MROD-Out

Introduction

The output part of the MROD module consists of a cluster of two output SHARCs. This cluster of SHARCs has three basic functions:

1. Drive an S-LINK
2. Connect to VMEbus
3. Receive TTC information.

Programmers viewpoint VMEbus

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

SHARCs:	A32: D32 RMW, BLT, Single Cycle as either Supervisory or Non Privileged and as either Program or Data access.
CR/CSR:	A24: D32, D16, D08(EO), D08(O) RMW, Single Cycle.
IRQ:	I(N) D08(O) ROAK where N is a selectable IRQ Level [1..7]
AM 0x10:	User-Defined AM.

SHARCs

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

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

Seen from the VME side this corresponds with the address range 0x00000000 to 0x003FFFFC (bits A0 and A1 are omitted since VMEbus is byte oriented).

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

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

Table 1: SHARC cluster memory map in VMEbus address space

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

CR/CSR

The modules CR/CSR is accessed through A24, CR/CSR (AM Code 0x2F) cycles.
The module accepts accesses tot the range of addresses belonging to the VME64x defined CR space (0x00000-0x00FFF) and the VME64x defined CSR space (0x7FC00-0x7FFFF). See the VME64x specification Chapter 10.
The full defined CR space is implemented in ROM (1024 bytes). The memory is currently filled with a VME64 CR format, which covers only 32 bytes (see VME64, Table 2-32).
Note that the CR area is read only. Writing to CR space causes a BERR_n to be asserted.

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

Offset 0x7FFFF: BAR
Upon reset (VME SYSRESET) the, module detects if it is put into a VME or VME64 back plane. This is detected because the geographical address pins in such crates are open. This means they will be pulled high on the board so the slot number will be zero with a parity error (VME64x, Observation 3.6). When this is the case, BAR[7..3] is loaded with the value of a LocalBAR jumpers.
When the module detects it is plugged into a VME64x back plane then BAR[7..3] is loaded with the inverted value of the geographical address pins (GA_n[4..0]) which corresponds to the slot number. However, when a parity error is detected, BAR[7..3] is set to the “Amnesia address” 0x1E (VME64x, Recommendation 3.8).
BAR[7..3] can be read and written by a VME A24 cycle to CR/CSR (AM Code 0x2F). Note that BAR[7..3] correspond to Address bits A[23..19] that are used to decode the boards CR/CSR in the A24 address space, so by writing the BAR the Base Address of this boards CR/CSR changes!
BAR[2..0] are not used.

Offset 0x7FFFB: Bit Set Register
Writing bits ‘1’ in this register ‘Sets’ the corresponding bits [7..0] of the Bit Set Register. Writing a bit ‘0’ has no effect to the corresponding bit. Reading the register returns the status of the bits. Note that reading the Bit Set Register yields the same result as reading the Bit Clear Register. See also VME64, Table 2-31 and VME64x, Table 10-6.
The following bits are implemented:
Bit 7: Put the board in Reset mode.

This causes the reset pin of the SHARCs to be asserted for 375 ns. Note that the internal CSR-Reset signal is set by writing a '1' to Bit 7 of the Bet Set Register. A '0' to '1' transition of the internal CSR-Reset signal generates a SHARC reset pulse. The SHARCs cannot be held in reset since they contain the arbiter for the host bus. The VMEbus needs to use the host bus when it wants to do a transfer (for example clear the internal CSR-Reset signal by writing to the Bit Clear Register).
Software (VME) determines the status of the internal CSR-Reset signal.

Bit 6: SYSFAIL driver enable.
This causes a '1' on the SYSFAIL_En pin. Software determines the status of this pin. The pin is currently not used.

Bit 5: Module Failed.
Hardware determines the status of this bit. Software can set this bit for test purposes (ModuleFail is currently hardwired FALSE in the FPGA).
Note that the definition for testing this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit 4: Module Enable.
This causes a '1' on the Module_En pin. Software determines the status of this pin. The pin is currently not used. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit 3: Module issued BERR_n.
Hardware determines the status of this bit. This bit is set whenever the module issued a BERR_n. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit 2: CRAM Owned.
Hardware determines the status of this bit. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard. Currently not implemented.

