EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH
ORGANISATION EUROPEENNE POUR LA RECHERCHE NUCLEAIRE

CERN - AB Department

AB-Note-2007-030-RF

THE DSP-CARRIER BOARD USED BY THE LEIR LOW-LEVEL RF SYSTEM

USER'S MANUAL

M. E. Angoletta, CERN, Geneva

Abstract

A new digital technology to implement beam control systems was tested in the PS Booster in 2004 and 2005 and commissioned in LEIR in 2006. The technology is based upon RF custom hardware that heavily exploits Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs) processing power. This architecture is extremely flexible in that it relies on a DSP-carrier board hosting one DSP and carrying different daughtercards. The LEIR beam control system deploys three DSP-carrier boards, which inter-communicate and exchange data continuously for the implementation of the various beam control loops.

This user's manual for the DSP-carrier board, release 1.0 (EDA-00990-V1), was written in 2004 and has been used by CERN and BNL developers since then. It describes the DSP-carrier board hardware, user settings and FPGA software; hints on the DSP code used with the board are also given. An additional VME board, called Rear Transition Module, is described because it acts as a DSP-carrier board extension.

Geneva, Switzerland

June 2007

TABLE OF CONTENTS

1. INTRODUCTION

A new digital technology to implement beam control systems was tested in the PS Booster in 2004 and 2005 [1, 2] and successfully commissioned in LEIR in 2006 [3]. The technology is based upon RF custom hardware that heavily exploits Digital Signal Processors (DSPs) and Field Programmable Gate Arrays (FPGAs) processing power. This architecture is based upon a DSP-carrier board hosting one DSP and carrying up to four, 1-site each daughtercards or up to two, 2-sites each daughtercards. Fast digital data links allow different DSP-carrier board to communicate and exchange data with low latency. The architecture is extremely flexible as it is modular: in fact, many combinations of daughtercards are possible for each DSP-carrier board. Additionally, specific features are customisable by DSP or FPGA programming. As an example, the LEIR beam control system deploys three DSP-carrier boards, carrying a total of six, 2-sites each daughtercards of three different types. The DSP-carrier boards inter-communicate and exchange data continuously for the implementation of the various beam control loops.

This manual describes the DSP-carrier board hardware, user settings and FPGA software. The Rear Transition Module (RTM) is also described as it acts as a DSP-carrier board extension.

2. BOARD OVERVIEW

The DSP-carrier board release 1.0 (EDA-00990-V1) is a ten-layers, 6U VME board with a 32-bit slave interface. It uses the VME64x bus, where the P1/J1 and P2/J2 connectors include 160 pins each by the addition of ‘z’ and ‘d’ rows. In particular, the J2/P2 connector provides 46 extra user-defined I/O pins, available as user defined I/O connections through a VME64x backplane RTM. The VME64x crate is a standard LHC crate fully supported by the AB/CO group; the crate power supply must however be located on top of it to allow full access to the backplane, where the RTM is located.

Figure 1 shows a schematic view of the board and corresponding RTM. The main blocks and the bus inter-connecting them are depicted; the number of data lines per each bus is also indicated. The DSP-carrier board hosts one ADSP-21160 DSP [4], shown as a red block. The ADSP-21160 DSP has six fast digital data links called linkports. These are half-duplex, byte-serial devices clocked at 40 MHz and designed to maintain signal integrity at high frequency. Linkports 0 and 1 are routed to the DSP-carrier board front panel while linkports 2 to 6 are sent to both one daughtercard site and to the RTM. The daughtercards set the linkports as tristate when not in use, hence allowing their use from the RTM. One MB of FLASH memory and eight MB of SRAM memory, indicated as green blocks, are also available. The FLASH memory is used by the DSP when booting; the SRAM stores control parameters and I/O data.

Four daughtercard sites are available, called Site 1 to Site 4 and indicated in Figure 1 with grey boxes. The DSP-carrier board can carry either up to four, one-site each daughtercards or up to two, two-sites each daughtercards. Each daughtercard is connected to the DSP global bus, one linkport, power supplies and flexible digital control signals. The two-site daughtercards must occupy Site 1 and Site 2 or Site 3 and Site 4 only. The DSP-carrier board takes up two VME slots when carrying one or more daughtercards.

The on-board resources are managed by six FPGAs; Figure 1 shows the four most important FPGAs as blue boxes. They carry out the following functions: VME interface, interrupt generation (on the DSP chip as well as on the VME bus), memory addressing, timing interfacing and counters implementation. Six digital inputs are passed from the RTM to the Timing interface FPGA and are used as triggers as well as in the counter implementation. A BNL-specific coded timing signal called “Event link” is also made available to the Timing interface via the RTM.

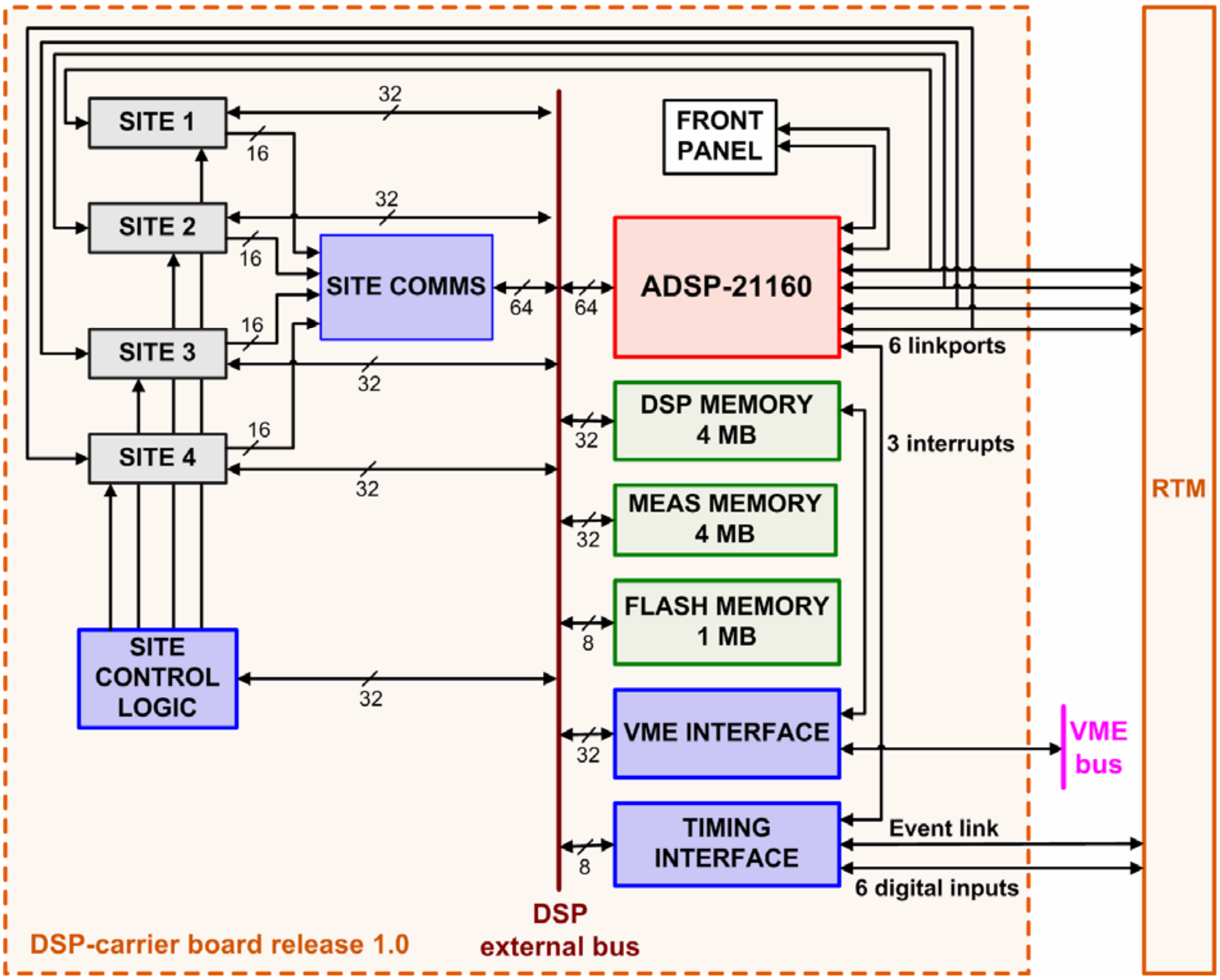


Figure 1: *Schematic view of the DSP-carrier board release 1.0 and corresponding RTM. The number of bits indicated with each bus refers to the data lines.*

Figure 2 shows the DSP-carrier board front panel. Two push-buttons allow reconfiguring some FPGAs and resetting the DSP. Two LEDs are also controllable from the DSP.

The *external synch* input allows synchronising daughtercards on different boards through a common pulse. A new tagged clock scheme has been introduced in the CERN implementation hence the *external synch* input is no longer in use.

The *external clock* signal allows clocking the DSP and the daughtercards with an external clock instead of with the 40 MHz quartz also available on the DSP-carrier board. The clock source selection is made via a DIP switch (see paragraph 3.6.2).

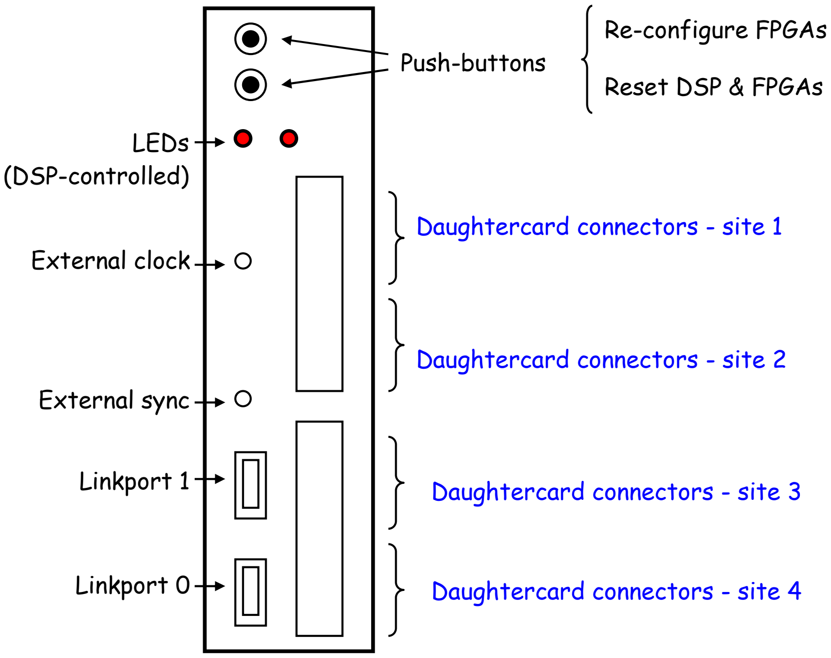

Push-buttons
Re-configure FPGAs
Reset DSP & FPGAs
LEDs
(DSP-controlled)
External clock
External sync
Linkport 1
Linkport 0
Daughtercard connectors - site 1
Daughtercard connectors - site 2
Daughtercard connectors - site 3
Daughtercard connectors - site 4

Figure 2: *DSP-carrier board front panel.*

3. HARDWARE

This paragraph outlines the characteristics of the board's main hardware components; the reader is referred to each components data sheet for more detailed information. The number each component has in the board schematic is also mentioned, typically in the paragraph heading. For instance, the DSP chip is the component U34, the FPGAs are components U1, U31, U32, U35, U36, U37 and so on. A slash "/" following a signal name or a line above it indicate that the signal is active low; examples are IRQ2/ and MS3/.

Figure 3 shows pictures of both sides of the DSP-carrier board, outlining the main components.

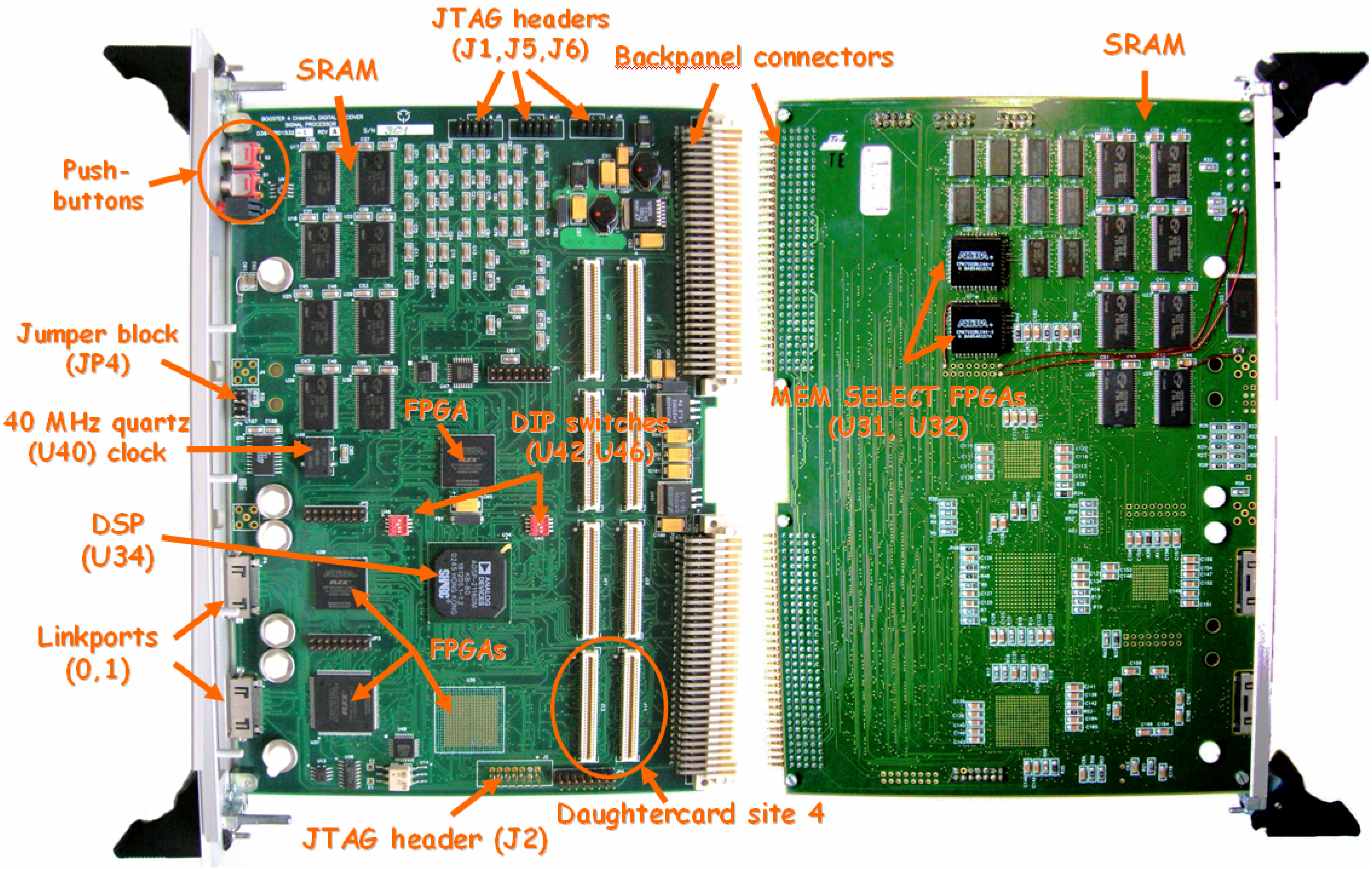


Figure 3: DSP-carrier board, front and rear view. The main board components are indicated.

