ATLAS Muon MDT

MROD Module:

MROD-X-In Programmer's Manual

Document Version:	1
Document Issue:	0
Document ID:	ATLAS-TDAQ-2005-XXX or EDMS Id XXXXXX
Document Date:	26-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-In 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-In Programmers Manual		
ID:	ATLAS-TDAQ-2005-XXX or EDMS Id XXXXXX		
Version	**Issue**	**Date**	**Comment**
1	0	26-Oct-2005	First version (draft)

Table of Contents

1 Introduction

The MROD-In is the input part of the MDT Precision Chamber Read-Out Driver. It is designed to receive data from two channels of 18 TDCs each. These two data streams are processed in two FPGAs; one per channel. The FPGAs are connected to one SHARC processor. Each MROD-In FPGA is connected to the MROD-Out FPGA via a RocketIO link [1]. The MROD-In SHARC processor communicates with the MROD-Out SHARC processors via SHARC Links (figure 1).

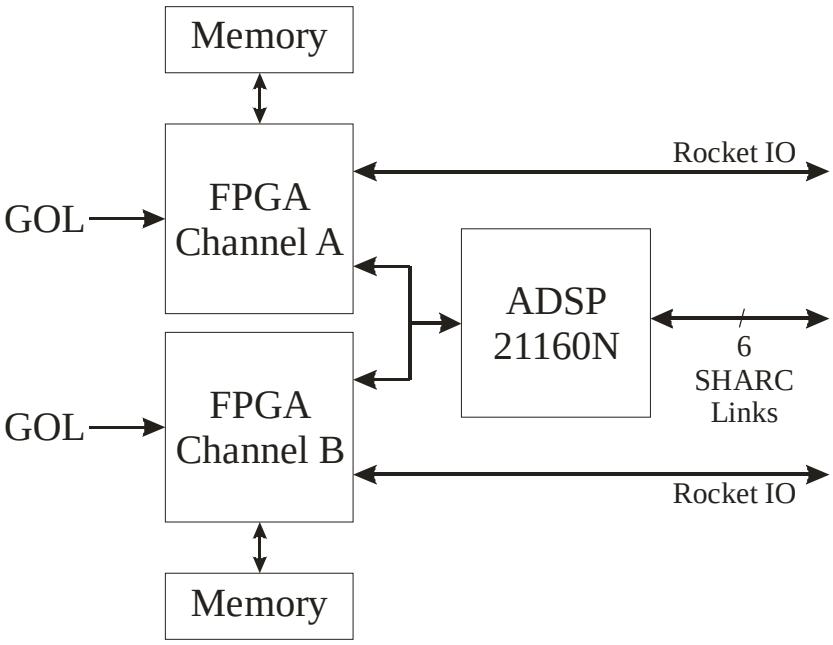

Figure 1: MROD-In block diagram.

The description below will focus on one channel and one FPGA since the processing of each channel is equal.

The input data for one channel of the MROD-In will be organised as can be seen in figure 2. Such data will be produced by the Chamber Service Module (CSM) and sent over a Gigabit Optical Link (GOL) to the MROD-In.

Separator	TDC 0	TDC 1	…	TDC 17	Separator	TDC 0	…

Figure 2: Input data format for one MROD-In channel.

A Separator word is used to mark the start of 18 time slots, one for each TDC. When there is no data available from a TDC then a NoData word is put in the data stream in the corresponding time slot.

The MROD-In channel can be programmed to recognise several words in the input data stream. These words are:

- A Separator word

- A TDC-Header word (= Begin Of TDC: BOT [2])

- A NoData word

- A TDC-Trailer word (= End Of TDC: EOT [2])

To do so, the FPGA on the MROD-In contains comparators that can be programmed with a bit pattern and a bit mask.

Separator:

When a Separator is recognised a time slot counter is started which will count from 0 up until 17. While counting, data from the input link is transferred to 18 partitions (each 8K words) of the buffer memory. Partition 0 is corresponding with time slot 0, partition 1 with time slot 1 etc. The only case where no data will be transferred to the buffer memory is when a NoData word is received. If there are more than 18 words sent after a Separator is recognised then the surplus of data will be discarded.

TDC-Header (BOT):

When a TDC-Header is recognised then bits [28:24] will be replaced by the value of the time slot counter (5 bits). In this way, data of the 18 TDCs can be uniquely identified. Note that the TDC-Header bits [27:24] normally contain the 4 bit TDC-ID as programmed by JTAG into the TDCs.

NoData word:

When a NoData word is sent by the CSM and recognised by the MROD-In NoData word comparator then this means that there was not yet data available from the TDC to be sent by the CSM in the corresponding time-slot. Nothing is stored in the corresponding partition.

TDC-Trailer (EOT):

When a TDC-Trailer is recognised then it is checked whether the Event-ID of the trailer falls within a 16 wide window of ‘Expected’ Event-IDs. This can lead to an ‘Accept’, an ‘Early’, or a ‘Late’ condition. The Event-ID in the TDC-Trailer is 12 bits wide, which means that the Event-ID can have 4096 different values. So, for 2048 values further than the Expected Event-ID + 8 (which is halfway the ‘Accept’ window) there is a roll over from ‘Early’ to ‘Late’ (see Table 1). Note that [Expected Event-ID + 7 – 2048] is the same value as [Expected Event-ID + 7 + 2048].

‘Early’ and ‘Late’ are error conditions, which can generate an interrupt on the SHARC. When the TDC-Trailer is accepted, a flag bit is set in the ‘Tetris register’. This register operates similar to the famous computer game called ‘Tetris’, hence the name ‘Tetris register’. The row of the flag bit in the Tetris register is determined by the low order 4 bits of the TDC-Trailer Event-ID. The column of the flag bit in the Tetris register is determined by the TDC time slot [0:17]. When all Trailers with the Expected Event-ID of enabled TDCs are received then a ‘Row-Out’ condition is sent to the output controller. So a row is complete when all enabled TDCs (see the TDC-Mask register) flagged the presence of their TDC-Trailer with the Expected Event-ID in the buffer memory.

A Row-Out condition will also occur when a row which is flagging the first, up to fourteenth following Event-ID is complete (Expected Event-ID + [1:14]). In this case the Row-Out condition for the Expected Event-ID is waiting for one or more TDC-Trailers with an Event-ID which was lost. The status of the flags of the Expected Event-ID row in the Tetris register tells us, which TDC-Trailer(s) was/were lost.

Expected Event-ID + 7 – 2048	Early
Expected Event-ID + 8 – 2048	Late
:	:
Expected Event-ID – 1	Late
Expected Event-ID	Accept
:	:
Expected Event-ID + 15	Accept
Expected Event-ID + 16	Early
:	:
Expected Event-ID + 7 + 2048	Early
Expected Event-ID + 8 + 2048	Late

Table 1: Expected Event-ID window

Furthermore a Row-Out condition immediately occurs whenever a TDC-Trailer is received with its Event-ID equal to Expected Event-ID + 15. This will only occur if the Event-ID difference between TDCs is more than 15, which is a very rare condition. Note that in this case the last row in the Tetris register is indexed so this Row-Out condition is an attempt to free up rows in the Tetris register in order to keep track of incoming TDC data.