Bit 1: Reserved

Bit 0: Reserved

Offset 0x7FFF7: Bit Clear Register

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

The following bits are implemented:

Bit 7: Get the board out of Reset mode.
Clears the internal CSR-Reset signal. Note that the reset pin of the SHARCs is asserted for 375 ns by a '0' to '1' transition on the internal CSR-Reset signal. The SHARCs cannot be held in reset since they contain the arbiter for the host bus (see also Bit Set Register above).
Software (VME) determines the status of the internal CSR-Reset signal.

Bit 6: SYSFAIL driver disable.
This causes a '0' on the SYSFAIL_En pin. Software determines the status of this pin. The pin is currently not used.

Bit 5: Module Failed.
This clears the module Fail flag. Hardware determines the status of this bit (ModuleFail is currently hardwired FALSE in the FPGA). Software can clear the bit. Note that the definition for testing this bit is not in the VME64 standard, but it is in the VME64x standard.

Bit 4: Module Enable.
This cases a '0' on the Module_En pin. Software determines the status of this pin. The pin is currently not used. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit 3: Clear BERR_n Flag.
Hardware determines the status of this bit. This bit is set whenever the module issued a BERR_n. Software can clear this flag. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard.

Bit 2: CRAM Owned.
Hardware determines the status of this bit. Note that this bit is only defined in the VME64x standard, not in the VME64 Standard. Currently not implemented.

Bit 1: Reserved

Bit 0: Reserved

Interrupt VMEbus

The module can issue a VME interrupt request. The interrupt priority level can be set under control of the SHARCs to any of the levels 7 to 1.
When the interrupt is acknowledged by an interrupt acknowledge cycle, the module puts an 8-bit status/ID on the VMEbus. The 8-bit Status/ID is programmable.
The module releases the VME IRQ line when the interrupt is acknowledged (Release On Acknowledge, ROAK).
VMEbus Mnemonic: I(N) D08(O) ROAK where N = 1..7
The Interrupt is issued using the "VMEbus BAR and IRQ" register (0x01). The StatusID is programmed in the "VMEbus IRQ Level and IRQ StatusID Pattern" register (0x00).
For a description of both registers see "Programmers viewpoint: SHARC"

AM10

The User-Defined AM10 is used as an ultimate way to reset the SHARCs.
The data bus on the MROD-Out board is shared by the VME-bus and the SHARCs. Therefore the data bus is arbitrated by the SHARCs using the arbiter of the host bus Interface. If, for any reason, one of the SHARCs ends-up in a state where its host bus arbiter is out of order, then the VME-bus is not accessible since the arbiter will never grant the data bus. This causes a hang-up situation because the "Put the board in Reset mode" bit of the Bit Set Register in CR/CSR space is not accessible anymore via the VME-bus.
In such a case, the SHARCs can be reset when a VME-bus cycle is generated with AM10 on the Base Address of the module (see table 2).

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

Table 2: AM10 Reset Address Decoding

Programmers viewpoint SHARC

SHARC cluster memory regions:

The SHARC uses a 64-bit data bus to interface to external memory. The data bus of the FPGA is connected to bits 63 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	MROD-Out internal registers	100	00	1, 2
MS1	Read TTC Event/Bunch-ID and Trigger-Type	001	00	1,3
MS2				1
MS3	S-LINK Interface	000	00	1, 4

Table 3: SHARC cluster memory regions

Remarks:

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

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

The MROD-Out SHARC cluster is connected to:

- VMEbus through a VME interface (See: "Programmers viewpoint VMEbus")
- S-LINK output (See also MS3 address space)
- TTC (Timing Trigger and Control) interface

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

Register	MS0 offset	Function	Write	Read
`0x00`	`0x01`	VMEbus IRQ Level and IRQ StatusID Pattern	`0x-----snn`	`0x00000snn`
`0x01`	`0x03`	VMEbus BAR and IRQ	`0x-------p`	`0xpns000nt`
`0x02`	`0x05`	S-LINK Status and Interrupt Register	`0x------p-`	`0x000000rn`
`0x03`	`0x07`	TTC Control/Status and Interrupt Register	`0x-------p`	`0x0000000p`
`0x04`	`0x09`	Resets and LEDs	`0x------nn`	`0x000000nn`

