LeMans

VX8 Carrier Board
Technical Reference

Document Number 500-00329

Revision 2.02

September 1998

Part Number 500-00329
Revision 2.02

Customer Feedback

At Spectrum, we recognize that product documentation that is both accurate and easy to use is important in aiding you in your new product development. We appreciate hearing your comments on how our product's documentation could be improved.

If you wish to comment on any Spectrum documentation then please fax or e-mail a completed copy of this page to us.

Full Name of Document: _____

Document Number: _____ Version Number: _____

If you have found a technical inaccuracy please describe it here:

_____

_____

_____

_____

_____

If you particularly liked or disliked an aspect of the manual then please describe it here:

_____

_____

_____

_____

_____

It may be helpful for us to call you to discuss your comments. If this would be acceptable please provide the following details:

Name: _____ Telephone #: _____

Organization: _____

Thank you for your time,

Spectrum Signal Processing Documentation Group

Fax: (604) 421-1764
Email: documentation@spectrumsignal.com

Contacting Spectrum…

Spectrum's team of dedicated Applications Engineers are available to provide technical support to you for this product. Our office hours are Monday to Friday, 8:00 AM to 5:00 PM, Pacific Standard Time.

Telephone	1-800-663-8986 or (604) 421-5422
Fax	(604) 421-1764
Email	support@spectrumsignal.com
Internet	http://www.spectrumsignal.com

When you contact us, please have the following information on hand:

- A concise description of the problem
- The name of all Spectrum hardware components
- The name and version number of all Spectrum software components
- The minimum amount of code that demonstrates the problem
- The version number of all software packages, including compilers and operating systems

Preface

Spectrum Signal Processing offers a complete line of DSP hardware, software and I/O products for the DSP Systems market based on the latest DSP microprocessors, bus interface standards, I/O standards and software development environments. By delivering quality products, and DSP expertise tailored to specific application requirements, Spectrum can consistently exceed the expectations of our customers. We pride ourselves in providing unrivaled pre and post sales support from our team of application engineers. Spectrum has excellent relationships with third party vendors which allows us to provide our customers with a more diverse and top quality product offering.

Spectrum achieved ISO 9001 quality certification in 1994.

As Spectrum's hardware products are static sensitive, please take precautions when handling and make sure they are protected against static discharge.

Table of Contents

List of Figures

List of Tables

1 Introduction

1.1. Purpose of This Manual

This manual describes the features, architecture, and specifications of the VX8 Carrier Board. It will help you understand how the function libraries described in the *VX8 Carrier Board Programming Guide* operate within the board. You can also use this information to program the board at a driver level, extend the standard hardware functionality, or obtain more information to do custom configurations.

In addition to the *VX8 Carrier Board Programming Guide* and this manual, the *VX8 Carrier Board Installation Guide* shows you how to install and configure the board, as well as loading and running some of the sample code provided with the developer's package.

Note: Use the software control libraries supplied with the VX8 C4x Support Software to initialize and transfer data to the VX8 hardware interfaces. The libraries should be used as supplied, without modification, due to the complexity of the interfaces.

1.2. Reference Documents

This guide is meant to be used with the following documents:

- ***VX8 Carrier Board Programming Guide*** available from Spectrum
- ***VX8 Carrier Board Installation Guide*** available from Spectrum
- *VMEbus Extensions for Instrumentation (VXIbus) VXI-1* Revision 1.4. Authored by the VXIbus Consortium, Inc.
- *TIM-40 TMS320C4x Module Specification* Version 1.01 available from Texas Instruments
- *TMS320C4x User's Guide* available from Texas Instruments
- *SCV64 VME64 Interface IC Data Book* available from Tundra Semiconductor
- *Atmel AT29C256 32 Kbyte PEROM Data Sheet* available from Atmel Semiconductor (This document is included with the VX8 Carrier Board documentation)
- *PC16552D data sheet* available from National Semiconductor
- *Spectrum TIM-40 PEROM Programming Guide* available from Spectrum (included with the VX8 Developer's Kit)
- *BALLISTIC VXIbus Interface Chip Data Sheet* available from Hewlett-Packard

2 General Description

2.1. Features

Spectrum's VX8 Carrier Board provides the processing engine for a VXIbus multiple DSP system.

2.1.1. TMS320C4x Nodes

Six TIM-40 sites and two on-board 60 MHz TMS320C40 (C40) processors are incorporated onto the VX8 Carrier Board. The embedded C40s are nodes A and B, and the TIM-40 sites are nodes C to H. Each node has one buffered C4x communication port brought to the front panel.

Each embedded node (A and B) features:

- One bank of 128k x 32 SRAM on both the local and global buses for a total of 1 Mbytes per C40 (Upgradeable to 512 x 32 SRAMs at the factory);
- One 32kx8 PEROM for booting or TIM-40 IDROM compatibility on the local bus;
- The routing of global bus signals to buffers to allow for HP Local Bus DMA controlled data writes to global SRAM;
- The capability to write to the HP Local Bus output FIFO or to access the shared global DRAM through the global bus connector; and
- The capability to access the SCV64 IC to act as a VXIbus master.

Node A has an additional 32k x 8 PEROM used for the board's boot kernel.

Node B has a DUART equipped with RS-232 drivers brought to the front panel Dual RS-232 asynchronous serial ports.

2.1.2. Bus Interfaces

The VX8 has a register based VXIbus interface incorporating an optional Hewlett Packard (HP) local bus interface. The HP Local Bus interface uses a high speed BALLISTIC chip and an intelligent DMA controller.

The VX8 can function as either a Master or as a Slave module on the VXIbus. VXIbus D32, D16, and D08E0 data access is supported in the following address modes:

Address Mode	Master	Slave
A32	Yes	Yes
A24	Yes	No
A16	Yes	VXI registers only

2.1.3. Diagnostic Support

Diagnostic and debugging support for the VX8 is provided through the following features:

- C language source symbolic debugger through front panel JTAG In and Out connectors to an XDS510
- On board Test Bus Controller for C language source symbolic debugger with WIN95/NT Intel VXI slot 0 controllers

2.2. Board Layout

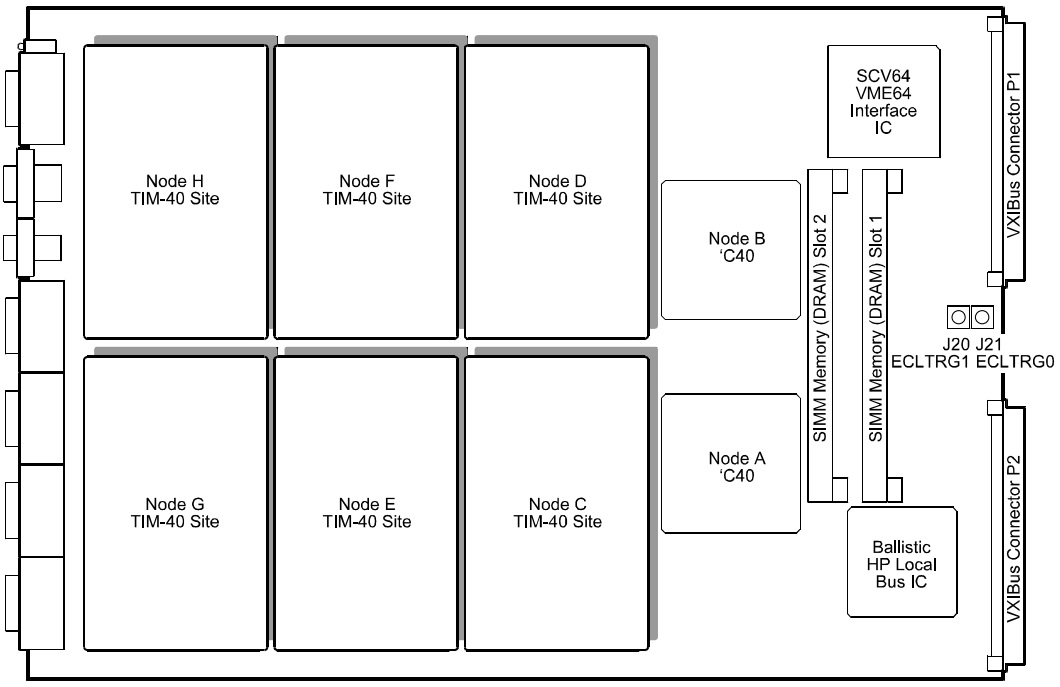


Figure 1 VX8 Carrier Board Layout

Note: J20 ECLTRG1 also generates SYNCLK.

2.3. Front Panel

The front panel of the VX8 has a variety of connectors and status LEDs as shown in the following illustration. The pinouts for the connectors are described in *chapter 12*. The LEDs are described in *section 7.2* of the *DRAM Shared Bus* chapter.

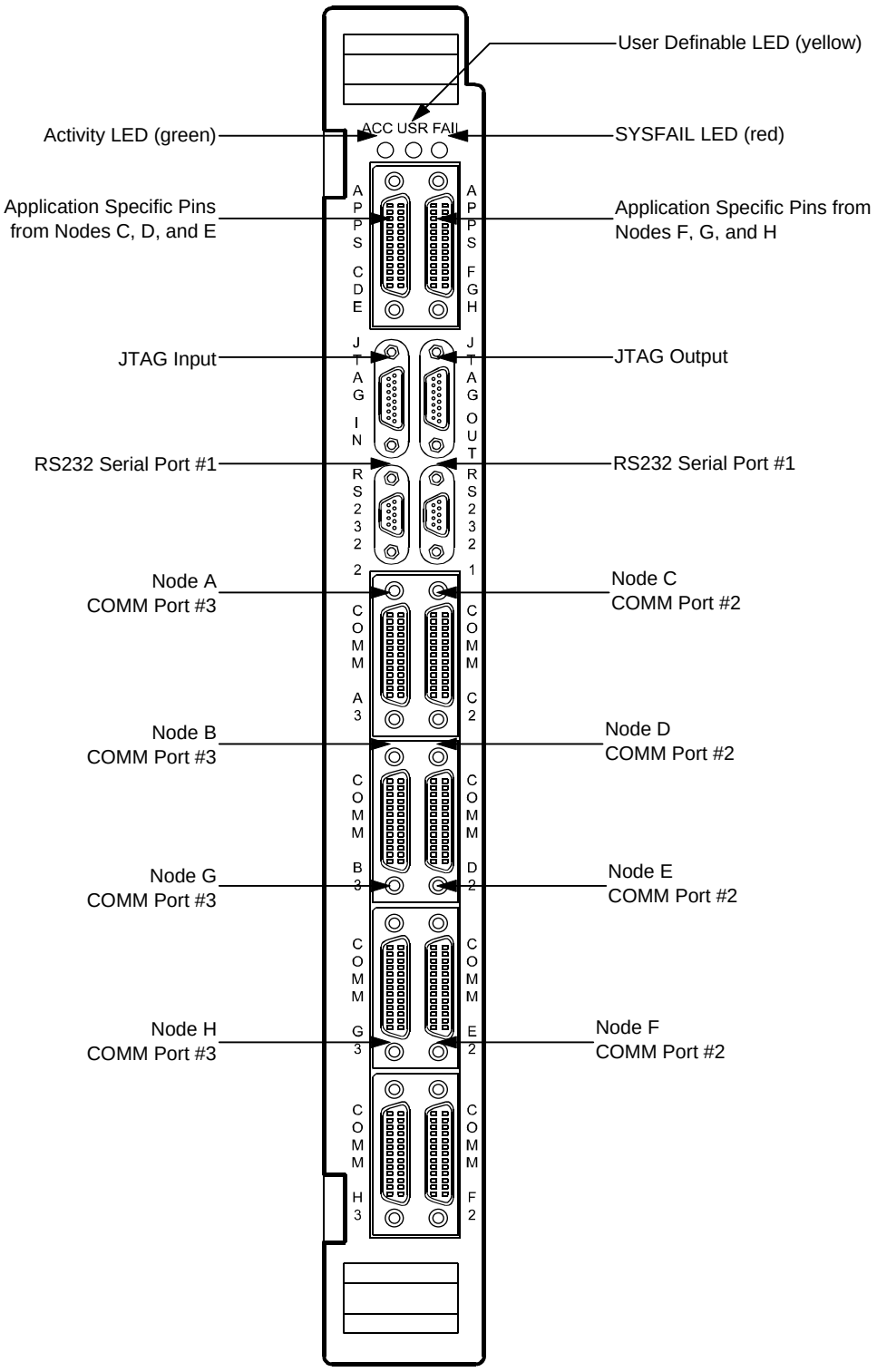


Figure 2 VX8 Front Panel

2.3.1. Status LEDs

LED	Color	Description
ACC	Green	VXIbus Activity LED. ON when there is activity between the VXIbus and the VX8 Carrier Board .
USR	Yellow	User Definable LED. ON when bit 0 (D0) of the LED register is set to "0". This write-only register is located at address 8B00 0000h on the Shared DRAM Bus.
FAIL	Red	VXIbus SYSFAIL LED. The LED is driven when the SCV64 chip drives the VXI SYSFAIL line.

2.3.2. Connectors

JTAG IN Texas Instruments' XDS510 or Spectrum's XDS3040 can be connected to the JTAG IN connector for use with a debug monitor.

JTAG OUT The JTAG OUT connector allows the VX8 to be part of a multi-module JTAG path.

RS232 Serial Ports Two RS232 Serial ports are supported by the Node B embedded 'C40 DSP.

Communication Ports One communication port from each of the Nodes is brought to the front panel via these connectors.

2.4. C4x Communication Port Architecture

The C4x Communications ports provide high speed parallel interface communications (~20 Mbytes/sec) to other DSP's and I/O sources. The communication is inherently bi-directional and point to point so there is no latency for access and a single COMM port can be used for half duplex communication between two devices.

The TMS320C40 provides 6 COMM Ports and the TMS320C44 provides 4 COMM Ports. COMM Port routing on the VX8 Carrier Board accounts for fewer COMM Ports on a C44 by ensuring that the front panel connections are valid for Spectrum's C40 and C44 based TIM-40 Modules.

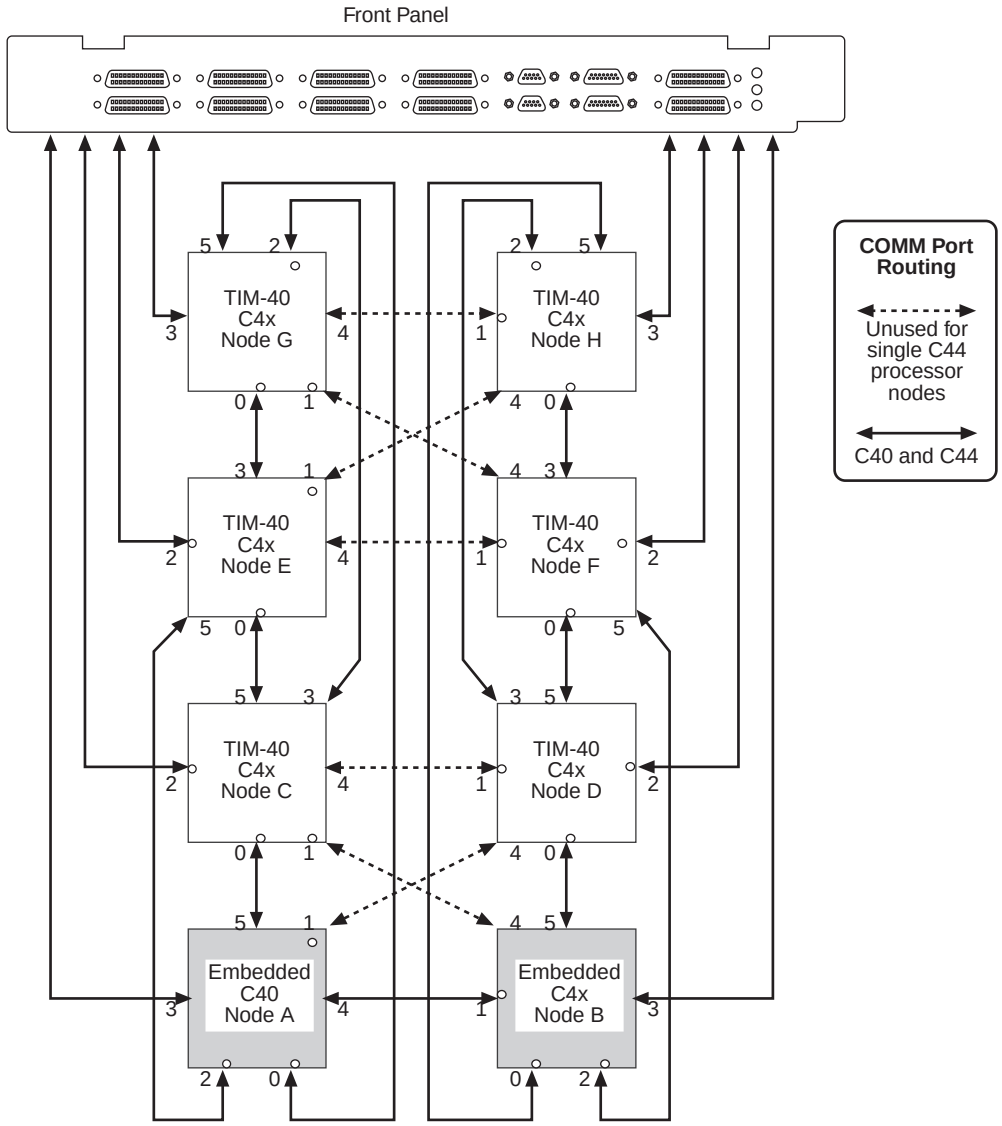


Figure 3 Communication Port Routing

Although the C44 does not use COMM ports 0 and 3, Spectrum's C44 based TIM-40 modules route COMM ports 1 and 4 to COMM ports 0 and 3 for compatibility with existing motherboard designs. As a result, COMM Ports 1 and 4 of single C44 based TIM-40 modules are not available. The COMM Port layout shown in Figure 4 ensures that the front panel COMM ports are valid for all current and planned Spectrum C40 and C44 based TIM-40 Modules.

Refer to the *TMS320C4x User's Guide* for further information on the C4x COMM Ports.

2.4.1. External Port Protocols

The comm ports on this board's front panel are 8 bit wide, buffered C4x ports that use a hardware handshaking (asynchronous) protocol to transfer data. The data is transferred from the source port to the target port using the STRB and RDY signals. The sequence of events for a typical data transfer is shown below:

1. Source port places data on data lines and asserts the STRB line low.
2. Target port sees that the STRB line is low and asserts the RDY line low in response.
3. Source port sees that RDY line is low, releases the STRB line (back high) and sends data across data lines.
4. Target port receives data and releases the RDY line (back high)

This handshaking process is transparent to the software.

Each comm port, although bi-directional, is set default as either a source or target. To prevent contention, source ports may only be connected to default target ports. All interconnected C4x processors must share the same RESET signal. This avoids contention when comm ports revert to their default direction after a reset.

When data needs to flow in the opposite direction, another handshaking protocol is used to reverse the data flow on the comm port. The current source comm port is said to "have the token" and the current target port needs to "request the token". This transfer is made using the REQ and ACK signals in the following manner:

1. Target port asserts the REQ line low.
2. Source port sees that the REQ line is low and asserts the ACK line low in response.
3. Target releases the REQ line (back high) and starts to drive the data lines (and swaps the drives on all handshake lines).
4. Source sees the REQ line go high, releases the ACK line (back high) and ceases driving the data lines (and swaps the drives on all handshake lines).

There is a short period of overlap on the line drives, but since all of the lines (data and handshake) are high at the time and the drivers are both driving high as well, there is no contention. This token swapping is done only when no data bytes are being transferred.

The comm port transactions are asynchronous and with 60MHz processors the throughput has been measured at ~15Mbytes/sec. Without buffering or routing length, the theoretical maximum comm port speed is half the processor clock rate. Comm ports can communicate between processors of different clock speeds if the clocks are within a ratio of 1:2 difference.

2.4.2. Bus Architecture

Several different communication buses are used on the VX8 Carrier board to connect the C40 processors, TIM-40 sites, memory devices, and interface circuitry. Although the buses are not the only way that devices are interconnected on the VX8, it is the primary means of data transfer between devices.

Local Bus

The Local Bus address range is specific to a single C4x DSP, and is therefore not shared with other processors or nodes. It is a private memory bus of a particular C4x.

