ATLAS Muon MDT

MROD Module:

MROD-X-Out Programmer's Manual

Document Version:	1
Document Issue:	0
Document ID:	ATLAS-TDAQ-2005-XXX or EDMS Id XXXXXX
Document Date:	25-Oct-2005
Document Status:	Draft

Abstract

The MROD-X module can be broken up into functional subsystems (four MROD-Ins and one MROD-Out). This document describes the details of the MROD-Out from a programmers point of view.

Institutes and Authors:

NIKHEF Amsterdam: Peter Jansweijer

Radboud University Nijmegen: Thei Wijnen

Table 0 Document Change Record

Title:	ATLAS MROD Module: MROD-X-Out Programmers Manual		
ID:	ATLAS-TDAQ-2005-XXX or EDMS Id XXXXXX		
Version	**Issue**	**Date**	**Comment**
1	0	25-Oct-2005	First version (draft)

Table of Contents

1 Introduction

The output part of the MROD-X module consists of one or two SHARC processors and a FPGA (called MROD-Out FPGA). The MROD-Out has four basic functions:

- Build events from the event fragments received from the MROD-Ins
- Connect to VMEbus
- Receive TTC information.
- Drive the ROL (S-LINK)

The MROD-X is a successor of the MROD-1 module. The SHARC processors in the MROD-1 module took care of the data transport from MROD-Ins to ROL. The MROD-X has two options, either run in MROD-1 mode where the data transport is done via the SHARC links or run in a mode where the data is transported via RocketIO links [1] between the MROD-In channels and the MROD-Out. In the latter case, the SHARC processors can be used for spying on the data.

2 Programmer's viewpoint VMEbus

The VME interface of the MROD module accepts the following VME cycles:

SHARCs: A32: D32
RMW, BLT, Single Cycle, as either Supervisory or Non Privileged and as either Program or Data access.

CR/CSR: A24: D32, D16, D08(EO), D08(O)
RMW, Single Cycle.

IRQ: I(N) D08(O) ROAK where N is a selectable IRQ Level [1:7]

AM 0x10: User-Defined AM.

2.1 SHARC processors

The IOP registers and the internal memory of the SHARC processors are mapped into A32 address space of VMEbus. The only transfers that are accepted are D32 transfers. Other transfer types (D16, D08 etc.) will cause a BERR_n to be generated. These transfers can be of type Single Cycle, BLT or RMW.

The memory map of each SHARC stretches from 0x00000000 to 0x000FFFFF (32 bit normal word). This memory can be accessed from VME via the SHARC host bus (see also figure 5-7 and figure 7-1 of the "SHARC DSP Hardware Reference" [2]; the host bus accesses contiguous addresses therefore note that SHARC A[0] is used).

Seen from the VME side this corresponds with the address range 0x00000000 to 0x003FFFFC (bits A[0] and A[1] are omitted since VMEbus is byte oriented).

The two SHARC processors are mapped into VME address space according to table 1.

A[31:28]	A[27:23]	A[22]	A[21:2]	A[1]	A[0]
"0000"	BAR [7:3]	"0"	Internal Memory Space SHARC A	0	0
		"1"	Internal Memory Space SHARC B		

Table 1: SHARC memory map in VMEbus address space

Five base address bits in the Base Address Register (See BAR in chapter 2.3.12 "Configuration Rom / Control and Status Register Capability" in the VMEbus specification [3]) determine the base address of the SHARC processors in the VME-crate. VMEbus address bits A[27:23] are compared to the BAR [7:3]. If a match occurs then the SHARC processors will be selected. Summarized, the VME address offset from the Base address is 0x00000000–0x003FFFFC for SHARC A, and 0x00400000–0x007FFFFC for SHARC B.

2.2 CR/CSR

CR/CSR of the module is accessed through A24, CR/CSR (AM Code 0x2F) cycles.

The module accepts accesses tot the range of addresses belonging to the VME64x defined CR space (0x00000-0x00FFF) and the VME64x defined CSR space (0x7FC00-0x7FFFF). See the VME64x specification Chapter 10 [4].

The full defined CR space is implemented in ROM (1024 bytes). The memory is currently filled with a VME64 CR format, which covers only 32 bytes (see VME64, Table 2-32 [3]).

Note that the CR area is read only. Writing to CR space causes a BERR_n to be asserted.

Although the complete defined CSR space address range is decoded, only the VME64 defined registers are implemented. The VME64 defined CSR area contains 3 addresses described below.

2.2.1 Offset 0x7FFFF: BAR

Upon reset (VME SYSRESET) the module detects if it is put into a VME or VME64 backplane. This is detected because the geographical address pins in such crates are open. This means they will be pulled high on the board so the slot number will be zero with a parity error (VME64x, Observation 3.6 [4]). When this is the case, BAR [7:3] is loaded with the value of a Local BAR jumper (J22).

When the module detects it is plugged into a VME64x backplane then BAR [7:3] is loaded with the inverted value of the geographical address pins (GA_n [4:0]) which corresponds to the slot number. However, when a parity error is detected, BAR [7:3] is set to the “Amnesia address” 0x1E (VME64x, Recommendation 3.8 [4]).

BAR [7:3] can be read and written by a VME A24 cycle to CR/CSR (AM Code 0x2F).

Note that BAR [7:3] correspond to Address bits A[23:19] that are used to decode the boards CR/CSR in the A24 address space, so by writing the BAR the Base Address of this boards CR/CSR changes!

BAR [2:0] are not used.

2.2.2 Offset 0x7FFFB: Bit Set Register

Writing bits ‘1’ in this register sets the corresponding bits [7:0] of the Bit Set Register. Writing a bit ‘0’ has no effect to the corresponding bit. Reading the register returns the status of the bits. Note that reading the Bit Set Register yields the same result as reading the Bit Clear Register. See also VME64, Table 2-31 [3] and VME64x, Table 10-6 [4].

The following bits are implemented:

Bit [7]: Put the board in Reset mode.

This causes the reset pin of the SHARC processors to be asserted for 375 ns. Note that the internal CSR-Reset signal is set by writing a ‘1’ to bit [7] of the Bet Set Register. A ‘0’ to ‘1’ transition of the internal CSR-Reset signal generates a SHARC reset pulse. The SHARC processors cannot be held in reset since they contain the arbiter for the host bus. The VMEbus needs to use the host bus when it wants to do a transfer (for example clear the internal CSR-Reset signal by writing

to the Bit Clear Register). Software (VME) determines the status of the internal CSR-Reset signal.

Bit [6]: SYSFAIL driver enable.

This causes a '1' on the SYSFAIL_En pin. Software determines the status of this pin. The pin is currently not used.

Bit [5]: Module Failed.

Hardware determines the status of this bit. Software can set this bit for test purposes (ModuleFail is currently hardwired FALSE in the FPGA). Note that the definition for testing this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit [4]: Module Enable.

This causes a '1' on the Module_En pin. Software determines the status of this pin. The pin is currently not used. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit [3]: Module issued BERR_n.

Hardware determines the status of this bit. This bit is set whenever the module issued a BERR_n. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit [2]: CRAM Owned.

Hardware determines the status of this bit. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard. This bit is currently not implemented.

