

## Project Specification

**Project Name: ATLAS Level-1 Calorimeter Trigger-  
Read-out Driver**

**Version: 1.2.2**

**13 May 2008**

<b>Approval</b>	<b>Name</b>	<b>Signature</b>	<b>Date</b>
<b>Project Manager</b>	<b>V. J. O Perera</b>		
<b>Customer</b>	<b>C. N. P Gee</b>		

**Authors: B. M Barnett, C. N. P. Gee, V. J. O. Perera**

**Distribution for all updates:**

<b>Project Manager:</b>	<b>V. J. O. Perera</b>
<b>Customer:</b>	<b>C. N. P. Gee</b>
<b>Group Leader responsible for Project:</b>	<b>R. N. J. Halsall</b>
<b>Project Design Engineers</b>	<b>I. P. Brawn</b>
	<b>W. Qian</b>
<b>Project Managers of related projects:</b>	<b>B. M. Barnett</b>
	<b>R. Staley</b>
	<b>P. Hanke</b>
	<b>U. Schäfer</b>
	<b>S. Silverstein</b>
<b>Account Manager:</b>	<b>S.P.H. Quinton</b>
<b>Level-1 Calorimeter Trigger Project manager</b>	<b>E. Eisenhandler</b>

**This document will evolve as the detailed design advances.**

## Contents

<b>1</b>	<b>Introduction</b>	<b>10</b>
1.1	Data transmission overview	10
1.2	ROD data handling	11
1.3	Data sources and content	12
1.4	Management of output S-Links	12
1.5	Data-rate limitation	12
1.6	Organisation of ROD Modules	13
<b>2</b>	<b>Technical Aspects of the ROD Module</b>	<b>14</b>
2.1	Functional Requirements	15
2.2	Readout Process	16
2.2.1	DAQ readout (slice) data	17
2.2.2	Single- and Multiple-timeslice data	18
2.2.3	Format of S-Link Event Fragments	18
2.2.4	G-Link Timeout	20
2.2.5	G-Link DAV Gap	20
2.2.6	Trigger Type Timeout	21
2.2.7	Handling S-Link Link Full	21
2.2.8	Handling ROD Busy	21
2.2.9	Event Counter and Event Counter Reset	21
2.2.10	RoI Data	22
2.3	Data Transfer Rates in the ROD	22
2.3.1	G-Link input data rates	22
2.3.2	Maximum Trigger Rates	23
2.3.3	S-Link output data rates	24
2.4	S-Link Mapping	24
2.4.1	RoI Data from CPM, JEM, and CMMs	25
2.4.2	DAQ Data from PreProcessor, CP and JEP	26
2.5	Zero suppression and compression	27
2.5.1	Zero suppression	28
2.5.2	Compression	28
2.6	Data Replay Mode	28
<b>3</b>	<b>Functional blocks</b>	<b>29</b>
3.1	Optical Inputs	29
3.2	G-Link	29
3.3	Input FPGA	29
3.4	Switch FPGA	29
3.4.1	Event Selection and Sampling	30
3.4.2	TTC-correlated Event Selection and Sampling	31
3.4.3	Bunch Crossing Number Buffer	31
3.4.4	Event Number Buffer	31
3.4.5	Trigger Type Buffer	31
3.4.6	Orbit Number Buffer	31
3.4.7	Status Words	32
3.4.8	Status Count	32
3.4.9	Data Count	32
3.4.10	Checkpoint Handling	32
3.5	S-Link	33
3.5.1	S-Link Hardware Interface	33
3.6	ROD S-Link Packet	34

3.6.1	Format Version Numbers	34
3.6.2	Module Identifiers	35
3.6.3	Subdetector Identifier	35
3.6.4	Run Number and Run Type	35
3.6.5	Detector-Specific Event Type	35
3.6.6	User Header	35
3.7	Calibration and Monitoring	36
3.7.1	Monitoring FPGA	36
3.7.2	Instruction RAM	36
3.7.3	Dual-Port RAM	37
3.7.4	PCI Interface	37
3.7.5	Auxiliary Front-Panel Output	37
3.8	VME Interface	37
3.8.1	Interrupt Capability	39
3.9	TTCrx Decoder Card	40
3.10	CANBus	40
3.11	Reset, Reload, and Module Initialisation	40
3.12	System ACE	40
3.13	Using Replay Mode	41
3.13.1	Input FPGA Settings for Replay	41
3.13.2	Switch FPGA Settings for replay	42
3.14	Motherboard	42
3.14.1	Signal Handling on the Motherboard	42
3.14.2	Front Panel Input and Outputs	43
3.14.3	Front Panel Monitoring (LEDs)	43
3.14.4	Rear Panel Input and Outputs	44
3.14.5	Test and Set-up Points	44
3.14.6	Ground Points	44
3.14.7	Power Requirements from VME	44
3.14.8	Crate Power Requirements	44
<b>4</b>	<b>Programming Model</b>	<b>45</b>
4.1	Guidelines	45
4.2	Notation	45
4.3	Addressing Modes	45
4.4	Memory Map for CR/CSR space	46
4.4.1	VME64-Defined Configuration ROM	46
4.4.2	VME64x Defined Configuration ROM	47
4.4.3	VME64 Defined CSRs	47
4.4.4	VME64x Defined CSRs	47
4.4.5	User Control & Status Registers (CSRs)	47
4.5	Memory Map for Normal VME space	48
4.6	Motherboard Register Map	49
4.6.1	Module Number Register	50
4.6.2	Control Mode Register	50
4.6.3	Control Pulse Register 1	51
4.6.4	Control Pulse Register 2	52
4.6.5	Addressing Control Register	52
4.6.6	Status Register 1	53
4.6.7	Status Register 2	54
4.6.8	Interrupt Register	55
4.6.9	VME CPLD firmware revision no	55

4.6.10	VME FPGA firmware revision no.	55
4.6.11	CAN Access Register A	56
4.6.12	CAN Access Register B	56
4.6.13	TTCrx Control Register	56
4.6.14	TTCrx Status Register	57
4.6.15	TTCI2cId Register	57
4.6.16	TTC Broadcast Data Register	58
4.6.17	TTC DQ Register	58
4.6.18	TTC Dump RAM	58
4.7	Input FPGA Register Map	59
4.7.1	Input FPGA Common Version	60
4.7.2	Input FPGA Status Register	60
4.7.3	Input G-Link Error Map	61
4.7.4	Input G-Link Error Count	62
4.7.5	Input Control Pulse Register	62
4.7.6	Input Data FIFO Busy Depth	63
4.7.7	Input Event Management FIFO Busy Depth	63
4.7.8	Input BCN FIFO Busy Depth	63
4.7.9	Input Channel Version Number	64
4.7.10	Input Channel Control Register	64
4.7.11	Input Channel Status Register	65
4.7.12	Input Channel Expected Slice Count	65
4.7.13	Input Channel Subslice Control Register	66
4.7.14	Input Channel Compression Control Register	66
4.7.15	Input Channel Source ID Register	67
4.7.16	Input Channel Neutral ID	67
4.7.17	Input Channel Event Management FIFO address pointers	68
4.7.18	Input Channel Bunch Number FIFO address pointers	68
4.7.19	Input Channel Data FIFO address pointers	68
4.7.20	Input Channel FIFO Depth Monitor Registers	69
4.7.21	Input Channel Event Management FIFO Access	69
4.7.22	Input Channel Bunch Number FIFO Access	69
4.7.23	Input Channel Data FIFO Access	69
4.8	Switch FPGA Register Map	70
4.8.1	Switch FPGA Revision	72
4.8.2	Switch Control Register	72
4.8.3	Switch FPGA Pulse Register	73
4.8.4	Switch FPGA Status Register	74
4.8.5	Run Number Register	75
4.8.6	ECR Count Register	76
4.8.7	Replay BCN Register	76
4.8.8	Replay Event Number Register	76
4.8.9	Replay Trigger Type Register	76
4.8.10	TTC SampleId Register	76
4.8.11	RoI Wordlimit Register	77
4.8.12	S-Link Control Register	77
4.8.13	S-Link Status Register	78
4.8.14	S-Link Routing	78
4.8.15	Sampling Control	79
4.8.16	Spy Status Register	79
4.8.17	Trigger Type Timeout Register	79

4.8.18	TTC FIFO Busy Depth.....	80
4.8.19	BCN FIFO Pointer Register .....	80
4.8.20	G-Link Timeout Register .....	80
4.8.21	Total L1A Number .....	80
4.8.22	Last Event L1ID Register.....	80
4.8.23	Last TTC L1ID Register .....	80
4.8.24	Detector Event Type Register .....	80
4.8.25	Orbit Counter Register .....	81
4.8.26	Orbit Counter Wrap Value.....	81
4.8.27	ECR Delay Register .....	81
4.8.28	ECR Error Counter .....	81
4.8.29	Format Version Register .....	81
4.8.30	User Header Register 1 .....	82
4.8.31	User Header Registers 2-4 .....	82
4.8.32	TrigType Buffer .....	82
4.8.33	BCN_Buffer .....	82
4.8.34	Event_Number Buffer.....	82
4.8.35	Orbit Number Buffer.....	83
4.8.36	S-Link 0-3 LFF time .....	83
4.8.37	S-Link 0-3 Source ID.....	83
4.8.38	S-Link 0-3 Spy Event Length 01.....	83
4.8.39	S-Link 0-3 Spy Event Length 23.....	84
4.8.40	S-Link 0-3 Spy Buffers .....	84
4.9	Monitoring FPGA Register Map .....	85
4.9.1	Monitor FPGA Revision.....	85
<b>5</b>	<b>Details of DAQ and RoI data and formats.....</b>	<b>86</b>
5.1	General guidelines .....	86
5.2	Overall Sub-block format .....	88
5.3	Neutral S-Link format .....	91
5.3.1	Output S-Link DAQ data Format (Fmt=0) .....	91
5.4	PreProcessor DAQ data .....	92
5.4.1	Input G-Link DAQ data format.....	92
5.4.2	General Structure of PPM S-Link Data. ....	95
5.4.3	Uncompressed S-Link DAQ output format. (Fmt = 1).....	95
5.4.4	Compressed S-Link DAQ output format. (Fmt = 2).....	97
5.4.5	Super-Compressed S-Link DAQ output format (Fmt=3) .....	97
5.5	CPM DAQ Data .....	98
5.5.1	Input G-Link DAQ data format.....	98
5.5.2	Output S-Link DAQ data format (Fmt=1) .....	98
5.6	JEM DAQ Data .....	100
5.6.1	Input G-Link DAQ data format.....	100
5.6.2	Output S-Link DAQ data format (Fmt=1) .....	100
5.7	CMM-CP DAQ .....	102
5.7.1	Input G-Link DAQ data format.....	102
5.7.2	Output S-Link DAQ data Format (Fmt=1) .....	103
5.8	CMM-Jet DAQ.....	103
5.8.1	Input G-Link Data format .....	103
5.8.2	Output S-Link DAQ data Format.....	104
5.9	CMM-Energy DAQ .....	105
5.9.1	Input G-Link DAQ Data format.....	105
5.9.2	Output S-Link DAQ data format .....	107

5.10	General features of RoI Formats .....	109
5.11	CPM RoI Data .....	109
5.11.1	Input G-Link RoI data format .....	109
5.11.2	Output S-Link RoI format .....	110
5.12	JEM Jet RoI .....	111
5.12.1	Input G-Link RoI data format .....	111
5.12.2	Output S-Link RoI data format .....	111
5.13	CMM-Jet- $E_T$ RoI .....	112
5.13.1	Input G-Link Data format .....	112
5.13.2	Output S-Link RoI data Format .....	112
5.14	CMM-Energy RoI .....	112
5.14.1	Input G-Link Data format .....	112
5.14.2	Output S-Link RoI data Format .....	112
5.15	S-Link Status word formats .....	112
<b>6</b>	<b>Manufacture, Testing and Maintenance .....</b>	<b>114</b>
6.1	Manufacturing .....	114
6.2	Testing .....	114
6.2.1	Test Strategy .....	114
6.2.2	Test equipment .....	114
6.3	Software .....	114
6.4	Installation .....	114
6.5	Maintenance and further orders .....	114
<b>7</b>	<b>Project Management .....</b>	<b>115</b>
7.1	Personnel .....	115
7.2	Project Responsibilities .....	115
7.3	Deliverables .....	115
7.4	Project plan (Milestones) .....	115
7.5	Design Reviews .....	116
7.6	Training .....	116
7.7	CAE .....	116
7.8	Costs and finance .....	116
7.9	Intellectual Property Rights (IPR) and Confidentiality .....	116
7.10	Safety .....	116
7.11	Environmental impact .....	116
7.11.1	Disposal .....	116
7.11.2	EMC .....	116
7.12	Handling Precautions .....	116
<b>8</b>	<b>References .....</b>	<b>117</b>
<b>9</b>	<b>Update History .....</b>	<b>118</b>
9.1	Version 1.0 (Post-review) .....	118
9.2	Version 1.05 .....	118
9.3	Version 1.06 .....	119
9.4	Version 1.07 .....	119
9.5	Version 1.08 .....	119
9.6	Version 1.09 .....	119
9.7	Version 1.1 .....	120
9.8	Version 1.2 .....	120
9.9	Version 1.2.1 .....	120
9.10	Version 1.2.2 .....	120
<b>Appendix A</b>	<b>S-Link rear transition module Connector Pin-out .....</b>	<b>121</b>
A.1	5-row P2 connector map to two S-LINK LSCs .....	121

A.2 5-row P3 connector map to two S-LINK LSCs.....	122
<b>Appendix B</b> <b>VME 64x J0 Connector Pin Assignment.....</b>	<b>123</b>
<b>Appendix C</b> <b>Stiffening and ESD strip details for 9U ROD .....</b>	<b>124</b>

## List of Figures

Figure 1: Overview of a typical data readout scheme .....	10
Figure 2: Use of DAV* to frame valid G-Link data. ....	11
Figure 3: Organisation of ROD input links and crates, showing one of two ROD crates. ....	14
Figure 4: ROD module with S-Link rear transition module .....	14
Figure 5: ROD Block Diagram .....	16
Figure 6: Overall structure of event fragment payload (CPM, JEM, and CMM data) .....	19
Figure 7: Overall structure of event fragment payload (PPM data) .....	19
Figure 8: Timing sequence of signals related to Event Counter Reset .....	22
Figure 9: RoI S-Link mapping, showing active (solid) and unused (dashed) data paths. ....	25
Figure 10: DAQ data S-Link mapping for JEP crates. ....	26
Figure 11: Rear S-Link Transition Module. ....	33
Figure 12: Format of User Header .....	36
Figure 13: Overall S-Link Sub-Block format. ....	88
Figure 14: Neutral S-Link format. ....	91
Figure 15: Bitstream format from one PPM ASIC channel .....	92
Figure 16: PPM Input G-Link data format .....	94
Figure 17: Uncompressed PPM S-Link output DAQ format .....	96
Figure 18: Uncompressed PPM S-Link output DAQ format - error block. ....	96
Figure 19: PPM Compressed S-Link DAQ data format .....	97
Figure 20: CPM G-Link DAQ data format .....	98
Figure 21: CPM S-Link DAQ data format .....	99
Figure 22: JEM G-Link DAQ data format (16-bit G-Link mode). ....	100
Figure 23: JEM S-Link DAQ data format .....	101
Figure 24: CMM-CP G-Link DAQ data format .....	102
Figure 25: CMM-CP S-Link DAQ data format .....	103
Figure 26: CMM-Jet input G-Link DAQ format .....	104
Figure 27: CMM-Jet output S-Link data format. ....	105
Figure 28: CMM-Energy G-Link DAQ input format .....	106
Figure 29: CMM-Energy Output S-Link DAQ data format .....	108
Figure 30: CPM G-Link RoI data format .....	109
Figure 31: S-Link RoI data formats .....	110
Figure 32: Jet RoI G-Link format (16-bit G-Link mode). ....	111
Figure 33: Format of S-Link status words. ....	113
Figure 34: Stiffening and ESD strip details for 9U ROD. ....	124



## List of Tables

Table 1: Numbers of RODs and S-Links .....	13
Table 2: ROD data sources and firmware types.....	17
Table 3: G-Link data rates into RODs .....	23
Table 4: Estimated S-Link Data Rates from each ROD .....	24
Table 5: G-Link use for RoI Data .....	26
Table 6: G-Link use for DAQ data .....	27
Table 7: ATLAS ROD Event Fragment, Version 3.0.....	34
Table 8: Minor Format Version Numbers.....	35
Table 9: Module identifier fields. ....	35
Table 10: ROD VME addresses and functions in both ROD crates.....	39
Table 11: System Ace standard firmware collections.....	41
Table 12: VME64x CR/CSR space .....	46
Table 13: Word-Id Values for Formatted Data.....	87
Table 14: Word-Id Values for Neutral Data .....	87
Table 15 : Sub-block header fields. ....	89
Table 16: Sub-status word fields.....	90

# 1 Introduction

This document describes the read-out driver (ROD) module for the ATLAS Level-1 Calorimeter Trigger processor. The specification is for the final production modules.

The Level-1 Calorimeter Trigger system [1] consists of a sequence of modules which generate the level-1 trigger bit patterns used by the Central Trigger Processor (CTP) [2] to form the level-1 trigger. In addition to forming the trigger, each stage of the electronics – PreProcessor Modules (PPMs) [3], Cluster Processor Modules (CPMs) [4], Jet/Energy Modules (JEMs) [5], and Common Merger Modules (CMMs) [6] – provides ancillary readout which can be used to check the trigger performance. These data are destined for the Readout Subsystem (ROS), a part of the ATLAS DAQ system, and are referred to in this document as “DAQ” data. Additionally, CPMs, JEMs, and some CMMs provide Region-of-Interest (RoI) information, which are sent via the RoI-Builder (RoIB) to the level-2 section of the higher-level trigger system. Copies of the RoI data are also sent, for checking, to the ROS.

The ROD module described in this document provides the data formatting, compression, buffering, and change of transmission standards needed to collect the DAQ and RoI data and forward them to the ROS and RoIB.

## 1.1 Data transmission overview

Data arrive at the RODs on optical links using a serial format generated by the G-Link chipset from Agilent [7]. The transmitting G-Link encodes 16 or 20 bits of user data into a 20 or 24-bit frame which it transmits serially at 800 or 960 Mbaud. The effective data transfer rate is 640 or 800 Mbit/s. The G-Link receiver recovers the clock and user data from the serial data stream, and also checks the framing bits to verify link stability. Where a transmitting module needs less than the full bit width of the G-Link, zero data are sent and the unused data pins may be grounded. On reception, data from unused pins are discarded, depending on the data format in use.

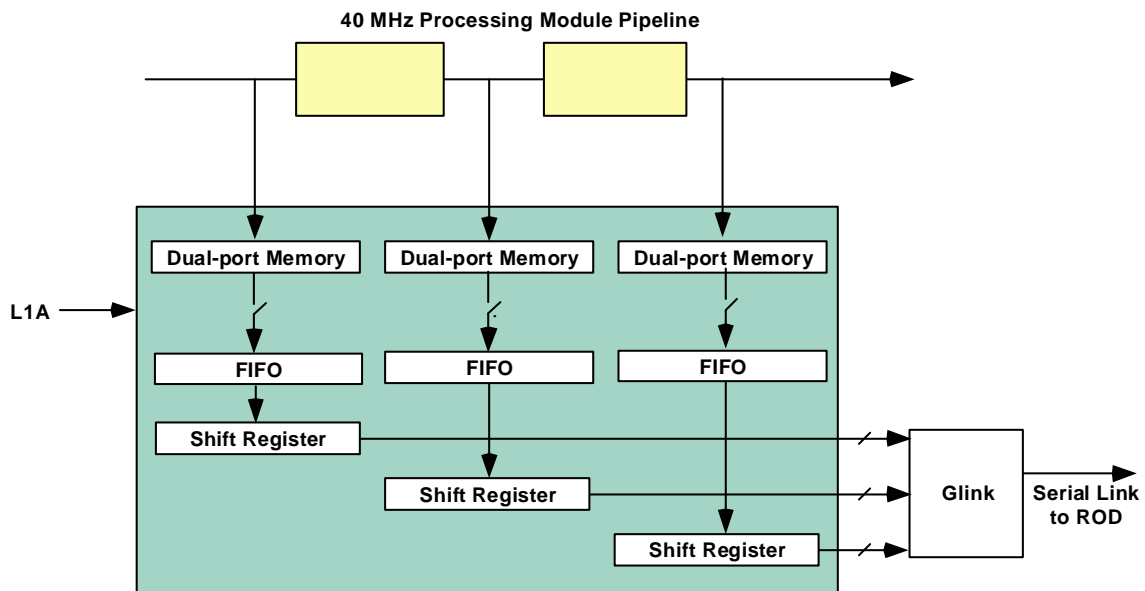


Figure 1: Overview of a typical data readout scheme

As illustrated in Figure 1, the readout process is initiated by the Level-1 Accept (L1A) signal generated by the CTP and distributed by the Timing, Trigger and Control (TTC) system [8]. All processing modules copy data from their 40.08 MHz pipelines into dual-port scrolling

memories. On receipt of an L1A signal, each module extracts data from its scrolling memories into FIFOs to await readout. Each FIFO is connected via a shift register to one of the G-Link user data pins. Logic on the module moves data from the FIFOs into the shift registers, asserts the Data Available (DAV\*) signal to the G-Link, and transmits the shift register contents for the event concerned. This typically involves sending one bit for each shift register on every G-Link clock edge. An odd parity bit is appended to each active G-Link pin when the shift register contents have been sent. The module finally deasserts DAV\* and G-Link returns to its quiescent state for at least one clock cycle. During quiescent periods, the transmitting G-Link sends fill frames (FF1) to maintain the lock between the transmitter and receiver.

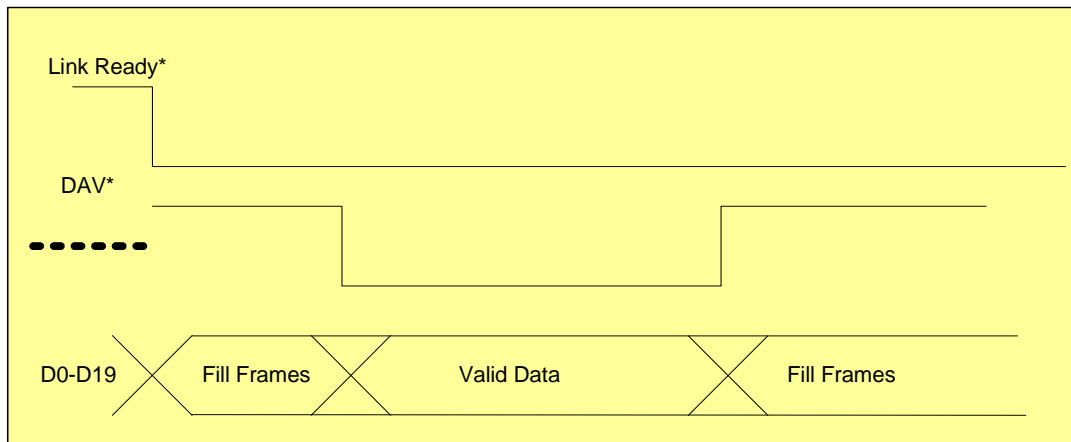


Figure 2: Use of DAV\* to frame valid G-Link data.

The DAV\* signal is used by the transmitting end to frame the serial data streams, and is recovered by the receiving G-Link device on the ROD. Data reception is initiated at the ROD end when the DAV\* becomes active. Up to 18 G-Links may be active on the ROD simultaneously receiving data. If for any reason one of the expected DAV\* signals is not received, that particular link times out (see section 2.2.4) and data from the other links are sent on to the DAQ or RoIB as normal.

When the link is down, the output DAV\* signal from the receiving G-Link oscillates as it tries to establish lock. To avoid confusion in the logic, the received DAV\* is qualified by the Link Ready\* signal before being used to extract the data from the G-Link.

## 1.2 ROD data handling

In the ROD, the transition from quiescent to active state of the receiving G-Link is signalled by the assertion of the receiver DAV\* signal. The G-Link receiver presents the recovered 40 MHz clock and parallel 16- or 20-bit user data at its output. Firmware in the ROD collects, processes and stores the data in 32-bit-wide FIFOs with the format of the output event record. The conversion process includes checks of parity and Bunch-Crossing Number (BCN) as well as zero-suppression or data compression. The ROD also receives the L1A signal via the TTC subsystem, shortly followed by the event trigger type needed in the ATLAS ROD fragment header. When all required information has been received from the TTC and G-Links, the ROD assembles a complete ATLAS event fragment with header, payload from G-Links, and trailer, and sends the completed fragment to the ATLAS readout subsystem.

### 1.3 Data sources and content

In normal use, each ROD processes either DAQ data only or RoI data only. The data input, processing, and output are significantly different in the two cases.

The DAQ readout data are intended for trigger monitoring and calibration, and include information collected near to the processing module inputs and outputs. These data, extracted parasitically from the 40.08 MHz pipelines, are associated in time with particular clock cycles and therefore with particular LHC bunch crossings, and are referred to as “timeslices”. All processing modules can sample a programmable number of consecutive timeslices. The minimum spacing between L1As means that a maximum of five timeslices can be read out in normal physics running, although longer readout is needed for calibration with the PPM. The time offset of the readout relative to the L1A may be programmed via registers in the modules. When all modules are properly configured, the data derived from a specified LHC bunch crossing may be tracked through the complete trigger system. Readout of multiple timeslices is useful in setting up and checking the internal digital timing of the trigger, and is essential to verify the correct operation of bunch crossing identification with the long calorimeter pulses.

RoI data are read out only for the single timeslice corresponding to the L1A. They provide the Level-2 trigger with information about the spatial position of e.m. and hadronic clusters and jets, and total and missing transverse energy measured in the calorimeters.

Full details of all data content and formats may be found in section 5.

### 1.4 Management of output S-Links

The ROD module is equipped with four S-Links to transmit the output data to the ROS and RoIB. The S-Link specification [9] defines a FIFO-like electrical interface providing data transport at up to 160 Mbyte/s per link, with flow control, status, and error detection signals. Several physical implementations of the interface exist.

The volume of RoI data handled by a ROD is typically a few percent of the S-Link bandwidth. A single S-Link carries the RoI data from an RoI ROD to the RoIB, and a second S-Link carries a copy to the ROS. The remaining two links are unused on RoI RODs.

For a ROD handling DAQ data, the data volume depends on the L1A rate, the occupancy of the detector (hence the LHC luminosity and background), the number of timeslices being read out, and the type and effectiveness of the zero-suppression or data compression algorithms. Some of these parameters are hard to predict, so the ROD must include configuration options to use from one to four S-Links to despatch the output data. All DAQ S-Links go only to the ROS. At high data rates, the input G-Link data volume may temporarily exceed the bandwidth of the S-Links, and buffers are provided to smooth the data flow.

### 1.5 Data-rate limitation

Limits on readout data rates are imposed by the G-Links between the trigger modules and RODs, and by the S-Links between the RODs and the ROS. It is also desirable to reduce the volume of data read out to DAQ in order to minimise the mass storage required for ATLAS events.

The number of G-Links handling DAQ data is kept to one per trigger module in order to minimise the power consumption per module, the number of RODs, and the system cost. (The number of RoI G-Links never needs to be more than one per module.) Once the full DAQ data rate can be handled by one G-Link per module, further data reduction can be

carried out by the RODs. The amount of data read out from each trigger module can be changed only by varying the number of timeslices of DAQ data that are read out, controlled by registers on the trigger modules. All modules of the same type should normally be set to read out the same number of timeslices. Except for the PPMs, the number of timeslices to be read out from a particular type of module is the same for all the different classes of data read out from a module. The number of timeslices of the two types of PPM data can be separately controlled, as the PPM cannot read out five timeslices of both FADC and LUT data at the full L1A rate of 100 kHz. At the maximum 100 kHz L1A rate, up to a total of eight timeslices (e.g. 5 FADC and 3 LUT) can be read. For most modules, more data can be read out at lower L1A rates.

Several tools are available to keep data rates within acceptable limits. Firstly, the number of timeslices read out from the trigger modules can be controlled. Secondly, the ROD can reduce the number of timeslices of each type of data independently. This may include entirely eliminating intermediate data that are not needed, especially when nominally equal to other data, e.g. at the two ends of data links in the trigger. Thirdly, data containing a large number of zeroes can be reduced in volume without loss of information by zero suppression. Finally, where some data values are much more frequent than others (e.g. in FADC data), loss-less data compression can be used to reduce data volume.

## 1.6 Organisation of ROD Modules

The four Cluster Processor crates each provide trigger signals covering one  $\phi$ -quadrant of the ATLAS detector, while the two Jet/Energy-Sum Processor crates each cover two  $\phi$ -quadrants. The eight PreProcessor crates are organised according to the topology of the different calorimeters. Twenty RODs are needed for the complete calorimeter trigger, enough to require two VME crates. Each ROD collects data from all the modules in one subsystem crate, so it is possible to carry most of the  $\phi$ -quadrant architecture into the two ROD crates, allowing the single-board computers in those crates each to access all the information from two quadrants (see [Figure 3](#)). In both ROD crates, this layout provides spare slots adjacent to DAQ RODs, and one empty slot adjacent to RoI RODs, so that expansion of the number of RODs is possible without completely reworking the connection database or adding another ROD crate.

In common with other trigger modules, the RODs determine their function by reading the geographical address of their VME slot, although this mechanism can be overridden for testing. Details of geographical addresses and functions are given in [section 3.8](#).

Each ROD can drive up to four output S-Links, depending on its position in the system. The total number of RODs and S-Links is given in [Table 1](#). G-Link and S-Link cabling details are defined in [\[10\]](#).

Subsystem	No of RODs	Links per ROD		Total Links	
		to DAQ	to RoIB	to DAQ	to RoIB
PreProcessor	8	4	0	32	0
CP DAQ	4	2	0	8	0
CP RoI	4	1	1	4	4
JEP DAQ	2	4	0	8	0
JEP RoI	2	1	1	2	2
<b>Totals</b>	<b>20</b>			<b>54</b>	<b>6</b>

Table 1: Numbers of RODs and S-Links

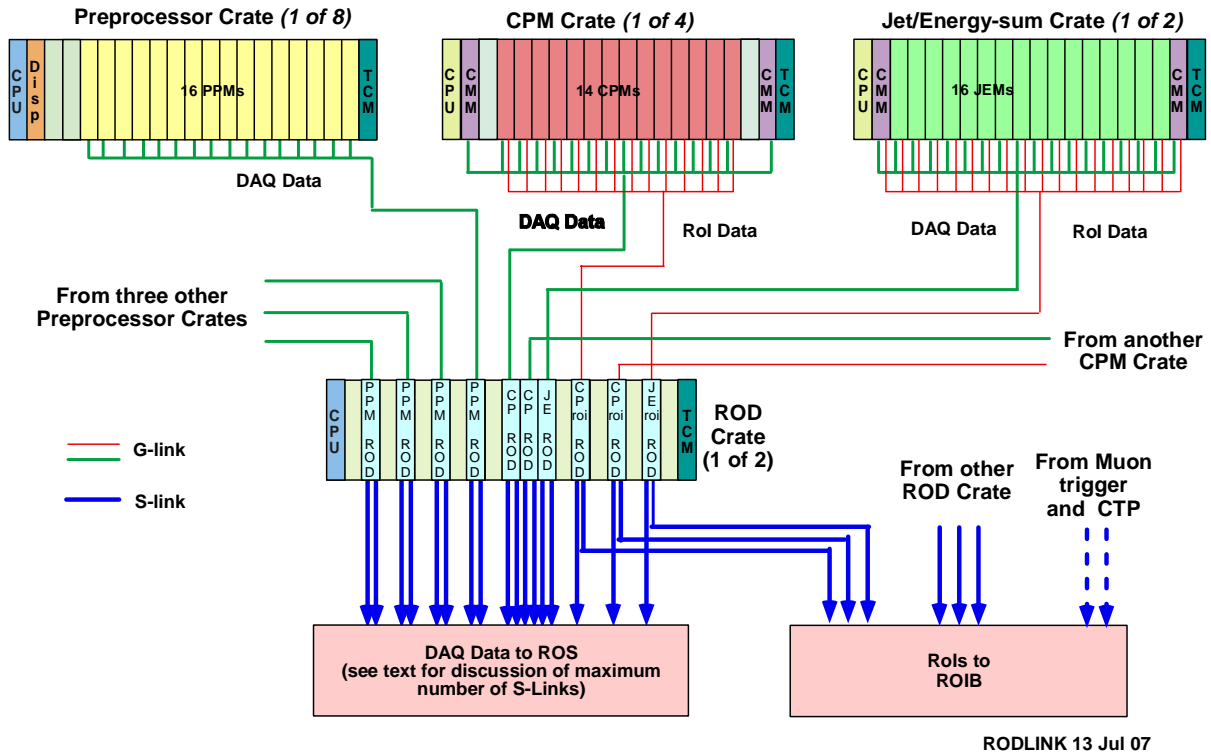
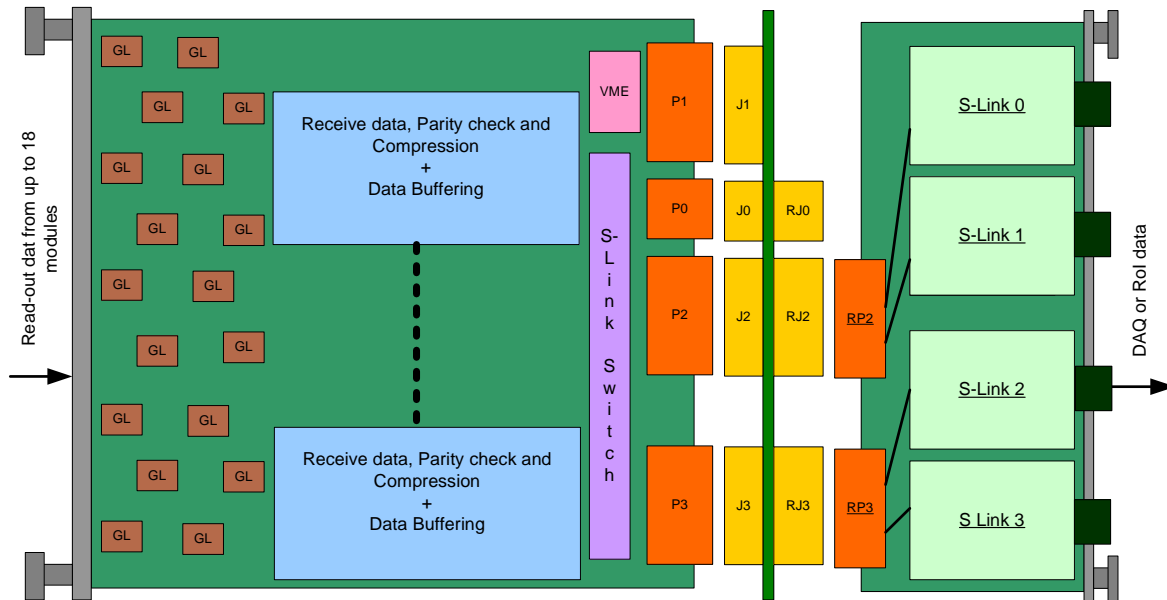


Figure 3: Organisation of ROD input links and crates, showing one of two ROD crates.

## 2 Technical Aspects of the ROD Module

The ROD module is a VME64x [11] slave conforming to IEEE1101.10 [12], 9U x 400 mm form factor with on-board G-Link devices interfacing to the processing modules (PPMs, CPMs, JEMs and CMMs). The interface to the ROS and RoIB is via S-Link cards hosted on a VME rear transition module similar to the CERN S2VME64x, as shown in Figure 4.