A Row-Out condition stores the status from the Expected Event-ID row of the Tetris register in an 'input to output FIFO' (I2O_FIFO). The Expected Event-ID is also stored.

Note that the input link data is not restricted to 18 TDC number slots. It could be less since the TDC number counter is forced to 0 after each received Separator word. However the frequency of the Separator words is limited due to the way the Tetris register is implemented. Separator words should at least be 3 clock cycles apart.

TDCs that are not in use or which have a malfunction can be disabled using the TDC-Mask register. When a TDC is disabled then the Tetris register will not wait for the corresponding trailer, which might never come.

The output controller reads the I2O_FIFO and gathers the data from the partitions in the buffer memory. These data are sent, either with or without Zero Suppression, to the output data stream. From here the output data stream is transferred into two parallel streams, one is put into the Event Data FIFO that is connected to the RocketIO link; the other may be put into the Event Data FIFO that is connected to the SHARC processor for spying purposes.

The number of data words within a single event is sent to an Event Length FIFO. As with the Event Data FIFOs described above, there are also two Event Length FIFOs operating in parallel, one connected to the RocketIO link, the other connected to the SHARC processor.

The SHARC processor can inspect the empty status of the Event Length FIFO that is connected to it, by means of one of its flag pins. The data stream that is sent to the SHARC processor is stored in a FIFO which is read by the SHARC under DMA control with a maximum bandwidth of 80 MB/sec. Note that the SHARC host bus is used by two MROD-In FPGAs (channel A and B) so the bandwidth of each FPGA is 40 MB/sec when the bandwidth is equally distributed over these two FPGAs. The Event Data FIFO DMA controller of channel A and B are connected to DMAR1_n and DMAR2_n of the SHARC respectively. The empty status of the Event Length FIFO of channel A and B are connected to flag 2 and 3 of the SHARC respectively. See also Table 3 and 7.

Data	Remark
0x8F000000	Begin Of Event Fragment (BOEF) This word is only used for communication between MROD-In and MROD-Out
LWC Pattern	Link Word Count (LWC)
BOL Pattern	Begin Of Link (BOL)
MROD-In Header (TLP)	Bits [31:24] MROD Header Pattern (Register 0x14) Bits [17:0] point out which TDCs are present in the data
TDC0 Header	If TDC0 was present
TDC0 Data	If TDC0 was present and had data
:	
TDC0 Trailer	If TDC0 was present
TDC1 Header	If TDC1 was present
TDC1 Data	If TDC1 was present and had data
:	
TDC1 Trailer	If TDC1 was present
:	
:	
MROD-In Trailer (TWC)	Bits [31:24] MROD Trailer Pattern (Register 0x15) Bits [23:12] contain the Event-ID Bits [11:0] contain the Event Word Count
MROD Header	Next event

Table 2: Output data format

The transfer of event data starts with a Begin Of Event Fragment (BOEF) word, followed by a Link Word Count (LWC) and Begin Of Link (BOL) data word [2]. Table 2 lists the format of these data.

A word that is read from the I2O_FIFO contains the status of the Tetris register and the Expected Event-ID. This information is put into the TDC Link Present (TLP) word which is transferred next. If a bit in the Tetris register was set then the corresponding partition in the buffer memory should be read until a TDC-Trailer is found with an event number equal to the Expected Event-ID in the I2O_FIFO status word. It is guaranteed that the output controller will find such a TDC-Trailer in the buffer memory since the Tetris register has the corresponding flag set.

There can be two conditions where the output controller will switch off the read-out for a certain partition. The first condition is when a partition in the buffer memory is full. Since the partitions are in fact circular buffers, the data in the partition will be overwritten. This means there is no unambiguous relation anymore between the content of the Tetris register and the data in the partition. The output controller could get confused because it is no longer guaranteed that it will find the TDC-Trailer with the Expected Event-ID.

The second condition where the output controller will switch off the read-out is when a single event is longer than a maximum event size. Read-out of the partition is shut down to prevent the data from that partition of blocking the output bandwidth.

Each channel of the MROD-In can be put in “Test Mode”. Some Control and Status registers can feed data to the link input. This test mode is internal to the MROD-In FPGA. Each MROD-In FPGA also has a “GOL Test Mode”. Data can be generated and sent out via the transmitter of the CSM link. When a loop back fibre is plugged in, these data can be used as input to the CSM link.

It is also possible for the SHARC to access the buffer Memory although this is only possible offline since the cycle-shared ports of the buffer memory are dedicated to the link input and the output controller of the FPGA during normal operation.

Several interrupt sources signal special conditions or errors.

All of the above can be controlled and monitored by means of a set of Control and Status Registers, which reside in the MS0 address space of the SHARC. Table 5 contains a listing these registers.

2 SHARC Memory Regions

2.1 General

SHARC Address line A[21] determines the MROD-In channel, which is being accessed as can be seen in Table 3. This table also lists the SHARC DMA request lines and the Event Length FIFO empty status for each channel.

MROD-In Channel	A[21]	Event Data FIFO DMA Channel	Event Length FIFO empty status (see also table 7)
A	'0'	DMAR1_n	Flag 2
B	'1'	DMAR2_n	Flag 3

Table 3: Access to the MROD-In input Channel A and B

The memory regions for the FPGAs are organized as can be seen in Table 4. Note that since address line A[21] of the SHARC is used to address the FPGA of channel A or B, memory regions MS0, MS1 and MS2 should be at least 4 MWord long (2 MWord for each channel).

The SHARC uses a 64 bit data bus to interface to external memory. The data bus of each 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" [3]).

Memory Region	Function	EBxWS	EBxAM	Remarks
MS0	Internal registers	100	00	1, 2
MS1	Read Output FIFO	001	00	1, 3
MS2	Buffer memory access when in SHARC Read/Write Mode	011	00	1, 4
MS3	Not Used	N.A.	N.A.	

Table 4: 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" [3].

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 = 011 means 3 Wait, 1 Hold Cycles
EBxAM = 00 Both Internal and External Acknowledge required

2.2 SHARC MS0 address space: Control and Status Registers

2.2.1 Registers 0x00 to 0x03: Separator Pattern & Mask

MS0 offset 0x01 to 0x07

Register 0x00 contains the 32 bit pattern that must match the Separator pattern on the input link. A 32 bit mask (register 0x02) defines the bits that must match. If a mask bit is set to '1' the corresponding bit in the pattern register must match the Separator pattern on the input link. If all bits match then a Separator is signalled.

Reading back registers 0x00 or 0x02 yields the value written into them.