Remarks:
'n' is any hexadecimal number 'r' is 0x00, 0x01, 0x04 or 0x05
'-' is don't care 's' is one out of 0x00 to 0x07
'p' is either 0x00 or 0x01
't' is one out of 0x00, 0x02, 0x04, 0x06, 0x08, 0x0A, 0x0C, 0x0E

Table 4: Registers in the MS0 address space

Register 0x00 (MS0 offset 0x01):
VMEbus IRQ Level and IRQ StatusID Pattern

Register 0x00 Bits [7..0] can be programmed by the MROD-Out SHARCs with an 8-bit StatusID pattern which is put onto the VMEbus when an interrupt acknowledge cycle is accepted (See VME64 Chapter 4).
Bits [10..8] determine to which VMEbus IRQ level the interrupt will be routed (see table 5) when it is triggered through the VMEbus BAR and IRQ register 0x01. Note that "000" will map to IRQ(7) which is the highest level.

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

Table 5: VMEbus IRQ Level select

Register 0x01 (MS0 offset 0x03): VMEbus BAR and IRQ

The MROD-Out SHARCs can assert an interrupt on the VMEbus when it writes a ‘1’ into bit 0 of this register (see tabel 6). The interrupt is routed to one of the seven VMEbus interrupt lines IRQ[7..1] according to the IRQ Level Select Bits in register 0x00. No interrupt will be triggered when there is an interrupt pending on any of the levels.
Reading register 0x01 gives information on the VME Base Address Register and any pending interrupt.
Bits[7..1] correspond to the IRQ[7..1] lines of the VMEbus. When one of these bits is read as ‘1’ then the corresponding IRQ level is pending. These bits are cleared by the hardware when the interrupt is acknowledged by VMEbus (See also register 0x00: VMEbus IRQ Level and IRQ StatusID Pattern).
Bits[23..19] give information about the VME selected base address. The Base Address Regiser (BAR) is loaded upon a Sysem Reset. Bits BAR[7:3] are read back in register 0x01 on bits [23:19].
Bit[24] determines whether the base address originated from the geographical address pins (when the module is plugged into a VME64x Backplane) or from the LocalBAR jumpers on board (when the module is plugged into an ‘old fashion’ backplane).
All other bits of register 0x01 are read back as ‘0’.

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

Table 6: VMEbus BAR and IRQ Register

Register 0x02 (MS0 offset 0x05): S-LINK Status and Interrupt Register

The S-LINK Link Return Lines LRL[3..0] and the status of the LDOWN# signal can be read through register 0x02.
An interrupt (IRQ0A) can be generated on SHARC A whenever the status of the Link Return Lines changes or when LDOWN_n is asserted. Both interrupt sources can be masked.

Data	Description	Write	Read
3..0	S-LINK LRL bits	X	S-LINK LRL(3..0)
4	Enable/Disable LRL Change Interrupt (SHARC IRQ0A)	'0': Disable Interrupt '1': Enable Interrupt	'0': Interrupt Disabled '1': Interrupt Enabled
5	Clear LRL Change Interrupt	'0': No action '1': Clear Interrupt	'0': No IRQ0A Pending '1': IRQ0A Pending
6	Enable/Disable LDOWN Interrupt	'0': Disable Interrupt '1': Enable Interrupt	'0': Interrupt Disabled '1': Interrupt Enabled
7	Clear LDOWN Interrupt or Read status	'0': No action '1': Clear Interrupt	'0': S-LINK LDOWN_n asserted
31..8	X	X	0x00

Table 7: S-LINK Status and Interrupt Register

The data format of the S-LINK Status and Interrupt Register is described in table 7. Note that bit 7 will read a '0' when LDOWN_n is asserted because LDOWN_n is a low active signal. Bit 7 will read the status op LDOWN_n independently of the mask bit for the interrupt (bit 6).