3.1 DSP (U34)

The DSP-carrier board hosts Analog Devices' ADSP-21160M DSP. This is a 32-bit floating point processor implementing a “Super Harvard” architecture and capable of Single Input multiple Output (SIMD) operation. The ADSP-21160M includes an 80 MHz core, a dual-ported on-chip SRAM, an integrated I/O processor with multiprocessing support and multiple internal buses to eliminate I/O bottlenecks.

Four Mbits of on-chip, zero wait-states SRAM are available. This internal memory is organized as two blocks of 2 Mbits each, which can be configured for different combinations of code and data storage. Each memory block is dual-ported for single-cycle, independent accesses by the core processor and I/O processor. Figure 4 shows the DSP internal memory space mapping.

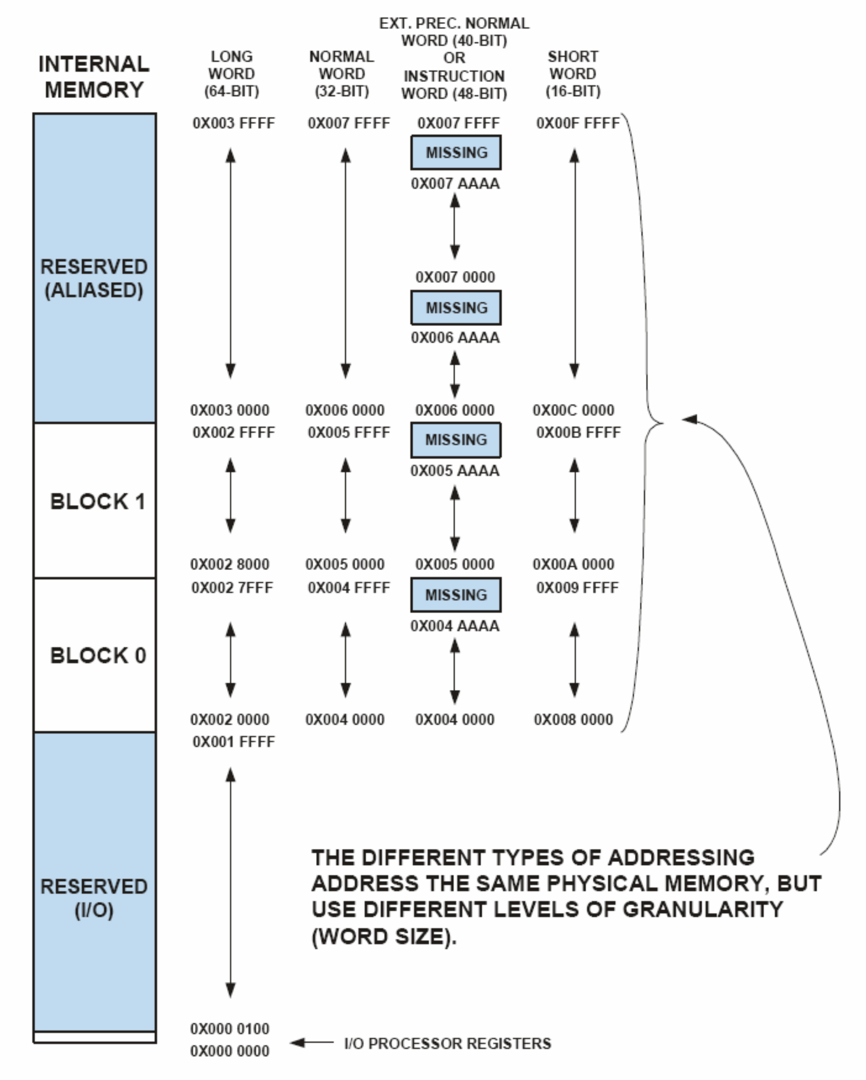


Figure 4: *ADSP21160M internal memory space.*

Three input lines (IRQ0/, IRQ1/ and IRQ2/) are available to trigger interrupts in the DSP and four flag pins (FLAG0, FLAG1, FLAG2 and FLAG3) can be configured via control bits to be either input or output.

The DSP’s external port extends the DSP’s address and data buses off-chip. Using these buses and external control lines, the DSP can interface with external memory, host processors, and other DSPs. Programmable memory wait states and external memory acknowledge controls are provided to allow interfacing to DRAM and peripherals with variable access, hold, and disable time requirements. The host interface allows easy connection to standard microprocessor buses, both 16-bit and 32-bit, with little additional hardware required. The external memory space has five regions: four banks (Bank 0-3) and an un-banked region. The DSP controls access to the banked regions with memory select lines (MS0/ to MS3/) in addition to the memory address. Access to the un-banked region is controlled only by the memory address. Each region of external memory may be configured for address range and wait-states. By mapping peripherals into different banks, systems can accommodate I/O devices with different timing requirements. Figure 5 shows the DSP external memory space, as a function of the memory select lines values; these are in fact used as chip selects for memories or other external devices, eliminating the need for external decoding logic. In the DSP-carrier board release 1.0 the external memory includes two blocks of SRAM and the daughtercards memory space.

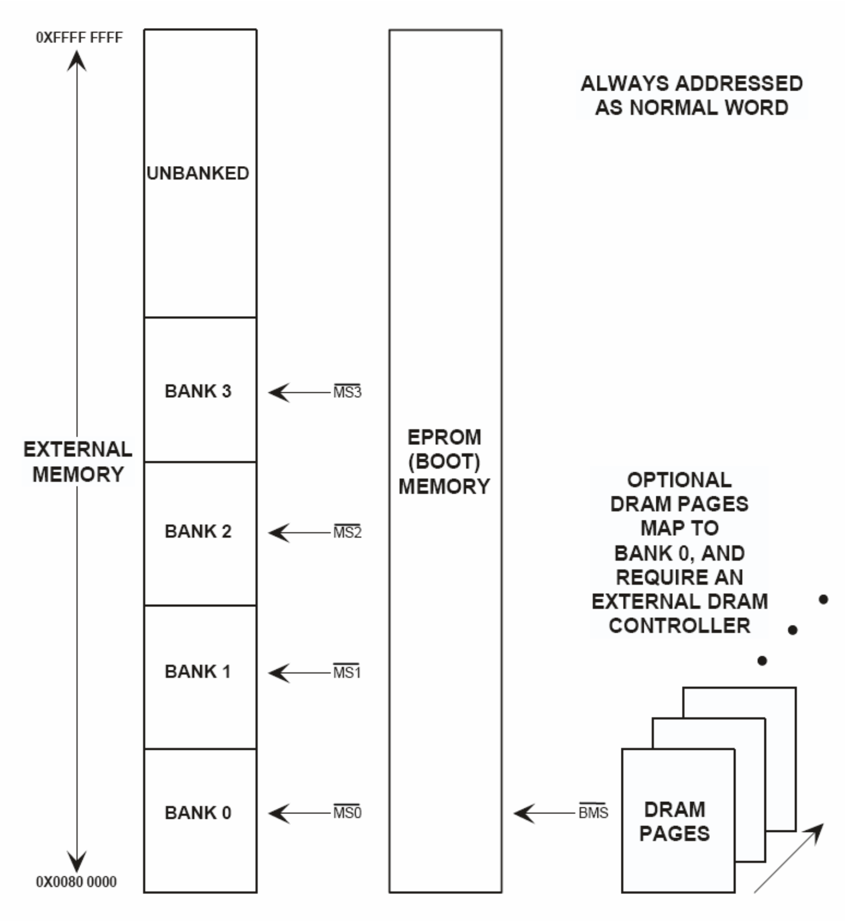


Figure 5: *ADSP21160M external memory space.*

Six, 8-bit linkports provide additional I/O capabilities, particularly useful for point-to-point inter-processor communication in multiprocessing systems. The linkports can operate independently, simultaneously and with a transfer rate of up to 80 MBytes/s each. Two synchronous serial ports are also available and provide an inexpensive interface to a wide variety of digital and mixed-signal peripheral devices.

The on-chip DMA controller allows zero-overhead data transfer without processor intervention. The DMA controller operates independently of the processor core, allowing DMA operations to occur while the core is simultaneously executing its program instructions. DMA transfers can occur between the ADSP-21160M’s internal memory and external memory, external peripherals, or a host processor. DMA transfers can also occur between the ADSP-21160M’s internal memory and its serial ports or linkports.

Shared-memory multiprocessing DSP systems can be easily implemented thanks to features such as the external port and linkports, providing integrated glue-less multiprocessing support. The ADSP-21160M DSP uses 3.3 volt I/O and a core voltage of 2.5 VDC. The core internal clock, called CLKIN, is generated by an on-chip Phase Locked Loop (PLL). Ratios of 2:1, 3:1, and 4:1 between the core and the external clock are supported.

3.2 FPGA (U1, U31, U32, U35, U36, U37)

The on-board FPGAs share tasks, data, address and control lines. Each FPGA is characterized not only by the number it has on the board schematics but also by a name referring to its function. In this document the FPGA will be referred to by their name written in italic. Figure 6 shows the FPGA location on the DSP-carrier board together with their name and IC number.

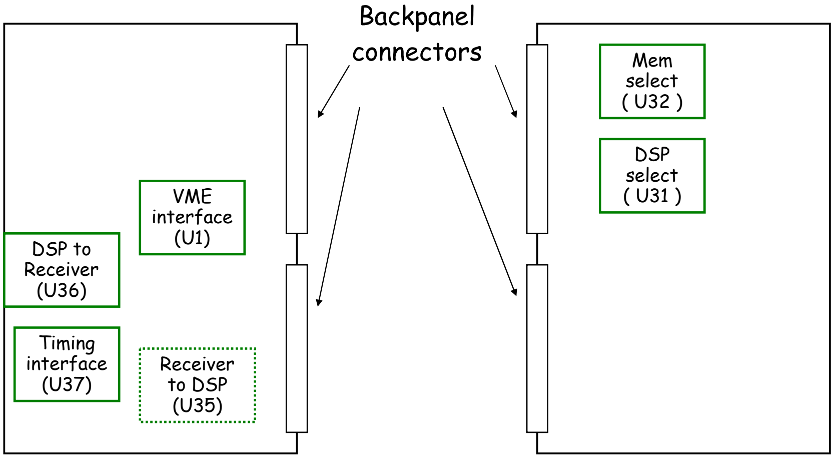


Figure 6: *Schematic view of both sides of the DSP-carrier board, showing the hosted FPGAs.*

In reality the *Receiver to DSP* (U35) FPGA cannot be physically installed on the board since its VCCINT pin is shorted to the ground plane via pin W20 on the device itself. In fact, it was discovered after the boards release 1 were manufactured and assembled that the board layout doesn't correspond to the schematics and the whole row W was displaced by one pin.

Table 1 lists the FPGAs according to the device family they belong to. The device type for *DSP Memory select* and *MEAS Memory select* FPGAs are incorrectly indicated in the board schematics as EPM7032AELC44, i.e. devices with a 3.3 V core instead of the equivalent device with a 2.5 Volt core.

FPGA NAME	COMPONENT NUMBER	DEVICE FAMILY	EXACT TYPE
VME interface	U1	FLEX	EPF10K30EFC256-1
Timing interface	U37	FLEX	EPF10K30ETC144-1
DSP to receiver	U36	FLEX	EPF10K30EFC256-1
Receiver to DSP	U35	FLEX	EPF10K30EFC484-1
DSP Memory select	U32	MAX	EPM7032BLC44-4
MEAS Memory select	U31	MAX	EPM7032BLC44-4

Table 1: *DSP-carrier board release 1.0 FPGAs name and type.*

All FLEX devices are connected by a 16-lines bus, referred to as the *emergency bus*, used to transfer missing information, when needed, to a device from another device that has access to it. More information on the emergency bus is given in paragraph 4.1.