RodOverview 14Oct06.vsd

Figure 4: ROD module with S-Link rear transition module

## 2.1 Functional Requirements

The purpose of the ROD module is to collect data from the PreProcessor Modules (DAQ), Cluster Processor Modules (DAQ and RoI), Jet/Energy Modules (DAQ and RoI), and Common Merger Modules (DAQ and RoI), and send the formatted data on to the ROS and to RoIB. The ROD module provides the following functionality:

- a) Receive data sent optically at 18 x 800 Mbit/s using 18 G-Link Rx chips on board in 16- or 20-bit frame mode, with handling of link loss and transmission abnormalities.
- b) Compare the transmitted bunch numbers with the on-board TTCrx generated numbers. If the two do not match, set an error flag in the readout data.
- c) Perform parity check on the incoming G-Link data.
- d) Perform compression or zero suppression on data from each channel as required.
- e) Provide options to select reduced numbers of timeslices from different sections of source modules.
- f) Provide an option to bypass all formatting and compression logic and write a copy of the raw incoming data as output.
- g) Write formatted data (see section 5 for data formats) to the FIFO buffers (internal to the FPGAs)
- h) Receive TTC signals including clock, short broadcasts (BCR, ECR, and user bits), event number and event type, and construct event headers.
- i) Select data to be transferred to the ROS, on the basis of 'trigger type' received from the TTC.
- j) Transfer the data from header and FIFO buffers, followed by an event trailer, as ATLAS standard event fragments, to a rear transition module housing four S-Links.
- k) Flag G-Link data frames received with parity and link errors
- l) Provide separate event buffers to spy on the data sent to S-Links, with an option to selecting events according to event type.
- m) Provide access to the spy event buffers from the crate single-board computer via VME
- n) Provide a BUSY signal to the Central Trigger Processor via ROD Busy Modules [13].
- o) Process complete events at up to 100 KHz sustained rate.
- p) For RoI data only, terminate RoI data transfer if the number of RoIs is greater than a programmable limit, discarding any additional RoIs, and flag in status words.
- q) Support CANBus board temperature and voltage monitoring.

The ROD module processes many different data formats, and should be adaptable to the different requirements by loading the appropriate firmware under computer control.

Figure 5 shows a block diagram (not the planned implementation) of the module.

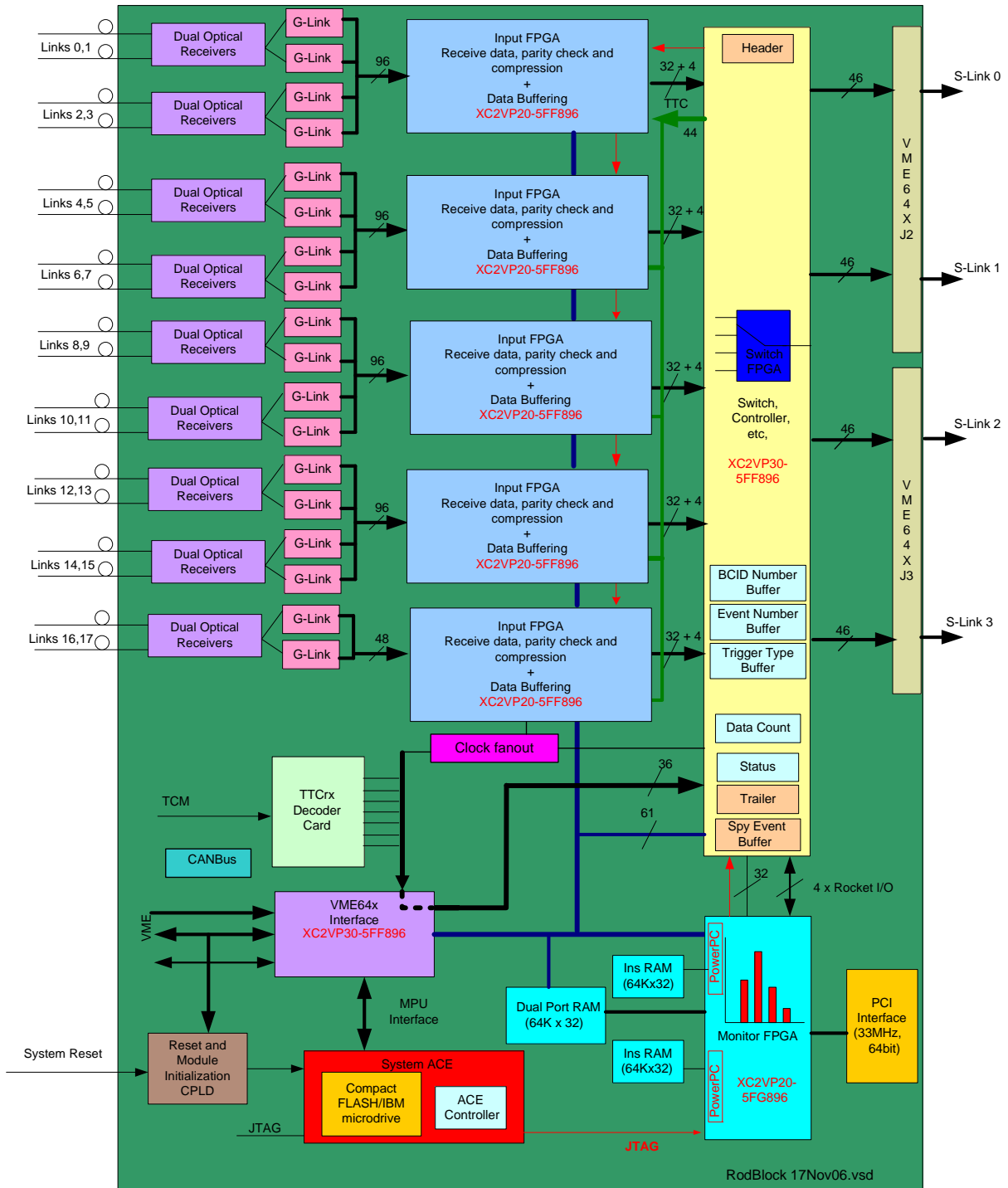


Figure 5: ROD Block Diagram

## 2.2 Readout Process

There are six types of DAQ input G-Link data and four of RoI input, as shown in Table 2 below. The data format for each source is described in the corresponding section. A neutral format is also provided to allow generic processing of data from any source for diagnostic use. Each data source has a corresponding firmware type. For completeness, the table also lists the other firmware types used in the ROD module. These are described later.



Firmware Type	Data Processing Type	Section
0	Neutral – all sources	<a href="#">5.3</a>
1	Pre-processor non-compressed DAQ	<a href="#">5.4</a>
2	CPM DAQ	<a href="#">5.5</a>
3	JEM DAQ	<a href="#">5.6</a>
4	CMM-CP DAQ	<a href="#">5.7</a>
5	CMM-Jet DAQ	<a href="#">5.8</a>
6	CMM-Energy DAQ	<a href="#">5.9</a>
7	CPM RoI	<a href="#">5.11</a>
8	JEM RoI	<a href="#">5.12</a>
9	CMM-Jet-E <sub>T</sub> RoI	<a href="#">5.13</a>
10	CMM-Energy RoI	<a href="#">5.14</a>
11	Pre-processor compressed DAQ	<a href="#">5.4</a>
12	CPLD	
13	Common	
14	Reserved for future Input FPGA code	
15	Reserved for future Input FPGA code	
16	VME FPGA	
17	Switch FPGA	
18	Monitor FPGA	

Table 2: ROD data sources and firmware types

### 2.2.1 DAQ readout (slice) data

Transfer of data from G-Links to S-Links involves both the Input and the Switch FPGAs.

G-Link receivers (one G-Link per source module) are attached to the Input FPGAs, four to FPGAs 0-3, and two to FPGA 4, as shown in [Figure 5](#). Each Input FPGA monitors the DAV\* signals from its G-Links, and starts receiving data when DAV\* goes active. The data are parity checked, zero-suppressed or compressed, formatted, and then stored in FIFO buffers to await readout. The amount of data received from each link is checked against the slice register.

When the Switch FPGA detects that all the Input FPGAs have completed processing for an event, it retrieves the event number and the Bunch-Crossing number from FIFOs, and places them in the header buffer FIFOs (one for each active S-Link) and then transfers the headers to the S-Links, followed by event data from the Input FPGA FIFOs, then followed by an event trailer. The number of S-Links and the data routing to the appropriate S-Link depends on the volume of data and type of data as described in [section 2.3](#).

The Switch FPGA also copies a selection of the data into the spy buffers. These data may subsequently be copied to the Monitor FPGA for analysis and/or transfer via the VME interface to a single board computer. In some cases (e.g. test run), event data may not be transferred out on the S-Link, but it is still required that data are written to the spy buffers.

When the Level-1 Accept (L1A) signal arrives, the Switch FPGA inserts the event number (ROD\_L1ID, counted in the ROD – see [2.2.9](#)) into its event number FIFO, and the Bunch-Crossing number (ROD\_BC, generated by the TTCrx) into its BC FIFO. These data are needed to build the S-Link header.

The Input FPGA puts another copy of ROD\_BC into its BC FIFO, which it uses to check the Bunch-Crossing number from the G-Link data. If there is a mismatch, a flag is set in the appropriate S-Link trailer and the lower six bits of the G-Link source Bunch-Crossing number are copied into the payload sub-status word. This condition should not occur if the system is in-step and operating correctly. On the first mismatch, and providing there were no parity errors, the controller records both Bunch-Crossing numbers in a register, together with the number of the G-Link involved, and sets a warning bit in the module status register. This information assists with problem diagnosis.

All the information needed to populate the main S-Link header comes from ROD registers. The source module identifiers needed to populate the sub-headers are assembled in the Input FPGAs from the SourceID registers. No crate or module number information is carried by the G-Links. The G-Link numbers for this calculation are listed in [Table 6](#).

### **2.2.2 Single- and Multiple-timeslice data**

All modules except the Preprocessor build the G-Link data frame by collecting data samples from their sequential processing stages, timed so that the data all relate to the same bunch crossing number. This tracks the time-evolution of the data through the algorithm – hence the term “timeslice”. All modules can collect a programmable number of consecutive timeslices in response to each L1A, these data being sent as a continuous G-Link transaction, with DAV\* de-asserted only after the last timeslice of each L1A. The ROD should be configured by software to expect the same number and should normally expect all modules of the same type to send the same number of timeslices.

Except for the Preprocessor, the number of timeslices collected from different parts of a processing module pipeline must be identical to keep the G-Link formats simple. However, the ROD is able to select a reduced number of timeslices for onward processing into S-Link data. Data so eliminated is used only to complete longitudinal G-Link parity checking, and thereafter all bits concerned are ignored, including any error indicators. This capacity is described in section [2.5](#).

In the case of the Preprocessor, the number of FADC and PPM\_LUT timeslices may be varied independently using a control register within the PPM ASIC, up to a maximum of seven timeslices of LUT data and 127 of FADC data. The ROD normally expects no more than 31 timeslices in total (a header format restriction). It may be possible for time-isolated events to contain more data for test purposes, depending on available FIFO capability.

The control logic checking the Bunch-Crossing number arriving in the G-Link data stream expects all timeslices to carry the same bunch-crossing number, that of the L1A.

### **2.2.3 Format of S-Link Event Fragments**

Each S-Link packet is built from four components, described in section [3.6](#). These are the ATLAS-standard ROD header, a data payload, a status block, and an ATLAS-standard ROD trailer.

The payload from each source module is constructed in an Input FPGA. It consists of a sequence of data blocks (each comprising sub-address, data and sub-status) for each active G-Link in turn (numbered g to h in the following figures). Within each G-Link data block, the data format depends on the data source. S-Link data from CPMs, JEMs and CMMs consist of a sequence a sub-blocks, one for each timeslice. This structure is illustrated in [Figure 6](#).

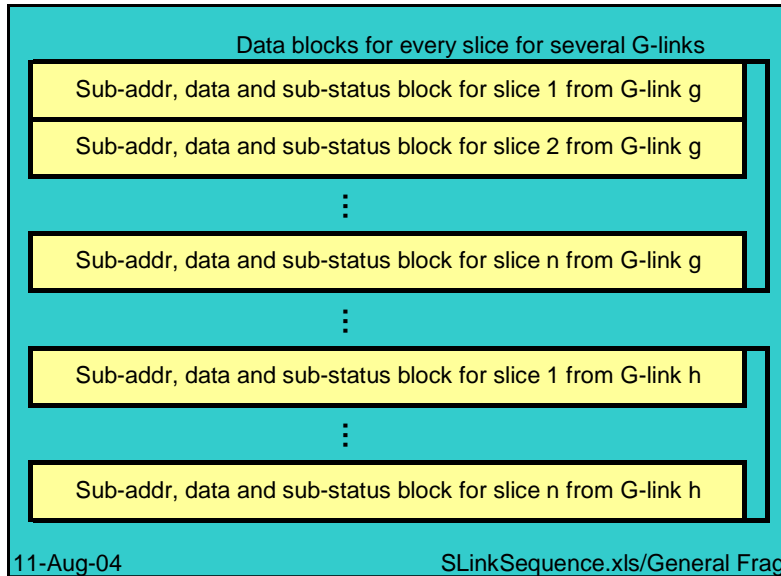


Figure 6: Overall structure of event fragment payload (CPM, JEM, and CMM data)

Data from the Preprocessor reflect the readout from the Preprocessor ASIC, and contain a sub-block for each channel (A-D) of each ASIC (0-15) per PPM. Each S-Link contains the data from an integral number of G-Links –usually four G-Links per S-Link. On each of the S-Links, the overall S-Link packet is as shown in Figure 7.

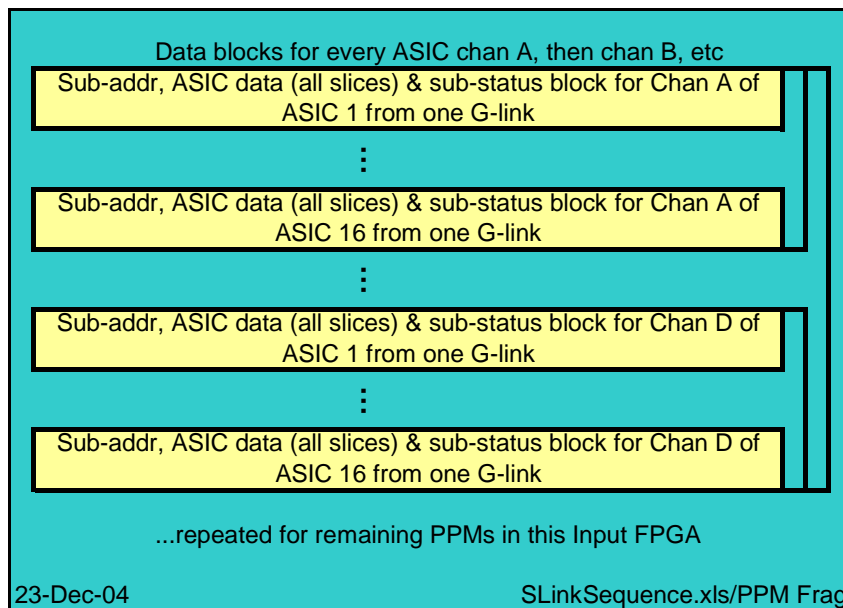


Figure 7: Overall structure of event fragment payload (PPM data)

The first word of each sub-block is a sub-address word. This includes a version number of the sub-block format, the slice number, and the source module crate and slot numbers. After the data part of the sub-block comes a sub-status word whose content includes error bits relating to the data of the sub-block. Errors are indicated by non-zero bits, so the sub-status may be zero-suppressed if no error bits are set.

The firmware responsible for each enabled G-Link always produces a sub-address word, even if all other data are eliminated from the sub-block by zero-suppression or compression. The sub-address is also produced if there is a timeout for a G-Link. Only if a G-Link is disabled by ROD control registers are no data or sub-status words produced.

To maintain the maximum throughput, the ROD should treat the active S-Links as separate channels. Event headers, data payload and event trailers should be transferred independently and concurrently to all active S-Links. Presence of Link Full Flag (LFF, i.e. Xoff) on one link does not prevent data transmission on other active links. There is an exception for RoI and DAQ-copy (diagnostic) modes, in which LFF on either the RoIB or ROS S-Links should suspend transmission on both. This is necessary since the two S-Links carry copies of the same data payload.

Memory-mapped VME access to the data buffers is needed for testing. Valid data lies between the read and write pointers, which should be read to ascertain the location and length of valid data.

#### **2.2.4 G-Link Timeout**

Cable and module faults may cause G-Link data not to arrive. A timeout mechanism is provided to identify and recover from such faults by generating a short status block for missing G-Link inputs, so that the rest of the readout can proceed unchanged. The Event Counter Reset signal (see section 2.2.9) ensures that the ROD recovers from the effect of temporary glitches.

In normal operation, the Switch FPGA waits for all Input FPGAs to signal that they have a formatted event sub-block available. Since the Input FPGA readout and formatting process is pipelined, formatted data from all active G-Links should be available within about 10 $\mu$ s after an L1A if the links were previously idle. Data transfer out of the Input FPGAs starts when they are all ready.

The timeout logic operates independently for each S-Link, using a separate timeout counter for each. The first data-ready signal from an Input FPGA routed to a given S-Link starts the timeout counter. After a programmable interval has elapsed (e.g. 11  $\mu$ s), other inputs routed to the S-Link are timed-out. Data are again transferred out of the Input FPGAs, with those that have timed out providing a timeout indicator block. The timeout condition is indicated via the sub-status word and copied by the Switch FPGA into the event trailer.

Once the timeout mechanism has been triggered, timeout indicator blocks are produced for all failing Input FPGA channels, even if formatted data arrive late and are ready by the time they are needed for readout. The late data cannot be distinguished (by the Input FPGA) from the following event.

The ROD should therefore be robust against removal and replacement of the G-Link input during normal operation.

#### **2.2.5 G-Link DAV Gap**

The ROD uses the G-Link DAV\* signal to frame data from each event. Input data formatting in the ROD is controlled by a state machine, which requires a small number of extra clock cycles to complete after the last incoming data are received. G-Link source modules must hold the DAV\* signal inactive for at least this period, referred to as the “DAV Gap”, even if another event is ready for transmission. The duration of the gap depends on the type of firmware in use in the ROD, so should be coded as a programmable parameter in the upstream readout firmware. The DAV Gap length is independent of the number of timeslices being read out.

The required value is at present 1 for all firmware variants except Neutral, which requires a DAV GAP of 3 ticks.

### **2.2.6 Trigger Type Timeout**

There is a time delay between receipt of the L1A and of the trigger type information. The minimum time is 1.05  $\mu$ s and the maximum unknown. If the trigger type does not arrive within a pre-defined time (programmable register) the event is built with the trigger type information set to 0xFF, and a timeout indicated in the status register.

### **2.2.7 Handling S-Link Link Full**

The Switch FPGA monitors the Link Full Flag (LFF) signal from all active S-Links, and pauses any data transfers within two clock cycles whenever the S-Link becomes temporarily full, as required by the S-Link specification. LFF may occur at any time, independently on any combination of the active S-Links, and the duration may be any period from 1 clock cycle upwards. Since the LFF signal prevents data transfer out of the ROD module, the incoming data may overflow the Input FPGA buffers. To prevent this, the ROD must assert "BUSY", asking the CTP to stop sending triggers, before the buffers get full.

Note that LFF is often referred to within ATLAS as Xoff. The ROD has the capability to generate an internal LFF signal under software control to pause traffic on any S-Link.

### **2.2.8 Handling ROD Busy**

The ROD continuously compares the depth of data stored in each Input FPGA to a programmable threshold. The ROD sets the BUSY bit in the status register and asserts the BUSY front-panel output whenever the threshold is exceeded, and removes these signals when all buffer levels fall to the threshold or below. Empty buffers never cause BUSY even if the threshold is zero, and BUSY is never asserted if the threshold is set equal to the physical buffer depth.

The BUSY signal is passed to the CTP via the BUSY modules. Several events may be awaiting readout, and more may be triggered before the BUSY takes effect. The threshold should be set to ensure that enough buffer space is available in the RODs to accommodate the maximum possible number of events, estimated to be about eight, of the maximum expected size.

If, despite the BUSY logic, the ROD buffers do overflow, data are lost. In this case the ROD sends whatever data are available, but always provides at least an event header and trailer with appropriate status bits set.

### **2.2.9 Event Counter and Event Counter Reset**

The CTP will transmit Event Counter Reset (ECR) signals at a programmable rate of around 1 Hz, depending on the LHC luminosity. ECR is counted in an eight-bit register which is copied into the ECRID field in the S-Link event header.

The ECR mechanism was designed long after the TTCrx ASIC design was complete, so the ASIC does not clear its internal Level-1 ID (event counter) on receipt of ECR. For this reason, the ROD must maintain its own 24-bit L1\_ID counter, which should be incremented on receipt of an L1A and cleared on receipt of ECR [14]. The value of this counter should be used to provide the ROD\_L1ID field in the ROD S-Link header.

ECR is also used to ensure that any unused event data left in the ROD following a timeout are removed cleanly. No L1A occurs during two dead-time intervals of around 1 ms each, generated in the CTP, immediately preceding and following an ECR. The limiting case is that eight events are triggered immediately prior to the start of ECR dead-time. The longest

G-Link readout is the 84-bit sequence per timeslice from the CPM. Eight such events of five timeslices each take just under 84  $\mu$ s to send to the ROD. The 1 ms is also long enough to transmit 100 events to the ROS or RoIB, and ROD BUSY is used to ensure that the data volume stored in the ROD never reaches this level. At a programmable interval (less than 1 ms) after the ECR, the ROD generates Buffer Reset, which clears the pending L1A counter and purges all Input FPGA buffers and the FIFOs holding Level-1 Event Type, BCN, and Orbit. An indicative timing sequence is shown in [Figure 8](#) below.

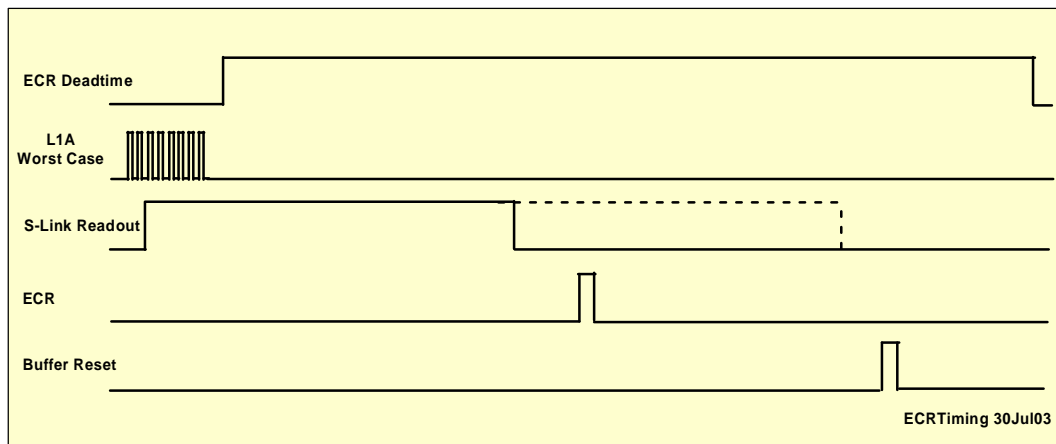


Figure 8: Timing sequence of signals related to Event Counter Reset

### 2.2.10 RoI Data

The principle of data transfer for the RoIs is as above, except for the bit field definitions and the requirement to limit the number of RoIs transmitted to the RoIB from each ROD. If more than a predetermined number of RoIs are received in any one ROD from the CPMs, JEMs, or CMMs, that ROD terminates the RoI transmission and indicates this via a bit in one of the status words.

Sorting the RoIs prior to discarding any beyond the transmission limit, as suggested in [15], is a complex operation and is not implemented.

## 2.3 Data Transfer Rates in the ROD

The data volume into and out of the RODs depends on several parameters, including the type of module, the number of timeslices being read out, the L1A rate, and the effectiveness of the zero suppression or compression algorithm (discussed in section 2.5). Compression efficiency depends strongly on the occupancy of calorimeter cells and hence the LHC luminosity and background conditions.

### 2.3.1 G-Link input data rates

[Table 3](#) shows some parameters for the G-Links from different source modules. The G-Link frame lengths and number of active bits (excluding padding zero bits) are given for one timeslice. For the PPM, the figures are for one timeslice of LUT data and five of FADC data, a combination likely to be used but requiring compression at full trigger rate. Also shown are the resulting incoming data rates in Mbytes/sec when running the trigger at 100 kHz. The final column shows the overall input rate for a ROD servicing a full crate of each type, including system-summing CMMs. Data rates for non-system-summing crates are slightly lower.



ROD Type	Source Module	G-Link frame length (bits)	G-Link active bits per link and per frame	Data rates at 100 kHz L1A and 3 (DAQ) , 5+1 (PPM DAQ), or 1 (RoI) timeslices (MB/s)	
				per G-Link	ROD Total
<b>PPM DAQ</b>	PPM	276	4412	55	882
<b>CP DAQ</b>	CPM	84	1680	63	882
	CMM	35	505	19	38
	Total				920
<b>JEP DAQ</b>	JEM	67	1046	39	628
	Jet-CMM	35	561	21	21
	En-CMM	35	590	22	22
	Total				671
<b>CP RoI</b>	CPM	22	348	4	61
<b>JEP RoI</b>	JEM	45	193	2	39
	Jet-CMM	35	16	0.2	0.2
	En-CMM	35	60	0.8	0.8
	Total				40

Table 3: G-Link data rates into RODs

These figures are computed directly from the data formats shown in section 5 (in fact the diagrams are EXCEL drawings produced by macros, which also compute the data lengths). For example, the CPM has a frame length of 84 bits. All 20 bits are used for each timeslice, making a total of 1680 bits. The 20 longitudinal parity bits are not repeated in a three-timeslice frame, so three timeslices require  $3*1680 - 2*20 = 5000$  bits = 625 bytes. At 100 kHz of L1A, this amounts to 62.5 Mbytes/sec per G-Link (excluding ROD headers and trailers). For 14 active G-Links this total incoming data rate is 875 Mbytes/sec.

An example for the JEM shows the effect of unused bits. The JEM has a frame length of 67 bits, but in the overall frame 22 bits are unused on pin D14 and four on D15. A frame thus provides  $67*16 - 26 = 1046$  bits. The calculation is completed assuming 16 active links.

### 2.3.2 Maximum Trigger Rates

ATLAS detectors, including the trigger, must be capable of readout at 75 kHz, and must be capable of upgrade to 100 kHz. In the calorimeter trigger, the CPM readout over G-Link requires 84 clock periods to read out each timeslice of data. To reach 100 kHz, the number of timeslices must be restricted to four or less ( $4 * 84 * 25 \text{ ns} = 8.4 \mu\text{s}$ ). A possible reduction in the CPM format would give a data length under 80 clock cycles. For the PPM, the ASIC can be configured to read out less than five timeslices, but the G-Link packet length of 276 clock ticks is already fast enough ( $276 * 25 \text{ ns} = 6.9 \mu\text{s}$ ). The JEM and CMM readout lengths, 67 and 35 per slice, respectively, are both short enough to allow 100 kHz readout of five timeslices.

A DAV gap is not included in the above calculations. If set to 1 tick as required for all except neutral format, it consumes about 0.2% of the available readout time at 100 kHz.

### 2.3.3 S-Link output data rates

Converting the G-Link data into 32-bit S-Link words causes the data to expand, due to the addition of addresses and internal headers. Zero suppression and data compression are needed to fit the data plus event headers and trailers into the available S-Link output bandwidth. The degree of data reduction depends on the detector occupancy, and the output data volume may be managed by choice of the number of timeslices read out from different stages of the various modules.

Table 4 shows the estimated data rates and the occupancy of the S-Links when the trigger tower occupancy is 5% of uncorrelated hits, the L1A rate is 100 kHz, and 3 timeslices (PPM 5+1 slices) are being read from all planned inputs per ROD, without discarding any intermediate data. It is pessimistically assumed that about 10 percent of RoIs survive after zero suppression. The data volumes scale linearly with L1A rates if no other conditions change. The calculation does not include the two S-Link control words, ATLAS-standard ROD header (9 longwords), user header (1 longword), status block (2 longwords), and trailer (4 longwords), which total 72 bytes per link per event, or 5.4 Mbytes per link at 75 kHz. This is 3.4% of the S-Link bandwidth.

ROD Type	Source Module	Uncompressed O/P (MB/s)	Compression Factor	Compressed O/P (MB/s)	S-Links/ROD	S-Link Occupancy
<b>PPM DAQ</b>	PPM	986	2	493	4	77%
<b>CP DAQ</b>	CPM	1395	9	152		
	CMM	65	3	22		
	Total	1459		174	2	54%
<b>JEP DAQ</b>	JEM	922	2.8	310		
	Jet-CMM	25	7	4		
	En-CMM	25	1	25		
	Total	972		358	4	56%
<b>CP RoI</b>	CPM	896	20	45	1+1, to DAQ & RoIB	28%
<b>JEP RoI</b>	JEM	58	2.9	20		
	CMM-J	25	7	4		
	CMM-E	25	1	25		
	Total	108		48	1+1, to DAQ & RoIB	30%

Table 4: Estimated S-Link Data Rates from each ROD

## 2.4 S-Link Mapping

The diagrams below show the connectivity between G-Link buffers and S-Links to achieve the data rates in Table 4. The configuration of the switching should be independent of the G-Link input firmware, so that the number of output S-Links in use on any ROD can be set as the L1A rate and data volume require. S-Link port 2 has a logic analyser header, and is included in all mapping arrangements.



### 2.4.1 RoI Data from CPM, JEM, and CMMs

Two copies of RoI S-Link output data are required, one destined for the RoIB and one for the ROS. These data differ only in the S-Link identifier value used in the header (see 3.6.2 ). Data are routed as shown in Figure 9. This connectivity may also be used for DAQ diagnostics at low data rates, in which case the duplicated output to S-Link 2 is not required.

G-Link inputs for RoI data from the CP and JEP subsystems are assigned as shown in Table 5. For testing purposes, it is necessary to operate when some of the source modules are unavailable. In this case, the corresponding G-Link inputs should be disabled in the G-Link disable register.

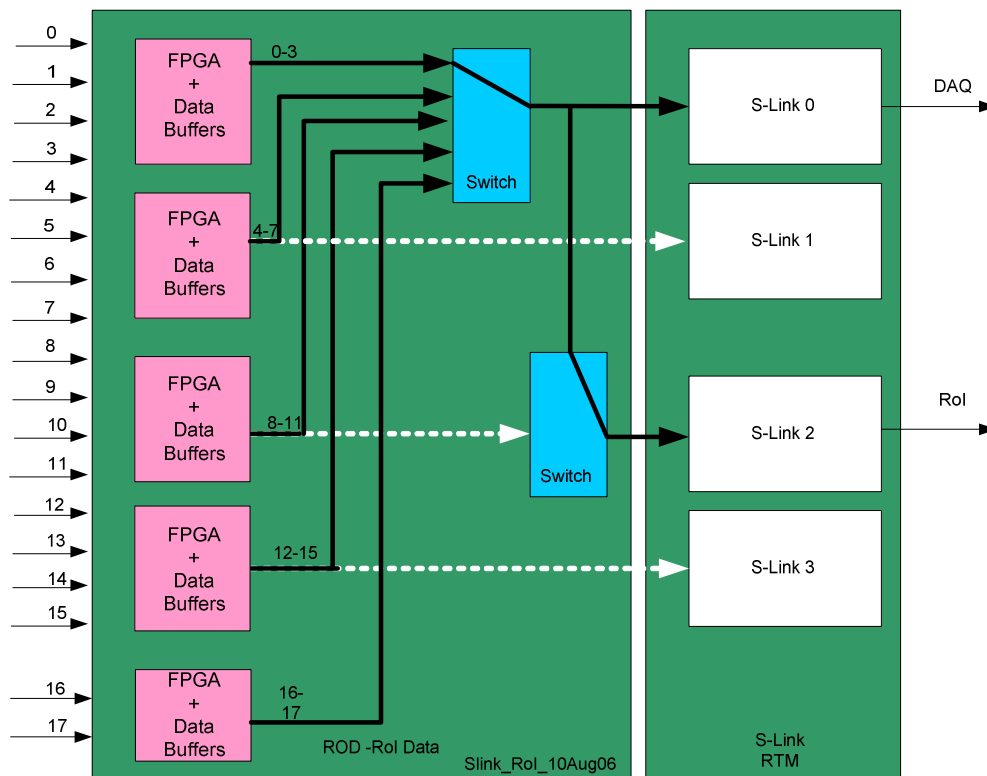


Figure 9: RoI S-Link mapping, showing active (solid) and unused (dashed) data paths.

Note that only one of the two JEP RoI RODs (that receiving data from the CMM system crate) has RoI input from CMMs.

Subsystem	G-Link	Use
CP (RoI)	0	Disabled by software
	1-14	CPM RoI
	15-17	Disabled by software
JEP (RoI)	0-15	JEM RoI
	16	CMM Energy (left) RoI (from JEP system-summing crate only)
	17	CMM Jet (right) RoI (from JEP system-summing crate only)

Table 5: G-Link use for RoI Data

### 2.4.2 DAQ Data from PreProcessor, CP and JEP

Figure 10 shows the connectivity between the ROD and the S-Links. Four S-Links are required to carry the data volume generated in the PPMs and the JEMs when running at the maximum L1A rate, while only two S-Links (numbers 0 and 2) are needed for the CPMs. At lower rates, data from PPMs and JEMs should also be presented on S-Links 0 and 2. At the lowest rates, using a single S-Link, as in Figure 9, is adequate.

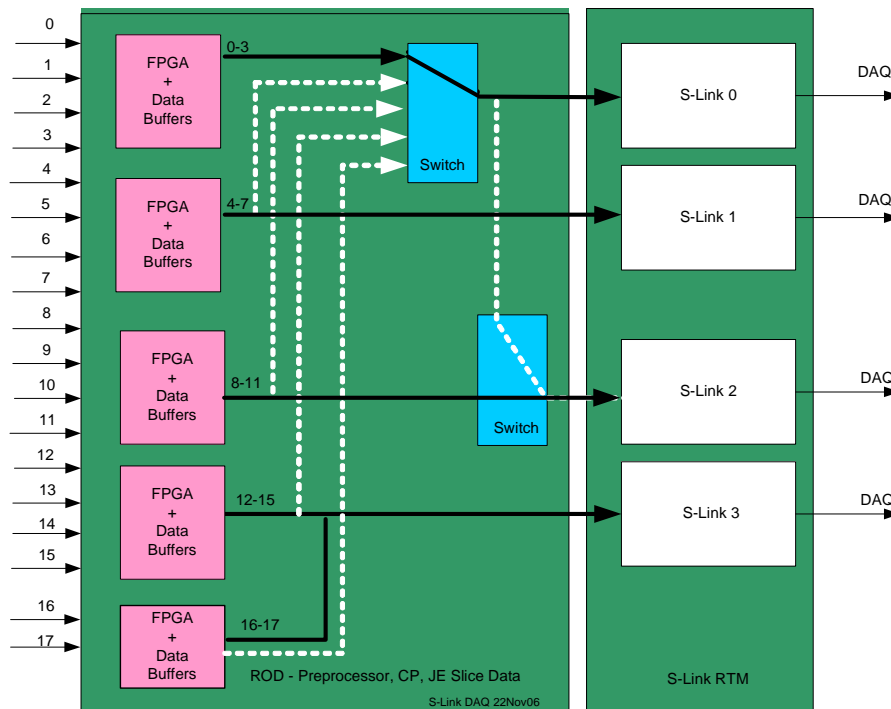


Figure 10: DAQ data S-Link mapping for JEP crates

G-Link inputs for DAQ data from the PreProcessor, CP and JEP subsystems are assigned as shown in Table 6. Each ROD handles all the DAQ data from a complete PreProcessor, CP or JEP processing crate.

Two RODs reading data from PreProcessor crates receive input from only 14 PPMs, which are located in slots 6-12 and 14-20. The G-Link inputs are assigned to mirror the occupied PPM crate slots.

Subsystem	G-Link	Use
PPM (DAQ)	0-15	PPM DAQ
	16-17	Unused
CP (DAQ)	0	Unused
	1-14	CPM DAQ
	15	Unused
	16	CMM e.m/tau. (left) DAQ
	17	CMM e.m (right) DAQ
JEP (DAQ)	0-15	JEM DAQ
	16	CMM Energy (left) DAQ
	17	CMM Jet (right) DAQ

Table 6: G-Link use for DAQ data

## 2.5 Zero suppression and compression

Various forms of data reduction are needed to reduce the size of the S-Link packets going to the ROS or RoIB. The S-Link output data are used downstream to check the trigger performance, so the data reduction process must be loss-less, i.e. it must be possible to recreate the ROD input accurately from compressed readout data. There are four basic tools for data reduction:

- The number of timeslices of data read out from the trigger modules is controlled on the modules themselves.
- The number of timeslices of data can be reduced further in the ROD. This includes the possibility of eliminating some data altogether (e.g. from the two ends of data links) by setting the number of timeslices to zero.
- Zero suppression — see section [2.5.1](#).
- Compression — see section [2.5.2](#).

The last three operations are carried out in the ROD Input FPGAs.

