref. 7].

Table 1 TGC ROD context

Number of RODs	24, one per 1/12th sector
Number of ROD crates	4, one per half-endcap
Number of TTC partitions	2, one per end-cap
Number of input Front End links, one from each Star Switch	28 per quadrant of 3 RODs, 224 total
Raw bandwidth per Front End Link	16 bits × 40MHz = 80MB/sec
Average fragment size (@1×estimated USA15 bkgnd)	22 bytes (incl 8 header bytes)
ROD input data rate (@5×estimated USA15 bkgnd	100MB/sec
Number of Read Out links, ROLs (to the ROS)	24
Output fragment size (@ 5×estimated USA15 bkgnd	~200 bytes per ROD
ROD output data rate (@ 5×estimate. USA15 bkgnd	~20MB/sec
S-Link maximum bandwidth (HOLA implementation)	160MB/sec

[1] The area of the detector read out by a Star Switch is also referred to as a Local Data Block (LDB).

[2] Original plans were for one ROD per octant tower, to match the MDT and RPC. Previous documents refer to this octant division.

2 Required functions

For each Level-1 Accept, an event fragment is read from each of the Front End links, checked, decoded, merged into an event along with information from the TTC system, formatted according to the ATLAS standard format and placed in the output buffer for transmission to the Read-out Buffer, ROB, of the ATLAS central Read-out System, ROS. In addition, all or a sample of events, can be sent for monitoring to the ROD Crate DAQ system running on the ROD Crate Processor (RCP) and / or to a monitoring computer via the gigabit ethernet output. The Level-1 Accept rate can be up to 100 kHz. The Front End event data may include data from just one, or up to three (0, ±1), bunch crossings.

In order to accomplish this, the ROD must perform the following major functions:

- Receive TTC information and queue expected event IDs and their BCID and trigger type.
- Collect event fragments from all the Front End links corresponding to each of the Event IDs.
- Detect and recover from input link errors and data errors.
- Assert RODBUSY to the ATLAS CTP module (Central Trigger Processor) when input FIFOs become almost full, but as infrequently as possible.
- Verify the input data structure; extract and decode the hit, coincidence and sector logic data.
- Format events into ATLAS standard ROB format.
- Optionally include the input raw data in the output format.
- Provide sampled events for monitoring the data to the Rod Crate Processor and / or to a monitoring computer via the gigabit ethernet output.
- Send the data to the ROS and / or Rod Crate Processor.
- Respond to S-Link flow control signals from the ROS.
- Inform the Rod Crate Processor when links fail, erroneous data is received or other exceptional conditions arise

The input transfer protocol and data format are detailed in Section 5. The output format to the ROB, described in [ref. 8], conforms to the ATLAS standards described in [ref. 9].

3 Design environment

The following design tools have been used to design the ROD:

Schematic capture	Mentor Graphics
Board layout	Mentor Graphics Expedition
FPGA design: top level and FE link handler	Mentor Graphics HDL (mixed graphics and VHDL)
FPGA design: main logic	Handel-C
FPGA functional simulation	Handel-C
FPGA synthesis	Mentor Graphics Precision
FPGA place & route	Xilinx ISE

4 Architecture

The ROD's functions listed in Section 2 are implemented by an asynchronous pipeline architecture of three main data-driven stages:

Input link handling and buffering: link error detection, event fragment framing, verifying fragment checksum, computing fragment word count, fragment buffering in FIFO

Event fragment processing: format verification, data error reporting, hit, coincidence, and sector logic decoding and re-formatting, hit and coincidence counting, associating hits with coincidences[1], hit – coincidence consistency checking[1]

Output event formatting: collection of TTC information, header, input fragment, hit and coincidence data, formatting into ATLAS standard ROB input event format, event sampling for monitoring.

"Asynchronous" here means that the data size and so execution clocks needed per fragment vary from fragment to fragment and from stage to stage. "Pipes", described below, are therefore needed to hold data between stages to compensate for fluctuations in execution time and fragment size. The input link handling stage is replicated, once per front end link. There are four identical fragment processors working in parallel, each with a dedicated transfer path from a group of three Front End channels, in order to attain the needed event throughput. A scheduler marshals the transfer of fragments to their fragment processor when both are ready. There is only one copy of the output formatter. As shown in Figure 2, the processing stages are implemented on different FPGAs. The input link handling and raw fragment buffers are

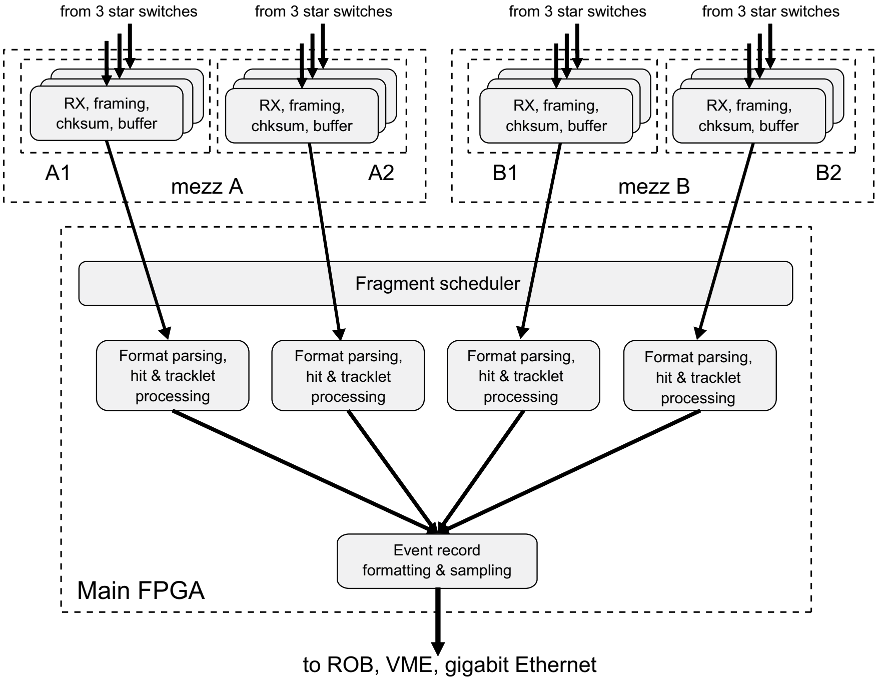
from 3 star switches
from 3 star switches
from 3 star switches
from 3 star switches
RX, framing, chksum, buffer
RX, framing, chksum, buffer
RX, framing, chksum, buffer
RX, framing, chksum, buffer
A1
mezz A
A2
B1
mezz B
B2
Fragment scheduler
Format parsing, hit & tracklet processing
Format parsing, hit & tracklet processing
Format parsing, hit & tracklet processing
Format parsing, hit & tracklet processing
Event record formatting & sampling
Main FPGA
to ROB, VME, gigabit Ethernet

Figure 2 Processing flow in the ROD. To handle the data rate, the same processing components are replicated and run in parallel. The first stage is replicated in four separate FPGAs on two mezzanine boards. There are four fragment processors. Point-to-point connections are used instead of a buses. The TTC information (L1 ID, BC ID, trigger type) is not shown.

[1] not currently implemented

implemented on two mezzanine boards, each with two smaller FPGAs. The scheduler, fragment processors and output formatter are implemented in one large FPGA on the main board. Not shown is the TTC and event header building on the main FPGA. The connections are via command-response links and FPGA-to-FPGA pipes (see below).

Pipes are used throughout the design as input and output data buffers and to marshal the data flow between the main data-driven stages as well as other parallel threads. They are implemented by Xilinx core FIFOs built from Block RAM with additional logic for synchronizing execution flow. Various pipe functions such as reading, writing, getting status are provided by Handel-C macro procedures. Similar procedures for exist for pipes within a clock domain, between clock domains, and between FPGAs. Pipes can be read out over the VMEbus. Pipes are usually used in pairs. The first, ‘data’ pipe, contains a number of variable length records or messages consisting of a variable number of FIFO words. For each record in the data pipe, the second, ‘control’ pipe, contains one word giving the number of words in the record and any error/status bits for that record. The control pipe is written only after a complete record has been written to the data pipe. The pipe consumer does a blocking read[1] on the control pipe; when that read succeeds the consumer knows that the next record is available and its length.

The physical block diagram of the TGC ROD is shown in Figure 3. Described below are each of the component systems. Several parameters of the FPGAs are shown in Table 2. The placement of parts on the main board is shown in Figure 4.