Register 0x03 (MS0 offset 0x07):
TTC Control/Status and Interrupt Register

The status of the Event/Bunch-ID and Trigger-Type FIFO can be monitored in register 0x03 (see table 8).
When the Event/Bunch-ID FIFO becomes full then an interrupt (IRQ2A) can be generated on SHARC A. The full interrupt status is indicated by bit 0 being a '1'. The interrupt can be cleared when a '1' is written to bit 0. The Event/Bunch-ID Full interrupt can be masked when bit 2 is '0'. Bit 4 of register 0x03 is the empty status of the Event/Bunch-ID FIFO.
When the Trigger-Type FIFO becomes full then an interrupt (IRQ2A) can be generated on SHARC A. The full interrupt status is indicated by bit 1 being a '1'. The interrupt can be cleared when a '1' is written to bit 1. The Trigger-Type Full interrupt can be masked when bit 3 is '0'. Bit 5 of register 0x03 is the empty status of the Trigger-Type FIFO.
Note that bit 0 and 1 indicate the source of the interrupt (which can be masked). They do not directly indicate the Full status of the Event/Bunch-ID and Trigger-Type FIFOs. Bits 4 and 5 however, directly indicate the empty status of the FIFOs.

Data	Description	Write	Read
0	Event/Bunch-ID FIFO Full	‘0’: No action ‘1’: Clear Interrupt	‘0’: not Full (or masked) ‘1’: Full
1	Trigger-Type FIFO Full	‘0’: No action ‘1’: Clear Interrupt	‘0’: not Full (or masked) ‘1’: Full
2	Event/Bunch-ID FIFO Mask	‘0’: Disabled ‘1’: Enabled	‘0’: Disabled ‘1’: Enabled
3	Trigger-Type FIFO Mask	‘0’: Disabled ‘1’: Enabled	‘0’: Disabled ‘1’: Enabled
4	Event/Bunch-ID FIFO Empty Status	X	‘0’: not Empty ‘1’: Empty
5	Trigger-Type FIFO Empty Status	X	‘0’: not Empty ‘1’: Empty
6	Flush Event/Bunch-ID And Trigger-Type FIFOs (If Selected)	‘0’: No action ‘1’: Flush	‘0’
7	Event/Bunch-ID And Trigger-Type FIFOs Flush Selection	‘0’: ECR_n (Default) ‘1’: Software Flush	‘0’: ECR_n (Default) ‘1’: Software Flush
31..8	X	X	0

Table 8: TTC Control/Status and Interrupt Register

Note also that the both DMA controllers for the Event/Bunch-ID and Trigger-Type FIFOs read ahead the first word in the output register of the FIFO. The Event/Bunch-ID FIFO is a Paged FIFO. Reading one word ahead is not enough to read a whole page so this FIFO will set its full flag after writing 128 words to it. The Trigger-Type FIFO however, will set its full flag after writing 129 words (128 words in the FIFO and one in the output register).
The Event/Bunch-ID and Trigger-Type FIFOs are flushed when an Event Counter Reset (ECR) occurs. If bit 7 is ‘0’, which is the default value, then this is done directly by the TTC ECR signal. When bit 7 is set to ‘1’ then the Event/Bunch-ID and Trigger-Type FIFOs can be flushed by the software. This gives the software the opportunity to verify that the FIFOs are empty since the ECR should only occur when the Data Acquisition pipeline is empty. Flushing the FIFOs under software control is done by writing to register 0x03 with bit 6 set to ‘1’. Note that the software is aware of the ECR event through its IRQ1A line (see “SHARC Interrupts”).

Register 0x04 (MS0 offset 0x09):
Resets and LEDs

Bits [2..0] of register 0x04 control the reset signals to the MROD-In. These reset signals (Rst0_n, Rst1_n and Rst_2n) are active low. The default value for these signals is ‘1’. If the SHARC on one of the MROD_In parts should be reset then the corresponding bit

[2..0] of register 0x04 should be written ‘0’ to assert the reset line. Be sure to write back ‘1’ to the bit to de-assert the reset line again.
The S-LINK can be reset when bit 3 of this register is taken from ‘0’ to ‘1`. This will initiate a S-LINK reset sequence. The URESET# signal of the S-LINK interface will be asserted. The S-LINK will assert LDOWN#. URESET# will be kept asserted until LDOWN# is de-asserted by the S-LINK. Bit 3 is reads back the value that it was written to.
Four LEDs are connected to bits [7..4]. Writing a ‘1’ puts on the light.
All bits of register 0x04 can be read back.