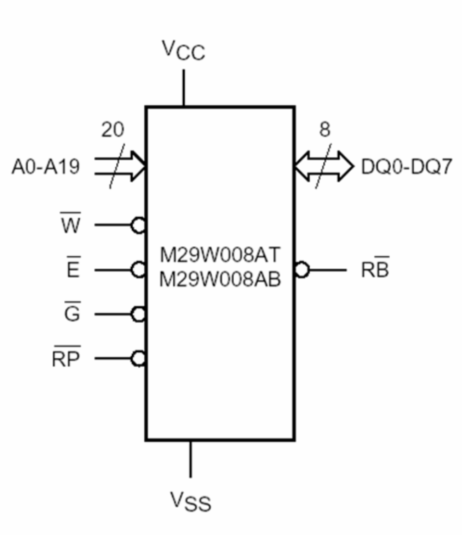
3.3 DATA MEMORY (U15 TO U30)

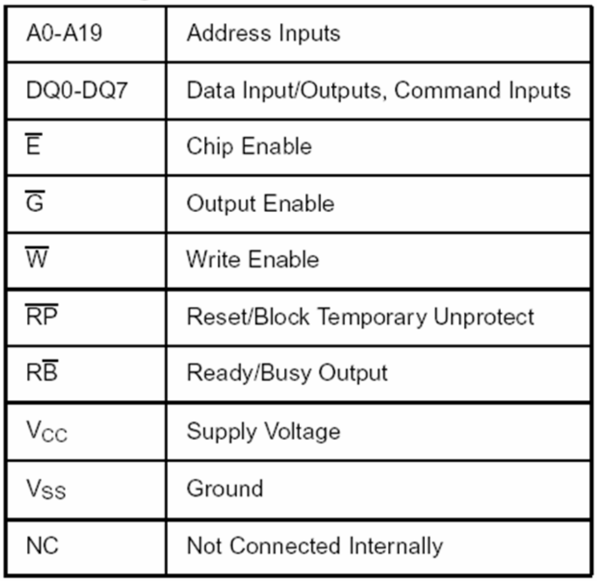
The on-board memory is implemented via sixteen 256 k by 16 bit static RAM chips (type Cypress CY7C1041-15ZC). They are organized in two separate banks of 32-bit memory as described below:

1. 4 MB DSP RAM, called DSP RAM in this document. This is a one-wait state static RAM whose access is controlled by the *DSP Memory select* FPGA.
2. 4 MB MEAS RAM. This is a one-wait state static RAM whose access is controlled by the *MEAS Memory select* FPGA. This is a one-wait state memory with two-port access.

3.4 FLASH MEMORY (U33)

The FLASH memory is an 8 Mbit (1Mb x8, Boot Block) Low Voltage Single Supply Flash Memory. The memory type is the M29W008AB. This is a non-volatile memory organised in blocks; the Boot Block is located at the bottom the memory address space. The memory can be erased electrically at block or chip levels and programmed in-system on a Byte-by-Byte basis using only a single 2.7 Volt to 3.6 Volt supply; for “program” and “erase” operations the necessary high voltages are generated internally. The array matrix organization allows each block to be erased and reprogrammed without affecting other blocks. Blocks can be protected against programming and erase-on-programming, and temporarily unprotected to make changes in the application. Each block can be programmed and erased over 100,000 times. Figure 7 shows the flash memory logic diagram and signal names.

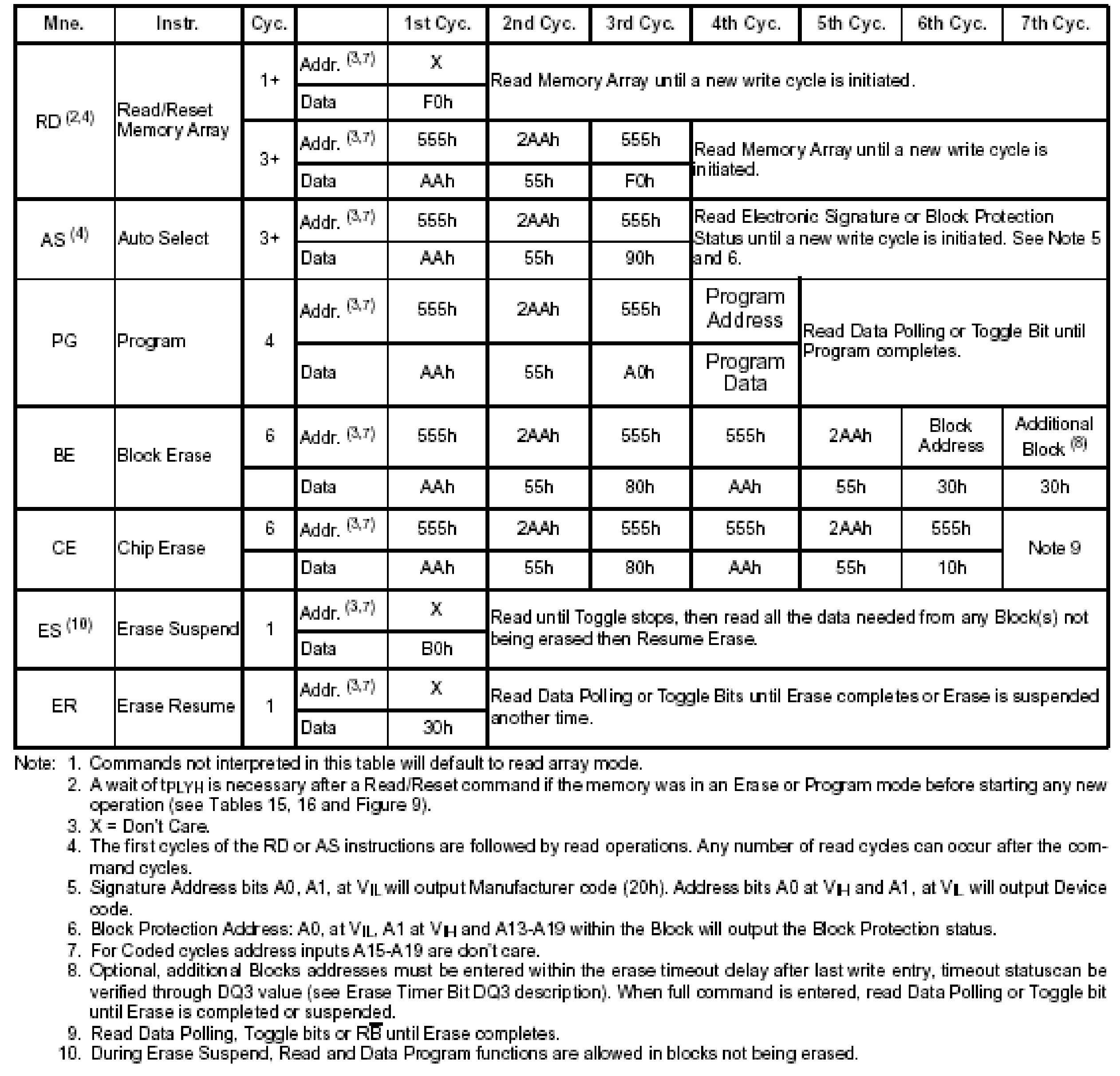




A0-A19	Address Inputs
DQ0-DQ7	Data Input/Outputs, Command Inputs
$\overline{E}$	Chip Enable
$\overline{G}$	Output Enable
$\overline{W}$	Write Enable
$\overline{RP}$	Reset/Block Temporary Unprotect
R$\overline{B}$	Ready/Busy Output
V_{CC}	Supply Voltage
V_{SS}	Ground
NC	Not Connected Internally

Figure 7 *Flash logic diagram and signal names. The flash used in the DSP carrier board is the M29W008AB (bottom-up version).*

Table 2 shows the available commands. The instructions for Program and Block or Chip Erase require further command inputs for additional data protection.



Mne.	Instr.	Cyc.		1st Cyc.	2nd Cyc.	3rd Cyc.	4th Cyc.	5th Cyc.	6th Cyc.	7th Cyc.
RD [2,4]	Read/Reset Memory Array	1+	Addr. [3,7]	X	Read Memory Array until a new write cycle is initiated.					
			Data	F0h						
		3+	Addr. [3,7]	555h	2AAh	555h	Read Memory Array until a new write cycle is initiated.			
			Data	AAh	55h	F0h				
AS [4]	Auto Select	3+	Addr. [3,7]	555h	2AAh	555h	Read Electronic Signature or Block Protection Status until a new write cycle is initiated. See Note 5 and 6.			
			Data	AAh	55h	90h				
PG	Program	4	Addr. [3,7]	555h	2AAh	555h	Program Address	Read Data Polling or Toggle Bit until Program completes.		
			Data	AAh	55h	A0h	Program Data			
BE	Block Erase	6	Addr. [3,7]	555h	2AAh	555h	555h	2AAh	Block Address	Additional Block [8]
			Data	AAh	55h	80h	AAh	55h	30h	30h
CE	Chip Erase	6	Addr. [3,7]	555h	2AAh	555h	555h	2AAh	555h	Note 9
			Data	AAh	55h	80h	AAh	55h	10h	
ES [10]	Erase Suspend	1	Addr. [3,7]	X	Read until Toggle stops, then read all the data needed from any Block(s) not being erased then Resume Erase.					
			Data	B0h						
ER	Erase Resume	1	Addr. [3,7]	X	Read Data Polling or Toggle Bits until Erase completes or Erase is suspended another time.					
			Data	30h						

Note:
1. Commands not interpreted in this table will default to read array mode.
2. A wait of t_{PLYH} is necessary after a Read/Reset command if the memory was in an Erase or Program mode before starting any new operation (see Tables 15, 16 and Figure 9).
3. X = Don't Care.
4. The first cycles of the RD or AS instructions are followed by read operations. Any number of read cycles can occur after the command cycles.
5. Signature Address bits A0, A1, at V_{IL} will output Manufacturer code (20h). Address bits A0 at V_{IH} and A1, at V_{IL} will output Device code.
6. Block Protection Address: A0, at V_{IL}, A1 at V_{IH} and A13-A19 within the Block will output the Block Protection status.
7. For Coded cycles address inputs A15-A19 are don't care.
8. Optional, additional Blocks addresses must be entered within the erase timeout delay after last write entry, timeout statuscan be verified through DQ3 value (see Erase Timer Bit DQ3 description). When full command is entered, read Data Polling or Toggle bit until Erase is completed or suspended.
9. Read Data Polling, Toggle bits or $\overline{RB}$ until Erase completes.
10. During Erase Suspend, Read and Data Program functions are allowed in blocks not being erased.

Table 2: *Flash memory instructions and commands.*

3.5 PUSH-BUTTONS (S1, S2)

The push-button S1 is the lower one in the DSP-carrier front panel. The button is connected to the *VME Interface* (U1), *Receiver to DSP* (U35), *DSP to Receiver* (U36) and *Timing interface* (U37) FPGAs CONFIG pin; when pressed, it reconfigures them.

The push-button S2 is the upper one in the DSP-carrier front panel. The push-button is connected to the RESET pin on the DSP (U34) and on the *Receiver to DSP* (U35), *DSP to Receiver* and *Timing interface* FPGAs. When pressed, it resets them.

3.6 USER-SELECTABLE SETTINGS

3.6.1 JUMPER BLOCK (JP4)

The jumper block JP4 allows selecting the clock distributed to FPGAs, DSP and daughtercards. Figure 8 shows its location on the board and its default settings.

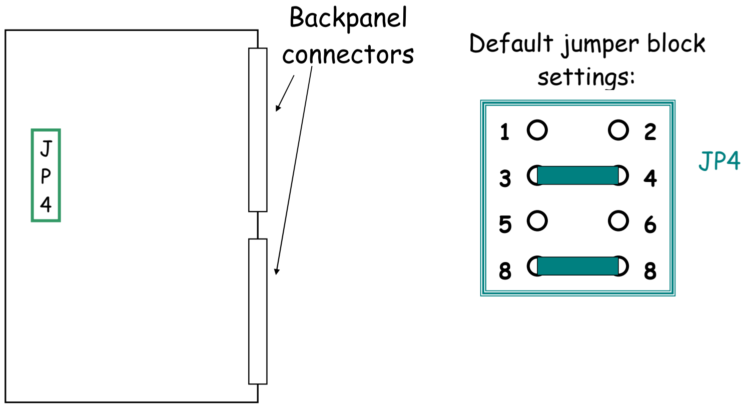


Figure 8: *Location of the jumper block JP4 on the DSP-carrier board. The default settings are shown on the right hand side.*

The jumper JP4 is connected to the fast CMOS clock distributor FCT3805 (U39). The FCT3805 allows single point-to-point transmission line driving in applications where one signal must be distributed to multiple receivers with low skew and high signal quality. It consists of two banks of drivers, each with a 1:5 fanout and its own output enable control: bank A provides clock signals to the DSP and to all FPGAs, while bank B provides clock signals to each daughtercard site. The two available clock sources are the 40 MHz quartz (U40) and the external clocks, fed to the board through the J4 input. Table 3 shows the possible jump settings.

	Pin JP4 IN	**Pin JP4 OUT**	
External clock	1	2	bank A
40 MHz quartz	3	4	
External clock	5	6	bank B
40 MHz quartz	7	8	

Table 3: *Jumper block JP4 input and output connections.*

3.6.2 DIP SWITCHES (U42, U46)

The board carries two, 4-positions, half-pitch DIP Switches (TDA04H0SK1), which select the DSP system configuration; these settings cannot be changed during DSP operation. The DIP switches inputs are grounded and their outputs are connected to different DSP input pins. Figure 9 shows their location and their default settings.

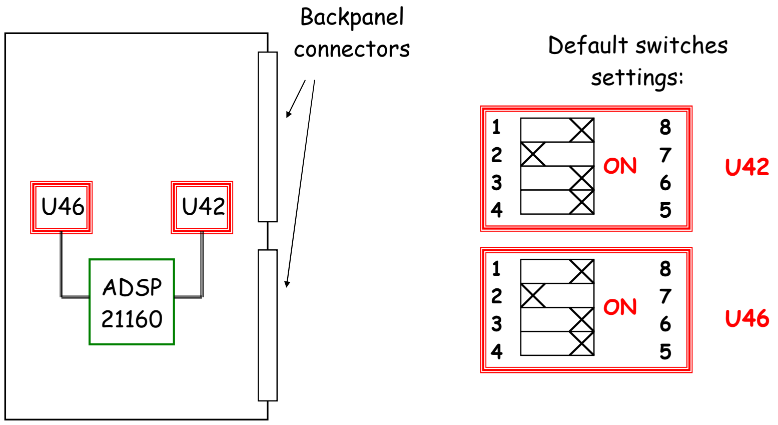
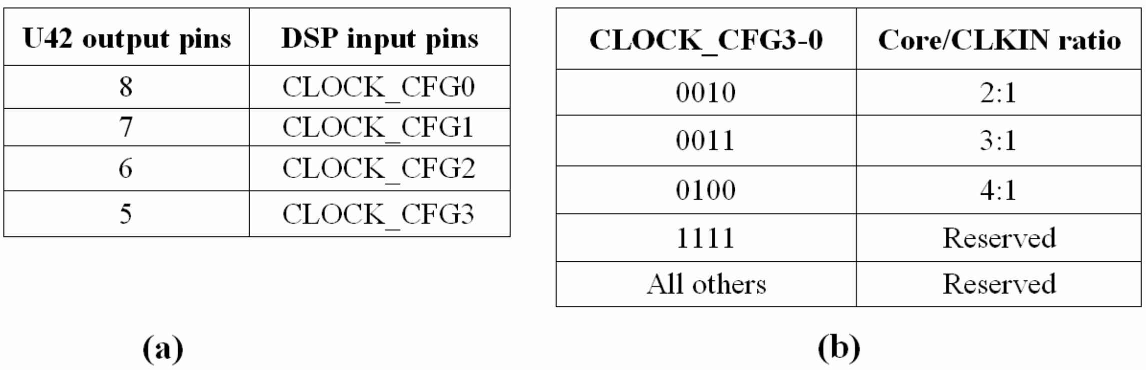


Figure 9: *Location of the U46 and U42 DIP switches on the DSP-carrier board. The default settings for the switches are shown on the right hand side.*

The U42 DIP switch selects the frequency of the DSP clock. The DSP employs an on-chip PLL, to provide clocks that switch at higher frequencies than the system clock (CLKIN). The PLL-based clocking methodology employed on the DSP influences the clock frequencies and behavior for the processor core, internal memory, serial, link, and external ports. The ratio of processor core clock frequency to CLKIN/external port clock frequency is determined by the CLK_CFG3-0 pins during reset. Table 4 shows the map between U42 DIP switch and the DSP input pins together with the Core/CLKIN ratio depending on different settings. The default settings correspond to a Core/CLKIN ratio of 2:1.



