MROD-In Programmers Manual

Introduction

The MROD-In 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 (figure 1). The description below will focus on one channel and one FPGA since the processing of each channel is equal.

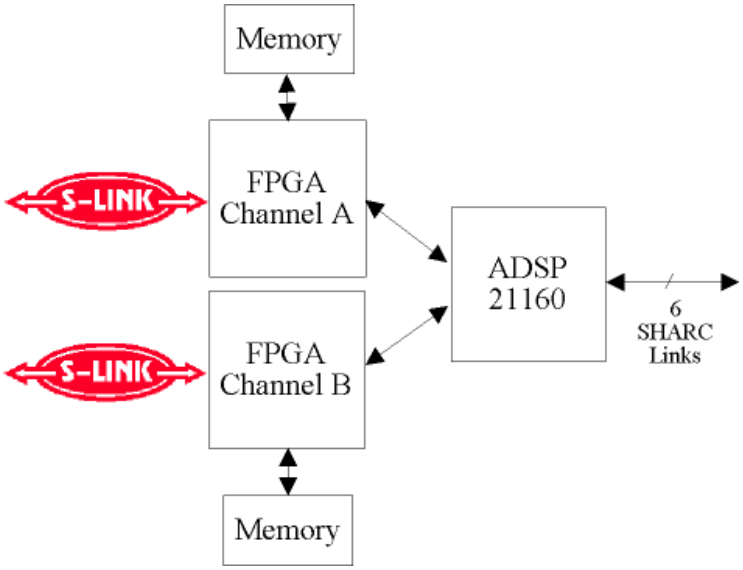

Figure 1: MROD-In block diagram.

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 CSM and send over an S-LINK to the MROD-In.

Separator + Parity Error	TDC 0	TDC 1	…	TDC 17	Separator + Parity Error	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 is programmed to recognise a Separator word and a TDC Trailer word. To do so, the FPGA on the MROD-In contains two comparators. For both, Separator and TDC Trailer, a bit pattern and a bit mask can be programmed. The pattern and mask bits are organised as two 32-bit words; in the last word, only bit 0 is significant.

This single bit signals whether or not the first 32-bit word needs to be an S-LINK control word or not.
The lower 18 bits [17..0] of the Separator word are used to signal TDC parity errors, one bit for each TDC. Bit 18 and 19 can possibly be used to propagate a serial bit stream from the front-end to the MROD-In.
When a Separator is recognised a time slot counter is started which will count from 0 up until 17. While counting, data from the S-LINK input 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 (NoData is defined as LD[31..0] = 0x0 and LCTRL_n = '1').
If there are more than 18 words send after a Separator is recognised then the surplus of data will be discarded.
When a TDC Trailer is recognised then the Event-ID of the trailer is checked whether it falls within a 16 wide window of 'Expected' Event-ID's. 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'. 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).

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

When all Trailers with the Expected Event-ID of enabled TDCs are received then a 'Row-Out' condition is send 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.
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 S-LINK input 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. This data is placed, either with or without Zero Suppression, in an output FIFO (Outp_FIFO) that is read by the SHARC under DMA control with a maximum bandwidth of 80 MB/sec. Table 2 lists the format of this data.

Data	Remark
MROD Header	Bits [31..24] MROD Header Pattern (Register 0x0C) 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 Trailer	Bits [31..24] MROD Trailer Pattern (Register 0x0D) Bits [11..0] contain Event Word Count
MROD Header	Next event

Table 2: Output data format

Note that the SHARC host bus is used by two FPGAs so the bandwidth of each FPGA is 40 MB/sec when the bandwidth is equally distributed over these two FPGAs.
The number of data words within a single event is send to an Event Length FIFO, which can be read by the SHARC. The empty status of the Event Length FIFO can be inspected by the SHARC by means of one of its flag pins.
Note that the SHARC services two MROD-In channels. The Outp_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 6.

Each word that is read from the I2O_FIFO contains the status of the Tetris Register and the Expected Event-ID. 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 to the SHARC.