Data	Description	Write	Read
0	Rst0_n	‘0’: Reset ‘1’: Normal Mode	‘0’: Reset ‘1’: Normal Mode
1	Rst1_n		
2	Rst2_n		
3	S-LINK Output	‘0’ -> ‘1’: Trigger S-LINK Reset Sequence	Read back last value written
4	LED0 (Red)	‘0’: LED Off ‘1’: LED On	‘0’: LED Off ‘1’: LED On
5	LED1 (Red)		
6	LED2 (Green)		
7	LED3 (Green)		
31..8	X	X	0x00

Table 9: Resets and LEDs Register

MS1 address space of the SHARC cluster: TTC information

The Timing Trigger and Control (TTC) information is distributed through the VME64x carte on 8 back plane lines TTC[7..0] (see Table 10). The MROD-Out receives the serial bit streams of signals TTC4_n and TTC5_n. The data values within SerialID and SerialTT are sent with the least significant bit first. The TTC signals are low active so the start bit has a value of '0' (high), which is guaranteed to be preceded by a '1' (low).

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

Table 10: Content of a group of 2 words as supplied by the Paged FIFO

The Event-ID and Bunch-ID from TTC4_n are stored in a Paged FIFO with 128 pages of 2 words. For readout the Paged FIFO behaves as a normal FIFO. The data read out is organized in groups of 2 words, as described in table 11.

Word	Description	Bit[31..25]	Bit[24]	Bit[23..12]	Bit[11..0]
0	Event-ID	0x000	'0'	0xnnn	0xnnn
1	Bunch-ID	0x000	'1'	0x000	0xnnn

Table 11: Content of a group of 2 words as supplied by the Paged FIFO

Bit 24 of the data determines whether the word which is read from the Paged FIFO is a Event-ID (Bits[23..0]) or a Bunch-ID (Bits[11..0]).

Whenever there are words available in the Event/Bunch-ID FIFO for read out a DMAR1_n is send to the SHARC cluster for each word. The SHARCs can then read the word from the Paged FIFO with a read cycle in the MS1 address space on any address which has A[1,0] = "01" (see Table 12).

The Trigger-Type from TTC5_n is stored in a normal 128 deep FIFO. Whenever there are words available in the Trigger-Type FIFO for read out a DMAR2_n is send to the SHARC cluster. The SHARCs can then read the word from the FIFO with a read cycle in the MS1 address space on any address which has A[1,0]= "11" (See Table 12).

SHARC A[1,0] (MS1 offset)		Description
“01”	0x01, 0x05, 0x09 etc.	Read Event/Bunch-ID
“11”	0x03, 0x07, 0x0B etc.	Read Trigger-Type

Table 12: Addresses in SHARC cluster MS1 address space

DMA Requests can be pipelined which means that the FPGA doesn’t wait for the read cycle that is the result of the DMA request. There can be a maximum of 4 pending DMA requests.
The empty status of the Event/Bunch-ID FIFO and the Trigger-Type FIFO is available when reading the TTC Control/Status and Interrupt Register (see register 0x03).
Normally it shouldn’t be necessary to read the empty status since no DMA request is send when the FIFOs are empty.
The Event/Bunch-ID FIFO and the Trigger-Type FIFO are read only, writing to address space MS1 has no effect.
When the TTC system asserts the Event Counter Reset (ECR corresponds to TTC1_n) then the content of both the Event/Bunch-ID FIFO and the Trigger-Type FIFO are flushed if the “Event/Bunch-ID And Trigger-Type FIFOs Flush Selection” bit in register 0x03 is set to ECR_n flush (bit = ‘0’).