4.1 Separation of functions to mezzanine boards

The TGC ROD is a 6U VME64x module. There are two ~3U “back” boards, each with six input optical front end links and two link handler/buffer FPGAs. They plug into the back of the main board, not into the VME backplane. They use one VME slot width to yield a double width VME module. The output link is a standard S-Link connector for an S-Link mezzanine board. The ROD accepts Timing, Trigger and Control (TTC) signals directly via fiber to a TTCrx chip mounted on a small mezzanine board. All mezzanine boards have only a single 64-pin connector (Common Mezzanine Card, IEEE1386 standard) to minimize mechanical tolerance requirements. They are oriented vertically to maximize air flow. The different mezzanine boards are described later in this section.

4.2 Main FPGA

The central data handling and processing element is the main board FPGA. The Xilinx Virtex-II family [ref. 10], specifically the XC2V3000 has been chosen. (See Table 2.) In addition to low skew clock distribution over the FPGA, the digital clock managers, DCMs, allow arbitrary clock frequency multiplication and division and configurable or real time adjustable phase shifts. The Virtex-II clock distribution is divided in quadrants. Although up to eight of the up to 16 global clocks can be used in a quadrant, there are constraints as to how. Clocks and clock domains are described in Section 4.7.

The I/O signals are divided in eight banks. Since all signals in a bank must use the same I/O standard, e.g. 3.3V TTL, 2.5V CMOS, etc., care must be taken with the IO signal pin assignment. Since the SRAM and CAM use 2.5V I/O signalling, complete banks must be dedicated to them. The rest of the design uses 3.3V I/O signalling. In order to meet these clock and I/O pin constraints, the FPGA I/O pins were assigned manually.

[1] By “blocking read” it is meant that the execution waits until the data to be read is available.

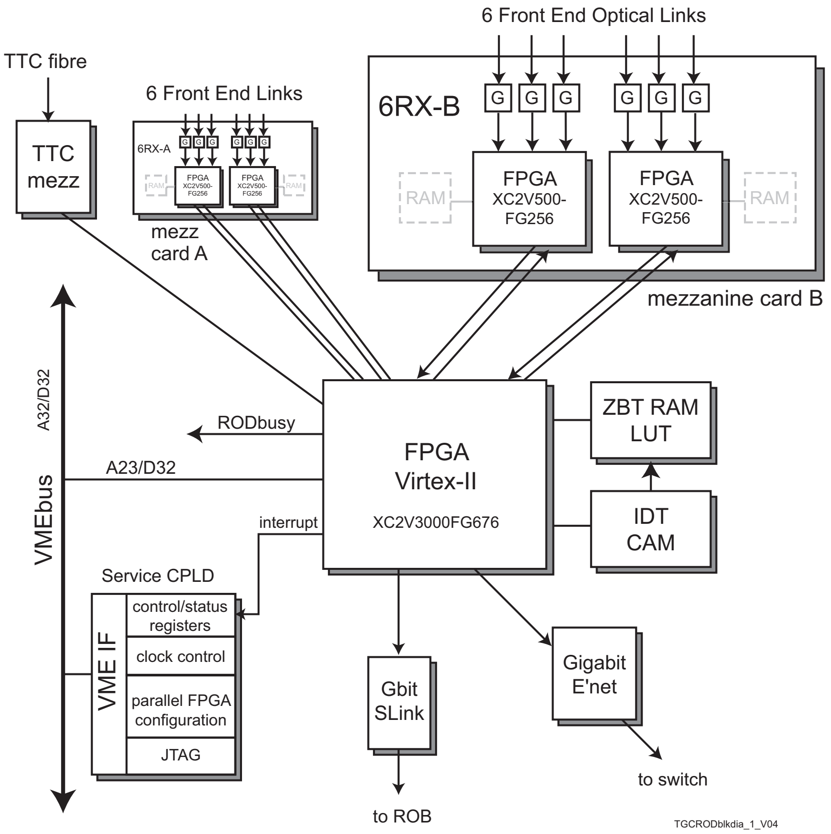

Figure 3 Block diagram of the TGC ROD. There are two front end link ~3U mezzanine boards (RXA, RXB) on the back and a TTC mezzanine and an S-Link mezzanine board on the front of a 6U VME64x main board.

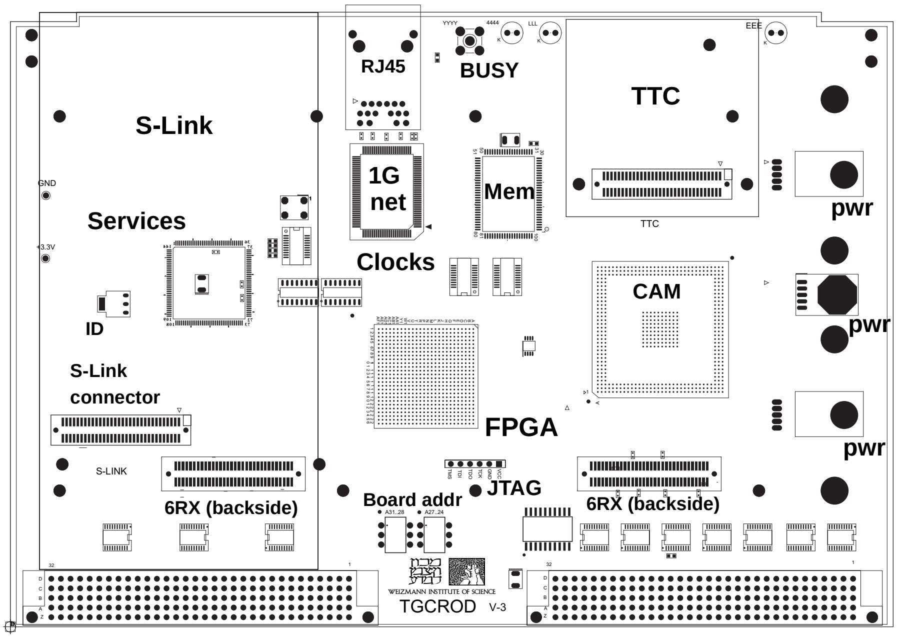

Figure 4 Placement of parts on the ROD 6U VME main board. The outlines of the S-link and TTCrms mezzanines are also shown. The two 6RX connectors and service PLD are on the back side.

Table 2 Some parameters of the FPGAs used. There are four RX mezzanine FPGAs, two on each mezzanine board.

	Mainboard	**RX mezzanine**	**RX total**
Family	Xilinx Virtex-II	Xilinx Virtex-II	
FPGA	XC2V3000FG676	XC2V1000FG256	
Input links	4 links from RX	3 FE links	12 FE links
Pins: package / user IO	676 / 484	256 / 172	
LUTs (FFs)	32,256	10,240	40,960
Maximum toggle freq / FF clock-to-output	750MHz / 0.45ns	750MHz / 0.45ns	
Block RAM	192K × 9bits	80K × 9bits	320K × 9bits
Digital clock managers	12	8	
Core voltage / CMOS process	1.5V / 0.12μ	1.5V / 0.12μ	

4.3 TTC signals

A small daughter card, TTCrms, with an on-board TTCrx chip supplies the TTC [ref. 11] signals, Bunch Clock (BC), Level-1 Accept (L1A), Bunch Counter Reset (BCR), and Event Counter Reset (ECR), and the Trigger Type to the main FPGA. The bunch crossing ID (BCID) and the Level-1 ID (L1ID) are counters internal to the FPGA that are incremented by BC and Level-1 Accept, respectively. On receiving BCR, BCID is not zeroed, but is instead loaded with a programmable offset that compensates for the time difference between arrival of BCR at the ROD and the time it resets the BCID in the on-chamber SlaveBoard ASICs. On receipt of L1A, the BCID and the full Level-1 event ID are stored in the EVENT ID FIFO in the FPGA. On receipt of ECR, the L1ID is set to 0 and the EVIDext (event ID extension) is incremented.

TTC Trigger type: The trigger type for an event arrives after the L1A. It is stored in a separate FIFO, since subsequent L1As may have arrived after a L1A, but before its matching trigger type.

Orbit Count: The ORBIT count (i.e. count of BCRs) is also added to the event header. The ORBIT count value allows correlating an event with the time-of-day which is used as the time stamp for DCS alarms and messages. The ORBIT count is held at zero until the first L1A of a run. The first L1A also interrupts the Rod Crate Processor which records the time-of-day of the first L1A.