Bit [1]: Reserved

Bit [0]: Reserved

2.2.3 Offset 0x7FFF7: Bit Clear Register

Writing bits '1' in this register clears the corresponding bits [7:0] of the Bit Clear Register. Writing a bit '0' has no effect to the corresponding bit. Reading the register returns the status of the bits. Note that reading the Bit Set Register yields the same result as reading the Bit Clear Register. See also VME64, Table 2-31 [3] and VME64x, Table 10-7 [4].

The following bits are implemented:

Bit [7]: Get the board out of Reset mode.

This bit clears the internal CSR-Reset signal. Note that the reset pin of the SHARC processors is asserted for 375 ns by a '0' to '1' transition on the internal CSR-Reset signal. The SHARC processors cannot be held in reset since they contain the arbiter for the host bus (see also Bit Set Register above). Software (VME) determines the status of the internal CSR-Reset signal.

Bit [6]: SYSFAIL driver disable.

This causes a '0' on the SYSFAIL_En pin. Software determines the status of this pin. The pin is currently not used.

Bit [5]: Module Failed.

This clears the module Fail flag. Hardware determines the status of this bit (ModuleFail is currently hardwired FALSE in the FPGA). Software can clear the

bit. Note that the definition for testing this bit is not in the VME64 standard, but it is in the VME64x standard.

Bit [4]: Module Enable.

This causes a '0' on the Module_En pin. Software determines the status of this pin. The pin is currently not used. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit [3]: Clear BERR_n Flag.

Hardware determines the status of this bit. This bit is set whenever the module issued a BERR_n. Software can clear this flag. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit [2]: CRAM Owned.

Hardware determines the status of this bit. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard. This bit is currently not implemented.

Bit [1]: Reserved

Bit [0]: Reserved

2.3 Interrupt VMEbus

The module can issue a VME interrupt request. The interrupt priority level can be set under control of the SHARC processors to any of the levels 7 to 1. When the interrupt is acknowledged by an interrupt acknowledge cycle, the module puts an 8 bit status/ID on the VMEbus. The 8 bit Status/ID is programmable. The module releases the VME IRQ line when the interrupt is acknowledged (Release On Acknowledge, ROAK).

VMEbus Mnemonic: I(N) D08(O) ROAK where N = [1:7]

The Interrupt is issued using the "VMEbus BAR and IRQ" register (0x01). The Status-ID is programmed in the "VMEbus IRQ Level and IRQ Status-ID Pattern" register (0x00). For a description of both registers see "Programmers viewpoint: SHARC".

2.4 AM10

The User-Defined AM10 is used as an ultimate way to reset the MROD-X. The whole MROD-X module is being reset when a VME-bus cycle is generated with AM10 on the Base Address of the module (see table 2). This reset is equal to a VME SYSRESET. All FPGAs are re-configured automatically. Note that this reset will be generated with a delay of 1 μs. in order to enable the VME interface (in the MROD-Out FPGA) to finish the AM10 cycle properly before all FPGAs are taken into a reconfiguration cycle.

A[31:24]	A[23:19]	A[18:1]
X	BAR [7:3]	X

Table 2: AM10 Reset Address Decoding

3 Programmer's viewpoint SHARC

3.1 SHARC memory regions:

The SHARC uses a 64 bit data bus to interface to external memory. The data bus of the FPGA is connected to bits [63:32] of the SHARC data bus. The SHARC should use "32 bit normal word addressing" on *odd* address (see also figure 7-1 and the note on page 7-3 of the "SHARC DSP Hardware Reference" [2]).

Memory Region	Function	EBxWS	EBxAM	Remarks
MS0	Internal registers	100	00	1, 2
MS1	Read TTC Event/Bunch-ID and Trigger-Type	001	00	1,3
MS2	Read Spy Data	001	00	1,3
MS3	S-LINK Interface	000	00	1, 4

Table 3: SHARC memory regions

Remarks:

1: EBxWS and EBxAM are sub-patterns of the Wait register described in table 5-4 and table 5-5 of the "SHARC DSP Hardware Reference" [2].

2: EBxWS = 100 means 4 Wait, 1 Hold Cycle
EBxAM = 00 Both Internal and External Acknowledge required

3: EBxWS = 001 means 1 Wait, 0 Hold Cycles
EBxAM = 00 Both Internal and External Acknowledge required

4: EBxWS = 000 means 0 Wait, 0 Hold Cycles
EBxAM = 00 Both Internal and External Acknowledge required

3.1.1 MS0 address space of the SHARC: Control and Status Registers

The MROD-Out SHARC processors are connected to:

- VMEbus through a VME interface (See: "Programmers viewpoint VMEbus")
- Event Builder (in the MROD-Out FPGA)
- TTC (Timing Trigger and Control) interface [5] via the TIM [6].
- S-LINK output (See also MS3 address space)

These items are controlled by a set of Control and Status registers which reside in the MS0 address space of the SHARC processors. Table 4 is a list of all the registers.

Register	MS0 offset	Function	Write	Read	Default After Reset
0x00	0x01	VMEbus IRQ Level and IRQ Status-ID Pattern	0x-----snn	0x00000snn	0x00000000
0x01	0x03	VMEbus BAR and IRQ	0x-------p	0xpns000nt	0x0pns0000 1)
0x02	0x05	S-LINK Status and Interrupt Register	0x----rsn-	0x0000snnn	0x00000800 2)
0x03	0x07	TTC Control/Status and Interrupt Register	0x-------p	0x0000000p	0x00030010
0x04	0x09	Resets, LEDs and Status	0x--nn-pnn	0x00nn0qnn	0x0000020F 3)
0x05	0x0B	Date & Revision ID Register	0x--------	0xyymmddxy	0xyymmddxy 4)
0x06	0x0D	ID1 Register	0x--------	0xiiiiiiii	0xiiiiiiii 5)
0x07	0x0F	ID2 Register	0x--------	0xffcciiii	0x01cciiii 5)
0x08	0x11	FPGA Temperature Register	0x-------q	0x0000000q	0x00000008 6)
0x09	0x13	Channel Enable Register	0x----snnn	0x0pnnsnnn	0x000063FF
0x0A	0x15	Format Version Register	0xnnnnnnnn	0xnnnnnnnn	0x03000000
0x0B	0x17	Module ID Register	0xnnnnnnnn	0xnnnnnnnn	0x00610080
0x0C	0x19	Run Number Register	0xnnnnnnnn	0xnnnnnnnn	0x00000000
0x0D	0x1B	Detector Event Type Register	0xnnnnnnnn	0xnnnnnnnn	0x00000000
0x0E	0x1D	BOT (TDC-Header) Pattern	0xnnnnnnnn	0xnnnnnnnn	0xA0000000
0x0F	0x1F	Reserved	0x--------	0x00000000	0x00000000
0x10	0x21	BOT Mask	0xnnnnnnnn	0xnnnnnnnn	0xE0000000
0x11	0x23	Reserved	0x-------0	0x00000000	0x00000000
0x12	0x25	EOT (TDC-Trailer) Pattern	0xnnnnnnnn	0xnnnnnnnn	0xC0000000
0x13	0x27	Reserved	0x--------	0x00000000	0x00000000
0x14	0x29	EOT Mask	0xnnnnnnnn	0xnnnnnnnn	0xF0000000
0x15	0x2B	Reserved	0x--------	0x00000000	0x00000000
0x16	0x2D	BOEF (Magic Word) Pattern	0xnnnnnnnn	0xnnnnnnnn	0x8F000000
0x17	0x2F	Reserved	0x--------	0x00000000	0x00000000
0x18	0x31	BOEF Mask	0xnnnnnnnn	0xnnnnnnnn	0xFF000000
0x19	0x33	Reserved	0x-------0	0x00000000	0x00000000
0x1A	0x35	LWC (Link Word Count) Pattern	0x--------	0xnnnnnnnn	0x81000000
0x1B	0x37	Reserved	0x--------	0x00000000	0x00000000
0x1C	0x39	LWC Mask	0xnnnnnnnn	0xnnnnnnnn	0xFF000000
0x1D	0x3B	Reserved	0x--------	0x00000000	0x00000000
0x1E	0x3D	Reserved	0x--------	0x00000000	0x00000000
0x1F	0x3F	Reserved	0x--------	0x00000000	0x00000000