Each channel of the MROD-In can be put in “Test Mode”. Some Control and Status registers can feed data to the S-LINK input. 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 S-LINK input and the output controller of the FPGA during normal operation.
Several interrupt sources signal special conditions or errors.
All 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.

SHARC Memory regions

SHARC Address line A[20] 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[20]	Output FIFO DMA Channel	Event Length FIFO empty status (see also table 6)
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[20] of the SHARC is used to address the FPGA of channel A or B, memory regions MS0, MS1 and MS2 should be at least 2 MWord long (1 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 to 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").

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 EBx=AM are sub-patterns of the Wait register described in table 8-3 and table 8.4 of the "SHARC Technical Specification", or table 5-4 and table 5-5 of the "SHARC DSP Hardware Reference".
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

SHARC MS0 address space: Control and Status Registers

Register 0x00 and 0x02 (MS0 offset 0x01 and 0x05): Separator Pattern & Mask

Register 0x00 contains the 14-bit pattern that must match the Separator data (bits [31..18]) on the S-LINK. A 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 data on the S-LINK. If all bits match then a Separator is signalled.
If a mask bit is reset to '0' the corresponding bit in the pattern register may be different from the bit in the S-LINK data.
Separator bits 17 to 0 are used to signal TDC parity errors therefore the corresponding bits in the Separator Pattern register (0x00) and Separator Mask register (0x02) are written don't care and read back as '0'. TDC Parity errors are signalled by IRQ0 (see register 0x0F).
Separator bit 18 and/or 19 can possibly be used to propagate a serial bit stream from the front-end to the MROD-In. Whenever a Separator word is recognised these bits are passed to register 0x19.
Note that when these bits 18 and/or 19 are used for serial data transmission then bits 18 and/or 19 in the Separator Mask register should be '0' so a Separator word is always recognised regardless the state of these bits.
Reading back registers 0x00 or 0x02 yields the value written into them.

Register 0x01 and 0x03 (MS0 offset 0x03 and 0x07): Separator Control Bit Pattern & Mask

Register 0x01 contains only one bit. This bit must match the control bit of the Separator data on the S-LINK but only if the mask bit (register 0x03) is set to '1'. Note that the S-LINK control bit is low active (LCTRL_n).
Reading back registers 0x01 or 0x03 yields either 0x00000000 or 0x00000001 depending on the value written into the bit 0 location. Bits 31 to 1 are always read back as '0'.

Registers 0x04 to 0x07 (MS0 offset 0x09 to 0x0F): TDC Header Pattern & Mask

The description of these registers is the same as for registers 0x00 to 0x03, except that registers 0x04 to 0x07 control the recognition of a TDC Header condition. The TDC Header needs to be recognized when data is to be Zero Suppressed.
There is a small difference between registers 0x00-0x03 and 0x04-0x07. Bits 23 down to 12 in the TDC Header are reserved for a 12 bit Event-ID. Therefore these bits are don't care when they are written to the TDC Header Pattern (0x04) and TDC Header Mask (0x06) registers. When these registers are read back bits 23 down to 12 will be '0'.