U42 output pins	DSP input pins
8	CLOCK_CFG0
7	CLOCK_CFG1
6	CLOCK_CFG2
5	CLOCK_CFG3

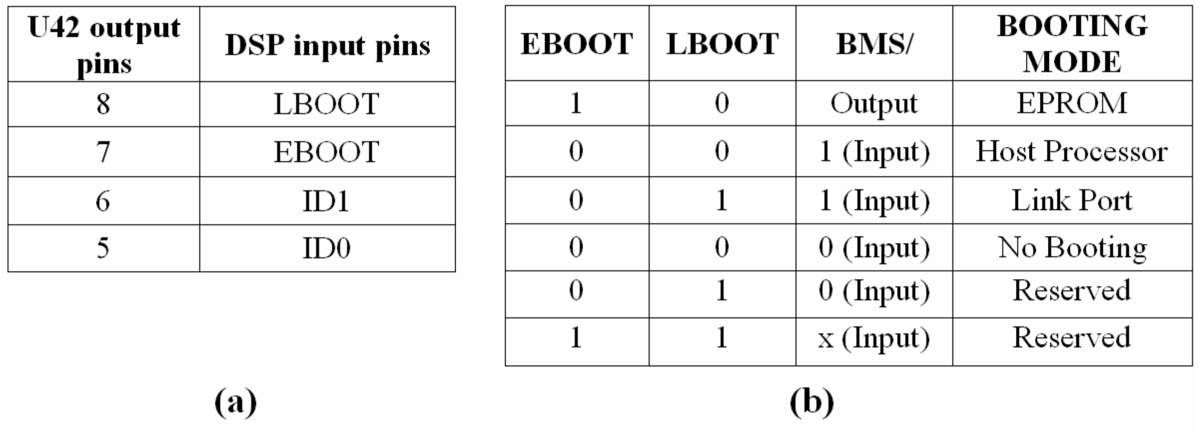
(a)

CLOCK_CFG3-0	Core/CLKIN ratio
0010	2:1
0011	3:1
0100	4:1
1111	Reserved
All others	Reserved

(b)

Table 4: *(**a**): relation between Dip Switch U42 and the DSP CLOCK_CFG input pins. (**b**): relation between the DSP CLOCK_CFG input pins and the Core/CLKIN ratio.*

The U46 DIP switch selects the DSP multiprocessor ID number and the DSP booting mode. The former determines which multiprocessing bus request (BR1-BR6) is used by each DSP; for example, ID = 001 corresponds to BR1, ID = 010 corresponds to BR2, and so on. In case of single processor systems the ID = 000 must be used. The DSP booting mode refers to the fact that programs can be automatically downloaded to the internal memory of a DSP after power-up or after a software reset. The DSP supports three booting modes: EPROM, host, and linkport. The BMS/ pin is a DSP input or output depending on the DSP booting mode selected. Table 5 shows the map between U46 DIP switch and the DSP input pins, together with the Booting mode as a function of the different settings. The DSP input ID2 is always grounded. The default settings correspond to an ID number = 000 and of EPROM boot enabled. As a consequence, the BMS/ line is a DSP output and is connected to the Flash memory.



U42 output pins	DSP input pins
8	LBOOT
7	EBOOT
6	ID1
5	ID0

(a)

EBOOT	LBOOT	BMS/	BOOTING MODE
1	0	Output	EPROM
0	0	1 (Input)	Host Processor
0	1	1 (Input)	Link Port
0	0	0 (Input)	No Booting
0	1	0 (Input)	Reserved
1	1	x (Input)	Reserved

(b)

Table 5: *(**a**): U46 DIP switch to DSP input pins map. (**b**): DSP booting as a function of EBOOT and LBOOT settings.*

3.7 JTAG (J1, J2, J5, J6) AND TEST (JP1, JP2, JP3, JP5) HEADERS

Several JTAG and test headers are installed on the board. Figure 10 shows their location on the board and their role.

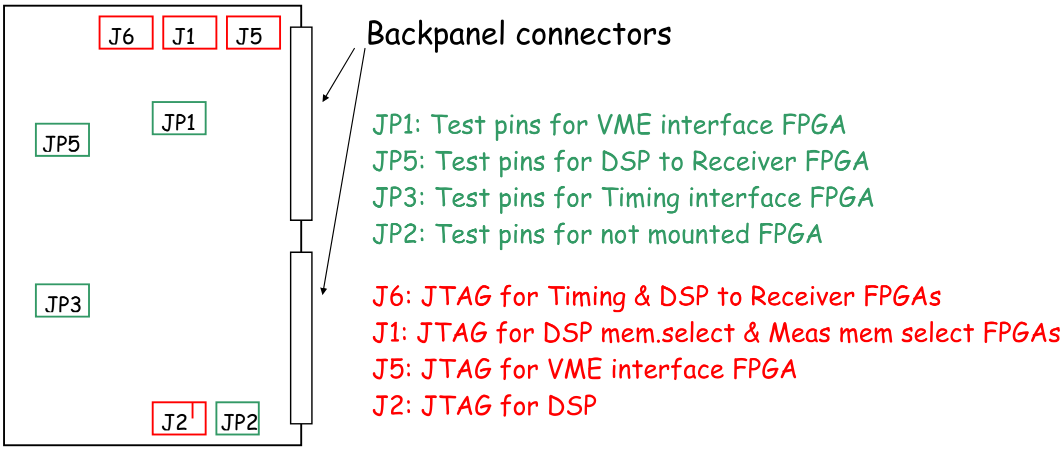

J6
J1
J5
JP1
JP5
JP3
J2
JP2
Backpanel connectors
JP1: Test pins for VME interface FPGA
JP5: Test pins for DSP to Receiver FPGA
JP3: Test pins for Timing interface FPGA
JP2: Test pins for not mounted FPGA
J6: JTAG for Timing & DSP to Receiver FPGAs
J1: JTAG for DSP mem.select & Meas mem select FPGAs
J5: JTAG for VME interface FPGA
J2: JTAG for DSP

Figure 10: *JTAG and test headers: role and location on the DSP-carrier board.*

The JTAG headers are used to load new code onto the different on-board FPGAs as well as to load code and debug the DSP. The DSP JTAG header can be distinguished from the FPGA headers by its missing pin.

The test headers are connected to FPGA I/O pins; signals that are output on them can be observed in real time via an oscilloscope.

3.8. RTM HARDWARE OVERVIEW

The RTM is the DSP-carrier board operational companion. It allows increasing the number of connections that can be made to the motherboard, it reduces clutter and it provides controlled impedance for signals. Two types of RTM are available. The one-slot RTM has been manufactured at BNL and bought by CERN for initial developments. The two-slot RTM has been designed and implemented at CERN to be used with a new and more powerful DSP-carrier board release. Figure 11 shows the front panel of both RTMs; the one-slot RTM is described in paragraph 3.8.1 while the to-slot RTM is described in paragraph 3.8.2.

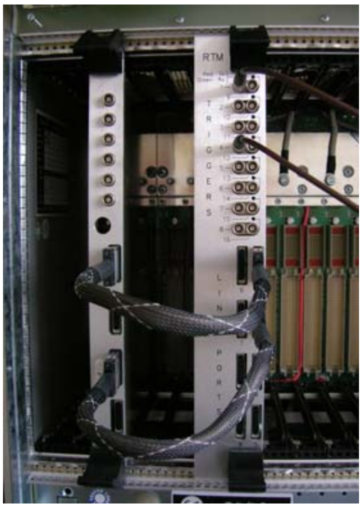


Figure 11: *RTM board, one-slot version (left hand side) and two-slot version (right hand side). Two linkport cables connecting the boards are also shown.*

3.8.1 ONE-SLOT RTM

The one-slot RTM provides 6 digital input ports, 4 linkports and one BNL event link to the DSP mother board via the VME 64x rear P2 connector. The BNL event link is a BNL-specific timing link, not used at CERN. Figure 12 shows the RTM front panel.

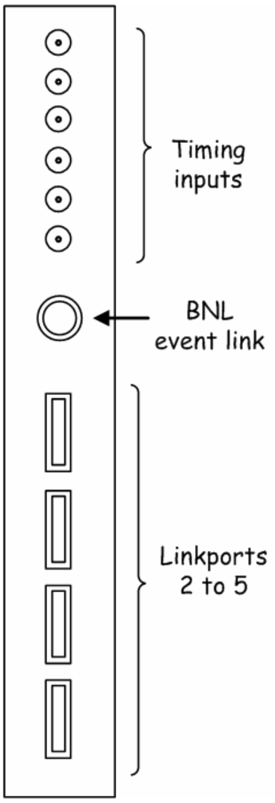


Figure 12: *One-slot RTM front panel – schematic view.*

The timing inputs are TTL signals connected directly to the *Timing interface* FPGA. These timing inputs are directly connected to the FPGA inputs hence no over-voltage should be applied, to avoid damaging the FPGA. Linkports 2 to 5 allow access to each unused linkport assigned to a daughter board. The BNL event link is a BNL-specific timing standard that allows encoding up to 256 different timings and is not used in the CERN implementation.

3.8.2 TWO-SLOT RTM

The two-slot RTM has been designed and manufactured at CERN to be used with a new release of the DSP-carrier board, not yet finalised. The RTM provides 16 digital I/O ports as well as 8 linkports to the DSP mother board through the VME 64x rear P2 connector.

The RTM consists of three boards: 1) the ‘mother board’ (EDA-01087) housing the 16 Trigger I/Os and 4 linkports, 2) a ‘daughter-card’ (EDA-01088) housing the remaining 4 linkports, and 3) a CTRV cable card (EDA-01105) which provides CTRV cable connections to another rear P2 connector. The mother board uses only the P2 connector. Figure 13 shows the RTM front panel. The 16 digital I/O ports can be configured in pairs as either inputs or outputs. They have protection circuitry which limits their inputs to LVTTL levels. They also have a voltage divider which should be used when they are driven by the high voltage blocking amplifiers. Lastly the 16 Trigger I/O ports connect to both the on-board LEMO connectors as well as to a flat cable header used by a CTRV module. The 16 digital I/O ports are used to deliver timing signals to the

DSP board through the P2 connector or provide general purpose outputs from the DSP board. Jumpers must be used to select the configuration of the digital I/O pairs as inputs or outputs. A LED above the trigger pairs lights green when they are configured as inputs and red when they are configured as outputs. Further jumpers are used to select whether as inputs they are to accept normal TTL/LVTTL levels or high voltage (30 Volt) levels from the blocking amplifier.

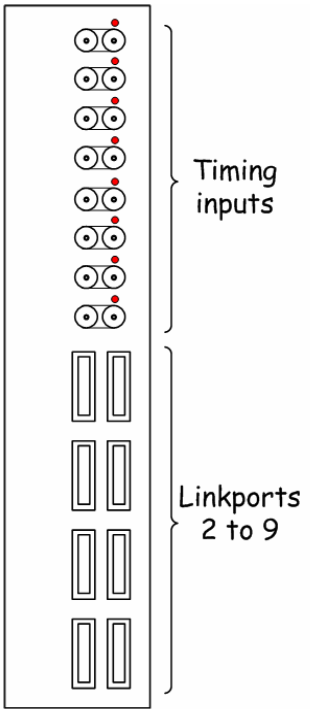


Figure 13: *Two-slots RTM front panel – schematic view.*

4. FPGA FUNCTIONALITY

Table 6 shows the functions carried out by the FPGAs hosted on the DSP-carrier board 1.0.

FPGA NAME	FUNCTIONS
VME interface	VME slave interface implementation, including interrupt capability. Partial address decoding for on-board memory space.
Timing interface	Timing and digital signal reception/processing. DSP interrupt generation.
DSP to receiver	Daughtercard sites control logic. Partial address decoding for daughtercard sites. FPGA and B-train counters implementation.
Receiver to DSP	Communication with daughtercard sites. FPGA not used.
DSP Memory select	Partial address decoding for DSP memory.
MEAS Memory select	Partial address decoding for MEAS memory.

Table 6: *Functions carried out by each FPGA.*

The *VME interface* and *Timing interface* FPGAs correspond to the blue blocks with the same name shown in Figure 1. The *DSP to receiver* and *Receiver to DSP* FPGAs correspond to the “Site control logic” and “Site comms” blue boxes, respectively. The *DSP Memory select* and *MEAS Memory select* FPGAs are not shown in Figure 1. More details on the actual function and implementation are given in paragraphs 4.1 through 4.5 and paragraph 5.