0x20	0x41	Reserved	0x--------	0x00000000	0x00000000
0x21	0x43	Reserved	0x--------	0x00000000	0x00000000
0x22	0x45	Reserved	0x--------	0x00000000	0x00000000
0x23	0x47	Reserved	0x--------	0x00000000	0x00000000
0x24	0x49	Reserved	0x--------	0x00000000	0x00000000
0x25	0x4B	Reserved	0x--------	0x00000000	0x00000000
0x26	0x4D	TWC (Trailer Word Count) Pattern	0xnn------	0xnn000000	0x8A000000
0x27	0x4F	Reserved	0x--------	0x00000000	0x00000000
0x28	0x51	TWC Mask	0xnnnnnnnn	0xnnnnnnnn	0xFF000000
0x29	0x53	Reserved	0x--------	0x00000000	0x00000000
0x2A	0x55	Error-Code Replace Patterns	0xn---n---	0xn000n000	0x5000D000
0x2B	0x57	Extended Event-ID	0xnn------	0xnnnnnnnn	0xFF000000
0x2C	0x59	RocketIO Status	0x--------	0xnnnnnnnn	0x0000FFFF
0x2D	0x5B	RocketIO Link Down, Spy Channel Full Interrupt (IRQ1)	0x000nnnnn	0x000nnnnn	0x00000000
0x2E	0x5D	Spy Event Length FIFO	0x--------	0x--------	0xnnnnnnnn
0x2F	0x5F	Spy Event Pre-scale	0x000nnnnn	0x000nnnnn	0x00000000
0x30 - 0x3F	0x61 - 0x7F	Reserved	0x--------	0x--------	0x--------

Remarks:
‘n’ is any hexadecimal number
‘-’ is don’t care
‘p’ is either 0x00 or 0x01 ‘q’ is 0x00, 0x01, 0x02 or 0x03
‘r’ is 0x00, 0x01, 0x04 or 0x05 ‘s’ is one out of 0x00 to 0x07
‘t’ is one out of 0x00, 0x02, 0x04, 0x06, 0x08, 0x0A, 0x0C, 0x0E
‘yymmdd’ is Date ID (year/month/day) ‘xy’ is Reversion ID: x.y
‘ff’ is DS2401 family ID ‘cc’ is DS2401 CRC value
‘iiiiiiiiiii’ is 48 bit DS2401 Identifier
Note: 1) Default value read depends on the VME Crate and BAR settings
Note: 2) Default value read depends on the status of LDOWN_n and LFF_n
Note: 3) Default value read depends on the status of ASP_n
Note: 4) Value depends on the current version of the FPGA content
Note: 5) Value depends on the Dallas DS2401 content
Note: 6) Default value read depends on the status of ALERT_n

Table 4: Registers in the MS0 address space

3.1.2 Register 0x00: VMEbus IRQ Level and IRQ Status-ID Pattern

MS0 offset 0x01

Register 0x00 Bits [7:0] can be programmed by the SHARC processors with an 8 bit Status-ID pattern which is put onto the VMEbus when an interrupt acknowledge cycle is accepted (See VME64 Chapter 4 [3]).

Bits [10:8] determine to which VMEbus IRQ level the interrupt will be routed (see table 5) when it is triggered through the VMEbus BAR and IRQ register 0x01. Note that "000" will map to IRQ [7] which is the highest level.

Bits [10:8]	IRQ Level
111	7
110	6
101	5
100	4
011	3
010	2
001	1
000	7

Table 5: VMEbus IRQ Level select (register 0x00)

3.1.3 Register 0x01: VMEbus BAR and IRQ

MS0 offset 0x03

The MROD-Out SHARC processors can assert an interrupt on the VMEbus when it writes a '1' into bit [0] of this register (see table 6). The interrupt is routed to one of the seven VMEbus interrupt lines IRQ [7:1] according to the IRQ Level Select Bits in register 0x00. No interrupt will be triggered when there is an interrupt pending on any of the levels. Reading register 0x01 gives information on the VME Base Address Register and any pending interrupt. When one of the bits [7:1] is read as '1' then the corresponding IRQ level is pending. These bits are cleared by the hardware when the interrupt is acknowledged by VMEbus (See also register 0x00: VMEbus IRQ Level and IRQ Status-ID Pattern).

Furthermore the value of the Base Address Register that is loaded upon a System Reset can be read from register 0x01.

All other bits of register 0x01 are read back as '0'.

Data	Description	Write	Read
0	IRQ Trigger	’1’ Trigger VME Interrupt	‘0’
7:1	VME IRQ(7:1) status	X	‘1’ if VME IRQ [7:1] is pending.
18:8		X	0x00
23:19	Base Address read back	X	BAR [7:3]
24	Module in VME64x Crate	X	‘0’: Module in VME64x Crate ‘1’: Module in VME or VME64 backplane
31:25	X	X	0x00

Table 6: VMEbus BAR and IRQ Register (0x01)

3.1.4 Register 0x02: S-LINK Status and Interrupt Register

MS0 offset 0x05

The S-LINK Link Return Lines (LRL [3:0]) and the status of the LDOWN_n signal can be read through register 0x02.

An interrupt can be generated on the IRQ0_n line of each SHARC whenever the status of the Link Return Lines changes, when LDOWN_n is asserted or when the S-Link FIFO Half Full flag is asserted. Each interrupt source can be masked.

The data format of the S-LINK Status and Interrupt Register is described in table 7.

A LDOWN_n condition while the LDOWN interrupt is enabled (see bit [6]) leads to an interrupt. The LDOWN interrupt can be cleared by writing a ‘1’ to bit [7] of this register.

Note that bit [11] will read the direct status of the LDOWN_n signal. This bit will read ‘0’ when LDOWN_n is asserted (active low).

Note that bit [9] will read a ‘1’ when the S-Link FIFO Half Full flag is asserted. This bit will read the FIFO status independently of the mask bit for the interrupt (bit [8]).