Register	MS0 offset	Function	Write	Read
0x00	0x01	Separator Pattern	0xnnnr----	0xnnnr0000
0x01	0x03	Separator Control Bit Pattern	0x-------p	0x0000000p
0x02	0x05	Separator Mask	0xnnnr----	0xnnnr0000
0x03	0x07	Separator Control Bit Mask	0x-------p	0x0000000p
0x04	0x09	TDC Header Pattern	0xnn---nnn	0xnn000nnn
0x05	0x0B	TDC Header Control Bit Pattern	0x-------p	0x0000000p
0x06	0x0D	TDC Header Mask	0xnn---nnn	0xnn000nnn
0x07	0x0F	TDC Header Control Bit Mask	0x-------p	0x0000000p
0x08	0x11	TDC Trailer Pattern	0xnn---nnn	0xnn000nnn
0x09	0x13	TDC Trailer Control Bit Pattern	0x-------p	0x0000000p
0x0A	0x15	TDC Trailer Mask	0xnn---nnn	0xnn000nnn
0x0B	0x17	TDC Trailer Control Bit Mask	0x-------p	0x0000000p
0x0C	0x19	MROD Header Pattern	0xnn------	0xnn000000
0x0D	0x1B	MROD Trailer Pattern	0xnn------	0xnn000000
0x0E	0x1D	Event Length FIFO	0x--------	0x0nnn0nnn
0x0F	0x1F	Interrupt Control IRQ0	0x------nn	0x0snnnnnn
0x10	0x21	S-LINK Interrupt; IRQ1	0x—rt----	0x00nnnnnn
0x11	0x23	Early and Late Event-ID; IRQ2	0x--------	0x000nnnnn
0x12	0x25	Test & S-LINK Control Status Register	0x----snnn	0x0000snnn
0x13	0x27	Test Link Data Register	0xnnnnnnnn	0xnnnnnnnn
0x14	0x29	Test Link Control Register	0x-------q	0x0000000q
0x15	0x2B	Maximum Event Size	0x-----nnn	0x00000nnn
0x16	0x2D	Expected Event-ID	0x-----nnn	0x00000nnn
0x17	0x2F	TDC Mask Register	0x---qnnnn	0x000qnnnn
0x18	0x31	Partition Read-out Enable	0x---qnnnn	0x000qnnnn
0x19	0x33	Serial Register	0x--------	0x0000000q

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, 0x04, 0x08 or 0x0C
‘s’ is one out of 0x00 to 0x07
‘t’ is 0x00 or 0x08

Table 5: Registers in the MS0 address space

Registers 0x08 to 0x0B (MS0 offset 0x11 to 0x17):
TDC Trailer Pattern & Mask

The description of these registers is the same as for the TDC Header Pattern and Mask registers 0x04 to 0x07, except that registers 0x08 to 0x0B control the recognition of a TDC Trailer condition.

Register 0x0C (MS0 offset 0x19):
MROD Header Pattern

Register 0x0C contains an 8 bits MROD Header Pattern in the bit positions 31 down to 24. For each read-out (one event), initiated by a Tetris register status word in the I2O_FIFO, a special first word is send to the output FIFO, which contains these 8 bits in the positions 31 down to 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 Read-out Enable register (register 0x18).
Rule: If one of the lower 18 bits is '1' then data of the corresponding TDC should be in the output stream.
Note: The following situation is an exception to this rule! If a partition full condition is met *after* the MROD header is send to the Output FIFO but *before* the corresponding partition is actually read-out then no data is found in the output stream since the corresponding partition was shut down before it was read-out. However, this will lead to a "Partition Full Interrupt" (see register 0x0F). This should be very rare!
Note: A Maximum Event Size overflow (see register 0x0E) will shut down the read-out as well but there should at least be some data of the corresponding TDC in the output stream!

Register 0x0D (MS0 offset 0x1B):
MROD Trailer Pattern

Register 0x0D contains an 8 bits MROD Trailer Pattern in the bit positions 31 down to 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 send to the output FIFO. A MROD Trailer word is send to the output FIFO when all TDC data of the event is send to the output FIFO. Bits 31 down to 24 of the MROD Trailer word contain the 8 bits in the positions 31 down to 24 of register 0x0D.
The lower 12 bits of the MROD Trailer which is send to the output FIFO contain the word count of the data in the event which was just send. The word count includes the MROD Header and MROD Trailer words.

Register 0x0E (MS0 offset 0x1D):
Event Length FIFO

Register 0x0E is a read only register. When the register is read, one word is read from the Event Length FIFO.

SHARC flags 2 and 3 (for Channel A and B, see table 6) signals whether the Event Length FIFO contains any data.
The event length is send to the Event Length FIFO during the transfer of the last word of the event (MROD-Trailer) to the Output FIFO. When the Output FIFO fills up, the data transfer from the buffer memory partitions is halted. In this case the event cannot be completed by sending the MROD-Trailer to the output FIFO (event length > 128) and thus the event length is not yet send to the Event Length FIFO. Although a part of the event is waiting to be read out of the Output FIFO, the Event Length FIFO may look empty. Normally this situation will not occur since the Output FIFO is continuously read with a DMA transfer.
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 Event-ID
Bits [11..00] contain the word count of the event. The MROD-Header and –Trailer are included in the word count.