The TTCrms daughter board is a small (53mx52mm) daughter board that consists of an optical receiver, TTCrx chip, optional PROM, CMC connector (64-pin CMC IEEE1386 SMD) and TTC Ready LED. All components are mounted on the connector side to allow fitting within a single-width VME module. The PROM is not installed, the main board FPGA can emulate it via the connector if needed. The connector is oriented vertically to minimize disruption of airflow. There are no switches or jumpers. Using a daughter board allows moving the TTCrx in the future to an upgraded version of the TGC ROD main board.

4.4 Receiver mezzanine boards

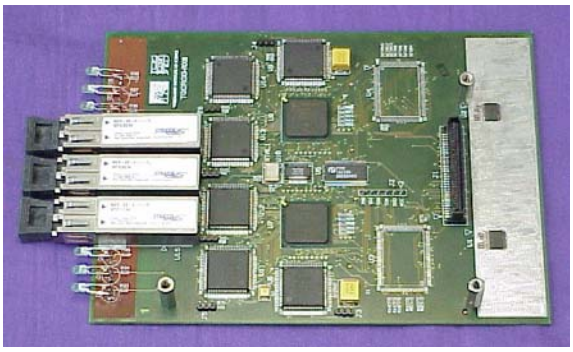

Figure 5 The 6RX board. The FPGAs are the two BGA packages in the center. Each opto-receiver package houses two opto-receivers. The heat spreading strip is on the right edge.

Each of the two 6RX mezzanine boards, Figure 5, serves six input Front End links. Three are serviced by each FPGA. The FPGAs implement the link handling as described above. In each FPGA there are 40 dual-ported RAM blocks providing 80KB of on-chip buffers. Typical fragment size at expected occupancy is 22 bytes. Should extra buffer space be needed, the boards allow for installation of an 512k×18-bit ZBT SRAM memory. Simulation and experience so far have not suggested more buffer space would be needed. The clock for the link receivers and FPGA is supplied by a custom 40.076MHz oscillator with 11psec r.m.s. jitter and is distributed point-to-point from a low skew clock buffer. (On the transmitter side, the link clock is the 40.076MHz BC clock distributed by the TTC system.) Due to the limited number of pins on the board-to-board connector, a zero-delay FET multiplexer allows reuse of the data pins for configuring the FPGAs on power-up. Each opto-receiver package houses two opto-receivers.

4.4.1 Input Front End links

The Agilent Technologies HDMP-1034A “G-Link” receiver chip [ref. 12] and the Stratos MTR-25-4-1-TL dual 850nm SFF optical receiver [ref. 13] are used to implement the Front End links. The G-Link is used on other ATLAS Front End links. The receiver operates without error even with 12dB of optical attenuation between it and the transmitter (an Infineon V23818 K305-L57 transceiver [ref. 14]). Since we do not need the bandwidth, the links are run at 40MHz, much less than their maximum 70MHz. This gives extra reliability and lower power consumption and a more than adequate 16bits×40MHz = 80MB/s data rate.

Crosstalk: Care must be taken in routing the high frequency, 800Mbits/sec, differential signals from the input link receivers. In order to eliminate crosstalk between the channels, the pairs are far apart. See Figure 6. Also the pairs are routed as strip lines, between power and ground planes, with a differential impedance of 100Ω specified for the traces. This also reduces EMI from the board. To be sure that there was no crosstalk, each channel was tested for crosstalk from its neighbours: Test data was sent to the channel under test and a pattern expected to be very noisy (alternating 0’s and 1’s) was sent to the neighbouring channels. No wrong data was received over several minutes of running at full link speed.

Event integrity: Each event is framed by a Beginning-of-Event and an End-of-Event word sent in Control Mode. Since these words cannot be event data words, re-synchronisation on event boundaries after any corruption of the data can be done without any ambiguity. To augment the Agilent HDMP-1034A/1032A error checking and to detect any Single Event Errors in the link chips, a 16-bit XOR check word is sent and checked for each event. Details of the event data format are in Section 5.

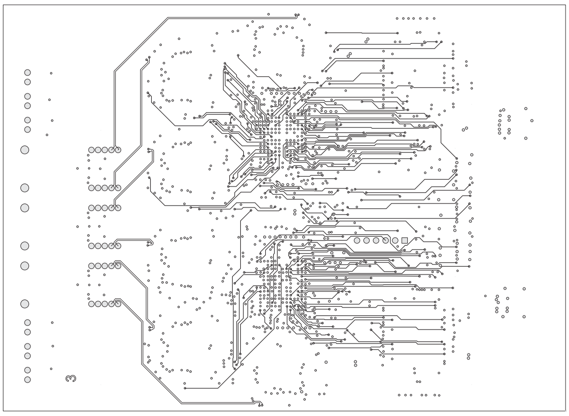

Figure 6 The PCB layer with the differential pairs carrying the 800Mbits/sec signal from the optical transceiver to the serializer. Note that the pairs are routed far from each other and between power/ground planes as striplines to minimize crosstalk and EMI.

1	signal	
2	ground	
3	diff strip lines from 1GHz serializers	
4	3.3V	
5	1.5V FPGA A core	1.5V FPGA B core
6	signal	
7	ground	
8	signal	

Figure 7 PCB layer stack-up for the 6RX board. To reduce EMI, the power planes do not extend as close to the board edges as do the ground planes.

4.4.2 6RX board power distribution and cooling

The 6RX board with all links running was measured to require 2.15A at 3.3V, i.e. 7.1W. The mezzanine connector has nine power and nine ground pins. The connector pins are rated by the manufacturer at 0.5A each. Each of the connector power pins is bypassed to allow it to be an AC ground. The signal, power and ground layer stack-up is shown in Figure 7. Extra large vias were placed very close to each of the power and ground pins of the connector and regulators to connect to the power and ground planes.

To supply its FPGA core, each 6RX FPGA has its own 3.3V to 1.5V linear regulator. A large tinned-copper flood area serves as a heat spreader for the regulators, as recommended in their data sheet. It is the full height of the board (11cm) by about 3cm width. (See Figure 5.) Heat conductive grease was put between the G-Link chips and the PCB. A power dissipation of 14Watts for one VME slot, spread fairly uniformly over the board, is very reasonable: the specified maximum for basic VME is 35W per slot. Furthermore, since there is a whole VME slot width available for the back-side board, the 6RX board sits at a distance of 1.5cm instead of 1.0cm as for standard CMC mezzanine boards such as S-Link. This gives better air flow, especially for the 1cm high optical receivers.

4.5 VME interface and Service CPLD

The VME interface and other board services are provided by the service CPLD, a 256 macrocell Xilinx CoolRunner XPLA3 CPLD (XCR3256TQ144, 144 IO pins). It provides the following services (See Section 7, "VME Registers", below for more details.):

The VME interface is a standard A32/D32 slave and interrupter. The A32/D32 address space that the board occupies is selected by two rotary switches. There are no other jumpers; the IRQ level is set in a programmable register. The supervisor/non-privileged bit in the VME Address Modifier is ignored. No advanced VME protocols are used. The VME64x geographical addressing and configuration space are not implemented. The current ROD FPGA design does not generate interrupts.

Other Services:

- byte-parallel configuration of the main FPGA. Configuring takes less than one second.
- some of the control and status registers
- two direct lines to and two from FPGA pins
- board reset
- control and programming of the two programmable clocks
- a Microwire port to read the silicon serial number of the VME board. A Dallas DS2401 chip [ref. 15] provides a unique serial number written by the manufacturer.
- JTAG for boundary scan of board components

4.6 Data link between the 6RX and main boards

The link between each RX FPGA and the main FPGA has two parts: a high bandwidth uni-directional link for block transfer and a low bandwidth bi-directional channel for short command–response pairs. The latter is used to query status and front end FIFO occupancy and to set-up the block transfer. The block transfer was originally intended to be implemented using Xilinx's "SelectLink" logic core [ref. 16] and this name remains even though the implementation is home-made. The main features of the FPGA-to-FPGA link are:

- There is a FIFO interface on both ends.
- Source, destination and link clocks can be different speeds.
- The link clock is forwarded from source to destination.
- The data path is 8-bits wide so a 16-bit data word is sent as two bytes.
- The link is "wrapped" by Handel-C to look like a pipe.

All transfers are validated by checking the XOR check word on arrival.