Near Global Bus

The Near Global Bus of the VX8 refers to the Global Bus of each TIM-40 site and the embedded C40 nodes. The SRAM located on these Global Buses is zero wait state from the DSP that owns it, but can be accessed by other DSP's, the HP Local Bus DMA Controller, and the VXIbus Slave Interface via the Global Shared Bus.

Global Shared Bus

The Global Shared Bus interconnects the:

- Buffered Global Buses of each TIM-40 site via the Global Connectors;
- Buffered Global Buses of the embedded C40 nodes A and B;
- DRAM Shared Bus; and
- HP-Local Bus Interface and registers.

32-bit buffers isolate the Global Shared Bus from all these areas except for the HP-Local Bus Interface, which is connected to the bus through a 1k x 32-bit FIFO.

DRAM Shared Bus

The DRAM Shared Bus enables the VXIbus slave interface to access the DRAM, Test Bus Controller, and the Global Shared Bus. It also allows a C4x DSP to access the DRAM, control / status registers, and the SCV64 as a VXI master. Two 72-pin SIMM sites allow expansion of the global shared DRAM using standard PC DRAM memory modules.

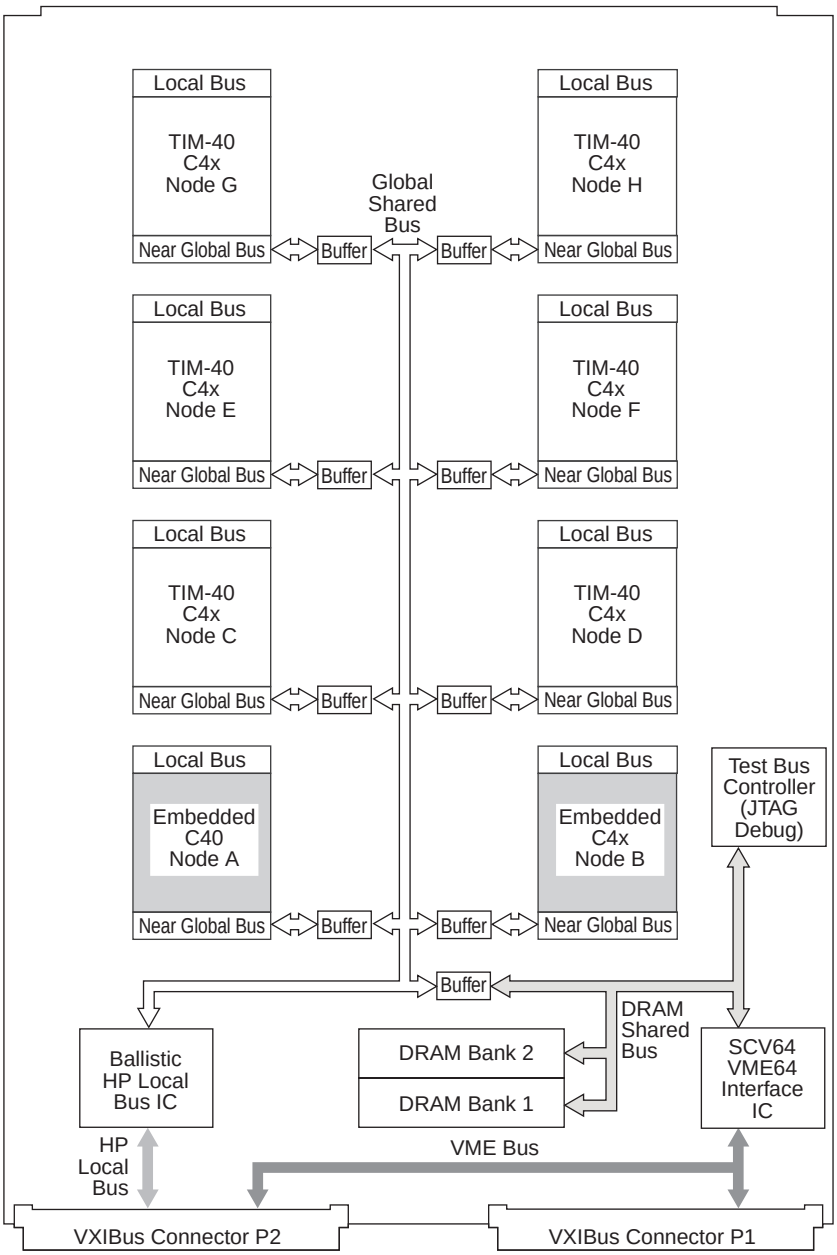


Figure 4 Bus Architecture

2.5. C4x Interrupt Architecture

The four configurable IIOF lines from each ‘C4x are used for interrupts:

- Between other C4x nodes on the board
- From the SCV64 VXIbus interface chip
- From the HP Local Bus interface
- From the Dual 16550 UART (DUART)
- From the VXIbus A16 Interface

The following figure shows the VX8 interrupt architecture.

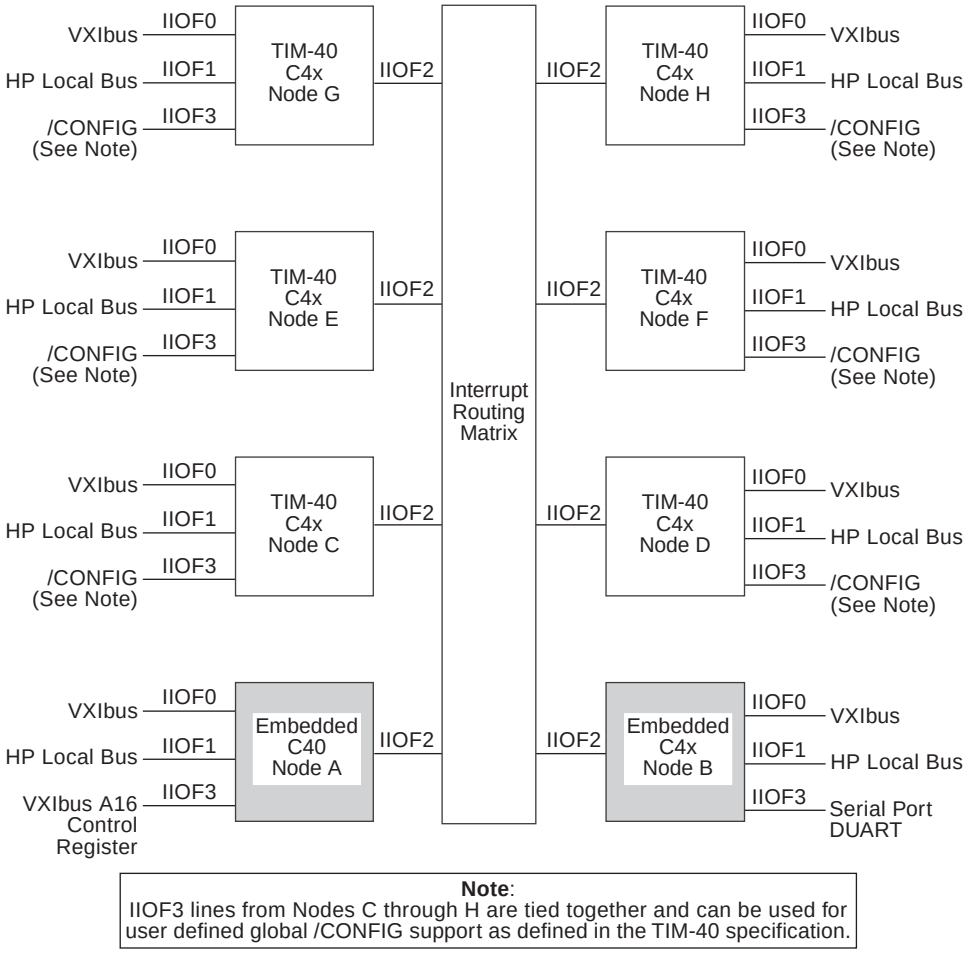


Figure 5 VX8 Interrupt Architecture

2.5.1. VXIbus Interrupts (IIOF0)

Interrupts from the SCV64 VXIbus interface chip are mapped to IIOF0 of all the C4x nodes. These level-triggered interrupts indicate that either a VXIbus interrupt has occurred or that one of the on-board SCV64 interrupt sources has occurred (SCV64 timer for example). The application program must determine the source of the interrupt. VXI interrupts can be enabled or disabled based on their interrupt level through registers in the SCV64 VXI interface chip. This allows software selectable receipt of interrupts of different priorities. Interrupts must be enabled before they can be generated. The interrupt vector received is latched for reading by the 'C4x servicing the interrupt through the IACK space in the shared memory map. The 'C4x's interrupt service routine must produce the IACK cycle through the SCV64.

To determine the source and initiate clearing of the interrupt, an interrupt acknowledge (IACK) cycle must be performed. VX8 software functions are available to configure and acknowledge the interrupts for this. Refer to the *VX8 Carrier Board Programmer's Guide* for information on these interrupt handling functions.

Vectored interrupts can also be generated from any node to the VXIbus using the internal SCV64 register set.

2.5.2. HP Local Bus Interrupts (IIOF1)

Three interrupts from the HP Local Bus interface are mapped to IIOF1 of all 'C4x nodes:

- End of Block (EOB)
- Write FIFO Almost Empty (WAE) sent once per transition of the WAE flag
- Write FIFO Almost Full (WAF)

These interrupts are ORed together onto the IIOF1 line. Use level-triggered interrupts to identify the source of the HP-Bus interrupt. The HINTENABLE, HINTSTAT, and HINTCLR registers on the Global Shared Bus are used to enable, identify, and clear the interrupts. Refer to the *HP Local Bus Interface* section in this manual for more information on these interrupts.

2.5.3. Interrupt Routing Matrix (IIOF2)

The IIOF2 lines from each node be used as an interrupt or as a general purpose I/O to signal other 'C4x nodes. This interrupt scheme is under software control, allowing IIOF2 lines to be tied together in any combination through the Interrupt Routing Matrix. The Interrupt Routing Matrix is configured by a register located on node A's local bus. For further information on the IIOF2 Interrupt Routing Matrix, see *section 3.8 Interrupt Routing Matrix*.

2.5.4. IIOF3 Interrupts

The IIOF3 line is used for three different purposes on the VX8 depending on which node it belongs to.

Node	Interrupt	Description
A	VXIbus A16	Whenever the host writes to the VXIbus A16 Interface Control Register an interrupt is sent to the IIOF3 line of the Node A C40. For further information on the VXIbus A16 interface see *section 10.4 VXIbus Slave Memory Map (A16)*.
B	DUART	Node B uses IIOF3 as the interrupt from the serial port DUART. The interrupt lines from Channel 0 and Channel 1 UARTs are ORed together and routed to Node B's IIOF3. For further information on the DUART interrupt see *chapter 4*.
C to H	C4x /CONFIG	The IIOF3 lines from the TIM-40 modules (nodes C to H) are not brought off the VX8. The lines are tied together on the VX8, allowing them to be used for user-defined global configuration as defined by the TIM-40 specification..

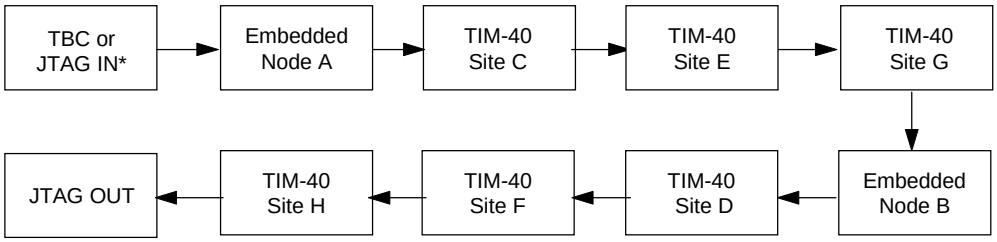
2.6. JTAG Debugging

A JTAG IN connector is provided on the front panel of the board for connection to an XDS510 from Texas Instruments or a DBC3040 from Spectrum. This allows use of the Texas Instruments' standard TMS320C4x debug monitor or third party debug monitors such as GO DSP's Code Maestro from an external PC or SUN workstation.

A JTAG OUT connector allows the VX8 to be part of a multi-module JTAG path. The open collector /CONFIG and /GRESET signals are bussed between boards via the JTAG connectors. The JTAG cable allows multi-board resetting (required if the front panel COMM ports are connected between boards) and /CONFIG to be bussed between devices.

Note: Because of the directional reset state of C4x COMM ports, all VX8s connected together via COMM port MUST be connected by their front panel JTAG connectors. The front panel JTAG connectors will reset JTAG connected VX8s together, thereby damage to COMM ports on reset will be avoided.

Each of the C4x processors has a JTAG interface for debugging purposes. The JTAG chain is controlled by the JTAG PAL, which routes the data lines to each available C40 node in turn. If a TIM site is not occupied then the JTAG chain bypasses that node. The full JTAG sequence is JTAG IN, Node A, C, E, G, B, D, F, H, JTAG OUT, as shown in the following figure. For multiple processor TIM-40 modules, refer to the TIM-40 module documentation for information on the order in which the processors are connected in the JTAG scan path. When an external debugger is connected to the JTAG IN connector of a board, the on-board Test Bus Controller is disabled.



*The Test Bus Controller (TBC) is disabled (bypassed) if an external debugger is connected to the JTAG IN connector.

Figure 6 JTAG Chain

For further details on the JTAG interface, refer to *the VX8 Carrier Board Programming Guide*.

The JTAG cable from the external debugger should be connected only to the JTAG IN of board 1. For multiple boards, the JTAG OUT of board 1 is connected to the JTAG IN of board 2, and so on.

Note: The hardware must be powered down before the JTAG chain is set up and the JTAG cables are connected.

---

3 Embedded C40 Node A

Node A consists of a 60 MHz TMS320C40 DSP embedded (installed) on the VX8 Carrier Board to function as a virtual TIM-40 site. Its features include:

- Two banks of zero wait state 128k x 32 SRAM. One bank is on the local bus and the other is on the global bus for a total of 1 MBytes per C40. The SRAM is upgradeable to 512k x 32 SRAMs at the factory
- Two 32k x 8 PEROMs on the local bus. One is used for TIM-40 defined IDROM support and the other is used for the Node A Boot kernel, providing VXIbus and code download support
- Global bus signals routed to buffers to allow for HP Local Bus DMA controlled data writes to global SRAM
- /CONFIG register for TIM-40 support
- A16 Control / Status register support to provide VXIbus A16 support on Node A's local bus
- The capability to write to the HP Local Bus output FIFO or to access the shared global DRAM through the global bus connector
- The capability to access the SCV64 IC to act as a VXIbus master
- Interrupt Route Matrix control register interface on Node A's local bus
- HP Local Bus DMA /Suspend support on Node A's local bus to allow stalling of DMA transfers for Node A access to Global Shared or DRAM Shared Bus (may be used to service VXIbus interrupts, preventing VXIbus timeouts and BUSERR)

3.1. Node A Block Diagram

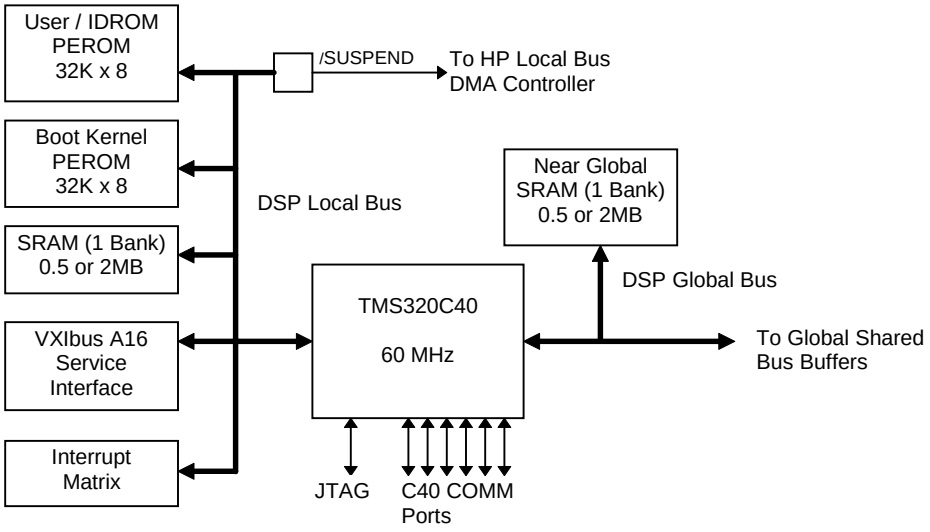


Figure 7 Node A Block Diagram

3.2. Node A Local Bus

3.2.1. Memory Map

Table 1 Node A Local Bus Memory Map

Strobe	C40 Address	Access
LSTRB0	0000 0000h 02FF FFFFh	Reserved for Internal C40 SRAM, Registers and Boot Loader
	0030 0000h 0FFF FFFFh	Local SRAM (128K or 512K x 32) and aliases
	1000 0000h	VXIbus A16 Slave Register Interface
	2000 0000h	Node A /CONFIG register
	3000 0000h	IIOF2 Enable / Routing Register
	4000 0000h	Boot Kernel PEROM Location
	5000 0000h	Reserved
	6000 0000h	HP Local Bus /SUSPEND Register
LSTRB1	7000 0000h	ID PEROM Location

3.2.2. Local Memory Interface Control Register

The LMICR value for the Node A C40 is **0x3D840000** for normal operation.

Table 2 Node A Local Memory Interface Control Register

Register Bits	Value	Description
STRB Switch	1	Inserts a cycle between STRB0 and STRB1 Reads
STRB Active	11101	0000 0000h to 3FFF FFFFh on LSTRB0, 4000 0000h to 7FFF FFFFh on LSTRB1
LSTRB1 Pagesize	10000	128 kwords page/bank size
LSTRB0 Pagesize	10000	128 kwords page/bank size
STRB1 WTCNT	000	Number of SW wait states - not used
STRB0 WTCNT	000	Number of SW wait states - not used
STRB1 SWW	00	External wait states only
STRB0 SWW	00	External wait states only

3.3. Node A Global Bus

3.3.1. Memory Map

Table 3 Node A Global Memory Map

Strobe	C40 Address	Access	Bus
STRB0	8000 0000h 87FF FFFFh	Near Global SRAM (128K or 512K x 32) and aliases. Unbuffered, zero wait state access from Node A C40.	Near Global RAM
STRB1 accesses to Global Shared and DRAM Shared Bus	8800 0000h	SCV64 Register Set	DRAM Shared BUS
	8900 0000h	VSTATUS Register	
	8A00 0000h	VCONTROL Register	
	8B00 0000h	LED0 Register (Write only)	
	8B00 0000h	Reserved	
	8C00 0000h	Reserved	
	8D00 0000h	Reserved	
	8E00 0000h	VXI IACK Space	
	8F00 0000h	Reserved	
	9000 0000h	HP Local Bus Register & WRITE FIFO Interface	Global Shared Bus
	9100 0000h	Reserved	
	9200 0000h	Reserved	
	9300 0000h	Node B Global Memory Space*	
	9400 0000h	Node C Global Memory Space *	
	9500 0000h	Node D Global Memory Space *	
	9600 0000h	Node E Global Memory Space *	
	9700 0000h	Node F Global Memory Space *	
	9800 0000h	Node G Global Memory Space *	
	9900 0000h	Node H Global Memory Space *	
	9A00 0000h	Reserved	
	9B00 0000h 9FFF FFFFh	Reserved	
	A000 0000h BFFF FFFFh	Shared DRAM plus Reflections	DRAM Shared Bus
	C000 0000h FFFF FFFFh	VXIbus Address Space C40 as VXIbus Master	

* A C40 cannot access it's own Near Global Memory with STRB1 on the Global Shared Bus. Therefore, Node A cannot access its own near global SRAM at 0x92000000 of the Global Shared Bus.

3.3.2. Global Memory Control Register

The GMCR value for the Node A C40 is **0x3A4C 0000**

Table 4 Node A Global Memory Control Register

Register Bits	Value	Description
STRB Switch	1	Inserts a cycle between STRB0 and STRB1 Reads
STRB Active	11010	8000 0000h to 87FF FFFFh on STRB0, 8800 0000h to FFFF FFFFh on STRB1
STRB1 Pagesize	01001	1 kwords page/bank size (required for DRAM pages)
STRB0 Pagesize	10000	128 kwords page/bank size
STRB1 WTCNT	00	Number of SW wait states - not used
STRB0 WTCNT	00	Number of SW wait states - not used
STRB1 SWW	00	External wait states only
STRB0 SWW	00	External wait states only

3.4. Node A Boot Kernel PEROM

A 32kx8 ATMEL Programmable Erasable Read Only Memory is used for Node A booting. It has been programmed to contain the boot kernel that supports VXIbus A16 control and offset register commands. It also provides SCV64 initialization for A32 base address and support for VX8 Carrier Board code download from the host. A jumper is used to prevent accidental erasure or corruption of this PEROM, and the default condition is for Boot Kernel PEROM writes disabled.