Register 0x0F (MS0 offset 0x1F):
Interrupt Control IRQ0

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 four interrupt sources for IRQ0. If register 0x0F is read back then bits 0 to 3 determine which of the four interrupt sources generated the interrupt.
Bit 0: I2O_FIFO Full Interrupt.
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 these bits in register 0x0F clears the interrupt.

Bits 4-7: Mask bits for the interrupt bits (bits 0 to 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 0x18) 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 0x15. Note that the Partition Read-out Enable Register (Register 0x18) can give more detailed information about which partition caused the interrupt.
Bit 7: TDC Parity Error Interrupt Mask. If ‘1’ then an interrupt is generated if a Separator word contains one or more bit’s, which signal a TDC Parity error (Separator word bits 17 down to 0).

Bits 25-8: Reflect the 18 Parity Error bits of the Separator word.
Bit 26: Overrun bit. If this bit is '1' then this means that there were one or more Separator words, which contained TDC Parity errors that weren't read-out in time. Note that during an overrun condition, bits 25 down to 8 of the register represent the status of the first error that occurred.

Register 0x10 (MS0 offset 0x21):
S-LINK Interrupt IRQ1

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 S-LINK interface.
There are two sources that can generate this interrupt, LDERR_n and LDOWN_n. Both of these can be masked out by bit 22 and 23 respectively.
When the cause of the interrupt is an LDERR_n condition then bit 19 of this register is set to '1' and bit 18 indicates whether the link error occurred during the transfer of a TDC data word (Bit 18 = '0') or whether the link error occurred during the transfer of another word (Separator, NoData or other data word etc. Bit 18 = '1').
If bit 18 = '0' then bits 17 down to 0 yields a buffer memory address. On this buffer memory address, the TDC data word is stored which contains an error indicated by the LDERR_n bit of the S-LINK. Note that:

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

(See also "SHARC MS2 address space: Buffer memory Access" below).
If bit 20 of this register is a '1' then there was an overrun condition. This means that there were one or more errors that weren't read-out in time. Note that during an overrun condition, bits 18 down to 0 of the register represent the status of the first error that occurred.
The LDERR_n interrupt can be cleared by writing a value which has bit 19 set to '1' to this register.
An LDOWN_n condition can lead to an interrupt as well. The (unmasked) status of the LDOWN_n line of the S-LINK is read back in bit 21 (Note that '0' read back means link is down).
Bit 17..0: The buffer memory address onto which the TDC data word with LDERR_n set is stored (in the case where bit 18 = '0').
Bit 18: Data / Other. If this bit is '0' then this means that an LDERR_n condition occurred during the transfer of a TDC data word, bits 17 down to 0 yields a buffer memory address. When '1' then the link error occurred during the transfer of another word.
Bit 19: LDERR_n. If this bit is '1' then this means that an LDERR_n condition generated the interrupt.
Bit 20: LDERR_n Overrun bit. If this bit is '1' then this means that there were one or more LDERR_n conditions that weren't read-out in time. Note that during an overrun condition, bits 17 down to 0 of the register represent the status of the first error that occurred.

Bit 21: When read, gives the status of the LDOWN_n line of the S-LINK (Note that '0' read back means link is down). This same bit can be found in register 0x12 bit 8.
Bit 22: LDERR_n Mask Bit. When set to '1' an LDERR_n condition leads to IRQ1_n to be asserted.
Bit 23: LDOWN_n Mask Bit. When set to '1' an LDOWN_n condition leads to IRQ1_n to be asserted.

Register 0x11 (MS0 offset 0x23):
Early and Late Event-ID; IRQ2

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).
Bits 11-0: Represent the Event-ID, which caused the interrupt.
Bits 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 down to 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-ID's, the Expected Event-ID register (register 0x16) is programmed with an Event-ID, which will be valid in the near future. 'Late' interrupts will be generated until the Event-ID's 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-ID's should fall within the Expected Event-ID window.