The command–response channel consists of a 4-bit bi-directional synchronous data bus with control and status signals: strobe, ack, service-call and all-links-ready-with-data. Four signals are shared by both RX FPGAs on the board: config-mode, RODBUSY, reset, spare. Config-mode sets the on-board multiplexer to route some connector signals to the FPGA parallel configuration ports. This allows the main FPGA to configure the RX FPGAs. (Signals used by the SelectLink are not multiplexed.) The commands and their response are shown in Table 3. The protocol is not fast, only 4-bits are sent per clock, but most commands are infrequent. When the event rate is high, all Front End FIFOs will contain data and checking the all-links-ready-with-data status line will eliminate the need to issue the get_status command to know if a front end link has data to transmit.

The bus clock is the Main FPGA design clock, is forwarded to the RX FPGA from the Main FPGA. The RX-side of the command–response channel's logic runs synchronously with the transmit side. In this way the data flow is synchronous over the connector. Should the RX FPGA design clock be different from the Main FPGA's (e.g. generated by a DCM from the 40MHz oscillator on the 6RX board), the re-synchronization between the domains is done inside the FPGA, as opposed to asynchronous hand shakes over the connector.

Table 3 Protocol for the Command–Response channel between RX and Main FPGAs

Command	Command byte			2-bytes returned	Notes
get_status	1	*reserved*		status	status for 4 FE Links: link ready, control FIFO empty, data overflow, busy (AlmostFull) flags
send_event	2	*reserved*	channel number	word count, flags	word count is the number of words to be transferred on the SelectLink
get_occupancy	3	*reserved*	channel number	occupancy count	used by DAQ to monitor FE FIFO occupancy
clear_overflows	4	4-bit channel mask		status	major error recovery
enable_links	5	4-bit channel mask		status	at start-up

4.7 Clocks and clock domains

There are several clock domains on the board and in the various FPGAs. (See Table 4.) Logic in each is separated from the others by a FIFO, except for the RX command–response channels as described above. (The local bus, derived from the VME bus, is asynchronous.) A 25MHz oscillator provides the “reference” clock to the FPGA and to the two programmable oscillators.

Table 4 ROD clocks

Clock	Frequency
FE links: RXA[0..7], RXB[0..7]	40.076MHz
6RX board reference clock	40.076MHz
RX FPGA design: A0, A1, B0, B1	40MHz
SelectLink clocks: A0, A1, B0, B1	>40MHz
Main board reference clock	25MHz
Main design	25–60MHz
CAM and SRAM	25–100MHz
Secondary design clock	25–100MHz
TTC BC	40.076MHz
S-Link	≤40MHz
Chipscope	25–100MHz
MAC TX, MAC RX	125MHz

Front End link domains: Each of the 12 Front End link G-link chips emits data synchronous to the clock reconstructed from its transmitter. This reconstructed clock is the clock for that channel’s link control logic and for clocking the received event into a FIFO.

RX FPGA design domain: This domain is between the front End Link FIFOs and the Select Link and RX command–response channel.

SelectLink domain The SelectLink transfer clock is independent of both transmitter and receiver design clocks.

TTC domain: The TTC signals are synchronous with the LHC beam crossing clock, BC. For each Level-1 Accept received, an event ID consisting of full L1ID, BCID, and orbit ID is stored into one FIFO, and trigger type, some micro-seconds later, into another FIFO.

FPGA design domain: This is the clock for the main logic in the Main FPGA. It is generated by the external programmable clock.

SRAM and CAM domain: The SRAM and CAM operate together in their own clock domain, probably at a faster clock rate than the Main FPGA design. The CAM transfers on both clock edges and requires a 2× clock for internal use. A Virtex FPGA DCM provides this synchronous 2× clock.

FPGA design secondary domain: Some logic in the FPGA may operate at a higher or lower clock rate than the main logic. This clock is generated from the reference clock.

Read Out Link domain: The data for the output link is read from the output event FIFO at a clock speed suitable for the output S-link, 40 MHz (the data width is 32 bits). This clock is generated from the reference clock by an FPGA DCM.

1-Gigabit Ethernet MAC TX domain The 1-gigabit ethernet MAC transmitter logic operates at 125MHz (the data width is 8 bits). This clock is produced from the board's 25MHz reference oscillator by the 1-gigabit ethernet physical layer chip. It is forwarded with the transmit data to the physical layer chip.

1-Gigabit Ethernet MAC RX domain The 1-gigabit ethernet MAC receiver logic operates at 125MHz (the data width is 8 bits). This clock is forwarded with the received data by the 1-gigabit ethernet physical layer chip.

Chipscope domain: The Xilinx Chipscope embedded logic analyzer tool uses a clock for sampling the signals being captured. This clock can be independently specified. It is generated by the external programmable clock.

4.8 S-Link output to the ATLAS Read Out System (ROS)

The Read Out Link to the Central ATLAS Read Out system (ROS) is an ATLAS standard Read Out Link, i.e. a HOLA S-Link mezzanine board [ref. 17]. Only 3.3V S-Link daughter boards are supported. Although the 32-bit width is expected, the output S-Link's width lines, UDW0 and UDW1, can be set by jumpers. The older ODIN S-Link accepted data with a clock up to 32MHz; the new HOLA S-Link, up to 40MHz.

4.9 Gigabit ethernet link

The ROD's gigabit ethernet link enables monitoring sampled events from all RODs without depending on the ATLAS Central DAQ and without being limited by the VMEbus bandwidth. This flexibility is especially helpful during setup and debugging the TGC system. The event sample is unbiased by the Level-2 trigger and the sampling rate can be very high. All the RODs are connected via a standard commercial gigabit network switch to one or more multi-core rack-mounted computers which accumulate histograms. The RODs, switch and monitoring computers form a private network. The monitoring computers are also connected to the ATLAS control network.

Each event is packaged in a single UDP packet. Jumbo packets, with up to 8KB event payload, are used. A ROD's IP address is 192.168.1.N, where N is 32+ the ROD's rodid (1 – 12 for Endcap A, 17 – 24 for Endcap C).

The ROD's gigabit link is implemented by the 1-gigabit ethernet MAC core provided by Xilinx[1] [ref. 18] and the physical device layer ASIC from Marvell (also used by the ATLAS ROBIN), using the GMII interface mode. Only the copper (1000Base-T) connection is implemented. The FPGA resources needed by the MAC are modest, i.e. ~600 of 14,336 slices, which is ~4% of the XC2V3000, 25 pins, two digital clock managers and two global clock buffers.

[1] We gratefully acknowledge receiving their 1-Gigabit Ethernet MAC core as a donation from Xilinx Corp.

4.10 SRAM and CAM Look-Up tables

The data stream from the TGC on-chamber read-out and trigger coincidence matrices consists of their input binary hit patterns (160 bits) and their resulting output from the coincidence logic (40 bits). The coincidence output is up to four $R,\Delta R$ or $\phi,\Delta\phi$ pairs. Although this comprises complete information, additional information very helpful for monitoring and later stages of the event processing can be extracted. This extraction is time consuming in software but is fast in hardware and can be done in the ROD. The following additional processing can be done:

- Identify the specific hits that make each coincidence. This is needed by Level-2 and doing it in software was observed to be time consuming for the RPCs.
- For monitoring, confirm that the coincidences found are consistent with the hits.
- Translate from data path format, i.e. link-id / SB-id / SB-channel, to a chamber oriented format where hits are identified by TGC layer, chamber, strip or wire, strip or wire number. This is useful for monitoring by chamber and essential for track finding.
- Subject to the limitation that the ROD sees only events that have been triggered by Level-1, check that there are no missing coincidences.
- Optionally, find low-p_T coincidences with asymmetric cones or cones larger than those of the Level-1 hardware.
- Find any second coincidence in the same quarter of the coincidence matrix. (Although rare, this cannot be output by the Slave Board ASIC.)

This functionality will be implemented in a way that will not compromise the primary functions of the ROD. It will couple to the main data flow in the ROD in an unobtrusive manner and will be activated or not according to a control register bit. The above functions can be performed simply and together by means of a ternary CAM and coupled SRAM. A CAM (Content Addressable, or associative, memory) takes an input pattern and outputs the address in the CAM where it is stored. In a ternary CAM each stored pattern has its own mask which selects the bits in the pattern to be compared. The match address can be used to address a conventional SRAM memory to provide the data associated with the matched pattern. We have selected the IDT 75K72100 18Mbit CAM [ref. 19] and the IDT 71T75702 512K×36 bits ZBT SRAM memory [ref. 20]. The CAM capacity is 18Mbits and can be configured with various word widths from 36 to 576 bits. Our patterns would use the 288-bit width allowing 64K patterns to be stored. Four 72-bit transfers to write the hit and coincidence bits to the CAM are required. The CAM is designed to work with a companion ZBT SRAM without additional logic. For details of the use of the CAM for hit pattern processing, see Section 4 and [ref. 21].