Data	Description	Write	Read
3:0	S-LINK LRL bits	X	S-LINK LRL(3:0)
4	Enable/Disable LRL Change Interrupt	‘0’: Disable Interrupt ‘1’: Enable Interrupt	‘0’: Interrupt Disabled ‘1’: Interrupt Enabled
5	LRL Change Interrupt	‘0’: No action ‘1’: Clear Interrupt	‘0’: No IRQ Pending ‘1’: IRQ Pending
6	Enable/Disable LDOWN Interrupt	‘0’: Disable Interrupt ‘1’: Enable Interrupt	‘0’: Interrupt Disabled ‘1’: Interrupt Enabled
7	LDOWN Interrupt	‘0’: No action ‘1’: Clear Interrupt	‘0’: No IRQ Pending ‘1’: IRQ Pending
8	S-Link FIFO Half-Full Interrupt Enable	‘0’: Disable Interrupt ‘1’: Enable Interrupt	‘0’: Interrupt Disabled ‘1’: Interrupt Enabled

9	Clear S-Link FIFO Half-Full Interrupt or Read status	'0': No action '1': Clear Interrupt	'1': S-Link FIFO Half Full asserted
10	S-Link Test Mode	'0': Normal Operation '1': Test Mode	'0': Normal Operation '1': Test Mode
11	LDOWN_n Status	X	'0' S-Link is Down '1' S-Link is Up
12	LFF_n Status (latched)	'0': No action '1': Clear Latched LFF_n Status	'0' LFF_n asserted '1' LFF_n de-asserted
13	LFF_n Status	X	LFF_n
14	S-Link Flush Mode	'0': Default '1': Flush S-Link	'0': Default '1': Flush S-Link
31:15	X	X	0x00

Table 7: S-LINK Status and Interrupt Register (0x02)

Bit [10] controls the S-Link test mode. When Bit [10] is set '1' the UTEST_n signal on the S-Link is made active (UTEST_n = '0').

Bit [12] will read the latched status of the LFF_n signal. This bit will read '0' when the S-Link FIFO Full signal is asserted. Writing a '1' to this register clears the latched status. Bit [13] will read the direct (unlatched) status of the LFF_n signal. This bit will read '0' when the S-Link FIFO Full signal is asserted (active low).

Bit [14] is the S-Link Flush Mode bit. When this bit is set to '1' then the system is behaving in a way as if the S-Link LSC is always able to send data (S-Link is always up and never full). When in Flush Mode the data is not written into the S-Link (i.e. S-Link FIFO write request is kept inactive). ***Be careful*** using this bit! It is meant for debugging and rate testing purposes only. During normal operation ***this bit should always be set to '0'***.

3.1.5 Register 0x03: TTC Control/Status and Interrupt Register

MS0 offset 0x07

The MROD-X module receives TTC information via the TIM [6] which distributes this information via the P3 backplane. The status of the Event/Bunch-ID and Trigger-Type FIFOs can be monitored in register 0x03 (see table 8).

There are two sets of Event/Bunch-ID and Trigger-Type FIFOs. One is connected to the event builder. The other is connected to the SHARC (see "MS1 address space of the SHARC: TTC information"). Both sets work independently.

3.1.5.1 Event/Bunch-ID and Trigger-Type FIFOs connected to the SHARC:

The SHARC Event/Bunch-ID and Trigger-Type FIFOs are both 511 words deep.

When the Event/Bunch-ID FIFO becomes full then an interrupt can be generated on the IRQ2_n line of each SHARC. The full interrupt status is indicated by bit [0] being a '1'. The interrupt can be cleared when a '1' is written to bit [0]. The Event/Bunch-ID Full interrupt can be masked when bit [2] is '0'.

When the Trigger-Type FIFO becomes full then an interrupt can be generated on the IRQ2_n line of each SHARC. The full interrupt status is indicated by bit [1] being a '1'. The interrupt can be cleared when a '1' is written to bit [1]. The Trigger-Type Full interrupt can be masked when bit [3] is '0'.

Note that bit [0] and bit [1] indicate the source of the interrupt (which can be masked). They do not directly indicate the Full status of the Event/Bunch-ID and Trigger-Type FIFOs.

The information of both Event-ID and Trigger-Type FIFOs is combined in one data stream and read-out via a DMA controller to the SHARC. Bit [4] is the Empty status for this data stream.

For debug purposes the Empty Flags of both FIFOs can be read in bit [16] (Event/Bunch-ID FIFO) and bit [17] (Trigger-Type FIFO). Both flags should always be equal since an equal amount of Event-IDs and Trigger-Types should have been received. Note that these Empty flags are different from the DMA controller TTC FIFO Empty signal (bit [4]). The DMA controller can read the FIFOs (making them empty) while the TTC FIFO Empty signal is still asserted since the data should still be read by the SHARC.

Data	Description	Write	Read
0	SHARC Event/Bunch-ID FIFO Full Interrupt	'0': No action '1': Clear Interrupt	'0': not Full (or masked) '1': Full
1	SHARC Trigger-Type FIFO Full Interrupt	'0': No action '1': Clear Interrupt	'0': not Full (or masked) '1': Full
2	SHARC Event/Bunch-ID FIFO Full Interrupt Mask	'0': Disabled '1': Enabled	'0': Disabled '1': Enabled
3	SHARC Trigger-Type FIFO Full Interrupt Mask	'0': Disabled '1': Enabled	'0': Disabled '1': Enabled
4	SHARC TTC FIFO Empty Status	X	'0': not Empty '1': Empty
5	Reserved	X	'0'
6	Generate software controlled Event Counter Reset (ECR) (If Selected)	'0': No action '1': Event Counter Reset	'0'
7	Event Counter Reset (ECR) source selection	'0': ECR_n (Default) '1': Software	'0': ECR_n (Default) '1': Software
8	Event Builder Event/Bunch-ID FIFO Full Interrupt	'0': No action '1': Clear Interrupt	'0': not Full (or masked) '1': Full
9	Event Builder Trigger-Type FIFO Full Interrupt	'0': No action '1': Clear Interrupt	'0': not Full (or masked) '1': Full
10	Event Builder Event/Bunch-ID FIFO Full Interrupt Mask	'0': Disabled '1': Enabled	'0': Disabled '1': Enabled
11	Event Builder Trigger-Type FIFO Full Interrupt Mask	'0': Disabled '1': Enabled	'0': Disabled '1': Enabled

12	ECR Interrupt	‘0’: No action ‘1’: Clear Interrupt	‘0’: No Interrupt ‘1’: ECR Interrupt
13	ECR Interrupt Mask	‘0’: Disabled ‘1’: Enabled	‘0’: Disabled ‘1’: Enabled
14	Reserved	X	‘0’
15	Reserved	X	‘0’
16	SHARC Event/Bunch-ID FIFO Empty Status	X	‘0’: not Empty ‘1’: Empty
17	SHARC Trigger-Type FIFO Empty Status	X	‘0’: not Empty ‘1’: Empty
18	Reserved	X	‘0’
19	Reserved	X	‘0’
20	Event Builder Event/Bunch-ID FIFO Empty Status	X	‘0’: not Empty ‘1’: Empty
21	Event Builder Trigger-Type FIFO Empty Status	X	‘0’: not Empty ‘1’: Empty
31:22	X	X	0

Table 8: TTC Control/Status and Interrupt Register (0x03)

3.1.5.2 Event/Bunch-ID and Trigger-Type FIFOs connected to the Event Builder:

The Event/Bunch-ID and Trigger-Type FIFOs that are connected to the event builder are both 511 words deep. When a Half-Full condition is met in one of the FIFOs then ROD-Busy will be asserted (if the proper mask bit [8] is set in the Channel Enable register: 0x09).