Registers 0x12 to 0x14 (MS0 offset 0x25 to 0x29):
Control Status Registers and Test Registers

Register 0x12 is a general purpose Control Status register. It controls:
- Test mode (see also Register 0x13 and 0x14)
- Zero Suppress Mode
- S-Link User Return Lines and LDOWN# status
- S-LINK reset
- Freeze mode for the data pipeline
- Four general purpose bits for LEDs (only two bits are output on FPGA pins)

Register 0x12 contains the following bits:

Bit 0: If in test mode, taking this bit from '0' to '1' writes test data from register 0x13 and 0x14 to the MROD-In S-LINK input.

Bit 1: '0' Normal operation (S-LINK)
'1' test mode operation.

Bit 2: '0' Normal operation. FPGA read mode; default on power up.
'1' test mode operation. SHARC has Read/Write access to buffer memory.

Bit 3: Zero Suppress. Output data to the SHARC 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 MROD Header word still contains a flag bit which signals that the zero suppressed TDC was in the data-stream.

Bit [4..7]: These bits represent the status of the S-LINK User Return Lines (URL[3..0]).

Bit 8: When read gives the status of the LDOWN_n line of the S-LINK (Note that '0' read back means link is down). This same bit can be found in register 0x10 bit 21.

Bit 9: Taking this bit from '0' to '1' resets the S-LINK.

Bit 10: 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.

Bit [14..11]: Four LEDs can be connected to these bits. Writing a '1' puts on the light. These bits can be read back. Due to pin shortage on the FPGA package, only LEDs[1..0] are fed to FPGA pins.

In test mode (bit 1 = '1'), the S-LINK input is disabled and taken over by the SHARC. The SHARC can now write data as if it came from the S-LINK.
Register 0x12 controls access to the buffer memory as well. The buffer memory has two cycle-share ports.
The first port is always dedicated to the S-LINK input.
During normal operation, the second port is connected to the FPGA in such a way that the FPGA can read 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 FPGA. Therefore bit 2 should only be set for debugging purposes.
For a detailed description of the Freeze and S-LINK Reset bits (bit 9 and 10), see also the chapter "Reset Facilities".

Register 0x13 contains the 32 data bits, which would normally come from the S-LINK LD[31..0] bus.

Register 0x14 contains the following bits:

Bit 0: Status of the LCTRL_n bit, which would normally come from the S-LINK.

Bit 1: Status of the LDERR_n bit which would normally come from the S-LINK.

After power-up, the MROD operates normal (that is, no test mode and the FPGA has read access to the buffer memory, Freeze on).
To run a test, write the following data in the corresponding registers:
0x12: 0x00000002 (Enter the test mode, Freeze off)
0x13: S-LINK Data word 1
0x14: S-LINK Corresponding status LCTRL and LDERR
0x12: 0x00000003 (Write Data Word 1 by taking bit 0 from 0 to 1)
0x12: 0x00000002
0x13: S-LINK Data word 2
0x14: S-LINK Corresponding status LCTRL and LDERR
0x12: 0x00000003 (Write Data Word 2 by taking bit 0 from 0 to 1)
0x12: 0x00000002
:
0x13: S-LINK Data word N
0x14: S-LINK Corresponding status LCTRL and LDERR
0x12: 0x00000003 (Write Data Word N by taking bit 0 from 0 to 1)
0x12: 0x00000000 (Go to normal operation, exit test mode)

Registers 0x15 (MS0 offset 0x2B):
Maximum Event Size

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 0x18. A “Read-out Maximum” interrupt will be generated, to signal the SHARC that something went wrong (see Register 0x0F).
Register 0x15 will power up to a default value of 1024.
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 FIFO. This depends on the full status of the output FIFO.

Register 0x16 (MS0 offset 0x2D):
Expected Event-ID

Register 0x16 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. The hardware of the MROD-In increments the Expected Event-ID when a complete event is gathered from its TDC inputs. The register can be inspected at any time.
It should be possible to synchronize the MROD-In during a run. Via the TTC system it is know 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 0x11) should be disabled until the MROD-In is in synchronisation.

Register 0x17 (MS0 offset 0x2F):
TDC Mask Register

Register 0x17 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.