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

The SHARC cluster can send data to the output S-LINK by writing data to address space MS3. The S_LINK output interface acknowledges each write cycle. When there is an S-LINK output FIFO full condition then the SHARC cluster write cycles to MS3 address space will no longer be acknowledged.
The purpose of the S-LINK output FIFO is to avoid a hang-up situation on the external bus of the SHARC cluster when a write cycle to the S-LINK is not Acknowledged because of an S-LINK Full (LFF_n) or a S-LINK Down (LDOWN_n) condition. This hang-up situation would extend as long as either LFF_n or LDOWN_n is asserted. A FIFO of 256 words is placed in the S-LINK Interface of the FPGA in between the SHARC cluster and the S-LINK. SHARC A or B can now verify (by reading Flag 1, see SHARC Flags) that it can write at least 256 words without the danger of a hang-up situation. Note that in fact 257 words can be transferred because of a pipeline register in front of the 256 word FIFO.
Data in the FIFO is send continuously (at 160 MB/sec) to the S-LINK output provided that the S-LINK is not down and the link is not full (LDOWN_n and LFF_n both de-asserted). Therefore the FIFO will normally be (nearly) empty since the maximum data transfer rate of the SHARCs to the FIFO in the FPGA does not exceed 160MB/sec.
When the S-LINK goes down or the link becomes full then eventually the FIFO in the FPGA can become full. When this is the case, a new write cycle from the SHARC cluster is not acknowledged so this cycle will be extended until the FIFO is not full.

For all addresses in the MS3 address space which have A[1,0] = “01” (0x01, 0x05, 0x09 etc.), the SHARC cluster writes S-LINK *DATA* words (S-LINK control line LCTRL# = ‘1’).
For all addresses in the MS3 address space which have A[1,0] = “11” (0x03, 0x07, 0x0B etc.), the SHARC cluster writes S-LINK *CONTROL* words (S-LINK control line LCTRL# = ‘0’).

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

Table 13: SHARC cluster to S-LINK data format

Remark: The S-LINK cannot be read in memory region MS3. To read the LRL(3..0) lines of the S-LINK, see the SHARC Control Status Register.

SHARC Flags:

Table 14 gives an overview of the SHARC flags.

Flag Number	Direction	Description
0 (SHARC A)	Output	ROD BUSY
1 (SHARC A)	Input	S-LINK Output FIFO empty
2 (SHARC A)	Output	S-LINK UTEST_n
3 (SHARC A)	Output	LED FLAG3a (Yellow)
0 (SHARC B)	Output	LED FLAG0b (Red)
1 (SHARC B)	Input	S-LINK Output FIFO empty (LED FLAG1b (Red))
2 (SHARC B)	Output	LED FLAG2b (Green)
3 (SHARC B)	Output	LED FLAG3b (Green)

Table 14: SHARC Flags

Flag 0A should be configured as an output and can be set by the SHARC to signal a ROD Busy condition to the Central trigger Processor via the Timing Interface Module (TIM).
Note that the SHARC Flag pins are configured as Inputs during and after a Reset. A pull-up resistor on the Flag 0 ensures that the ROD Busy signal is asserted during and after a reset. The red ‘ROD Busy’ LED indicates the status of this signal.
Flag 1A and Flag 1B should be configured as an Input. Flag 1A and 1B signal the status of the S-LINK output FIFO in the FPGA.
Flag 2A should be configured as an Output. Flag 2 is connected to the UTEST_n pin of the S-LINK Output. When Flag 2 is ‘1’, then the output S-LINK operates in the normal way, when Flag 2 is ‘0’ then the output S-LINK is put into test mode (see S-LINK specifications).
There are five LEDs connected to Flags 3A, 0B..3B. These LEDs can be used to signal various events. Setting the flags to ‘1’ will light up the LED.

SHARC Interrupts:

Table 15 gives an overview of the SHARC interrupt lines.
IRQ0A_n has two interrupt sources, which can be enabled (see S-LINK Status and Interrupt Register). First, IRQ0A_n is asserted when the LRL lines of the S-LINK change. Second, IRQ0A_n is asserted when the S-LINK signal LDOWN_n is asserted.
IRQ1A_n is directly coupled to the Event Counter Reset signal on the TTC-bus (TTC1_n).
IRQ2A_n is asserted when either the Event/Bunch-ID FIFO or the Trigger-Type FIFO is full (see TTC Control/Status and Interrupt Register).

IRQ Number	Description
0A	S-LINK
1A	Event Counter Reset (ECR)
2A	Event/Bunch-ID or Trigger-Type FIFO Full
0B	Not Used
1B	Not Used
2B	Not Used

Table 15: SHARC Interrupts