4.1 EMERGENCY BUS

The *Timing interface*, *DSP to Receiver*, *Receiver to DSP* and *VME interface* FPGAs are connected via a bus called "*emergency bus*" and composed of 16 lines. This bus has been introduced to transfer signals from one FPGA to another in case of need or of circumstances unforeseen at the moment of the board design. Table 7 shows the six lines allocated in the emergency bus, the transferred signals, their use and the source/destination FPGAs.

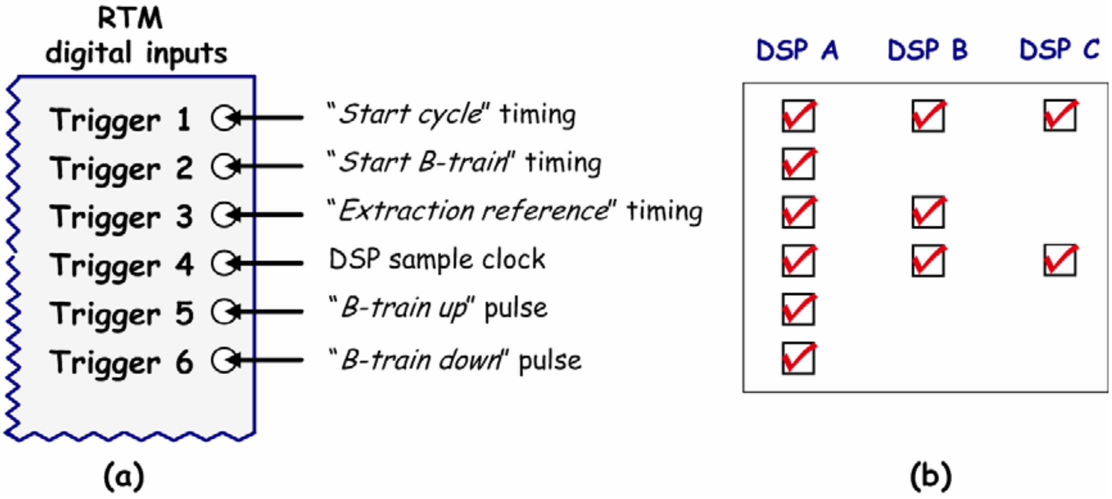
LINE NB	SOURCE FPGA	DESTINATION FPGA	MEANING	USE
0	*Timing interface*	*DSP to receiver*	"*B-train up*" pulse	B-train counter implementation
1	*Timing interface*	*DSP to receiver*	"*B-train down*" pulse	
2	*Timing interface*	*DSP to receiver*	"*B-train reset*" timing	B-train counter reset
3	*Timing interface*	*DSP to receiver*	"*Start cycle*" timing	DSP IRQ2/ generation and FPGA counter reset.
4	*DSP to receiver*	*VME interface*	DSP WRL/ line	MEAS MEM access implementation
5	*DSP to receiver*	*VME interface*	DSP RDL/ line	
10	*Timing interface*	*DSP to receiver*	DSP sample clock	DSP IRQ1/ generation FPGA counter implementation

Table 7: *"Emergency bus" signal allocation, showing the six lines currently used. Source and destination FPGAs are indicated.*

The signals originated from the *Timing interface* FPGA are digital inputs coming from the RTM. The signals originated from the *DSP to receive* FPGA are DSP input/output lines.

4.2 DIGITAL INPUTS AND INTERRUPTS

Figure 14 shows digital inputs to the RTM according to the trigger number they are connected to. The allocation for each DSP used in the LEIR beam control system is also shown.



	DSP A	DSP B	DSP C
Trigger 1	☑	☑	☑
Trigger 2	☑		
Trigger 3	☑	☑	
Trigger 4	☑	☑	☑
Trigger 5	☑		
Trigger 6	☑		

Figure 14: *Digital inputs to the RTM and allocation in each DSP used for the LEIR beam control system.*

Triggers are voltage pulses with a TTL electrical format and different duty cycles. Table 8 shows the RTM digital inputs together with their source and their use.

The "*Start cycle*" timing is issued by a timing receiver module at the start of each user cycle; it resets the *FPGA counter* and generates the DSP interrupt IRQ1/. More details on the *FPGA counter* are given in paragraph 4.3.

The "*Start B-train*" timing is issued by a timing receiver module at the start of each user cycle; it resets the *B-train counter*. More details on the *B-train counter* are given in paragraph 4.3.

The "*Extraction reference*" timing is issued by a timing receiver module at a known time interval prior to the beam extraction and generates the DSP interrupt IRQ1/.

The DSP sample clock has a programmable frequency and it is issued by a timing receiver module; typical frequencies used in LEIR are of 80 kHz or higher.

The "*B-train up*" and "*B-train down*" pulses are generated by the accelerator magnetic measurement system and are used for the B-train counter implementation. More details on the *B-train* are given in paragraph 4.3.

TRIGGER NB	**DIGITAL INPUT**	**SOURCE**	**USE**
1	"*Start cycle*" timing	Accelerator timing system	DSP IRQ2/ generation and FPGA counter reset.
2	"*Start B-train*" timing	Accelerator timing system	B-train counter reset.
3	"*Extraction reference*" timing	Accelerator timing system	DSP IRQ2/ generation.
4	DSP sample clock	Accelerator timing system	DSP IRQ1/ generation FPGA counter implementation
5	"*B-train up*" pulse	Accelerator magnetic measurement system	B-train counter implementation.
6	"*B-train down*" pulse		

Table 8: *Digital inputs to the RTM together with their source and their use.*

4.3 B-TRAIN AND FPGA COUNTERS

Two counters are implemented in the *DSP to receiver* FPGA: the *B-train counter* and the *FPGA counter*. As shown in Table 8, the digital inputs needed for their implementation and reset are originally acquired by the *Timing interface* FPGA then passed on to the *DSP to receiver* FPGA, where the counters are implemented. This is due to the *Timing interface* having an 8-bit data interface with the DSP, which does not provide a sufficient resolution for implementing the counters.

The *B-train counter* sums, in real time, the number of "B-train up" pulses and subtracts the number of "B-train down" pulses received. The counter is reset to the fixed value "B-train offset" by the "Start B-train" timing. The "B-train offset" value is controlled by the DSP and for the LEIR LLRF is set to zero. The B-train counter is used to implement the frequency loop.

The *FPGA counter* sums in real time the number of “DSP sample clock” pulses. The counter is reset to zero by the “Start cycle” timing. The *FPGA counter* provides the DSP with a measure of the time during the cycle.

4.4 FPGA MEMORY MAPPING

Table 9 shows the registers mapped onto the *VME interface*, *Timing interface* and *DSP to receiver* FPGAs. The addresses as seen from the DSP and from the VME bus are also shown.

REGISTERS	FPGA	DSP ADDRESS	VME OFFSET FROM BASE ADDRESS
Reset DSP	*VME interface*	0x0000 0000	0x0040 0000
Host bus request and grant	*VME interface*	Not available	0x00E0 0000
External interrupt	*Timing interface*	0x00E0 000B	Not available
FPGA counter	*DSP to receiver*	0x00E0 0009	Not available
B-train value	*DSP to receiver*	0x00E0 000D	Not available
B-train offset	*DSP to receiver*	0x00E0 000F	Not available
B-train validation bit	*DSP to receiver*	0x00E0 0011	Not available
***DSP to receiver* version number**	*DSP to receiver*	0x00E0 0011	Not available
***Timing interface* version number**	*Timing interface*	0x00E0 0015	Not available

Table 9: *FPGA-implemented registers; addresses are detailed as seen from the DSP and from the VME bus.*

An overview of the FPGA-mapped registers is given below.

Reset DSP: this register is mapped the first location in the internal memory space, i.e. to the SYSCON (System Configuration) register. Setting to 1 the bit 0 (called SRST, system reset) of this register results in the DSP reset.

Host bus request and grant: this register reflects the value of the Bus Grant Request signal, and is used for the VME access to the DSP MEM, FLASH memory and daughtercards memory space.

External interrupt: this register contains a value one to six to indicate which is the timing signal that has been received by the board via the transition module and has generated the DSP interrupt. Upon reception of an interrupt the DSP must read this memory location so as to re-enable the interrupts again.

FPGA counter: see paragraph 4.3.

B-train value: see paragraph 4.3.

B-train offset: see paragraph 4.3.

B-train validation bit: this is the bit 0 of the register and shows whether the B-train value has been properly initialised. If so, it can be used by the DSP in the beam control implementation.

DSP to receiver version number: this register contains a number identifying the *DSP to receiver* FPGA code version. The DSP reads it after a reboot and publishes it at higher level.

Timing interface version number: this register contains a number identifying the *Timing interface* FPGA code version. The DSP reads it after a reboot and publishes it at higher level.

4.5 MEMORY MAPPING

This paragraph details the mapping of the DSP internal, external and memory-mapped I/O (namely the daughtercards) as it is done from the DSP as well as from the VME bus. It should be noted that the DSP addresses in 32-bit words while the VME bus addresses in bytes.

The VME base address of each board is hard-coded in the VME interface. The three VME base addresses currently allocated to the DSP-carrier boards are shown in Table 10.

	VME BASE ADDRESS
DSP A	0x0200 0000
DSP B	0x0300 0000
DSP C	0x0400 0000

Table 10: *VME base addresses; they are hard-coded in the VME interface FPGA.*

Table 11 shows the on-board and DSP internal memory mapping, as seen by the DSP and by the VME bus. It should be noted that the FLASH memory is not accessible from the DSP. The DSP internal memory can be accessed from the VME bus, however this feature is not used. Finally, from a DSP access viewpoint the DSP RAM is composed of odd memory locations only, and the MEAS RAM of even locations only. The reason for this is detailed in paragraph 4.5.1.

		DSP ADDRESS		**VME OFFSET FROM BASE ADDRESS**
		Range	**Notes**	**Range**
DSP Internal memory block #1 and block #2	start	0x0004 0000		0x50 0000
	stop	0x0005 FFFF		0x57 FFFC
DSP RAM	start	0x0080 0000	Odd locations only	0x0060 0000
	stop	0x00A0 0000		0x009F FFFF
MEAS RAM	start	0x00A0 0000	Even locations only	0x00A0 0000
	stop	0x00BF FFFF		0x00DF FFFF
FLASH RAM	start	N/A		0x00000000
	stop	N/A		0x003F FFFF

Table 11: *On-board and DSP internal memory mapping, as seen by the DSP and by the VME bus. DSP addresses are expressed in 32-bit words while VME addresses are expressed in bytes.*

Table 12 shows how the daughtercards memory space is mapped from the VME bus and from the DSP. The memory address is partially decoded by the *DSP to receiver* FPGA, which controls two lines (SEL_0 and SEL_1) for each daughter-card site. Only 15 address bits (A0 to A14) are sent to each daughtercard site owing to the limited number of pins on the daughtercard-to-motherboard connector. In fact, 69 pins are available on each connector. With 15 address lines only it is not possible to address the whole 256k x 16 bit of SRAM located on the daughter board. To allow accessing it, a paging mechanism has been implemented on the daughtercard FPGA. The address ranges defined for site 2 and site 4 are not currently used with the new, two-sites each daughtercards.

			DSP ADDRESS		VME OFFSET FROM BASE ADDRESS
			Range	Notes	Range
SITE 1	SEL 0	start	0x00C8 0000	Odd locations only	0x00F0 0000
		stop	0x00C8 7FFF		0x00F0 FFFF
	SEL 1	start	0x00C8 8000		0x00F1 0000
		stop	0x00C8 FFFF		0x00F1 FFFF
SITE 2	SEL 0	start	0x00C9 0000	Odd locations only	0x00F2 0000
		stop	0x00C9 7FFF		0x00F2 FFFF
	SEL 1	start	0x00C9 8000		0x00F3 0000
		stop	0x00C9 FFFF		0x00F3 FFFF
SITE 3	SEL 0	start	0x00CA 0000	Odd locations only	0x00F4 0000
		stop	0x00CA 7FFF		0x00F4 FFFF
	SEL 1	start	0x00CA 8000		0x00F5 0000
		stop	0x00CA FFFF		0x00F5 FFFF
SITE 4	SEL 0	start	0x00CB 0000	Odd locations only	0x00F6 0000
		stop	0x00CB 7FFF		0x00F6 FFFF
	SEL 1	start	0x00CB 8000		0x00F7 0000
		stop	0x00CB FFFF		0x00F7 FFFF

Table 12: *Daughtercard memory space as mapped by the DSP and by the VME bus. DSP addresses are expressed in 32-bit words while VME addresses are expressed in bytes.*

4.5.1 DSP MEMORY MAP

Figure 15 shows the DSP external port word alignment. The 64-bit data bus is divided into two 32-bit lanes, each one controlled by an independent set of strobes: WRH/ (Write High) and RDH/ (Read High) control the access to the 32 most significative bits (MSB) while RDL/ (Read Low) and WRL/ (Write Low) control the access to the 32 less significative bits (LSB).

The DSP MEM is addressed by using the WRH/ and RDH/ strobes; hence only odd external memory locations (e.g. 0x80 0001, 0x80 0003 etc) are accessible from the DSP.

The MEAS MEM is addressed by using the WRL/ and RDL/ strobes. As a consequence, only even memory locations (e.g. 0xA0 0000, 0xA0 0002 etc) can be addressed by the DSP. The RDL/ and WRL/ signals are routed to the *VME interface* FPGA from the *DSP to receiver* FPGA, as shown in paragraph 4.1. This is done to correct a problem in the board design, namely the fact that the lower 32 data pins have been connected to the MEAS MEM instead of the 32 higher ones. This problem will be corrected in a future version of the board, therefore the MEAS MEM will include odd memory locations only, as is the case for the DSP MEM.