Registers 0x01 and 0x03 are reserved.

2.2.2 Registers 0x04 to 0x07: BOT (TDC-Header) Pattern & Mask

MS0 offset 0x09 to 0x0F

The description of these registers is the same as for the Separator Pattern and Mask registers 0x00 to 0x03, except that registers 0x04 to 0x07 control the recognition of a BOT condition.

2.2.3 Registers 0x08 to 0x0B: EOT (TDC-Trailer) Pattern & Mask

MS0 offset 0x11 to 0x17

The description of these registers is the same as for the Separator Pattern and Mask registers 0x00 to 0x03, except that registers 0x08 to 0x0B control the recognition of an EOT condition.

2.2.4 Registers 0x0C to 0x0F: NoData Pattern & Mask

MS0 offset 0x19 to 0x1F

The description of these registers is the same as for the Separator Pattern and Mask registers 0x00 to 0x03, except that registers 0x0C to 0x0F control the recognition of a 'NoData' word condition.

Register	MS0 offset	Function	Write	Read	Default After Reset
0x00	0x01	Separator Pattern	0xnnnnnnnn	0xnnnnnnnn	0xD0000000
0x01	0x03	Reserved	0x--------	0x00000000	0x00000000
0x02	0x05	Separator Mask	0xnnnnnnnn	0xnnnnnnnn	0xF0000000
0x03	0x07	Reserved	0x-------0	0x00000000	0x00000000
0x04	0x09	BOT (TDC-Header) Pattern	0xnnnnnnnn	0xnnnnnnnn	0xA0000000
0x05	0x0B	Reserved	0x--------	0x00000000	0x00000000
0x06	0x0D	BOT Mask	0xnnnnnnnn	0xnnnnnnnn	0xE0000000
0x07	0x0F	Reserved	0x-------0	0x00000000	0x00000000
0x08	0x11	EOT (TDC-Trailer) Pattern	0xnnnnnnnn	0xnnnnnnnn	0xC0000000
0x09	0x13	Reserved	0x--------	0x00000000	0x00000000
0x0A	0x15	EOT Mask	0xnnnnnnnn	0xnnnnnnnn	0xF0000000
0x0B	0x17	Reserved	0x-------0	0x00000000	0x00000000
0x0C	0x19	No Data Pattern	0xnnnnnnnn	0xnnnnnnnn	0x00000000
0x0D	0x1B	Reserved	0x--------	0x00000000	0x00000000
0x0E	0x1D	No Data Mask	0xnnnnnnnn	0xnnnnnnnn	0xF0000000
0x0F	0x1F	Reserved	0x-------0	0x00000000	0x00000000
0x10	0x21	Error-Code Replace Patterns	0xn---n---	0xn000n000	0x5000D000
0x11	0x23	Reserved	0x--------	0x00000000	0x00000000
0x12	0x25	LWC (Link Word Count) Pattern	0xnnnnnnnn	0xnnnnnnnn	0x81000000
0x13	0x27	BOL (Begin Of Link) Pattern	0xnnnnnnnn	0xnnnnnnnn	0x18000000
0x14	0x29	TLP (TDC Link Present; MROD-In Header) Pattern	0xnn------	0xnn000000	0x89000000
0x15	0x2B	TWC (Trailer Word Count; MROD-In Trailer) Pattern	0xnn------	0xnn000000	0x8A000000
0x16	0x2D	Event Length FIFO	0x--------	0x0nnn0nnn	
0x17	0x2F	Interrupt Control IRQ0	0xn-----nn	0xnsnnnnnn	0x00000000
0x18	0x31	TDC (Parity) Error Individual Interrupt Mask	0x---qnnnn	0x000qnnnn	0x0003FFFF
0x19	0x33	Input link Interrupt; IRQ1	0x------qs	0x000000qs	0x00000000 1)
0x1A	0x35	Early and Late Event-ID; IRQ2	0x--------	0x000nnnnn	0x00000000
0x1B	0x37	Test & Control Status Register	0x-s--nvsn	0x0snnnnnn	0x00060940 2)
0x1C	0x39	Test Link Data Register	0xnnnnnnnn	0xnnnnnnnn	0x00000000
0x1D	0x3B	Test Link Control Register	0x-------q	0x0000000q	0x00000003
0x1E	0x3D	Maximum Event Size	0x-----nnn	0x00000nnn	0x00000400
0x1F	0x3F	Expected Event-ID	0x-----nnn	0x00000nnn	0x00000000
0x20	0x41	TDC Mask Register	0x---qnnnn	0x000qnnnn	0x0003FFFF
0x21	0x43	Partition Read-out Enable	0x---qnnnn	0x000qnnnn	0x0003FFFF
0x22	0x45	Separator Flags Register	0x--------	0xnnnnnnnn	0x00000000

0x23	0x47	Date & Revision ID Register	0x--------	0xyymmddxy	0xyymmddxy 3)
0x24	0x49	FPGA Temperature Register	0x-------q	0x0000000q	0x00000008 4)
0x25	0x4B	Zero Suppress Override Count	0x---pnnnn	0x000pnnnn	0x00010400
0x26	0x4D	Channel Busy Mask Register	0x---snnnn	0x000snnnn	0x0007FFFF
0x27	0x4F	Channel Busy Status Register	0x--------	0x000snnnn	0x00000000
0x28	0x51	TDC Limit Register	0x-----nnn	0x00000nnn	0x00000060
0x29	0x53	SHARC Spy Count Register	0x---pnnnn	0x000pnnnn	0x00010000
0x2A	0x55	GOL Test FIFO	0xnnnnnnnn	0x00000000	0x00000000
0x2B - 0x3F	0x57 - 0x7F	Reserved	0x--------	0x--------	0x--------

Remarks:

‘n’ is any hexadecimal number ‘s’ is one out of 0x00 to 0x07

‘-’ is don’t care ‘t’ is 0x00 or 0x08

‘p’ is either 0x00 or 0x01 ‘u’ is 0x00 or 0x02

‘q’ is 0x00,0x01,0x02 or 0x03 ‘v’ is 0x00,0x02,0x04,0x06,0x08,0x0A,0x0C,0x0E

‘r’ is 0x00,0x04,0x08 or 0x0C

‘yymmdd’ is Date ID (year/month/day) ‘xy’ is Reversion ID: x.y

Note: 1) Default value read depends on the status of LDOWN_n

Note: 2) Default value read depends on the status of TDO and LDOWN_n

Note: 3) Value depends on the current version of the FPGA content

Note: 4) Value depends on the status of the MAX 1618 ALERT pin

Table 5: Registers in the MS0 address space

2.2.5 Registers 0x10: Error-Code Replace Patterns

MS0 offset 0x21

When an error is signalled on the incoming data stream from the CSM then the word-ID bits [31:28] of the data word that contains the error is replaced by an Error-Code. The original word-ID bits are moved to bits [27:24] in order to make debugging easier because the original word type can probably still be traced (there is no guarantee however, since the error could have effected one of the word-ID bits).

