MROD-In PCB Options

S-LINK input connectors

Since the S-LINK inputs of the MROD-1 receive their power from the VME motherboard, the VCC pins on the S-LINK input connectors of the MROD-In are not used. However, they are grounded on the MROD-In board to guarantee signal integrity for the other S-LINK signals on the connector.
Pins 17 and 20 on the S-LINK input connectors of the MROD-In that are normally used for the S-LINK signals UDW0 and UDW1. The Channel A input connector carries the SysRst_n and Sharc_Rst_n signals respectively. On the Channel B input connector these pins are not connected.

Buffer memory

The ZBT memory may be extended from 1 MB to 4 MB. Different topologies are possible. Table 1 lists the usage of the buffer memory address lines when accessing the memory.

Buffer Memory Address Lines	Selection
19	'0' IC1 (= IC3 for Channel B) '1' IC2 (= IC4 for Channel B)
18	Used for 512K x 36 memories
17	Used for 256K x 36 memories
16	Used for 128K x 36 memories

Table 1: Buffer memory address lines

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"). This means that the SHARC A[0] address line is always '1' during a transfer on an odd address (and '0' for even addresses).
It is important to note the difference between the Buffer memory address lines and the SHARC address lines. 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. Therefore only IC1 (or IC3 for Channel B) can be selected. 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.

Size [MB]	IC1 (IC3)	IC2 (IC4)	SHARC Address Space Channel A (MS2 offset)	SHARC Address Space Channel B (MS2 offset)	SHARC Access Remarks
1	256K x 36		0x00000001 - 0x0007FFFF	0x00100001 - 0x0017FFFF	
1	128K x 36	128K x 36	0x00000001 - 0x0003FFFF	0x00100001 - 0x0013FFFF	First 128K only
2	512K x 36		0x00000001 - 0x000FFFFF	0x00100001 - 0x001FFFFF	
2	256K x 36	256K x 36	0x00000001 - 0x0007FFFF	0x00100001 - 0x0017FFFF	First 256K only
4	512K x 36	512K x 36	0x00000001 - 0x000FFFFF	0x00100001 - 0x001FFFFF	First 512K only

Note: that IC2 and IC4 cannot be accessed by the SHARC.

Table 2: Memory map with different memory options.

JTAG Chain

The JTAG chain includes 7 devices as listed in table 3:

JTAG no	Device	Device Function	Download File
1	EPC2	Configuration device 2 for Channel A FPGA	FPGA_1.POF
2	EPC2	Configuration device 1 for Channel A FPGA	FPGA.POF
3	EP20K200EQC240	Channel A FPGA	FPGA.SOF
4	EPC2	Configuration device 2 for Channel B FPGA	FPGA_1.POF
5	EPC2	Configuration device 1 for Channel B FPGA	FPGA.POF
6	EP20K200EQC240	Channel B FPGA	FPGA.SOF
7	ADSP-21160	SHARC-II	

Table 3: JTAG Chain

It is not completely clear which EPC2 device asserts nINIT_CONF after programming of the EPC2 device(s). Altera states, “only the *first* EPC2 has its nINIT_CONF pin tied to the FPGA” (see Table 4 of the ‘Configuration Devices for APEX & FLEX Devices Data Sheet’). It is not clear what the definition of ‘first’ is in this context.

It could mean the ‘first’ configuration device for each FPGA. These are de EPC2 devices that have their nCS pin connected to the CONF_DONE pin of the FPGA. From table 3 can be seen that this would be JTAG devices 2 and 5.
On the other hand, ‘first’ could mean the first device in the JTAG chain, which is configuration device 2 of Channel A FPGA. If this is the case then Channel B can never be serviced!
Altera doesn’t support (at least not in the Quartus software) a JTAG instruction, which can be send to any device in the chain.
For maximum flexibility the nINIT_CONF lines of the EPC2 devices are merged via resistors R11 and R12 for Channel A and R13 and R14 for Channel B (see figure 1).
It should be noted that the content of any newly programmed EPC2 always could be loaded into the FPGAs when SysRst_n is asserted.