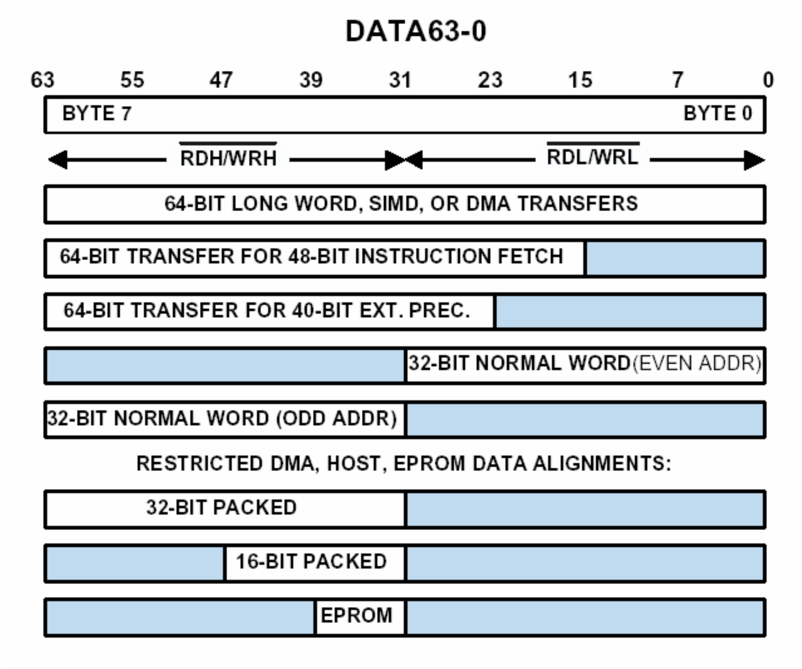

DATA63-0
63
55
47
39
31
23
15
7
0
BYTE 7
BYTE 0
RDH/WRH
RDL/WRL
64-BIT LONG WORD, SIMD, OR DMA TRANSFERS
64-BIT TRANSFER FOR 48-BIT INSTRUCTION FETCH
64-BIT TRANSFER FOR 40-BIT EXT. PREC.
32-BIT NORMAL WORD(EVEN ADDR)
32-BIT NORMAL WORD (ODD ADDR)
RESTRICTED DMA, HOST, EPROM DATA ALIGNMENTS:
32-BIT PACKED
16-BIT PACKED
EPROM

Figure 15: *External port word alignment.*

4.5.2 VME BUS MEMORY MAP

The board is equipped with a VME slave interface, i.e. the DSP cannot access memory on another VME board over the VME bus. Locations in the memory blocks listed in tables 11 and 12 are mapped contiguously from the VME bus.

4.6 DSP MEMORY ACCESS

The DSP accesses the external memory and the memory-mapped peripherals as the bus master and in asynchronous access mode. Figure 16 shows a typical DSP external memory asynchronous read or write cycle.

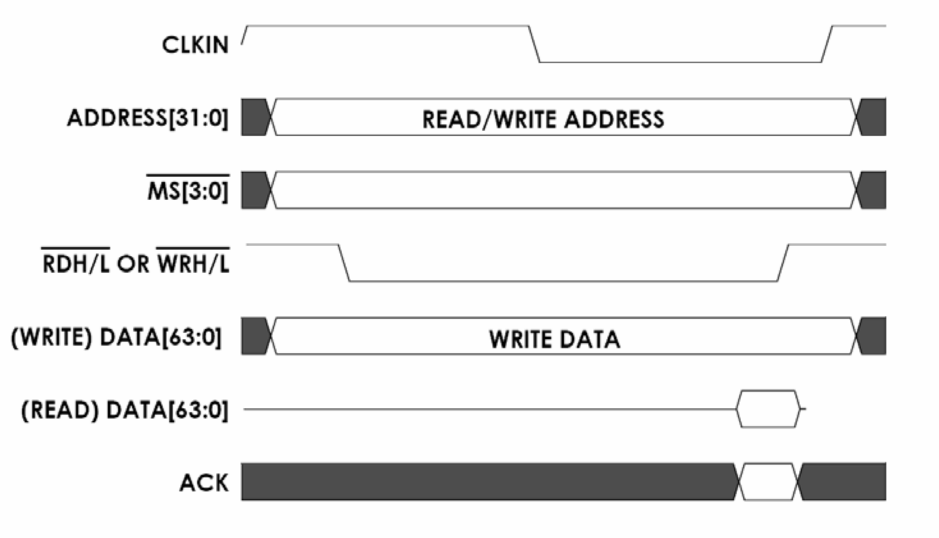
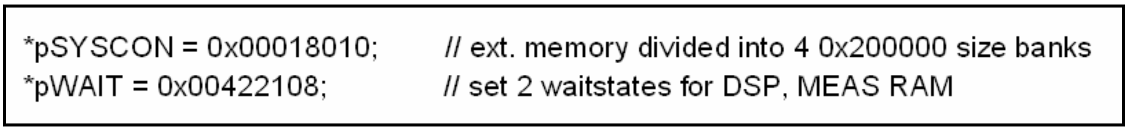


Figure 16: *DSP external memory asynchronous access cycle.*

The access protocol is selected at the beginning of the DSP program by writing to the System Configuration (SYSCON) and External Memory Waitstate and Access Mode (WAIT) registers. Figure 17 shows the DSP code used to initialise the two registers.



```
*pSYSCON = 0x00018010;      // ext. memory divided into 4 0x200000 size banks
*pWAIT = 0x00422108;        // set 2 waitstates for DSP, MEAS RAM
```

Figure 17: *Initialisation values for the System Configuration (SYSCON) and External Memory Waitstate and Access Mode (WAIT) registers*

All external memory (i.e. DSP MEM, MEAS MEM) and the memory-mapped I/O (namely the daughtercards space) are defined as one block hence the DSP allocates to them the same number of Wait States (WS) when accessing a location within this space. Two is the minimum WS number to reliably access this block; one additional hold state is added by the DSP, hence the minimum time needed to access an external memory location is 100 ns.

The DSP can also access daughtercard data via the linkports, i.e. avoiding the global bus.

4.7 VME BUS MEMORY ACCESS AND ARBITRATION

The access to the on-board memory is controlled by the *VME interface* FPGA. The VME bus is always treated as a high priority task, therefore it is granted mastership of the global bus whenever it requires it.

4.7.1 DSP MEM, FLASH MEMORY AND DAUGHTERCARD MEMORY SPACE

The DSP MEM, FLASH memory and daughtercards memory space are accessed via the DSP external bus. Interleaved DSP and VME bus accesses are not possible owing to the lack of a register buffering the data during the long VME cycle. This is in fact needed to release access to the DSP external bus during VME transfers. In case of competition between DSP and VME bus, priority is given to the latter; hence the DSP remains stalled until the VME bus access is completed. DSP access to the daughtercards memory space via the external bus will be impossible, while DSP access via the linkport will still be possible as it is external bus-independent.

The DSP lines Host Bus Request (HBR/), Host Bus Grant (HBG/), Chip Select (CS/) and Host Bus Acknowledge (REDY) must be driven according to a pre-defined protocol to allow a host processor to gain access to the DSP external bus. The *VME interface* FPGA implements this protocol, according to the DSP requirements for an asynchronous host processor access. The host processor signals the start of a read/write operation from the VME bus by clearing the FPGA register "*Host bus request and grant*". The end of the VME bus access is signalled by setting the same register to the value of 1.

4.7.2 MEAS MEM

The MEAS MEM has two access ports, one from the VME bus and the other from the DSP; the access arbitration is managed by the *MEAS MEM select* and *VME interface* FPGAs. Even when the VME bus is accessing the MEAS MEM memory space, the DSP can still access memory blocks which are mapped on the global bus, such as the daughter-cards memory space and the DSP MEM.

The MEAS MEM is not a dual port memory: only one between VME bus and DSP can access it for reading or writing at a given moment. The VME bus has higher priority over the DSP, hence it will access the MEAS MEM whenever it requires it. If the DSP then tries to access any memory location in the MEAS MEM, it will retrieve indefinite data without any error message being issued. A software semaphore must then be implemented to prevent this from happening.

5. DSP INTERRUPTS

5.1 VME BUS-TO-DSP INTERRUPTS

VME bus-to-DSP interrupts are currently not needed but could be implemented on the board.

5.2 DSP-TO-VME BUS INTERRUPTS

DSP-to-VME bus interrupts have been implemented as Release On ACKnowledge (ROAK) and successfully tested at CERN. The interrupt level and its STATUS/ID are hard-coded in the VME Interface FPGA. The values used for each DSP board are shown in Table 13.

	INTERRUPT LEVEL	**STATUS/ID**
DSP A	2	220 = 0xDC = 1101 1100
DSP B	2	221 = 0xDD= 1101 1101
DSP C	2	222 = 0xDE = 1101 1110

Table 13: *Interrupt levels and STATUS/ID values used depending on the DSP board.*

The DSP generates an interrupt by setting to 0 and then to 1 the pin 0 (FLG0) of the Flag Value Register. This is a non-memory mapped, universal system register, with a reset value equal to 0x0; the pin must be set as output pin during initialisation. When a flag pin is an output and the program sets the pin's bit in FLAGS, the DSP outputs a high. The I/O direction (input or output) selection of each bit is controlled by its FLG[3:0] bits in the MODE2 register.

5.3 DSP EXTERNAL INTERRUPTS

The DSP chip has three pins available for external interrupts. In order of decreasing priority, they are IRQ2/, IRQ1/ and IRQ0/. Table 14 shows the distribution of IRQ lines to the different elements in the DSP-carrier board.

IRQ/ pin	**DISTRIBUTION**
IRQ0/	▪ Each daughter-card site. ▪ *Timing interface* FPGA. ▪ *Receiver to DSP* FPGA (FPGA not used).
IRQ1/	▪ *Timing interface* FPGA. ▪ *Receiver to DSP* FPGA (FPGA not used).
IRQ2/	▪ *Timing interface* FPGA. ▪ *Receiver to DSP* FPGA (FPGA not used).

Table 14: *Distribution of IRQ interrupt pins to different board elements.*

IRQ0/ is generated by the daughtercards and corresponds to an alarm situation.

IRQ1/ is generated by the *Timing interface* FPGA and triggers the DSP periodic routine that actually implements the beam control loops.

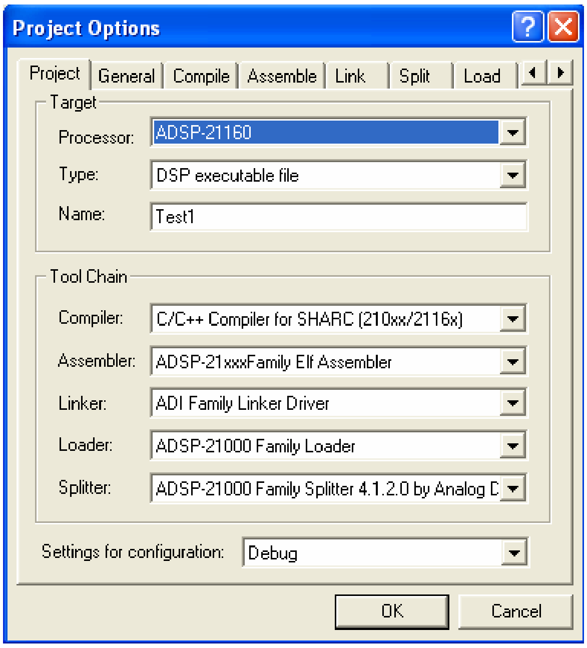
IRQ2/ is generated by the *Timing interface* FPGA and corresponds to an external timing event. A dedicated register ("External interrupt", see paragraph 4.4) contains a value 0 to 6 indicating which timing signal has triggered the interrupt. The interrupts remains latched until the DSP reads this register; only then a new incoming interrupt can be treated.

6. DSP CODE LOADING AND OPERATION

This paragraph discusses the possible ways to interact with the DSP; the preferred one depends on the development stage as well as on the number of the boards composing the system. During early development and for a single-board test, JTAG operation is often more convenient. For further development and final exploitation, the use of the DSP code loader and of the DSP driver (hence allowing operating the DSP from the VME bus) is typically preferred.

6.1 JTAG OPERATION

The DSP executable can be directly uploaded to the DSP from the development PC via the JTAG interface and the corresponding emulator. The DSP can then be fully operated from the JTAG header (J2) on the DSP board, i.e. it is possible to execute the code in a step-by-step fashion, displaying memory and register contents. To this purpose the DSP code should be created as "DSP executable file" (extension "dxe "), as shown in Figure 18. This corresponds to building a stand-alone executable, fully resolved and linked, that can be uploaded to a processor target. A "DSP executable file" includes: a) DSP instructions (code and data); b) symbol table and section information; c) target processor memory layout; d) debug information.



Target

Processor: ADSP-21160

Type: DSP executable file

Name: Test1

Tool Chain

Compiler: C/C++ Compiler for SHARC (210xx/2116x)

Assembler: ADSP-21xxxFamily Elf Assembler

Linker: ADI Family Linker Driver

Loader: ADSP-21000 Family Loader

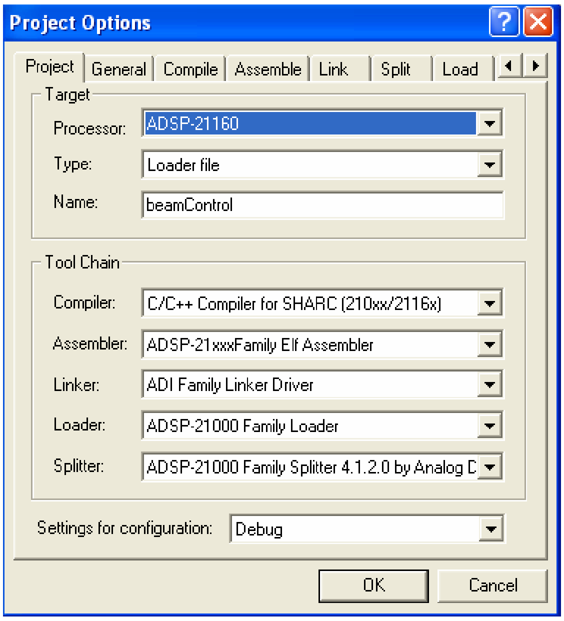
Splitter: ADSP-21000 Family Splitter 4.1.2.0 by Analog D

Settings for configuration: Debug

Figure 18: *Project tab of the SP project options for a DSP executable file. The screen dump is taken from VisualDSP++ v. 3.5.*

6.2 DSP CODE LOADER AND FLASH MEMORY

The DSP code can be uploaded and stored to the on-board FLASH memory via a dedicated program; the DSP can then boot from the FLASH memory following a reset or a power-off. The DSP code must be generated as “loader file” (extension “ldr”); this file is composed of the DSP executable code and of a loader header, which moves the program data from external to internal memory at boot time. Figure 19 shows how to select a loader file as the output in the DSP project option menu. Figure 20 shows other settings in the DSP project file. In particular, the Boot Format should be specified as ASCII in the “LOAD” tab for the loader to function.