When a TDC Parity Error was signalled by the CSM (bit [27] of the data word was set to ‘1’ by the CSM) then bits [31:28] will be replaced with the TDC Parity Error-Code bits [31:28] of the Error-Code Replace Patterns register.

When an input link parity check fails (bit [26] of the data word serves as a parity bit over all other 31 bits in the data word) then bits [31:28] will be replaced with the input link Parity Check Failure Error-Code bits [15:12] of the Error-Code Replace Patterns register.

When both errors (TDC Parity Error and input link Parity Check Failure) occur at the same time then the input link Parity Check Failure is given priority.

Summarizing register 0x10 bits:

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

2.2.6 Registers 0x11: Reserved

MS0 offset 0x23

This register is reserved. Writing doesn't care, reading always yields 0x00000000.

2.2.7 Register 0x12: LWC (Link Word Count) Pattern

MS0 offset 0x25

Register 0x12 contains a 32 bits LWC Pattern. This word is sent as the LWC word in the event fragment.

2.2.8 Register 0x13: BOL (Begin Of Link) Pattern

MS0 offset 0x27

Register 0x13 contains a 32 bits BOL Pattern. This word is sent as the BOL word in the event fragment.

2.2.9 Register 0x14: TLP (TDC Link Present; MROD-In Header) Pattern

MS0 offset 0x29

Register 0x14 contains an 8 bits TLP pattern in the bit positions [31:24]. For each read-out (one event), initiated by a Tetris register status word in the I2O_FIFO, a special word is sent to the output data stream, which contains these 8 bits in the positions [31:24] (see table 2). The lower 18 bits of this word give information about which TDCs have data (in sequence). This is actually the AND of the status from the Tetris register and the Partition Read-out Enable register (register 0x21).

Rule: If one of the lower 18 bits is '1' then data of the corresponding TDC should be in the output data stream.

Note: The following situation is an exception to this rule! If a partition full condition is met *after* the TLP word is sent to the output data stream but *before* the corresponding partition is actually read-out then no data is found in the output data stream since the corresponding partition was shut down before it was read-out. However, this will lead to a "Partition Full Interrupt" (see register 0x17). This should be very rare!

Note: A Maximum Event Size overflow (see register 0x1E) will shut down the read-out as well but there should at least be some data of the corresponding TDC in the output data stream.

2.2.10 Register 0x15: TWC (Trailer Word Count; MROD-In Trailer) Pattern

MS0 offset 0x2B

Register 0x15 contains an 8 bit TWC pattern in the bit positions [31:24]. For each read-out (one event), initiated by a Tetris register status word in the I2O_FIFO, a sequence of TDC data read from the partitions in the buffer memory is sent to the output data stream. A TWC

word is sent to the output data stream when all TDC data of the event have been sent. Bits [31:24] of the TWC word contain the 8 bits in the positions [31:24] of register 0x15 (see table 2).

Bits [23:12] of the TWC which is sent to the output data stream contain the Event-ID of the data which was just sent. Bits [11:0] contain the word count of the data in the event which was just sent. The word count includes the TLP and TWC words.

2.2.11 Register 0x16: Event Length FIFO

MS0 offset 0x2D

Register 0x16 is a read only register. When the register is read, one word is read from the Event Length FIFO that is connected to the SHARC. SHARC flags 2 and 3 (for Channel A and B, see table 7) signals whether the Event Length FIFO contains any data.

A spy mechanism (see register 0x29) selects events to be sent to the SHARC processor. If an event is selected then the data is transferred into the Event Data FIFO. The event length is sent to the Event Length FIFO during the transfer of the last word (TWC) to the Event Data FIFO. When the Event Length FIFO becomes full then the transfer of data to the Event Data FIFO is stopped. The SHARC is reading the Event Data FIFO using an endless DMA thus filling up its internal buffers. The SHARC can moderate this data stream when it stops reading the Event Length FIFO so this FIFO becomes full (after 16 Event Length entries).

Note that in rare occasions when in “MROD-X Mode” without debug (see register 0x1B, bits [4:5], table 6) and when the spy pre-scale factor is 0 (register 0x29) and when events are read from the buffer memory back to back then the Event Length FIFO may contain 17 entries before Full is signalled.

Note that when the Event Data FIFO is not read using endless DMA and when in “legacy MROD-1 mode” or in “MROD-X debug mode” (see register 0x1B, bits [4:5], table 6) and when the Event Data FIFO fills up (after 512 entries), then data transfer from the buffer memory partitions is halted. In the case where an event fragment is longer than 512 words, the head of the event fragment resides in the Event Data FIFO but the tail (TWC word) did not yet pass the output controller. Thus the event length is not yet sent to the Event Length FIFO. Although a part of the event fragment is waiting to be read-out from the Event Data FIFO, the Event Length FIFO may look empty.

Warning, reading an empty Event Length FIFO returns the last value read from the FIFO although this should be handled as garbage data.

Bits [27:16] contain the associated Event-ID

Bits [11:0] contain the word count of the event. The BOL, LWC, TLP and TWC words are included in the word count.

Note that this word count is always two more (BOL and LWC) than the word count in the TWC word.

2.2.12 Register 0x17: Interrupt Control IRQ0

MS0 offset 0x2F

This register controls the interrupt output IRQ0_n to the SHARC. The IRQ0_n lines from both MROD-In channels are wired OR-ed.

There are six interrupt sources for IRQ0. If register 0x17 is read back then bits [0:3] and bits [28:29] determine which of the six interrupt sources generated the interrupt.

Bit [0]: I2O_FIFO Full Interrupt. The I2O_FIFO is 512 words deep.

Bit [1]: Buffer Memory Partition Full Interrupt.

Bit [2]: Read-out Maximum Interrupt.

Bit [3]: TDC (Parity) Error Interrupt.

Writing a '1' to one of the bits [0:3] in register 0x17 clears the interrupt.

Bit [7:4]: Mask bits for the interrupt bits [0:3]. An interrupt bit can only be set when the corresponding mask bit is a '1'.

Bit [4]: I2O_FIFO Full Interrupt Mask. If '1' then an I2O_FIFO Full condition generates an interrupt.

Bit [5]: Buffer Memory Partition Full Interrupt Mask. If '1' then a Buffer Memory Partition Full condition generates an interrupt. Note that the Partition Read-out Enable Register (Register 0x21) can give more detailed information about which partition caused the interrupt.

Bit [6]: Read-out Maximum Interrupt Mask. If '1' then an interrupt is generated if an Event is read from a partition but the event contains more data words than is programmed in the "Maximum Event Size" register 0x1E. Note that the Partition Read-out Enable Register (Register 0x21) can give more detailed information about which partition caused the interrupt.