A further requirement is that suppression and compression must handle the full data rate into the ROD at the maximum specified L1A rate of 100 kHz and at the incoming G-Link clock frequency of 40 MHz. All such suppression and compression are therefore done in pipelined firmware. The CPUs in the ROD FPGAs are not used for this purpose.

For diagnostic use, it must be possible to process data without suppression or compression. This capability is provided by the “neutral” data format. However, a limit on the maximum size of an S-Link packet is imposed by both the ROS and RoIB, so uncompressed data are unlikely to be acceptable in normal running.

### 2.5.1 Zero suppression

The G-Link input data from some modules will contain many zero values. Examples include the RoI data from the CPMs and the JEMs, and DAQ data from the CPMs, JEMs, CMMs, and PPMs (LUTs).

The ROD is capable of zero-suppressing such data. To do this, incoming data are discarded if all data, error, and saturation bits are zero, and the surviving data are then labelled with channel, module, and crate addresses. Details of the criteria for zero suppression are provided with each data format in section 5.

### 2.5.2 Compression

Data from the PreProcessor consists of raw FADC samples and one or more PPM\_LUT values per channel. For much of FADC readout, the data have small non-zero values close to the FADC pedestal, and zero suppression is not effective. Lossless data compression is needed, and three output formats are specified:

- an uncompressed format, in which the output data contain the full data value for each input datum. This format is used only for diagnostics or when the numbers of timeslices to be read out do not match the compressed (or super-compressed) format. This format is as close as possible to the input data format.
- a compressed format, in which the PPM\_LUT output and FADC data are packed into short bit fields, with longer bit fields available to encode larger values. This format, intended to handle only 1 LUT sample and five FADC samples, is described in [16].
- a super-compressed format, similar to the compressed format, except that data are omitted for those PreProcessor channels in which the PPM\_LUT result is zero and the FADC samples are all below a programmable (per G-Link) threshold. This format is loss-less only for those PreProcessor channels with a non-zero PPM\_LUT output or an energetic FADC sample.

Details are given with the data formats in section 5.

## 2.6 Data Replay Mode

The ROD provides a data replay mode for testing the switch and monitoring functions, and to provide test S-Link data for the ROS and RoIB. Details are included in section 3.13.

Data to be replayed must be loaded by software into the FIFOs in the Input FPGA, together with the details of the event lengths it contains. When in replay mode, the Input FPGA ignores the G-Links, and instead sends data directly from the FIFOs associated with enabled channels. In replay mode, the data read from the FIFOs is also recycled back to the FIFO input so that an indefinite number of events may be sent.

Replay data readout is initiated, as in normal ROD operation, by a L1A, either arriving via the TTC or generated internally to the ROD by VME command to the pulse register. FIFOs should be cleared by software before terminating replay mode.

## 3 Functional blocks

This section describes the functional blocks shown in [Figure 5](#).

### 3.1 Optical Inputs

The physical connections between the ROD module and the processor modules (PPM, CPM, JEM, and CMM) are via optical links operating at 960 Mbaud. The receivers are small form-factor commercial 1.25 Gigabit Ethernet compliant, optical dual receiver modules with LC type connectors (the M2R-25-4-1-TL from STRATOS Lightwave [\[17\]](#) ). Nine of these modules fit on the ROD to interface to 18 links.

### 3.2 G-Link

The interface between the optical receivers and the FPGAs uses a commercial “G-Link” serial to parallel converter and clock recovery receiver chip HDMP1024 (Rx) from Agilent Technologies. This chip can receive 16-bit or 20-bit data. The corresponding parallel to serial converter on the processor modules (PPM, CPM, and CMM) is the transmitter chip HDMP1022 (Tx). The JEM uses the compatible HDMP1032 (Tx). The 18 HDMP1024 receiver chips on the ROD each connect to a single processor module via the optical receivers. Depending on the position within the system, not all 18 G-Links are required at all times.

The G-Link chips operate in “Simplex method 2”, in which the receiver requires a synchronisation signal from the transmitter. The synchronisation signal is routinely sent during DAV gaps. The G-Link receiver does not require a local reference clock, although a 40 MHz crystal oscillator on board is provided to generate a clock source for testing when the TTC is not present. To prevent problems, selection of a TTC source is prohibited if the TTC is not present, or not ready.

Two G-Link input pins, DIV0 and DIV1, define the frequency range over which the G-Link can operate. DIV1 is pulled to ground on the PCB, while DIV0 is under control of the Input FPGA firmware. At present it is driven as the inverse of the mod20sel level, which is set by VME. This is due to the difference in frequency ranges in the 16- and 20-bit modes of the G-Link.

VME-readable link loss and error counters must be provided for each G-Link.

### 3.3 Input FPGA

Four G-Links feed data into each Input FPGA (which is equivalent in functionality to the previous (6U) prototype ROD [\[18\]](#)). Of the five input FPGAs, four receive data from four G-Links each and the fifth from two G-Links only. The input FPGAs receive the data, check for protocol or parity errors, compress the data if required and then write them, in S-Link data format, into internal block RAMs. Each G-Link input has one FIFO buffer approximately 8k deep x 32 bits wide. These buffers are VME-accessible and can also be used in replay mode as a data source, as described in section [2.6](#). All input FPGAs feed the S-Link formatted data onto the ‘switch FPGA’. The Xilinx XC2VP20-5FF896 device is used.

### 3.4 Switch FPGA

The Switch FPGA assembles, routes, and can spy on complete S-Link event packets. The S-Link ROD header, trailer and status words are constructed using buffers and registers in the

Switch FPGA itself (see sections 3.4.3 to 3.6.5). The data payload is collected from the Input FPGA buffers.

The Switch FPGA has the capability under firmware control to route the data, as required, to one or all of the four S-Links mounted on the rear transition module.

Four 8k x 32-bit wide buffers are provided, one per S-Link, to spy on the output S-Link data stream. The spy memory buffers each hold a copy of the data from a small number (e.g. 4) of events (data corresponding to the same L1A from all the Input FPGA data buffers routed to that S-Link). The spy buffers are filled, read and released in time order. The spy data can be read by an embedded or on-board CPU, or transferred to a single-board computer via the VME bus for monitoring and analysis. The contents of the spy buffers can also be copied to the Monitoring FPGA (see section 3.7.1) using high-speed (Rocket I/O) data links on the FPGA for further processing.

Each of the four spy buffers is further internally sub-divided into four event buffers (so that there are 16 event buffers in total, four per S-Link). The Switch FPGA maintains an “occupied” flag for each event buffer. When an event has been selected for sampling, it is copied into the next unoccupied event buffer (for the corresponding S-Link), the occupied flag is set, and the length of the event is written into the appropriate position in the event length buffer. The decision to sample an event requires that the same event buffer should be available in the spy area for all active S-Links.

It must be possible, for special runs, to disable the S-Link outputs, but the event samples should still enter the spy memories.

The Xilinx XC2VP30-5FF896 device is used for the Switch FPGA function. This device has sufficient memory capacity (1136 block rams of 16k bits each) to accommodate the required firmware concurrently with the Chipscope software and with room for future expansion.

### ***3.4.1 Event Selection and Sampling***

Events are selected as candidates for sampling into the spy buffers based on the trigger type followed by pre-scaling. The trigger type is first masked with a programmable mask, and the event is deemed eligible for sampling if the result equals a programmable value. Eligible events are pre-scaled before becoming sampling candidates. Candidate events are actually sampled if the same spy buffer is free for each S-Link, and all contributing S-Links are sampled into the corresponding memories.

The ROD supports two sampling algorithms:

- By default, the ROD randomly samples selected events. In this mode, pre-scaled eligible events are copied from the event stream whenever spy memory is available to contain the resulting event fragments.
- For testing, and possible use in calibration, a 100% sampling mode is available. In this mode, internal LFF is asserted whenever a spy memory is full. This pauses the output flow of events, and the BUSY logic will reduce the L1A rate accordingly. To sample all events of all types, all bits of the trigger type should be masked off, the result required to be zero, and the pre-scaling factor set to 1.

Once an event fragment has been copied to spy memory, the fragment length is written into the fragment length register, and the buffer is marked as occupied. The occupied flag remains set until cleared by monitoring firmware or software.

### **3.4.2 TTC-correlated Event Selection and Sampling**

It is essential that all RODs in the system be able to sample the same event. To support this, the spy buffer controller responds to a TTC command value 0x04. At high trigger rates, the ROD may be working on events when the command is received, and the strategy is designed to accommodate this.

When the command is received, a “TTC Sample Pending” flag is set. When the next L1A arrives, the sample flag is attached to it, for example by adding a 13<sup>th</sup> bit to the bunch crossing number. The L1A is queued for processing, but the event is eventually considered for sampling. At this time, the sampling controller detects the flag, clears all Spy buffers, and re-initialises the pre-scaling counter. The flagged and following events are then processed as normal and are stored in the empty spy memories. If the selection criteria are identical in all RODs, the same events should be sampled.

As it clears the spy buffers in this way, the controller also writes a zero into the TTC\_SampleID register. When the event copying is complete, the L1A number of the flagged event is written into the register. Reading this before and after copying an event for processing, a monitoring program can determine if the data it read is likely to have been overwritten. There is a single such register governing all spy memories.

### **3.4.3 Bunch Crossing Number Buffer**

This is 256 deep x 16 bits wide FIFO buffer to queue the bunch crossing number (12 bits) generated by the TTCrx chip (see section 3.9) on a level-1 accept signal. When the first (chronologically) Input FPGA signals that it has a complete data block available, the first BCN in the queue is transferred to the event headers, ready to be transferred out on the S-Links. If the buffer overflows, the BCN\_FIFO\_Overflow bit should be set in Switch status register 2.

### **3.4.4 Event Number Buffer**

This is a 256 deep x 32 bits wide FIFO buffer to queue the extended event number formed from the 24-bit event number (L1-ID) counter and the eight bit Event-Counter-Reset (ECR) counter. When the first (chronologically) Input FPGA signals that it has a complete data block available, the first 32-bit number in the queue is transferred to the headers, ready to be transferred out on the S-Links. The ECR register is incremented each time an event counter reset command is received over the TTC, and is set to zero when a run number is loaded into the Run Number Register at the start of a run. If the buffer overflows, the Event\_Number\_FIFO\_Overflow bit should be set in Switch status register 2

### **3.4.5 Trigger Type Buffer**

This is 256 a deep x 8 bits wide FIFO buffer to queue the level-1 trigger type received by the TTCrx chip following a level-1 accept signal. When the first (chronologically) Input FPGA signals that it has a complete data block available, the first Trigger Type number in the queue is transferred to the headers, ready to be transferred out on the S-Links. If the buffer overflows, the Trig\_Type\_FIFO\_Overflow bit should be set in the Switch FPGA Status Register.

### **3.4.6 Orbit Number Buffer**

This is 256 deep x 16 bits wide FIFO buffer to queue the orbit number sampled from the orbit count register when the TTCrx chip receives a level-1 accept signal. When the first



(chronologically) Input FPGA signals that it has a complete data block available, the first orbit number in the queue is transferred to the headers, ready to be transferred out on the S-Links. If the buffer overflows, the Orbit\_Number\_FIFO\_Overflow bit should be set in Switch status register 2.

### **3.4.7 Status Words**

Two status words are included in each event, to indicate, separately on each active S-Link, the status of the data sent on that link. In events with no errors, both words are zero. In the case of RoIs, a uniform content for these words has been agreed, in which the first word contains error bits and the second contains other status bits. Details of the bit content of the words are included in the event formats in section 5

### **3.4.8 Status Count**

This 32-bit register is used to indicate the status summary using bits [31:16]. This field is not defined yet, therefore should be set to hex 0000. The bit field [15:0] is used to indicate the number of status words in the event fragment, which is two in this design.

### **3.4.9 Data Count**

This is a separate 32-bit counter for each S-Link, indicating the number of data words in the corresponding event fragment data payload.

### **3.4.10 Checkpoint Handling**

A checkpoint is a mechanism to allow rapid increment of run number when minor adjustments in ATLAS operating conditions are needed. The behaviour required of the ROD is described in [19].

A checkpoint is accompanied by a Sweeper Event, identified by the trigger type, which is set to 0x07. As the Switch starts the assembly of each event, the trigger type is inspected. When the value 0x07 is recognised, the ROD performs the following actions:

1. Immediately asserts the ROD\_Busy signal
2. Completes the assembly and S-Link transmission of the Sweeper Event.
3. Performs end of run operations as necessary. These functions are not yet known. When the Monitor FPGA functions are fully defined, this is likely to involve transmitting a flag to the monitor so that end-of-run calculations can be completed. In this case, the Monitor FPGA signals completion back to the Switch FPGA.
4. On completion of end-of-run operations, increment the low-order 24 bits of the run number register.
5. Release the ROD-Busy signal.



## 3.5 S-Link

### 3.5.1 S-Link Hardware Interface

S-Link [4] is the standard interface used in ATLAS to transfer data collected by RODs to the DAQ and Level-2 (RoI) subsystems. The S-Link interface, shown in Figure 11, is a rear transition module similar to the CERN S2VME64X [20] transition module, modified to use the 3.3V signalling and power used by the commercially available “HOLA” S-Link [21]. The transition module includes 5V power traces which were used in tests with prototype RODs but are not powered in the production ROD.



Figure 11: Rear S-Link Transition Module

Up to four HOLA S-Link source modules plug on to this rear transition module. The transition module obtains its input data and returns status indicators via the user-defined pins on the J2 connector, and an added J3 connector. Appendix A shows the pin-out. The transition module is equipped with headers to allow logic analyser access to all S-Link pins of S-Link 2 while the S-Link daughterboards are in place.

Separate event fragments, each with the ATLAS-standard format, are transferred concurrently and asynchronously over each active S-Link for each Level-1 accept. The ROD must be capable of driving all S-Links at close to their rated sustained maximum of 160 Mbytes/s. This is needed when running at high trigger rates, and may also be reached at lower trigger rates if compression is disabled, or during calibration runs.

### 3.6 ROD S-Link Packet

S-Link packets consist of beginning- and end-of fragment control words surrounding ATLAS-standard S-Link ROD fragments, both for DAQ and RoI data. The ROD fragments start with an ATLAS-standard event header and end with a standard event trailer. These enclose the event data payload. The overall format is defined in reference [22] and illustrated below.

Content			Type
Begin of Fragment (B0F00000 Hex)			Control
Start of Header Marker (0xEE1234EE )			Data
Header Size (9 – number of words in the header excluding control word)			Data
Major Format Version No [31:16]=0x0300		Minor Format Version No [15:0] = 0x1XXX	Data
Reserved [31:24] = 0	Sub-detector ID [23:16]	Module ID [15:0]	Data
Run Type	Run Number (24 bit)		
ECRID [31:24]	ROD_L1ID (24 bit event no) [23:0]		Data
ROD_BCN (12 bit Bunch Crossing Number)			Data
Level1-1 Trigger Type			Data
Detector Specific Event Type Orbit Counter (16 bits)	Sequence Number (12 bits)	Type (4 bit)	Data
Payload – User Header followed by formatted event data.			Data
Status Word-1			Data
Status Word-2			Data
No. of Status elements = 2			Data
Number of data elements			Data
Status block position – 1(data first)			Data
End of Fragment (E0F00000 Hex)			Control

Table 7: ATLAS ROD Event Fragment, Version 3.0

#### 3.6.1 Format Version Numbers

The Major Format Version is defined in [22] to be 0x0301 (ROD format version 3.1, Data format version 4.0). However, it is possible that this number will change before ATLAS starts. Therefore both the major and minor version numbers are copied from the [Format Version Register](#) into the ROD S-Link header. Software is responsible for setting both fields to the required values.

The value loaded by software divides the minor format into two fields – the top nibble, specifying major version, should be set to 0 for the 6U ROD and 1 for the 9U ROD. The remaining 12 bits should then be used to specify the 9U data format number. Software byte decoders should thus interpret minor version number values like 0x1XXX to mean 9U ROD data, and 0x0XXX would mean 6U ROD data from the 2004 test-beam. This document describes 9U data format number 2, giving the Minor Format Version a value 0x1002.

Format numbers for the 9U ROD are defined by the version of this document in which they appear, as listed in [Table 8](#) (which will be extended if new formats are used). Changes to formats of individual sub-blocks should be indicated by increasing the “Vers” field in the sub-block header. If this field is fully used, or if more substantial changes are applied to the sub-blocks or to the overall ROD fragment format, the Minor Format Version number should be increased.

Rod Spec Version	Minor Format Version
6U	0x0XXX
9U spec version 1.09	0x1001
9U spec version 1.2	0x1002

Table 8: Minor Format Version Numbers.

### 3.6.2 Module Identifiers

The Module Identifiers used in the S-Link header are built from three bit-fields identifying the upstream crate number(c), the data type – DAQ or RoI(r), and the S-Link number (s), with the bits from most- to least-significant thus: r0ssccc. Possible values are listed in [Table 9](#).

Subsystem	Crate ( c )	RoI ( r )	S-Link ( s )
PreProcessor	0-7	0	0-3
Cluster Processor	0-3	0-1	0-1
Jet-Energy Processor	4-5	0-1	0-3

Table 9: Module identifier fields.

### 3.6.3 Subdetector Identifier

The Subdetector Identifiers are defined by ATLAS as follows:

PPM Readout	0x71		
CP DAQ Readout	0x72	CP RoI	0x73
JEP DAQ Readout	0x74	JEP RoI	0x75

### 3.6.4 Run Number and Run Type

The run number is a 24-bit field. The upper 8 bits contain the run type. Run types currently defined are: 0 = physics, 1 = Calibration, 2 = Cosmics, 15=test.

### 3.6.5 Detector-Specific Event Type

The detector-specific event type provides support for simulation of the trigger when ROD-Busy might otherwise break the correspondence between the hardware and simulated LIAs. The low 16 bits of this header field (sequence number and type) are copied from the lower 16 bits of the 32-bit detEventType register. The upper 16 bits (orbit count) are loaded from the orbit-count FIFO.

The orbit counter is a 32-bit counter which is incremented by the appropriately qualified BCR, and overflows to zero at the number of orbits specified in the orbit wrap value register. The orbit-counter should be reset to zero by either one of the orbit counter reset or TTC reset bits of the pulse control register.

The FIFO should be reset via the FIFO reset bit of the pulse register.

### 3.6.6 User Header

The ATLAS ROD fragment format makes no specific allowance for a detector-specific ROD header, so it must be treated as part of the data payload. The user header therefore appears at the beginning of the data payload, immediately after the ATLAS-standard header, but only in

DAQ data, not in RoI data. The user header format is shown in Figure 12. The low four bits of the first header word contain the total number of header words, including the first. The required number of header words is copied from the User Header Registers in the Switch FPGA.

The remainder of the first user header word contains six fields. Four fields indicate the offset to the triggered slice within the data read out (i.e. a value of 0 indicates that the triggered slice is the first slice read out). Separate offsets are provided for PPM LUT and FADC readout, while a common value is assumed for the CP system and another common value for the JEP system. The next field contains the Lower Bound value used in PPM data compression, and the uppermost nibble contains a Word-Id which should be set by software to 0xF.

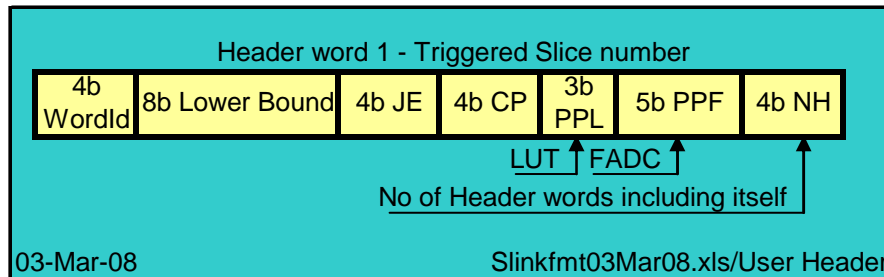


Figure 12: Format of User Header

## 3.7 Calibration and Monitoring

### 3.7.1 Monitoring FPGA

The ROD uses firmware for all data input, reformatting, and output operations, so the on-chip Power PCs are not be used for any of these tasks in the main event flow from G-Link to S-Link. However, CPU capacity may be required in particular for calibration, and may be used on the ROD for event analysis and monitoring. The monitoring FPGA interfaces to the switch FPGA and can transfer data from the spy buffers on the switch FPGA for further ‘processing’ on this FPGA as required. The processing of the data can be done by firmware or interfacing to a commercial PMC card (33 MHz, 64-bit) or using the two on-chip Power PCs. In all cases, output results from CPU calculations are written to VME-accessible memories. A Xilinx XC2VP20-5FF896 device is used.

The following example illustrates how this capability might be used. A calibration run may consist of bursts of 100 events at each of several pulse heights. As data for each event arrive at the ROD, the uncompressed data format is generated and the event fragment written to one of the spy memories (but not sent to the ROS). The onboard CPU immediately copies this event, releases the spy memory, and accumulates statistics relating to the pulse height. When all 100 events have been read, the CPU calculates the mean and sigma of the measured energy, stores them in a VME-readable memory, sets a status bit in the CPU Interface register to indicate completion, and optionally generates a VME interrupt. The crate single-board computer monitors the RODs, and as soon as all have finished, initiates a burst of events at the next energy. When all energies have been processed, the crate CPU reads the complete results block from the ROD.

### 3.7.2 Instruction RAM

Two blocks of external RAMs are provided as instruction RAMs for the two on-chip Power PCs on the monitoring FPGA. They use 128 Mbit synchronous DRAM devices which can be

upgraded to 512 Mbit if required in the future. They are organised to provide 8M x 32-bit words (32 Mbyte) per CPU.

### **3.7.3 Dual-Port RAM**

The monitoring FPGA also interfaces to a 64K x 32-bit dual-port RAM which is used to output the results of any processed spy data. The contents of this Dual-port RAM can be read via the VME.

### **3.7.4 PCI Interface**

The monitoring FPGA also provides an interface to an optional PMC daughter card conforming to PCI 33 MHz, 64 bit interface standard.

### **3.7.5 Auxiliary Front-Panel Output**

A single-pole LEMO-00 connector is provided on the module front panel. This is intended for use if additional hardware signalling from the ROD is required, for example at some point during a calibration run. It is driven from the VME FPGA but tracking is provided to connect to the Switch and monitoring FPGAs. The signal may also be activated and removed by VME command.

## **3.8 VME Interface**

A single board computer in the ROD crate will access all module registers over the VME bus. However sufficient tracks should be provided to allow the ROD module to become a VME master in case this is required in a future upgrade.

The VME interface is provided via a separate FPGA which implements the CR/CSR space required for VME64x and all other VME functions required for this module.

The VME interface is VME64x standard with the following:

- Capable of D32, A32 cycles
- Implement CR/CSR space
- Interrupt capability.
- Capable of VME64x block transfers (as A32/D32 slave to memories/FIFOs)
- Use five-row connectors as per VME64x specification
- Pin assignment for J0 (VME64x) shall be compliant to the TCM-VME card. The pinout is listed in Appendix B.
- Pin assignment for J1 and J2 shall be to VME64x specification and the J2 user defined pins of J2 and J3 are compliant to S-Link rear transition module.
- Will not support hot swap capability
- Use geographical addressing for accessing the module as described below:

Extended geographical addressing is provided in the VME crate through standard VME lines GA0\*-GA4\* and additional application-specific lines GA5\*-GA7\* which provide the crate number. ROD crates are numbered only 0 and 1, so only bit GA5\* is actively used in RODs. The ROD uses the geographical addressing lines to determine:

- Its function in the system, and hence the default firmware to load on power-up;

- The TTCrx address;
- The CANBus node address;
- The Module VME base address and configuration space base address;
- Module identifiers used in the S-Link data (e.g. CPM and CP-Crate numbers);

The TTCrx address is obtained according to the scheme in [23], using the six least significant bits of the geographical address thus: 00'G5G4G3G2'G1G000'0110. The setting of the least significant six bits is necessary to address all modules individually, and requires the I2C address to be set to 6.

VME base addresses are similarly obtained, with each module occupying 8 Mbytes of address space, thus 1100'G<sub>4</sub>G<sub>3</sub>G<sub>2</sub>G<sub>1</sub>'G<sub>0</sub>000'0000'0000'0000'0000'0000. This scheme matches the PreProcessor addressing scheme, allowing testing of both types of modules in the same crate (caution: suitable powering is needed). The CR/CSR space base address in A24 space is G<sub>4</sub>G<sub>3</sub>G<sub>2</sub>G<sub>1</sub>'G<sub>0</sub>000'0000'0000'0000'0000. Addressing is summarised in Table 10. Where the function is shown in parentheses, the slot will normally be empty, but the function indicated will be loaded if the module is placed there.

Slot	Function	TTCrx	CANBus	VME Base	CR/CSR Base
1	Crate SBC	n/a	n/a		
2	Empty				
3	PPM DAQ	0x00C6		0xC180'0000	0x18'0000
4	(PPM DAQ)	0x0106		0xC200'0000	0x20'0000
5	PPM DAQ	0x0146		0xC280'0000	0x28'0000
6	(PPM DAQ)	0x0186		0xC300'0000	0x30'0000
7	PPM DAQ	0x01C6		0xC380'0000	0x38'0000
8	(PPM DAQ)	0xC206		0xC400'0000	0x40'0000
9	PPM DAQ	0x0246		0xC480'0000	0x48'0000
10	(PPM DAQ)	0x0286		0xC500'0000	0x50'0000
11	CP DAQ	0x02C6		0xC580'0000	0x58'0000
12	CP DAQ	0x0306		0xC600'0000	0x60'0000
13	J/E DAQ	0x0346		0xC680'0000	0x68'0000
14	(JE DAQ)	0x0386		0xC700'0000	0x70'0000
15	CP RoI	0x03C6		0xC780'0000	0x78'0000
16	(CP RoI)	0x0406		0xC800'0000	0x80'0000
17	CP RoI	0x0446		0xC880'0000	0x88'0000
18	(CP RoI)	0x0486		0xC900'0000	0x90'0000
19	J/E RoI	0x04C6		0xC980'0000	0x98'0000
20	(JE RoI)	0x0506		0xCA00'0000	0xA0'0000
21	TCM	n/a		0x680000 (A24)	N/A

Table 10: ROD VME addresses and functions in both ROD crates.

The crate single-board computer and TCM [24] must be VME64x compliant, with addresses chosen to avoid conflict with RODs and each other.

### 3.8.1 Interrupt Capability

Following the recommendations of reference [25], the ROD is capable of generating interrupts on any of the seven IRQn\* lines. The IRQn\* should release the interrupt line automatically during the interrupt acknowledge cycle (“ROACK” mechanism). The module should provide an 8-bit Status Id value in response to an 8-bit D08(O) interrupt acknowledge cycle. The computer interprets this value as an interrupt vector. The Interrupt Register allows any one of the IRQn\* lines and any Status Id value to be selected.

Tracking is provided to allow any of the main FPGA devices (Input, Switch, and Monitoring) to raise an interrupt. The board-level interrupt source should be latched, and further interrupts from the same board FPGA device are inhibited until the register has been cleared under software control. It is expected that the principal interrupt user will be the monitor FPGA and/or daughter PCI card.



### 3.9 TTCrx Decoder Card

The TTC system provides the clock and control signals via the TCM, which are then decoded on the ROD module via a plug-in TTCrx decoder card [26] which hosts a CERN TTCrx chip. The main 40.08 MHz clock for the module is software selectable to be either clock40des1 or clock40des2, and fanned-out using appropriate zero delay clock buffers on the ROD module. The TTCdec internal crystal clock is also available for testing purposes. VME access to the internal registers of the TTCrx chip is provided over the I2C interface, following the common scheme used on other processing modules in the calorimeter trigger.

All TTCrx output pins should be tracked to the switch FPGA so that all the relevant information such as the bunch crossing number, event number, trigger type are extracted, to construct the S-Link packet (see section 3.6). A subset of TTCrx information, including the bunch number, is required for comparison with incoming data in the input FPGAs. The TTCrx address is obtained from the module VME64x geographical address, as described in section 3.8.

### 3.10 CANBus

The ATLAS Detector Control System (DCS) will monitor the physical state (temperature and voltages) of the ROD module. The interface to the DCS is via the TCM on the ROD crate. The ROD interface to the TCM is provided by a Fujitsu CANBus micro-controller (MB90F591A). The voltages and the temperature to be monitored are connected to the micro-controller via suitable sensors. The CANBus controller address is obtained from the module VME64x geographical address, as described in section 3.8.

### 3.11 Reset, Reload, and Module Initialisation

A CPLD is provided to initialise the module and reload the firmware if required. The CPLD has a VME-accessible register to monitor the DONE signals of the FPGAs as well as restart the loading by issuing an appropriate signal to the System ACE chip. This is also initiated following a VME System reset issued by the VME crate controller. A separate Reset function is also available for the module and all FPGAs. This function resets all hardware and firmware logic blocks but does not reload any firmware.

### 3.12 System ACE

All FPGAs are configured using the Xilinx System ACE (CF) solution. The configurations are held on a Compact Flash card (CF) in the form of complete firmware collections for the module, and downloaded to the FPGAs via the JTAG chain. The appropriate collection (one of eight) is selected via three pins on the System ACE chip. These three pins are driven by the module initialisation CPLD by interpreting the crate geographical address pins, but can later be updated through a VME register implemented on the CPLD. The standard collections are listed in Table 11. Test collections can be loaded by exchanging the compact flash card.

It is important that Compact Flash cards are prominently labelled so that only those intended for the ROD are inserted into the ROD module.



Configuration Address	ROD Type	Crate Slot
0	PPM DAQ Uncompressed	3 - 9
1	PPM DAQ Compressed	none
2	PPM DAQ Super-Compressed	none
3	CP DAQ	10-13
4	JE DAQ	14 - 15
5	CP RoI	16 - 18
6	JE RoI	19 - 20
7	Neutral	none

Table 11: System Ace standard firmware collections.

The compact flash card can be removed (front panel access) and programmed on a PC. The front panel also provides a connector for the JTAG port so that the module can be configured directly from a PC via JTAG, mainly during initial module test phase.

Space is reserved in the programming model to provide VME access to the System Ace registers should this be required in future.

### 3.13 Using Replay Mode

In replay mode, data previously stored on the ROD is used to construct S-Link ROD fragments in place of data received over the G-Links. To operate this mode, settings are needed for each active channel in the Input FPGAs, and to provide the event headers in the Switch FPGA. Once these values are set, the ROD should generate replay packets on receipt of L1A. A software L1A can be generated by a bit in the Control Pulse register 2 if required.

#### 3.13.1 Input FPGA Settings for Replay

- Input G-Links are disabled by clearing the *G-Link enable* bit in the Input Channel Subslice Control Register. All G-Link signals are ignored from this point. Setting this bit also stops the storing of bunch numbers in the Input FPGA BCN FIFO. This is necessary as the FIFO is only emptied during G-Link data processing.
- Input FPGA FIFOs are reset using the *Clear FIFOs* bit in the Input Control Pulse Register.
- Required event data are written into the Input Channel Data FIFOs, starting at offset 0. The data loaded are copied directly into the S-Link ROD fragments, so must include any required sub-block headers and trailers.
- The corresponding Input Channel Data FIFO write pointer is moved to the next location above the data written.
- The count of the number of words loaded is written into the Input Channel Event Management FIFO, and the write pointer for this FIFO increased by one.

### **3.13.2 *Switch FPGA Settings for replay***

- The values required in the ROD S-Link header are loaded into the Replay BCN, Replay Event Number, Replay Trigger Type registers.

## **3.14 Motherboard**

All the above functionality is implemented on a 9U motherboard. The motherboard is a standard 16-layer 2mm thick 9U x 400 mm VME64x module and complies with IEEE 1101.1/IEEE 1101.10. It is a single-width module with a front panel and ejector/extractor handles as specified in IEEE 1101.10.

The ROD module has strengthening bars and ESD strip as described in Appendix C.

### **3.14.1 *Signal Handling on the Motherboard***

The track between the optical receiver module and the G-Link receiver chip must be as short as possible, and implemented as a 50  $\Omega$  impedance transmission line. All other signals on the motherboard are 60  $\Omega \pm 10\%$ . The smallest track size is 75 micron with 100 micron gap.

### 3.14.2 Front Panel Input and Outputs

#### 3.14.2.1 Front Panel Inputs

Description	Signal Name	Connector
Data-In	Optical Gigabit	Small form factor LC
Compact Flash	CF specification rev.2	1.27 mm card connector
JTAG	Xilinx parallel cable IV	2 mm Molex (87333-1420)

The JTAG port connects to the ribbon cable which is supplied as standard with Xilinx Parallel IV programming cables.

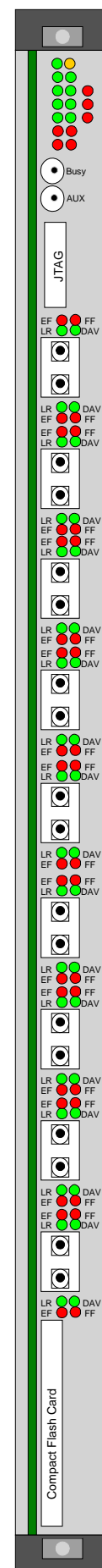
#### 3.14.2.2 Front Panel Outputs

Description	Logic	Connector
ROD Busy	TTL (active low)	Single pole LEMO 00
Aux Out	TTL	Single pole LEMO 00

#### 3.14.3 Front Panel Monitoring (LEDs)

Description	Signal name	Colour
System ACE Error	AERR	Green
System ACE Status	ASTAT	Green
FPGAs Configured	DONE	Green
2.5 V VCCO	2.5IO	Green
2.5 V AUX	2.5AX	Green
3.3 V VME	3.3	Green
5 V VME	5	Green
3.3 V VCCO	3.3IO	Green
3.3 V Rest	3.3R	Green
2.5 V RocketIO	2.5RI	Green
3.0 V PCI	3	Green
1.5 V	1.5	Green
Reset	RST	Yellow
TTC Ready	TTC	Green
System Clock Active	CLK	Green
VME Access in progress	BSEL	Yellow
Level-1 Accept	L1A	Yellow
S-Link Full	SLFULL(1-4)	Red
ROD Busy	RDBSY	Red
OR of G-Link Errors	GLERR	Red
G-Link Active	DAV<n>, n=0-17	18 x Green
G-Link Ready	LR<n>, n=0-17	18 x Green
Data FIFO Empty	FE<n>, n=0-17	18 x Red
Data FIFO Full	FF<n>, n=0-17	18 x Red

- The front panel layout is shown in the drawing opposite (not to scale)



- Since space is limited on the front panel, the LEDs should be 2 mm
- Fast signals are stretched so that they will light the LEDs for long enough to be seen.

### **3.14.4 Rear Panel Input and Outputs**

- 1) VME P1/J1, 5 Row 160 pin
- 2) VME P0/J0, 5 + 2 row hard metric 2 mm, 95 pins
- 3) VME P2/J2, 5 Row 160 pin
- 4) VME P3/J3, 5 Row 160 pin

### **3.14.5 Test and Set-up Points**

There are logic analyser probe points on various data paths to aid in debugging the module. These include the DAV\* and Link Ready signals for each G-Link. A JTAG boundary scan port is provided.

### **3.14.6 Ground Points**

Ground points are provided for scope probe grounding in exposed areas of the motherboard.

### **3.14.7 Power Requirements from VME**

Voltage rail	Current/VME Slot
+5V	10A
+3.3V	12A
48V	2A

Note:

1. All additional voltages (3.3V, 2.5V) are derived from the 48V via DC-DC converters. This is done to reduce the current requirement at lower voltages, avoiding the need for VME64xP.
2. Estimated power dissipation of each module is approximately 100 Watts.
3. It is assumed that a maximum of 11 modules can reside in a ROD crate (limited by the 48V supply).

### **3.14.8 Crate Power Requirements**

1. 200 amps @ 5V (two bricks)
2. 200 amps @3.3 V (two bricks)
3. 24 amps @48V (two bricks)

## 4 Programming Model

### 4.1 Guidelines

These are to aid the software control of the module.

1. All registers can be read by the computer, hence there are no 'write only' registers.
  - All Status Registers shall be Read-Only registers.
  - All Control Registers shall be Read/Write registers.
  - Pulse registers shall return 0 when read.
2. If the computer tries to read or write a register which the module itself is able to modify at the same time, the data integrity cannot be guaranteed. The only exceptions are the Control Register and the Status Register. Therefore it is recommended that before any software tries to read or write to any registers the status of the module should be already known or be ascertained by reading the Status Register.
3. When the address space occupied by the module is accessed, it always responds with a handshake to avoid a bus error.
4. The power-up condition of all registers is all zeros, unless otherwise stated.
5. Where possible, bits are assigned to registers by logical function rather than trying to minimise the register count.