Project Options
Project | General | Compile | Assemble | Link | Split | Load
OK
Cancel

Target

Processor: ADSP-21160

Type: Loader file

Name: beamControl

Tool Chain

Compiler: C/C++ Compiler for SHARC (210xx/2116x)

Assembler: ADSP-21xxxFamily Elf Assembler

Linker: ADI Family Linker Driver

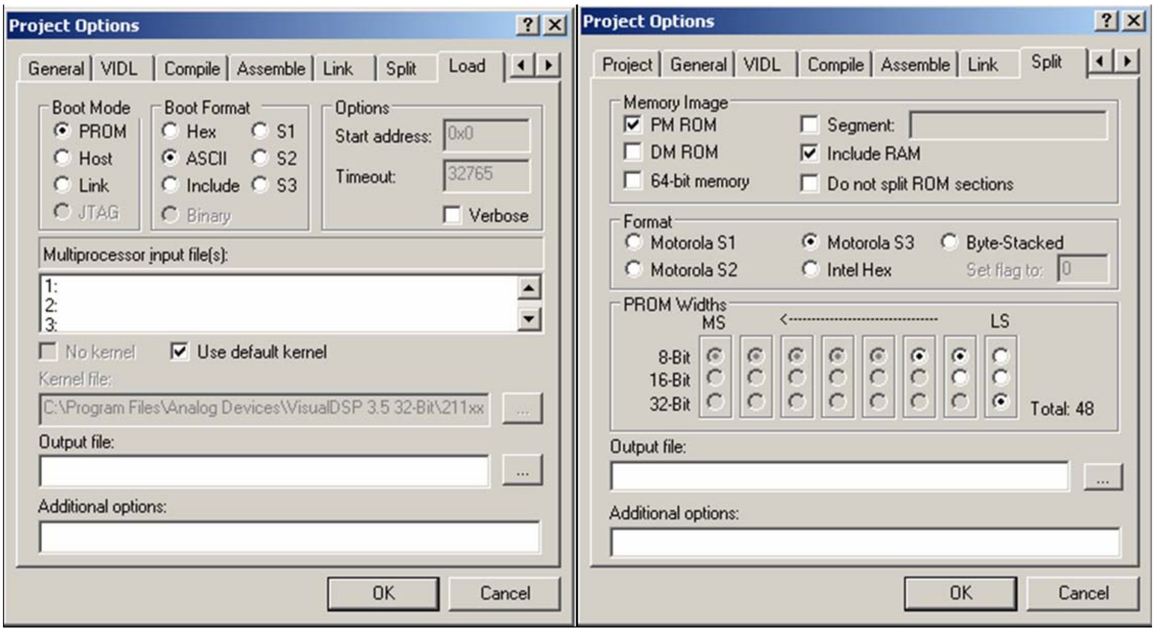
Loader: ADSP-21000 Family Loader

Splitter: ADSP-21000 Family Splitter 4.1.2.0 by Analog D

Settings for configuration: Debug

.

Figure 19: *“Project” tab of the DSP project options when creating a DSP loader file. Screen dump taken from VisualDSP++ v. 3.5.*


Project Options
General | VIDL | Compile | Assemble | Link | Split | Load
OK
Cancel
Project Options
Project | General | VIDL | Compile | Assemble | Link | Split
OK
Cancel

Boot Mode: ● PROM ○ Host ○ Link ○ JTAG

Boot Format: ○ Hex ○ S1 ● ASCII ○ S2 ○ Include ○ S3 ○ Binary

Options: Start address: 0x0 Timeout: 32765 □ Verbose

Multiprocessor input file(s):
1:
2:
3:

□ No kernel ☑ Use default kernel

Kernel file: C:\Program Files\Analog Devices\VisualDSP 3.5 32-Bit\211xx

Output file:

Additional options:

Memory Image: ☑ PM ROM □ DM ROM □ 64-bit memory □ Segment: ☑ Include RAM □ Do not split ROM sections

Format: ○ Motorola S1 ○ Motorola S2 ● Motorola S3 ○ Intel Hex ○ Byte-Stacked Set flag to: 0

PROM Widths MS <------------ LS

8-Bit	●	●	●	●	●	●	●	○
16-Bit	○	○	○	○	○	○	○	○
32-Bit	○	○	○	○	○	○	○	●

Total: 48

Output file:

Additional options:

Figure 20 *“Load” and “Split” tabs in the DSP project option. Screen dump taken from VisualDSP++ v. 3.5.*

A “loader program” running under LynxOS has been developed to load the .ldr file onto the FLASH memory. The program handles the FLASH read/write memory protocols described in paragraph 3.4.

6.3 DSP DRIVER

A VME driver running under LynxOS has been developed at CERN, based upon the initial BNL driver code running under VxWorks. The VME driver can be installed in a front-end crate and called by a FESA Real-Time Task as well as by a test executable. The main actions implemented are the following:

a. Read / write from DSP MEM. The protocol described in paragraph 4.7.1 is implemented.

b. Read / write from MEAS MEM.

c. Detection of a DSP-to-VME bus interrupt and of the DSP generating it.

d. Reset of a DSP, specified via its VME base address.

7. ACKNOWLEDGMENT

The author wishes to acknowledge the contribution of Joe Delong, now with Motorola, for the design of the DSP-carrier board release 1.0.

8. REFERENCES

[1] M.E. Angoletta, J. DeLong (BNL), A. Findlay, F. Pedersen, *“Feasibility Tests Of A New All-Digital Beam Control Scheme For LEIR”*, AB-Note-2004-004-RF.

[2] M.E. Angoletta, J. Bento, A. Blas, A. Findlay, P. Matuszkiewicz, F. Pedersen, A. Salom-Sarasqueta, J. H. DeLong, *“Beam Tests Of A New Digital Beam Control System For The CERN LEIR Accelerator”*, PAC05, Knoxville, USA.

[3] M.E. Angoletta, A. Findlay, *“First experience with the new LEIR digital beam control system”*, AB-Note-2006-003-RF.

[4] Analog Devices, *“ADSP-21160 SHARC® DSP Hardware Reference”*, Revision 3.0, November 2003.

APPENDIX A – DSP-CARRIER BOARD J2/P2 PINS ALLOCATION

Table 15 gives the custom pin layout implemented by the J2/P2 connector in the DSP board release 1.0. The pin numbers are shown together with the corresponding letters used in the board schematics. The user-defined pins are coloured in blue.

Use of VMEbus P2/J2 pins on DSP-carrier board					
Pin	**Row z**	**Row a**	**Row b**	**Row c**	**Row d**
1 (A)	TRIG1	LINK2_CLK	+5 VDC	LINK4_CLK	
2 (B)	GND		GND		
3 (C)	TRIG2	LINK2_ACK	RETRY*	LINK4_ACK	
4 (D)	GND		A24		
5 (E)	TRIG3	LINK2_DATA0	A25	LINK4_DATA0	
6 (F)	GND	LINK2_DATA1	A26	LINK4_DATA1	
7 (G)	TRIG4	LINK2_DATA2	A27	LINK4_DATA2	
8 (H)	GND	LINK2_DATA3	A28	LINK4_DATA3	
9 (I)	TRIG5	LINK2_DATA4	A29	LINK4_DATA4	
10 (J)	GND	LINK2_DATA5	A30	LINK4_DATA5	
11 (K)	TRIG6	LINK2_DATA6	A31	LINK4_DATA6	
12 (L)	GND	LINK2_DATA7	GND	LINK4_DATA7	
13 (M)	EVENT_LINK_S1		+5 VDC		
14 (N)	GND	LINK3_CLK	D16	LINK5_CLK	
15 (O)	EVENT_LINK_S2		D17		
16 (P)	GND	LINK3_ACK	D18	LINK5_ACK	
17(Q)			D19		
18 (R)	GND	LINK3_DATA0	D20	LINK5_DATA0	
19 (S)		LINK3_DATA1	D21	LINK5_DATA1	
20 (T)	GND	LINK3_DATA2	D22	LINK5_DATA2	
21 (U)		LINK3_DATA3	D23	LINK5_DATA3	
22 (V)	GND	LINK3_DATA4	GND	LINK5_DATA4	
23 (W)		LINK3_DATA5	D24	LINK5_DATA5	
24 (X)	GND	LINK3_DATA6	D25	LINK5_DATA6	
25 (Y)		LINK3_DATA7	D26	LINK5_DATA7	
26 (Z)	GND		D27		
27 (AA)		SP0_DT	D28	SP1_DT	
28 (AB)	GND	SP0_DR	D29	SP1_DR	
29 (AC)		SP0_TCLK	D30	SP1_TCLK	
30 (AD)	GND	SP0_RCLK	D31	SP1_RCLK	
31 (AE)		SP0_TFS	GND	SP1_TFS	GND
32 (AF)	GND	SP0_RFS	+5 VDC	SP1_RFS	VPC

Table 15: *Custom pin assignment in the J2/P2 connector for the DSP board release 1.0.*

APPENDIX B – TWO-SLOTS RTM J2/P2 PINS ALLOCATION

Table 16 shows the custom pin layout implemented by the J2/P2 connector in the two-slot RTM. The user-defined pins are coloured in blue.

Use of VMEbus P2/J2 pins on the two-slot RTM module					
Pin	**Row z**	**Row a**	**Row b**	**Row c**	**Row d**
1	TRIG1	LINK2_CLK	+5 VDC	LINK4_CLK	LINK7_CLK
2	GND		GND		LINK7_ACK
3	TRIG2	LINK2_ACK	RETRY*	LINK4_ACK	LINK7_DATA0
4	GND		A24		LINK7_DATA1
5	TRIG3	LINK2_DATA0	A25	LINK4_DATA0	LINK7_DATA2
6	GND	LINK2_DATA1	A26	LINK4_DATA1	LINK7_DATA3
7	TRIG4	LINK2_DATA2	A27	LINK4_DATA2	LINK7_DATA4
8	GND	LINK2_DATA3	A28	LINK4_DATA3	LINK7_DATA5
9	TRIG5	LINK2_DATA4	A29	LINK4_DATA4	LINK7_DATA6
10	GND	LINK2_DATA5	A30	LINK4_DATA5	LINK7_DATA7
11	TRIG6	LINK2_DATA6	A31	LINK4_DATA6	LINK8_CLK
12	GND	LINK2_DATA7	GND	LINK4_DATA7	LINK8_ACK
13	TRIG7		+5 VDC		LINK8_DATA0
14	GND	LINK3_CLK	D16	LINK5_CLK	LINK8_DATA1
15	TRIG8		D17		LINK8_DATA2
16	GND	LINK3_ACK	D18	LINK5_ACK	LINK8_DATA3
17	TRIG9		D19		LINK8_DATA4
18	GND	LINK3_DATA0	D20	LINK5_DATA0	LINK8_DATA5
19	TRIG10	LINK3_DATA1	D21	LINK5_DATA1	LINK8_DATA6
20	GND	LINK3_DATA2	D22	LINK5_DATA2	LINK8_DATA7
21	TRIG11	LINK3_DATA3	D23	LINK5_DATA3	LINK9_CLK
22	GND	LINK3_DATA4	GND	LINK5_DATA4	LINK9_ACK
23	TRIG12	LINK3_DATA5	D24	LINK5_DATA5	LINK9_DATA0
24	GND	LINK3_DATA6	D25	LINK5_DATA6	LINK9_DATA1
25	TRIG13	LINK3_DATA7	D26	LINK5_DATA7	LINK9_DATA2
26	GND		D27		LINK9_DATA3
27	TRIG14	LINK6_CLK	D28		LINK9_DATA4
28	GND	LINK6_ACK	D29		LINK9_DATA5
29	TRIG15	LINK6_DATA0	D30	LINK6_DATA4	LINK9_DATA6
30	GND	LINK6_DATA1	D31	LINK6_DATA5	LINK9_DATA7
31	TRIG16	LINK6_DATA2	GND	LINK6_DATA6	GND
32	GND	LINK6_DATA3	+5 VDC	LINK6_DATA7	VPC

Table 16: *Custom pin assignment in the J2/P2 connector for the two-slot RTM..*

APPENDIX C – PROCEDURES FOR FPGA PROGRAMMING

It is possible to program the FPGAs by using the JTAG headers detailed in paragraph 3.7. Three different procedures must be applied depending on the target FPGA.

1. TIMING INTERFACE AND DSP TO RECEIVER FPGAS

The *Timing interface* and *DSP to receiver* FPGAs are SRAM based devices programmed via the same JTAG interface. The steps a) to f) below must be followed to program the FPGAs.

a) Connect the JTAG cable to the J6 header.