Bit [7]: TDC (Parity) Error General Interrupt Mask. If '1' then an interrupt is generated if a TDC (Parity) Error was signalled (bit [27] = '1') in the data from the input link and the TDC (Parity) Error Individual Interrupt Mask bit for the corresponding TDC is set (see register 0x18).

Bit [25:8]: Reflect the 18 (Parity) Error bits, one for each TDC.

Bit [26]: Overrun bit. If this bit is '1' then this means that there were one or more TDC (Parity) Error Interrupts that weren't read-out in time. Note that during an overrun condition, bits [25:8] of the register represent the status of the first error that occurred.

Bit [27]: Reserved

Bit [28]: Set when the Event Length FIFO (that is connected to the SHARC Processor) gets full, thus causing an IRQ0. The interrupt can be cleared by writing a '1' to this bit. Bit [30] is the corresponding mask bit. Note that when the Event Length FIFO is full, no more data will be sent to the Event Data FIFO. The threshold for the Event Length FIFO full flag is set to 16. This IRQ source can be helpful to tune the data rate for spying and monitoring.

Bit [29]: Set when the Event Data FIFO (that is connected to the SHARC Processor) gets full, thus causing an IRQ0. The interrupt can be cleared by writing a '1' to this bit. Bit [31] is the corresponding mask bit. Normally this condition will never occur since the SHARC should read the Event Data FIFO using an endless DMA. The Output FIFO is 512 words deep. This IRQ source can be helpful to tune the data rate for spying and monitoring.

Bit [30]: Event Length FIFO Full interrupt mask bit (see bit [28]).

Bit [31]: Event Data FIFO Full interrupt mask bit (see bit [29]).

2.2.13 Register 0x18: TDC (Parity) Error Individual Interrupt Mask

MS0 offset 0x31

With this register, (Parity) Errors in the data stream from individual TDCs can be masked out for the generation of a TDC (Parity) Error Interrupt.

Bit [0:17] corresponds to TDC0 to TDC17 respectively. When a bit is set '1' then an interrupt is generated when a TDC (Parity) Error was signalled (bit [27] = '1') in the data stream from the corresponding TDC.

Note that the TDC (Parity) Error General Interrupt Mask (see register 0x17 bit [7]) should also be enabled in order to generate an interrupt.

2.2.14 Register 0x19: Input Link Interrupt; IRQ1

MS0 offset 0x33

This register controls the interrupt output IRQ1_n to the SHARC. The IRQ1_n lines from both MROD-In channels are wired OR-ed.

IRQ1_n occurs whenever an error is detected on the input link. There are two sources that can generate this interrupt, an input link Parity Check Failure and LDOWN_n. Both of these can be masked out by bit [4] and bit [5] respectively.

The link is tested by inserting parity in bit [26]. By default 'odd' parity is chosen, so when the parity is put into bit [26] then all 32 data bits should have odd parity. 'Even' parity can be chosen if bit [9] of the Test & input link Control Status Register (0x1B) is set to '1'.

Bit [0]: Input link Parity Check Failure. If this bit is '1' then this means that an input link Parity Check Failure generated the interrupt. The input link Parity Check Failure interrupt can be cleared by writing a '1' to bit [0].

Bit [1]: Input link Parity Check Failure Overrun bit. If this bit is '1' then this means that there were one or more input link Parity Check Failure conditions that weren't read-out in time.

Bit [2]: LDOWN interrupt. If this bit is '1' then this means that a LDOWN condition generated the interrupt. The LDOWN interrupt can be cleared by writing a '1' to bit [2].

Note that the status of the LDOWN_n signal can be read back directly (unlatched) in register 0x1B bit [8].

Bit [3]: Reserved, read back as '0'.

Bit [4]: Input link Parity Check Failure Mask Bit. When set to '1' an input link Parity Check Failure condition leads to IRQ1_n to be asserted.

Bit [5]: LDOWN_n Mask Bit. When set to '1' a LDOWN_n condition leads to IRQ1_n to be asserted.

2.2.15 Register 0x1A: Early and Late Event-ID; IRQ2

MS0 offset 0x35

This register controls the interrupt output IRQ2_n to the SHARC. The IRQ2_n lines from both MROD-In channels are wired OR-ed.

IRQ2_n occurs whenever a TDC-Trailer is received with an Event-ID, which is out of the Expected Event-ID window. Such an Event-ID is either 'Early' or 'Late' (see table 1).

Bit [11:0]: Represent the Event-ID, which caused the interrupt.

Bit [16:12]: Represent the TDC slot number, which caused the interrupt.

Bit [17]: If this bit is '1' then the Event-ID is 'Early'. For example, when the Expected Event-ID is 51 and the received Event-ID for a TDC is 102, which is 'in the future' with respect to the Expected Event-ID.

Bit [18]: If this bit is '1' then the Event-ID is 'Late'. For example, when the Expected Event-ID is 51 and the received Event-ID for a TDC is 50, which is 'in the past' with respect to the Expected Event-ID.

Bit [19]: Overrun bit. If this bit is '1' then this means that there were one or more errors that weren't read-out in time. Note that during an overrun condition, bits [18:0] of the register represent the status of the first error that occurred.

Writing any value to this register clears IRQ2_n.

Take care in enabling IRQ2 on the SHARC. When the MROD-In is being synchronised with respect to the Event-IDs, the Expected Event-ID register (register 0x1F) is programmed with an Event-ID, which will be valid in the near future. 'Late' interrupts will be generated until the Event-IDs from the TDC-Trailers are within the Expected Event-ID window. Therefore it is advisable to enable IRQ2, only when the MROD-In is in synchronisation and Event-IDs should fall within the Expected Event-ID window.

2.2.16 Registers 0x1B: Test & Control Status Register

MS0 offset 0x37

Register 0x1B is a general purpose Control Status register. It contains the following bits:

Bit [0]: If in test mode, taking this bit from '0' to '1' writes test data from register 0x1C and 0x1D to the MROD-In input link (edge sensitive).

Bit [1]: '0' Normal operation (input link); default on power up.

'1' test mode operation (internal feedback in the MROD-In FPGA). In test mode (bit [1] = '1'), the input link is disabled and taken over by the SHARC. The SHARC can now write data as if it came from the input link.

Bit [2]: '0' Normal operation. FPGA read mode; default on power up.

'1' test mode operation. SHARC has Read/Write access to buffer memory. The buffer memory has two cycle-share ports. The first port is always dedicated to the input link. During normal operation, the second port is dedicated to the output controller in the MROD-In FPGA which reads data from the buffer memory. For test purposes the SHARC may read or write the buffer memory as well. When bit [2] of the Test Control register is set then the second port is switched from 'FPGA Read mode' to SHARC 'Read/Write mode'.

Note that bit [2] distorts the operation of the output controller of the MROD-In FPGA. Therefore bit [2] should only be set for debugging purposes.