The 2MB SRAM (Flow-through ZBT, Zero Bus Turnaround) can be addressed either by the FPGA or by the CAM and can run up to 100MHz. The memory can also be used for conventional look-up tables and buffers. The CAM and SRAM are accessed only via the FPGA. A 4MB SRAM memory chip will be available in the same package and pin-out.

4.11 Power distribution

The main power source is the 3.3V from the VME64x backplane. Unfortunately, in addition, 1.2V, 1.5V and 2.5V are also needed as shown in Table 5. These are provided by linear regulators with 3.3V input. There are no switching regulators on the ROD board. Distribution is via power planes. Only voltages used for chip cores are put on split power planes. The stack-up is done so that split planes are never adjacent to signal planes. The TTCrx optical receiver diode uses 5V from the VME bus.

Table 5 Components using 2.5V, 1.5V and 1.2V

	3.3V	2.5V	1.5V	1.2V
FPGA	IO	IO	core	—
SRAM	—	all	—	—
1-Giga Phys	IO	core	core	—
CAM	—	IO	—	core

Prior to the full design, we estimated that the FPGA core would require less than 3A at 1.5V. Estimates for the CAM core are less than 2A. The gigabit phys requires 3.3V: 120mA, 2.5V: 240mA, 1.5V: 325mA. The SRAM requires 2.5V: 275mA at 100MHz. The HOLA S-link source requires 3.3V: 550mA. The two 6RX boards require at most 3.3V: 3.0A each. (We measured 2.15A with all links running, but with only a simple FPGA design.) From these we estimated the upper limit on current to be supplied via the VME 3.3V lines to be 14A. The current actually measured by the crate for the 3.3V with no CAM / SRAM operating was 10A. TheVME64x specification rates the ten 3.3V pins at 1.25A each. This rating assumed all 160 pins in the connector were carrying the maximum current. The recently approved ANSI / VITA standard, 1.7-2003 [ref. 22] takes into consideration that the other signal pins not only do not carry significant currents, but act as heat sinks for the power pins. The maximum current for the VME 3.3V power pins is given as 2A per pin, giving a maximum current for the ROD module of 20A. Recall that, for the ROD, this power is dissipated over two slots.

4.12 Facilities for debugging, testing and instrumenting

Running with simulated events:
A board with four Front End Link transmitters can be mounted on the prototype-0 ROD board instead of the receiver card. Simulated events can be loaded into its memories, 512KB SRAM per Front End Link and sent over fibers to a ROD. The event transmissions can be driven by the TTCvi random trigger generator at full LHC rates. Events have been simulated with hits and track occupancies according to the ATLAS cavern background simulation. The same events can be read into the Handel-C code running in simulation mode, as can, and were, real test beam events. Deliberately corrupted events have been used to test error recovery.

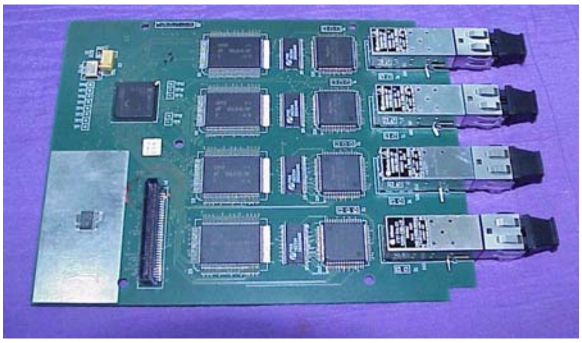

Figure 8 The 4TX board for sending simulated event data to the ROD. There is one 512KB SRAM per link for holding events. The control FPGA is on the left.

Chipscope, the Xilinx embedded configurable logic analyzer, is regularly used to examine signals inside the FPGA. It communicates with the FPGA via the FPGA JTAG lines.

An external logic analyzer can be connected to the S-link connector instead of the S-Link. This provides 46 pins to extract internal FPGA signals.

The exception pipe A very powerful tool is the pipe used to route 4-byte messages from the FPGA to the VME processor. When the pipe is not empty a VME interrupt is made and the VME processor reads the full contents of the pipe. Each message consists of a 2-bit message type (info / debug, event error, system error), a 6-bit message id and three bytes of context information. The info / debug message type can be used to trace execution and show intermediate values.

Pipe read-out Internal pipes can be read via the VMEbus in order to examine intermediate data.

Occupancy In order to monitor buffer occupancy, the number of words in each FIFO can be read via the VMEbus at any time.

4.13 Front Panel LEDs and connectors

Connectors	LEDs
BUSY output to the ROD BUSY module (LEMO)	RODBUSY
The TTCrx input fibre (ST connector)	VME module select
12 fiber Front End Link inputs (LC connector)	TTC ready
S-Link fiber output (duplex LC connector)	Level-1 Accept
	G-Link lock for each Front End Link
	S-Link PWR, TST, ERR, UP, XOF, ACT

5 Front End Link data format

The Front End links send events from the Star Switch which collects data from up to 18 Slave Board ASICs on the detectors. Each event is a record framed by words sent in G-link control mode. There is no flow control; the ROD must issue RODBUSY to the Central Trigger Processor when its buffers become almost full. The 160 wire or strip input bits and 40 coincidence matrix output bits from the on-chamber Slave Board ASIC are divided into 8-bit cells. Only the non-zero cells and their cell id are transmitted on the Front End links from the Star Switch to the ROD. Since the TGC occupancy is low, this substantially reduces the number of words from the Star Switch that must be processed. Table 6 shows the format for the Front End link data.

Originally only headers for Slave Boards with data were to be sent to the ROD.Work is in progress to add a pre-processing stage in the ROD mezzanine board FPGA to discard headers for which there is no data. There is one such pre-processor per Front End link. Also the second Slave Board header was to be sent only on error, i.e. when its data words were non-zero. The pre-processing stage will also discard these headers when there is no error. The discarded words account for more than 80% of the data transferred. Without discarding them, the down stream processors cannot handle a 100kHz Level-1 rate with five times the expected occupancy, as required by ATLAS.

The output format of the pre-processor is similar to the FE format. Since the SSW RXID is needed to look-up a unique slave board location id, the three BCID and the two unused bits in the Slave Board header are replaced by the 5-bit RXID from the second Slave Board header word. Word type 0b010 is used to communicate exceptions to the subsequent processing stages.

Table 6 Front End Link data format.

Event Header 000

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">000</td><td colspan="2">RecType</td><td colspan="4">SSWID</td><td colspan="23">RX mask pattern (1=enabled, 0=disabled)</td></tr>
</table>

Record Type (RecType) is **01** in this format version, hard-wired in FPGA
SSWID is set by a dip-switch on each SSW board.

SLB header 010 This word was intendended to be sent only if there was data for that SLB, but it is sent for every SLB.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">010</td><td colspan="5">SLBID</td><td>0</td><td colspan="3">BCmap</td><td colspan="3">Mod Type</td><td>0</td><td colspan="4">L1ID</td><td colspan="12">BCID</td></tr>
</table>

BCmap shows 3 BC data lines read by SSW. The 3 bits are {next, current, previous} BCs; 1=read. 0=discarded
SLBID, Mod Type, L1ID and BCID are all SLB's data. See SLB documents.
Unfortunately there are duplicate SLBIDs for some Star Switches. The RXID must be used with a Look-up table loaded at configuration to generate a unique SBLOC for all the Slave boards read by SSW.

SLB header 011 –0 This word was intended to be sent only when the low 20 bits were non-zero, but it is sent for every SLB.

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">011</td><td>0</td><td>0</td><td colspan="5">RXID</td><td colspan="2">0</td><td colspan="4">RX FIFO status</td><td colspan="8">SLB-OVF</td><td colspan="8">RX-OVF</td></tr>
</table>

RXID is the SSW RX ID (SSW input port) from 0 to 22.
RX FIFO status tells what amount of data are stored in RX-FIFO then.
SLB-OVF is SLB's data. See SLB documents.
RX-OVF is RX-FIFO overflow counter. It is the value when this word is sent from the SSW RX to SSW TX.