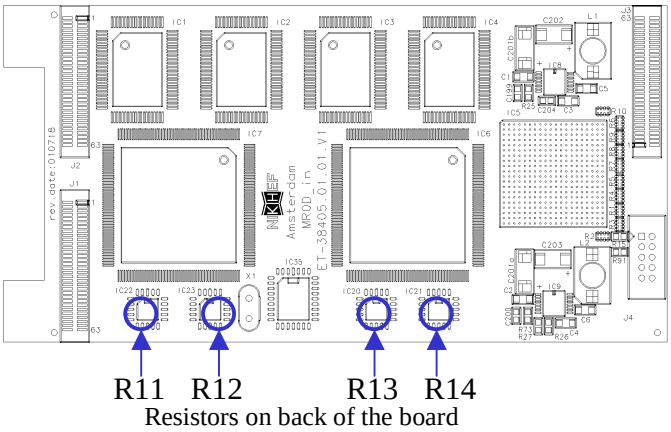

Figure 1: Location of R11, R12, R13 and R14

ByteBlaster connector J4

Connector J4 is used to connect the ByteBlaster to the MROD_In. By default, the ByteBlaster doesn’t drive the TRST_n line so the TRST_n signal is de-asserted (pulled ‘1’ by a pull-up resistor). This should be the case in order to program the EPC2 and/or the FPGA devices.
The ADSP-21160 (SHARC) datasheet states however that “TRST_n must be asserted (pulsed low) after power-up or held low for proper operation of the ADSP-21160”.
This is a conflicting situation.
To satisfy the rule for the SHARC described above, the TRST_n line of the SHARC is connected to GND via a 0 ohm resistor R17 (see figure 2).
Note that according to the JTAG standard (IEEE Std 1149.1), TRST_n is optional. The TAP controller can always be put into the “Test-Logic-Rest” state by holding TMS high while at least five rising-edged are applied to TCK.
Pin 2 of the ByteBlaster connector is not used and is left unconnected by the ByteBlaster itself.

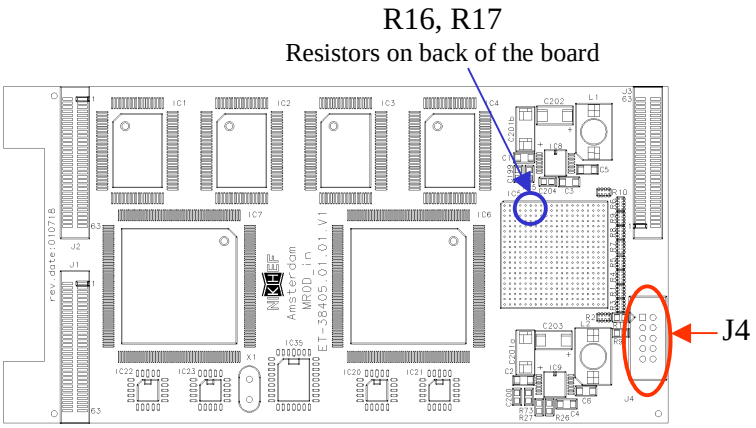

Figure 2: Location of R16, R17 and J4

EZ-ICE

When the EZ-ICE is used for simulation of the SHARC it might be necessary to drive TRST_n. If so, remove R17 and place a 0 ohm resistor at R16 (see figure 2). Note that the SHARC might not function properly after power-up when J4 is left open (See ByteBlaster connector J4 above). Make sure an EZ-ICE connector is put into J4, which is driving TRST_n in a proper way.
The EMU_n signal of the SHARC is available on pin 2 of J4 (which is not used by the ByteBlaster). When 0 ohm resistor R15 is removed, pin 2 of J4 is left unconnected on the MROD-In board (see figure 3).
Make sure that the EZ-ICE connector fits J4!

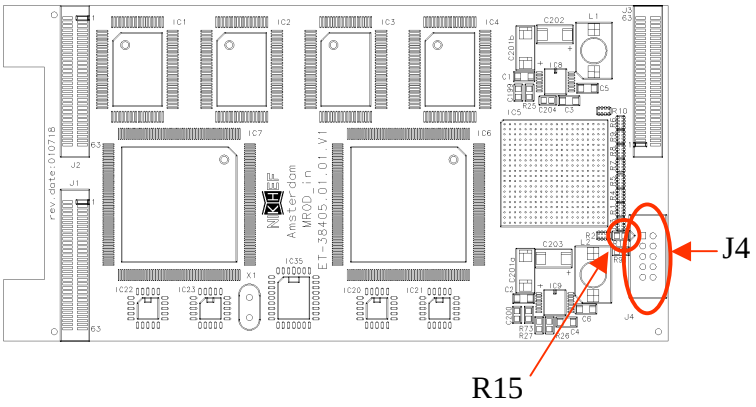

Figure 3: Location of R15