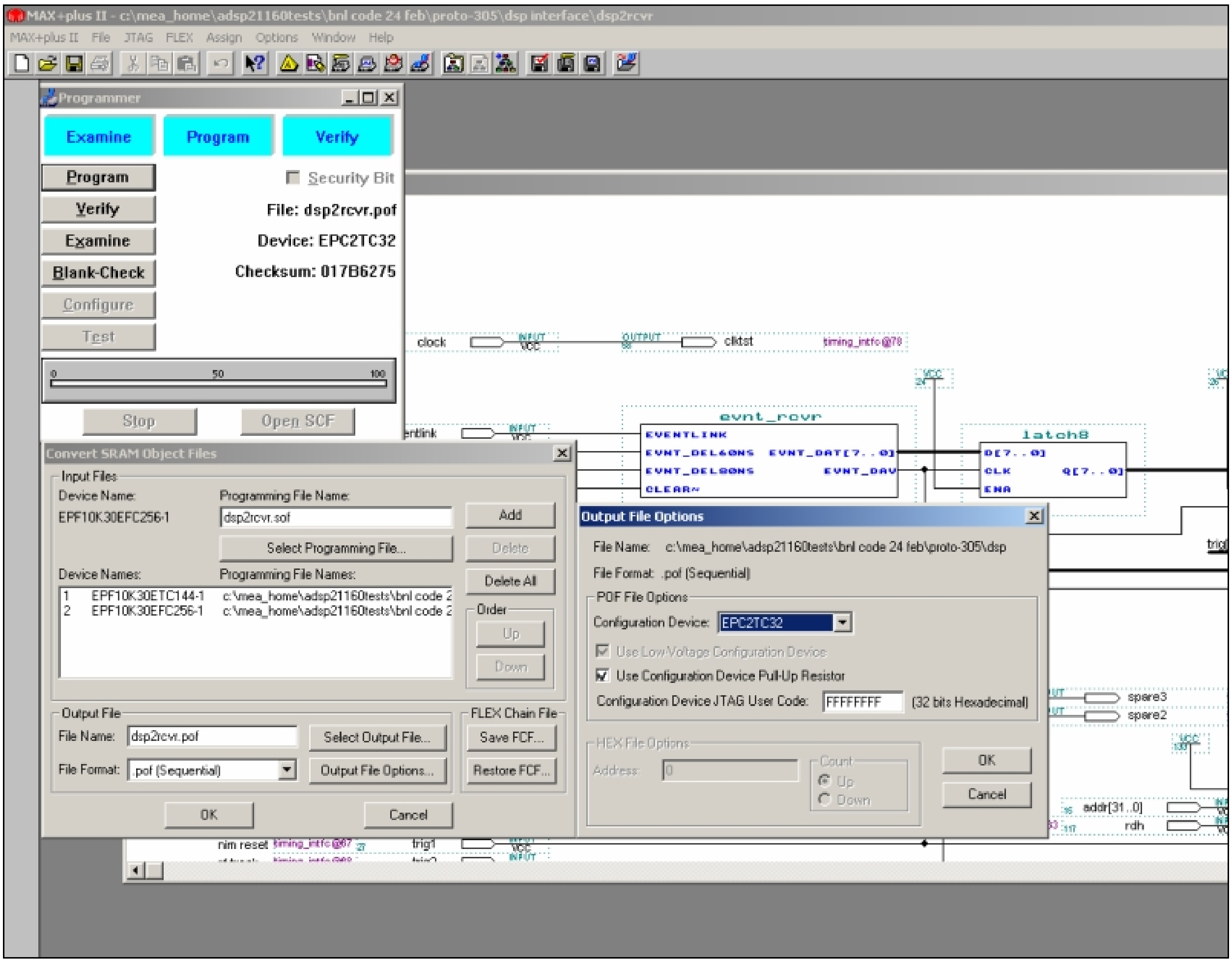
b) Open the programmer, by selecting MAX+plus II then Programmer.

c) Leave the Programmer window in the background and open the “Convert SRAM object Files”.

d) Select the two .sof files to be used for the generation of a .pof file. Currently their names are *timing_intfc.sof* and *dsp2rcvr.sof*. Define a name for the resulting .pof file. In the *Output File Option* the configuration device has to be set as EPC2TC32, as shown in Figure 21.

e) Click on OK and create the resulting .pof file.

f) In the Programmer window select the .pof file that was just created. Check also that the Device is the EPC2TC32. Press Program, and the code will be moved to SRAM.



MAX+plus II - c:\mea_home\adsp21160tests\bnl code 24 feb\proto-305\dsp interface\dsp2rcvr

Programmer

□ Security Bit

File: dsp2rcvr.pof

Device: EPC2TC32

Checksum: 017B6275

Convert SRAM Object Files

Input Files

Device Name: EPF10K30EFC256-1

Programming File Name: dsp2rcvr.sof

Device Names: / Programming File Names:

1 EPF10K30ETC144-1 c:\mea_home\adsp21160tests\bnl code 2

2 EPF10K30EFC256-1 c:\mea_home\adsp21160tests\bnl code 2

Output File

File Name: dsp2rcvr.pof

File Format: .pof (Sequential)

Output File Options

File Name: c:\mea_home\adsp21160tests\bnl code 24 feb\proto-305\dsp

File Format: .pof (Sequential)

POF File Options

Configuration Device: EPC2TC32

☑ Use Low Voltage Configuration Device

☑ Use Configuration Device Pull-Up Resistor

Configuration Device JTAG User Code: FFFFFFFF (32 bits Hexadecimal)

HEX File Options

Address: 0

Count: ● Up ○ Down

Figure 21: *MAX+plus II screen dump showing how to load the FPGA code onto the Timing interface and DSP to receiver FPGAs.*

2. VME INTERFACE FPGA

To load the compiled code one should:

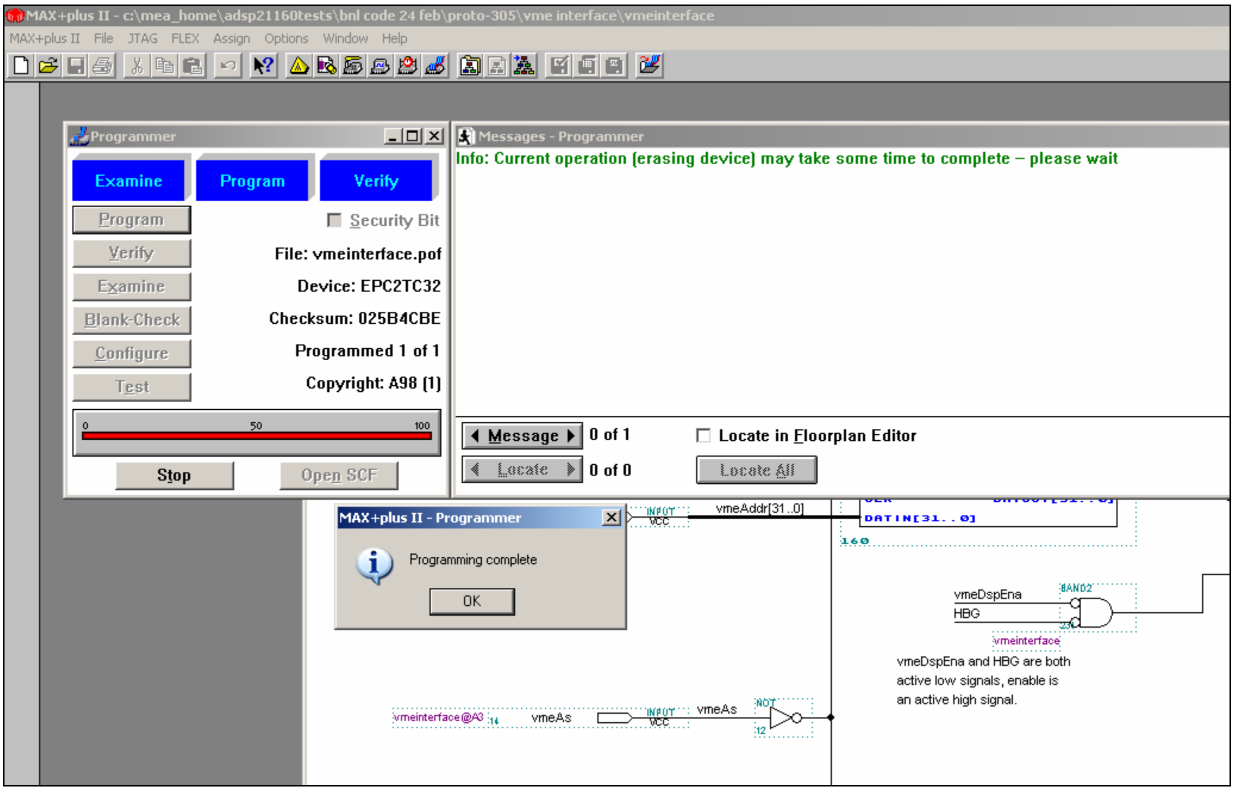
a) Connect the JTAG cable to the J5 header.

b) Open the programmer, by selecting MAX+plus II then Programmer.

c) Select *vmeinterface.pof* as the file to load.

d) Check the Device is the EPC2TC32, as shown in figure 22.

e) Press the Program button.



MAX+plus II - c:\mea_home\adsp21160tests\bnl code 24 feb\proto-305\vme interface\vmeinterface

Messages - Programmer

Info: Current operation (erasing device) may take some time to complete – please wait

□ Security Bit

File: vmeinterface.pof

Device: EPC2TC32

Checksum: 025B4CBE

Programmed 1 of 1

Copyright: A98 (1)

Message 0 of 1

Locate 0 of 0

□ Locate in Floorplan Editor

MAX+plus II - Programmer

Programming complete

vmeDspEna and HBG are both active low signals, enable is an active high signal.

Figure 22: *MAX+plus II screen dump showing how to load the FPGA code onto the VME interface FPGA.*

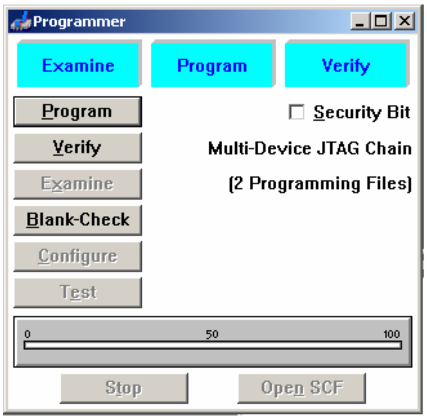
3. DSP MEMORY SELECT AND MEAS MEMORY SELECT FPGA

To load the compiled code one should:

a) Connect the JTAG cable to the J1 header.

b) Open the programmer, by selecting MAX+plus II then Programmer.

c) Leave the Programmer in the background and set JTAG to Multi-Device JTAG chain. This will change the appearance of the programmer to be as shown in figure 23.



Multi-Device JTAG Chain

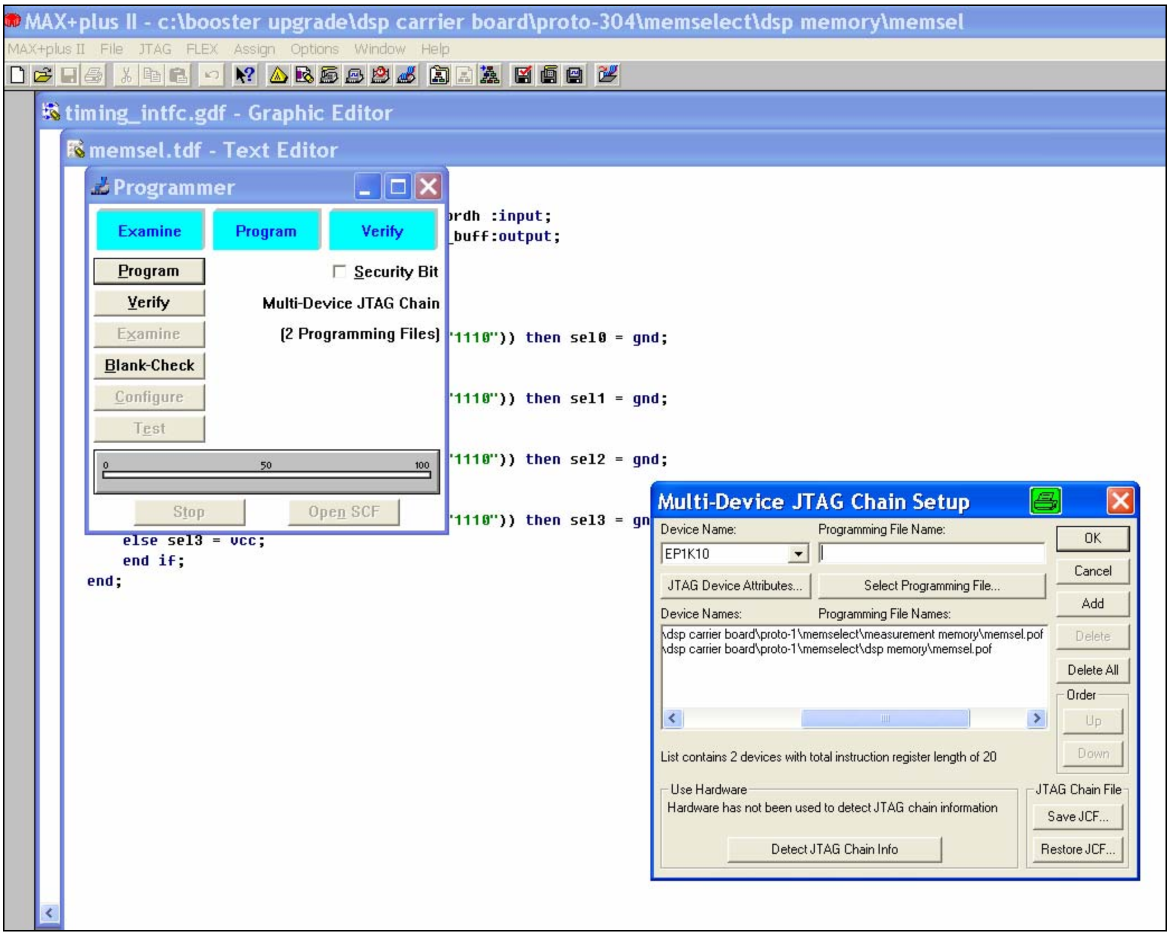
(2 Programming Files)

Figure 23: *Programmer window after the Multi-device JTAG chain option has been selected.*

d) Setup the files that will be downloaded, by clicking JTAG-> Multi-device JTAG chain setup. The two files that have to be included are *memselect\measurement memory\memsel.pof* and *memselect\dsp memory\memsel.pof*.

e) By pressing the button “Detect JTAG chain info” it is possible to check in the hardware whether the selected JTA chain corresponds to the hardware. The JTAG chain details can also be saved as a .jcf file.

f) Press Program to download the code.



Multi-Device JTAG Chain

(2 Programming Files)

```
rdh :input;
_buff:output;

"1110")) then sel0 = gnd;

"1110")) then sel1 = gnd;

"1110")) then sel2 = gnd;

"1110")) then sel3 = gn
    else sel3 = vcc;
    end if;
end;
```

Device Name: EP1K10

\dsp carrier board\proto-1\memselect\measurement memory\memsel.pof
\dsp carrier board\proto-1\memselect\dsp memory\memsel.pof

List contains 2 devices with total instruction register length of 20

Hardware has not been used to detect JTAG chain information

Figure 24: *MAX+plus II screen dump showing how to load the FPGA code onto the DSP MEM select and MEAS MEM select FPGAs.*