All memory on the card is directly mapped into VME address space so as to be conveniently accessible by the computer.

### 4.2 Notation

The names of the bit fields within registers are written in *italic*. The names of the signal lines are *italic*. Bit fields are labelled Input to Card and Output from Card rather than Write and Read to make it clear whether it is the card or the computer which is doing the writing.

In this document, a byte is always an 8-bit field, a word is always 16 bits and a long-word is always 32 bits.

Setting a bit field means writing a 1 to it, clearing it means writing a 0.

### 4.3 Addressing Modes

The module conforms to the VME64x standard, and responds to the following VME Address Modifier codes.

AM code	Function
2F	A24/D32 CR/CSR space
09	A32/D32 Single Transfer
0B	A32/D32 Block Transfer

Block transfer mode is used only for mapped memories and FIFOs.

## 4.4 Memory Map for CR/CSR space

The VME64x standard requires a 512k byte CR/CSR space, accessed through AM code 2F. The overall structure is illustrated in [Table 12](#).

Address Range	Content
0x00000 – 0x0007F	VME64 Defined Configuration ROM
0x00083 – 0x00FFF	VME64x Defined Configuration ROM
0x7FBE0 – 0x7FBFF	User Control & Status Registers (CSR)
0x7FC00 – 0x7FFF3	VME64x Defined CSRs
0x7FFF7 – 0x7FFFF	VME64 Defined CSRs

Table 12: VME64x CR/CSR space

The VME64x standard requires configuration space to be A24, D08(O) addressable. Although the ATLAS VME interface specification relaxes this to make only A24, D32 mandatory to meet the standard, the module should respond in CR/CSR space to D08(O), D16, and D32 cycles. Configuration space contains those registers needed to load the FPGA devices, control the geographical address, and read the FPGA firmware version numbers. The CR/CSR space is provided jointly by the VME CPLD and FPGA devices, with pulse functions implemented in the CPLD to allow the module to be reset if the VME FPGA load fails.

Some of the CR/CSR registers are mirrored (with the same bit definitions) in normal VME address space. However, CR/CSR space does not contain mirrors of registers and memories needed to control the module once the geographical address has been established and all FPGA devices loaded.

### 4.4.1 VME64-Defined Configuration ROM

All bytes are read only. Bytes other than those listed should return a zero value when read. Data values in this area are implemented in memory 1 byte wide and addressed at every fourth VME address.

Address Offset	Bytes	Value	Description
0x03	1		Checksum
0x07 – 0x0F	3		ROM Length to Checksum
0x13	1	0x84	CR Data Access Width
0x17	1	0x84	CSR Data Access Width
0x1B	1	2	VME64 CR/CSR space
0x1F	1	43	ASCII C
0x23	1	53	ASCII R
0x27-0x2F			Manufacturers ID
0x33 – 0x3F	4		Board ID
0x43 – 0x4F			Revision

#### 4.4.2 VME64x Defined Configuration ROM

All bytes are read only. Bytes other than those listed should return a zero value when read. Data values in this area are implemented in memory 1 byte wide and addressed at every fourth VME address.

Address Offset	Bytes	Value	Description
0xB3 – 0xBB	3	0x7FBE0	Offset to beginning of User CSR –pace
0xBF – 0xC7	3	0x7FBFF	Offset to end of User CSR Space
0xE3	1	4	Slave Characteristics (implements P2)
0xF7	1		Interrupter Capabilities (levels asserted)
0x103	1	0x84	Function 0 data access width
0x123 – 0x13F	8	0xA00	Function 0 AM code Mask (AM 09, 0B, not–2F)
0x623 – 0x62F	4		Function 0 address decoder mask

#### 4.4.3 VME64 Defined CSRs

All bytes are read/write. Bytes other than those listed should return a zero value when read. Data values in this area are implemented in registers 1 byte wide and addressed at every fourth VME address.

Byte Address (hex)	Type	Register Name	Size in bytes	Description
0x7FFFF	RW	BAR	1	Base Address Register
0x7FFFB	RW	BitSetRegister	1	Bit Set Register
0x7FFF7	RW	BitClearRegister	1	Bit Clear Register

#### 4.4.4 VME64x Defined CSRs

All bytes are read/write. Bytes other than those listed should return a zero value when read. Data values in this area are implemented in registers 1 byte wide and addressed at every fourth VME address.

Byte Address (hex)	Type	Register Name	Size in bytes	Description
0x7FF63-0x7FF6F	RW	Function 0 ADER	4	Function 0 Address/Data compare

#### 4.4.5 User Control & Status Registers (CSRs)

The USER CSR registers are mirrored in normal VME address space, and have the same functions, bit- and byte-ordering as in that space

Byte Address (hex)	Register			Size in bytes	Description
	Loc	Type	Name		
0x7FBE0	V	RO	ModuleId	4	<a href="#">Module Number Register</a>
0x7FBE4	C	RW	ControlPulseReg1	4	<a href="#">Control Pulse Register 1</a>
0x7FBE8	C	RW	AddressControlReg	4	<a href="#">Addressing Control Register</a>
0x7FBEC	C	RO	StatusReg1	4	<a href="#">Status Register 1</a>
0x7FBF0	C	RO	VmeId	4	<a href="#">VME CPLD firmware revision no</a>

#### 4.5 Memory Map for Normal VME space

The ROD module is addressed using an A32 address bus. The ROD occupies a 128 Mbyte address space, using 27 least significant bits of the A32 address bus. The 5 most significant bits of the 32-bit address are by default be obtained from the Geographical address lines, using the address allocation in [Table 10](#) on page 39.



## 4.6 Motherboard Register Map

The register map is identical for all 9U ROD firmware variants. It contains addresses managed by the VMEILD (C) and VME FPGA(V) in the general motherboard address range, and address blocks for the Input, Switch, and Monitoring FPGAs.

Byte Address (hex)	Register			Size in bytes	Description
	Loc	Type	Name		
00000	V	RO	ModuleNumber	4	Module Number Register
00004		RW	ControlModeReg	4	Control Mode Register
00008	C	RW	ControlPulseReg1	4	Control Pulse Register 1
0000C	V	RW	ControlPulseReg2		Control Pulse Register 2
00010	C	RW	AddressControlReg	4	Addressing Control Register
00014	C	RO	StatusReg1	4	Status Register 1
00018	V	RO	StatusReg2	4	Status Register 2
0001C	V	RW	InterruptControlReg	4	Interrupt Register
00020	C	RO	VmeId	4	VME CPLD firmware revision no
00024	V	RO	VmeFPGAId	4	VME FPGA firmware revision no
00030	V	RW	CanAccessA	4	CAN Access Register A
00034	V	RW	CanAccessB	4	CAN Access Register B
00100	V	RW	TtcrxControl	4	TTCrx Control Register
00104	V	RO	TtcrxStatus	4	TTCrx Status Register
00108		RW	TtcI2Cid	4	TTCI2cId
0010C		RO	TtcBrcst	4	TTC Broadcast
00110		RO	TtcDq	4	TTC DQ Register
00200		RO	TtcDump	32	TTC Dump RAM
00300		RW	Reserved for Ace	40	Reserved for System Ace Registers
40000		RW	InputFPGA0	256K	Input FPGA Register Map, FPGA 0
80000		RW	InputFPGA1	256K	Input FPGA Register Map, FPGA 1
C0000		RW	InputFPGA2	256K	Input FPGA Register Map, FPGA 2
100000		RW	InputFPGA3	256K	Input FPGA Register Map, FPGA 3
140000		RW	InputFPGA4	256K	Input FPGA Register Map, FPGA 4
180000		RW	SwitchFPGA	256K	Switch FPGA Register Map
1C0000		RW	MonitoringFPGA	256K	Monitoring FPGA Register Map

#### 4.6.1 Module Number Register

The ROD module identification register is the unique identifier for the module, and is accessible at the module base address.

All bit fields are outputs from card

Bit	Assignment	Description
0-15	RAL Number (0x2420, 9248 <sub>10</sub> )	Same for each module type, number recorded in RAL register
16-23	Module Serial number	A unique number assigned to each module
24-27	Issue	PCB Issue number, initially 1
28 -31	PCBUpdate	PCB Hardware update no, set by solder pads on the PCB.

#### 4.6.2 Control Mode Register

All bit fields are inputs to card.

Bit	Descriptive Name	Signal–
0 - 2	<i>Clock Source Select</i>	
3	<i>Replay Mode</i>	<u>ENPBK</u>
4	<i>Interrupt Enable</i>	
5–31	<i>Null</i>	

##### **Clock Source Select**

###### **Bit 0**

When this bit is set, the ROD uses the TTC clocks. When this bit is cleared, the ROD uses the on-board 40 MHz oscillator. Trying to set this bit when the TTC is not ready has no effect (the bit remains cleared).

###### **Bit 1-2**

This sets the board clock source. Values have the following meanings:

[2:1]	Clock Selected
00	40 MHz Crystal Oscillator on Board
01	TTC Clock40Des1
10	TTC Clock40Des2
11	TTC Clock40

The default condition is 00. To prevent problems, selection of a TTC source is prohibited if the TTC is not present, or not READY. In this situation the selection reverts to the default when the register is written to.

##### **Enable Replay Mode**

###### **Bit 3**

Writing a 1 to this bit causes the Input FPGAs FIFOs to be used as a data playback memory. In this mode, input from G-Links is ignored, and data from the Input FPGA memory enters the switch FPGA in response to L1A. All other module functions continue unchanged in this mode. Input G-Links are not used. See section 2.6.

### **Enable Interrupts**

#### **Bit 4**

Writing a 1 to this bit allows the module to generate VME interrupts.

### **4.6.3 Control Pulse Register 1**

All bit fields are inputs to card.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>	
0 (LSB)	<i>Reset Motherboard</i>	<i>Power-on Reset</i>	Pulse (200ns)
1	<i>Reset VME FPGA</i>	<i>Reset</i>	Pulse (200ns)
2	<i>Reset Input FPGAs</i>		
3	<i>Reset Switch FPGA</i>		
4	<i>Reset Monitor FPGA</i>		
5	<i>Reset TTC</i>		Pulse (200ns)
6	<i>Reset System ACE</i>		Pulse (200ns)
7	<i>null</i>		
8	<i>null</i>		
9	<i>Reset CAN controller</i>		Pulse (200ns)
10-31	<i>Null</i>		

#### **Reset Motherboard**

##### **Bit 0**

Writing a 1 to this bit resets the module to the power-on state. Writing a zero has no effect. This bit always reads as zero.

#### **Reset VME FPGA**

##### **Bit 1**

Writing a 1 to this bit resets the VME FPGA firmware. Writing a zero has no effect. This bit always reads as zero.

#### **Reset Input FPGAs**

##### **Bit 2**

Writing a 1 to this bit resets the Input FPGA firmware. Writing a zero has no effect. This bit always reads as zero.

#### **Reset Switch FPGA**

##### **Bit 3**

Writing a 1 to this bit resets the Switch FPGA firmware. Writing a zero has no effect. This bit always reads as zero.

#### **Reset Monitor FPGA**

##### **Bit 4**

Writing a 1 to this bit resets the Monitor FPGA firmware. Writing a zero has no effect. This bit always reads as zero.

#### **Reset TTC**

##### **Bit 5**

Writing a 1 to this bit resets the TTC decoder daughterboard and the TTCrx device. Writing a zero has no effect. This bit always reads as zero.

### **Reset System Ace**

#### **Bit 6**

Writing a 1 to this bit resets the System Ace controller. Writing a zero has no effect. This bit always reads as zero.

### **Reset CAN controller**

#### **Bit 9**

Writing a 1 to this bit resets the Can controller to the power on state. Writing a zero has no effect. This bit always reads as zero.

## **4.6.4 Control Pulse Register 2**

All bit fields are inputs to card.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>	
0	<i>Reset FPGA FIFOs</i>	<u><i>FIFO-Reset</i></u>	Pulse (200ns)
1	<i>Reset Switch Spy memories</i>	<u><i>Spy-Reset</i></u>	Pulse (200ns)
2	<i>Clear Errors</i>	<u><i>CLRPE</i></u>	Pulse (200ns)
3	<i>Generate software L1A</i>		Pulse (200ns)
4 -31	<i>Null</i>		

### **Reset FPGA FIFOs**

#### **Bit 0**

Writing a 1 to this bit resets all FIFOs to the empty state in all Input, Switch, and Monitoring FPGAs. Writing a zero has no effect. This bit always reads as zero.

### **Reset Spy Memories**

#### **Bit 1**

Writing a 1 to this bit resets input memories, flags, and processing in the Monitoring FPGA. Writing a zero has no effect. This bit always reads as zero.

### **Clear Errors**

#### **Bit 2**

Writing a 1 to this bit clears all error indicators and counters on the module. Writing a zero has no effect. This bit always reads as zero.

### **Generate software L1A**

#### **Bit 3**

Writing a 1 to this bit causes the ROD to generate a software L1A, which is intended for use in replay mode. Writing a zero to has no effect. This bit always reads as zero.

When the bit is written, the module should behave as if an L1A had been received via the TTC system, with the Bunch Crossing Number, Event Number, and Trigger Type taken respectively from the Replay\_BCNC, Replay\_EventNo and Reply\_TrigType registers. Replay\_EventNo should be incremented after being used.

## **4.6.5 Addressing Control Register**

All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0 (LSB)	<i>System Ace Configuration Mode</i>	
1 – 3	<i>System Ace Configuration Address</i>	
4 – 11	<i>Geoadd Bypass</i>	
12-31	<i>Null</i>	

### **System Ace Configuration Mode**

#### **Bit 0**

This bit drives the System ACE pin CFGMODEPIN. When this bit is set the ACE Controller will start the configuration process immediately following a System ACE reset. When clear, the ACE Controller will not start the configuration process after a reset until the CFGSTART bit is set in the CONTROLREG register of the ACE MPU interface. By default this bit is set.

### **System Ace Configuration Address**

#### **Bits 1-3**

This field contains the firmware collection number, as listed in [Table 11](#). A different collection may be loaded under software control by writing a new configuration number and then setting the Reset Motherboard bit in [Control Pulse Register 1](#).

### **Geoadd Bypass**

#### **Bits 4-11**

This field contains the module geographical address (5-bit slot number, 2-bit crate number) used at module power-up to determine the TTC and CAN addresses and the firmware collection to be loaded. A different geographical address (used after power-up for the TTC and CAN addresses only) may be set under software control by writing the new value to this field and then setting the Reset Motherboard bit in [Control Pulse Register 1](#). The firmware collection number is not altered by this field after power up – use the System Ace Configuration Address in this register to do this.

## **4.6.6 Status Register 1**

All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0	<i>VME FPGA load Done</i>	
1 – 5	<i>Input FPGAs 0-4 loaded</i>	<u><i>DONE_INP0-4</i></u>
6	<i>Switch FPGA loaded</i>	<u><i>DONE_SW</i></u>
7	<i>Monitor FPGA loaded</i>	<u><i>DONE_MON</i></u>
8	<i>System ACE Done</i>	
9	<i>System ACE Error</i>	
10	<i>System ACE POR Test</i>	
11–31	<i>null</i>	

### **VME FPGA loaded**

#### **Bit 0**

This bit is set if the VME FPGA configuration has loaded successfully.

**Input FPGAs 0-4 loaded**

Bit 1-5

This bit is set if all Input FPGA configurations have loaded successfully.

**Switch FPGA loaded**

Bit 6

This bit is set if the Switch FPGA configuration has loaded successfully.

**Monitor FPGA loaded**

Bit 7

This bit is set if the Monitor FPGA configuration has loaded successfully.

**System Ace done**

Bit 8

This bit is set to 0 when the configuration operation is complete and to 0 when the configuration is idle. While configuration is underway, the bit alternates from 0 to 1.

**System Ace Error**

Bit 9

This bit is set to 0 if an error occurs and to 1 if no errors are detected. The value alternates between 0 and 1 if no Compact Flash device is found.

**System Ace POR Test**

Bit 10

This bit is set to 1 while the System ACE is performing the power-on reset and to 0 when the reset is complete.

**4.6.7 Status Register 2**

All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0 (LSB)	<i>Combined G-Link Error</i>	
1	<i>TTC Ready</i>	<u><i>TTC_READY</i></u>
2	<i>TTC selected</i>	<u><i>TTC_S2</i></u>
3-31	<i>null</i>	

**Combined G-Link Error**

Bit 0

This bit is set to 1 if at least one of the Input FPGA G-Link error counters is non-zero. It is cleared when all those counters are zero.

**TTC Ready**

Bit 1

This bit is set to 1 if the TTCrx chip reports ready (indicating lock to TTC clock).

**TTC Selected**

Bit 2

This bit is set to 1 if the TTC clock is selected and enabled for the module or to 0 if the on-board oscillator is in use.

#### 4.6.8 Interrupt Register

This register controls the interrupts from this module. The bit field will be defined later

#### 4.6.9 VME CPLD firmware revision no

This register contains the type and revision number of the CPLD providing the interface to VME64. The first revision is number 1. All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-15	Module Type (0x2420)	
16-23	CPLD Code Type (12)	
24-31	CPLD Code Revision	

##### Module Type

Bits0-15

These 16 bits return the module type (0x2420 for the 9U ROD).

##### CPLD Code type

Bits 16-23

These eight bits fields are set to 12. They are part of a more general scheme to describe the firmware types, using the values in [Table 2](#). The code type refers to the VME CPLD.

##### FPGA Code Revision

Bit 24-31

This 8-bit field contains the CPLD code revision number. The first revision is numbered 1.

#### 4.6.10 VME FPGA firmware revision no.

This register contains the type and revision number of the FPGA providing the interface to VME64. The first revision is number 1. All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-15	Module Type (0x2420)	
16-23	FPGA Code Type (16)	
24-31	FPGA Code Revision	

##### Module Type

Bits0-15

These 16 bits return the module type (0x2420 for the 9U ROD).

##### FPGA Code type

Bits 16-23

This eight-bit field is set to 16. It is part of a more general scheme to describe the firmware types, using the values in [Table 2](#). The code type refers to the VME FPGA.

##### FPGA Code Revision

Bit 24-31

This 8-bit field contains the VME FPGA code revision number. The first revision is numbered 1.



#### 4.6.11 CAN Access Register A

Bit	Descriptive Name	Signal name
0	<i>Can Reset</i>	
1	<i>Can Program</i>	
2-31	<i>Null</i>	

##### **CAN Reset**

###### **Bit 0**

Writing a 1 to this bit resets the CANBus processor. Writing a zero has no effect. This bit always reads as zero.

##### **CAN Program**

###### **Bit 1**

Writing a 1 to this bit places the CANBus processor in programming mode. Programming is done using the D15 connector at the top of the S-Link rear transition module. Writing a zero places the processor in running mode. The CANBus processor should always be reset (by setting the CAN Reset bit) at the same time as a change of mode. The default register setting is zero (CANBUS running).

#### 4.6.12 CAN Access Register B

Bit	Descriptive Name	Signal name
0 – 3	<i>Can Interrupt Request</i>	<u><i>CAN_IRO[0:3]</i></u>
4-31	<i>Null</i>	

##### **CAN Interrupt Request**

###### **Bits 0-3**

Writing a 1 to one of these bits causes an interrupt request to the CANBus processor at the corresponding priority. Writing a zero has no effect. These bits always read as zero.

#### 4.6.13 TTCrx Control Register

All bit fields are inputs to card. Power-up condition initially sets all bits to 0. A TTCrx I/O operation takes place whenever this register is changed, unless the I2C bus is busy (see TTC Status register below). An I2C operation is aborted if the reset TTCrx Controller bit is set.

Bit	Descriptive Name	Signal name
0-7	<i>Data to TTCrx</i>	<u><i>TTCWDATA</i></u>
8	<i>Write</i>	<u><i>TTCWRITE</i></u>
9-13	<i>TTC Register Number</i>	<u><i>RRCREGNO</i></u>
14	<i>Unused</i>	
15	<i>Reset TTCrx Controller</i>	<u><i>TTCRESET</i></u>
16-31	<i>Null</i>	

##### **Data to TTCrx**

###### **Bits 0-7**

This 8-bit field contains data to be written to the TTCrx chip.

**Write****Bit 8**

When set to 1, defines the operation as a write to TTCrx. When set to 0, the operation is a read.

**TTC Register Number****Bits 9-13**

Specify the TTCrx register number to be read or written

**Reset TTCrx Controller****Bit 15**

When set to 1, resets the TTC controller logic and aborts any I2C operation in progress.

**4.6.14 TTCrx Status Register**

All bit fields are outputs from card. Power-up condition will initially set all bits to 0.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-7	<i>Data from TTCrx</i>	<u><i>TTCRDATA</i></u>
8	<i>I2C Busy</i>	<u><i>I2CBUSY</i></u>
9	<i>I2C Error</i>	<u><i>I2CERR</i></u>
10-31	<i>Unused</i>	

**Data from TTCrx****Bits 0-7**

This 8-bit field contains data read from the TTCrx chip.

**I2C Busy****Bit 8**

When set to 1, indicates that an I2C transaction is underway

**I2C Error****Bits 9**

When set to 1, indicates that an I2C error has occurred

**4.6.15 TTCI2cId Register**

All bit fields are inputs to card.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-5	<i>TTCrx I2C ID</i>	<u><i>TtcI2cId</i></u>
6-31	<i>Unused</i>	

**TtcI2cId**

The TTC base ID used by the I2C FPGA to communicate with the TTCrx. This must be set via VME before I2C communication can be established. It should be set to match the content of the TTCrx base address register I2C\_ID (0:5) (see TTCrx specification).

#### 4.6.16 TTC Broadcast Data Register

All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-1	<i>Null</i>	
2-5	<i>TTC BRCST data (2: 5)</i>	<u><i>TTCBRCST (2: 5)</i></u>
6-7	<i>TTC BRCST data (6:7)</i>	<u><i>TTCBRCST (6: 7)</i></u>
8-31	<i>Unused</i>	

##### **TTC BRCST data (2:5)**

The most recent value of BRCST data bits (2:5) output from the TTCrx (that were qualified by the TTC strobe BRCSTSTR1).

##### **TTC BRCST data (6:7)**

The most recent value of BRCST data bits (6:7) output from the TTCrx (that were qualified by the TTC strobe BRCSTSTR2).

#### 4.6.17 TTC DQ Register

All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-3	<i>TTC DQ</i>	<u><i>TTCDQ</i></u>
4-31	<i>Unused</i>	

##### **TTC DQ**

The most recent DQ value output from the TTCrx (that was qualified by the TTC strobe DOUTSTR).

#### 4.6.18 TTC Dump RAM

All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-7	<i>TTC Dump Data</i>	<u><i>TTCDump</i></u>
8-31	<i>Unused</i>	

##### **TTC Dump Data**

A block of RAM 16 words deep that captures data from TTC Error and Configuration Dumps. Data from the TTC are mapped to the RAM using DQ as the address.

## 4.7 Input FPGA Register Map

The registers are chosen to suit all firmware types, so the following registers are used in all types of ROD

Byte Offset (hex)	Register type	Register Name	Size in bytes	Description
00000	RO	<i>FPGA Version</i>	4	Input FPGA Common Version
00004	RO	<i>FPGA Status</i>	4	Input FPGA Status Register
00008	RO	<i>G-Link Error Map</i>	4	Input G-Link Error Map
0000C	RO	<i>G-Link Error Count</i>	4	Input G-Link Error Count
00010	RW	<i>Control Pulse</i>	4	Input Control Pulse Register
00014	RW	<i>DataFifoBusyDepth</i>	4	Input Data FIFO Busy Depth
00018	RW	<i>EventMntBusyDepth</i>	4	Input Event Management FIFO Busy Depth
0001C	RW	<i>BCNBusyDepth</i>	4	Input BCN FIFO Busy Depth
Channel 0 Registers				
00800	RO	<i>Channel 0 Version</i>		Input Channel Version Number
00804	RW	<i>Chan 0 Control</i>	4	Input Channel Control Register
00808	RO	<i>Chan 0 Status</i>	4	Input Channel Status Register
0080C	RW	<i>Chan 0 Slice Cnt</i>	4	Input Channel Expected Slice Count
00810	RW	<i>Chan 0 Subslice</i>	4	Input Channel Subslice Control
00814	RW	<i>Chan 0 Compressn</i>	4	Input Channel Compression Control
00818	RW	<i>Chan 0 SourceId</i>	4	Input Channel Source ID
0081C	RW	<i>Chan 0 Neutral Id</i>	4	Input Channel Neutral ID
Channel 0 FIFO pointers and access				
00820	RW	<i>Chan 0 EvtMgtPtrs</i>	4	Input Channel Event Management FIFO address pointers
00824	RO	<i>Ch 0 BCNFifoPtrs</i>	4	Input Channel Bunch Number FIFO address pointers
00828	RO	<i>Chan 0 Data Fifo ptrs</i>	4	Input Channel Data FIFO address pointers
0082C	RW	<i>Chan 0 Fifo Mnt0</i>	4	Input Channel FIFO Depth Monitor Registers
00830	RW	<i>Chan 0 Fifo Mnt1</i>	4	Input Channel FIFO Depth Monitor Registers
01000	RW	<i>Ch 0 EvtMgt Access</i>	2048	Input Channel Event Management FIFO Access

01800	RW	<i>Ch 0 BCN Fifo Access</i>	64	<a href="#">Input Channel Bunch Number FIFO Access</a>
02000	RW	<i>Ch 0 Reserved</i>		
08000	RW	<i>Ch 0 Data FIFO Access</i>	32768	<a href="#">Input Channel Data FIFO Access</a>
010800	RW	<i>Chan 1 Registers and FIFOs</i>	48	<a href="#">Registers for Channel 1</a>
020800	RW	<i>Chan 2 Registers and FIFOs</i>	48	<a href="#">Registers for Channel 2</a>
030800	RW	<i>Chan 3 Registers and FIFOs</i>	48	<a href="#">Registers for Channel 3</a>

#### 4.7.1 Input FPGA Common Version

This register describes the formatting type and revision number of the Input FPGA common firmware. All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-15	<i>Module Type (0x2420)</i>	
16-23	<i>FPGA Code type (13)</i>	
24-31	<i>FPGA Code Revision no</i>	

##### Module Type

Bits0-15

These 16 bits return the module type (0x2420 for the 9U ROD).

##### FPGA Code type

Bits 16-23

This eight-bit field is set to 13. It is part of a more general scheme to describe the firmware types, using the values in [Table 2](#). The common code type refers to the shared FPGA code which provides the interfaces to the G-Link, VME, Switch FPGA, TTC, and control functions.

##### FPGA Code Revision

Bit 24-31

This 8-bit field contains the Input FPGA common code revision number. The first revision is numbered 1. Revisions are numbered separately for each code type.

#### 4.7.2 Input FPGA Status Register

This register provides status information from resources in the Input FPGA common firmware. All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0	<i>DLL Locked</i>	
1	<i>BCN FIFO Full</i>	
2	<i>BCN FIFO Empty</i>	
3	<i>BCN FIFO Busy</i>	
1-31	<i>Null</i>	

**DLL Locked****Bit 0**

When set to 1, this bit indicates that the internal DLL in the FPGA has successfully locked to the main board clock.

**BCN FIFO Full****Bit 1**

When set to 1, this bit indicates that at least one of the Input FPGA BCN FIFOs is full.

**BCN FIFO Empty****Bit 2**

When set to 1, this bit indicates that at least one of the Input FPGA BCN FIFOs is empty.

**BCN FIFO Busy****Bit 3**

When set to 1, this bit indicates that at least one of the Input FPGA BCN FIFOs is busy (i.e. not empty).

**4.7.3 Input G-Link Error Map**

All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-19	<i>Failing G-Link data pin</i>	
20-23	<i>Failing G-Link data Map</i>	
24-27	<i>Failing G-Link Link Map</i>	
28-31	<i>G-Link down Map</i>	<u>0</u>

This register contains error indicators for enabled G-Links only.

**Failing G-Link Data Pin****Bits 0-19**

This 19-bit field contains one bit for each G-Link bit D0-D19. A bit in this field is set whenever a longitudinal parity error has been detected on the corresponding G-Link pin of any of the four input G-Links. This field is cleared by the G-Link Error Reset bit in the input FPGA control pulse register.

**Failing G-Link Data Map****Bit 20-23**

This 4-bit field contains one bit for each G-Link connected to this FPGA. A bit corresponding to the G-Link number is set whenever a parity error is detected on the corresponding G-Link. This field is cleared by the G-Link Error Reset bit in the input FPGA control pulse register.

**Failing G-Link Protocol****Bits 24-27**

This 4-bit field contains one bit for each G-Link connected to this FPGA. A bit corresponding to the G-Link number is set whenever a protocol error is detected on the corresponding G-Link. This field is cleared by the G-Link Error Reset bit in the input FPGA control pulse register.

### **G-Link Link Lost**

**Bits 28-31**

This 4-bit field contains one bit for each G-Link connected to this FPGA. A bit corresponding to the G-Link number is set whenever a link down error is detected on the corresponding G-Link. This field is cleared by the G-Link Error Reset bit in the input FPGA control pulse register.

#### **4.7.4 Input G-Link Error Count**

All bit fields are outputs from card.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-15	<i>Counted G-Link errors</i>	
16-31	<i>Counted G-Link link down</i>	

### **Counted G-Link Errors**

**Bit 0-15**

This 16-bit register counts 40 MHz clock ticks on which any of the G-Link data parity or protocol errors are present, ignoring disabled G-Links. On power-up, all bits are set to 0, but the counter will count errors until all enabled links are transmitting valid data. The counter is reset by the G-Link Error Reset bit in the input FPGA control pulse register.

### **Counted G-Link Link Down**

**Bit 16-31**

This 16-bit register counts 40 MHz clock ticks on which any of the enabled G-Links is asserting link down, ignoring disabled G-Links. On power-up, all bits are set to 0, but the counter will increment until all enabled links are stable. The counter is reset by the G-Link Error Reset bit in the input FPGA control pulse register.

#### **4.7.5 Input Control Pulse Register**

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-3	<i>G-Link Reset, device 0-3</i>	
4	<i>Clear Errors</i>	
5	<i>Clear FIFOs</i>	
6	<i>Reset FIFO Monitor</i>	
7-31	<i>NULL</i>	<u>0</u>

### **G-Link Reset**

**Bit 0**

Writing a 1 to this bit resets the corresponding G-Link attached to this FPGA. Writing a zero has no effect. This bit always reads as zero.

### **Reset Errors**

**Bit 4**

Writing a 1 to this bit clears all error indicators and counters on this FPGA. Writing a zero has no effect. This bit always reads as zero.



### **Clear FIFOs**

#### **Bit 5**

Writing a 1 to this bit will empty all data from all input FIFOs in this FPGA. Writing a zero has no effect. This bit always reads as zero. The results are undefined if this bit is pulsed while events are entering the device.

### **Reset FIFO Monitor**

#### **Bit 6**

Writing a 1 to this bit will zero the [Input Channel FIFO Depth Monitor Registers](#). This bit always reads as zero.

## **4.7.6 Input Data FIFO Busy Depth**

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-12	<i>Data Busy Depth</i>	
13-31	<i>NULL</i>	<i>0</i>

### **Data Busy Depth**

#### **Bit 0-12**

This field sets the depth of data in the event FIFO above which BUSY is generated. If set to the maximum value, 0x1FFF, BUSY cannot be generated from this FIFO.

## **4.7.7 Input Event Management FIFO Busy Depth**

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-8	<i>Management FIFO Busy Depth</i>	
8-31	<i>NULL</i>	<i>0</i>

### **Management FIFO Busy Depth**

#### **Bit 0-8**

This field sets the number of events in the event management FIFO above which BUSY is generated. If set to the maximum value, 0x1FF, BUSY cannot be generated from this FIFO.

## **4.7.8 Input BCN FIFO Busy Depth**

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-8	<i>BCN Busy Depth</i>	
9-31	<i>NULL</i>	<i>0</i>

### **BCN Busy Depth**

#### **Bit 0-8**

This field sets the depth of data in the BCN FIFO above which BUSY is generated. If set to the maximum value, 0x1FF, BUSY cannot be generated from this FIFO.

#### 4.7.9 Input Channel Version Number

All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-15	Module Type (0x2420)	
16-23	FPGA Code type	
24-31	FPGA Code Revision no	

##### Module Type

Bits 0-15

These 16 bits return the module type (0x2420 for the 9U ROD).

##### FPGA Code type

Bits 16-23

These eight bits describe the data formatting capability per G-Link. Values are given in [Table 2](#).

##### FPGA Code Revision

Bit 24-31

This 8-bit field contains the Input FPGA code revision number. The first revision is numbered 1. Revisions are numbered separately for each code type.

#### 4.7.10 Input Channel Control Register

Bit	Descriptive Name	Signal name
0	G-Link enable	
1	Mode 20 Select	
2	BCN FIFO Disable	
3-31	NULL	<u>0</u>

##### G-Link Enable

Bit 0

When set to 1, this 1-bit field enables the corresponding input G-Link. In replay mode, this bit enables replay from the Input FPGA FIFO into the S-Link streams. When set to 0, the bit enables VME access to the Data FIFO and EVENT Management FIFO, and their pointers. On power-up, the bit should be set to 0, so that the G-Link input is disabled to prevent noise injection into the FPGA.

##### Mode 20 Select

Bit 1

This bit determines the G-Link data width. When set to 1, the width is 20 bits. When set to 0, the width is 16 bits.

##### BCN FIFO Disable

Bit 2

When set to 1, this bit disables the BCN FIFO filling and emptying for the channel, and enables VME access to the BCN FIFO & pointers.

#### 4.7.11 Input Channel Status Register

Bit	Descriptive Name	Signal name
0	<i>G-Link lock</i>	
1	<i>Data FIFO Full</i>	
2	<i>Event Monitor Full</i>	
3	<i>Data FIFO Empty</i>	
4	<i>Event Monitor Empty</i>	
5	<i>Data FIFO Busy</i>	
6	<i>Event monitor Busy</i>	
7-31	<i>NULL</i>	<u>0</u>

##### **G-Link lock**

Bit 0

This bit is set if the G-Link associated with the channel is asserting Link ready.

##### **Data FIFO Full**

Bits 1, 3, 5

These bits are set if the data FIFO associated with the channel is full, empty, or partially full respectively.

##### **Event Monitor Full**

Bits 2, 4, 6

This bit is set if the Event Monitor associated with the channel is full, empty, or partially full respectively.

#### 4.7.12 Input Channel Expected Slice Count

Bit	Descriptive Name	Signal name
0-2	<i>Primary Slice Count</i>	
3-7	<i>null</i>	
8-12	<i>Secondary Slice Count</i>	
13-31	<i>NULL</i>	<u>0</u>

##### **Primary Slice Count**

Bits 0-3

This field contains the expected number of slices of module readout, and is used to populate the NS1 and NS2 fields in the sub-block header for the channel. For the PreProcessor, this register determines the number of LUT slices read. The maximum permitted value is 7 and the minimum 1. See note on secondary slice count for handling larger values.

##### **Secondary Slice Count**

Bits 8-11

This field is set nonzero by software only for the PreProcessor. It contains the expected number of FADC slices. The maximum permitted number is 31 and the minimum 0. If it becomes necessary to handle larger numbers of slices than the bit fields will accommodate, both NS1 and NS2 will be set to zero to indicate that the slice counts used are stored elsewhere.

#### 4.7.13 Input Channel Subslice Control Register

Bit	Descriptive Name	PPM	CPM	JEM	CMM
0-4	<i>Subslice Map comp 0</i>	FADC	Ser Input	LVDS In	BP Input
5-9	<i>Subslice Map comp 1</i>	BCID_LUT	Hits	Hits	Local Sum
10-14	<i>Subslice Map comp 2</i>	-	-	-	Remote In
15-19	<i>Subslice Map comp 3</i>	-	-	-	Sys Sum
20-24	<i>Subslice Map, comp4</i>	-	-	-	-
25-29	<i>Subslice Map, comp 5</i>	-	-	-	-
30-31	<i>NULL</i>	-	-	-	-

#### **Subslice Map, components 0-5**

Bits 0-4, etc

Each module may provide up to six readout components. Except for the Pre-processor, each component produces the same number (up to five) of sub-slices of G-Link data. Selected slices may be discarded in the ROD on a per-component, per-slice basis. The PreProcessor may provide separately controlled numbers of slices for its two components (FADC and BCID\_LUT output).