Register 0x18 (MS0 offset 0x31):
Partition Read-out Enable

Register 0x18 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 readout 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 since data should be read from the Output FIFO and the Event Length FIFO fast enough.
Note that the Partition Full condition is met when a partition contains 5 to 6K words instead of the maximum possible 8K words. This is due to the fact that the FPGA contains a pipeline to the buffer memory and only the upper 3 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:
`WritePointer[12..10] = ReadPointer[12..10] – 2`
Note the following example: when the read pointer is 0x0423 a partition full condition is met when the write pointer increases to 0x1C00, a difference less then 6K 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 to the SHARC
In either case IRQ0 is asserted (if enabled, see register 0x0F Interrupt Control IRQ0)
The SHARC can read the TDC read-out enable register (register 0x18). 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.

Registers 0x19 (MS0 offset 0x33):
Serial Register

Register 0x19 is a read only register. Only bits 0 and 1 are significant, all other bits are read back as '0'.
Register 0x19 is updated whenever a Separator word is recognised in the input data stream. Bits 0 and 1 of register 0x19 follow the values of bits 18 and 19 of the Separator word, which was received.
Note that the Separator Mask bits 18 and/or 19 should be '0' (register 0x02) in order to use the option of receiving serial data through these bits.

SHARC MS1 address space: Output FIFO

A value form the output FIFO is read when a read cycle on any address in MS1 occurs. Address line A[20] of the SHARC determines the channel to be read (see table 3).
Note that reading from an empty output FIFO will return the value of the last word which was read before the FIFO went empty. Normally this will never occur since the Output FIFO will be readout using DMA, in such a mode that a DMA request is send by the FPGA for each word in the Output FIFO.

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 0x12) is '1' then the SHARC gets Read/Write access to the buffer memory. Be carefull, 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").
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[19..1] are re-routed to Buffer Memory address lines A[18..0] while buffer memory address line A[19] is always '0' during an access by the SHARC. The SHARC can only address up to 2 MB of memory instead of the full 4 MB.
Note that SHARC address line A[20] is being used to select either Channel A or Channel B.

SHARC MS3 address space: Not Used

MS3 address space is not used on the MROD-In board.

SHARC Flags:

Table 6 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 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 14-21 "Flag Outputs" of the ADSP-21160 Technical Specification.
Flag 2 and 3 must be configured as inputs and are connected to the Empty output of the 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 Event Length FIFO of Channel A
3	Input	Empty status of Event Length FIFO of Channel B

Table 6: SHARC Flags

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 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, or a TDC Parity Error was received (see register 0x0F).
IRQ1 is asserted when an S-LINK error (LDERR_n) occurs or when the S-LINK goes down (LDOWN_n), see register 0x10.
IRQ2 is asserted whenever a TDC trailer is received with an Event-ID, which is out of the Expected Event-ID window ('Early' or 'Late', see register 0x11 and table 1).

Reset Facilities

There are three reset signals: “POR” (Power-Up Reset), “SYSRESET_n” and “Rst_n”.
A Power-Up Reset or an asserted “SYSRESET_n” signal resets the SHARC, starts a re-configuration of the FPGAs and resets the FPGAs (including the Pipeline, Address Generators, Tetris Register, I2O_FIFO, Output FIFO. Event Length FIFO etc.).
Asserting “Rst_n” resets the FPGAs (including the Pipeline, Address Generators, Tetris Register, I2O_FIFO, Output FIFO. Event Length FIFO etc.) but not the S-LINK and the S-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 0x12 bit 10). This means that data in the S-LINK input FIFO is not read into the data pipeline in the FPGA.
An S-LINK reset can be generated by taking bit 9 of register 0x12 from ‘0’ to ‘1’. This resets the S-LINK and the S-LINK FIFO. The S-LINK will force LDOWN# low until the initialisation phase of the S-LINK is complete. LDOWN# will then go high again. The S-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 S-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 S-LINK FIFO, which might contain garbage. The SHARC resets the S-LINK(s) by taking bit 9 of register 0x12 from ‘0’ to ‘1’. The S-LINK is reset and the S-LINK FIFO in the FPGA is cleared. The Freeze condition can now be taken away (clear bit 10 of register 0x12).