Bit [3]: Zero Suppress. Output data is zero suppressed if this bit is '1'. This means that Headers and Trailers of TDCs that do not have data are flushed from the data-stream. Note that the TLP word still contains a flag bit which signals that the zero suppressed TDC was in the data-stream.

Bit [4]: Enable Event Building via RocketIO (see table 6).

'0' enables the legacy MROD-1 mode. In this mode the output data stream is stopped when the spy channel connected to the SHARC becomes full, i.e. either the Event Data FIFO or the Event Length FIFO becomes full.

'1' enables the MROD-X mode. In this mode the output data stream is stopped only when the Event Data FIFO or the Event Length FIFO in the RocketIO link (connected to the MROD-Out FPGA) become full.

Note that in legacy MROD-1 mode, the output data is also send over the RocketIO data channel but the data over this channel could be corrupted when a full condition is occurring.

Bit [5]: MROD-X debug mode (see table 6). When this bit is '1' then read-out is stopped as soon as one of the output streams (RocketIO Event Data FIFO, RocketIO Event Length FIFO, SHARC Event Data FIFO or SHARC 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 [4]	Bit [5]	SHARC spy channel Full (= Event Data FIFO or Event Length FIFO full)	RocketIO Full (= Event Data FIFO or Event Length FIFO full)	Mode
0	0	√		MROD-1
0	1	√		MROD-1
1	0		√	MROD-X
1	1	√	√	MROD-X debug

Table 6: Output data stream FIFO Full flags taken into account

Bit [6]: Taking this bit from '0' to '1' resets the RocketIO Link (edge sensitive).

Bit [7]: When read gives the status of the LDOWN_n line of the RocketIO Link (Note that '0' read back means link is down).

Bit [8]: When read gives the status of the LDOWN_n line of the input link (Note that '0' read back means link is down).

Bit [9]: '0' (default) 'Odd' Parity Check on the input link, '1' for 'Even' Parity Check on the input link. Note that bit [26] of the input link data is defined as the parity bit.

Bit [10]: Taking this bit from '0' to '1' resets the input link (edge sensitive).

Bit [11]: When this bit is '1' the data pipeline in the FPGA is in 'Freeze' mode. Note that the default mode (after a Rst_n) is Freeze ON. For a detailed description of the Freeze and input link Reset bits (bit [10:11]), see also the chapter "Reset Facilities".

Bit [15:12]: Four LEDs can be connected to these bits. Writing a '1' puts on the light. These bits can be read back.

Bit [20:16]: Five spare bits that read the SFP status signals. Bit [16] is MOD_DEF0, bit [17] is MOD_DEF1, bit [18] is MOD_DEF2, bit [19] is TX_Fault and bit [20] is RX_LOS.

Bit [23:21]: Channel ID bits (read only). “000” corresponds to input Channel 1A, “001” to 1B, “010” to 2A etc.

Bit [24]: GOL Test Mode Run. When this bit is set to ‘1’ then the content of the GOL Test FIFO is transmitted. When this bit is ‘0’ the transmission is stopped.

Bit [25]: GOL Test Mode Circulate bit. When this bit is set to ‘1’ then the GOL Test FIFO is switched into circular mode. Words that are read-out of the FIFO (and that are transmitted) are immediately written into the GOL Test FIFO again. When this bit is set to ‘0’, then the SHARC can fill the GOL Test FIFO by writing to the GOL Test FIFO register (0x2A). Note that the SHARC should take care not to overflow the FIFO (which is 4095 words deep). See Appendix A for more details about the Test Mode Data format.

Bit [26]: GOL Test Mode Trigger select bit. When this bit is set to ‘1’ then triggered test mode is selected. Each time there is a Level 1 Accept then one event (which resides in the GOL Test FIFO) is sent. When this bit is ‘0’ then un-triggered test mode is used and events are sent one after another. See Appendix A for more details about the Test Mode Data format.

Bit [31:27]: Reserved and read back as ‘0’.

2.2.17 Registers 0x1C: Test Link Data Register

MS0 offset 0x39

Register 0x1C is used in test mode and contains the 32 data bits, which would normally come from the input link LD [31:0] bus.

2.2.18 Registers 0x1D: Test Link Control Register

MS0 offset 0x3B

Register 0x1D is reserved. The value of register 0x1D should be 0x00000003 (default after power up).

2.2.19 Registers 0x1E: Maximum Event Size

MS0 offset 0x3D

The 12 bit value in this register determines the maximum allowable event size for a TDC. If the number of words in an event, which is being read-out from a partition in the buffer memory, is more than the value in this register then the event is truncated. Read-out for the partition is stopped by clearing the corresponding partition read-out enable bit in register 0x21. A “Read-out Maximum” interrupt will be generated, to signal the SHARC that something went wrong (see Register 0x17).

Register 0x1E will power up to a default value of 0x400 (1K Words).

Note that the value 0 for this register will give unpredictable results and should be avoided.

Note also that due to pipelining, one extra word *could* be transferred to the output data stream. This depends on the full status of the output FIFOs.

2.2.20 Register 0x1F: Expected Event-ID

MS0 offset 0x3F

Register 0x1F contains 12 bits, which represents the Event-ID that is to be expected as the next Event-ID to receive from the TDCs. Normally this register is only written during initialisation of the MROD-In. When a complete event is gathered from its TDC inputs then the hardware of the MROD-In increments the Expected Event-ID. The register can be inspected at any time.

It should be possible to synchronize the MROD-In during a run. Via the TTC system [4] it is known which Event-ID is currently used. This value, plus an added margin due to events which can already be in the pipeline of the Data Acquisition System, can be programmed in the Expected Event-ID register. The MROD-In then catches up as soon as this Event-ID comes along. Note that there will be 'Late' conditions so probably IRQ2 (register 0x1A) should be disabled until the MROD-In is in synchronisation.

2.2.21 Register 0x20: TDC Mask Register

MS0 offset 0x41

Register 0x20 contains an 18 bits TDC-Mask. Writing a '1' to a bit enables the corresponding TDC. If the bit is written '0' then the corresponding TDC is disabled and the Tetris register will not wait for the corresponding TDC-Trailer to arrive in order to generate a Row-Out condition. Bit [0] corresponds with TDC0; bit [1] with TDC1 and so on.

2.2.22 Register 0x21: Partition Read-out Enable

MS0 offset 0x43

Register 0x21 contains 18 partition read-out enable bits. When a bit is '1' then the corresponding buffer memory partition is enabled for read-out, if it is '0' then the partition is skipped during read-out. Bit [0] corresponds with TDC0; bit [1] with TDC1 and so on.

There are two conditions where read-out for a TDC is disabled by the hardware. The first condition is when the corresponding TDC partition in the buffer memory is full. Since the partitions are organized as circular buffers, there is a danger of overwriting data, which was not read-out yet, so it is no longer possible to have a consistent image of the TDC event data in the buffer memory with respect to the content of the Tetris register. Therefore read-out may not be able to find the TDC-Trailers anymore for which it is looking in the buffer memory. To avoid read-out problems the read-out is disabled. This condition should never occur because the trigger should already have been moderated via the ROD-Busy signal (see register 0x26) when the buffer memory partition became half full.