Caution: If the Boot Kernel PEROM is overwritten or corrupted, the VX8 Carrier Board will not respond to VXIbus host Resource Manager commands. As a result, the Resource Manager will assert the VXIbus A16 reset bit and the board will be entirely non-functional. The user would have to manually release the VXIbus A16 reset bit in the A16 Control Register and use an external debugger card with pod (XDS-510 / XDS3040) to re-burn the correct code into the Boot Kernel PEROM.

3.5. Node A IDROM PEROM

A second PEROM is included at 0x7000 0000 on the Node A C40 local bus to provide virtual TIM-40 support. Its use is entirely at the discretion of the user and could be used for TIM-40 IDROM information, tables, etc.

3.6. Node A Local Bus SRAM

The Node A Local Bus SRAM consists of one bank of 128kx32 or 512kx32 zero wait state SRAM located at 0x300000. This provides contiguous memory support between C40 internal RAM and Bank 1 of external SRAM.

3.7. Node A IIOF Lines

The four IIOF lines of the Node A C40 are assigned and configured according to the following table. User applications running on Node A must configure the 'C40's Interrupt Flag Register (IIF) accordingly for interrupts to be received.

Table 5 Node A IIOF Lines

Line	Source	Configuration	Comment
IIOF0	SCV64 VXIbus Interface	Level-triggered interrupts enabled	See *VX8 Carrier Board Programmer's Guide* for information on interrupt handling functions.
IIOF1	HP Local Bus Interface	Level-triggered interrupts enabled	See *chapter 8.*
IIOF2	Interrupt Routing Matrix	User-defined	See *chapter 3*.
IIOF3	VXIbus A16 Control Register	**Do not configure**	Initialized by the Boot Kernel PEROM upon start up.

3.8. Interrupt Routing Matrix

The IIOF2 output of one node can be tied to the IIOF2 inputs of one or more other nodes through the Interrupt Routing Matrix. This inter-processor interrupt capability allows un-buffered connection between the IIOF2 lines of two or more processors. The routing matrix is configured and enabled via the IIOF2 Routing Matrix Register located at address 3000 0000h on the local bus of Node A's embedded C40. The matrix supports a number of configurations allowing a processor to interrupt one or more other processors with support for bi-directional interrupts.

The Interrupt Routing Matrix consists of eight 8:1 analog multiplexers — one for the IIOF2 line of each 'C4x node — connected as shown in the following figure. The multiplexers provide eight signal lines that can be used to tie the IIOF2 lines to each other. The Interrupt Routing Matrix register configures the multiplexer for each node. Analog switch technology makes this interrupt scheme inherently bi-directional. Therefore, you must ensure that two or more DSPs that are connected together cannot assert IIOF2 at the same time.

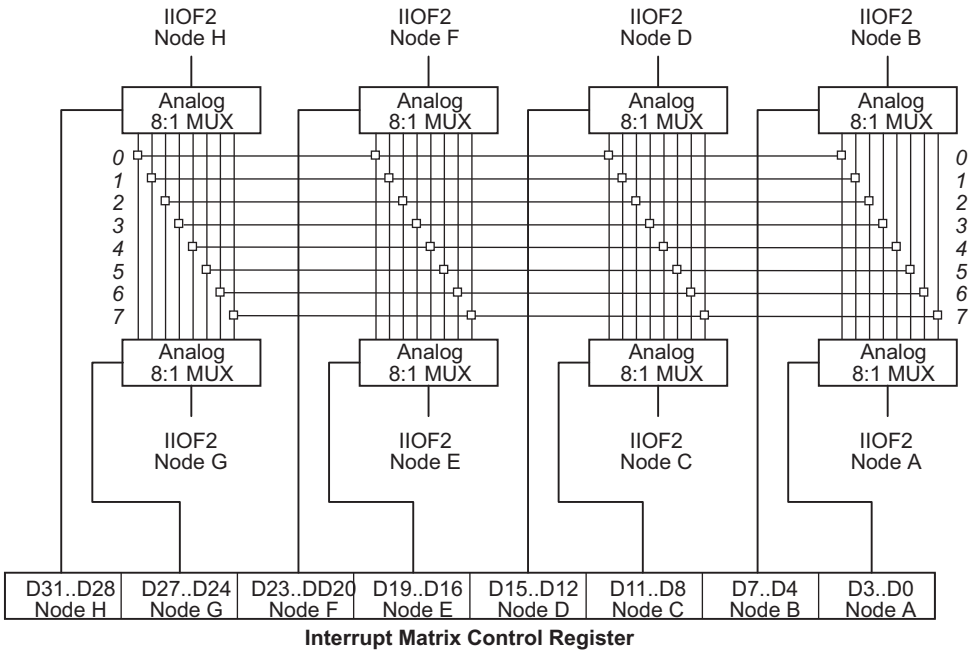


Figure 8 Node A Interrupt Routing Matrix

Caution: When connecting the IIOF2 lines together, only **one** IIOF2 line can be configured as an output. If two or more outputs are tied together the 'C40 DSPs can be damaged. The routing matrix does not prevent this and the analog multiplexers provide a physical 5Ω connection. Therefore, you must ensure that the IIOF2 lines of each 'C4x are properly configured (inputs or outputs) before enabling the interrupt routing matrix.

There are three select bits and one enable bit for each processor node in the IIOF2 Interrupt register corresponding to the select and enable lines for the multiplexors connected to the IIOF2 lines of each node as shown.

D31	D30	D29	D28	D27	D26	D25	D24
Node H /EN	Node H S2	Node H S1	Node H S0	Node G /EN	Node G S2	Node G S1	Node G S0

D23	D22	D21	D20	D19	D18	D17	D16
Node F /EN	Node F S2	Node F S1	Node F S0	Node E /EN	Node E S2	Node E S1	Node E S0

D15	D14	D13	D12	D11	D10	D9	D8
Node D /EN	Node D S2	Node D S1	Node D S0	Node C /EN	Node C S2	Node C S1	Node C S0

D7	D6	D5	D4	D3	D2	D1	D0
Node B /EN	Node B S2	Node B S1	Node B S0	Node A /EN	Node A S2	Node A S1	Node A S0

S0, S1, and S2 Select Bits The three select bits of each processor select which of the eight multiplexed signals are routed to its IIOF2 line.

/EN Enable Bit The enable bit (active low) of each processor determines if the multiplexor is turned on (low) or tri-stated (high).

3.8.1. How to Set Up the Interrupt Routing Matrix

To set up the Interrupt Routing Matrix the Interrupt Flag Register (IFF) of each of the ‘C4x DSPs and the Node A Interrupt Routing Matrix register must be configured.

Once the structure of the interrupt routing matrix has been determined so that you know which IIOF2 lines will be tied together and which processor will be generating the interrupt, configure the matrix according to the following steps.

Caution: Ensure that only **one** IIOF2 is configured as an output for a group of IIOF2 lines that are tied together. All other lines must be configured as IIOF2 inputs. The ‘C4x processors can be damaged if multiple IIOF2 lines configured as outputs are connected together.

---

1. Configure the ‘C4x IFF register of each node so that the IIOF2 lines are level-triggered interrupt *inputs*, except for the ‘C4x generating the interrupt. Configure this processor as a level-triggered interrupt *output*. Refer to *TMS320C4x User’s Guide* for information on this register.

2. Configure the Interrupt Routing Matrix Register located at 3000 0000h on Node A’s local bus to connect the IIOF2 lines of two or more processors together.

    - One of the eight multiplexed lines is chosen and its 3-bit value is written to the S0, S1, and S2 bits of each of the nodes to be connected.
    - The enable bit is set low to enable the connection for each node being connected.

3.8.2. Interrupt Routing Matrix Examples

The following examples show how the Interrupt Routing Matrix can be configured.

Example One This first example shows a simple configuration with Node A generating interrupts to be received by Node B.

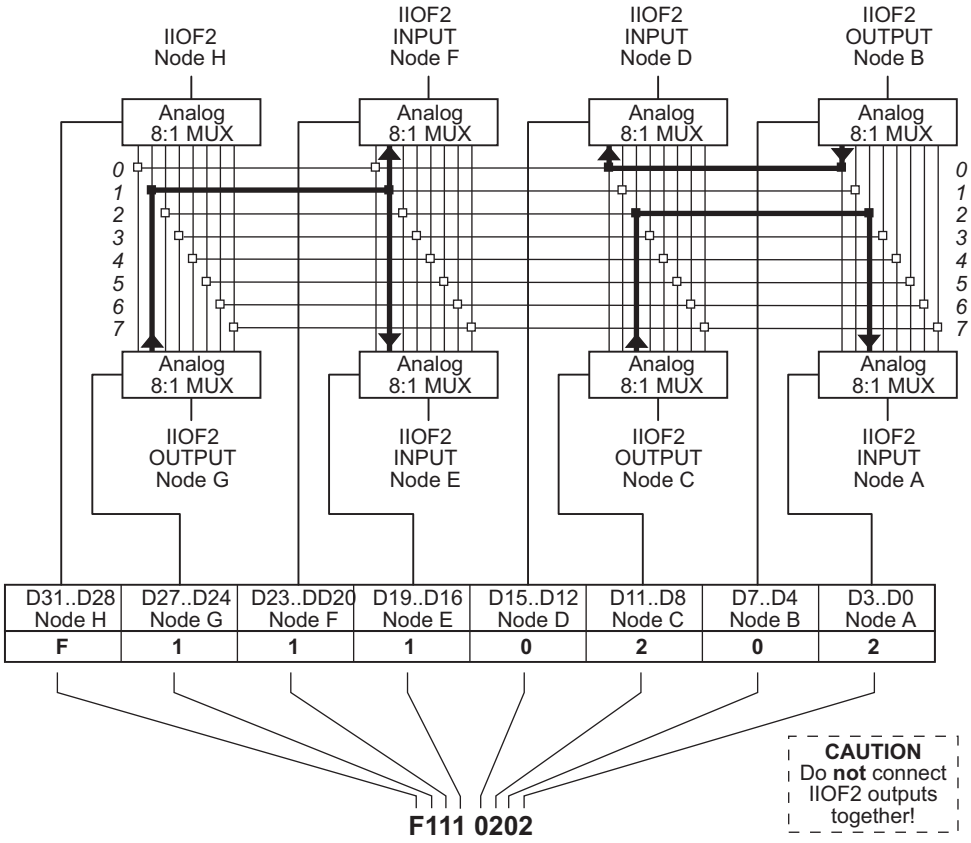


Figure 9 Interrupt Routing Matrix Example One

- The IIF registers of Nodes A and B are set so that their IIOF2 lines are level-triggered interrupts. Node A is configured as the output; Node B is configured as the input.

- The value “FFFF FF00” is written to the Interrupt Routing Matrix register. This ties the IIOF2 lines of nodes A and B together on multiplexer signal line 0. The Interrupt Routing Matrix is disabled for all other nodes as indicated by the value “F”.

Example Two The second example shows a more complex configuration.

- Node B generates interrupts to Node D.
- Node C generates interrupts to Node A.
- Node G generates interrupts to Nodes E and F.

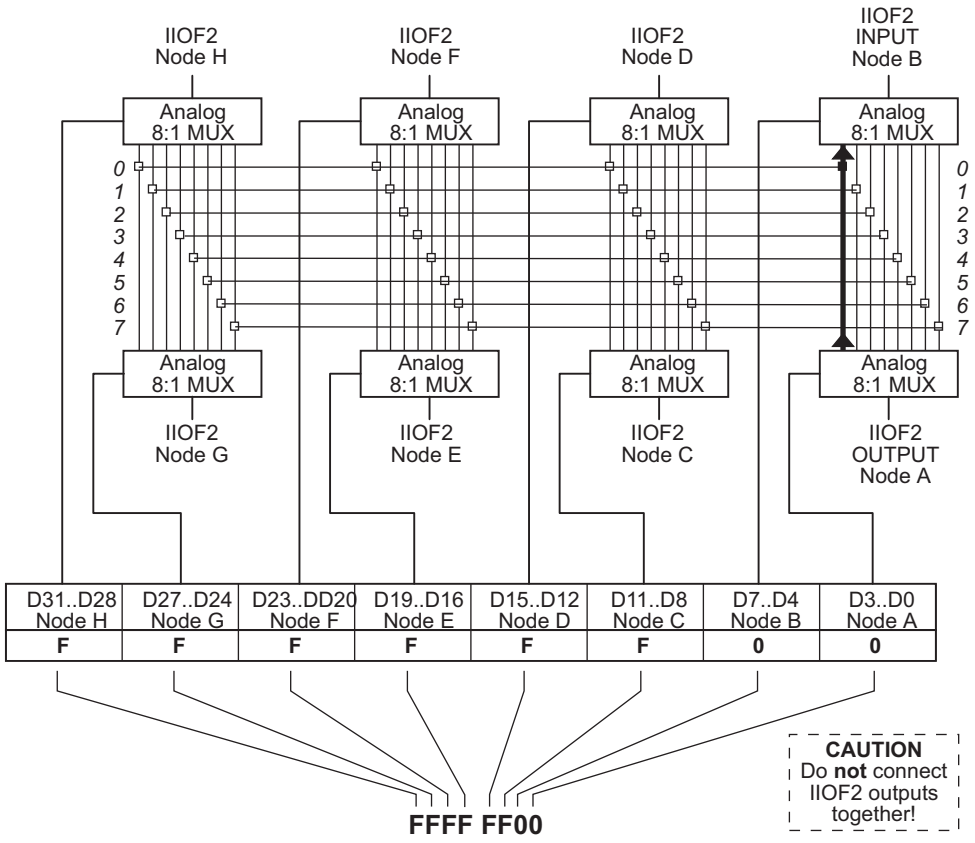


Figure 10 Interrupt Routing Matrix Example Two

- The IIF registers of Nodes A, B, C, D, E, F, and G are set so that their IIOF2 lines are level-triggered interrupts. Their direction is set as follows:

Input/Output	Nodes
Output	B, C, G
Input	A, D, E, F.

Note that only one node from each group of connected nodes is configured as an output.

- The value “F111 0202” is written to the Interrupt Routing Matrix register. This causes the nodes to be connected to the following multiplexer signal lines:

Node	A	B	C	D	E	F	G	H
Multiplexer Signal Line	2	0	2	0	1	1	1	none

The Interrupt Routing Matrix is disabled for node H as indicated by the value “F”.

Further examples of Interrupt Matrix Register values are given in the following table.

Table 6 Interrupt Routing Matrix Examples

IIOF2 Interrupt Matrix Register Value	**IIOF2 physically inter-connected Nodes**
FFFF FFFFh	None
FFFF FF00h	Nodes A connected to B
FFFF F000h	Nodes A, B, and C connected (allowing A to interrupt B and C for example)
FFFF 0011h	A connected to B, C connected to D
0011 2233h	A connected to B, C connected to D, E connected to F, G connected to H
0000 0000h	All nodes connected (allowing any one node to interrupt all nodes)

3.9. VXIbus A16 Slave Register Interface

Although the SVC64 device interfaces the VXIbus to the VX8 Carrier Board, A16 requests from the VXIbus hosts are serviced through the A16 register interface residing on Node A’s local bus. When the host writes to the A16 registers of the VX8 Carrier Board, the Node A C40 is interrupted on IIOF3. The C40 reads the VXIbus A16 control and offset registers and takes appropriate action, writing to the A16 status register if necessary.

To simplify this access, the VXIbus A16 registers can be read directly from Node A’s Local Bus. The VXIbus A16 interface is entirely under software control; the boot kernel services these interrupts before application code is downloaded, and the user must link in the A16 interrupt service routine and enable the interrupt signal IIOF3. The register addresses and descriptions are as follows:

Table 7 Node A VXI A16 Registers

C40 Address	**Register**
1000 0002h	Read A16 Control Register Write A16 Status Register
1000 0003h	Read A32 Offset Register Write clears the A16 interrupt to Node A

Complete descriptions of these registers can be found in section *10.5 VXIbus Slave Memory Map (A16).*

3.10. /CONFIG Register

The Node A /CONFIG Register provides support for the TIM-40 specified open collector /CONFIG line. Upon power up or reset on board logic pulls the /CONFIG line low on the board. Because it is open collector, the line will only go high when all C40's have released their individual /CONFIG signals. The Node A C40 releases its /CONFIG line by writing a 1 to address 2000 0000h.

This is a user application function for TIM-40 modules that does not affect the normal operation of the VX8.

3.11. HP Local Bus DMA /SUSPEND Register

VXIbus service requests can time out due to HP Local Bus DMA transfers that monopolize the VX8. If these DMA transfers deny Node A access to the Global Shared Bus when Node A is trying to get from the Global shared Bus to the DRAM bus to service the VXIbus interrupt, a VXIbus timeout can occur. This can be prevented by using the /SUSPEND register on Node A's local bus to force the DMA Controller to relinquish the Global Shared Bus after completing its current cycle. This gives Node A access to the Global Shared Bus and the DRAM Bus to service the interrupt. To do this, the /SUSPEND line to the DMA controller is asserted. The /SUSPEND line is controlled by the state of the /SUSPEND register located at address 6000 0000h of Node A's local bus.

To assert the /SUSPEND line, write the value 0x1 to this register. To release the line, write the value 0x0 to the register. The HP Local Bus DMA Controller reacquires the Global Shared Bus and continues the operation that was in progress prior to /SUSPEND being asserted.

The /SUSPEND Register has no effect on other C4x DSP's or VXIbus Slave accesses to TIM-40 or embedded node near global SRAM accesses on the Global Shared Bus.

Caution: Both HP Local Bus and VXIbus transfers through the SCV64 have priority over traffic from the C4x nodes. Therefore, HP Local Bus and VXIbus transfers can monopolize the VX8 to the point that C4x nodes are denied access to the Global Shared Bus. This can prevent VXIbus interrupts from being acknowledged by the C4x nodes. The /SUSPEND register can be used to prevent HP Local Bus DMA transfers from monopolizing the VX8. C4x *lock out* due to VXIbus transfers can be minimized through configuration of the SCV64.

---

4 Embedded C40 Node B

Node B consists of an embedded 60 MHz TMS320C40 DSP implemented as a virtual TIM-40 site on the VX8 Carrier Board. Its features include:

- Two banks of zero wait state 128k x 32 SRAM. One bank is on the local bus and the other is on the global buses for a total of 1 MBytes per C40. The SRAM is upgradeable to 512k x 32 at the factory
- One 32k x 8 PEROM on the local bus for TIM-40 defined IDROM support
- Global bus signals routed to buffers to allow for HP Local Bus DMA controlled data writes to global SRAM
- /CONFIG register for TIM-40 support
- The capability to write to the HP Local Bus output FIFO or to access the shared global DRAM through the global bus connector
- The capability to access the SCV64 IC to act as a VXIbus master
- Support of two RS-232 serial ports via a dual 16C550 UART

4.1. Node B Block Diagram

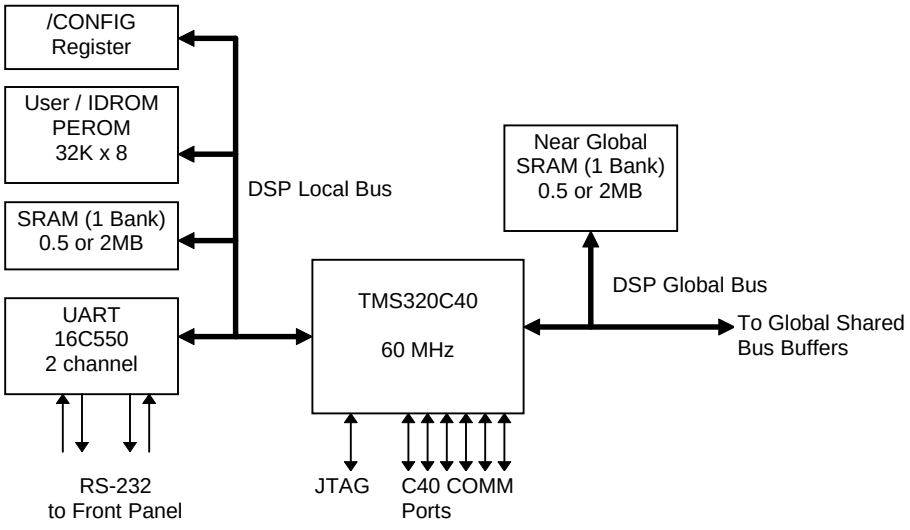


Figure 11 Node B Block Diagram

4.2. Node B Local Bus

4.2.1. Memory Map

Table 8 Node B Local Memory Map

Strobe	C40 Address	Access
LSTRB0	0000 0000h 002F FFFFh	Reserved for Internal C40 SRAM, Registers and Boot Loader
	0030 0000h 0FFF FFFFh	Local SRAM (128K or 512K x 32) and aliases
	1000 0000h 1FFF FFFFh	Reserved
	2000 0000h 3FFF FFFFh	Node B /CONFIG register
	4000 0000h 4FFF FFFFh	PEROM Boot Location
	5000 0000h 5FFF FFFFh	16C552 DUART Registers
	6000 0000h 6FFF FFFFh	Reserved
LSTRB1	7000 0000h 7FFF FFFFh	ID PEROM Location (Alias of 0x4000 0000)

4.2.2. Local Memory Interface Control Register

The LMCR value for the Node B C40 is **0x3D840000** for normal operation.