This should prevent the FIFOs from ever becoming full. However, when the Central Trigger Processor (CTP) is not responding to the asserted ROD-Busy, then a full condition can occur. In this case an interrupt can be generated on the IRQ2_n line of each SHARC.

For the Event/Bunch-ID FIFO the full interrupt status is indicated by bit [8] being a ‘1’. The interrupt can be cleared when a ‘1’ is written to bit [8]. The Event/Bunch-ID Full interrupt can be masked when bit [10] is ‘0’.

When the Trigger-Type FIFO becomes full then an interrupt can be generated on the IRQ2_n line of each SHARC. The full interrupt status is indicated by bit [9] being a ‘1’. The interrupt can be cleared when a ‘1’ is written to bit [9]. The Trigger-Type Full interrupt can be masked when bit [11] is ‘0’

Note that bit [8] and bit [9] indicate the source of the interrupt (which can be masked). They do not directly indicate the Full status of the Event/Bunch-ID and Trigger-Type FIFOs.

For debug purposes the Empty Flags of both FIFOs can be read in bit [20] (Event/Bunch-ID FIFO) and bit [21] (Trigger-Type FIFO). Both flags should always be equal since an equal amount of Event-IDs and Trigger-Types should have been received.

The SHARC Event/Bunch-ID and Trigger-Type FIFOs, as well as the Event/Bunch-ID and Trigger-Type FIFOs that are connected to the event builder, are cleared when an Event Counter Reset (ECR) occurs. If bit [7] is ‘0’ (the default value), then this is done directly by

the TTC ECR signal. When bit [7] is set to '1' then the Event/Bunch-ID and Trigger-Type FIFOs can be cleared by the software. This gives the software the opportunity to verify that the FIFOs are empty since the ECR should only occur when the Data Acquisition pipeline is empty. Clearing the FIFOs under software control is done by writing to register 0x03 with bit [6] set to '1'.

When an ECR condition (either TTC or a Software Controlled) occurs then an interrupt (IRQ2_n) is generated if the "ECR Interrupt Mask" bit [13] is set to '1'. The ECR interrupt status is indicated by bit [12] being a '1'. The interrupt can be cleared when a '1' is written to bit [12].

3.1.6 Register 0x04: Resets, LEDs and Status

MS0 offset 0x09

Register 0x04 controls resets and LEDs.

Bit [3:0]: Control the reset signals to the four MROD-Ins. These reset signals (Rst0_n, Rst1_n, Rst2_n and Rst3_n) are active low. The default value for these signals is '1'. If the SHARC on one of the MROD-Ins should be reset then the corresponding bit [3:0] of register 0x04 should be written '0' to assert the reset line. Be sure to write back '1' to the bit to de-assert the reset line again.

Bit [7:4]: Four LEDs are connected to these bits. Writing a '1' puts on the light.

Bit [8]: The S-LINK is reset when this bit is taken from '0' to '1'. This will initiate an S-LINK reset sequence. The URESET_n signal of the S-LINK interface will be asserted. The S-LINK will assert LDOWN_n. URESET_n will be kept asserted until LDOWN_n is de-asserted by the S-LINK. Bit [8] is reads back the value that it was written to.

Bit [9]: Reads the status of the Addressable Scan Port (ASP) that connects the FPGA JTAG chain to the Module Test and Maintenance (MTM) bus [7] on the VME64x backplane. When bit [9] reads '0' then the ASP is connected and the FPGA JTAG chain is controlled by the MTM-bus. This is also signalled by a green front panel LED ("ASP Conn").

Bit [15:10]: Reserved and read back 0x0.

Bit [23:16]: Used to reset the individual RocketIO links.

Note: the status of the 8 RocketIO links can be read in the RocketIO Status register 0x2C.

Data	Description	Write	Read
0	Rst0_n	'0': Reset '1': Normal Mode	'0': Reset '1': Normal Mode
1	Rst1_n		
2	Rst2_n		
3	Rst3_n		
4	LED0 (Red)	'0': LED Off '1': LED On	'0': LED Off '1': LED On
5	LED1 (Red)		
6	LED2 (Green)		
7	LED3 (Green)		
8	S-LINK Output	'0' -> '1': Trigger S-LINK Reset Sequence	Read back last value written

9	ASP Connect	X	‘0’: Connected ‘1’: Disconnected
15:10	Reserved	X	0x00
16	RocketIO 1A Reset	‘1’: Reset	Read back reset status
17	RocketIO 1B Reset	‘1’: Reset	Read back reset status
18	RocketIO 2A Reset	‘1’: Reset	Read back reset status
19	RocketIO 2B Reset	‘1’: Reset	Read back reset status
20	RocketIO 3A Reset	‘1’: Reset	Read back reset status
21	RocketIO 3B Reset	‘1’: Reset	Read back reset status
22	RocketIO 4A Reset	‘1’: Reset	Read back reset status
23	RocketIO 4B Reset	‘1’: Reset	Read back reset status
31:24	Reserved	X	0x00

Table 9: Resets, LEDs and Status Register (0x04)

3.1.7 Registers 0x05: Date & Revision ID Register

MS0 offset 0x0B

Register 0x05 is a read only register. This register contains a value that uniquely identifies the configuration of the FPGA via a date and revision number. The Date ID field is 24 bits wide (bits [31:8]) and is formatted as a year/month/day combination (0xyymmdd). The Revision ID field is 8 bits wide (bits [7:0]).

The value of this register is determined during synthesis of the VHDL code. The operating system date and a revision number are automatically passed to the synthesis tool using a TCL script.

3.1.8 Registers 0x06, 0x07: ID1, ID2 Register

MS0 offset 0x0D, 0x0F

Registers 0x06 and 0x07 reflect the 64 bits that are read-out of an identifier chip (Dallas DS2401 [8]) which contains a 48 bit unique number. The bits in these registers are defined as described in Table 10 and 11.

Data	Description
31:0	ID(31:0)

Table 10: ID1 Register (0x06)

Data	Description
15:0	ID(47:32)
23:16	CRC
31:24	Family ID (0x01)

Table 11: ID2 Register (0x07)

3.1.9 Registers 0x08: FPGA Temperature Register

MS0 offset 0x11

The bits in register 0x08 are connected to a MAX1618 “Remote Temperature Sensor” device [9] with a SMBus Serial Interface. This interface consists of a SMB-Clk (input to the MAX1618) and a SMB-Data line that can be either input or output.

Bit [0]: drives the SMB-Clk line.