This register controls the elimination of sub-slices from module readout via a 5-bit field for each component. A sub-slice from the component is discarded if the corresponding bit is set in this register. For example, if the register is set to 0x400, the first sub-slice from component two is discarded. The components of source modules are as indicated in the table.

#### 4.7.14 Input Channel Compression Control Register

This register controls data compression during formatting. The normal setting for all inputs is Zero Suppression/Compression only, which in the case of the PPM provides bit-field compression. Zero suppression / Compression with threshold is used only for PPM data. When neutral format is use, these settings are ignored.

Bit	Descriptive Name	Signal name
0-1	<i>Compression Level</i>	
2-3	<i>Null</i>	
4-7	<i>LUT FADC Slice Position</i>	
8-15	<i>FADC Baseline Lower Bound</i>	
16-31	<i>Compression Threshold</i>	

#### **FPGA Compression Level**

Bits 0-1

This 2-bit field contains the compression type as follows:

Compression Level	Function
0	No Compression
1	Zero Suppression / Compression only
2	Zero Suppression/ Compression and Threshold

### **LUT FADC Slice Position**

#### **Bits 4-7**

This 4-bit field contains the FADC slice number (0 being the first slice) corresponding to the LUT value. For readout of 5 slices with the FADC peak in the third slice, the value should be 2. The value is used in compression to suppress the 'External BCID' status bits for the FADC data if they can be deduced from the LUT information.

### **FADC Baseline Lower Bound**

#### **Bits 8-15**

This 8-bit field contains the lower bound of the window for encoding the observed FADC minimum value for a channel as a short field during data compression. This lower bound is used for all 64 PPM channels on the associated G-Link input.

### **Compression Threshold**

#### **Bits 16-31**

This 16-bit field contains the compression threshold used in PreProcessor data compression level 2 mode. The data block for a channel is suppressed if the LUT output is zero and the raw FADC data values are all less than this threshold.

### **4.7.15 Input Channel Source ID Register**

This register holds the crate and module number connected to the input G-Link, for insertion into the output data. Any values arriving in any G-Link frame should be discarded and replaced by values from this register. All bit fields are input to card.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-3	<i>Crate Number</i>	
4-7	<i>Null</i>	<u>0</u>
8-11	<i>Module Slot Number</i>	
12-31	<i>Null</i>	<u>0</u>

### **4.7.16 Input Channel Neutral ID**

This register sets the Word-IDs to be used for sub-block header, sub-status, and data payload in neutral format. Software should load the appropriate values from [Table 14](#).

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-3	<i>Neutral Header ID</i>	
4-7	<i>Neutral Payload ID</i>	
8-31	<i>Null</i>	

### **Neutral Header ID**

#### **Bits 0-3**

This 4-bit field contains the Word-ID to be used for neutral sub-block headers and sub-status words. The firmware uses the value as given for sub-headers. For sub-status words, firmware replaces the least significant bit by 1.

### **Neutral Payload ID**

Bits 4-7

This 4-bit field contains the Word-ID to be used for the data payload of neutral format data.

#### **4.7.17 Input Channel Event Management FIFO address pointers**

This register provides read-only access to the internal FIFO address counters.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-12	<i>Write pointer</i>	
13-15	<i>null</i>	<u>0</u>
16-28	<i>Read pointer</i>	<u>0</u>
29-31	<i>null</i>	

#### **Event Management FIFO Pointers**

These 13-bit fields contain the offset in FIFO memory to the address to be used for the next write or read operation. The event management FIFO contains the lengths of events awaiting transfer to the Switch FPGA

#### **4.7.18 Input Channel Bunch Number FIFO address pointers**

This register provides read-only access to the internal FIFO address counters. All bit fields are output from card.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-12	<i>Write pointer</i>	
13-15	<i>null</i>	<u>0</u>
16-28	<i>Read pointer</i>	<u>0</u>
29-31	<i>null</i>	

#### **Bunch Number FIFO Pointers**

These 13-bit fields contain the offset in FIFO memory to the address to be used for the next write or read operation. The Bunch Number FIFO contains the bunch numbers used to check incoming G-Link data against the L1A bunch number from the TTC.

#### **4.7.19 Input Channel Data FIFO address pointers**

This register provides read-only access to the internal FIFO address counters. All bit fields are output from card.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-12	<i>Write pointer</i>	
13-15	<i>null</i>	<u>0</u>
16-28	<i>Read pointer</i>	<u>0</u>
29-31	<i>null</i>	

### Channel Data FIFO Pointers

These 13-bit fields contain the offset in FIFO memory to the address to be used for the next write or read operation. The Channel Data FIFO contains the event data in S-Link format awaiting transfer to the Switch FPGA.

#### 4.7.20 Input Channel FIFO Depth Monitor Registers

These registers provide a monitor of the occupied depth of the Input FPGA data FIFOs, using an 8-bin histogram. A signal is sent by the formatting logic as the last word of an event is inserted into the data FIFOs. The common logic computes the occupied depth of the FIFO at this moment (by subtracting the read and write pointers), and then divides this value by the length of the FIFO and multiplies by 8. The result is a value between 0 and 7, which is used to address an array of 8 bytes forming a histogram. The addressed byte is incremented by 1. Whenever one of the bytes reaches 255, ALL of the bins should be divided by 2. The 8 bins are read through the two registers, starting with the first bin in bits 0-7 of the first register and the remainder in the second register. The bins are cleared by writing 1 to the Reset FIFO Monitor bit in the [Input Control Pulse Register](#).

#### 4.7.21 Input Channel Event Management FIFO Access

These addresses provide read-write access to the internal FIFO used to manage the events in the data FIFO. The information includes the lengths of event blocks awaiting transfer to the Switch FPGA. Each word is formatted as follows

Bit	Descriptive Name
0-12	<i>Length in words of block in Data FIFO, including subheader &amp; sub-status.</i>
13	<i>BCN mismatch flag</i>
14	<i>Data FIFO overflow flag</i>
15	<i>LVDS error flag</i>
16	<i>G-Link error flag</i>
17-31	<i>null</i>

Bits 0-12 are used by the Switch input logic to determine where the data blocks stop. Bits 13-16 are sent to the Switch to build Status Word 1.

#### 4.7.22 Input Channel Bunch Number FIFO Access

These addresses provide read-write access to the internal FIFO storing the bunch numbers to be used for input G-Link event checking.

Byte offset (hex)	Contents
0-0x3D	Bits <12:31> = 0; bits <0:11>=bunch number

#### 4.7.23 Input Channel Data FIFO Access

These addresses provide read-write access to the internal FIFOs. The FIFOs are implemented as dual-port memories, one assigned to the Switch FPGA and one shared between VME and G-Link. G-Link access must be disabled (in the [Input Channel Control Register](#)) to enable VME access.



## 4.8 Switch FPGA Register Map

Small changes to the memory map and detailed assignment of bits are possible during the trigger commissioning phase to accommodate requirements not foreseen at the time of writing this document.

Byte Offset (hex)	Register type	Register Name	Size in bytes	Description
0000	RO	<i>FPGA Revision</i>	4	Switch FPGA Revision
0004	RW	<i>Switch Control</i>	4	Switch Control Register
0008	RW	<i>Switch Pulse</i>	4	Switch FPGA Pulse Register
000C	RO	<i>Switch Status</i>	4	Switch FPGA Status Register
0010	RW	<i>Run Number</i>	4	Run Number Register
0014	RW	<i>ECR Number</i>	4	ECR Count Register
001C	RW	<i>Replay_BCN</i>	4	Replay BCN Register
0020	RW	<i>Replay_EventNo</i>	4	Replay Event Number Register
0024	RW	<i>Reply_TrigType</i>	4	Replay Trigger Type Register
0028	RO	<i>TTC_SampleId</i>	4	TTC SampleId Register
002C	RW	<i>RoI_Wordlimit</i>	4	RoI Wordlimit Register
0030	RW	<i>S-Link Control</i>	4	S-Link Control Register
0034	RW	<i>S-Link Status</i>	4	S-Link Status Register
0038	RW	<i>S-Link Routing</i>	4	S-Link Routing
004C	RW	<i>SamplingControl</i>	4	Sampling Control
0050	RO	<i>Spy Status</i>	4	Spy Status
0054	RW	<i>TrigTypeTimeout</i>	4	Trigger Type Timeout Register
0058	RW	<i>TtcFifoBusyDepth</i>	4	TTC FIFO Busy Depth
005C	RO	<i>BCNFifoPointer</i>	4	BCN FIFO Pointer Register
0060	RW	<i>G-Link Timeout</i>	4	G-Link Timeout Register
0064	RO	<i>Total_LIA_No</i>	4	Total LIA Number
0068	RO	<i>Last Event LIID</i>	4	Last Event LIID Register
006C	RO	<i>Last TTC LIID</i>	4	Last TTC LIID Register
0074	RW	<i>DetEvTypeLow16</i>	4	Detector Event Type Register
0078	RO	<i>OrbitCount</i>	4	Orbit Counter Register

007C	RW	<i>OrbitWrap</i>	4	Orbit Counter Wrap Value
0080	RW	<i>ECR Delay</i>	4	ECR Delay Register
0084	RO	<i>ECR Error Counter</i>	4	ECR Error Counter
0090	RW	<i>Switch Format Version</i>	4	Format Version Register
00E0	RW	<i>UserHeader1</i>	4	User Header Register 1
00E4	RW	<i>UserHeader2-4</i>	3*4	User Header Registers 2-4
0100	RW	<i>TrigType Buffer</i>	256	TrigType Buffer
0200	RW	<i>BCN_Buffer</i>	512	BCN_Buffer
0400	RW	<i>Event_Number</i>	1k	Event_Number Buffer (256*32 bits)
0800	RW	<i>Orbit Buffer</i>	512	Orbit Number Buffer (256*16 bits)
01000	RW	<i>S-Link 0 LFF time</i>	4	S-Link 0-3 LFF time
01004	RW	<i>S-Link 0 Source Id</i>	4	S-Link 0-3 Source ID
01010	RO	<i>S-Link 0 SpyEvLen01</i>	4	S-Link 0-3 Spy Event Length 01
01014	RO	<i>S-Link 0 SpyEvLen23</i>	4	S-Link 0-3 Spy Event Length 23
08000	RW	<i>Chan 0 Spy_Buffer</i>	32K	S-Link 0-3 Spy Buffers (4*2K*32 bit)
11000	RW	<i>SLink 1 LFF time</i>	4	S-Link 0-3 LFF time
11004	RW	<i>S-Link 1 Source Id</i>	4	S-Link 0-3 Source ID
11010	RO	<i>S-Link 1 SpyEvLen01</i>	4	S-Link 0-3 Spy Event Length 01
11014	RO	<i>S-Link 1 SpyEvLen23</i>	4	S-Link 0-3 Spy Event Length 23
18000	RW	<i>Chan 1 Spy_Buffer</i>	32K	S-Link 0-3 Spy Buffers
21000	RW	<i>SLink 2 LFF time</i>	4	S-Link 0-3 LFF time
21004	RW	<i>S-Link 2 Source Id</i>	4	S-Link 0-3 Source ID
21010	RO	<i>S-Link 2 SpyEvLen01</i>	4	S-Link 0-3 Spy Event Length 01
21014	RO	<i>S-Link 2 SpyEvLen23</i>	4	S-Link 0-3 Spy Event Length 23
28000	RW	<i>Chan 2 Spy_Buffer</i>	32K	S-Link 0-3 Spy Buffers
31000	RW	<i>S-Link 3 LFF time</i>	4	S-Link 0-3 LFF time
31004	RW	<i>S-Link 3 Source Id</i>	4	S-Link 0-3 Source ID

31010	RO	<i>S-Link 3 SpyEvLen01</i>	4	<a href="#">S-Link 0-3 Spy Event Length 01</a>
31014	RO	<i>S-Link 3 SpyEvLen23</i>	4	<a href="#">S-Link 0-3 Spy Event Length 23</a>
38000	RW	<i>Chan 3 Spy_Buffer</i>	32K	<a href="#">S-Link 0-3 Spy Buffers</a>

#### 4.8.1 Switch FPGA Revision

This register describes the revision number of the firmware. All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-15	<i>Module Type (0x2420)</i>	
16-23	<i>FPGA Code type (17)</i>	
24-31	<i>FPGA Code Revision no</i>	

##### Module Type

###### Bits 0-15

These 16 bits return the module type (0x2420 for the 9U ROD).

##### FPGA Code type

###### Bits 16-23

This eight-bit field is set to 17. It is part of a more general scheme to describe the firmware types, using the values in [Table 2](#). The type 14 refers to the Switch FPGA.

##### FPGA Code Revision

###### Bit 24-31

This 16-bit field contains the switch FPGA code revision number. The first revision is numbered 1.

#### 4.8.2 Switch Control Register

This register controls the selection of events from input FPGAs and for sampling in spy buffers.

Bit	Descriptive Name	Signal name
0	<i>Sample All Events</i>	
1-5	<i>Enable Input FPGA</i>	
6-7		
8-12	<i>User Header Disable</i>	
13	<i>Ignore TTC FIFO Empty</i>	
14	<i>Enable ECR Error Reset</i>	
15	<i>Synchronous Sampling Mode</i>	
16-31	<i>Null</i>	

##### Sample All Events

###### Bit 0

When set to 1, this bit causes internal LFF to be generated whenever all spy memory buffers are occupied. The internal LFF should be removed when at least one spy buffer is free.

**Enable Input FPGA****Bit 1-5**

These bits are used to control data routing in those readout schemes where the mapping between Input FPGAs and S-Links is not one-to-one, i.e., where data from several Input FPGAs are cascaded to one S-Link. When a bit here is set to 1, the data from the corresponding Input FPGA are included in the data cascade; otherwise that Input FPGA is not read out. Except for the last Input FPGA, which is only ever routed to an S-Link via a cascade, these bits have no effect when one-to-one mapping schemes are used. For this reason these bits should not be used to mask out active data channels (use the Input Channel Control Register for this).

**Input Header Disable****Bits 8-12**

Setting a bit to 1 disables the insertion of event headers for the channels from the corresponding Input FPGA.

**Ignore TTC FIFO empty****Bit 13**

For normal operation this bit should be 0. When it is set to 1, the Switch logic ignores the status of the FIFOs containing the TTC information and acts as if they always contain data. The TTC information in the event header is thus unreliable and the readout process is driven entirely by the receipt of data on the G-Links. This mode of operation allows a large part of the readout logic to be tested in isolation from a TTC system.

**Enable ECR Error Reset****Bit 14**

When set to 1, this bit enables the error recovery logic, described in section [2.2.9](#), which removes fragments of incomplete events. When set to 0, the error-handling logic is disabled. Logic to reset the event counter is always enabled and is not affected by the setting of this bit.

**Synchronous Sampling Mode****Bit 15**

When set to 1, this bit enables synchronous sampling of events. In this mode, the spy sampling buffers are reset only by TTC short broadcast 0x04, and VME command by setting the Clear all Buffers bit in the Switch FPGA Pulse Register is ignored. When set to zero, the TTC short broadcast is ignored and the sampling buffers are reset only by VME command.

**4.8.3 Switch FPGA Pulse Register**

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0	<i>Clear all Buffers</i>	
1	<i>Clear orbit counter</i>	
2	<i>Clear TTC FIFOs</i>	
3	<i>Fake L1A</i>	
4	<i>Clear L1A Counter</i>	
5	<i>Clear ECR Error Counter</i>	
6-31	<i>Null</i>	

#### **Clear all Buffers**

##### **Bit 0**

When set to 1, this bit causes all spy buffers to be cleared. Writing a 0 has no effect. This bit always reads as 0.

#### **Clear Orbit Counter**

##### **Bit 1**

When set to 1, this bit causes the orbit count register to be set to zero. Writing a 0 has no effect. This bit always reads as 0.

#### **Clear TTC FIFOs**

##### **Bit 2**

This bit is intended for testing only. When set to 1, it clears any pending entries from TTC FIFOs in the Switch. This leads to excess data fragments in the Input FPGAs, providing a means to test this aspect of the ECR error recovery logic. Writing a 0 has no effect. This bit always reads as 0.

#### **Fake L1A**

##### **Bit 3**

This bit is intended for testing only. When set to 1, it generates entries in the Switch FPGA FIFOs for L1ID, Orbit and TriggerType, but not BCN. This leads to misaligned FIFOs which should be detected and reset by the ECR Error Reset logic. Writing a 0 has no effect. This bit always reads as 0.

#### **Clear L1A Counter**

##### **Bit 4**

When set to 1, this bit causes the total L1A Counter to be set o zero. Writing a zero has no effect. The bit always reads as 0.

#### **Clear ECR Error Counter**

##### **Bit 5**

When set to 1, this bit causes the ECR Error Counter to be set o zero. Writing a zero has no effect. The bit always reads as 0.

### **4.8.4 Switch FPGA Status Register**

All bit fields are outputs from card.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal</b>
0	<i>BCN FIFO Overflow</i>	
1	<i>Event No FIFO Overflow</i>	
2	<i>TrigType FIFO Overflow</i>	
3	<i>OrbitNo FIFO Overflow</i>	
4	<i>BCN FIFO Over Threshold</i>	
5	<i>EventNo FIFO Over Threshold</i>	
6	<i>TriggerType FIFO Over Threshold</i>	
7	<i>Orbit No FIFO Over Threshold</i>	
8	<i>Switch FPGA BUSY asserted</i>	
4-31	<i>Null</i>	

#### ***BCN FIFO Overflow***

##### **Bit 0**

This bit is set to 1 if the BCN FIFO Overflows. It is cleared by the FIFO reset in the pulse register.

#### ***Event No FIFO Overflow***

##### **Bit 1**

This bit is set to 1 if the Event No FIFO Overflows. It is cleared by the FIFO reset in the pulse register.

#### ***TrigType FIFO Overflow***

##### **Bit 2**

This bit is set to 1 if the Trigger Type FIFO Overflows. It is cleared by the FIFO reset in the pulse register.

#### ***Orbit No FIFO Overflow***

##### **Bit 3**

This bit is set to 1 if the Orbit Number FIFO Overflows. It is cleared by the FIFO reset in the pulse register.

#### ***BCN, Event No, TriggerType and OrbitNumber FIFOs Over Threshold***

##### **Bits 4-7**

When one of these bits is set, the depth of the corresponding FIFO is above the threshold level for BUSY generation, and the corresponding input into the common Switch BUSY signal is active.

#### ***Switch FPGA BUSY asserted***

##### **Bit 8**

When this bit is set, the overall Switch BUSY signal is active. This is ORed with corresponding signals from the Input FPGAs to create the overall ROD\_BUSY signal.

#### **4.8.5 Run Number Register**

This register contains the current run number for use in the S-Link headers.

#### 4.8.6 ECR Count Register

Bit	Descriptive Name	Signal name
0-7	<i>ECR Count</i>	
8-31	<i>Null</i>	

This 8-bit register counts the number of TTC Event-Counter-reset commands received over the TTC, wrapping back to 0 after reaching 255. The value is inserted into the top 8 bits of the Event Number buffer on receipt of each L1A. It may be cleared by writing a zero at any time. Any other value may be written if needed to cure synchronisation problems.

#### 4.8.7 Replay BCN Register

Bit	Descriptive Name	Signal name
0-11	<i>Replay_BCN</i>	
12-31	<i>null</i>	

##### **Replay\_BCN**

Bits 0-11

This field contains the 12-bit bunch crossing number to be used in the event header when in replay mode instead of the value provided by the TTC.

#### 4.8.8 Replay Event Number Register

Bit	Descriptive Name	Signal name
0-23	<i>Replay_Event Number</i>	
24-31	<i>null</i>	

##### **Replay\_Event Number**

Bits 0-23

This field contains the 24-bit Level-1 ID number to be used in the event header when in replay mode instead of the value provided by the TTC.

#### 4.8.9 Replay Trigger Type Register

Bit	Descriptive Name	Signal name
0-7	<i>Replay Trigger Type</i>	
8-31	<i>null</i>	

##### **Replay\_Trigger Type**

Bits 0-7

This register contains the 8-bit trigger type to be used in the event header when in replay mode.

#### 4.8.10 TTC SampleId Register

This 32-bit register contains the level-1 ID of the first event following a TTC sample command, written by the sampling controller.

#### 4.8.11 RoI Wordlimit Register

This 32-bit register contains the maximum number of payload words (excluding header and trailer) which should be copied into an S-Link packet. The power-on value is zero, meaning no limit.

#### 4.8.12 S-Link Control Register

Bit	Descriptive Name	Signal name
0(LSB)	<i>S-Link 0 Enable Output</i>	
1	<i>S-Link 0 Enable LFF</i>	
2	<i>S-Link 0 soft LFF</i>	
3-7	<i>Reserved</i>	
8	<i>S-Link 1 Enable Output</i>	
9	<i>S-Link 1 Enable LFF</i>	
10	<i>S-Link 1 soft LFF</i>	
11-15	<i>Reserved</i>	
16	<i>S-Link 2 Enable Output</i>	
17	<i>S-Link 2 Enable LFF</i>	
18	<i>S-Link soft LFF</i>	
19-23	<i>Reserved</i>	
24	<i>S-Link 3 Enable Output</i>	
25	<i>S-Link 3 Enable LFF</i>	
26	<i>S-Link 3 soft LFF</i>	
27-31	<i>Reserved</i>	

##### **S-Link Enable**

**Bits 0, 8, 16, & 24**

When set to 1, these bits enable the corresponding S-Link. Apart from clocks, signals should not be sent to disabled S-Links, and incoming signals should be ignored. S-Links will only be used if also enabled in the S-Link routing register. On power-up, these bits should be set to zero to prevent transmission or reception while the S-Link state is unknown.

##### **S-Link Enable LFF**

**Bits 1, 9, 17, & 24**

When set to 1, these bits enable normal S-Link flow-control processing on the ROD. When set to 0, any LFF received from the corresponding S-Link should be ignored (i.e. the incoming LFF signals is ANDed with the corresponding LFF Enable).

##### **S-Link soft LFF**

**Bits 2, 10, 18, & 26**

When set to 1, these bits generate a software level to be ORed with incoming S-Link LFF signals.



#### 4.8.13 S-Link Status Register

Bit	Descriptive Name	Signal name
0 (LSB)	<i>S-Link 0 Present</i>	
1	<i>S-Link 0 Link Down</i>	
2	<i>S-Link 0 Link Full</i>	
3-7	<i>Reserved</i>	
8	<i>S-Link 1 Present</i>	
9	<i>S-Link 1 Link Down</i>	
10	<i>S-Link 1 Link Full</i>	
11-15	<i>Reserved</i>	
16	<i>S-Link 2 Present</i>	
17	<i>S-Link 2 Link Down</i>	
18	<i>S-Link 2 Link Full</i>	
19-23	<i>Reserved</i>	
24	<i>S-Link 3 Present</i>	
25	<i>S-Link 3 Link Down</i>	
26	<i>S-Link 3 Link Full</i>	
27-31	<i>Reserved</i>	

##### **S-Link Present**

Bits 0, 8, 16, & 24

When set to 1, these bits indicate that an S-Link daughter module is physically present in the corresponding S-Link slot.

##### **S-Link Link Down**

Bits 1, 9, 17, & 27

When set to 1, these bits indicate that the corresponding S-Link is asserting Link Down, for example because no fibre is connected.

##### **S-Link Full**

Bits 2, 10, 18, & 28

When set to 1, these bits indicate that the corresponding S-Link is asserting Link Full. This is normally only a transient condition.

#### 4.8.14 S-Link Routing

This register determines the routing from Input FPGAs 0-5 to S-Links 0-3. The following values are defined:

Value	S-Link 0	S-Link 1	S-Link 2	S-Link 3
0	Input 0	Input 1	Input 2	Inputs 3& 4
1	Input 0 + 1	Unused	Input 2+3+4	Unused
2	All Inputs	Unused	All Inputs (copy)	

Data is sent over an S-Link only if active in the routing register and if enabled in the S-Link control register

#### 4.8.15 Sampling Control

This register controls the selection of events for sampling in spy buffers.

Bit	Descriptive Name	Signal name
0-15	<i>SamplePreScale</i>	
16-23	<i>Trigger Type Mask</i>	
24-31	<i>Masked Trigger Type Value</i>	<u>0</u>

##### **Sample Pre-Scale**

Bits 0-15

This 16-bit field contains the sampling pre-scale factor. A hidden register is pre-loaded with this value and decremented as each event is considered for sampling. If the result after decrementing is non-zero, the event is not sampled. If the result is zero, the event is sampled (if a spy buffer is free), and the hidden register is re-loaded with the starting value.

##### **Trigger Type Mask**

Bits 16-23

The contents of this register are ANDed with the trigger type as the first stage of the event sampling mechanism.

##### **Masked Trigger Type Value**

Bits 24-31

The contents of this register are compared with the value obtained by ANDing the trigger type with the mask. If the two values match, the event is retained as an eligible event for pre-scaling.

#### 4.8.16 Spy Status Register

This register controls the selection of events for sampling in spy buffers.

Bit	Descriptive Name	Signal name
0-3	<i>SpyBuf0Occupied</i>	
4-7	<i>Reserved</i>	
8-11	<i>SpyBuf1Occupied</i>	
12-15	<i>Reserved</i>	
16-19	<i>SpyBuf2Occupied</i>	
20-23	<i>Reserved</i>	
24-27	<i>SpyBuf3Occupied</i>	
28-31	<i>Reserved</i>	<u>0</u>

##### **SpyBufOccupied**

Bits 0-3, 8-11, 16-19, 24-27

This register contains four bytes, each relating to the spy buffer associated with one S-Link. Each spy buffer is further subdivided into four event buffers. If any event buffer is occupied by an event fragment, the corresponding bit is set in this register.

#### 4.8.17 Trigger Type Timeout Register

This register contains the number of ticks the ROD should wait for trigger type before timing out. The power-up value is zero.

Bit	Descriptive Name	Signal name
0-7	<i>Timeout tick count</i>	
8-31	<i>Null</i>	

#### 4.8.18 TTC FIFO Busy Depth

This register sets the number of entries in the TTC FIFO above which Busy will be generated. If set to the maximum of 0xFF, busy will never be generated from the TTC FIFO.

#### 4.8.19 BCN FIFO Pointer Register

This register gives read-only access to the read and write pointers of the BCN FIFO.

Bit	Descriptive Name	Signal name
0-7	<i>Write address pointer</i>	
8-15	<i>Null</i>	
16-23	<i>Read address pointer</i>	
24-31	<i>Null</i>	<u>0</u>

#### 4.8.20 G-Link Timeout Register

This register sets the timeout value in ticks use for the G-Link Timeout. Only the low 16 bits are used.

Bit	Descriptive Name	Signal name
0-15	<i>Timeout Value</i>	
16-31	<i>Null</i>	

#### 4.8.21 Total LIA Number

This register returns the total number of LIAs received from the TTC. The counter is reset by a bit in the [Switch FPGA Pulse Register](#).

#### 4.8.22 Last Event LIID Register

This register gives the extended LIID of the last event to be built. This will normally also be the last event transmitted, unless Xoff is asserted by the S-Link.

#### 4.8.23 Last TTC LIID Register

This register gives the extended LIID of the last L1A received from the TTCrx.

#### 4.8.24 Detector Event Type Register

This register contains the value to be loaded into the low 16 bits of the ROD header detector event type.

Bit	Descriptive Name	Signal name
0-15	<i>DetectorEvttype Low16</i>	
16-31	<i>Null</i>	

#### 4.8.25 Orbit Counter Register

This register contains the 16-bit orbit counter. The value is incremented by each Bunch Counter Reset and cleared by Clear Orbit Counter bit in the Switch Pulse register and by the Reset TTC bit in the Control Pulse Register.

Bit	Descriptive Name	Signal name
0-15	<i>Orbit Counter</i>	
16-31	<i>Null</i>	

#### 4.8.26 Orbit Counter Wrap Value

This register determines the wrapping (maximum) value of the orbit counter. The orbit counter is incremented by each Bunch Counter Reset. If such an increment would cause the orbit counter to exceed the wrap value, the new value should instead be set to zero.

#### 4.8.27 ECR Delay Register

This 32-bit register contains the delay in ticks after arrival of the TTC ECR signal before the ROD can generate the ECR\_Error\_Reset. The value should be chosen to lie within the deadtime imposed by the CTP – e.g. it should be less than 40,000 if the CTP deadtime is 1 ms.

#### 4.8.28 ECR Error Counter

This 32-bit register counts the number of times that an error condition requiring a reset was detected in the ROD following receipt of the TTC ECR signal. The counter is reset by the Clear ECR Error Counter bit in the Switch Pulse Register

#### 4.8.29 Format Version Register

Bit	Descriptive Name	Signal name
0-11	<i>MinorFormat Version No</i>	
12-15	<i>ROD version</i>	
16-31	<i>MajorFormat Version No</i>	

This register provides the major and minor ROD header format versions described in section [3.6.1](#).

##### **Minor Format Version No**

##### **Bits 0-11**

These 12 bits should be set to specify the firmware release – an ensemble of firmware version/revisions for the complete ROD. [Table 8](#) provides the correspondence between ROD spec versions and this value.

### **ROD Version**

#### **Bits 12-15**

These four bits should be set to 0x1 to indicate the 9U ROD.

### **Major Format Version No**

#### **Bits 16-31**

These 16 bits specify the Major Format Version and should be set by software to match the ATLAS Data format version in use – see [22].

### **4.8.30 User Header Register 1**

This register is the first of four copied into the S-Link packet immediately following the ROD header and before any G-Link data. Together, they provide the User Header (3.6.6). The least significant 4 bits also determine the total number of user header words written, up to a maximum of four, by copying the User Header Registers 2-4.

The remaining bits label the triggered slice in the modules, but are not actively used in the firmware.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-3	<i>LengthOfUserHeader</i>	
4-8	<i>PPM_LUT_LIA Slice</i>	
9-11	<i>PPM_FADC_LIA Slice</i>	
12-15	<i>CP Subsystem Slice</i>	
16-19	<i>Jet-Energy Subsystem Slice</i>	
20-27	<i>Compression Lower Bound</i>	
28-31	<i>Word Id – set to 0xF</i>	

### **4.8.31 User Header Registers 2-4**

These registers are copied into the ROD S-Link packet in order until the number of registers reached in header register one has been reached.

### **4.8.32 TrigType Buffer**

This register gives read-only access to the FIFO containing trigger-types received from the TTC system and awaiting insertion into S-Link Headers. The buffer is organised as 256 1-byte values.

### **4.8.33 BCN\_Buffer**

This register gives read-only access to the FIFO containing Bunch-Crossing Numbers received from the TTC system and awaiting insertion into S-Link Headers. . The buffer is organised as 256 16-bit words, of which bits [12:15] are unused.

### **4.8.34 Event\_Number Buffer**

This register gives read-only access to the FIFO containing Event Level-1 ID Numbers received from the TTC system and awaiting insertion into S-Link Headers. The buffer is organised as 256 32-bit words.

#### 4.8.35 Orbit Number Buffer

This register gives read-only access to the FIFO containing Orbit Numbers sampled on receipt of L1A from the TTC system and awaiting insertion into S-Link Headers. . The buffer is organised as 256 16-bit words.

#### 4.8.36 S-Link 0-3 LFF time

These four registers provide a simple rate-meter to measure the proportion of time for which the corresponding S-Links are inactive because LFF is asserted. The ROD contains four (hidden) 31-bit registers, one per S-Link, counting 40 MHz clock cycles during which LFF is asserted. A fifth (hidden) 31-bit register counts all clock cycles. When it overflows (approximately every 50 seconds), the contents of the first four are copied into the low 31 bits of the S-Link 0-3 LFF-time registers, and all five counting registers are reset. A further (hidden) 1-bit register is inverted each time this process happens, and is copied into the high-order bit of all four LFF-time registers.

When read by software, the low 31 bits of each LFF-time register contain the wanted time, while the high-order bit should be the same for all four registers. Very rarely, a 50-second update cycle may happen while the registers are being read. In this case, the top bits of each register are different. A repeat read of all four registers should happen well within the 50-second update period and will provide data for the four S-Links measured over the same interval.

#### 4.8.37 S-Link 0-3 Source ID

These four registers provide the source identifier field for the S-Link packet header described in section 3.6.

Bit	Descriptive Name	Signal name
0-15	<i>Module ID</i>	
16-23	<i>SubDetector ID</i>	
24-31	<i>Reserved – must be zero</i>	<u>0</u>

##### **Module ID**

###### **Bits 0-15**

This 16-bit field contains the module identifier for the ROD. Values are described in section 3.6.2

##### **Subdetector ID**

###### **Bits 16-23**

This 8-bit field contains the centrally-allocated subdetector ID for the ROD. Values are listed in section 3.6.3.

#### 4.8.38 S-Link 0-3 Spy Event Length 01

These registers provide the length in longwords of the ROD-format events captured in spy buffers 0 and 1.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-15	<i>Event Length in spy buffer 0</i>	
16-31	<i>Event Length in spy buffer 1</i>	

#### **4.8.39 S-Link 0-3 Spy Event Length 23**

These registers provide the length in longwords of the ROD-format events captured in spy buffers 2 and 3.

<b>Bit</b>	<b>Descriptive Name</b>	<b>Signal name</b>
0-15	<i>Event Length in spy buffer 2</i>	
16-31	<i>Event Length in spy buffer 3</i>	

#### **4.8.40 S-Link 0-3 Spy Buffers**

These four mapped address ranges each provide access to the spy buffers. Spy buffers are organised in longwords. For each S-Link, four spy event buffers are provided, starting at byte offsets 0x0, 0x2000, 0x4000 and 0x6000. Each of the individual buffers is 2048 longwords in length.

## 4.9 Monitoring FPGA Register Map

The memory map and detailed assignment of bits may change during the trigger commissioning phase to accommodate requirements not foreseen at the time of writing this document.

The following registers are used in all types of ROD

Byte Offset (hex)	Register type	Register Name	Size in bytes	Description
0	RO	<i>FPGA Version</i>	4	<a href="#">Monitor FPGA Revision</a>
4	RW	<i>Monitor Control</i>	4	Monitor FPGA Control
8	RW	<i>Monitor Pulse</i>	4	Monitor FPGA Pulse
C	RO	<i>Monitor Status</i>	4	Monitor FPGA Status
	RW	<i>PC DRAM</i>	16M	Power-PC Instruction RAM
	RW	<i>PC Output RAM</i>	256K	64K*32 bit PowerPC result output RAM

### 4.9.1 Monitor FPGA Revision

This register describes the revision number of the firmware. All bit fields are outputs from card.

Bit	Descriptive Name	Signal name
0-15	<i>Module Type (0x2420)</i>	
16-23	<i>FPGA Code type (18)</i>	
24-31	<i>FPGA Code Revision no</i>	

#### **Module Type**

##### **Bits 0-15**

These 16 bits return the module type (0x2420 for the 9U ROD).

#### **FPGA Code type**

##### **Bits 16-23**

This eight-bit field is set to 18. It is part of a more general scheme to describe the firmware types, using the values in [Table 2](#). The type refers to the Monitor FPGA.

#### **FPGA Code Revision**

##### **Bit 24-31**

This 16-bit field contains the monitor FPGA code revision number. The first revision is numbered 1.

Other registers in the Monitor FPGA will be defined when the monitoring function is implemented.



## 5 Details of DAQ and RoI data and formats

### 5.1 General guidelines

Firmware in the Input FPGAs converts data from 16- or 20-bit G-Link to 32-bit S-Link format. The following guidelines have been used in deciding on data formats:

1. Each active G-Link contributes one or more variable-length data block(s) to an S-Link event fragment. All data blocks have the same overall structure – a one-word block header, a payload containing the reformatted G-Link data, and a block status word. The block status word should be omitted if all error indicators are zero.
2. The block header includes slice, module and crate numbers. These are all derived from registers in the ROD, not from G-Link content.
3. On PreProcessor G-Links and S-Links, the data order starts with the first channels of all PPASICs and then other channels in the same order. For all other G-Links and S-Links, the order starts with all data for the first timeslice, followed by all data for subsequent timeslices.
4. Part of the payload data may be suppressed when nonzero if corresponding bits are set in the subslice control register.
5. Zero suppression is done while the G-Link data are being reformatted. This reduces the size of the data between block header and block status. If all the G-Link data are zero suppressed, a data block must still be produced so that the overall sequence of blocks is preserved - so at least a block header is needed for every PPASIC or timeslice read out.
6. A longitudinal parity bit is sent on each G-Link pin at the end of data for each readout timeslice. This is intended to allow verification of the G-Link data transmission. Any errors detected should cause an error bit to be set in the block status word for the slice concerned.
7. Any G-Link protocol errors should also be indicated in the block status word of the timeslice in which they occur. G-Link timeout and link-down errors should also be recorded in this status word.
8. The G-Link data include a bunch-crossing number for each timeslice. Where multiple timeslices are read out, the G-Link data should contain the same bunch-crossing number, that of the L1A, for all timeslices of the event. The ROD should check the number for each slice. If a mismatch is found, the ROD should insert the BCID Mismatch error bit and the low six bits of the BCID (from the G-Link) into the sub-status word.
9. Any parity or link status or error indicators in the G-Link or S-Link data represent failure by using the value '1'. Zero indicates no error and helps in zero suppression.
10. S-Link payload data are labelled using the high-order bits of each 32-bit word (the “Word Id”). The meaning of each Word Id depends on the S-Link category, so analysis software must first identify the specific link concerned (from the source identifier fields in the S-Link ROD header) before decoding the payload. [Table 13](#) lists the Word-Id values used, starting from the leftmost (high-order) bits. When using neutral formatting, the distinction between different categories of data disappears, and the simplified (compatible) values in [Table 14](#) are used. In both cases, the firmware uses the word-id patterns xyz0 and xyz1 in handling sub-block headers and sub-status indicators.