Note that the Partition Full condition is met when a partition contains 8000 to 8064 words (8K – 192 to 8K -128) instead of the maximum possible 8192 (8K) words. This is due to the fact that the FPGA contains a pipeline to the buffer memory and only the upper 7 bits of the read- and write pointers of the partitions are used (to save hardware resources in the FPGA) to determine a full condition. The full condition is met when:

```
Write Pointer [12:6] = Read Pointer [12:6] – 2
```

Note the following example: when the read pointer is 0x043F (bits [12:6] = 16) a partition full condition is met when the write pointer increases to 0x0380 (bits [12:6] = 14), a difference of 8001 words.

The second condition where read-out for a TDC is disabled is when during read-out of a single event a maximum event size is exceeded. In such a case it is likely that a TDC is out of order. Such a condition forces the read-out for that particular TDC to be disabled. This prevents lots of wrong data from blocking the output bandwidth. In either case IRQ0 is asserted (if enabled, see register 0x17 Interrupt Control IRQ0).

The SHARC can read the Partition Read-out Enable register (register 0x21). This gives the SHARC information about which condition and TDC caused the interrupt. The SHARC can write to the register as well, but enabling read-out should only be done during initialisation of the MROD-In. If the read-out is enabled during runtime then the behaviour depends on the values of the read- and write-pointers of the buffer memory partition. Therefore this could result in an immediate assertion of the partition full interrupt, which shuts down the output again. It could also result in a maximum event length interrupt since the relation between the buffer memory partition data and the Tetris register is lost so the hardware may keep searching for a TDC-Trailer which was overwritten.

2.2.23 Registers 0x22: Separator Flags Register

MS0 offset 0x45

Register 0x22 is a read only register. The register is updated with the bits of the Separator word, each time it is recognised in the input data stream (see registers 0x00 to 0x03).

Bits that are not used for Separator recognition (see Separator Mask registers 0x01) can be used to transfer data from the CSM to the MROD-In.

Note that the SHARC must poll the content of the Separator Flags Register so this is not a high-speed data channel. The Separator word could typically be used to transfer for example a front-end ID.

2.2.24 Registers 0x23: Date & Revision ID Register

MS0 offset 0x47

Register 0x23 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.

2.2.25 Registers 0x24: FPGA Temperature Register

MS0 offset 0x49

The bits in register 0x24 are connected to a MAX1618 “Remote Temperature Sensor” device [5] 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.

2.2.26 Registers 0x25: Zero Suppress Override Count Register

MS0 offset 0x4B

Bits [15:0] of register 0x25 determine how often a complete (non-zero suppressed) read-out of an event is generated when zero suppress mode is active. When bit [16] is cleared ‘0’ then Zero Suppress Override mode is disabled.

When zero suppress mode is active (bit [3] of the Test & Input Link Control Status Register, 0x1B) then all TDCs that do not have data are suppressed in the data stream. This means that the TDC header and trailer pairs with no TDC data in between are removed. In order to make debugging possible, once every so often, a complete event will be outputted without zero suppression. The default value for register 0x25 is 0x10400 which means that every 1K events, a complete event is output in non zero suppress mode.

Note that (when bit [16] = ‘1’) the first event is always completely outputted. Note also, that loading value 0x10000 in the Zero Suppress Override Count Register means that all events are completely outputted.

2.2.27 Registers 0x26: Channel Busy Mask Register

MS0 offset 0x4D

If one of the 18 partitions in the buffer memory gets half full, or when the I2O_FIFO gets half full (after 256 entries), then the Channel Busy signal is asserted. With register 0x26, each buffer memory partition and the I2O_FIFO can be masked out so that the corresponding half full condition will not lead to a Channel Busy.

Bit [0] corresponds with TDC0; bit [1] with TDC1 and so on. Bit 18 corresponds to the I2O_FIFO half full flag.

When a bit is cleared (‘0’) then the corresponding half full signal is masked out and does not generate a Channel Busy signal.

Note that when read-out of a buffer memory partition is shut down (see Partition Read-Out Enable Register 0x21) this will not automatically mask the corresponding bit in the Channel Busy Mask register.

Register 0x26 will power up to a default value of 0x7FFFF. This means that a Channel Busy signal is generated whenever one of the buffer memory partitions or the I2O_FIFO becomes half full.

2.2.28 Registers 0x27: Channel Busy Status Register

MS0 offset 0x4F

This register is a read-only register. It has information on what buffer memory partition(s) lead to a Channel Busy signal being asserted. Bit [0] corresponds with the buffer memory partition of TDC0; bit [1] with TDC1 and so on. Bit [18] corresponds to the I2O_FIFO half full flag.

Note that a bit will read ‘0’ if the corresponding mask bit in the Channel Busy Mask Register (0x26) is cleared, irrespective of the buffer memory partition status.

2.2.29 Registers 0x28: TDC Limit Register

MS0 offset 0x51

This 12 bit register determines how many TDC data words are passed to the output stream at maximum. The amount of TDC data words is the value of this register plus 1.

The purpose of this register is to limit the event size for each TDC. When a TDC has more data words than the pre-programmed limit value then the surplus of the TDC data words are flushed and will not be send to the output stream. However, the EOT (TDC Trailer) will be passed to the output stream. Note that the EOT word contains a word count. This counter holds the original (unlimited) amount of TDC data words + 2 for the TLP (= MROD-In Header) and TWC (= MROD-In Trailer) words.

In contrast to the Maximum Event size register (0x1E), the TDC Limit register keeps reading from the buffer memory partition until an EOT is found. Readout is not shutdown as is the case when the Maximum Event size is passed.

Register 0x28 will power up to a default value of 0x060 (96+1 Words).

2.2.30 Registers 0x29: SHARC Spy Count Register

MS0 offset 0x53

Bits [15:0] of register 0x29 determine how often a complete read-out of an event is send to the SHARC for debugging and/or monitoring purposes. When bit [16] is cleared ‘0’ then spy mode is disabled.

The default value for register 0x29 is 0x10000 which means that every event is outputted (i.e. all events are spied).

Note that (when bit [16] = ‘1’) the first event of a run is always sent.

2.2.31 Registers 0x2A: GOL Test FIFO

MS0 offset 0x55

Via this register the GOL input link Test FIFO can be filled with test data. This register is a write only register. When the register is read, the result will be 0x00000000.

The GOL Link Test Mode bits [26:24] in the Test and Control Status register (0x1B) control the flow of the test data that are sent out via the transmitter of the input link (so it can be looped back externally).

2.2.32 Registers 0x2B– 0x3F: Reserved

MS0 offset 0x57– 0x7F

Registers 0x2B to 0x3F are reserved.