Bit [1]: value of the SMB-Data line (read or write)

Bit [2]: determines whether the SMB-Data line is driven by the FPGA or not. A ‘0’ means the FPGA drives the SMB-Data line (with the value of bit [1]), ‘1’ the SMB-Data line is an input to the FPGA and driven by the MAX1618 (the value can be read from bit [1]). For further details, see the MAX1618 datasheet.

Bit [3]: reads the value of the ALERT pin of the MAX1618 device.

3.1.10 Registers 0x09: Channel Enable Register

MS0 offset 0x13

Register 0x09 determines which MROD-In Channels are enabled and take part in the event building.

Bit [7:0]: If ‘1’ then this means that the corresponding CSM input channel is enabled. Note that enabling a channel also means that its Busy signal is taken into account when generating the ROD-Busy signal.

Bit [8]: If ‘1’ then a Half-Full condition on either the Event-ID/Bunch-ID FIFO or the Trigger-Type FIFO that are connected to the event builder will lead to assertion of the ROD-Busy signal. Note that this does not apply to the Event-ID/Bunch-ID and Trigger-Type FIFOs that are connected to the SHARC.

Bit [9]: is a master busy bit. When this bit is set ’1’ then ROD-Busy is asserted independent of any other the Busy flags.

Bit [10]: determines whether event building via the RocketIO transceivers is enabled.

Bit [11]: MROD-X debug mode. When this bit is ‘1’ then read-out is stopped as soon as one of the output streams (S-Link, Spy Event Data FIFO or the Spy Event Length FIFO) becomes full. *During normal operation this bit should be ‘0’* since the spy channel to the SHARC (Event Data FIFO and Event Length FIFO) should *never stop the main data stream.*

Bit [12]: determines whether event building via the RocketIO transceivers is working without an active TTC system that sends Event-ID, Bunch-ID and Trigger-Type for each trigger. The event builder waits until these are available before gathering the data and sending the full event fragment to the S-Link. In RocketIO test mode the event builder simply gathers the incoming data, builds the event fragment and the header is filled with a faked Event-ID whereas the Bunch-ID and Trigger-Type are filled with all zeroes.

Bit [13]: Empty flag of the Spy Event Data FIFO. This FIFO can be read by the SHARC under DMA control (using DMAR2, see description of MS2 address space). The FIFO is only provided for monitoring and test purposes. The FIFO will only collect the events selected by the spy pre-scale (register 0x2F).

Data	Description
0	Enable Channel 1A
1	Enable Channel 1B
2	Enable Channel 2A
3	Enable Channel 2B
4	Enable Channel 3A
5	Enable Channel 3B
6	Enable Channel 4A
7	Enable Channel 4B
8	Mask TTC Interface Busy
9	Assert ROD-Busy
10	Enable Event Building via RocketIO
11	MROD-X Debug Mode
12	Enable RocketIO test mode
13	Spy Event Date FIFO Empty flag
14	Spy Event Length FIFO Empty flag
15	Reserved
16	Busy Status Channel 1A
17	Busy Status Channel 1B
18	Busy Status Channel 2A
19	Busy Status Channel 2B
20	Busy Status Channel 3A
21	Busy Status Channel 3B
22	Busy Status Channel 4A
23	Busy Status Channel 4B
24	Busy Status TTC Interface
31:25	Reserved

Table 12: Channel Enable Register (0x09)

Bit [14]: Empty flag of the Spy Event Length FIFO. This FIFO stores the event lengths of the events that were selected for spying. See register 0x2E for more details.

Bit [23:16]: Busy status of all the input channels.

Bit [24]: Busy status of the TTC interface.

The ROD Busy signal is available as a NIM signal at the front. A ROD Busy LED lights up if this signal is active. The ROD Busy is send to the Central trigger Processor (CTP) via the Timing Interface Module (TIM).

3.1.11 Registers 0x0A: Format Version Register

MS0 offset 0x15

This register contains the value of the "Format Version" that should be written in the output data stream.

3.1.12 Registers 0x0B: Module ID Register

MS0 offset 0x17

This register contains the value of the "Module ID" that should be written in the output data stream.

3.1.13 Registers 0x0C: Run Number Register

MS0 offset 0x19

This register contains the value of the "Run Number" that should be written in the output data stream.

3.1.14 Registers 0x0D: Detector Event Type Register

MS0 offset 0x1B

This register contains the value of the "Detector Event Type" that should be written in the output data stream.

3.1.15 Register 0x0E to 0x11: BOT (TDC-Header) Pattern & Mask

MS0 offset 0x1D to 0x23

In order to be able to recognize a BOT word in the data stream, the 32 bit pattern and mask Registers 0x0E to 0x11 should be set.

Register 0x0E contains the 32 bit pattern that must match the BOT pattern in the data stream. A 32 bit mask (register 0x10) defines the bits that must match. If a mask bit is set to '1' the corresponding bit in the pattern register must match the bit in the data stream. If all bits match then a BOT is signalled.

Reading back registers 0x0E or 0x10 yields the value written into them.

Registers 0x0F and 0x11 are reserved.

3.1.16 Register 0x12 to 0x15: EOT (TDC-Trailer) Pattern & Mask

MS0 offset 0x25 to 0x2B

The description of these registers is the same as for the BOT Pattern and Mask registers 0x0E to 0x11, except that registers 0x12 to 0x15 control the recognition of an EOT.

3.1.17 Register 0x16 to 0x19: BOEF (Begin Of Event Fragment) Pattern & Mask

MS0 offset 0x2D to 0x33

The description of these registers is the same as for the BOT Pattern and Mask registers 0x0E to 0x11, except that registers 0x16 to 0x19 control the recognition of a BOEF.

3.1.18 Register 0x1A to 0x1D: LWC (Link Word Count) Pattern & Mask

MS0 offset 0x35 to 0x3B

The description of these registers is the same as for the BOT Pattern and Mask registers 0x0E to 0x11, except that registers 0x1A to 0x1D control the recognition of an LWC.

3.1.19 Register 0x1E to 0x25: Reserved

MS0 offset 0x3D to 0x4B

Registers 0x1E to 0x25 are reserved.

3.1.20 Register 0x26 to 0x29: TWC (Trailer Word Count) Pattern & Mask

MS0 offset 0x4D to 0x53

The description of these registers is the same as for the BOT Pattern and Mask registers 0x0E to 0x11, except that registers 0x26 to 0x29 control the recognition of a TWC.

3.1.21 Registers 0x2A: Error-Code Replace Patterns

MS0 offset 0x55

The MROD-In puts error replacement codes in the data stream whenever there is an error signalled on the CSM input link (see the MROD-In Programmers Manual for details).

In order to be able to recognize these codes in the data stream, they must be defined in the Error-Code Replace Patterns Register.

Bit [31:28] TDC Parity Error, Error-Code Replace bits

Bit [27:16] Not used, writing doesn't care, reading yields 0x000

Bit [15:12] Input Link Parity Check Failure Error-Code Replace bits

Bit [11:0] Not used, writing doesn't care, reading yields 0x000

3.1.22 Registers 0x2B: Extended Event-ID Register

MS0 offset 0x57

This register is actually a counter that can be preloaded with an 8 bit value. The counter is incremented each time an Event Counter Reset (ECR) is received from the TTC system. When register 0x2B is read then the actual value of the counter is read back.