Table 9 Node B Local Memory Interface Control Register

Register Bits	Value	Description
STRB Switch	1	Inserts a cycle between STRB0 and STRB1 Reads
STRB Active	11101	0000 0000h to 3FFF FFFFh on LSTRB0, 4000 0000h to 7FFF FFFFh on LSTRB1
LSTRB1 Pagesize	10000	128 kwords page/bank size
LSTRB0 Pagesize	10000	128 kwords page/bank size
STRB1 WTCNT	000	Number of SW wait states - not used
STRB0 WTCNT	000	Number of SW wait states - not used
STRB1 SWW	00	External wait states only
STRB0 SWW	00	External wait states only

4.3. Node B Global Bus

4.3.1. Memory Map

Table 10 Node B Global Memory Map

Strobe	C40 Address	Access	Bus
STRB0	8000 0000h 87FF FFFFh	Near Global SRAM (128K or 512K x 32) and aliases (unbuffered zero wait state access)	Near Global RAM
STRB1 accesses to Global Shared Bus and DRAM Shared Bus	8800 0000h	SCV64 Register Set	DRAM Shared BUS
	8900 0000h	VSTATUS Register	
	8A00 0000h	VCONTROL Register	
	8B00 0000h	LED0 Register (Write only)	
	8C00 0000h	Reserved	
	8D00 0000h	Reserved	
	8E00 0000h	VXI IACK Space	
	8F00 0000h	Reserved	
	9000 0000h	HP Local Bus Register & WRITE FIFO Interface	Global Shared Bus
	9100 0000h	Reserved	
	9200 0000h	Node A Global Memory Space*	
	9300 0000h	Reserved	
	9400 0000h	Node C Global Memory Space*	
	9500 0000h	Node D Global Memory Space*	
	9600 0000h	Node E Global Memory Space*	
	9700 0000h	Node F Global Memory Space*	
	9800 0000h	Node G Global Memory Space*	
	9900 0000h	Node H Global Memory Space*	
	9A00 0000h	Reserved	
	9B00 0000h 9FFF FFFFh	Reserved	
	A000 0000h BFFF FFFFh	Shared DRAM plus Reflections	DRAM Shared Bus
	C000 0000h FFFF FFFFh	VXIbus Address Space C40 as VXIbus Master	

* A C40 cannot access it's own Near Global Memory with STRB1 on the Global Shared Bus. Therefore, Node B cannot access its own near global SRAM at 0x93000000 of the Global Shared Bus.

4.3.2. Global Memory Control Register

The GMCR value for the Node B C40 is **0x3A4C 0000**. It is identical to the GMCR for Node A.

Table 11 Node B Global Memory Control Register

Register Bits	Value	Description
STRB Switch	1	Inserts a cycle between STRB0 and STRB1 Reads
STRB Active	11010	8000 0000h to 87FF FFFFh on STRB0, 8800 0000h to FFFF FFFFh on STRB1
LSTRB1 Pagesize	01001	1 kwords page/bank size (required for DRAM pages)
LSTRB0 Pagesize	10000	128 kwords page/bank size
STRB1 WTCNT	00	Number of SW wait states - not used
STRB0 WTCNT	00	Number of SW wait states - not used
STRB1 SWW	00	External wait states only
STRB0 SWW	00	External wait states only

4.4. Node B IDROM PEROM

A PEROM is included on the Node B C40 to provide virtual TIM-40 support. Its use is entirely at the discretion of the user and could be used for TIM-40 IDROM information, tables, etc.

4.5. Node B Local Bus SRAM

The Node B Local Bus SRAM consists of one bank of 128kx32 or 512kx32 zero wait state SRAM located at 0x300000. This provides contiguous memory support between C40 internal RAM and Bank 1 of external SRAM.

4.6. Node B IIOF Lines

The four IIOF lines of the Node B C40 are assigned and configured according to the following table. User applications running on Node B must configure the ‘C40’s Interrupt Flag Register (IIF) accordingly for interrupts to be received.

Table 12 Node B IIOF Lines

Line	Source	Configuration	Comment
IIOF0	SCV64 VXIbus Interface	Level-triggered interrupts enabled	See *VX8 Carrier Board Programmer's Guide* for information on interrupt handling functions.
IIOF1	HP Local Bus Interface	Level-triggered interrupts enabled	See *chapter 8*.
IIOF2	Interrupt Routing Matrix	User-defined	See *chapter 3*.
IIOF3	Serial Port DUART	Level-triggered interrupts enabled	

4.7. Node B /CONFIG Register

The Node B /CONFIG Register provides support for the TIM-40 specified open collector /CONFIG line. Upon power up or reset on board logic pulls the /CONFIG line low on the board. Because it is open collector, It will only go high when all C40's have released their individual /CONFIG signals. The Node B C40 releases its /CONFIG line by writing a 1 to address 2000 0000h.

4.8. Node B DUART RS-232 Communications

The two RS232 serial ports on the front panel are supported by a 16C550 Asynchronous Dual UART on Node B. The DUART is National Semiconductor's NS16C552 (or compatible) device that provides two input and two output channels.

Note: Functions are provided in the VX8 C4x Support Software Library that can be used to initialize the device, configure interrupts, and transfer data between the DSP and the DUART.

It is clocked from the BAUDCLK of the SCV64 (32 MHz / 13 or 2.461538 MHz) and provides asynchronous communications to 153.85 kbps. The highest standard data rate provided is 38.4 kbps. The 16C550 uses a standard divide by 16 on the master clock to generate its baud rate clock. A second divisor register is then used to set the input and output baud rate according to the following formula:

Baud Rate = 32 MHz / (13 * 16 * Divisor Register)

A divisor register of 1 will give 153.85 kbps, a divisor register of 4 will give 38.46 kbps (38.4 kbps with an error of 0.16%).

The UART data and control signals TX0, RX0, CTS0, and RTS0 for port 1; and TX1, RX1, CTS1, and RTS1 for port 2 are provided. Each port is brought out to the front panel through a high density 9 pin D sub-miniature connector.

Complete information on National Semiconductor's NS16C552 is given in the *PC16552D data sheet*. The registers for the device are mapped to Node B's local bus memory as shown in the following table.

Table 13 Node B DUART Registers

Port 1 Address	Port 2 Address	Register
5000 0000h	5000 0008h	• RCVR Buffer (Read), XMTR Holding Register (Write) • Divisor Latch (LSB)
5000 0001h	5000 0009h	• Interrupt Enable • Divisor Latch (MSB)
5000 0002h	5000 000Ah	• Interrupt Identification (Read), FIFO Control (Write) • Alternate Function
5000 0003h	5000 000Bh	Line Control
5000 0004h	5000 000Ch	MODEM Control
5000 0005h	5000 000Dh	Line Status
5000 0006h	5000 000Eh	MODEM Status
5000 0007h	5000 000Fh	Scratch

5 TIM-40 Nodes C to H

5.1. Introduction

The VX8 Carrier Board can accommodate single and double-width TIM-40 modules on its six TIM-40 module sites. Each site has three 80-way connectors, supporting the top, bottom and optional Global Bus connectors.

The TIM-40 module specification was developed by Texas Instruments to deliver the full performance of TMS320C4x DSP's on a standardized parallel processing element. The standard provides physical, electrical interface, and system characteristics for module development. Modules are installed on TIM-40 carrier boards which connect the modules to each other and to the host system. Spectrum offers a variety of TIM-40 modules with various configurations including single and multiple TMS320C40, single and multiple TMS320C44 with SRAM, DRAM, or specialized I/O.

Complete information on the TIM-40 specification can be found in the *TIM-40 TMS320C4x Module Specification* available from Texas Instruments.

5.2. Global Shared Bus Access

The Global Bus SRAM (the Near Global Memory) on each site is buffered to the Global Shared Bus permitting:

- DMA transfers from the HP Local Bus to the Near Global Memory of any site
- C4x writes to the HP Local Bus output FIFO
- C4x access to the shared global DRAM
- C4x access to the SCV64 VXIbus interface

In order for the C4x DSPs on the TIM-40 modules to access to the Global Shared Bus the module must:

- Use 60 MHz C4x DSPs
- Be configured to derive their clock from the VX8 Carrier Board

> **Note:** TIM-40 modules that have a jumper to select ON module or OFF module clock source must choose the OFF module clock source.

If the TIM-40 module is intended to serve as HP Local Bus DMA Target or if Near Global SRAM access from VXIbus host or another C4x processing node is desired then 60 MHz TIM-40 modules with zero wait state SRAM on the primary processor's global bus MUST be used.

Note: Secondary C4x DSP's on Twin Processor TIM-40 modules generally do not have access to the global bus connector, so they will be unable to access the Global Shared Bus. TIM-40 modules that do not support the optional Global Connector will also not have access to the VX8 system resources.

5.3. TIM-40 C40 Global Bus Memory Map

Refer to the particular *TIM-40 Module's User Manual* for details such as architecture, amount of SRAM, number of DSP's, and the capabilities of the C4x Local Bus. Some changes to the Global Memory Control Register (GMCR) of the primary processor are required for use with the VX8 carrier board. This manual does not describe the Local Bus interface memory map of C4x DSP's on particular TIM-40 modules.

Table 14 Global Memory Map for Nodes C to H

Strobe	Address	Description	Bus
STRB0 Accesses	8000 0000h 87FF FFFFh	Zero Wait State Near Global SRAM and Aliases on TIM-40 Module	Near Global SRAM
STRB1 Accesses to the Global Shared Bus or the DRAM Shared Bus	8800 0000h	SCV64 Register Set	DRAM Shared Bus
	8900 0000h	VSTATUS Register	
	8A00 0000h	VCONTROL Register	
	8B00 0000h	LED0 Register	
	8C00 0000h	Reserved	
	8D00 0000h	Reserved	
	8E00 0000h	VXI IACK Space	
	8F00 0000h	Reserved	
	9000 0000h	HP Local Bus Register & WRITE FIFO Interface	Global Shared Bus*
	9100 0000h	Reserved	
	9200 0000h	Node A Near Global Memory Space*	
	9300 0000h	Node B Near Global Memory Space*	
	9400 0000h	Node C Near Global Memory Space*	
	9500 0000h	Node D Near Global Memory Space*	
	9600 0000h	Node E Near Global Memory Space*	
	9700 0000h	Node F Near Global Memory Space*	
	9800 0000h	Node G Near Global Memory Space*	
	9900 0000h	Node H Near Global Memory Space*	
	9A00 0000h	Reserved	
	9B00 0000h 9FFF FFFFh	Reserved	
	A000 0000h BFFF FFFFh	Shared DRAM plus Reflections	DRAM Shared Bus
	C000 0000h FFFF FFFFh	VXIbus Address Space C40 as VXIbus Master	

*A C4x DSP on any TIM-40 site cannot access its own Near Global SRAM using STRB1 accesses on the Global Shared Bus.

5.4. TIM-40 IIOF Lines

The four IIOF lines of the TIM-40 C4x nodes are assigned and configured according to the following table. User applications running on these nodes must configure the Interrupt Flag Register (IIF) of each C4x accordingly for interrupts to be received.

Table 15 TIM-40 Node IIOF Lines

Line	Source	Configuration	Comment
IIOF0	SCV64 VXIbus Interface	Level-triggered interrupts enabled	See *VX8 Carrier Board Programmer's Guide* for information on interrupt handling functions.
IIOF1	HP Local Bus Interface	Level-triggered interrupts enabled	See *chapter 8* of this manual.
IIOF2	Interrupt Routing Matrix	User-defined	See *chapter 3* of this manual.
IIOF3	/CONFIG	User-defined	Typically used for TIM-40 /CONFIG support.

5.5. Global Memory Control Register

The GMCR value for the TIM-40 based C4x DSP's is **0x3A4C 0000**

Table 16 TIM-40 Global Memory Control Register

Register Bits	Value	Description
STRB Switch	1	Inserts a cycle between STRB0 and STRB1 Reads
STRB Active	11010	8000 0000h to 87FF FFFFh on STRB0, 8800 0000h to FFFF FFFFh on STRB1
LSTRB1 Pagesize	01001	1 kwords page/bank size (required for DRAM pages)
LSTRB0 Pagesize	10000	128 kwords page/bank size
STRB1 WTCNT	00	Number of SW wait states - not used
STRB0 WTCNT	00	Number of SW wait states - not used
STRB1 SWW	00	External wait states (/RDY control) only
STRB0 SWW	00	External wait states (/RDY control) only

Page Size Register

The Page Size Register is used to indicate that the hardware must perform a full RAS/CAS access to the DRAM when a DRAM page is crossed. As a result, it must be configured to the page size of the DRAM SIMMs that are installed. If no DRAM SIMMs are installed, then this register is not used.

RDY Timing The GMCR must be configured for external /RDY control on STRB0 and external RDY control on STRB1. Logic control on the VX8 Carrier Board will provide RDY timing to produce zero wait state access to SRAM on STRB0 when the DSP accesses it.

5.6. User-Defined TIM-40 Pins

Eight user-defined TIM-40 pins are routed to the front panel connectors on the VX8 Carrier Board from each TIM-40 site. Extreme care was taken during layout of the Application Specific Connector traces on the VX8 Carrier Board to ensure excellent signal integrity and immunity from digital noise on the board. The following diagram illustrates a cross-section of the physical layout of the traces.

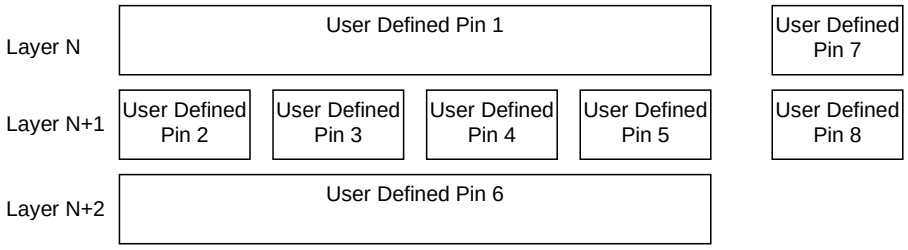


Figure 12 User-Defined Pin PCB Trace Cross Section

When designing a custom TIM-40 module, route signal grounds to user-defined pins 1 and 6. User-defined pins 7 and 8 should be used for a differential signal, or a signal + ground.

6 Global Shared Bus

Note: The VX8 C4x Support Software library provides a set of functions for accessing, locking, unlocking and transferring data along the Global Shared Bus. Refer to the *VX8 Carrier Board Programming Guide* for details.

6.1. Shared Bus Arbitration

All processing nodes on the VX8 Carrier Board have access to the Global Shared Bus and the DRAM Shared Bus. These shared buses are the gateway to system resources such as the HP Local Bus Interface, VXIbus Interface (Master), SCV64 (VXIbus Interrupts), front panel LED, DRAM, and other Processing Nodes' Near Global SRAMs. As a result, the global bus may be in demand by the HP Local Bus DMA controller, the SCV64 DMA controller, VXIbus masters, and DSPs on populated nodes.

To arbitrate shared bus requests from all sources, a bus locking mechanism is used in conjunction with the C4x interlock operations. When a device requests the Global Shared Bus, it is granted the bus according to the following priority:

- **Level 1** - The SCV64 Local bus Master, which can be either a SCV64 DMA or a VXIbus slave access to the VX8, has the highest priority. Ownership of the bus cannot be relinquished due to the VXIbus time-out. If a VXIbus Slave Access is required, then the current Global Shared Bus cycle is completed before granting the bus to the VXIbus interface.
- **Level 2** - The Local Bus is issued the global bus when ownership of the bus lies with C4x nodes. The HP Local Bus DMA Controller can give up the bus to a /SUSPEND signal issued by Node A, or a level 1 request.
- **Level 3** - C4x Read/Write access. Any C4x can request the bus by asserting its /LOCK line (VX8_Lock). The instruction cycle that asserts the /LOCK line will be extended by logic on the VX8 Carrier Board until the bus is granted to the locking DSP. The bus is issued to C4x DSPs by the arbiter in a round-robin fashion. A LOCKed C4x cannot be pre-empted by another DSP, only by a level 1 or level 2 requester. The C4x currently granted access to the bus must relinquish the bus by performing a VX8_Unlock function. After a C4x is preempted from the shared bus, ownership will be returned to it for completion of the next shared bus cycle.

Caution: C4x nodes cannot preempt one another on the Global Shared Bus. Care must be taken in your software design to ensure that processors do not starve one another on the Global Shared Bus. Processors which attempt to lock to the global bus when it is currently locked will stall until the global shared bus is unlocked by the current owner. Processors accessing their Near Global SRAM will stall while that node's Global Shared SRAM is locked by another level 1, 2, or 3 requester.

6.2. Global Shared Bus /SUSPEND DMA Transfer

The Node A embedded C40 can assert a control signal to the HP Local Bus DMA Controller that causes the DMA Controller to release the Global Shared Bus after a current cycle has completed. This allows the Node A C40 access to the Global Shared Bus to service VXIbus interrupts. Refer to the *Node A Embedded C40 Local Memory Map* for /SUSPEND register details. When the /SUSPEND control line is de-asserted, the DMA transfer will resume from the previous address. The /SUSPEND functionality does not affect the integrity of the data being transferred by DMA, but will significantly affect DMA throughput.

The /SUSPEND signal causes the HP Local Bus DMA Controller to release the Global Shared Bus. However, the /SUSPEND signal does not affect other DSPs that may have locked the Global Shared Bus, and does not affect VXIbus slave accesses to Near Global SRAM. Application software must allow Node A to access the Global Shared Bus and the DRAM Shared Bus during DMA transfers in order for VXIbus interrupts to be serviced.

HP Local Bus DMA transfers can be stalled if Node A's interrupt subroutines are delayed by other nodes that delay or lockout Node A from bus ownership.

Note: The VX8 C4x Support Software library provides a function for suspending HP Local Bus DMA transfers. Refer to the *VX8 Carrier Board Programming Guide* for details.

7 DRAM Shared Bus

7.1. Shared DRAM

The VX8 Carrier Board supports up to 2 Banks of shared DRAM in its two 72-pin SIMM sockets, SIMM 1 and SIMM 2. The SIMM sockets, which can be left un-populated, support both 4 and 32 Mbyte 60 ns SIMMs. This DRAM is accessible by any of the DSPs on the Global Shared Bus and the VXIbus slave interface. The DRAM is **not** accessible by the HP Local Bus interface.

For information on Shared Bus Arbitration, see *section 6.1* .

DRAM refresh is handled in hardware and does not require any C4x resources (IRQs, timers, or MIPs) to perform, however if a DSP tries to access DRAM when a refresh cycle is in progress, the DSP's cycle will be extended until the refresh cycle has completed.

7.2. User-Defined LED

The user-defined yellow LED is accessed through the LED register on the Shared DRAM Bus allowing it to be controlled by the application software. The write-only LED register is active low. To turn the LED on, write a "0" to the register. Refer to TIM-40 module or Embedded C40 memory maps for address information. The VX8 C4x Support Software library provides support for controlling this LED.