2.3 SHARC MS1 address space: SHARC Event Data FIFO

A value from the SHARC Event Data FIFO is read when a read cycle on any address in MS1 occurs. Address line A[21] of the SHARC determines the channel to be read (see table 3).

Note that reading from an empty Event Data FIFO will return the value of the last word, which was read before the FIFO went empty. Normally this will never occur since the Event Data FIFO will be read-out using DMA, in such a mode that a DMA request is sent by the FPGA for each word in the Event Data FIFO.

2.4 SHARC MS2 address space: Buffer Memory Access

For test purposes the SHARC may read or write the buffer memory. When bit [2] of the Test Control register (Register 0x1B) is '1' then the SHARC gets Read/Write access to the buffer memory. Be careful, this distorts the operation of the output controller of the FPGA; this mode should only be used for debugging purposes!

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" [3]).

It is important to note the difference between the Buffer memory address lines and the SHARC address lines:

```
Buffer Memory Address = SHARC MS2 Offset * 2 + 1
```

The SHARC address lines A[20:1] are re-routed to Buffer Memory address lines A[19:0] so the SHARC can address 1 MWords (= 4 MB). Currently the buffer memory is 256 KWords (1 MB) and this memory resides in the lower 256 KWords.

Note that SHARC address line A[21] is being used to select either Channel A or Channel B.

2.5 SHARC MS3 address space: Not Used

MS3 address space is not used by the MROD-In SHARC.

2.6 SHARC Flags:

Table 7 gives an overview of the SHARC flags.

Flag 0 and Flag 1 must be configured as outputs. These flags control the reset pins of the FPGA of Channel A and B. Note that the reset signal is low active. See also 'Reset Facilities' below.

Note that the SHARC core frequency operates at twice the frequency of CLKIN. If a flag output should be active for one external clock cycle then the core should execute a NOP instruction. See also page 11-20 "Flag Outputs" of the ADSP-21160 Hardware Reference [3].

Flag 2 and 3 must be configured as inputs and are connected to the Empty output of the SHARC Event Length FIFOs of the MROD-In Channel A and B respectively.

Flag Number	Direction	Description
0	Output	FPGA Reset Channel A
1	Output	FPGA Reset Channel B
2	Input	Empty status of Channel A SHARC Event Length FIFO
3	Input	Empty status of Channel B SHARC Event Length FIFO

Table 7: SHARC Flags

2.7 SHARC Interrupt Lines

The three interrupt lines of the SHARC are shared (wire-or-ed) by the FPGAs of Channel A and Channel B.

IRQ0 (see register 0x17 and 0x18) is asserted when:

- the I2O_FIFO is full
- a buffer memory partition is full
- an event was read-out which was longer than a pre-programmed maximum
- a TDC (Parity) Error was received
- SHARC Event Data FIFO is full
- SHARC Event Length FIFO is full.

IRQ1 (see register 0x19) is asserted when:

- an input link Parity Check Failure occurs
- the input link goes down (LDOWN_n)

IRQ2 is asserted whenever an EOT (TDC-Trailer) is received with an Event-ID, which is out of the Expected Event-ID window (‘Early’ or ‘Late’, see register 0x1A and table 1).

3 Reset Facilities

There are four reset signals: “POR” (Power-On Reset), “SYSRESET_n”, “Rst_n” and “SHARC_Rst_n”.

A Power-On Reset or an asserted “SYSRESET_n” signal generates a General Reset that resets the SHARC, starts a re-configuration of the FPGAs and resets the FPGAs (including the Registers, Pipeline, Address Generators, Tetris register, I2O_FIFO, Event Data FIFOs, Event Length FIFOs etc.).

Asserting “Rst_n” resets the FPGAs (including the Registers, Pipeline, Address Generators, Tetris register, I2O_FIFO, Event Data FIFOs, Event Length FIFOs etc.) but not the input link and the input link FIFO (which resides in the FPGA). The “Rst_n” signals of the two FPGAs can be controlled by the SHARC using flag 0 and 1 respectively.

Note that in all cases, the FPGA enters Freeze mode (See Register 0x1B bit [11]). This means that data in the input link FIFO is not read into the data pipeline in the FPGA.

An input link Reset can be generated by taking bit [10] of register 0x1B from ‘0’ to ‘1’. This resets the input link and the input link FIFO. The input link will force LDOWN_n low until the initialisation phase of the input link is complete. LDOWN_n will then go high again. The input link is now up and running again.

The following sequence should give a proper initialisation. First there will be a Power-Up Reset or a “SYSRESET_n”. After that, the SHARC is booted through link 4. In the mean time garbage data had a chance to enter the input link FIFO in the FPGAs. Therefore the SHARC activates flags 0 and/or 1 (Rst_n for Channel A and/or B). The FPGA(s) are now reset and the data pipeline wakes up in Freeze mode so no data is read from the input link FIFO, which might contain garbage. The SHARC resets the input link(s) by taking bit [10] of register 0x1B from ‘0’ to ‘1’. The input link is reset and the input link FIFO in the FPGA is cleared. The Freeze condition can now be taken away (clear bit [11] of register 0x1B).

Asserting “SHARC_Rst_n” resets the SHARC. Note that this also affects the SHARC flag pins 0 and 1 that drive the “Rst_n” signals to the MROD-In FPGAs. “SHARC_Rst_n” can be asserted via a register in the MROD-Out.

A Appendix: GOL Test Mode Data format

The Test FIFO should contain data in CSM format, which is a sequence of one separator word (0xD0000000) followed by 18 TDC data words or zero words (0x0) when that TDC has no data. Each sequence is followed by another.

The GOL Test generator may operate in Transparent mode (Reg. 0x1B, bit [25] ='0') or in Circular mode (bit [25] ='1'). In Transparent mode, all data written by the SHARC to the FIFO register (Reg. 0x2A) will once be transmitted on the GOL link. In Circular mode (in both un-triggered and triggered mode) the whole sequence of preloaded data in the FIFO will be transmitted repeatedly on the GOL link.

When using the GOL Test in Circular mode, a special separator word ("End of Event": 0xD0E00000) is needed to mark the end of event: all TDCs have sent the trailer word belonging to the same event.

The GOL test generator uses the "End Of Event" Separator to stop sending when the "Run" bit is turned off (Register 0x1B, bit [26:24] changes from "011" to "010" or "111" to "110"). Note that in Circular mode, the event numbers in the TDC header and trailer words are updated with a new 12 bit event number. This event number is a counter that is incremented each time an "End of Event" separator word from the FIFO is passed to the GOL transmitter.

4 References

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

2 The MROD data format and the tower partitioning of the MDT chambers, T. Wijnen, Radboud University of Nijmegen:

http://doc.cern.ch/archive/electronic/cern/others/atlnot/Note/daq/daq-2003-023.pdf

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

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

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

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

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/.