<b>S-Link Category</b>	<b>Word Id</b>	<b>Data Content</b>
CP Subsystem DAQ (subdetector ID = 0x72, ModuleID[RoI bit] = 0)	00	CMM DAQ readout – hit sum
	01	CPM DAQ readout – trigger towers
	10	CPM DAQ readout – hits
	1100	CPM DAQ readout – sub-header
	1101	CPM DAQ readout – sub-status
	1110	CMM DAQ readout – sub-header
	1111	CMM DAQ readout – sub-status
CP Subsystem RoI (subdetector ID = 0x73, ModuleID[RoI bit] = 1)	00	CP RoI
JEP Subsystem DAQ (sub-detector ID =0x 74, ModuleID[RoI bit] = 0)	00	CMM DAQ readout
	01	JEM DAQ readout
	10	JEM DAQ readout – hits
	1100	JEM DAQ readout – sub-header
	1101	JEM DAQ readout – sub-status
	1110	CMM DAQ readout – sub-header
	1111	CMM DAQ readout – sub-status
JEP Subsystem RoI (sub-detector ID =0x 75, ModuleID[RoI bit] = 1)	100	JET RoI
	101	JET ET RoI
	0100	Energy RoI Ex value
	0110	Energy RoI Ey value, SumET hits
	0101	Energy RoI SumET value, ETMiss hits
PPM Subsystem (subdetector ID = 0x71)	0	PreProcessor DAQ readout
	1100	PPM DAQ readout – sub-header
	1101	PPM DAQ readout – sub-status

Table 13: Word-Id Values for Formatted Data

<b>S-Link Category</b>	<b>Word Id</b>	<b>Data Content</b>
All	00	All payload data.
	1100	CPM, JEM and PPM- sub-header
	1101	CPM, JEM and PPM-status
	1110	CMM – sub-header
	1111	CMM – sub-status

Table 14: Word-Id Values for Neutral Data

## 5.2 Overall Sub-block format

The structure used by all S-Link data blocks is shown in Figure 13. The Sub-status word may be suppressed if all error bits are zero, but the Sub-block header is always present.

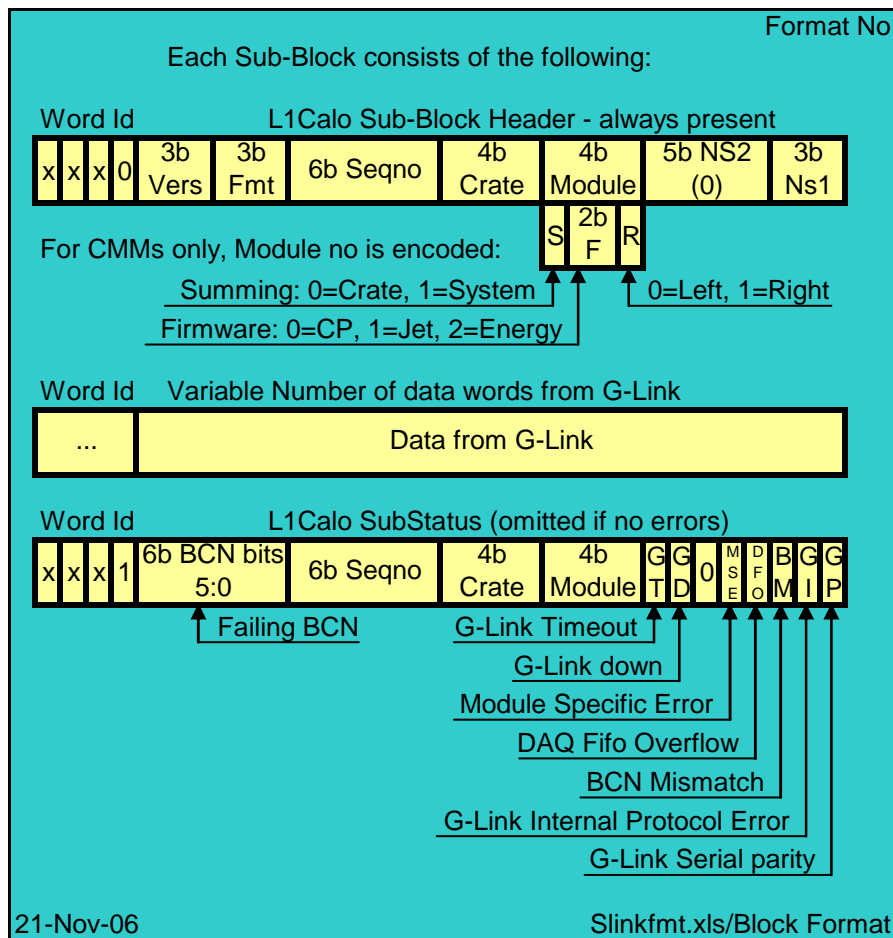


Figure 13: Overall S-Link Sub-Block format.

The data fields in the sub-block header are listed in Table 15. The data fields in the Sub-status word are listed in Table 16:

Word	Field	Width (bits)	Data Content
Sub-Block Header	Ns1	3	Number of timeslices expected. In PPM, number of LUT slices.
	Ns2	5	In PPM data, no of FADC timeslices expected. Otherwise zero.
	Module	4	Module number: JEMs 0-15, CPMs 1-14. For CMMs only, physical position plus firmware function as illustrated.
	Crate	4	Source Module Crate Number
	Seqno	6	For PPM uncompressed data, number of the first ASIC channel; for PPr compressed data, compression firmware version number; for other modules, slice number counting from 0.
	Fmt	3	Format identifier. 0 indicates neutral format data, 1 indicates formatted and zero suppressed, 2 indicates bit-field compressed, 3 indicates bit-field compressed and thresholded.
	Vers	3	Format version number, initially 1. This number will increment if the data format is changed.
	Word-ID	4	Block header identifier. Values for different links are listed in <a href="#">Table 13</a>

Table 15 : Sub-block header fields.

Word	Field	Width (bits)	Data Content
Sub-Status word	GP	1	A G-Link serial parity error was detected. The failing pin number is latched in the G-Link error map register
	GI	1	A G-Link internal protocol error was detected.
	BM	1	The G-Link and TTC Bunch crossing numbers mismatched. The low six bits of the failing G-Link BCN are copied to the BCN field in this word.
	DFO	1	DAQ FIFO overflow – L1A signals arrived too fast for the source module to read out.
	MSE	1	Module-Specific Error – see the detailed module payload.
	GD	1	The G-Link is down (link ready inactive)
	GT	1	G-Link Timeout - no data was received when the timer expired.
	Module	4	4-bit Module number, 0-15, as sub-block header
	Crate	4	4-bit Source Module Crate Number, as sub-block header
	Seqno	6	6-bit Slice number (or PPM ASIC number or PPM Compression firmware version), as in sub-block header.
	BCN	6	Set to the failing bunch-crossing number if the check fails. See BM above.
	Word-ID	4	Sub-status identifier. Values for different links are listed in <a href="#">Table 13</a>

Table 16: Sub-status word fields.

The sub-status word may be suppressed if none of the error bits is set.

### 5.3 Neutral S-Link format

In this format, the 16 or 20-bit incoming G-Link data are written to the S-Link without any decoding or parity checking, and with no attempt to identify the boundary between timeslices or to selectively suppress timeslice components. In 16-bit mode, the G-Link data are zero-extended to 20 bits. Neutral format is intended primarily for diagnostic purposes.

#### 5.3.1 Output S-Link DAQ data Format (Fmt=0)

The output format consists of a standard sub-block header word, a series of G-Link data words, and a standard substatus word, as illustrated in Figure 14. The incoming data are not parsed, and the length of the S-Link packet is determined by the duration of the G-Link DAV\* signal. The ROD cannot detect BCN mismatch or G-Link Serial parity errors even if present, and always sets these error bits to zero. The Word-Id values are set by software, depending on the module being read out, as listed in Table 14.

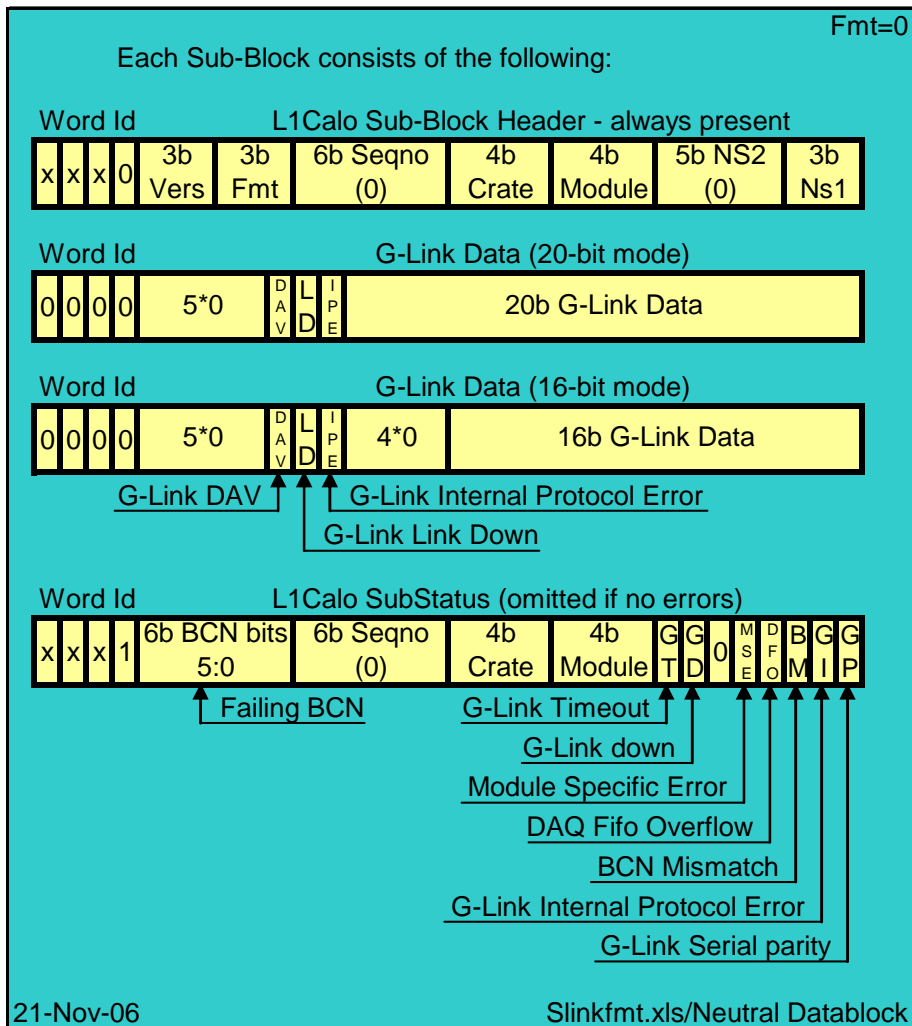


Figure 14: Neutral S-Link format.

The data fields are as follows:

Word	Field	Width (bits)	Data Content
G-Link Data	G-Link Data	16 or 20	16-bit data zero-extended to 20 bits, or 20-bit data copied directly from G-Link
	IPE	1	Internal G-Link Protocol Error this tick
	LD	1	G-Link Link Down this tick. This bit should never be set and is included for diagnostic checking only.
	DAV	1	G-Link Data Available (DAV*) for this tick. This bit should always be set and is included for diagnostic checking only.

## 5.4 PreProcessor DAQ data

The data are provided by the 16 PPM ASICs, and contain separately programmable numbers of timeslices of 10-bit FADC data and 8-bit LUT output data. The data frame also includes the 12-bit bunch-crossing number and a set of error flags.

### 5.4.1 Input G-Link DAQ data format

The ROD input G-Link format is based on the PreProcessor ASIC bit stream, illustrated in [Figure 15](#) for the case of one LUT and five FADC slices. The 11-bit LUT and FADC data also include bunch-crossing identification information, formatted as shown in [Figure 16](#).

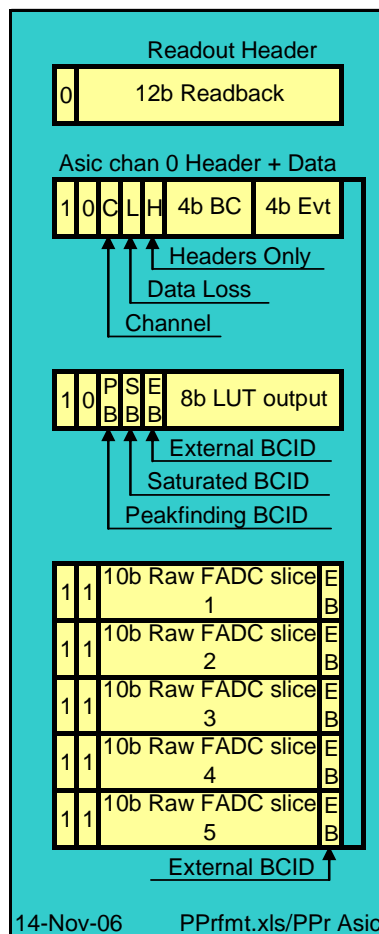


Figure 15: Bitstream format from one PPM ASIC channel

The ASIC data are collected in the PPM RemFPGA, where most of the header data are removed, and error bits added before G-Link transmission, as follows:

- The readback header indicates to the RemFPGA that event data follows. It is discarded.
- Reference copies of the event counter and the bunch counter are maintained in the REMFPGA on the PPM. The ASIC event and bunch counters are compared to the reference copies, and then discarded. If there is a mismatch, event or bunch mismatch error bits are set in the error field.
- The Headers-only flag is set when timeslices have been discarded because the ASIC data derandomiser is nearly full. This bit is propagated to the ASIC FIFO Full bit in the error field.
- The Data Loss bit is set if the ASIC derandomiser FIFO actually overflows. This bit also causes the ASIC FIFO Full bit to be set in the error field.
- The Channel indicator carries the channel number (0 or 1).It is ignored.

The resulting G-Link format is shown in [Figure 16](#). Each G-Link pin carries one bit of the bunch number (D0-D11) or a zero bit, followed by data from the four channels of one ASIC, followed by an error bit field and finally by the longitudinal parity bit. Each ASIC channel contains a programmable number (n, m) of 11-bit data fields for FADC timeslices (n) and LUT timeslices (m). Values for n and m can be set independently in the PPM in the ranges 0-127 and 0-7, but ROD input FPGA FIFOs may overflow if large numbers are used, and they are restricted by the sub-block header to the range 0-15. Reasonable values will normally be smaller. Assuming that the FIR filter uses the full number of FADC samples, the count of FADC samples should be at least four more than the count of LUT samples to enable software to re-compute the LUT outputs. This makes likely the use of value pairs 1+5, 3+7 and perhaps 5+9. Combinations with proportionally more LUT samples might be needed to check the digital part of the system.

The values must be pre-loaded by software into ROD Input FPGA registers. Including one bunch number bit, ten error bits, and one overall longitudinal parity bit, the overall length of the G-Link frame is  $4 \times (11 \times (n + m)) + 11 + 1$  clocks.

The data fields are as follows:

Word	Field	Width (bits)	Data Content
1	BC	12	Bunch Crossing Number, to be checked in ROD against L1A BC number
LUT Data	LUT output	8	Calibrated $E_T$ value extracted from LUT content
	EB	1	External BCID was active this clock cycle
	SB	1	Saturated BCID was active this clock cycle
	PB	1	Peak-finding BCID was active this clock cycle
FADC Data	FADC Output	10	Data as used from the FADC
	EB	1	External BCID value for this clock cycle.



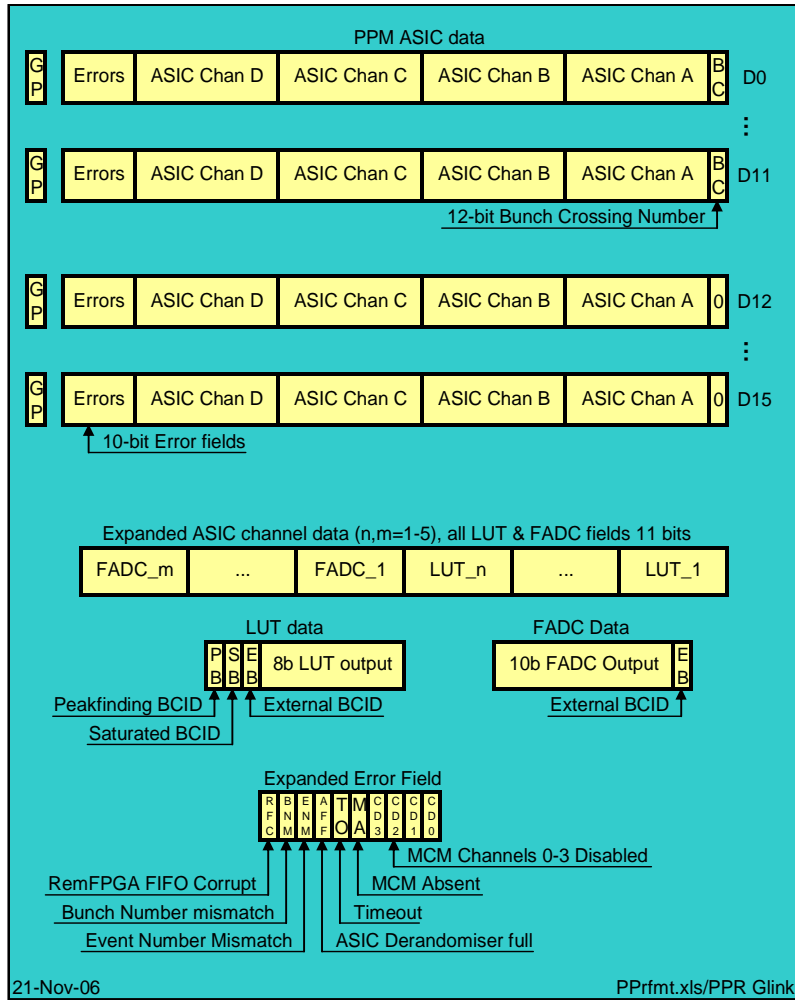


Figure 16: PPM Input G-Link data format

The error fields are as follows:

Word	Field	Width (bits)	Data Content
Error	CD0-3	1	MCM channel 0 – 3 is disabled in software
	MA	1	This MCM is absent from the module
	TO	1	One or more channels did not produce data in response to L1A
	AFF	1	L1As are coming too fast and the readout FIFO in the PPr ASIC has become full.
	ENM	1	Event Number Mismatch between REMFPGA reference value and data from the ASIC
	BNM	1	Bunch Number Mismatch between REMFPGA reference value and data from the ASIC
	RFC	1	REMFPGA FIFO corrupt – occurs if FIFO overflows.

### 5.4.2 *General Structure of PPM S-Link Data.*

The PPM data arrive in parallel on 16 G-Link pins, and the output order is chosen to minimise the data storage needed during reformatting. The S-Link data therefore contains all channel-A information from all G-Link pins (i.e. from all PPr ASICs) before starting channel B data.

Three PPM-specific formats are provided. The uncompressed S-Link format is provided for intermediate diagnostic purposes and for readout of large numbers of timeslices. Compressed format will normally be used in running, and requires exactly 5 FADC and 1 LUT timeslices from all PPMs. Super-compressed format is an optional (lossy) format which may be used if data rates become too high. It is also possible that different compression types may be produced in future, for example to accommodate 3 FADC and 1 LUT timeslices giving a reduced data volume.

The PPM has a range of internal error checks, and the resulting bits do not map well onto the common error field in the sub-status word. The presence of PPM-specific errors is therefore indicated in the Module-Specific Error bit in the sub-status word, with the details inside the PPM data payload.

### 5.4.3 *Uncompressed S-Link DAQ output format. (Fmt = 1)*

The output data are of fixed format. Five sub-blocks are written, four containing 16 ASIC channels each, and the fifth containing errors.

For each group of 16 ASIC channels, the data are written to the S-Link in the order of arrival over G-Link, with each 11-bit datum padded with five zeros, and the resulting 16-bit fields packed two to an S-Link word. When running with one LUT and five FADC timeslices, this results in a data block containing eight longwords of 2 LUT samples each, followed by eight longwords each for FADC samples one to five. This format is illustrated in [Figure 17](#).

The Error block is similarly formatted, with the 16 ten-bit error fields grouped two to a 32-bit word, as illustrated in [Figure 18](#). Because of the possible wide range in numbers of timeslices supported by this format, the G-Link serial parity and BCN Mismatch checks cannot be made in firmware, and it is necessary to inspect the error block to obtain complete information about errors. It is important to note that the errors are reported in this error block, not in the sub-status word of the preceding data sub-blocks.

In each sub-block, the first ASIC channel is identified by a 0, 16, 32 or 48 in the “seqno” field. The error block is indicated by a seqno value of 63. Future format version may change the grouping from 16 per block, so analysis software should check the seqno and not assume that the number of channels per block is permanently fixed to 16.

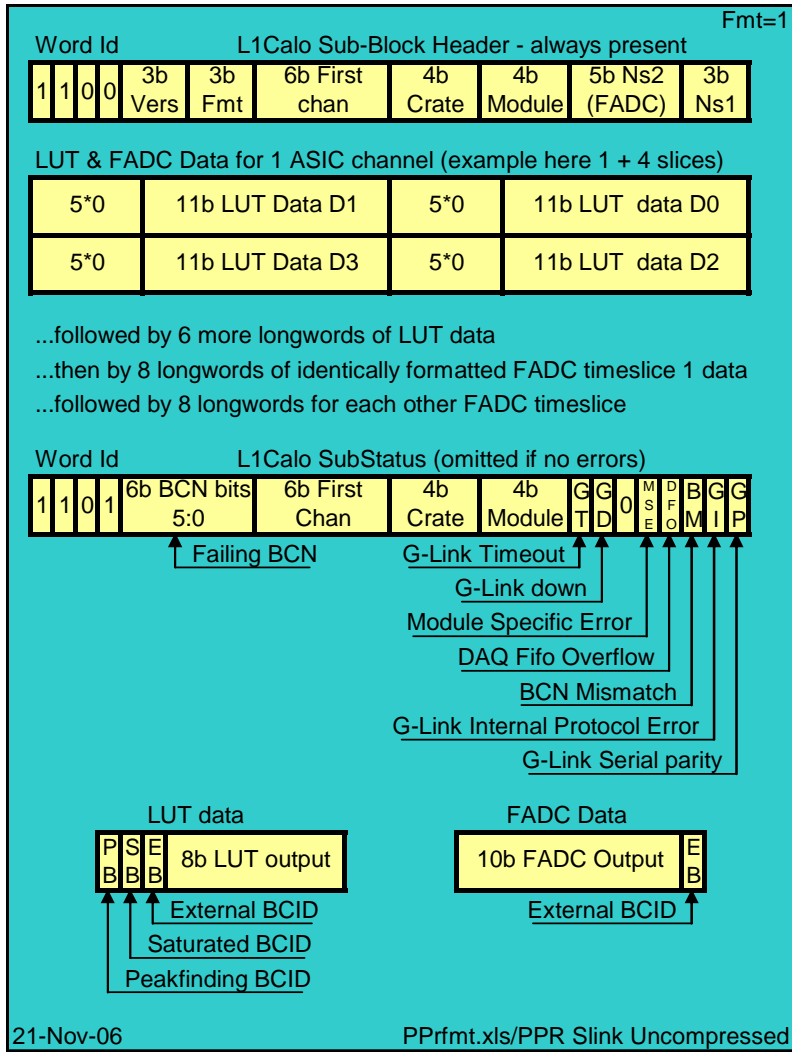


Figure 17: Uncompressed PPM S-Link output DAQ format

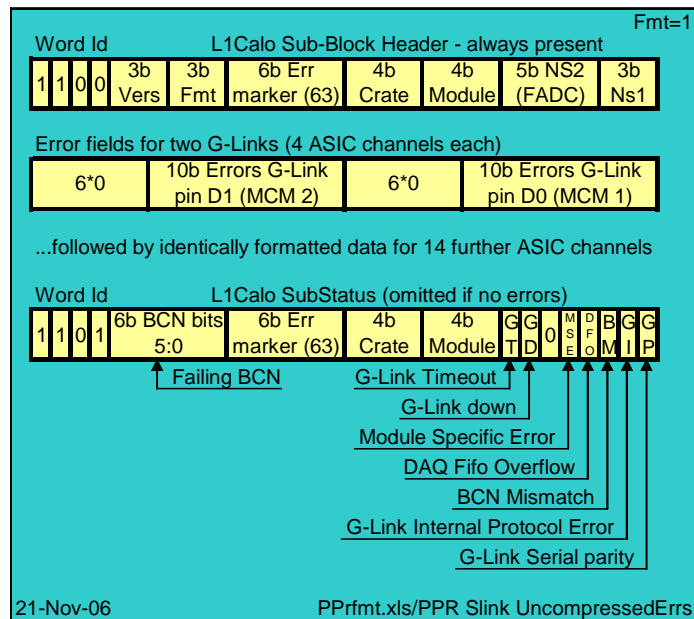


Figure 18: Uncompressed PPM S-Link output DAQ format - error block.

#### 5.4.4 Compressed S-Link DAQ output format. (Fmt = 2)

The compressed format consists of a bit stream occupying the low 31 bits of S-Link longwords, with the high bit used as a word identifier. Data from the 16 ASIC “A” channels are concatenated to form an extended bit-stream, which flows continuously over the 31-bit fields of as many longwords as necessary, starting with bit 0 on the right of the first word. Data from “B”, “C” and “D” channels follows immediately. The block is padded with zeros to occupy a whole number of S-Link longwords, preceded by a sub-block header and followed by a sub-status word (may be zero-suppressed) to create the overall block format illustrated Figure 19.

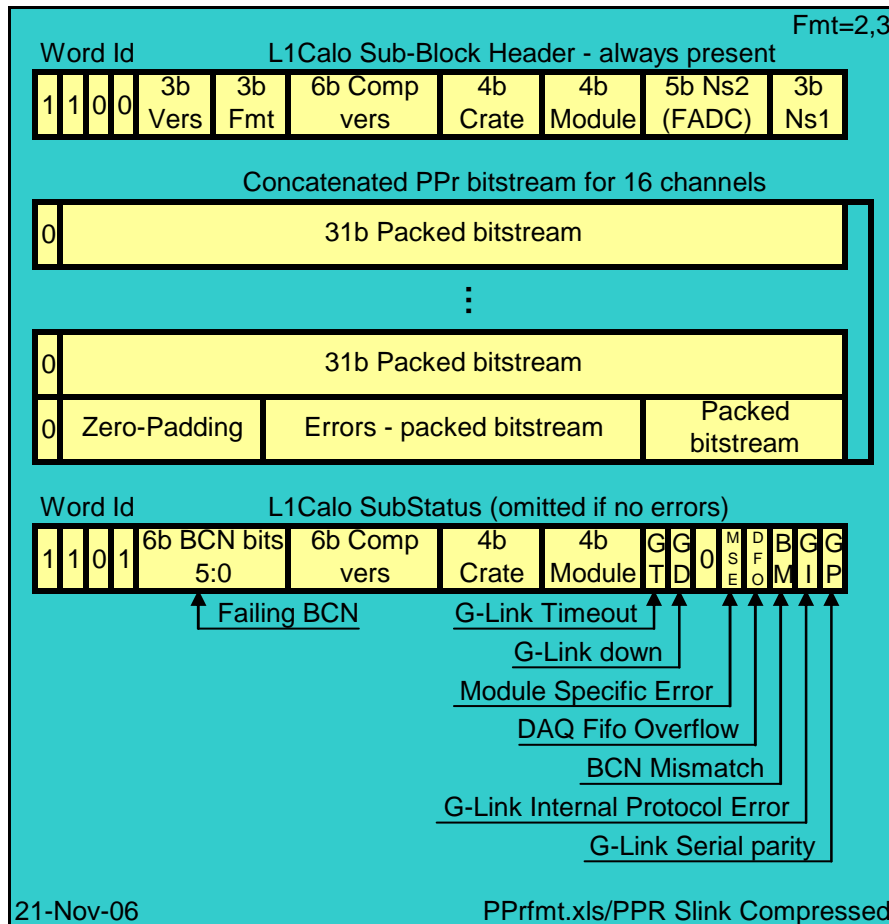


Figure 19: PPM Compressed S-Link DAQ data format

Details of the internal structure of the bitstream are contained in [16].

#### 5.4.5 Super-Compressed S-Link DAQ output format (Fmt=3)

The super-compressed format is a bit stream similar to the compressed format, except that data are omitted for those PreProcessor channels in which the PPM\_LUT result is zero and the FADC samples are all below a programmable (per G-Link) threshold. Details of the internal structure of the bitstream will be developed as needed and documented in a separate note.

## 5.5 CPM DAQ Data

### 5.5.1 Input G-Link DAQ data format

Most of the data are produced by the 20 Serialiser FPGAs, and for each input trigger tower consist of eight bits of trigger tower data, a parity error indicator and an LVDS link down indicator. The data also include the 48 threshold hit bits (on D0-D15) and the 12-bit bunch-crossing number (on D16 – D19).

One time-slice of the serial data format of the CPM Serialiser FPGA received by the ROD module is shown in [Figure 20](#). Up to five time-slices may be sent per event.

These data are 84 bits in length, but could be reduced without information loss to 76 bits by removing alternate LVDS link-down and parity error bits (which are duplicates on adjacent A and B data). This change would enable 5-slice readout at 100 kHz, but has not been implemented at present since there is no need for 5-slice CPM readout at full rate.

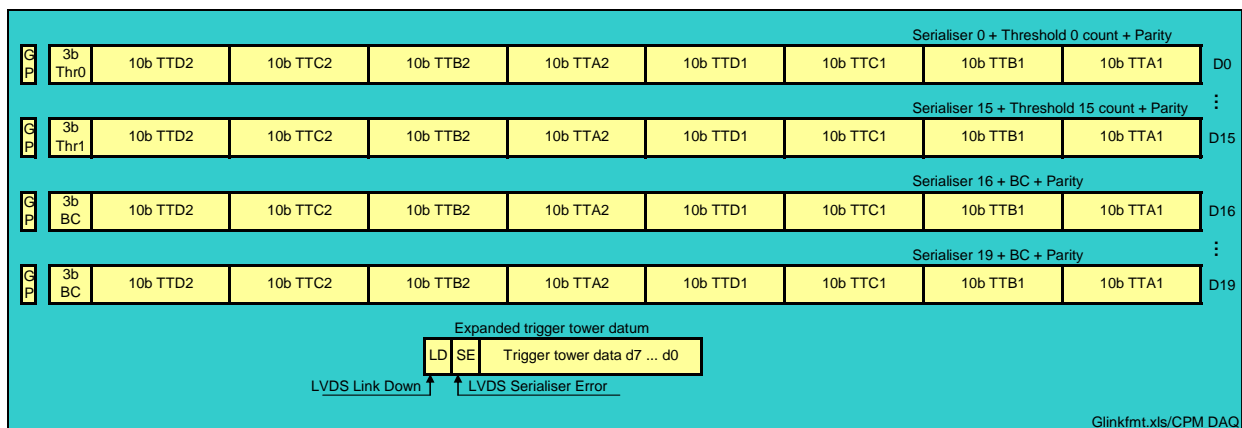


Figure 20: CPM G-Link DAQ data format

### 5.5.2 Output S-Link DAQ data format (Fmt=1)

The data format within the event fragment is as shown in [Figure 21](#). Data from each pair of trigger towers are written to one S-Link word, with the link down indicators as a separate bit field. The trigger tower pair number identifies the pair (from 0 to 3) on the incoming G-Link, and the input FPGA number (from 0 to 19) is identical to the G-Link pin number.

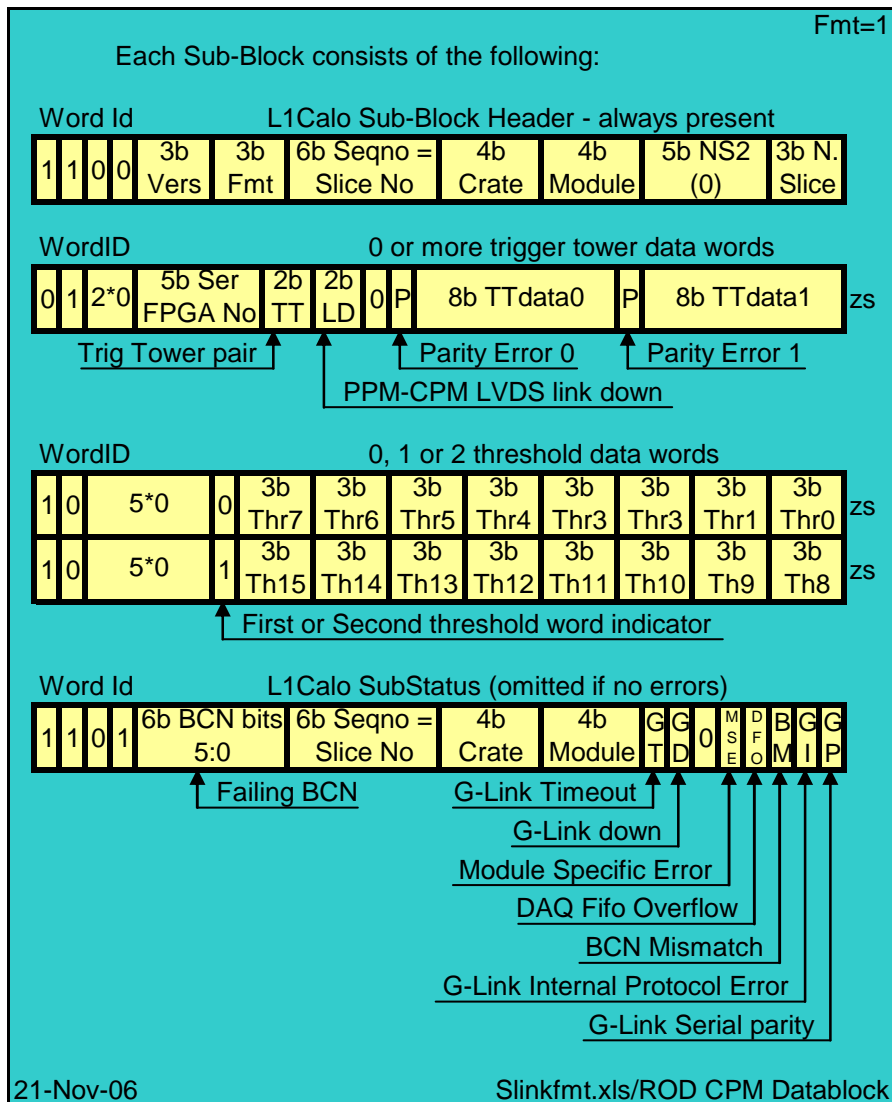


Figure 21: CPM S-Link DAQ data format.

These words should be suppressed when: both 8-bit tower-data fields are zero; both adjacent parity error fields are zero; and both PPM-CPM link down error bits are zero.

Threshold hit data are assembled into two words, the first containing data from pins D0-D7 and the second data from D8-D15. Either of these words should be zero-suppressed if all the threshold data it contains are zero.

## 5.6 JEM DAQ Data

### 5.6.1 Input G-Link DAQ data format

The JEM G-Links operate in 16-bit mode. Incoming LVDS data are packed into 11-bit frames, containing 9 data bits plus the error detect (SE) and link down (LD) per Input channel. The LVDS link may also be used in 10-bit data mode, and in this case the SE bit becomes a data bit. A total of 44 pairs of electromagnetic and hadronic jet element components are sent per timeslice, with three pairs on pins D0 – D13, and the remaining two pairs on D14.