7.3. C40 Debug Capability

A 74ACT8990 Test Bus Controller is included on the board and support for C Source Symbolic Debugging using the Texas Instruments' standard TMS320C4x debug monitor is provided for WIN95/NT based VXI hosts. For non WIN95 hosts, or for complete non-intrusive debugging, a JTAG IN connector is provided on the front panel of the board for connection to an XDS510 from Texas Instruments or an XDS3040 from Spectrum. This allows use of the Texas Instruments' standard TMS320C4x debug monitor or third party debug monitors such as GO DSP's Code Maestro from an external PC or SUN workstation.

On board logic automatically detects TIM-40 modules and routes JTAG signals appropriately.

A JTAG out connector is also provided on the front panel for interconnection of multiple board's JTAG paths. On Board logic automatically detects and re-routes the data path if another board is connected to the JTAG out connector. The JTAG out connector allows an entire system of C4x DSPs on multiple carrier boards to be debugged without power down and hardware re-configuration.

8 HP Local Bus Interface

The HP local bus interface connects the Global Shared Bus to the HP local bus on connector P2 of the VX8. It allows the Near Global RAM of any of the C4x nodes to be accessed from the HP Local Bus

The interface consists of the BALLISTIC HP Local Bus interface chip from Hewlett Packard, read/write FIFOs, and a DMA controller. The DMA controller is used to transfer data from the BALLISTIC HP Local Bus interface to the Global Shared Bus; it is **not** used to transfer data from the Global Shared Bus to the HP local bus.

The HP Local Bus interface combines the high speed data interface between VXIbus I/O modules with a flexible DMA Controller to provide a powerful I/O solution. Hewlett Packard has a number of multi-channel analog I/O cards that communicate using HP Local Bus.

Note: The HP Bus interface does not have access to the Shared DRAM Bus (the VXIbus interface or the global shared DRAM). The VXIbus interface can not write directly into the HP Local Bus WRITE FIFO.

The VX8 C4x Support Software library provides a number of functions for accessing, configuring, and operating the HP Local Bus interface. Refer to the *VX8 Carrier Board Programming Guide* for details.

8.1. HP Local Bus Theory of Operation

The Local Bus defined in VXI systems is a daisy chained bus that connects adjacent modules through the VXI mainframe. Pins on row C of connector P2 send data to pins on row A of the next module's P2 connector along the VXI mainframe as shown in the following diagram.

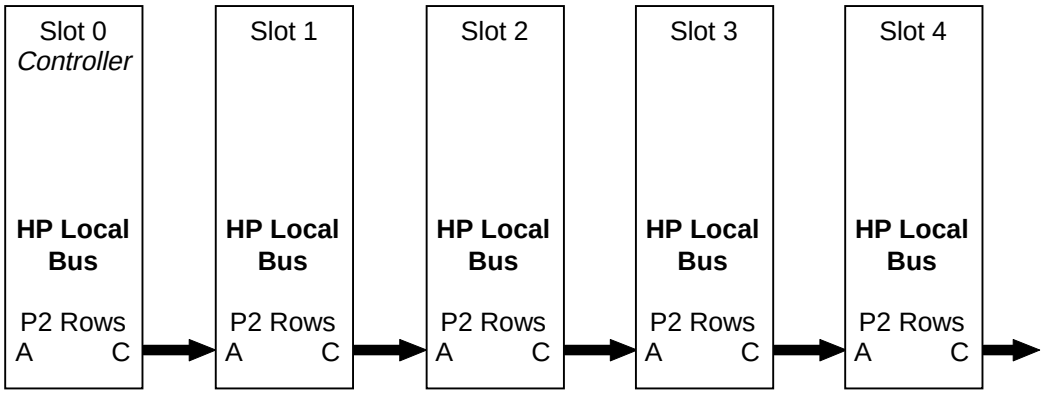


Figure 13 HP Local Bus Operation

The VXI HP local bus is a one-way bus that passes data from slot N to slot N+1. Slot 0 is the VXIbus controller and can only be a data source. In other words, it cannot receive data, but only send data to slot 1.

8.2. HP Local Bus Interface Block Diagram

A block diagram of the HP local bus interface is shown in the following figure.

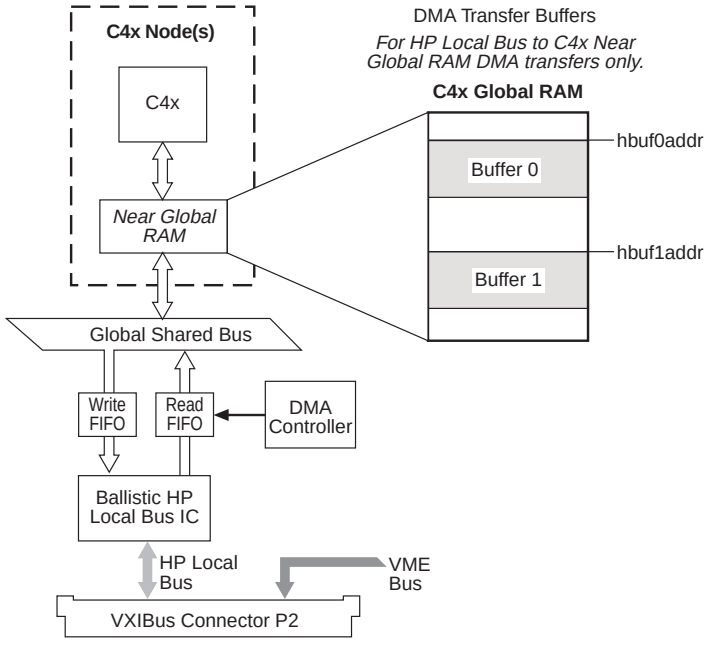


Figure 14 HP Local Bus Interface

BALLISTIC IC The BALLISTIC HP Local Bus interface chip from Hewlett Packard functions as the interface transceiver and contains the handshake logic. Hardware support is also provided for a number of control and status registers. See chapter *9, BALLISTIC HP Local Bus Implementation*, for information specific to the BALLISTIC chip.

FIFOs Each of the FIFOs is a 1k x 36 bit FIFO buffer that transfers 32 bit words to and from the Global Shared Bus. The extra 4 bits are used for End Of Block (EOB) and Frame control used by the HP local bus.

DMA Controller The DMA controller only controls the Read FIFO. Therefore, the HP local bus read operation is inherently faster than the write operation.

8.3. HP Local Bus READ DMA Operation

DMA transfers are configured and enabled by any of the C4x DSPs. In general, it is a good idea to have a single processing node responsible for initialization and interrupt servicing of the HP Local Bus interface. The setup required for DMA Operation is as follows:

1. After RESET of the HP Local Bus Interface, the READ and WRITE FIFO almost full and almost empty flag levels are configured by writing to the WRITE FIFO.
2. The HTARGET register is written to select the target of the DMA transfer. This can be one, several, or all processing nodes on the Global Shared Bus.
3. The HBUF0ADDR is written to set the start address in Near Global SRAM of the target processor(s) for the DMA Buffer 0.
4. The HBUF1ADDR is written to set the start address in Near Global SRAM of the target processor(s) for the DMA Buffer 1.
5. The HINTENABLE register is written to enable interrupt on End Of Block.
6. The DMA Enable bit is set in the HCONTROL register to start DMA transfers (once the required parameters described below are met.)

The general characteristics of the DMA Controller are:

- The DMA Controller requests the Global Shared Bus only if DMA transfers are enabled.
- The DMA Controller will **not** request the Global Shared Bus if there is a pending DMA end of block interrupt. A DSP must clear this interrupt flag for the next block DMA transfer to continue.
- The DMA Controller is typically configured to double buffer the incoming data. Block 0 is filled first, then an interrupt is sent to the DSP. The DSP clears the interrupt causing Block 1 to be filled.

- The DMA Controller will only request the Global Shared Bus if the FIFO is almost full or if an End Of Block (EOB) data signal has been received from the HP Local Bus.
- The DMA Controller will release the Global Shared Bus if the FIFO is almost empty AND an EOB has not yet been received. It will reacquire the Global Shared Bus when the FIFO reaches the almost full point OR if an EOB is received.
- If a C4x DSP is performing an access, the DMA controller will be granted the bus after the DSP completes the current cycle.
- If a VXIbus slave access to Near Global SRAM is occurring, the DMA Controller is delayed until the SCV64 releases the Global Shared Bus.
- Once taking ownership of the bus, the DMA Controller presents the upper address lines (mapped from the HTARGET register), and the Near Global SRAM bus of the target DSP(s) is acquired.
- The DMA Controller automatically generates addresses.

Data writes from the DMA Controller to Near Global SRAM of a target DSP will not affect the operation of other DSP's access to their own Near Global SRAM.

For example, if the DMA target is Node A, then Node B can access its own Near Global SRAM without any latency. If Node B wishes to access the Global Shared Bus, then it will be held off by having it's cycle extended while the DMA transfer is in progress. Node A will be held off from using its Global Bus (accessing its Near Global SRAM) while the DMA transfer is in progress. In general it is a good idea for the Node A C40 to wait for the EOB interrupt before accessing the Global Shared Bus, otherwise it will have its cycle extended and will be unable to service other interrupts or use the COMM ports.

Operation of the HP Local Bus DMA Controller is highly dependent on the block size of the READ data and the flag settings for the READ FIFO.

Block Size	HP Local Bus DMA Controller Operation
Smaller than Almost Full Flag	1. The Local Bus DMA controller performs no action until the End of Block data signal is received from the Local Bus. 2. The DMA controller recognizes that a complete block of data is in the FIFO and requests the Global Shared Bus. 3. The DMA transfer begins - data words are read from the READ FIFO and are written to Near Global SRAM of the target DSP. 4. Sequential addresses are written to SRAM until the EOB signal is read from the FIFO. 5. The DMA controller sends an interrupt to all DSP's and stalls until the interrupt is serviced.
Larger than Almost Full Flag	1. The FIFO fills to the almost full point before EOB is received from the source. 2. When the FIFO reaches the almost full flag, the DMA Controller acquires the Global Shared Bus and starts a DMA transfer to Near Global SRAM. 3. The DMA transfer continues until the FIFO reaches the almost empty flag, at which point the DMA Controller releases the Global Shared Bus, and waits for the almost full flag. 4. This sequence of acquiring the bus when the FIFO is almost full, then releasing the bus when the FIFO is almost empty continues until an EOB is received from the HP Local Bus. An EOB in the FIFO signals the DMA Controller to acquire the bus (if it doesn't already own it) and complete the DMA transfer.

Note: The C4x DSP's cannot read data directly from the HP Local Bus READ FIFO. Only the DMA Controller has access to the incoming data.

8.4. HP Local Bus WRITE Operation

Writes to the HP Local Bus from the VX8 Carrier Board must be performed by a C4x DSP writing to the WRITE FIFO of the HP Local Bus Interface. There is no DMA support for HP Local Bus data generation.

Hardware logic on the board counts the data written to the FIFO and presents EOB, FRAME and WRITEDONE with the appropriate data bits. The general setup and operation of the HP Local Bus WRITE functionality is as follows:

1. Following HP Local Bus RESET, a C40 sets up the READ and WRITE FIFO almost full and almost empty flags by writing to the FIFO.
2. A C40 writes to the HP Local Bus Register space to setup HBLOCK (the desired block size for outgoing data), and HEOB for the state of FRAME and WRITEDONE at the end of the block transfer.

3. Any C40 can output data by acquiring the Global Shared Bus and writing directly to the WRITE FIFO. Typically the WRITE operation can be handled in an interrupt service routine, where the DSP receives an interrupt when the WRITE FIFO reaches the almost empty point. The DSP then fills the FIFO and continues processing data until it receives the next interrupt. Of course, implementation details will vary depending on the speed of the Local Bus data sink and the Block Size used.
4. The hardware counter counts the data written into the FIFO and asserts EOB, FRAME and WRITEDONE when the counter reaches 0.
5. The counter automatically re-loads to the HBLOCK value so that the C40 can continue writing data without any further configuration.

8.5. HP Local Bus Interface Register Description

The following table shows the hardware registers that are configured to support the VX8 Carrier Board:

Table 17 HP Local Bus Interface Registers

Address	Access	Register Name	Function
9000 0000	R/W	HCONTROL	HP Bus Control Register. Global control of the interface. Includes Reset, modes, DMA.
9000 0001	R	HSTATUS	HP Bus Status Register. Indicates buffer being or to be filled, FIFO status, HP Local Bus status signals.
9000 0002	W	HRESTART	Provides RESTART pulse to BALLISTIC chip.
9000 0003	R/W	HREADEOB	Causes READDONE to be asserted indicating end of transfer to this module.
9000 0004	R/W	HWRITEEOB	Sets the values of FRAME and WRITEDONE to be written to the FIFO when the block counter reaches 0.
9000 0005	W	HBUF0ADDR	DMA buffer 0 start address register (24 bits - 10. Aligned on a 1k boundary).
9000 0006	W	HBUF1ADDR	DMA buffer 1 start address register (24 bits - 10. Aligned on a 1k boundary).
9000 0007	W	HBLOCK	DMA block size register for WRITEs (24 bits).
9000 0008	W	HWAIT	Sets the number of wait states for DMA transfers.
9000 0009	W	HTARGET	Target Register for the node to receive data.
9000 000A	R/W	TTLTRG	TTL open collector trigger lines to P2. *Part of the VXIbus interface*.
9000 000B	R/W	ECLTRG	Bussed ECL data line to P2. *Part of the VXIbus interface*.
9000 000C	R	HINTSTAT	Interrupt source status register.
9000 000C	W	HINTCLR	Interrupt source clear register.
9000 000D	R/W	HINTENABLE	Interrupt enable register.
9000 0010	W	FIFO WRITE	Writes to this address will clock the 32 data bits on the bus into the WRITE FIFO.
R = Read Only, W = Write Only, R/W = Read/Write			

8.5.1. HCONTROL

D15	D14	D13	D12	D11	D10	D9	D8
						DMA INCR	DMA ON

D7	D6	D5	D4	D3	D2	D1	D0
MODE3	MODE2	MODE1	MODE0	STRIP	CONT	Reserved	/RESET

DMA INCR Enables address increment on DMA transfer - allowing for RAM or PORT interfaces. Active High.

DMA ON Enables DMA transfers. Active High.

MODE[3..0] Assert MODE[3..0] bits directly to the BALLISTIC chip. Refer to *Chapter 8* for BALLISTIC Mode details.

STRIP Asserts STRIP line to the BALLISTIC IC. The Frame bit on an HP Local Bus data stream will be stripped, allowing the device to append blocks to a frame of data originating from other device(s). For the last block, FRAME should be asserted to mark the end of frame (see *Section 8.5.5* WRITEEOB and *Section 8.5.8* HBLOCK).

CONT Asserts CONT line to the BALLISTIC IC. CONT will override the PAUSED/RESTART handshake. The IC will immediately perform a transition from a PAUSED state and continuously operate in its current mode.

/RESET Resets all state machines, asserts reset line to both read and write FIFOs.

Note: STRIP is intended to be used as a control bit for the non-standard BALLISTIC modes. The user should be extremely familiar with the BALLISTIC IC before using this functionality.

CONT is intended to be asserted during normal operation to provide continuous mode changes without pausing. Modification of this value should only be used for the non-standard BALLISTIC modes. The user should be extremely familiar with the BALLISTIC IC before using this functionality.

8.5.2. HSTATUS

D15	D14	D13	D12	D11	D10	D9	D8
				/WE	/WAE	/WAF	/WF

D7	D6	D5	D4	D3	D2	D1	D0
/RE	/RAE	/RAF	/RF	PAUSE	DMA BUFFER	Reserved	/RESET

/WE Write FIFO Empty status flag. For writes to the FIFO from the Global Shared Bus.

/WAE Write FIFO Almost Empty status flag. For writes to the FIFO from the Global Shared Bus.

/WAF Write FIFO Almost Full status flag. For writes to the FIFO from the Global Shared Bus.

/WF Write FIFO Full status flag. For writes to the FIFO from the Global Shared Bus.

/RE Read FIFO Empty status flag. For DMA transfers to the Global Shared Bus from the FIFO.

/RAE Read FIFO Almost Empty status flag. For DMA transfers to the Global Shared Bus from the FIFO.

/RAF Read FIFO Almost Full status flag. For DMA transfers to the Global Shared Bus from the FIFO.

/RF Read FIFO Full status flag. For DMA transfers to the Global Shared Bus from the FIFO.

PAUSE Mirror of PAUSED output bit from the BALLISTIC IC.

DMA BUFFER Indicates which buffer is active (either being filled or to be filled next).

/RESET Mirrors the /RESET bit in the control register.

8.5.3. HRESTART

A write to this register provides a 100 ns pulse to the RESTART input bit of the BALLISTIC IC. Any data value will cause the pulse to be written.

Note: HRESTART is intended to be used as a control pulse for the non-standard BALLISTIC modes. The user should be extremely familiar with the BALLISTIC IC before using this functionality.

8.5.4. HREADEOB

D7	D6	D5	D4	D3	D2	D1	D0
							READ DONE

READ DONE Asserts the READDONE input bit to the BALLISTIC IC. Since PACK is low, the next EOB received will trigger the end of transfer. See section *9.4, BALLISTIC Read Interface*, for a description of PACK.

8.5.5. HWRITEEOB

D7	D6	D5	D4	D3	D2	D1	D0
						WRITE DONE	FRAME

FRAME Sets the value of the FRAME input bit that is written to the BALLISTIC IC for the last word in a block (indicated by the block counter = 0).

WRITE DONE Sets the value of the WRITEDONE input bit that is written to the BALLISTIC IC for the last word in a block (indicated by the block counter = 0).

8.5.6. HBUF0ADDR

D23 - D10	D9 - D0
START ADDR	

Start Addr Start address of the HP Bus DMA Buffer 0. This address must be aligned on a 1kword boundary (the lower 10 data bits are ignored).

8.5.7. HBUF1ADDR

D23 - D10	D9 - D0
START ADDR	

Start Addr Start address of the HP Bus DMA Buffer 1. This address must be aligned on a 1kword boundary (the lower 10 data bits are ignored).

8.5.8. HBLOCK

D23 - D0
Block Size

Block Size HP Bus Interface block size. For HP Local Bus WRITE transfers only; it indicates the number of transfers to the BALLISTIC IC before EOB is asserted. Also, FRAME and WRITEDONE are asserted based on this counter and the HWRITEEOB register.

Note: The block size for HP Local Bus DMA READs is determined by the Local Bus data source. The DMA Controller uses the End Of Block signal to switch the DMA Buffer locations in Near Global SRAM.

8.5.9. HWAIT

D7	D6	D5	D4	D3	D2	D1	D0
					DMA Transfer Wait States		

Wait States

Sets the number of wait states for DMA transfers to Near Global SRAM. The number of wait states can be set from three to ten 60 MHz cycles with 000 (binary) being 3 cycles and 111 (binary) being 10 cycles. This register field can be used to vary the write pulse width given to port-style data receivers.

- If the target node is a 60 MHz C4x based TIM-40 modules, a value of 001 (4 cycles) should be used for a sustained throughput of 60 Mbytes/sec.
- If the target node is an embedded node, a value of 000 should be used, for a maximum throughput of 80 Mbytes/second sustained while there is data in the READ FIFO.

Note: C4x based TIM-40 modules that do not have zero wait state SRAM on their near global bus **cannot** be used as DMA Controller targets for data transfers. Although the DMA cycle can be slowed down to accommodate “slow” targets, this feature is intended for hardware oriented ports only.

8.5.10. HTARGET

D7	D6	D5	D4	D3	D2	D1	D0
		HTARGET Value					

The HP Local Bus DMA Target register sets which node receives the DMA data transfers. It is also used to specify any of the DMA Broadcast modes. The bit values are the following:

Table 18 DMA Broadcast Mode Bit Values

HTARGET Value D[5:0]	C40 or TIM-40 Node
0x12	A
0x13	B
0x14	C
0x15	D
0x16	E
0x17	F
0x18	G
0x19	H
0x1A	A, B, C, D, E, F, G, H
0x1B	C, D, E, F, G, H
0x1C	C, D, G, H
0x1D	E, F, G, H
0x1E	C, D

8.5.11. TTLTRG

D7	D6	D5	D4	D3	D2	D1	D0
/TTLTRG7	/TTLTRG6	/TTLTRG5	/TTLTRG4	/TTLTRG3	/TTLTRG2	/TTLTRG1	/TTLTRG0

A Write to this register latches the lower 8 data bits of the Global Shared Bus. The data is provided to P2 through open collector buffers. A Read of this register returns the status of the /TTLTRG lines.