The Extended Event-ID bits serve as an 8 bit extension for the 24 bit normal Event-ID bits. Therefore the Extended Event-ID register is connected to bits [31:24] of the data bus. Bits [23:0] read the Event-ID value that is last received from the TTC system. Note that if the register is read, immediately after an Event Counter Reset then Bit [23:0] of the register value will probably be unequal to zero since these bits still hold the last received Event-ID which is the one send just before the Event Counter Reset.

3.1.23 Registers 0x2C: RocketIO Status

MS0 offset 0x59

This register is read only and reads the status of the 8 RocketIO transceivers that are connected to the 8 MROD-Ins.

Bits [7:0] read the Link Down Status of the input channels; ‘0’ => Link is Down, ‘1’ => Link is Up.

Bits [15:8] read the Synchronisation Status of the input channels; ‘0’ => Receiver is *Out of* Synchronisation with the transmitter on the MROD-In, ‘1’ => Receiver is *In* Synchronisation with the transmitter on the MROD-In.

Bits [23:16] read the Link Error status; ‘0’ => Okay, ‘1’ => Link Error occurred.

Bits [31:24] read the Link Full status; ‘0’ => Not Full, ‘1’ => Link transmit FIFOs (from MROD-Out to MROD-In) are Full.

Data	Description
7:0	LDown Status Channel [4B,4A,3B,3A,2B,2A,1B,1A]
15:8	RxInSync Status Channel [4B,4A,3B,3A,2B,2A,1B,1A]
23:16	Link Error Status Channel [4B,4A,3B,3A,2B,2A,1B,1A]
31:24	Link Full Status Channel [4B,4A,3B,3A,2B,2A,1B,1A]

Table 13: RocketIO Status Register (0x2C)

3.1.24 Registers 0x2D: RocketIO Link Down, Spy Channel Full Interrupt (IRQ1)

MS0 offset 0x5B

Whenever one of the 8 RocketIO links between the MROD-In FPGAs and the MROD-Out FPGA is going down, or when the spy channel (Event Data FIFO or Event Length FIFO) becomes full, an interrupt can be generated on the SHARC (A and/or B) via the IRQ1_n interrupt signal.

Bit [7:0]: Give information about which RocketIO [4B,4A,3B,3A,2B,2A,1B,1A] caused the interrupt. Writing a ‘1’ to one of the bits [7:0] clears the corresponding interrupt.

Bit [15:8]: Interrupt mask bits. A link down interrupt will only occur when the corresponding link goes down and the interrupt is enabled (mask bit is set to '1').

Bit [16]: Is '1' when a Spy Event Length FIFO Full condition occurred, causing an IRQ1. The interrupt can be cleared by writing a '1' to this bit. Bit [18] is the corresponding mask bit. Note that when the Spy Event Length FIFO is full, no more Spy Events will be sent to the Spy Event Data FIFO.

Bit [17]: Is '1' when a Spy Event Data FIFO Full condition occurred, causing an IRQ1. The interrupt can be cleared by writing a '1' to this bit. Bit [19] is the corresponding mask bit. Normally this condition will never occur since the SHARC should read the Spy Event Data FIFO using an endless DMA. However the FIFO is read-out by the SHARC with a maximum bandwidth of 80 MByte/sec. This means that the FIFO needs to be deep in order not to fill when a large event is spied (which can be burst into the FIFO with 160 Mbytes/sec). Therefore the Spy Event Data FIFO is 16K words deep.

Table 14 lists the corresponding RocketIO channel and register bits.

IRQ Bit	Mask Bit	Interrupt Source
0	8	LDown Channel 1A
1	9	LDown Channel 1B
2	10	LDown Channel 2A
3	11	LDown Channel 2B
4	12	LDown Channel 3A
5	13	LDown Channel 3B
6	14	LDown Channel 4A
7	15	LDown Channel 4B
16	18	Spy Event Length FIFO Full
17	19	Spy Event Data FIFO Full

Table 14: RocketIO Link Down, Spy Channel Full Interrupt Register (0x2D).

3.1.25 Registers 0x2E: Spy Event Length FIFO

MS0 offset 0x5D

Register 0x2E is a read only register. When the register is read, one word is read from the Spy Event Length FIFO.

The SHARC can determine whether this FIFO is empty by reading bit [14] of the Channel Enable Register (0x09).

The spy event length is sent to the Spy Event Length FIFO during the transfer of the last word of the spy event to the Spy Event Data FIFO. When the Spy Event Length FIFO becomes full then the transfer of spy events to the Spy Event Data FIFO is stopped.

The SHARC is reading the Spy Event Data FIFO using an endless DMA thus filling up its internal buffers. The only way to stop the transfer of spy event data is when the SHARC stops reading the Spy Event Length FIFO so that this FIFO becomes full (after 4 Spy Event Length entries). Spy event data is now no longer selected for transfer to the SHARC.

Bit [31:20]: Event-ID bits

Bit [19:0]: Contain the word count of the spy event.

3.1.26 Registers 0x2F: Spy Event Pre-scale

MS0 offset 0x5F

Bits [15:0] of register 0x2F determine how often a complete event is passed to the SHARC processors. When bit [16] is cleared (‘0’) then spy mode is disabled. The default value for register 0x2F is 0x0 (spying disabled).

Note that (when bit [16] = ‘1’) the first event of a run is always outputted. Note also, that loading value 0x10000 in the Spy Event Pre-scale Register means that all events are passed to the SHARC processors (i.e. all events are spied).

3.2 MS1 address space of the SHARC: TTC information

The Timing Trigger and Control (TTC) [5] information is distributed via the TIM [6] over the P3 backplane through the VME64x carte on 8 backplane lines TTC [7:0] (see Table 15). The MROD-Out FPGA receives the serial bit streams of signals TTC4_n and TTC5_n. The data values within Serial-ID and Serial-TT are sent with the least significant bit first. The TTC signals are low active so the start bit has a value of ‘0’ (high), which is guaranteed to be preceded by a ‘1’ (low).

Signal	Description	MROD-Out Action
TTC0_n	L1-Accept	Not used
TTC1_n	Event Counter Reset	Clears Event/Bunch-ID and Trigger-Type FIFOs (see also Register 0x03)
TTC2_n	Bunch Counter Reset	Not used
TTC3_n	Issue Calibration Pulse	Not used
TTC4_n	Serial-ID: EV-ID (24 Bits) and BC-ID (12 Bits)	Write Event/Bunch-ID FIFOs
TTC5_n	Serial-TT: Trigger-Type (8 bits) and Reserved (2 bits)	Write Trigger-Type FIFOs
TTC6_n	Reserved	Not used
TTC7_n	Spare	Not used

Table 15: TTC-Bus signal definition

The Event-ID and Bunch-ID from TTC4_n, the Trigger-Type from TTC5_n and the Extended Event ID Count (stored in register 0x2B) are sent to two sets of FIFOs. One set is coupled to the event builder; the other set is read-out by the SHARC.

Read out by the SHARC is controlled via the DMAR1_n signal. A read cycle on any address in MS1_n is a valid read cycle for reading out the TTC information.