Copies of the backplane jet hit counts plus transmitted parity, three 8-bit energy sums plus transmitted parity, and the 12-bit bunch-crossing number, are transmitted on D15. The jet counts contain twelve 2-bit fields from JEMS 0, 7, 8 and 15 which process main and forward jets, or eight 3-bit fields from the other JEMs processing main jets only.

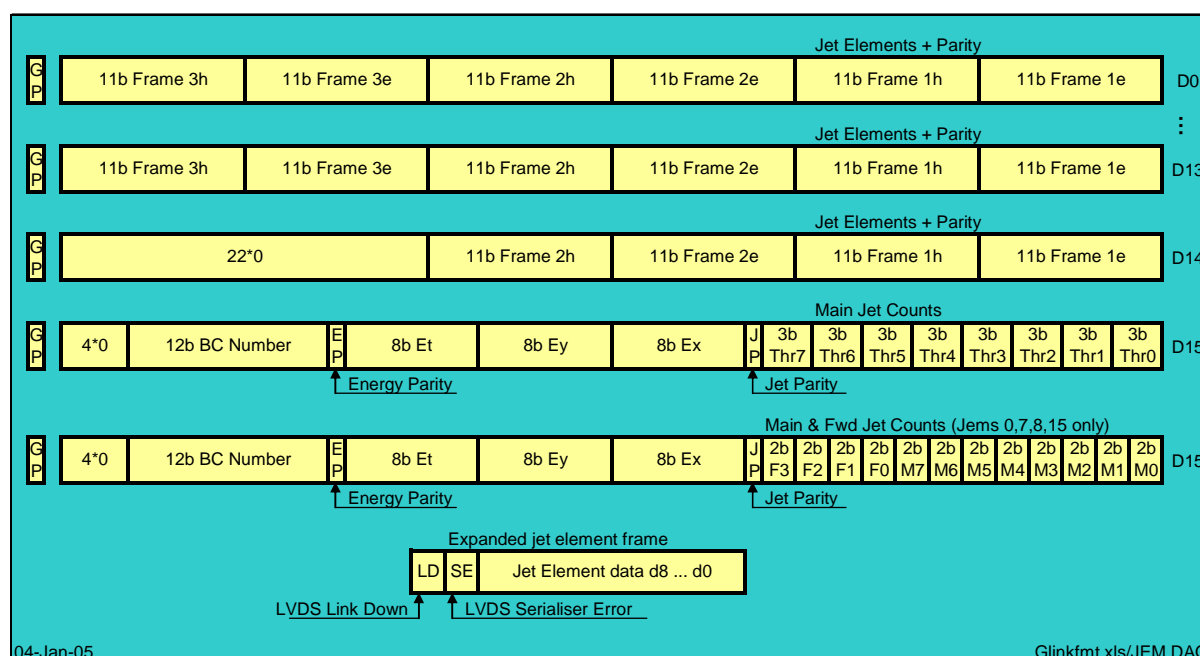


Figure 22: JEM G-Link DAQ data format (16-bit G-Link mode).

### 5.6.2 Output S-Link DAQ data format (Fmt=1)

The S-Link data format is as shown in Figure 23. Each pair of jet elements (one e.m. and one hadronic) is written to one S-Link word, with parity and link error indicators and a counter to indicate the pair number per G-Link pin. The G-Link pin number (0-14) completes these data fields.

Any of these data words should be suppressed if both 11-bit data fields (9-bit data plus two error indicators) are zero.

The jet hit counts and energy sub-sums are written with a source Id as illustrated, and each should be suppressed if the low 24 bits are all zero. The two jet hit formats do not need to be distinguished in this processing, but should be labelled separately as indicated.

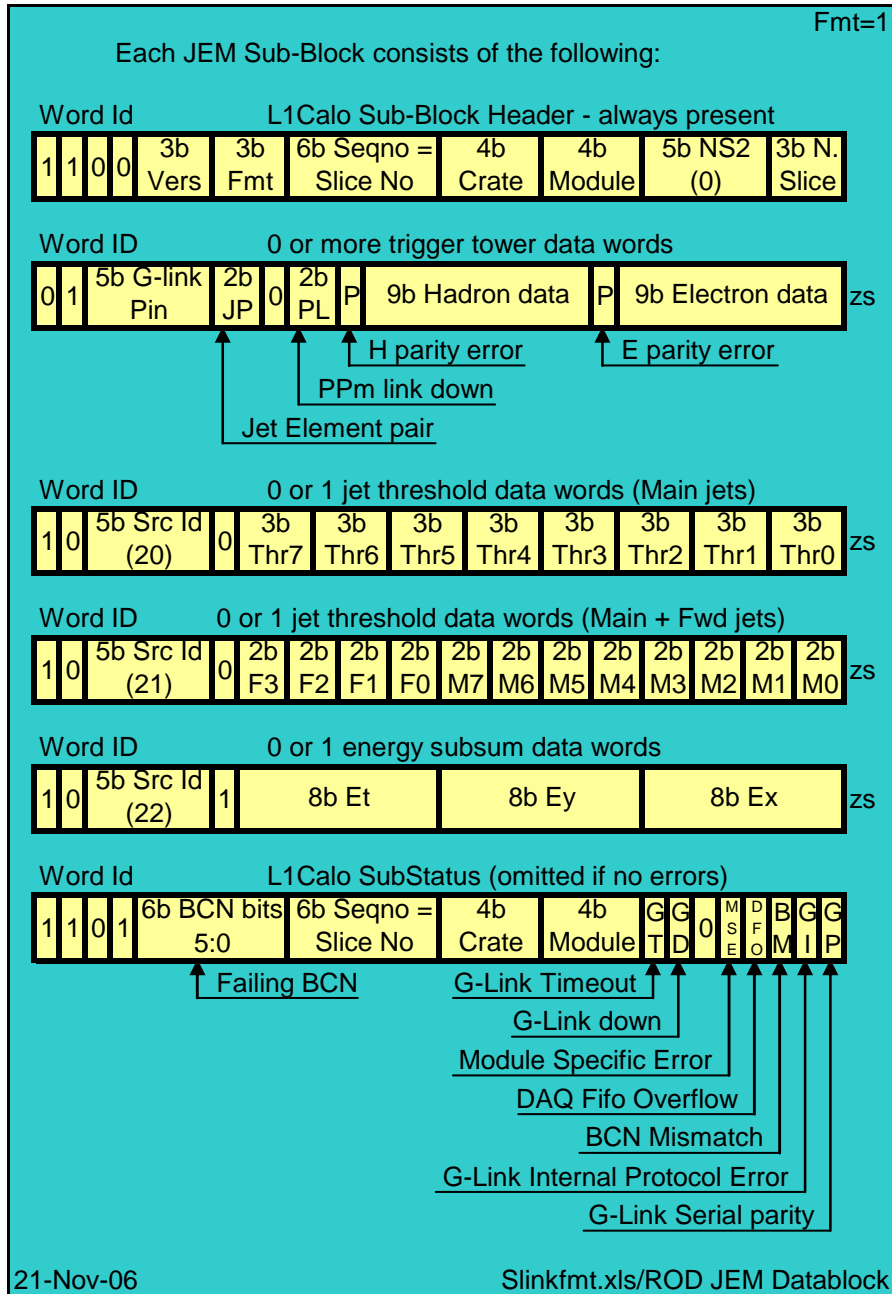


Figure 23: JEM S-Link DAQ data format



## 5.7 CMM-CP DAQ

### 5.7.1 Input G-Link DAQ data format

CMM CP, Jet and Energy G-Link data packets are the same length in clock ticks, and where possible are similarly formatted. Figure 24 shows the CMM-CP G-Link data format. Most of the data are copies of the incoming backplane hits from each CPM, and consist of 24 data bits and a backplane parity error indicator (D0-D13). D17 contains the sub-sums for this crate. For system-summing CMMs, D14-D16 contain summed hit counts from remote CMMs with cable parity error indicators, and D18 the system total hit counts. For other CMMs, these data are set to zero. The 12-bit bunch-crossing number is packed into D0-D11, and the DAQ FIFO overflow bit is sent on D13.

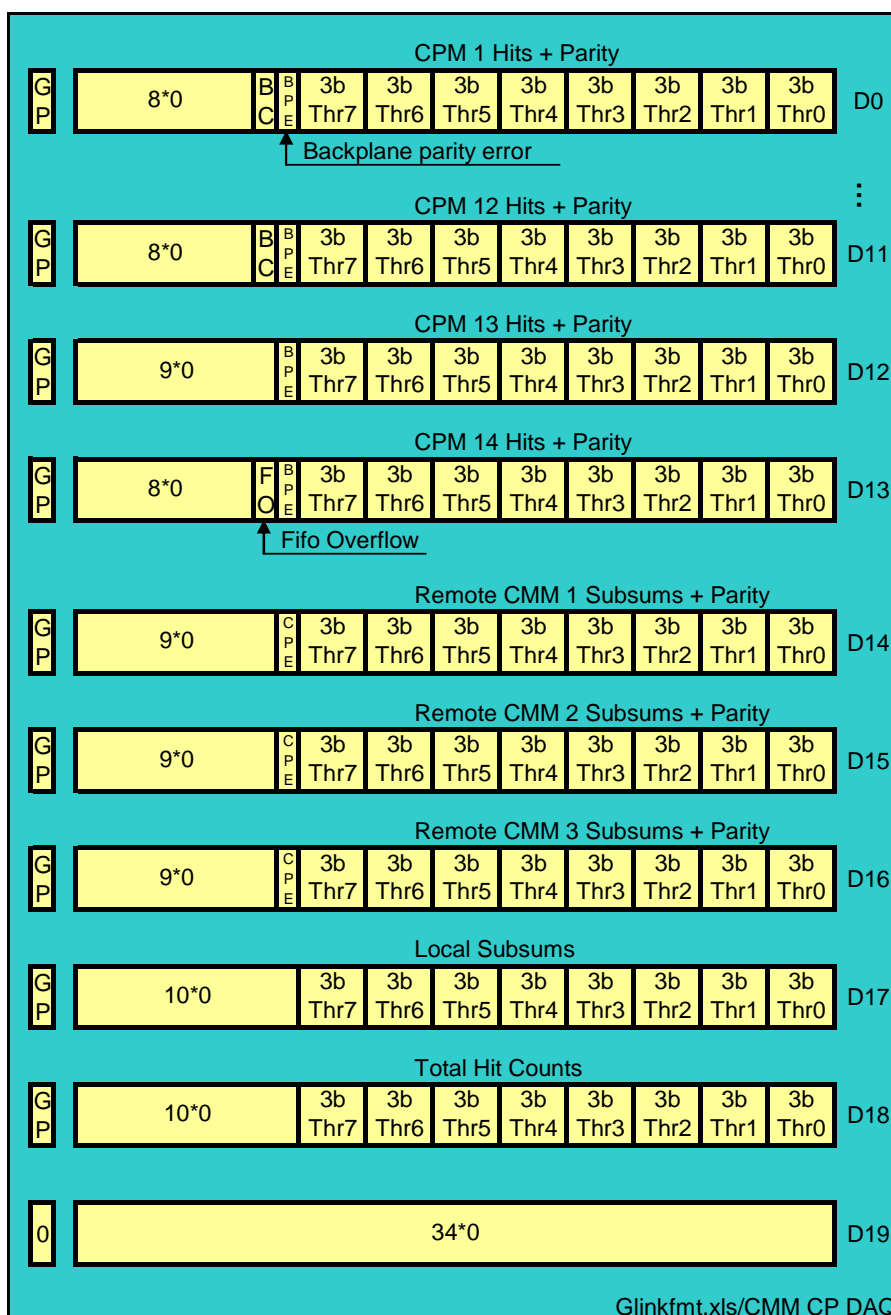


Figure 24: CMM-CP G-Link DAQ data format

### 5.7.2 Output S-Link DAQ data Format (Fmt=1)

The CMM-CP S-Link format is as shown in Figure 25. The data are packed into words containing eight 3-bit threshold counts with the accompanying error indicator. A data source field, identical to the G-Link pin number, labels each word. The data may be zero-suppressed if the low 25 bits are all zero. The DAQ FIFO error bit is copied into the SubStatus word.

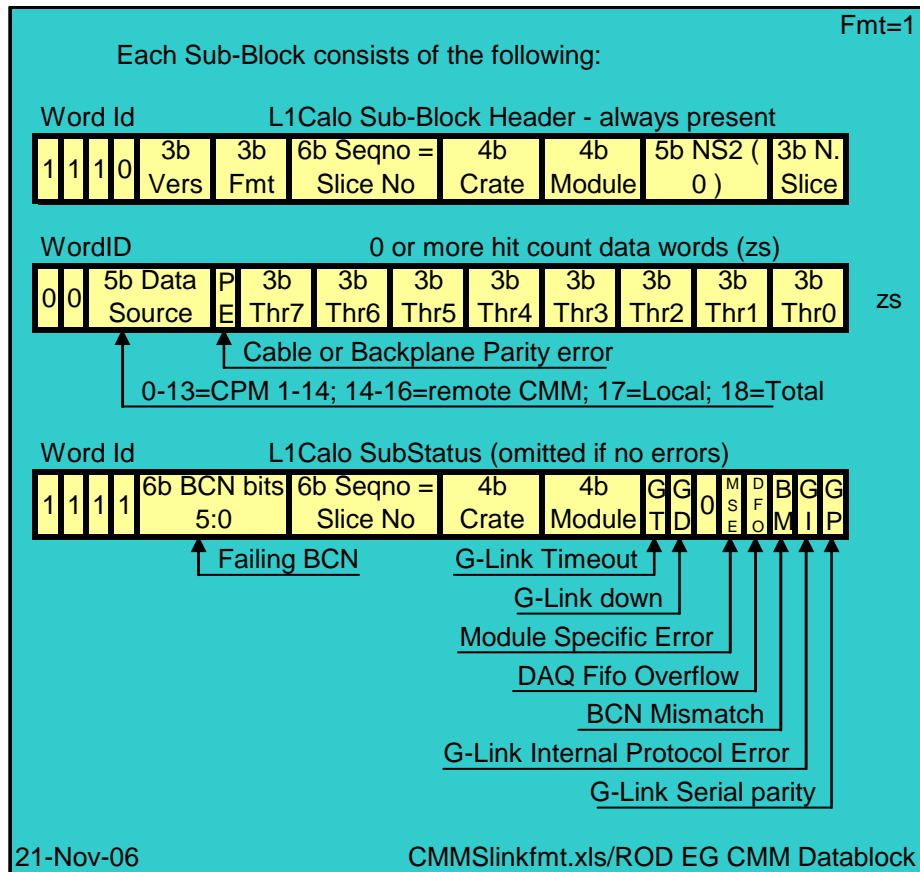


Figure 25: CMM-CP S-Link DAQ data format

## 5.8 CMM-Jet DAQ

### 5.8.1 Input G-Link Data format

Figure 26 shows the CMM-Jet G-Link data format. Most of the data are copies of the incoming backplane hits from JEMs, and consist of 24 data bits and a backplane parity error indicator (D0-D15). If the corresponding JEM is summing some forward jets, an alternative format is used for the low 24-bits. The 12-bit bunch-crossing number is packed into D0-D11, and the DAQ FIFO overflow bit is sent on D15. The local main jet sub-sums and local forward jet sub-sums (Left and Right) are packed into D17 and D19. For system-summing CMMs only, Remote input sub-sums and overall total sums are packed as indicated onto D16, D18 and D19, and the Jet Hit map output to CTP is packed into D16. These fields are otherwise set to zero.

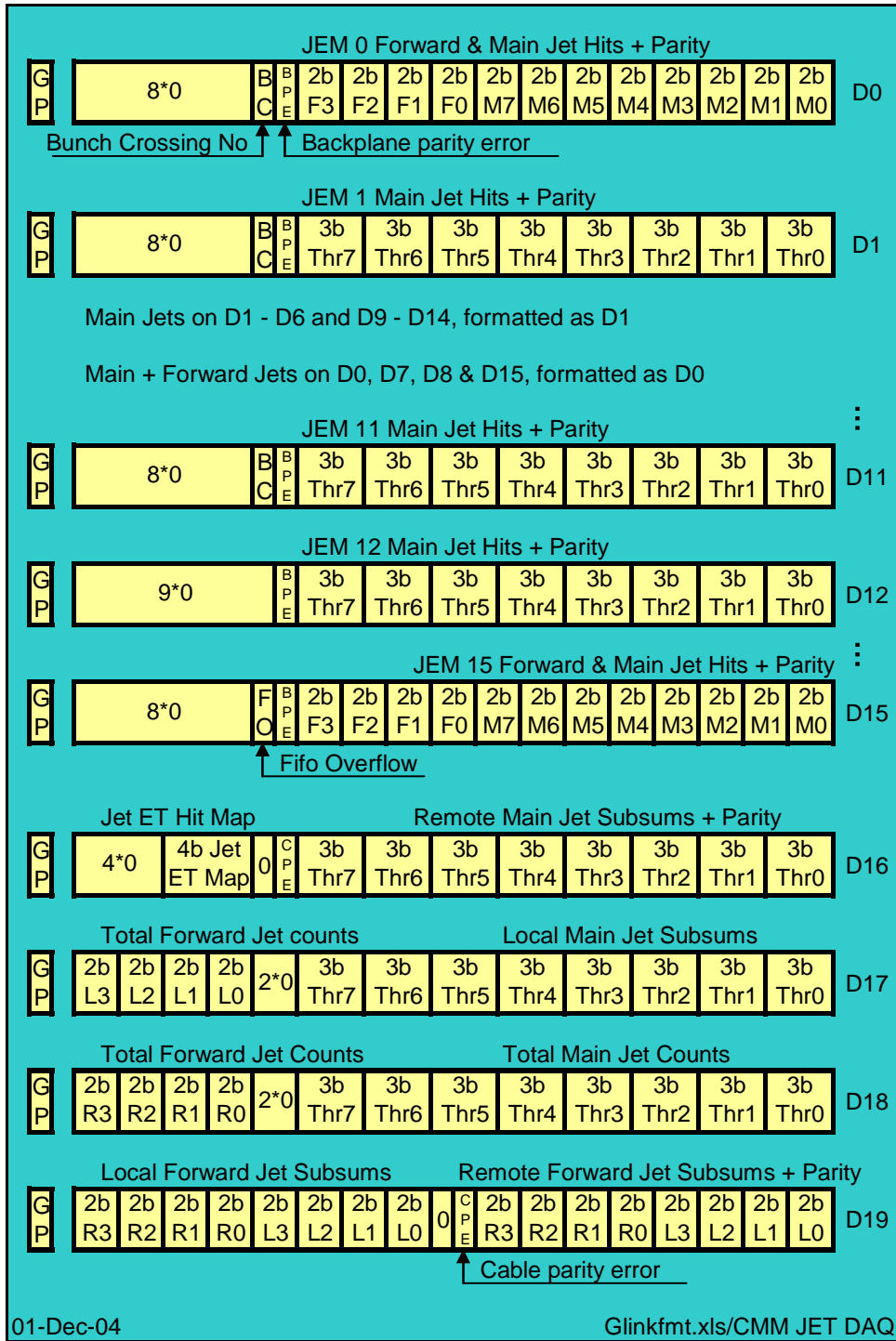


Figure 26: CMM-Jet input G-Link DAQ format

### 5.8.2 Output S-Link DAQ data Format

The CMM-Jet S-Link DAQ format is as shown in Figure 27, and contains of a mixture of 2-bit and 3-bit hit counts. Eight of the two- or three-bit hit counts are packed into each S-Link word with the accompanying error indicator.

The three-bit hit counts are labelled by their source G-Link pin number, while the two-bit hit counts have similar label fields with values of 20 or above.

Any of these data words should be zero-suppressed if the low 25 bits are all zero.

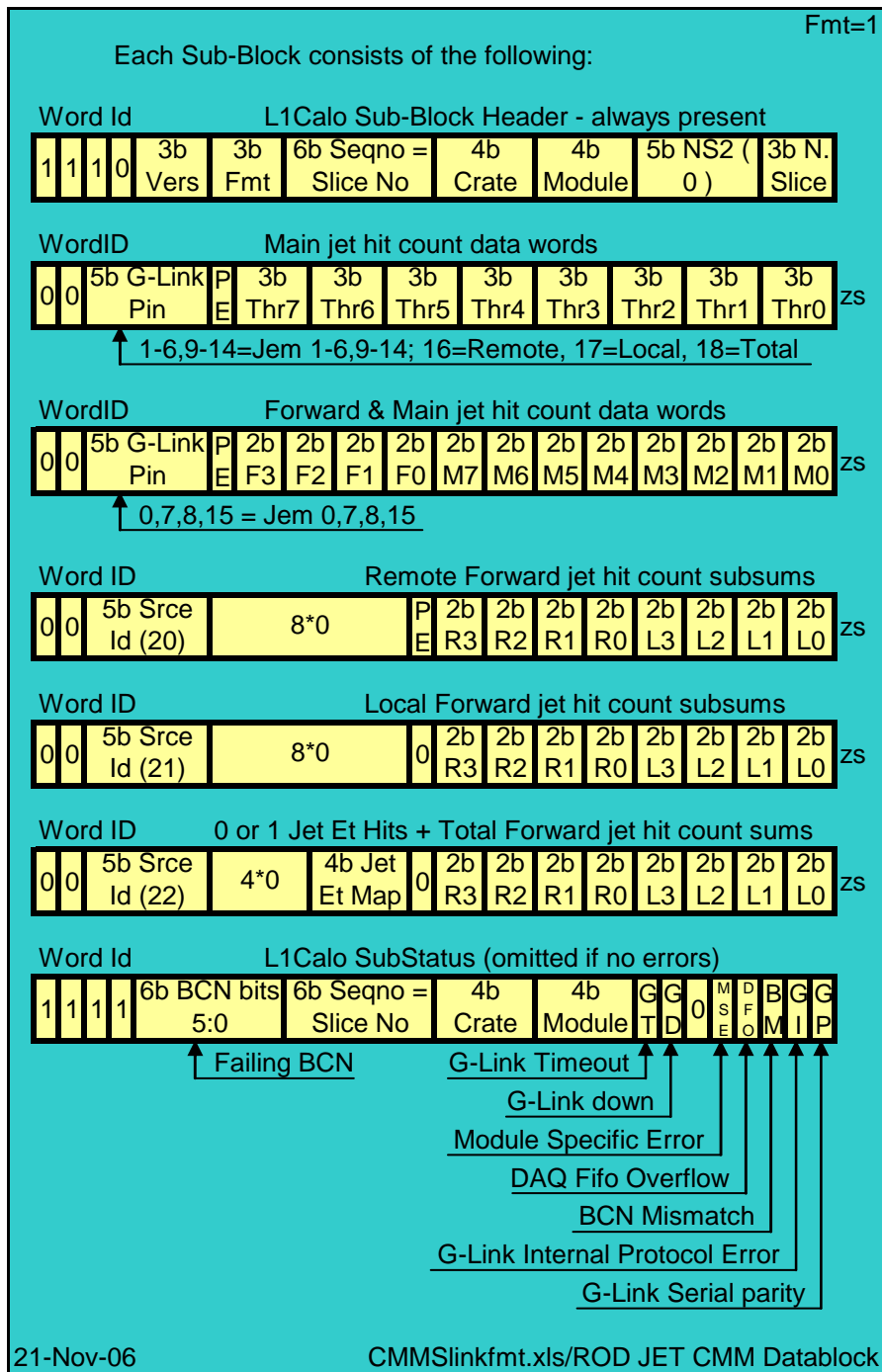


Figure 27: CMM-Jet output S-Link data format.

## 5.9 CMM-Energy DAQ

### 5.9.1 Input G-Link DAQ Data format

Figure 28 shows the CMM-Energy G-Link data format. Data on pins D0-D15 are copies of the 24-bit energies and parity error indicator from JEMs. The 12-bit bunch-crossing number is packed into D0-D11, and the DAQ FIFO overflow bit is sent on D15. The local sub-sums are packed into D17 and part of D19. For system-summing CMMs only, remote input sub-sums and overall total sums are packed as indicated onto D16, D18 and D19, and the outputs to CTP are packed onto D14-D15. These fields are otherwise set to zero.

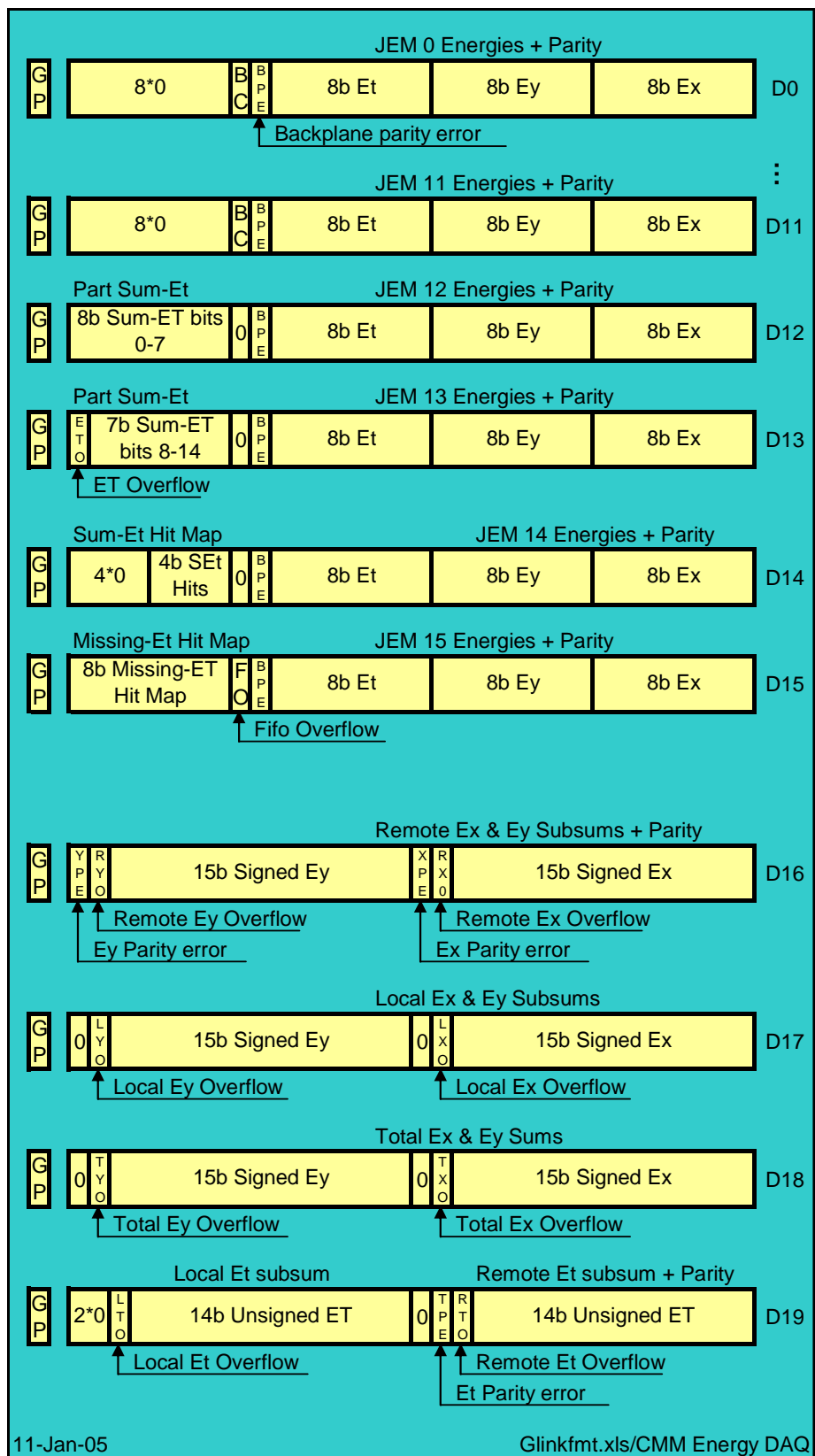


Figure 28: CMM-Energy G-Link DAQ input format

### 5.9.2 *Output S-Link DAQ data format*

The CMM-Energy S-Link format is as shown in [Figure 29](#). The incoming JEM energy words are labelled by their source G-Link pin number, while the other data have similar label fields with values of 20 or above.

Remote and local sub-sums are formatted as shown, with overflow and parity error indicators and label field values from 20 to 25. The total  $E_x$  and  $E_y$  and the missing- $E_T$  data follow, with label field values 26 and 27. Note that the missing- $E_T$  is available in the module only as LUT hit-bit outputs – the actual value is never computed. Finally the Sum- $E_T$  value and hits appear with label field set to 28.

Any of the above data words should be zero-suppressed if all bits are zero apart from word-ID and label field.

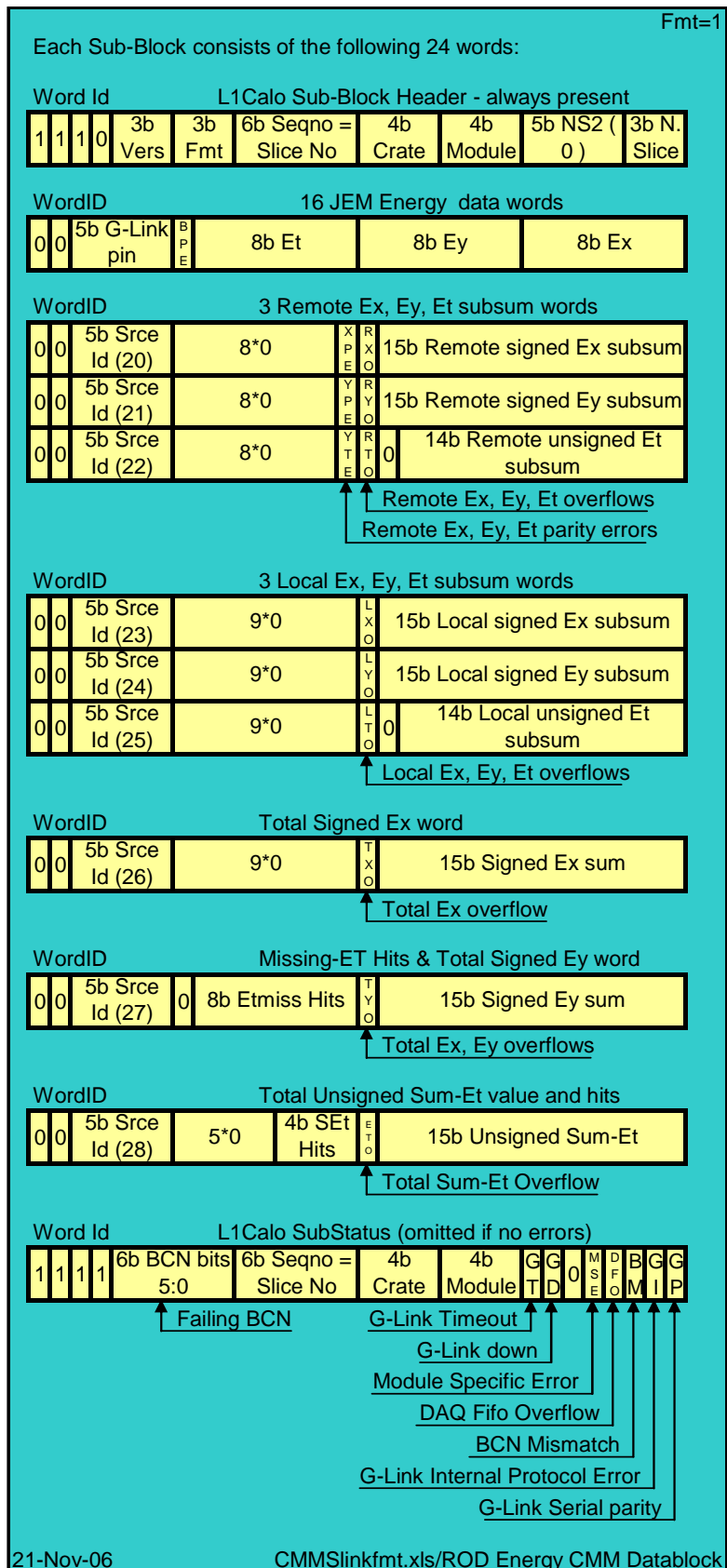


Figure 29: CMM-Energy Output S-Link DAQ data format

## 5.10 General features of RoI Formats

RoI data are built into ATLAS-standard S-Link event fragments. However, the data payload differs from DAQ data, in that no sub-block structure is present. Instead, each RoI word contains the full system address (crate number, module number, chip number, location within chip) to identify its position fully within the trigger system.

The RoI builder imposes S-Link fragment length limitations. The maximum payload length is set by software into the RoI-Wordlimit register in the switch FPGA, taking into account the S-Link header and trailer lengths. When assembling an RoI event fragment, the Switch FPGA firmware should count the number of payload words added to each fragment. If the limit is reached, the fragment should be terminated with the standard trailer, with the Limited RoI Set (LRS) bit set. Any remaining RoI data for the event being processed should be discarded.

## 5.11 CPM RoI Data

### 5.11.1 Input G-Link RoI data format

The RoI data received by the ROD is shown in Figure 30. Data from each CP chip are sent on two pins, one each for the left (L) and right (R) RoI clusters, with L always on an even pin number. Data from the eight CP chips therefore occupy the 16 pins D0 to D15 of the G-Link starting from chip 0 (left and right) on D0 and D1. The 12 bit bunch crossing number is appended from D0 to D11 with the LSB at D0. An overall longitudinal parity is added to the first 16 pins, with the remainder set to zero.

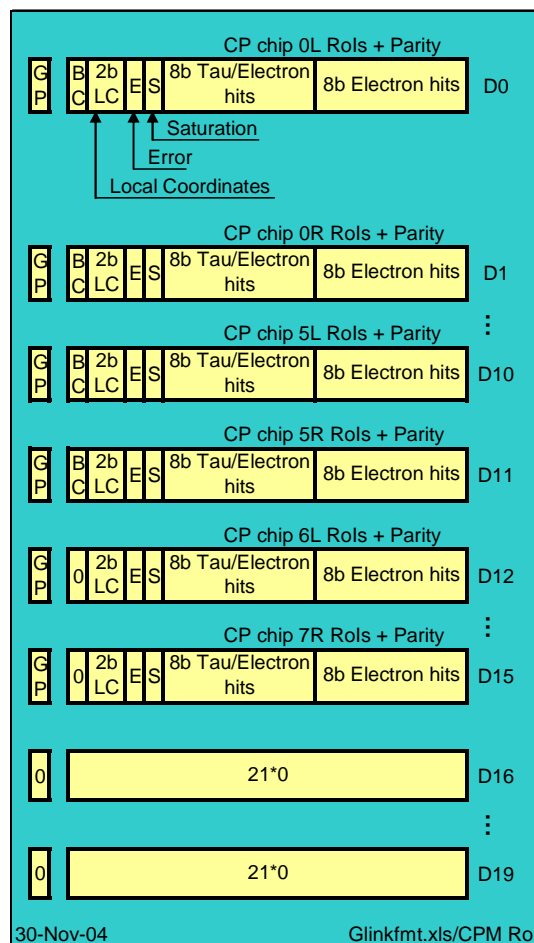


Figure 30: CPM G-Link RoI data format



### 5.11.2 Output S-Link RoI format

The data word format is as shown in Figure 31. Each word contains the two eight-bit threshold maps plus the error and saturation bits from one G-Link pin. The upper bit of the local coordinates is equal to the LSB of the G-Link pin number. The CP chip number is obtained from the next 3 remaining bits of the G-Link pin number. The CPM number and crate number are obtained from ROD Input Channel Source ID register in the corresponding Input FPGA.

These S-Link words should be suppressed if both 8-bit fields and the adjacent saturation and error bits are all zero.