8.5.12. ECLTRG

D7	D6	D5	D4	D3	D2	D1	D0
						ECLTRG0	ECLTRG0 DATA

ECLTRG0 DATA A write to this register latches the data bit. The data is provided to P2 through Emitter Coupled Logic (ECL) drivers. The output from this latch is then re-synchronized to the ECLTRG1 clock signal, so if there is no clock present on ECLTRG1, then the value of ECLTRG0 will not be clocked out to the P2 connector. When read, this bit returns the value of the data in this register.

ECLTRG0 When read, this bit returns the actual value on the ECL line. This bit is read only.

8.5.13. HINTSTAT

D7	D6	D5	D4	D3	D2	D1	D0
	Write FIFO Almost Full	Write FIFO Almost Empty	EOB (DMA Reads)				

This register is read by the C40 to determine which of the 3 HP Local Bus interrupt sources caused the interrupt on the HP Local Bus interrupt line IIOF1. HINTCLR is then written to clear the source. These bits are active high.

8.5.14. HINTCLR

D7	D6	D5	D4	D3	D2	D1	D0
	Write FIFO Almost Full	Write FIFO Almost Empty	EOB (DMA Reads) Buffer Switch				

This register is written to by the C40 to clear the appropriate interrupt source bits for IIOF1. For example a write to this register with 0x1D as data will clear the EOB Buffer Switch interrupt source and trigger that the next block of data should be transferred to Near Global SRAM by the DMA Controller.

8.5.15. HINTENABLE

D7	D6	D5	D4	D3	D2	D1	D0
	Write FIFO Almost Full	Write FIFO Almost Empty	EOB (DMA Reads) Buffer Switch				

This register is written to by the C40 to enable the appropriate interrupt source. If none of the EOB control bits are enabled then the EOB bit will be set upon receipt of an EOB, but no interrupt will be generated. In this case, the next DMA block will be transferred regardless of the EOB bit status.

8.6. HP Local Bus Interface Performance

The transfer rate performance of the various hardware interfaces associated with the HP Local Bus Interface is shown in the following figure and table.

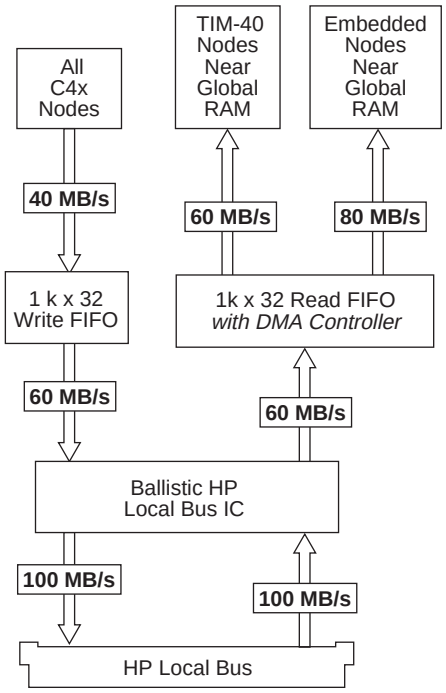


Figure 15 HP Local Bus Data Transfer Rates

Table 19 HP Local Bus Interface Performance

Source	Destination	Max. Sustained Transfer Rate
HP Local Bus	BALLISTIC Interface IC	100 Mbytes/sec
BALLISTIC Interface IC	Read FIFO	60 Mbytes/sec
Read FIFO	Embedded C40 Near Global SRAM (via DMA Controller)	80 Mbytes/sec
Read FIFO	TIM-40 based C40 Near Global SRAM (via DMA Controller)	60 Mbytes/sec
C4x DSP (Internal RAM or Local Bus)	Write FIFO	40 Mbytes/sec
Write FIFO	BALLISTIC Interface IC	60 Mbytes/sec
BALLISTIC Interface IC	HP Local Bus	100 Mbytes/sec

From this information the maximum sustained data transfer rates of the HP Local Bus interface can be derived as shown in the following table. Data rate values are given in Mbytes per second.

Source	Destination	Rate	Comment
HP Local Bus	Any C4x Near Global SRAM	60 MB/s	DMA Transfer
Any C4x Internal RAM or the C4x Local Bus	HP Local Bus	40 MB/s	Direct write transfer

Table 20 HP Local Bus Interface Maximum Sustained Transfer Rates

---

Note: The 80 Mbyte/s DMA transfer rate allowed between the Near Global RAM of an embedded C40 and the Read FIFO allows the Global Bus to be freed up for other data transfers.

The number of wait states for the DMA Controller writing to Near Global SRAM can be programmed into the HWAIT register. Therefore, slow ports can be accommodated; but at the expense of diminished DMA throughput.

- Zero wait state cycle requires three 60 MHz clock cycles for 80 MBytes/sec transfers from FIFO to embedded C40 Near Global SRAM.
- One wait state cycle requires four clock cycles for 60 MBytes/sec transfers from FIFO to TIM-40 based C40 (or broadcast) Near Global SRAM.
- Two wait state cycle requires five clock cycles for 48 MBytes/sec transfers.
- Three wait state cycle requires six clock cycles for 40 MBytes/sec transfers.
- Four wait state cycle requires seven clock cycles for 34 MBytes/sec transfers.
- Five wait state cycle requires eight clock cycles for 30 MBytes/sec transfers.
- Six wait state cycle requires nine clock cycles for 27 MBytes/sec transfers.
- Seven wait state cycle requires ten clock cycles for 24 MBytes/sec transfers.

Transfer rates slower than one wait state are provided for use with non-standard TIM-40 modules. The anticipated use for this model is to a “dedicated” port, FIFO, or memory device.

9 BALLISTIC HP Local Bus Implementation

9.1. BALLISTIC Chip Overview

BALLISTIC is the implementation of a high speed interconnection scheme used on VXI products. The IC was developed by Hewlett Packard explicitly for this purpose and its feature set is the following:

- Daisy Chained Local Bus transfers up to 100 MBytes/sec
- Emitter Coupled Logic (ECL) signaling levels for data and control strobes
- Asynchronous operation for flexibility between all HP Local Bus Master and Slaves
- 149 pin PGA package equipped with heat sink for robust operation within the VXI environment

The BALLISTIC HP Local Bus interface chip is capable of operating in any one of five different transfer ***modes***, and it is configurable to switch between these modes during different ***phases*** of a data flow cycle. For example, an HP Local Bus module can be configured to be in PIPE mode (passing data to the next module without modifying or examining it), then at the end of the current block of data, automatically switch to CONSUME mode (consume data by receiving the next block and not passing the data on). The next End Of Block (EOB) signal will cause the module to return to PIPE mode. The modes and state sequences are defined in this section.

9.2. Data Transfer Modes

There are five configurations that a module on the local bus can be in during any phase of a data flow cycle.

Pipe Data is being passed straight through from left to right (slot N-1 to slot N+1)

Generate Data is being generated in the current module and being transferred to the module on the right. (slot N to slot N+1)

Consume Data is being read in the current module from the module on the left. (slot N-1 to slot N)

Transform Data is being read in from the left, then processed in the current module. The processed data is then sent out the right side for another module to receive. (slot N-1 to slot N to slot N+1)

Eavesdrop Data is being read from a sending module on the left, and is also passed on to the module on the right. (slot N-1 to slot N and slot N-1 to slot N+1)

9.3. State Sequences for Mode Changes

For most data flow sequences, state changes can be handled automatically by the BALLISTIC IC. The five default states are defined, as well as a number of sequences that allow change of mode.

If you use function calls from the supplied libraries (for example, specifying the mode via **VX8_HPSetMode**), then the state sequences and transitions are transparent to your application.

Table 21 BALLISTIC State Sequences

MODE[3..0] Bits	MODE	State Sequence	Comments
0000	P	P|P|P|…	Pipe
0010	C	C|C|C|…	Consume
0011	E	E|E|E|…	Eavesdrop
0100	G	G|G|G|…	Generate
0101	T	T|T|T|…	Transform
0110	PC	PC|PC|PC|…	See *BALLISTIC Data Sheet*
0111	PE	PE|PE|PE|…	See *BALLISTIC Data Sheet*
1000	PG	PG|PG|PG|…	Append
1001	PT	PT|PT|PT|…	See *BALLISTIC Data Sheet*
1100	CP	CP|CP|CP|…	Strip
1101	EP	EP|EP|EP|…	See *BALLISTIC Data Sheet*
1110	GP	GP|GP|GP|…	Insert
1111	TP	TP|TP|TP|…	See *BALLISTIC Data Sheet*
1010	CPT	C|PT|PT|…	See *BALLISTIC Data Sheet*
1011	CPTP	CP|TP|TP|…	See *BALLISTIC Data Sheet*

The criteria for transitions of modes can be explained in terms of End Of Block (EOB) and Frame bits. The following state diagram shows the criteria that BALLISTIC follows for state transitions. The states (P, G, C, E, and T) have already been described. The STRIP state clears the frame bit from the last byte of data being piped through. The FRAME state sets the frame bit from the last byte of data in the GENERATE or TRANSFORM states. The state transition qualifiers are the following:

READDONE (RD) The module has read all the data it has been programmed to during one data flow cycle

WRITEDONE (WD) A module has written all the data it has been programmed to in one data flow cycle

End Of FRAME (EOF) An End Of Frame indicator has been received from the left local bus

The following state transition table describes which state transition qualifiers cause each state sequence. The simple state transitions (i.e. P|P|P|..) occur only on the End Of Frame transition qualifier and are omitted from further description. The following figure shows the twelve complex state transitions and highlights the state transition qualifiers for each.

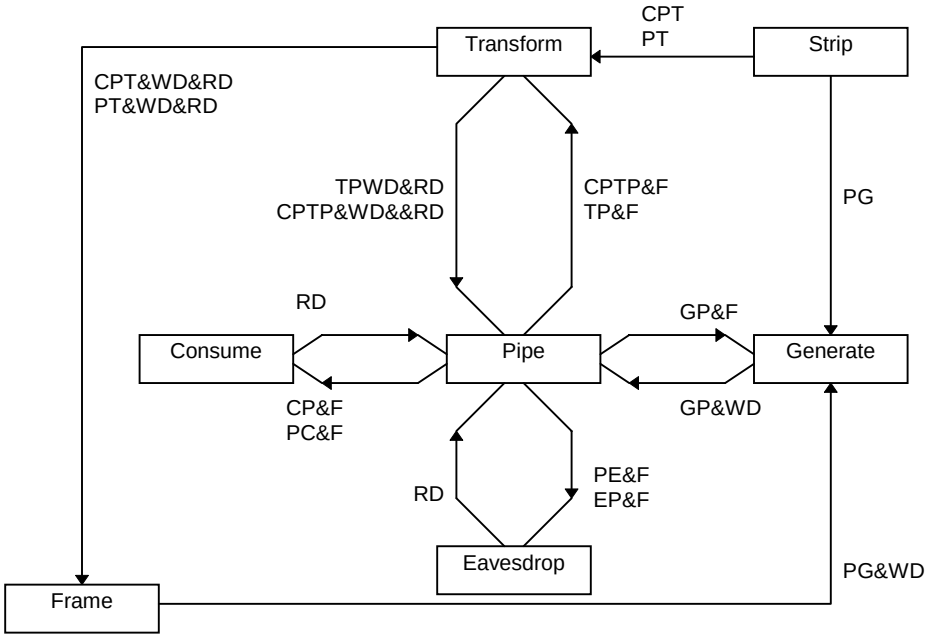


Figure 16 HP BALLISTIC IC State Transition Diagram

Table 22 BALLISTIC State Transitions

Transition #	Current State	Next State	MODE	Qualifier(s)
1	PIPE	TRANSFORM	CPTP TP	EOF EOF
2	PIPE	CONSUME	CP PC	EOF EOF
3	PIPE	GENERATE	GP	EOF
4	PIPE	EAVESDROP	PE EP	EOF EOF
5	PIPE	GENERATE via STRIP	PG	EOF
6	PIPE	TRANSFORM via STRIP	PT CPT	EOF EOF
7	CONSUME	PIPE	CP PC	RD RD
8	EAVESDROP	PIPE	PE EP	RD RD
9	TRANSFORM	PIPE	TP CPTP	RD&WD RD&WD
10	TRANSFORM	PIPE via FRAME	CPT PT	RD&WD RD&WD
11	GENERATE	PIPE	GP	WD
12	GENERATE	PIPE via FRAME	PG	WD

A complete description of the state transitions is given below, listed by Transition #:

1. In modes CPTP or TP, the transition from PIPE to TRANSFORM occurs on End Of Frame.
2. In modes CP or PC, the transition from PIPE to CONSUME occurs on End Of Frame.
3. In mode GP, the transition from PIPE to GENERATE occurs on End Of Frame.
4. In modes PE or EP, the transition from PIPE to EAVESDROP occurs on End Of Frame.
5. In mode PG, the transition from PIPE to GENERATE occurs on End Of Frame, however the FRAME bit is stripped off the end of the PIPE data transfer allowing data blocks to be appended during the GENERATE phase.
6. In mode PT or CPT, the transition from PIPE to TRANSFORM occurs on End Of Frame, however, the FRAME bit is stripped off the end of the PIPE data transfer allowing the received blocks of data to be received, transformed, and appended on the end of the PIPE frame of data.
7. In modes CP or PC, the transition from CONSUME to PIPE occurs when READDONE is asserted to BALLISTIC at End of Block.

8. In modes PE or EP, the transition from EAVESDROP to PIPE occurs when READDONE is asserted to BALLISTIC at End of Block.
9. In modes TP or CPTP, the transition from TRANSFORM to PIPE occurs when both READDONE and WRITEDONE are asserted at the appropriate End of Block.
10. In modes CPT or PT, the transition from TRANSFORM to PIPE occurs when both READDONE and WRITEDONE are asserted at the appropriate End of Block, however a FRAME bit is attached to the TRANSFORM data block to indicate that the Frame of data is now complete.
11. In mode GP, the transition from GENERATE to PIPE occurs when WRITEDONE is asserted to BALLISTIC at End of Block.
12. In mode PG, the transition from GENERATE to PIPE occurs when WRITEDONE is asserted to BALLISTIC at End of Block, however a FRAME bit is attached to the GENERATE data block to indicate that the Frame of data is now complete.

9.4. BALLISTIC Read Interface

9.4.1. WIDTHIO[1..0], PACK and PACKIN[1..0]

During a READ access of the BALLISTIC (the VX8 on board logic is receiving the data from BALLISTIC and will write it into the READ FIFO), WIDTHIO and PACK determine the data formats. For use on the VX8 Carrier Board, WIDTHIO is set to “11” indicating that the received data is always packed with bits 31:24 representing the first byte received. PACK is set to 0, and the combination of WIDTHIO and PACK indicates that only EOB and FRAME are valid.

Internal to the BALLISTIC, there are two registers - one that packs the received data, and one for allowing the module to READ the received data. The PACKIN control signals determine how many bytes are packed before transfer to the output register. On the VX8 board, PACKIN is configured to “11”, so 4 bytes are packed before transfer.

Note: EOB3 is received as a data bit and is presented with data to the VX8 control logic. For simplicity, EOB3 and FRAME3 are referred to as EOB and FRAME in this document.

9.4.2. READDONE

As seen in *section 9.3 State Sequences for Mode Changes*, READDONE is used to indicate the end of a READ for a data flow cycle. The READ cycle is terminated when READDONE is asserted (logic HIGH) and an End Of Block transfers from the packing register to the output register.

READDONE support is implemented as a single register bit on the VX8 Carrier Board. The expected mode of operation is that the user will assert the READDONE bit before setting the DMA Enable bit (if a single block of data should trigger the change of state),

or just prior to clearing the End Of Block interrupt that indicated receipt of the next to last data block before the change of state.

9.5. BALLISTIC WRITE Interface

9.5.1. WRITEDONE and EOB

As seen in *section 9.3 State Sequences for Mode Changes*, WRITEDONE is used to indicate the end of a WRITE for a data flow cycle. The WRITE cycle is terminated when WRITEDONE is asserted (logic HIGH) and a 32 bit word of data is written to BALLISTIC. On the VX8 Carrier Board, the WRITEDONE and EOB functionality is controlled by the block counter. The user sets the WRITEDONE bit at any point during a block WRITE (may be prior to the first piece of data being written, but must be prior to the last piece of data in a block being written). The on board control logic will count down to the last piece of data and assert the EOB and WRITEDONE bits with the last piece of data to ensure that these bits are written to BALLISTIC correctly.

9.6. Non-Standard BALLISTIC Modes

Use of the CONT, RESTART, and STRIP bits with the various pause modes of the BALLISTIC IC make it possible to configure extremely flexible non-standard BALLISTIC modes.

Note: The user should be extremely familiar with the BALLISTIC IC before attempting to use this functionality.

The following table indicates the various transfer modes and when the BALLISTIC will enter the PAUSED state for each mode. The BALLISTIC will only enter the PAUSED state when the CONT bit is not asserted, indicating that continuous operation is not required. The expected mode of operation is for the advanced user to configure the mode, complete a block or frame transfer resulting in BALLISTIC entering the PAUSED state. The user then re-configures the mode bits and writes to the RESTART register to restart the BALLISTIC.

Table 23 Non-Standard BALLISTIC Modes

MODE[3..0] Bits	MODE	Comments
0000	P	(P) pause on EOB
0001	P	(P) pause on EOF
0010	C	(C) pause on READDONE
0011	E	(E) pause on READDONE
0100	G	(G) pause on WRITEDONE
0101	T	(T) pause on READDONE and WRITEDONE
0110	PC	(P) pause on EOF (C) pause on READDONE
0111	PE	(P) pause on EOF (E) pause on READDONE
1000	PG	(P) pause on EOF and strip FRAME (G) pause on WRITEDONE
1001	PT	(P) pause on EOF and strip FRAME (T) pause on READDONE and WRITEDONE
1010	CPT	(C) pause on READDONE (P) pause on EOF and strip FRAME (T) pause on READDONE and WRITEDONE (repeat as PT)
1011	CPTP	(C) pause on READDONE (P) pause on EOF (T) pause on READDONE and WRITEDONE (P) pause on EOF (repeat as TP)
1100	CP	(C) pause on READDONE (P) pause on EOF
1101	EP	(E) pause on READDONE (P) pause on EOF
1110	GP	(G) pause on WRITEDONE (P) pause on EOF
1111	TP	(T) pause on READDONE and WRITEDONE (P) pause on EOF

10 VXIbus Interface

The VX8's register based VXIbus interface can function as either a Master or as a Slave on the VXIbus. VXIbus D32, D16, and D08EO data access is supported in the following address modes:

Address Mode	Master	Slave
A32	Yes	Yes
A24	Yes	No
A16	Yes	VXI registers only

The interface IC used for the VXIbus interface is the SCV64 from Tundra Semiconductor. It includes Master and Slave capabilities, as well as interrupt control. The SCV64 supports A32 / D32, D16, and D08EO accesses to the DRAM Shared Bus, and the on-board decode logic allows byte, word, and double word access to the shared DRAM memory from the host or from other VXIbus Masters. The Global Shared Bus only supports D32 accesses, so VXIbus slave accesses to the Near Global SRAM of the embedded C40's or the Near Global SRAM of the C4x DSP's on TIM-40 modules must be A32/D32.

The SCV64 does not support A16 VXIbus Slave accesses, so an external A16 register set is used to support the VXI registers.

Note: Unaligned transfer (UAT) slave accesses are **not** supported.

10.1. Status LEDs

Two LEDs on the front panel indicate status of the VX8 on the VXIbus. The red LED (SYSFAIL) is driven directly by hardware monitoring the VXIbus SYSFAIL line driven by the VX8. The green LED (ACTIVE) is driven directly by hardware and indicates accesses between the VXIbus and the VX8 for both slave and master modes.

10.2. VXIbus Master Memory Map

As seen from the VX8 C4x nodes, the physical memory for the VXI address space is mapped from address C000 0000h to FFFF FFFFh as shown in the following memory map.