For each TTC entry in the SHARC TTC FIFO the DMAR1_n signal will be asserted twice since the complete TTC information is packed into two 32 bit words

After the first DMAR1_n signal, the SHARC can read-out the Extended Event-Id and the 24 bit Event-Id as it was supplied by the TTC system. Immediately after this read cycle, a second read cycle can be started to read-out the Trigger-Type and Bunch-ID (See Table 16).

Word	Bit [31:28]	[27:24]	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
0	Extended Event-ID Register 0x2B		Event-ID					
1	Reserved (0x000)			Trigger-Type		Bunch-ID		

Table 16: Content of a group of 2 words as supplied to the SHARC

The empty status of the Event/Bunch-ID FIFO and the Trigger-Type FIFO is available when reading the TTC Control/Status and Interrupt Register (see register 0x03). Normally it shouldn't be necessary to read the empty status since no DMA request is send when the FIFOs are empty.

The Event/Bunch-ID FIFO and the Trigger-Type FIFO are read only. Writing to address space MS1 has no effect.

When the TTC system asserts the Event Counter Reset (ECR corresponds to TTC1_n) then the content of both the Event/Bunch-ID FIFOs and the Trigger-Type FIFOs are cleared if the Event Counter Reset source selection (bit [7] in register 0x03) is set to its default value ('0').

3.3 MS2 address space of the SHARC: Read Spy Event Data

It is possible to spy on the event data that is send to the Read Out Link. Once every so often a complete event can be sent to a FIFO that is connected to the SHARC processors. Read out by the SHARC is controlled via the DMAR2_n signal. A read cycle on any address in MS2_n is a valid read cycle for reading out the Spy Event Data FIFO.

The amount of events that are sent to the SHARC is determined by the Spy Event Pre-scale register (0x2F, see registers in MS0_n address space).

3.4 MS3 address space of the SHARC: S-LINK output

When the "Enable Event Building via RocketIO" (see bit [10] in the Channel Enable Register, 0x09) is disabled then the SHARC processors can send data to the output S-LINK by writing data to address space MS3. The S_LINK output interface acknowledges each write cycle. When there is an S-LINK output FIFO full condition then the SHARC write cycles to MS3 address space will no longer be acknowledged.

The purpose of the S-LINK output FIFO is to avoid an external bus stall of the SHARC processors when a write cycle to the S-LINK is not Acknowledged because of an S-LINK Full (LFF_n) or a S-LINK Down (LDOWN_n) condition. This stall situation would extend as long as either LFF_n or LDOWN_n is asserted. A FIFO of 1024 words is placed in the S-LINK Interface of the FPGA in between the SHARC processors and the S-LINK. SHARC A or B can now verify (by reading Flag 1, see SHARC Flags) that it can write at least half the FIFO depth (512 words) without the danger of a stall situation. Note that the FIFO is in fact 1025 words deep because of a pipeline register in front of the 1024 word FIFO.

Data in the FIFO is send continuously (at 160 MB/sec) to the S-LINK output provided that the S-LINK is not down and the link is not full (LDOWN_n and LFF_n both de-asserted). Therefore the FIFO will normally be (nearly) empty since the maximum data transfer rate of the SHARC processors to the FIFO in the MROD-Out FPGA does not exceed 160MB/sec.

For all addresses in the MS3 address space which have A[1:0] = "01" (0x01, 0x05, 0x09 etc.), the SHARC writes S-LINK *DATA* words (S-LINK control line LCTRL_n = '1').

For all addresses in the MS3 address space which have A[1:0] = "11" (0x03, 0x07, 0x0B etc.), the SHARC writes S-LINK *CONTROL* words (S-LINK control line LCTRL_n = '0').

Data	Description	Write	Read
31:0	S-LINK LD(31:0)	Write data to S-LINK	X

Table 17: SHARC to S-LINK data format

Remark: The S-LINK cannot be read in memory region MS3. To read the LRL [3:0] lines of the S-LINK, see the S-Link Status and Interrupt register (0x02) in MS0 address space.

3.5 SHARC Flags:

Table 18 gives an overview of the SHARC flags.

Flag Number	Direction	Description	LED
0 (SHARC A)	Output		Red
1 (SHARC A)	Input	S-LINK Output FIFO Half Full	Red
2 (SHARC A)	Output	S-LINK UTEST_n	Green
3 (SHARC A)	Output		Green
0 (SHARC B)	Output		Red
1 (SHARC B)	Input	S-LINK Output FIFO Half Full	Red
2 (SHARC B)	Output		Green
3 (SHARC B)	Output		Green

Table 18: SHARC Flags

Flag 1A and Flag 1B should be configured as an Input. Flag 1A and 1B signal the status of the S-LINK output FIFO in the MROD-Out FPGA.

All other flags should be configured as output. They drive surface mount LEDs (on the board, not on the front panel of the module) for general purpose signalling. Setting a flags to '1' will light up the LED.

Note that the SHARC Flag pins are configured as Inputs during and after a Reset.

3.6 SHARC Interrupts:

Table 19 gives an overview of the SHARC interrupt lines. The interrupt signals for SHARC A are the same as for SHARC B.

IRQ0_n has three interrupt sources, which can be enabled (see S-LINK Status and Interrupt Register):

- LRL lines of the S-LINK change.
- S-LINK signal LDOWN_n is asserted
- S-Link output FIFO becomes half full.

IRQ1_n is used to signal a RocketIO link down condition (see also register 0x2D)

IRQ2_n is asserted when either the Event/Bunch-ID FIFOs or the Trigger-Type FIFOs are full (see TTC Control/Status and Interrupt Register 0x03).

IRQ Number	Description
0A	S-LINK
1A	RocketIO Link Down
2A	Event/Bunch-ID or Trigger-Type FIFOs Full
0B	S-LINK
1B	RocketIO Link Down
2B	Event/Bunch-ID or Trigger-Type FIFOs Full

Table 19: SHARC Interrupts

4 References

1 RocketIO Transceiver User Guide: http://direct.xilinx.com/bvdocs/userguides/ug024.pdf

2 ADSP-21160 SHARC DSP Hardware Reference: http://www.analog.com/Processors/Processors/sharc/technicalLibrary/manuals/32BitIndex.html

3 American National Standard for VME64; ANSI/VITA 1-1994

4 American National Standard for VME64 Extensions; ANSI/VITA 1.1-1997

5 http://ttc.web.cern.ch/TTC/intro.html

6 http://www.hep.ucl.ac.uk/atlas/sct/tim/

7 IEEE 1149.5 MTM bus

8 DS2401 datasheet, http://www.maxim-ic.com/DS2401

9 MAX1618 datasheet, http://www.maxim-ic.com/MAX1618

This document has been prepared using the Test Report Document Template version 0.2 provided and approved by the ATLAS TDAQ and DCS Connect Forum. For more information, go to

http://atlas-connect-forum.web.cern.ch/Atlas-connect-forum/.

This template is based on the SDLT Single File Template that has been prepared by the IPT Group (Information, Process and Technology), IT Division, CERN (The European Laboratory for Particle Physics) and then converted to MS Word. For more information, go to http://framemaker.cern.ch/.