SLB trailer 011 –1 This word appears after SLB data words only when there is an error

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="3">011</td><td>1</td><td>SEU</td><td>OVF</td><td colspan="2">LVDSlnk</td><td colspan="8">RX error state</td></tr>
</table>

LVDSlnk=LVDS links status. 2bits are {now,old}. 1=Not linked. 0=Linked.
SEU = SLB SEU flag. See SLB documents.
OVF = RX-FIFO overflow flag. If OVF=1, some overflows have happened in this RX data.

SLB data 100, 101, 110

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>In any order:</td></tr>
<tr><td colspan="3">100</td><td colspan="5">cell address</td><td colspan="8">cell bitmap</td><td>Cell data for Current BC data</td></tr>
<tr><td colspan="3">101</td><td colspan="5">cell address</td><td colspan="8">cell bitmap</td><td>Cell data for Previous BC data</td></tr>
<tr><td colspan="3">110</td><td colspan="5">cell address</td><td colspan="8">cell bitmap</td><td>Cell data for Next BC data</td></tr>
</table>

PAD word 110

<table>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td></td></tr>
<tr><td colspan="3">110</td><td colspan="5">11111</td><td colspan="8">0</td><td>i.e. 0xDF00</td></tr>
</table>

Event Trailer 111

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="3">111</td><td colspan="9">0x1CA</td><td>Glnk</td><td>T1C</td><td>NRC</td><td>T2C</td><td colspan="16">XOR check sum</td></tr>
</table>

Glnk = Glink TX status. "Locked" signal of Glink Tx. 1=Not locked. 0=Locked
T1C = Timeout1_count_frag. Time-out to collect the event fragment from all the enabled input ports.
NRC = Nores_count_flag. No response from RX FIFO of "enabled" (not masked) input port.
T2C = Timeout2_count_flag. Time-out to collect the event fragment from each enabled RX FIFO.
These three flags are reset at every event.

The XOR operation includes the first word(16bits) of the event header through the first word of the event trailer. When the result is XOR'ed with the XOR checksum word, the result becomes zero. (the XOR does not include the 0x0B0F an 0x0E0F framing words)

Framing

Each event is preceeded by the 32-bit word 0x0000'0B0F and followed by the 32-bit word 0x0E0F'0000, both words are sent in G-link control mode.

6 Firmware design

Figure 9 shows a simplified data flow diagram of the main FPGA design. The diagram is explained here. Data flow between processes is via pipes (see Section 4, "Architecture"). The event manager reads a Level-1 ID and starts a thread to read its associated data and to write it to the event header pipe. For each of the four board-to-board FPGA-to-FPGA links there is a fragement scheduler and a fragment processor. Each scheduler polls its associated FPGA, via its RX command-response channel, to see if a fragment for this event is ready for transfer. If the fragment processor is also available, a command is sent to start the transfer and the response provides the length of the fragment. The fragment processor is then sent the SSW id, fragment

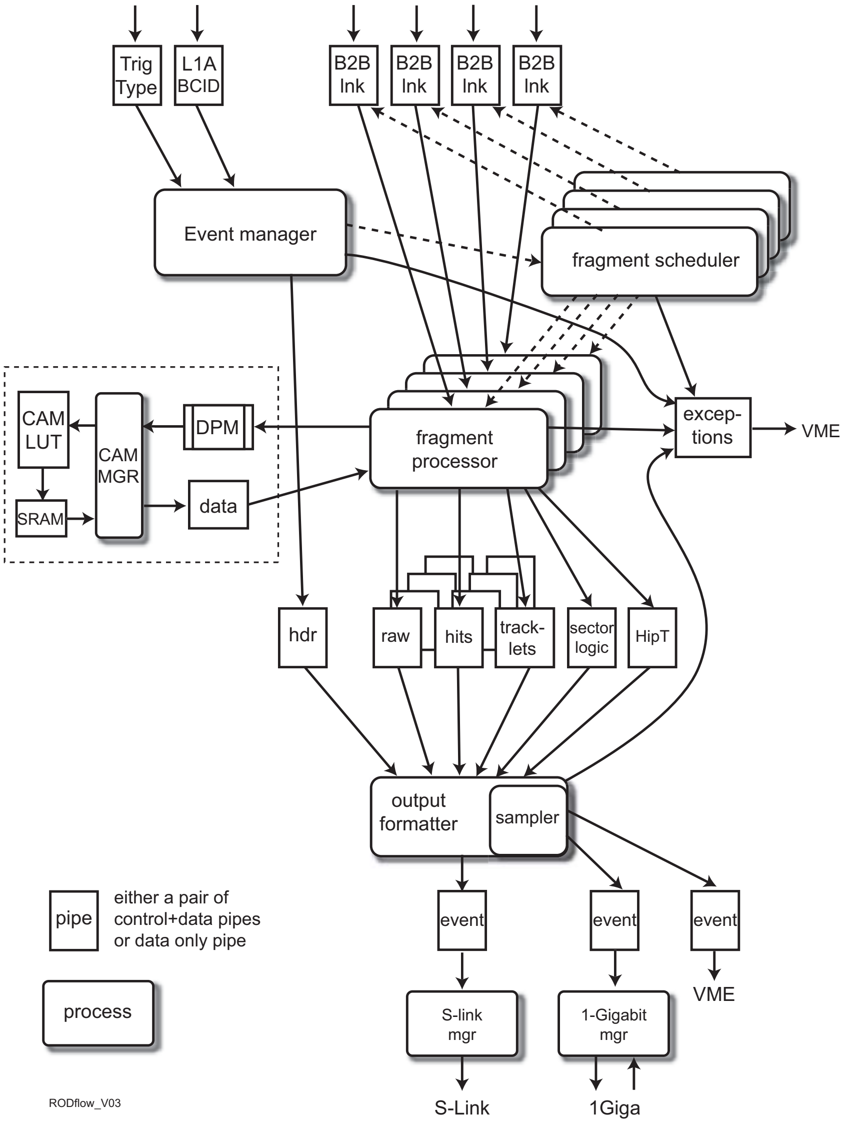

Figure 9 Simplified data flow diagram of the FPGA design. Processes are shown as rounded boxes; pipes are shown as rectangles. Fragment schedulers control the transfers over the board-to-board links. Only one set of arrows to and from the raw data and hit pipes, which are replicated for each fragment processor, are shown. There are only single instances of the exception, sector logic, HipT, and tracklet pipes. The CAM/LUT logic (inside the dashed rectangle) is not yet implemented.

length and other flags and begins to process the fragment as it is being transferred. For simplicity, all fragments of the current event are processed before any fragments of the next event are processed. The four links and processors work in parallel.

The output of each fragment processor is split into several output pipes, e.g. hits, tracklets, etc. When all fragments have been processed the number of items in each data pipe are written to the respective control pipes. Finally the event header is completed with error and other information from fragment processing and is written to its pipe. The pipes can hold the data from several events.

Fragment processing Fragment processing includes detailed checks of the data and format integrity. The 8-bit bit-maps of the non-zero cells of the 200-bit Slave Board ASIC output (160 bits of channel hits, 40 bits of coincidence logic output) are decoded to one or more individual hits and coincidences. The Sector logic input (e.g. HipT output) and output fragments are decoded. Optionally, the input to the fragment processor can also be written out.

Exceptions Processes can issue exception messages, e.g. various kinds of invalid data, too many hits, etc. Since these are expected to be infrequent, they are serialized (by means of a semaphore) from all processes into one pipe. The ROD crate processor reads the exception pipe in response to a service call interrupt request made whenever the pipe is not empty. Table 7 shows the list of possible exceptions that the fragment processor may issue. This gives an idea of the level of error and consistency checking done.

Output formatter and event sampler The output formatter works in parallel to the fragment processing. It is driven by the header item in the header pipe, which includes the TTC information, and the control word for each of its input pipes. The formatter then knows ahead of time how many hits, tracklets, etc. will be in the event record and can build the appropriate event header, followed by the event data organized into a hit block, a tracklet block, a raw data block, a sector logic output block, etc. The output formatter also samples events for transmission to the VME processor and/or the gigabit ethernet link. Sampling can be based on trigger type, presence of errors, or other “filter” conditions. Events are passed to these links only when their pipes are not almost-full, unless the operating mode is to force all events via these links. Otherwise, the S-Link output is never blocked by them. Should the ROS pause the S-Link for a long period, eventually all the intermediate FIFOs will fill and at least one input FE link FIFO on an 6RX board will become almost-full. This causes RODBUSY to be active.