Table 24 VXIbus Master Memory Map

C4x Address	VXIbus Address Space	VXIbus Address Range (bytes)
C000 0000h - FDFF FFFFh	VXI A32 Space	0000 0000h - F7FF FFFFh
FE00 0000h - FE3F FFFFh	Default A24:D16 Space	0000 0000h - 00FF FFFFh
FE40 0000h - FE7F FFFFh	Default A24:D32 Space	0000 0000h - 00FF FFFFh
FE80 0000h - FFFF BFFF	VXI A32 Space	FA00 0000h - FFFE FFFFh
FFFF C000h - FFFF FFFFh	Default A16:D16 Space	0000 0000h - 0000 FFFFh

Because the C4x DSP is 32 bit word addressed and the VXIbus is byte addressed, address translation is used for the C40's to master the VXIbus.

C40 Global Bus Address Lines A0 to A29 map to SCV64 address lines KADDR2 to KADDR31. SCV64 address lines KADDR[1..0] are set by the VCONTROL register on the DRAM Shared Bus. This control register also sets the KSIZE[1..0] inputs to the SCV64 that determine the number of bytes accessed. There is an overhead involved for D8, D16, or D24 transfers because the C40 must write the lower address bits, then perform the SCV64 access. D32 accesses will keep the lower address bits at [0,0], so sequential accesses are supported for higher throughput.

Note: If KSIZE[0..1] and KADDR[0..1] are non-zero, those SCV64 registers that require D32 cycles for access will be inaccessible.

The VXIbus Master interface supports A32/D32, D16, D08(EO), A24/D32, D16, D08(EO) and A16/D32, D16, D08(EO). The C40 is responsible for filling the proper byte lanes for writes, and must take endian conversion into account for non-D32 transfers. The VXIbus access functions provided in the VX8 C4x Support Software library ensure that this happens.

10.3. VXIbus Master Support Registers

The VX8 has the following hardware registers to support VXIbus mastering. The VX8 C4x Support Software library provides functions to access these registers.

Table 25 VXIbus Master Support Registers

Offset from VXI Register Base	Register Name	Function
0x0 R/W	VCONTROL	VXI Control Register. Used to set lower address lines, KSIZE and KFC for VXI master accesses.
0x0 READ and R/W	VSTATUS	VXI Status Register. Used to determine interrupt levels, interrupt vector status, and cycle status.

10.3.1. VCONTROL

D31..D8
Reserved

D7	D6	D5	D4	D3	D2	D1	D0
Reserved	KFC2	KFC1	KFC0	KSIZE1	KSIZE0	KADDR1	KADDR0

KFC 2..0 Sets the type of access performed as User, Supervisor Program, or Data accesses. This directly affects the address modifiers used for the VXI Master cycle.

KSIZE 1..0 Sets the number of bytes transferred. Directly affects D32, D16, or D8 address modifiers.

KADDR1..0 Sets the lower address bits (0, 1) of the SCV64 local bus. Always set to "00" for D32.

10.3.2. VSTATUS

D31..D8
Reserved

D7	D6	D5	D4	D3	D2	D1	D0
Reserved			VECTERR	ACCERR	KIPL2	KIPL1	KIPL0
			R	R/W[1]	R	R	R

VECTERR Status of the interrupt vector currently on the local data bus. Set if the vector is not valid. A non-vectored interrupt would set this bit, denoting a non-valid vector value on the bus.

ACCERR Status of the last bus cycle access made by the SCV64. Set on an error. Cleared by a write of a "1" to the ACCERR bit in the VSTATUS register.

KIPL2..0 The interrupt level of pending interrupts in the SCV64.

Note: Although VSTATUS is a read only register, a write to this address is used to clear the interrupt to the DSP's if it was caused by Bus Error (ACCERR asserted).

10.4. VXIbus Slave Memory Map (A32)

Table 26 VXIbus Slave Memory Map (A32)

VXI Byte Address BASE +	Access
0000 0000h	Test Bus Controller Registers
0000 0010h 007F FFFFh	Reserved
0080 0000h	Node A Global Memory Space
00C0 0000h	Node B Global Memory Space
0100 0000h	Node C Global Memory Space
0140 0000h	Node D Global Memory Space
0180 0000h	Node E Global Memory Space
01C0 0000h	Node F Global Memory Space
0200 0000h	Node G Global Memory Space
0240 0000h	Node H Global Memory Space
0280 0000h	Node A, B, C, D, E, F, G, H Global Memory Space
02C0 0000h	Node C, D, E, F, G, H Global Memory Space
0300 0000h	Node C, D, G, H Global Memory Space
0340 0000h	Node E, F, G, H Global Memory Space
0380 0000h	Node C, D Global Memory Space
02C0 0000h 03FF FFFFh	Reserved
0400 0000h 07FF FFFBh	Shared DRAM
07FF FFFCh 07FF FFFFh	Location Monitor

The VXIbus slave interface is inherently byte addressed and cannot map all the DRAM, registers, and C4x Near Global SRAM into the VXIbus space. As a result, the VXIbus Master may not have access to all of the global memory space of each C40. The VXI Slave interface is limited to the lower 1 MWords (4 MBytes) of each C4x DSP's global bus.

Note: The SCV64 internal registers are not mapped into the VXIbus Slave Memory Map. Node A C40 must be used to setup the A32 base address and enable A32 accesses according to the A16 VXIbus registers.

Note: The SCV64 is configured to always respond as a D32 local port.

10.4.1. Near Global SRAM Access

The VXIbus slave interface can read or write directly to the Near Global SRAM of the embedded C40's and the C4x DSP's on TIM-40 modules that are connected to the Global Shared Bus. Support is also provided for broadcast capability from any VXIbus Master to the Near Global SRAM of all DSP's installed on the VX8 Carrier Board. Arbitration for the DRAM Shared Bus, the Global Shared Bus, and the particular DSP's near global bus is performed automatically in hardware and is completely transparent to the user for slave accesses. Only A32/D32 accesses are allowed to Near Global SRAM.

Caution: Accesses from the VXIbus slave interface to Near Global SRAM are intrusive. As a result all other Global Shared Bus accesses during the transfer are held off. This data path is intended for code download and possibly transfer of control information from the host to the DSPs.

The VX8 C4x Support Software library provides functions for accessing Near Global SRAM.

10.4.2. Shared DRAM Access

The VXIbus slave interface can read or write directly to the shared DRAM on the Shared DRAM Bus. Arbitration for the DRAM Shared Bus is performed automatically in hardware and is completely transparent to the user. A32 / D32, D16, D08EO accesses are supported. The SCV64 DMA can also access the DRAM.

Note: Accesses from the VXI slave interface to shared DRAM are not intrusive to the Global Shared Bus, and as a result will not affect HP Local Bus Interface DMA transfers or C4x DSP to Global Shared Bus transfers. This data path is the recommended operating mode of real time data transfer between the VX8 Carrier Board and other boards on the VXIbus (including the host controller).

The VX8 C4x Support Software library provides functions for accessing Shared DRAM.

10.5. VXIbus Slave Memory Map (A16)

DIP switch S1 on the VX8 selects the module's A16 logical address. The VXIbus A16 interface consists of only the first four 16-bit registers in the A16 space. The Control, Status, and Offset registers are also accessible on the local bus of the Node A C40. The kernel program initializes these registers and the VX8 C4x Support Software library provides functions to access them. The registers, with their addresses and access privileges are shown in the following table.

Table 27 VXIbus A16 interface Registers

Register	VXIbus Address	VXIbus Access	Node A Address	Node Access
ID Register	BASE + 0000	Read Only	none	none
Device Type	BASE + 0002	Read Only	none	none
Control	BASE + 0004	Write Only	0x1000 0002	Read Only
Status		Read Only		Write Only
Offset	BASE + 0006	Read/Write	0x1000 0003	Read Only

10.5.1. ID Register

D15	D14	D13	D12	D11	D10	D9	D8
Device Class		Address Space		Manufacturer's ID			

D7	D6	D5	D4	D3	D2	D1	D0
Manufacturer's ID							

Device Class This card is a register based device and the code is "11".

Address Space This defines the type of interface and is set to "01" for A32/A16 memory.

Manufacturer ID This number uniquely identifies the board manufacturer. Spectrum's ID is 111010010000 binary (0xE90).

10.5.2. Device Type Register

D15	D14	D13	D12	D11	D10	D9	D8
Required Memory				Model Code			

D7	D6	D5	D4	D3	D2	D1	D0
Model Code							

Model Code This code is set by the manufacturer to define their different products. The code for the VX8 is 010101010101 (0x555).

Required Memory This code defines the size of the memory space occupied by the card in VXI space. The code is “0100” for 128 Mbytes.

10.5.3. Control Register

D15	D14	D13	D12	D11	D10	D9	D8
A32 Enable			Firmware Revision Query	Self Test Result Query			

D7	D6	D5	D4	D3	D2	D1	D0
						SYSFAIL Inhibit	Reset

A32 Enable Set by the host to tell the card to enable its A32 VXIbus interface (the SCV64). When set to 1, A32 access is enabled; when set to 0, A32 access is disabled.

Firmware Revision Query Set by the host to request the Node A C40 to load the firmware revision code into location 8000 0000h of its global SRAM. It is cleared by another write from the host. Active High.

Self Test Query Set by the host to request the Node A C40 to load the self test results into location 8000 0001h of its global SRAM. It is cleared by another write from the host. Active High.

SYSFAIL Inhibit Inhibits the card from asserting the /SYSFAIL line to the VXIbus when it is set to 1. This prevents a single failed device from bringing down the entire VXIbus. Active high.

Reset Set by the host to reset the card. The card stays reset until the reset bit is cleared by the host. The entire card is reset by this bit with the exception of the A16 register set. Active High.

An interrupt is sent to the Node A C40 whenever the host writes to any of the Control register bits. This advises it of a change in the A16 register parameters. The interrupt is cleared when the C40 issues a write access to the Offset register at address 0x1000 0003; data is not actually written to the Offset Register, but this action causes the on-board logic to clear the A16 interrupt.

10.5.4. Status Register

D15	D14	D13	D12	D11	D10	D9	D8
A32 Active	MODID	/CONFIG	Firmware Revision Query Stat.	Self Test Result Query Stat.			

D7	D6	D5	D4	D3	D2	D1	D0
				Ready	Passed		

A32 Active Set by the card when the SCV64 has been configured and enabled for A32 transactions. Active high.

/MODID Set by hardware to reflect the state of the card's slot MODID* line. When MODID* is low, the /MODID bit is low.

/CONFIG Set to a logic 1 by the card when all of the C40 nodes are configured (have completed their boot sequences) and ready for code download. Logic 0 indicates that at least one of the C40 nodes in the system is still in it's configuration sequence. Active low.

Revision Query Status This bit mirrors the state of the same bit in the control register. It is cleared when the control register bit is cleared by the host. Active High.

Self Test Query Status This bit mirrors the state of the same bit in the control register. It is cleared when the control register bit is cleared by the host. Active High.

Ready Set by the card when it is ready to interact with the VXIbus. Active high.

Passed Set by the card when the power up self tests are successfully completed. Active high.

Note: All Status register bits are cleared by a system reset (SYSRESET or VXIbus A16 Control register bit). When the reset is released they resume normal operation.

10.5.5. Offset Register

D15	D14	D13	D12	D11	D10	D9	D8
Offset	Offset	Offset	Offset	Offset			

D7	D6	D5	D4	D3	D2	D1	D0

Offset The top five bits of this register are set by the host to define the location of the card's A32 VXI port in the VXI map. Only the top five bits are needed due to the size of memory occupied by the card in VXI space (128 MBytes).

An interrupt is sent to the Node A C40 whenever the host writes to the Offset Register. This advises it of a change in the A16 register parameters. The interrupt is cleared when the C40 issues a write access to the Offset register address 0x1000 0003; data is not actually written to the Offset Register, but this action causes the on-board logic to clear the A16 interrupt.

10.6. VXIbus Interrupt Handler Support

The VX8 Carrier board can act as an interrupt handler for all interrupt levels on the VXIbus. Internal registers of the SCV64 can be configured to select the interrupt levels to which the VX8 responds. Refer to the *SCV64 data book* for information on the internal register set.

Software functions to enable, disable, identify, and acknowledge VXIbus interrupts are supplied with the VX8 C4x Support Software library. Refer to the *VX8 Carrier Board Programming Guide* for details.

VXIbus interrupts are transmitted to the IIOF0 of each node on the VX8 Carrier Board. The following sources can cause an IIOF0 interrupt:

- VXIbus external interrupt (KIPL lines)
- SCV64 DMA Controller (KIPL lines)
- SCV64 Timers (KIPL lines)
- SCV64 Location Monitor (KIPL lines)
- VXIbus ERROR detected by the on-board logic on a VXIbus access from a DSP (BUSERR or timeout)

Note: Although all processors have access to the VCONTROL, VSTATUS, and SCV64 registers, it is best to use node A to service VXIbus interrupts. Node A is the only node that can use /SUSPEND to service interrupts during HP Local Bus DMA transfers.

10.7. Determining the Interrupt Source (VXI IACK Cycle)

The VX8 C4x Support Software library uses the following procedure to identify and acknowledge VXIbus interrupts:

1. The DSP reads the VSTATUS register.
   - If the ACCERR bit is set, a VXIbus error caused the interrupt. The interrupt is then cleared by writing to the VSTATUS register location. The user's code must take appropriate action on ACCERR.
   - If the ACCERR is not set then the interrupt was caused by an interrupt on one or more of the KIPL levels.
2. If the interrupt was caused by an interrupt on one or more of the KIPL levels, then the DSP reads from the /IACK memory space on the Global Bus Memory Map. To do this, the DSP must first read the state of the KIPL lines from the VSTATUS register, and then set up the VCONTROL register and determine the address to read as shown below.

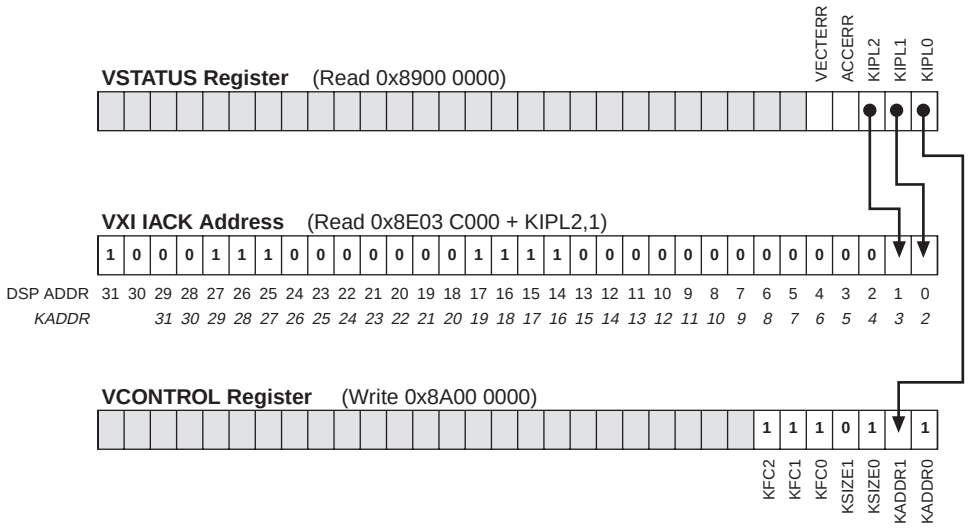


Figure 17 Setting the VXI IACK Address

The KIPL lines indicate the interrupt level. The interrupt level is mapped to the KADDR lines read by the SCV64. The Interrupt levels assigned to the KADDR lines of the SCV64 are given in the following table.

Table 28 KADDR Interrupt Levels

Interrupt Level	KADDR [3..1]
7	111
6	110
5	101
4	100
3	011
2	010
1	001

3. The SCV64 recognizes that an /IACK cycle is in progress and determines if it was an internal (SCV64 DMA Controller / SCV64 Timer) or external (VXIbus) interrupt.

4. If the source was an external VXIbus interrupt, an /IACK cycle is initiated on the VXIbus.

5. The /IACK cycle is acknowledged by the interrupter which puts its interrupt vector on the lower 8 data bits of the DSP's data bus.

6. The DSP completes the /IACK cycle, stores the received vector, and then checks the VSTATUS register again to determine if the vector is valid.

   - If VECTERR is **not** set then the DSP knows that the interrupt was caused by a VXIbus interrupt request.
   - If VECTERR is set, then the interrupt was not an external VXIbus interrupt request and could only be caused by an internal SCV64 interrupt source. The DSP should then access the SCV64 internal registers and clear the source of the interrupt.

10.8. VXIbus Interrupter Support

The VX8 Carrier Board can act as an interrupter on the VXIbus. This interrupt support is provided within the SCV64, and any C4x DSP with access to the Global Shared Bus can write to the SCV64 internal registers to setup the vector and cause assertion of the VXIbus interrupt. Refer to the *SCV64 data book* for information on the internal register set.

The VX8 C4x Support Software library supplies a function to generate a VXIbus interrupts. Refer to the *VX8 Carrier Board Programming Guide* for details.

10.9. VXI Local Bus Specific Signals

The VX8 implements these VXI local bus specific signals in the following manner.

CLK10± CLK10± is a 10 MHz ECL clock generated by the Slot 0 Controller in a VXIbus mainframe. These differential signals are converted to TTL levels on the VX8 Carrier Board and can be routed to TCLK1 of each TIM-40 module site (a jumper selects between CLK10 and ECLTRG1).

ECLTRG1 ECLTRG1 is defined as the clock line for backplane synchronous signaling. It's frequency and characteristics can be configured by the system designer, and the VX8 Carrier Board can always receive this signal, but can also generate this signal through the SMB connector. ECLTRG1 is converted to a TTL signal and routed to jumper JP1. The user can select between the TTL versions of CLK10 or ECLTRG1 to be routed to the TIM-40 modules on TCLK1. ECLTRG1 is also routed to the on board logic used to re-synchronize ECLTRG0 to the ECLTRG1 clock. The C4x DSP's on TIM-40 modules must not configure TCLK1 as an output if it is connected to the TIM-40 module connector.

ECLTRG0 ECLTRG0 is defined as the data line for backplane synchronous signaling. The VX8 Carrier Board can both receive and generate this signal. The board always receives this signal, and it is converted to TTL signal levels and routed to User-defined pin 12 on all TIM-40 modules. The board can generate this data in one of two ways: by a C40 writing to a register in the HP Local Bus register set, or through the SMB connector JP2 If the C40 is writing to the ECLTRG register, then this signal is synchronized with ECLTRG1, converted to ECL and sent to the P2 connector. The VX8 C4x Support Software provides functions for accessing the ECLTRG0 line.

TTLTRG*(7..0) TTLTRG(7..0) are TTL level, open collector signals that can be used to communicate between boards, or used to trigger events at low speeds. The VX8 Carrier Board can both receive and generate the TTLTRG lines. A write to the TTLTRG will latch the lower 8 data bits of the Global Shared Bus and immediately provide them to the P2 connector. A read of the TTLTRG will return the value seen on the TTLTRG lines on the lower 8 data bits of the Global Shared Bus. The remaining bits of the Global Shared Bus are not defined. The VX8 C4x Support Software provides functions for accessing the TTLTRG lines.

10.10. SCV64 Register Map

The register set for the SCV64 is accessible on the DRAM Shared Bus from any of the processor nodes. The base address for the register set is 8800 0000h. A complete listing of the register locations, as accessed from the processor node DSPs, is shown in the following table. Complete information on the SCV64 and its registers can be found in the *SCV64 data book*.