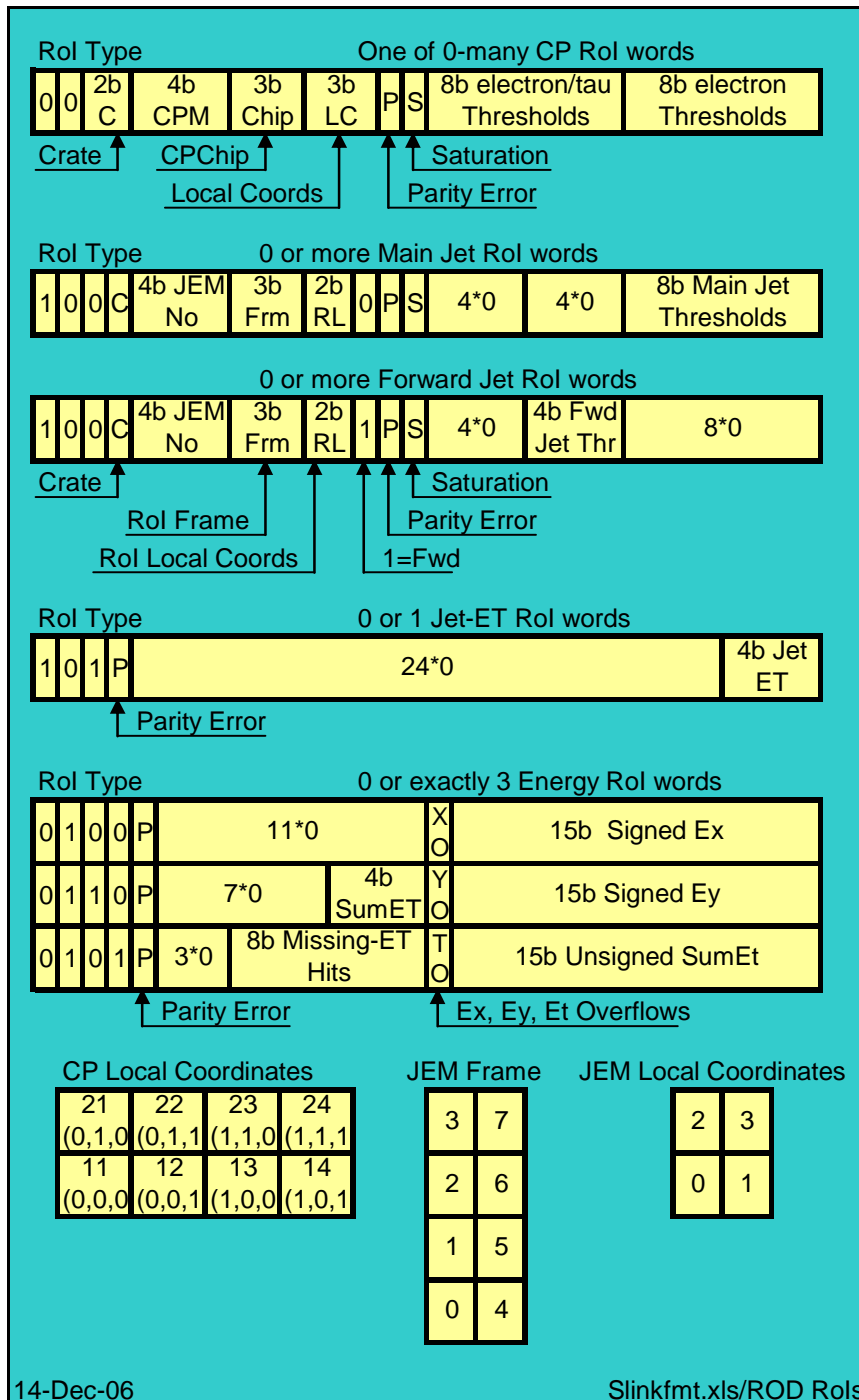


Figure 31: S-Link RoI data formats

## 5.12 JEM Jet RoI

### 5.12.1 Input G-Link RoI data format

The Jet RoI Data received by the ROD are shown in Figure 32. For main jets, four RoI frames are received on each of the G-Link pins D0 and D1. From JEMs handling forward jets, four additional forward jet RoI frames are received on each of the G-Link pins D3 and D4. Other JEMs set these fields to zero. Pin D2 of the G-Link carries the BC number. Longitudinal parity bits are appended on pins D0-D4 by all JEMs (so GP is set to 1 on pins D3-D4 from JEMs which don't send forward jet data). All unused G-Link data lines are set to zero.

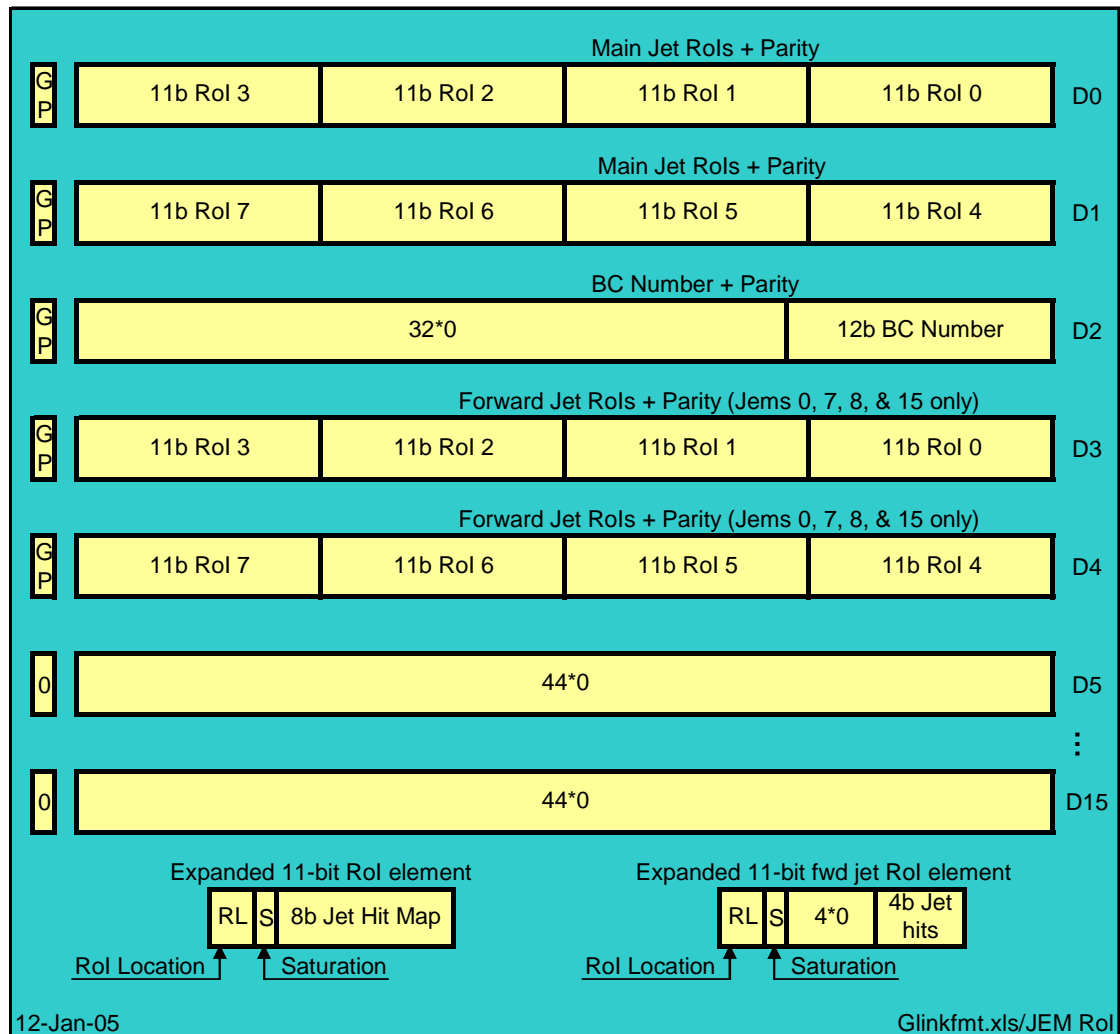


Figure 32: Jet RoI G-Link format (16-bit G-Link mode).

### 5.12.2 Output S-Link RoI data format

The S-Link format of jet RoIs is included in Figure 31. Each S-Link word contains thresholds, saturation and RoI location bits copied from one incoming RoI frame. The parity error bit is set if the incoming pin-level parity check failed. The RoI Frame number is set to the RoI element number (0-7), and the JEM number and crate number are obtained from ROD Input Channel Source ID register in the corresponding Input FPGA.

These S-Link words should be suppressed if all 12 threshold data bits and the Saturation and Parity Error bits are zero.

## 5.13 CMM-Jet- $E_T$ RoI

### 5.13.1 Input G-Link Data format

The G-Link format of the Jet- $E_T$  RoI data is identical to that for one slice of jet readout corresponding to the L1A as shown in [Figure 26](#). From this, the ROD checks the Bunch Crossing number, and then uses only the jet- $E_T$  hit data to build the RoI datum.

### 5.13.2 Output S-Link RoI data Format

The S-Link format of the Jet- $E_T$  RoI is included in [Figure 31](#). A single word is produced, containing the four-bit Jet- $E_T$  bits copied from incoming G-Link pin D16. The parity error bit is set if the incoming pin-level parity check failed on D16. There is no other addressing information.

The S-Link word should be suppressed if the Jet- $E_T$  value and parity error bits are zero.

## 5.14 CMM-Energy RoI

### 5.14.1 Input G-Link Data format

The format of Energy RoI data is identical to that for one slice of Energy-merging readout corresponding to one L1A as shown in [Figure 28](#). From this, the ROD checks the Bunch Crossing number, and then uses only the Total  $E_X$ ,  $E_Y$ ,  $E_T$  and Sum- $E_T$  and missing- $E_T$  threshold data to build the RoI datum.

### 5.14.2 Output S-Link RoI data Format

The S-Link format of the Energy RoI is included in [Figure 31](#). Three words are produced. The first contains the  $E_X$  total transverse energy component and overflow from G-Link pin D18. The second contains the  $E_Y$  component and overflow from G-Link pin D18 and the four Sum- $E_T$  threshold hits from D14. The third contains the unsigned  $E_T$  sum from G-Link pins D12 and D13 and the eight missing- $E_T$  threshold hits from D15. There is no addressing information.

The S-Link words are not zero-suppressed.

## 5.15 S-Link Status word formats

As indicated in [Table 7](#), two status words are included in the S-Link event fragment immediately before the trailer (these come at the very end of the S-Link packet, and are not to be confused with the sub-status words provided at the end of each sub-block). These provide an overall summary of the readout status, and adopt the convention that 0 indicates success and 1 indicates failure in each bit. The least significant half of each longword conforms to a common format agreed with the Level-1 Muon Trigger, while the remaining bits have private formatting. The format of the words is shown in [Figure 33](#)

Status word 1 indicates various fatal errors. Status word two gives other information to interpret the contents of the event fragment. Some commonly-defined errors cannot occur in the L1Calo ROD, and the bits are set to zero. These include event number mismatch and Internal Data Transfer Error.

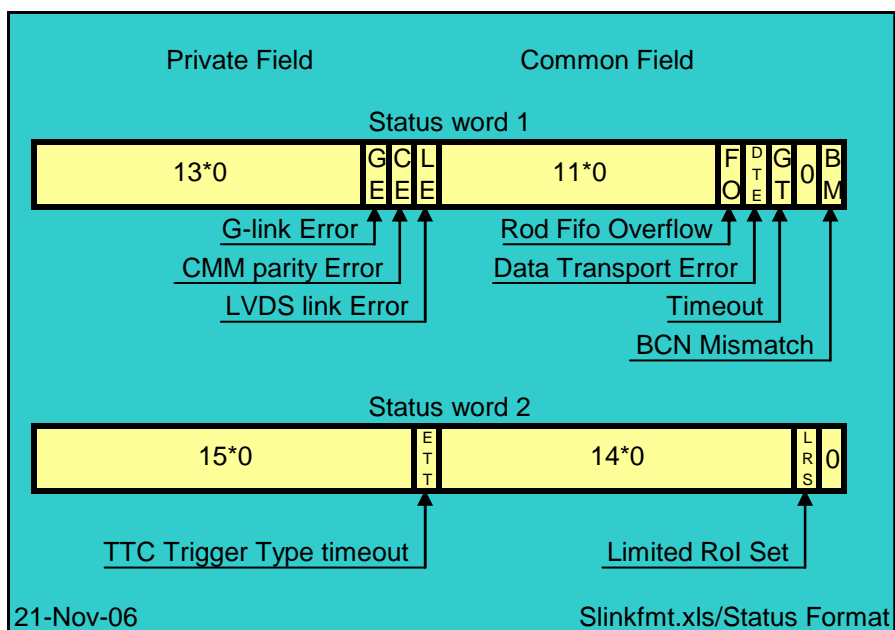


Figure 33: Format of S-Link status words.

Word	Field	Content
1	BM	BCN Mismatch detected on one or more Input FPGAs.
	GT	Timeout in switch FPGA due to non-arrival of incoming G-Link data.
	DTE	Data Transport Error – OR of private errors in word 1.
	FO	Data Overflow in one of the ROD FIFOs
	LE	LVDS link error between PreProcessor and CPM or JEM.
	CE	Parity error in CMM backplane or cable real-time data (can be set only if CMM data is included in the payload.)
	GE	G-Link error entering ROD (G-Link protocol or serial parity error). Detected in Input FPGAs.
2	LRS	Limited RoI Set - transmission was terminated at the maximum number of RoIs (can be set only for RoI data)
	ETT	Event type Timeout - timeout waiting for trigger type. Value of zero used in event fragment.

## **6 Manufacture, Testing and Maintenance**

### **6.1 Manufacturing**

An outside manufacturer makes the PCBs and performs the component assembly.

### **6.2 Testing**

#### **6.2.1 Test Strategy**

Testing is done in stages:

1. Boundary scan to detect and correct any assembly faults
2. Firmware simulation tests
3. Test the ROD data transfer using LabView environment
4. System tests with ATLAS DAQ software.
5. System tests including data transfer to a single board computer
6. Test RoI transfer to level-2 (RoIB) and ATLAS DAQ (ROS)
7. Event buffer data processing

A test plan was provided at the FDR.

#### **6.2.2 Test equipment**

- (1) 9U VME crate
- (2) 32 bit VME interface
- (3) Computer to run software
- (4) DSS module + G-Link Tx cards
- (5) S-Link RTM
- (6) Logic analyser
- (7) Oscilloscope
- (8) TTC system (TTCvi, TTCvx)
- (9) TCM
- (10) VME Extenders

### **6.3 Software**

A test engineer from the System Support Group will develop the LabView test software to carry out the initial standalone tests.

The test software for other tests will be developed within RAL PPD.

### **6.4 Installation**

The ROD module is designed to be used in a 9U CERN LHC-standard ROD crate with VME J0, J1, J2 and J3 backplanes. The module is provided with a front panel containing sockets and monitoring LEDs.

### **6.5 Maintenance and further orders**

This is a production module with an expected life span of at least five years. A spares policy has been defined so that replacement components are available. Facilities to continue firmware development for the expected life of the module are required.

## 7 Project Management

### 7.1 Personnel

		RAL Ext.	RAL Location
Customer:	C. N. P. Gee	6244	R1, 1.39
Project Manager:	V. Perera	5692	R68, 2.31
Project Engineers:	J. Edwards	6814	R68, 2.28
	I. Brawn	6816	R68, 2.23
	W. Qian	6128	R1, 1.36

### 7.2 Project Responsibilities

Development of the ROD module is shared between RAL TD and PPD. The principal responsibilities are as follows:

RAL E&ID:

1. Hardware: Schematic design and capture, module layout, component procurement, manufacturing oversight.
2. Firmware: VME FPGA and CPLD; Switch FPGA; Input FPGA (skeleton); Monitoring FPGA (initially only to access data, no data-dependent calculations).
3. Testing: Power, JTAG, FPGA loading, Register access (LabView environment).

RAL PPD:

1. Firmware: Input FPGA (G-Link to S-Link format conversions)
2. Testing: Detailed behavioural tests (online software environment).

The module has 10 G-Link data sources with different formats. A collaborative firmware management tool is being evaluated, and regular meetings are held to monitor progress and ensure command development paradigms.

### 7.3 Deliverables

Specifications of the deliverables are given in section 3

1. Initially two preproduction ROD Modules and firmware. When all tests are passed, 20 installed production modules, three in test rigs and three spares are needed – in total 26 modules.
2. Updated specification document with implementation details
3. JTAG test
4. Initial test with LabView software

### 7.4 Project plan (Milestones)

1. PDR 1 July 2003
2. Interim Design Review June 2004
3. Start testing August 2004
4. Combined Final Design/Production Readiness Review August 2006
5. Concluding Review – Date to be determined later.

## **7.5 Design Reviews**

The customer must be present at the PDR and the combined FDR/PRR.

The progress of the project is reported on a monthly basis in the ATLAS Calorimeter First-level Trigger project monitor form.

## **7.6 Training**

If required, training is carried out as required on the job. Special training may be required when using the PowerPC capabilities of the Xilinx FPGAs

## **7.7 CAE**

Cadence for schematic capture, VHDL and FPGA design tools, and LabView for test software.

## **7.8 Costs and finance**

All manufacturing, assembly and component costs are charged to FK42300.

## **7.9 Intellectual Property Rights (IPR) and Confidentiality**

All background and foreground Intellectual Property Rights in this project remain with CCLRC. The customer has unrestricted rights to items listed under deliverables (7.3). If the customer requires other data, then an appropriate protective agreement should be in place before releasing such data.

## **7.10 Safety**

General laboratory safety codes apply.

## **7.11 Environmental impact**

### ***7.11.1 Disposal***

RAL will dispose of the modules at end of their life.

### ***7.11.2 EMC***

Since the electronics must function as designed, without malfunction or unacceptable degradation of performance due to electromagnetic interference (EMI) within their intended operational environment, the electronics shall comply with specifications intended to ensure electromagnetic compatibility.

## **7.12 Handling Precautions**

Anti-static precautions must be taken when handling the module to prevent damage to expensive components.

## 8 References

- [1] ATLAS First-Level Trigger TDR, CERN/LHCC/98-14, at <http://ATLASinfo.cern.ch/ATLAS/GROUPS/DAQTRIG/TDR/tdr.html>
- [2] Central Trigger Processor Specification, [https://edms.cern.ch/file/107447/0/ctp\\_spec.pdf](https://edms.cern.ch/file/107447/0/ctp_spec.pdf)
- [3] PreProcessor Specification (under revision) <http://hepwww.rl.ac.uk/ATLAS-L1/Modules/Modules.html>
- [4] CPM Specification, <http://hepwww.rl.ac.uk/ATLAS-L1/Modules/Modules.html>
- [5] JEM Specification, <http://hepwww.rl.ac.uk/ATLAS-L1/Modules/Modules.html>
- [6] CMM Specification, <http://hepwww.rl.ac.uk/ATLAS-L1/Modules/Modules.html>
- [7] Low Cost Gigabit Rate Transmit/receive Chip set from Agilent: <http://literature.agilent.com/litweb/pdf/5966-1183E.pdf>
- [8] TTC documents at <http://www.cern.ch/TTC/intro.html>
- [9] S-Link specification at <http://www.cern.ch/hsi/S-Link/>
- [10] Level-1 Trigger Cabling specification, <https://edms.cern.ch/document/399348>
- [11] VME 64x specifications in the VMEbus Handbook, Peterson, 4<sup>th</sup> ed.
- [12] IEEE 1101.1 and IEEE 1101.10 specification
- [13] CERN Busy Module, [https://edms.cern.ch/file/319209/1/rod\\_busy\\_manual\\_2.pdf](https://edms.cern.ch/file/319209/1/rod_busy_manual_2.pdf)
- [14] BCID and LVL1ID after a Reset, <https://edms.cern.ch/document/405318>
- [15] Specification of the LVL1/LVL2 trigger interface, <https://edms.cern.ch/file/107485/1/L1L2020411.pdf>
- [16] Compressed format for PreProcessor data, at <https://edms.cern.ch/document/761046>
- [17] Dual optical receiver product datasheet [http://www.stratoslightwave.com/PDF/31-M2R\\_25\\_4\\_1\\_TL.pdf](http://www.stratoslightwave.com/PDF/31-M2R_25_4_1_TL.pdf)
- [18] Prototype ROD Specification, <http://hepwww.rl.ac.uk/ATLAS-L1/Modules/Modules.html>
- [19] Run number and run transitions, <https://edms.cern.ch/document/454588>
- [20] CERN S2VME64X description, <http://hsi.web.cern.ch/HSI/S-Link/devices/vme64x/>
- [21] HOLA S-Link interface, <http://hsi.web.cern.ch/HSI/S-Link/devices/hola/>
- [22] The Raw event format in the ATLAS Trigger & DAQ, ATL-D-ES-0019, Version 4.0, at <https://edms.cern.ch/document/445840>
- [23] Use of TTC System and BUSY Network, <http://hepwww.ph.qmul.ac.uk/11calo/doc/pdf/TTCBusy.pdf>
- [24] TCM Specification, <http://hepwww.rl.ac.uk/ATLAS-L1/Modules/Modules.html>
- [25] ATLAS Read Out Driver VME Interface - Draft specification, [http://atlas.web.cern.ch/Atlas/GROUPS/FRONTEND/documents/ROD\\_VME83.pdf](http://atlas.web.cern.ch/Atlas/GROUPS/FRONTEND/documents/ROD_VME83.pdf)
- [26] TTC Decoder Card, <http://hepwww.rl.ac.uk/ATLAS-L1/Modules/Components.html>



## 9 Update History

Draft 0.5 Initial draft for review.

Draft 0.54 Inserted introductory section including diagram of ROD organisation. Added Table of Contents. Split technical aspects and functional aspects into separate chapters. Added diagrams of S-Link packet structure. Added sections on link timeouts, event counter reset, details of ROD data rates. Updated S-Slink header details. Added VME, TTC and CAN addresses. Added update history chapter. Captioned all figures and renumbered all sections, tables, figures with cross-references. Added detail, figures and uniform structure for appendices.

Draft 0.6. Moved appendix A to become section 5 to eliminate problems with appendix headings (MSWord limitation). Renumbered references.

Draft 0.61. Rewrite parts of sections 3 and 4. Change to optical input, 18+2 channels.

Draft 0.62. Defined Register model for Main VME space. Defined interrupt capability. Promote register model to main section.

Draft 0.63. Add neutral event format and common block header & trailer.

Draft 0.7 Implement review comments. Change ROD Block Diagram. Revised S-Link transition module pin-out. Added VME J0 pinout. Update memory map for motherboard and add spy buffer details to switch FPGA.

Draft 0.71. Clarify sampling details for TTC correlated sampling. Add TTC SampleId Register.

Draft 0.73. Updated G-Link input formats

Draft 0.74. Corrected figs 6 & 7.

Draft 0.75. Changed Input FPGA channel control pulse register to register for all channels. Inserted diagrams and descriptions of data formats.

Draft 0.76. Inserted calculation of data rates in tables. Corrected S-Link header to Spec v 2.4. Defined FPGA version registers. Minor tidy to Input FPGA register map.

### 9.1 *Version 1.0 (Post-review):*

Many minor text clarifications

Update to CMM-Energy formats for mapped Sum- $E_T$  and Missing- $E_T$  Bits (G-Link & S-Link).

Change PreProcessor and JEM formats to use 16-bit G-Link mode. Update to JEM S-Link format to match CPM hit format.

Update to Jet RoI Format (F bit moved).

Added orbit count FIFO and register connected to S-Link ROD header.

Description of Energy RoI added.

Defined TTC command for synchronised event sampling

Added overflow bits for switch FIFOs in switch status register 2

Defined CMM G-Link and S-Link format for forward jet readout

Modified Xoff-time counters to detect update of data during VME readout.

Added registers for S-Link user header with register support.

Added System Ace register definitions.

Changed routing of G-Links 17/18 to go to S-Link 3 rather than 4.

### 9.2 *Version 1.05*

Added description of local event counter.

Corrected figure illustrating PPM S-Link sequencing

Correct PPM G-Link format to 16-bit G-Link

Add resetting of Orbit Counter

JEM DAQ – added definition of forward jet fields. Changed S-Link format to include source IDs for hit words.

Detailed corrections to CMM S-Link formats

Added definition of User event header

Added definition of FADC data in PPM readout.

Added link down counters. Added Input FIFO occupancy monitor. Minor updates to names of data fields in formats for internal consistency between G-Link and S-Link names.

### **9.3 Version 1.06**

Update to PPM G-Link format – simplification by removing ASIC headers.

Correct spellings and broken links.

PPM Compressed format – error reporting deferred to sub-status at end of last data block.

PPM Super compressed – send channel hit map rather than channel number at head of packed data.

Further update to PPM format to specify error field, remove event number from G-Link, define error S-Link block and compressed error formats.

Removed System Ace registers from VME MAP (not implemented). Left reserved space.

Reordered VME MAP.

Updates for Word-IDs for Energy RoIs, PPM DAQ, neutral format and User Header (from rod9UBMB04).

Update neutral format, explicitly set to zero BCnumber check errors.

### **9.4 Version 1.07**

S-Link numbers in diagrams run 0-3; G-Link numbers run 0-17.

Remove checkpoint handling requirement (abandoned by ATLAS)

Specify DAV gaps required.

Minor clarifications to text.

Added module-specific error bit to substatus word

Updated ROD header to ATLAS version 3.0

Added reference to D Sankey ROD formats

### **9.5 Version 1.08**

Created for FDR/PRR

Fixed text overlay problem in figures4 & 5 and pagination problem.

### **9.6 Version 1.09**

Post-FDR Version containing review updates. Large number of minor text updates and clarifications.

PPM Compressed format firmware type changed to 16.

CAN Register A bits defined (register was already present).

Unused Switch registers at 0x40 to 0x48 removed.

ALL data formats: LS1 and NS2 fields split 5+3, to benefit PreProcessor.

PreProcessor – separated MCM disable bit into 4 separate channel disable bits.

RoI details changed – Energy RoI and specification of JEM RoI location

Formatting changes copied into L1L2 interface doc.v1.3c

Switch register map updates: TrigtypeTimeout register duplicate removed at 0x070; description of TTC timeout added;

SourceID duplicate removed in switch at 0x18. Reordered switch register descriptions to match address map. Correct length and describe spy buffer format.

## **9.7 Version 1.1**

Updates to switch FPGA relating to event sampling and ERC Error Reset behaviour:  
Added Control Register bits to enable ECR Error reset and Synchronous Sampling; Added pulse register bits to generate errors for testing error recovery; Added Status register bits to display individual inputs into BUSY; added delay register for generation of ECR Reset signal; added ECR Error Counter; added pulse register bit to clear ECR error counter.

Change Orbit Counter register to 16 bits as in firmware.

Make Format Version register software R/W (was R/O) and describe values.

Update allocation of ROD addresses and functions in section 3.8, table 10 and figure 3.

Corrected text describing FPGA Code Type as four bits – should read eight bits like tables.

## **9.8 Version 1.2**

Changes to firmware codes described in table 2, to allow all Input FPGA codes to fit in the 4-bit interface in the common Input FPGA code interface.

Change in definition of Common User Header, to allow inclusion of FADC compression lower bound.

## **9.9 Version 1.2.1**

Propagate firmware code changes in table 2 to register descriptions for individual FPGAs, e.g. 4.6.10

## **9.10 Version 1.2.2**

Removed redundant Event Length descriptor from Switch FPGA.

Inserted G-Link Timeout Register, Last event L1ID Register, Total L1A Number Register, Last TTC L1ID Register.

Added switch pulse register bit to clear Total L1A number register.

## Appendix A S-Link rear transition module Connector Pin-out

### A.1 5-row P2 connector map to two S-LINK LSCs

		<b>Z</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>LSC SLOT A</b>	1	UD3	UD2	Vcc	UD1	UD0
	2	Gnd	UD5	Gnd	Gnd	UD4
	3	UD7	Gnd	nc	UD6	Gnd
	4	Gnd	UD10	nc	UD9	UD8
	5	UD14	UD13	nc	UD12	UD11
	6	Gnd	UD17	nc	UD16	UD15
	7	UD20	UD19	nc	Gnd	UD18
	8	Gnd	UD23	nc	UD22	UD21
	9	UD25	Gnd	nc	UD24	Gnd
	10	Gnd	UD28	nc	UD27	UD26
	11	UCTRL#	UD31	nc	UD30	UD29
	12	Gnd	UWEN#	Gnd	Gnd	UDW0
	13	UCLK	Gnd	Vcc	UDW1	UTEST#
	14	Gnd	LDOWN#	nc	LFF#	URESET#
	15	LRL1	LDERR#	nc	LRL0	Gnd
	16	Gnd	LRL3	nc	LRL2	Gnd
<b>LSC SLOT B</b>	17	UD3	UD2	nc	UD1	UD0
	18	Gnd	UD5	nc	Gnd	UD4
	19	UD7	Gnd	nc	UD6	Gnd
	20	Gnd	UD10	nc	UD9	UD8
	21	UD14	UD13	nc	UD12	UD11
	22	Gnd	UD17	Gnd	UD16	UD15
	23	UD20	UD19	nc	Gnd	UD18
	24	Gnd	UD23	nc	UD22	UD21
	25	UD25	Gnd	nc	UD24	Gnd
	26	Gnd	UD28	nc	UD27	UD26
	27	UCTRL#	UD31	nc	UD30	UD29
	28	Gnd	UWEN#	nc	Gnd	UDW0
	29	UCLK	Gnd	nc	UDW1	UTEST#
	30	Gnd	LDOWN#	nc	LFF#	URESET#
	31	LRL1	LDERR#	Gnd	LRL0	Gnd
	32	Gnd	LRL3	Vcc	LRL2	Vcc

## A.2 5-row P3 connector map to two S-LINK LSCs

(uses P2-type backplane)

		<b>Z</b>	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>
<b>LSC SLOT C</b>	1	UD3	UD2	3.3V	UD1	UD0
	2	Gnd	UD5	Gnd	Gnd	UD4
	3	UD7	Gnd	nc	UD6	Gnd
	4	Gnd	UD10	nc	UD9	UD8
	5	UD14	UD13	nc	UD12	UD11
	6	Gnd	UD17	nc	UD16	UD15
	7	UD20	UD19	nc	Gnd	UD18
	8	Gnd	UD23	nc	UD22	UD21
	9	UD25	Gnd	nc	UD24	Gnd
	10	Gnd	UD28	nc	UD27	UD26
	11	UCTRL#	UD31	nc	UD30	UD29
	12	Gnd	UWEN#	Gnd	Gnd	UDW0
	13	UCLK	Gnd	3.3V	UDW1	UTEST#
	14	Gnd	LDOWN#	nc	LFF#	URESET#
	15	LRL1	LDERR#	nc	LRL0	Gnd
	16	Gnd	LRL3	nc	LRL2	Gnd
<b>LSC SLOT D</b>	17	UD3	UD2	nc	UD1	UD0
	18	Gnd	UD5	nc	Gnd	UD4
	19	UD7	Gnd	nc	UD6	Gnd
	20	Gnd	UD10	nc	UD9	UD8
	21	UD14	UD13	nc	UD12	UD11
	22	Gnd	UD17	Gnd	UD16	UD15
	23	UD20	UD19	nc	Gnd	UD18
	24	Gnd	UD23	nc	UD22	UD21
	25	UD25	Gnd	nc	UD24	Gnd
	26	Gnd	UD28	Rs-IN	UD27	UD26
	27	UCTRL#	UD31	Rs-OUT	UD30	UD29
	28	Gnd	UWEN#	Rs-DTR	Gnd	UDW0
	29	UCLK	Gnd	Rs-RTS	UDW1	UTEST#
	30	Gnd	LDOWN#	Rs-CTS	XOFF#	URESET#
	31	LRL1	LDERR#	Gnd	LRL0	Gnd
	32	Gnd	LRL3	3.3V	LRL2	3.3V

Note: Signal names starting with U are input to S-Link

Signal names starting with L are output from S-Link

Signal names ending with # are active low signals

UDW0 and UDW1 indicate S-Link data width (both low for 32 bit wide data).

LDERR# signal only used in destination cards. e.g. loop back capability with destination card

## Appendix B VME 64x J0 Connector Pin Assignment

Pin	Row f	Row e	Row d	Row c	Row b	Row a	Row Z
1							
2							
3							
4							
5							
6							
7							
8							
9			TTCrx+				
10			TTCrx-				
11							
12							
13		GA<7>					
14		GA<6>					
15		GA<5>					
16		CAN-	CAN+				
17							
18							
19							

## Appendix C

## Stiffening and ESD strip details for 9U ROD

1. Isolate the front panel from ground plane (grounded through chassis)
2. Stiffening bars made out of standard 5mm x 10 mm aluminium bars
3. Stiffening bar holes tapped through for M2.5 threaded bolts
4. Place the horizontal bars 6 mm from edge of PCB and 10 mm from front edge of PCB
5. Place the vertical bar 6 mm from the bottom edge of PCB and 3 mm from the edge of the connector (optional).
6. Electrical connection between the stiffening bars, connected to the chassis via front panel connection to the board
7. Connect each ESD strip to ground plane via two 1 M $\Omega$  resistors in series
8. Width (w) of the ESD strip is  $5 < w > 2.5$  mm gold finished
9. Note the drawing is not to scale

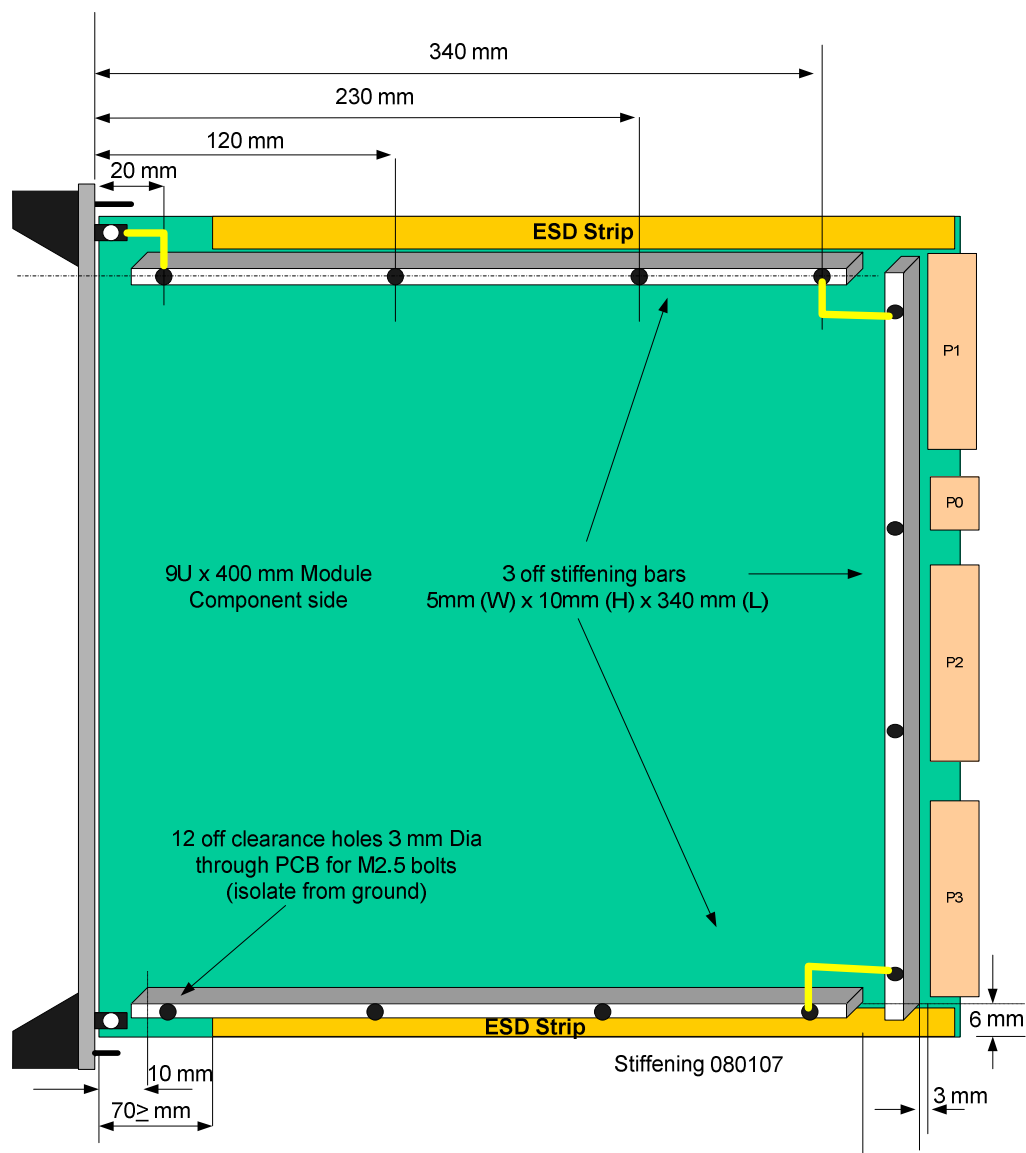


Figure 34: Stiffening and ESD strip details for 9U ROD.