S-Link and 1-Gigabit link managers control the data flow to the S-link and gigabit link according to the XON/XOFF protocols from the ROS and gigabit destinations. The managers each operate in a clock domain different from the main design and each other.

Ternary CAM Look-up table performs the functions described in Section 4.10. An array of 8-bit dual port memories are used to assemble the 200-bit data word, plus extra control and ID bits as the search pattern for the CAM. (The CAM contains up to 64K patterns.) Each fragment processor has two dedicated assembly areas. Only the non-zero 8-bit cells need to be written. A bit map indicates which cells are non-zero. The CAM manager polls the DPM for requested searches and issues four 72-bit DDR writes to the CAM. Searches can be streamed to the CAM one after the other. After a latency of a few clocks the results of the search are read from the SRAM and sent back to the fragment processor via a pipe. The results consist of several words of data associated with the pattern matched. These words would include, for example, the list of hits in chamber-oriented format, marked according to whether or not they are in the coincidence corresponding to the pattern, hit-coincidence consistency flag, etc. The CAM Manager operates in its own clock domain, up to 100MHz.

Table 7 List of possible exceptions from the fragment processor. For a detailed interpretation of the exception messages, see the document, Endcap Muon trigger system: ROD error messages [ref. 23].

Front End link G-link error	RXfifo has overflowed
Invalid FE link framing words	SSW reports T1C, NRC, T2C, or GlinkNoLock error.
Input FE event is too long or FE FIFO overflow	Bad XOR checksum from mezz board
Unexpected nulls in FE input	Invalid XOR event checksum
Unrecognized record type	First word is not header
Invalid or unexpected Star Switch ID	Word is out of order
Data from disabled SSW RX ID	Error in request to send an event via RXlink
Bad End-of-event marker received, not 0xFCA	RX ID is duplicated in the event
A Front End link has timed out - abandoned	Sector Logic BCID[2:0] does not match its SB BCID
SBid does not match SBinfo table	SBtype does not match SBinfo table
Unexpected SB BCID	Sector Logic reports G-Link error
Unexpected SB L1 Event ID(lo 4)	
Invalid cell address	SSW: SB fifo overflow
Too many hits in event	SSW: RX fifo overflow
Timeout expired for at least one FE link	SSW: new input LVDS link down
No End-of-event marker received	SSW: old input LVDS link down
L1ID mismatch (TTC EVID FIFO vs local).	SSW: RX error state
Event has WC=0 or WC > max WC	SSW: Slave board had SEU
WC not 0 after EoE marker	Formatted output event not sampled: too long.
Unexpected trigger bits	Xmit err from mezz board.

7 VME Registers

The registers presented to the VME bus are shown in Table 8, for the Service PLD registers and Table 9. for the FPGA registers. The FPGA status and control registers give a good idea of the options available. Refer to the .h files in the code repository for the latest versions.

Table 8 Register set 0 (Service CPLD)

offset	R/W	name		# bits	SVC pin	pin type
00'0000	R	**SR0**	**Status register 0**	**8**		
		0	/INIT	1		
		1	FPGA configuration done	1	1	IN
		2	interrupt pending	1	1	IN
		3	RODBUSY	1	1	IN
		4	TTC ready	1	1	IN
		5	output S-link ready	1	1	IN
		7:6	fromFPGA	2	2	IN
00'0004	RW	**CR0**	**Control Register 0**	**8**		
		0	interrupt enable	1		
		1	force_RODBUSY	1		
		2	microwire port for uniqueID	1	1	IO
		3	TTC_SDA	1	1	IO
		4	TTC_SCL	1	1	OUT
		5	TTC_ENA	1	1	OUT
		7:6	toFPGA	2	2	OUT
00'0008	W	**CMR0**	**Command Register 1**	**3**		
		0	reset board	1	1	OUT
		1	start configuration	1		
		2	test interrupt	1		
00'002C	W	**CFG**	**Configuration data register**			
			data[7:0]	8	2	OUT
00'0010	RW	**PHYS**	**E'net Phys config**	4		
		0	/Reset	1	1	OUT
		1	coma (deep sleep)	1	1	OUT
		6	maintenance port clock	1	1	OUT
		7	maintenance port data	1	1	OUT
00'0014	W	**CLK**	**Clock control reg**	6		
		3:2	Enable Clock[1..0]	2	2	OUT
		1:0	data[1..0]	2	2	OUT
		5	strobe	1	1	OUT
		4	programming clock	1	1	OUT
00'0018	R	**BVER**	**Board version**	4		
00'001C	RW	**IRQ**	**Interrupt request level**			
			irqlevel	4	7	OUT
00'0020	R	**SW**	**Software INT**			
00'0024	RW	**IVR**	**Interrupt vector register**			
		7:0	vector	8		
00'0028	R	**SERR**	**TTC Single Error**	8	1	IN
00'002C	R	**DERR**	**TTC Double Error**	8	1	IN
00'0030	RW	**CFGMEZZ**	**Set mezz mpx for configure**			
		0	control bit	1		

Table 9 Register set 1 (internal to main FPGA). See next section for control and status register bits.

FPGA regs (1MB space) (see TGCrod_FPGAreg.h for details of bits			
offset	**R/W**	**name**	
10'0000	R	SR1	FPGA Status Register 1
10'0020	R	GBSR	Gigabit Ethernet status register
10'0014	R	FVER	Firmware version
10'0030	R	FERDY	FE link ready flags (from G-link chips)
10'0004	R	FFR	FE FIFO full indicators: hi 16: ovfl, Lo 16: busy
10'000C	R	FEMT	Low 12 bits: FE FIFO empty flags (4 rx * 3 FEchan)
10'0028	R	FEOUT	high 16: LDB_dropped[12]; low 16: LDB_timedout[12];
10'001C	R	BTIME	accumulated RODBUSY time (in microsecs) 32 bits=1.2 hours
10'002C	R	STALD	stalled pipe write flags
10'0010	R	DBGR	data for debugging
10'0034	R	DIAG	diagnostic bits
10'0018	R	DIAG2	2nd set of diagnostic bits
10'0008	R	ERRS	error flags for the current event
10'0040	R	LINF0	last info msg issued, for debugging, from frag proc0
10'0044	R	LINF1	last info msg issued, for debugging, from frag proc1
10'0048	R	LINF2	last info msg issued, for debugging, from frag proc2
10'004C	R	LINF3	last info msg issued, for debugging, from frag proc3
10'0050	R	LINF4	last info msg issued, for debugging, from fragment scheduler
10'0054	R	LINF5	last info msg issued, for debugging, from outpur formatter
10'0024	R	L1AR	Last L1A received
10'0108	R	L1AP	L1A being processed by frag processors
10'0100	R	L1AX	Last L1A transmitted
10'0104	R	VMFW	number of formatted event words transfered to VME
10'010C	R	NGIG	number of events transfered out via the gigabit ethernet link
10'0110	R	NEVS	number of events processed
10'0200	RW	FCR1	Control Register 1
10'0204	W	CMR1	Command register 1
10'0230	RW	SVCR	Service Call request Register
10'020C	WS	SVCA	acknowledge that requested Service Call has been executed
10'0214	RW	RUN	run number to be included in formatted events
10'022C	RW8	EL1ID	Extended L1 ID
10'0234	RW	RODID	ROD ID, high bit is side
10'0220	RW	TGCC0	FE connection config reg: 4bits each: ch7LDB ... ch0LDB
10'0224	RW	TGCC1	FE connection config reg: 4bits each: ch11LDB ... ch8LDB
32'0000	RW13	SBINFO	data for each SB at SSWid*32 + RXID: SBID(5) SBloc(5) SBtype(3)
10'0210	RW16	BCOF	BC offset
10'0238	RW	EMUTE	bits to mute error reporting for each event error check
10'023C	RW	EMUTS	bits to mute error reporting for each system error check
10'0240	RW	TTACC	a bit on means monitor events with this Trigger Type bit on
10'0218	RW16	TSTN	number of test events to send, 0 means infinite
10'0208	RW16	MXR	Memory Extension Register (extra 4 bits of memory to/from SRAM)
10'030n	R	occupancy counters for FE link n, for n= 0..b	
10'0330	R	FIFO occupancy counters: exceptions, evid, history, hipt, evout, tracklets, giganet, sector logic, evhdr, raw0, raw1, raw2, raw3, hit0, hit1, hit2, hit3	
10'0370			
20'0000	R	2MB: 8K word (64KB) window each, for up to 16, FIFOs to allow use of DMA	
21'0000			
...			
2F'FFFC			
40'0000	RW	2MB ZBT RAM memory space	
5F'FFFC			
60'0000	RW	2MB IDT LUT/IP-Coproc space	
7F'FFFC			

7.1 FPGA control, status and command register bits

Refer to the file, TGCrod_FPGARegMV1.h, in the code repository for the latest version.