Table 29 SCV64 Register Map

Address	Description	Register
8800 0000	DMA Local Address	DMALAR
8800 0001	DMA VMEbus Address	DMAVAR
8800 0002	DMA Transfer Count	DMATC
8800 0003	Control and Status	DCSR
8800 0004	VMEbus Slave Base Address	VMEBAR
8800 0005	Rx FIFO Data	RXDATA
8800 0006	Rx FIFO Address Register	RXADDR
8800 0007	Rx FIFO Control Register	RXCTL
8800 0008	VMEbus/VSB Bus Select	BUSSEL
8800 0009	VMEbus Interrupter Vector	IVECT
8800 000A	Access Protect Boundary	APBR
8800 000B	Tx FIFO Data Output Latch	TXDATA
8800 000C	Tx FIFO Address Output Latch	TXADDR
8800 000D	Tx FIFO AM Code and Control Bit Latch	TXCTL
8800 000E	Location Monitor FIFO Read Port	LMFIFO
8800 000F	SCV64 Mode Control	MODE
8800 0010	Slave A64 Base Address	SA64BAR
8800 0011	Master A64 Base Address	MA64BAR
8800 0012	Local Address Generator	LAG
8800 0013	DMA VMEbus Transfer Count	DMAVTC
8800 0014 8800 001F	Reserved Reserved	
8800 0020	Status Register 0	STAT0
8800 0021	Status Register 1	STAT1
8800 0022	General Control Register	GENCTL
8800 0023	VMEbus Interrupter Requester	VINT
8800 0024	VMEbus Requester Register	VREQ
8800 0025	VMEbus Arbiter Register	VARB
8800 0026	ID Register	ID

Address	Description	Register
8800 0027	Control and Status Register	CTL2
8800 0028	Level 7 Interrupt Status Register	7IS
8800 0029	Local Interrupt Status Register	LIS
8800 002A	Level 7 Interrupt Enable Register	7IE
8800 002B	Local Interrupt Enable Register	LIE
8800 002C	VMEbus Interrupt Enable Register	VIE
8800 002D	Local Interrupts 1 and 0 Control Register	IC10
8800 002E	Local Interrupts 3 and 2 Control Register	IC32
8800002F	Local Interrupts 5 and 4 Control Register	IC54
88000030	Miscellaneous control register	MISC
88000031	Delay line control register	DLCT
88000032	Delay line status register 1	DLST1
88000033	Delay line status register 2	DLST2
88000034	Delay line status register 3	DLST3
88000035	Mailbox register 0	MBOX0
88000036	Mailbox register 1	MBOX1
88000037	Mailbox register 2	MBOX2
88000038	Mailbox register 3	MBOX3
88000039 8800007F	Reserved Reserved	

10.11. SCV64 Default Configuration

Upon initialization the SCV64 DMA transfer count size is 20 bits. Both read and write FIFOs are enabled for decoupled operation. Burst mode is disabled for the receive FIFO and DMA as the VX8 does not support local bursts.

The VXIbus is set as follows:

- Fair mode
- Release when done
- Bus request level 3
- Release on BCLR
- Bus ownership timer is set for 8 µs

IRQ1 is an interrupt pin only. This is needed in the event that the host/backplane floats the IRQ1 line causing the SCV64 to constantly go into bi-mode. Other interrupt levels are set as follows:

Interrupt	Level
/VMEINT	5
Location monitor	4
Timer	3

10.12. Deadlocks and Deadlock Resolution

Deadlocks occur when a C4x on the VX8 Carrier Board attempts to master the VXIbus while another card is attempting to access the Carrier Board's memory. Deadlock situations are resolved by the VX8 hardware as follows:

Deadlock Situation	Hardware Resolution
The DSP attempts a VXIbus master cycle while the host (or other VXIbus master) attempts a DRAM bus access.	1. The DSP is backed off (by extending its cycle) to the buffers between the DRAM Shared Bus and the Global Shared Bus. 2. When the VXIbus master finishes its transfer, the DRAM Shared Bus is granted to the DSP so that it can then retry it's VXIbus master cycle.
The DSP attempts a VXIbus master cycle while the host (or other VXIbus master) attempts to access the Near Global SRAM of another DSP.	1. The DSP is backed off to the buffers between the DSP's Global Bus and the Global Shared Bus. 2. When the VXIbus master finishes its transfer the Global Shared Bus is granted to the DSP. 3. The DRAM Shared Bus is granted to the DSP so that it can then retry its VXIbus master cycle.
The DSP attempts a VXIbus master cycle while the host (or other VXIbus master) attempts to access the Near Global SRAM of that DSP.	1. The DSP is backed off all the way to the DSP. 2. When the VXIbus master finishes its transfer, the Global Bus is granted to the DSP. 3. The DRAM Shared Bus is granted to the DSP so that it can then retry it's VXIbus master cycle.

11 Specifications

11.1. Cycle Types and Cycle Performance

This section describes the number of wait states required for accesses to the various memory devices, interface IC's, and registers. The 60 MHz TMS320C4x DSP runs off the H1 clock running at 30 MHz. A zero wait state read from external SRAM is one cycle, while a zero wait state write to external SRAM is two cycles. Refer to the *TMS320C4x User's Guide* for detailed timing information.

Note: These cycle times are for sustained data transfers after the Source has acquired the relevant bus. The time required to acquire the bus is dependent on whether or not some device is using that bus. These cycle times also assume no interruption (i.e. VXIbus slave access pre-empting C40 -> DRAM).

Table 30 VX8 Timing Cycles (30 MHz Cycles)

Access Type	From a C40 Using Local or Global Bus	From a C40 Over Global Shared Bus	From VXIbus (Slave Access)
Local Bus PEROM	5 Read / 7 Write	N/A	N/A
Local Bus DUART	4 R/W	N/A	N/A
VXIbus A16 Registers	2 Read / 3 Write	N/A	N/A
IIOF2 Interrupt Control Register	3 Write	N/A	N/A
/SUSPEND Register	3 Write	N/A	N/A
Local SRAM	1 Read / 2 Write (120 / 60 Mbytes/sec)	N/A	N/A
Global SRAM	1 Read / 2 Write - own Near Global SRAM (120 / 60 Mbytes/sec)	3 R/W - other DSP's Near Global SRAM (40 Mbytes/sec)	5 R/W - any DSP's Near Global SRAM (24 Mbytes/sec)
Internal SRAM	1 Read / 1 Write (120 / 120 Mbytes/sec)	N/A	N/A
Global DRAM	N/A	5 Read / 6 Write (24 / 20 Mbytes/sec)	6 R/W (17 Mbytes/sec)
HP Local Bus Registers	N/A	1	N/A

11.2. C40 COMM Port Performance

Path	Data Transfer Rate
Onboard	24 Mbytes per second
Via Front Panel	15 Mbytes per second

11.3. Physical Specifications

Since the VX8 Carrier Board provides a modular and highly configurable processing environment, the exact power consumption and electrical specifications will be different in each configuration.

Caution: The user of a particular configuration of the VX8 Carrier Board must ensure that the total current consumption and cooling requirements of their VXI system do not exceed the capabilities of the mainframe.

Specification	Base (No TIM Modules)	6 MDC40SS TIM Modules
Current Consumption		
+5.0 V + 5, - 2.5%	3.44 A typical	7 A typical
-5.2 V ± 5%	700 mA typical	731 mA typical
-2.0 V ± 5%	165 mA typical	176 mA typical
+12.0 V	0 mA	0 mA
-12.0 V	0 mA	0 mA
+24.0 V	0 mA	0 mA
-24.0 V	0 mA	0 mA
Cooling Airflow	1.78 litres/second @ 0.07 mm H_2O	3.28 litres/second @ 0.27 mm H_2O

12 External Interface Connectors

The external interfaces to the VXI-C40 board consist of P1 and P2 VME / VXI connectors, COMM Port connectors, Application Specific connectors, JTAG connectors, LED's, and serial port connectors.

12.1. P1 VXI Connector

The P1 VXI Connector uses a standard 96 pin DIN, 3 row connector with upper and lower shields.

Table 31 P1 VXI Connector Pinout

Pin #	(A Row) Signal	(B Row) Signal	(C Row) Signal
1	D00	BBSY*	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	GND	BG2OUT*	GND
10	SYSCLK	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	GMD	A18
21	IACKIN*	SERCLK	A17
22	IACKOUT*	SERDAT*	A16
23	AM4	GND	A15
24	A07	IRQ7*	A14

Pin #	(A Row) Signal	(B Row) Signal	(C Row) Signal
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	-12V	+5VSTANDBY	+12V
32	+5V	+5V	+5V

12.2. P2 VXI Connector

The P2 VXI Connector uses a standard 96 pin DIN 3 row connector with two external rows for shielding. The P2 pin connections follow the slot 1-12 (not 0) pin definitions. The pinout is as follows:

Table 32 P2 VXI Connector Pinout

Pin Number	(Row A) Signal	(Row B) Signal	(Row C) Signal
1	ECLTRG0	+5V	CLK10+
2	-2V	GND	CLK10-
3	ECLTRG1	RESERVED	GND
4	GND	A24	-5.2V
5	LBUSA0 (LDATA0IN)	A25	LBUSC0 (LDATA0OUT)
6	LBUSA1 (LDATA1IN)	A26	LBUSC1 (LDATA1OUT)
7	-5.2V	A27	GND
8	LBUSA2 (LDATA2IN)	A28	LBUSC2 (LDATA2OUT)
9	LBUSA3 (LDATA3IN)	A29	LBUSC3 (LDATA3OUT)
10	GND	A30	GND
11	LBUSA4 (LDATA4IN)	A31	LBUSC4 (LDATA4OUT)
12	LBUSA5 (LDATA5IN)	GND	LBUSC5 (LDATA5OUT)
13	-5.2V	+5V	-2V
14	LBUSA6 (LDATA6IN)	D16	LBUSC6 (LDATA6OUT)
15	LBUSA7 (LDATA7IN)	D17	LBUSC7 (LDATA7OUT)
16	GND	D18	GND
17	LBUSA8 (LDAVIN)	D19	LBUSC8 (LDAVOUT)
18	LBUSA9 (LREQOUT)	D20	LBUSC9 (LREQIN)
19	-5.2V	D21	-5.2V
20	LBUSA10 (LFRAMEIN)	D22	LBUSC10 (LFRAMEOUT)
21	LBUSA11 (LEOBIN)	D23	LBUSC11 (LEOBOUT)
22	GND	GND	GND
23	TTLTRG0*	D24	TTLTRG1*
24	TTLTRG2*	D25	TTLTRG3*
25	+5V	D26	GND
26	TTLTRG4*	D27	TTLTRG5*
27	TTLTRG6*	D28	TTLTRG7*
28	GND	D29	GND
29	RSV2	D30	RSV3
30	MODID*	D31	GND

Pin Number	(Row A) Signal	(Row B) Signal	(Row C) Signal
31	GND	GND	+24V
32	SUMBUS - NC	+5V	-24V

12.3. ECL Connectors

The ECL connectors J20 and J21 uses SMB RF Vertical Coaxial Connections.

J20 routs an external ECL level signal into the VXI ECL Trig 1 line (backplane). This line drives the sync_Clk line to a PLD that uses the signal to synchronize outbound ECL data. It can also be used to drive the TIM site TRGCLK lines via link JP1.

J21 routs an external ECL level signal into the VXI ECL Trig0 line (backplane). This line is also driven from the PLD synch ECL data line mentioned above. This output can also be routed to the TIM site TRGDAT lines via link JP1.

12.4. C40 COMM Port Connections

There are 8 front panel COMM Ports arranged in 4 input/output pairs. These pairs use AMP 786199-1 keyed 26/26-pin SCSI connectors. The bottom connector of the pair is a COMM Port that powers up in the input or receiver state. This connector is keyed. The top connector of the pair is a COMM Port that powers up in the output or transmit state. This connector is not keyed.

Table 33 C40 In (Bottom) COMM Port Connector Pinout

Pin Number	Top Row	Pin Number	Bottom Row
1	CRDY*	14	GND
2	CSTRB*	15	GND
3	CACK*	16	GND
4	CREQ*	17	GND
5	CD7	18	GND
6	CD6	19	GND
7	CD5	20	GND
8	CD4	21	GND
9	CD3	22	GND
10	CD2	23	GND
11	CD1	24	GND
12	CD0	25	DIRIN
13	GND	26	DIROUT

Table 34 C40 Out (Top) COMM Port Connector Pinout

Pin Number	Top Row	Pin Number	Bottom Row
1	CRDY*	14	GND
2	CSTRB*	15	GND
3	CACK*	16	GND
4	CREQ*	17	GND
5	CD7	18	GND
6	CD6	19	GND
7	CD5	20	GND
8	CD4	21	GND
9	CD3	22	GND
10	CD2	23	GND
11	CD1	24	GND
12	CD0	25	DIROUT
13	GND	26	DIRIN

The DIRIN and DIROUT signals of the connector are used for protection. A COMM Port drives DIROUT and senses DIRIN. The keyed connectors only allow the default (power up) input port to be connected with the default output. The input port DIROUT is connected to the output port DIRIN, and vice versa.

A transmitting output port (owns the token) drives DIROUT high. An input port (does not own the token) drives DIROUT low. Under normal operation when a transmitter drives DIROUT high and senses DIRIN as low its output buffer is enabled. Also, when a receiver drives DIROUT low and senses DIRIN high its input buffer is enabled. If DIRIN equals DIROUT on a port, its data buffer is disabled.

In COMM Port (Bottom)	Out COMM Port (Top)
1. Powers up as a receiver (no token).	1. Powers up as transmitter (has token).
2. DIRIN pulled low via 10K-ohm resistor	2. DIRIN pulled high via 10K-ohm resistor
3. If left unconnected both DIROUT and DIRIN are low, disabling the buffers.	3. If left unconnected both DIROUT and DIRIN are high, disabling the buffers.

When a device is receiving, it does not have the token. To be a transmitter, it requests the token from the transmitter. The pin numbers on transmit and receive ports are then swapped (all other lines -handshake and data- are on the same pins for transmit and receive). The direction signals in the VX8's COMM Port interface are only used to drive the buffers. The direction of the port itself is determined by which port has the token. Refer to the *TMS320C4x Users Guide* from Texas Instruments for diagrams showing the token transfer process. Timing diagrams for the DIRIN/DIROUT lines are shown in the following figure.

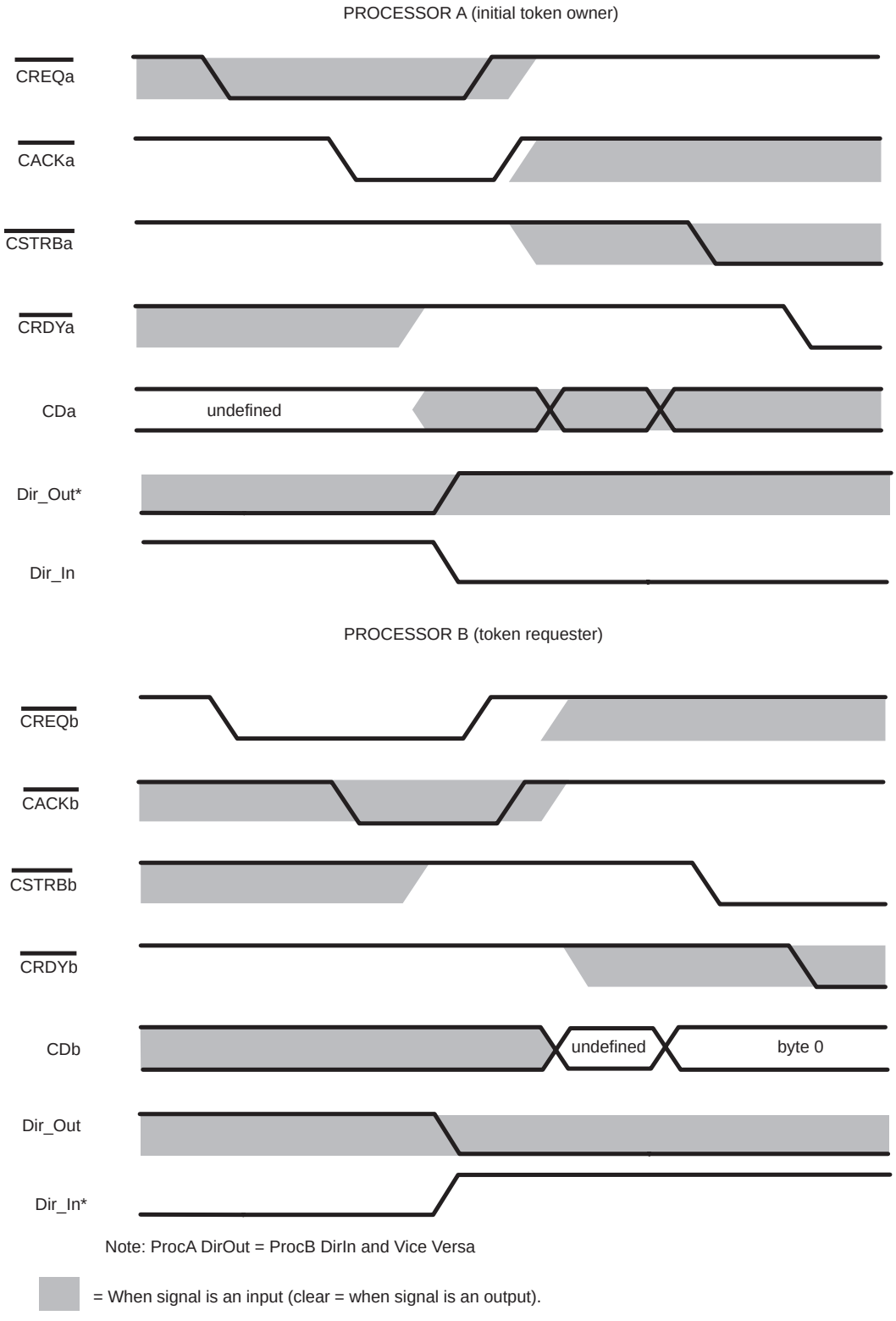


Figure 18 Token Transfer Timing Diagram

Under normal operation, DIROUT is never the same as DIRIN. If DIRIN and DIROUT are both high, the buffers are disabled; the port is dead until each interface is reset to power-up conditions (one port receiving and one transmitting). This would only happen if one or both of the ports gets confused about who has the token and both think they are transmitting. Note that all signals are buffered and the port is bi-directional.

The comm ports on this board's front panel are fitted with buffers to protect the processors from ESD damage and from noise on unterminated inputs.

12.5. JTAG Connectors

JTAG connection is provided by a single stacked, shielded connector. The upper connector is used for JTAG out, and the lower connector is used for JTAG in. A stacked 15 / 15 pin Micro D connector is used for JTAG In and Out.

Table 35 JTAG IN Connector Pinout

Pin Number	Top Row	Pin Number	Bottom Row
1	TMS	9	CONFIG
2	/TRST	10	/GRESET
3	TDI	11	GND
4	Presence Detect (+5V)	12	EMU1
5	TDO	13	GND
6	TCK_RET	14	GND
7	TCK	15	GND
8	EMU0		

Table 36 JTAG OUT Connector Pinout

Pin Number	Top Row	Pin Number	Bottom Row
1	TMS	9	CONFIG
2	/TRST	10	/GRESET
3	TDO	11	GND
4	SENSE	12	EMU1
5	TDI	13	GND
6	TCK_RET	14	GND
7	TCK	15	GND
8	EMU0		

12.6. RS-232 UART Connector

The DUART with RS-232 drivers is interfaced to one of the embedded C40's, and is brought to the front panel with a stacked Micro D 9 pin Receptacle (Female) Connector. Pinouts for the two connectors are identical, with Channel 1 on the top, and Channel 2 on the bottom.

Table 37 RS-232 Connector Pinout

Pin Number	Input (I) / Output (O)	Name
1	N/A	NC
2	I	RX - Received data
3	O	TX - Transmitted data
4	O	NC
5	I	Signal Ground
6	N/A	NC
7	N/A	RTS - Request to Send
8	N/A	CTS - Clear to Send
9	O	NC

12.7. Application Specific Connector

The Application Specific Connector uses the same connector as the C40 COMM Ports. It is used to route the user-defined pins from the TIM-40 connectors to the front panel.

Table 38 Top Application Specific Connector Pinout

Pin Number	Top Row User-Defined Pins NODE:PIN	Pin Number	Bottom Row User-Defined Pins NODE:PIN
1	C:1	14	C:5
2	C:2	15	C:6
3	C:3	16	C:7
4	C:4	17	C:8
5	D:1	18	D:5
6	D:2	19	D:6
7	D:3	20	D:7
8	D:4	21	D:8
9	E:1	22	E:5
10	E:2	23	E:6
11	E:3	24	E:7
12	E:4	25	E:8
13	GND	26	GND

Table 39 Bottom Application Specific Connector Pinout

Pin Number	Top Row User-Defined Pins NODE:PIN	Pin Number	Bottom Row User-Defined Pins NODE:PIN
1	F:1	14	F:5
2	F:2	15	F:6
3	F:3	16	F:7
4	F:4	17	F:8
5	G:1	18	G:5
6	G:2	19	G:6
7	G:3	20	G:7
8	G:4	21	G:8
9	H:1	22	H:5
10	H:2	23	H:6
11	H:3	24	H:7
12	H:4	25	H:8
13	GND	26	GND