Table 10 Control Register 1: FCR1

Bit name	Meaning
`CR1_ALLFMT`	ALL formatted events are sent to VME
`CR1_ERRFMT`	Formatted events with any UNMUTED error bits on are sent to VME
`CR1_FLTFMT`	Filtered formatted events are sent to VME
`CR1_GIGA_SAMPLE`	Sampled formatted events are sent to gigabit ethernet
`CR1_FLTFMT_GIGA`	Only sampled FILTERED formatted events are sent to gigabit ethernet
`CR1_INCLSSW`	Include copy of raw fragments received from SSW in the formatted event
`CR1_SYSMAP`	Enable sysmap mode where raw data output is forced and hits, trks, etc. are not decoded
`CR1_INFOOFF`	Suppress writing all info msgs to exception pipe
`CR1_IGNORE_ID`	Ignore the ROD TTCrx data for checking BCID and L1D consistency; use that from first slave board received instead
`CR1_TTYPE_INCL`	Include Trigger type from TTCrx
`CR1_SLNK_SVC`	Enable SVC on S-link down
`CR1_SLNK_TSTCONT`	If set, S-link test runs continuously, else once
For debugging	
`CR1_FMTENA`	Enable building formatted events for output
`CR1_SLNK_FORCE`	If set, write to S-link without checking link ready
`CR1_TEST_RXCMD`	Send an RX getstatus command
`CR1_JUMBO_PKTS`	Enable jumbo packets for gigabit interface
`CR1_TEST_INT`	Test interrupt by faking exception pipe svc
`CR1_DIAGNOSE`	Allow diagnostic state, which forces flushed output to be written to raw data pipe
`CR1_TSTEVS`	Send test events, number according to TSTNUM reg, reset by ROD when done
`CR1_FAKE_L1A`	Fake L1A internally to ROD FPGA, at 40MHz / 512 = 78kHz
`CR1_TESTPULSE`	Write out hits in test pulse mode format: 8 bit map per input cell
Set by host program logic	
`CR1_OUTLENA`	Enable output S-link
`CR1_TTCENA`	Enable TTC and Event ID FIFO
`CR1_ASD_HSTENA`	Enable histogramming of ASD occupancy
`CR1_GLOBAL`	Place holder for the bit that signals global control of this register. Not used by FPGA

Table 11 Status Register 1: SR1 This register is read only

Width	Field name	Meaning
16	SR1_PENDING	SVC pending flag for each SVC
4	SR1_L1IDXPCT	Current Level-1 ID expected (lo 4 bits)
1	SR1_SLNK_LFF	Output S-link full flag
1	SR1_SVCPEND	SVC request waiting for acknowledge
1	SR1_SLNK_LDOWN	Output S-link is down
1	SR1_SLNK_TST	S-link test in progress
1	SR1_SLNK_RST	S-link reset in progress
1	SR1_FRAGWAIT	Waiting for first (non-dropped) fragment
1	SR1_TTYPEWAIT	Waiting for trigger type
1	SR1_TTCrxREADY	TTCrx on TTCrms mezz board is ready
1	SR1_TRIGWAIT	Waiting for trigger
1	SR1_DONEWAIT	Waiting for all LDBs to be processed
1	SR1_HSTRY_ENA	Recording in history buffer is enabled; set inactive by significant event in ROD logic

Table 12 Command Register 1 CMR1. This register is write only.

Bit name	Meaning
CMR1_CLRASD	Clear ASD histogram
CMR1_ENA_HSTRY	Enable writing to history buffer
CMR1_SLNK_RST	Do S-link reset
CMR1_SLNK_TST	Do S-link test, reset by FPGA at end of test, if test once
CMR1_ORBITCNT	Clear orbit counter
CMR1_FAKE_L1A	Generate a fake L1 trigger

8 References

Note In Acrobat Reader, click on URLs.

1 Endcap Muon trigger system: Read-out Driver Requirements and Specification, http://cern.ch/atlas-tgc/doc/ROD_PDR.pdf

2 Atlas First Level Trigger Technical Design Report, CERN / LHCC 98-14, ATLAS TDR 12, 30 June 1998, see Chapter 12.6, http://atlasinfo.cern.ch/Atlas/GROUPS/DAQTRIG/TDR/tdr.html

3 Endcap Muon Level-1 Trigger – TDR update, June 2000, see Chapter 12.6, http://cern.ch/atlas-tgc/doc/lvl1muonTDRupdate.pdf

4 The TGC ROD Crate web page, http://atlas-tgc.web.cern.ch/atlas-tgc/ROD/TGCROD.html

5 Endcap Muon Trigger System: Front End data rates and ROD input buffer simulation, http://cern.ch/atlas-tgc/doc/FEbuffering.pdf

6 The TGC ROD, LECC2002 workshop Colmar, September 2002,
paper: http://atlas-tdaqtalks.web.cern.ch/ATLAS-TDAQtalks/colmar_papers/B48.pdf
slides: http://atlas-tdaqtalks.web.cern.ch/ATLAS-TDAQtalks/colmar_papers/B48_presentation.pdf
ATLAS ROD Workshop, Annecy, November 2002, slides:
http://agenda.cern.ch/askArchive.php?base=agenda&categ=a021794&id=a021794s4t4/transparencies

7 TGC ROD Performance, http://atlas-proj-tgc.web.cern.ch/atlas-proj-tgc/ROD/RODperf.pdf, http://atlas-proj-tgc.web.cern.ch/atlas-proj-tgc/ROD/RODperf_Jun02.ppt

8 Endcap Muon Trigger System: ROD output format to the ROB, http://cern.ch/atlas-tgc/doc/ROBformat.pdf

9 The event format in the ATLAS DAQ prototype-1
http://atddoc.cern.ch/Atlas/postscript/Note050.ps
also ATLAS Note: ATL-DAQ-98-129.

10 The Xilinx Virtex-II series of FPGAs, see: http://www.xilinx.com/

11 Timing, Trigger and Control (TTC) Systems for LHC Detectors, http://ttc.web.cern.ch/TTC/intro.html

12 Agilent Technologies, HDMP-1034 gigabit receiver, (G-link), http://www.agilent.com/view/physical

13 Stratos M2R-25-4-1-TR Optical Gigabit Ethernet / Fibre Channel 850nm SFF 2x5 Dual Receivers – 1.25 / 1.0625GBaud. This part is now obsolete and has been replaced by another. http://www.stratosoptical.com/

14 Infineon Technologies Small Form Factor (SFF) transceiver with LC connector (AC-coupled) V23818 K305 L57, http://www.infineon.com/cmc_upload/0/000/012/561/1dot0625_GBd_Fiber_Channel_1dot3_Gigabit_Ethernet.pdf

15 Silicon serial number, Dallas Semiconductor Inc., part DS2401, http:www.dalsemi.com/DocControl/PDFs/2401.pdf

16 Virtex-II SelectLink Communications Channel, http://www.xilinx.com/bvdocs/appnotes/xapp263.pdf

17 CERN S-Link Home page: http://hsi.web.cern.ch/HSI/s-link/

18 1-Gigabit Ethernet MAC Core with PCS / PMA Sublayers (1000BASE-X) or GMII, v4.0, DS200, http://www.xilinx.com/ipcenter/catalog/logicore/docs/gig_eth_mac.pdf

19 IDT 18Mbit Ternary CAM, http://www.idt.com/products/pages/Network_Search_Engines-PL116_Sub330_Dev558.html

20 IDT ZBT SRAM 512×36 bits, http://www.idt.com/products/pages/ZBT-71T75702.html

21 Use of a CAM for hit pattern processing in the TGC ROD, in preparation.

22 Increased Current Level For 96 Pin & 160 Pin DIN / IEC Connector Draft Standard, ANSI / VITA 1.7-2003, http://www.vita.com/vso/draftstd/Vita1-7R5.pdf

23 Endcap Muon trigger system: ROD error messages, http://atlas-proj-tgc.web.cern.ch/atlas-proj-tgc/ROD/RODerrmsgs.pdf