SHELDON INSTRUMENTS

SI-C33DSP USER'S GUIDE
February 2012

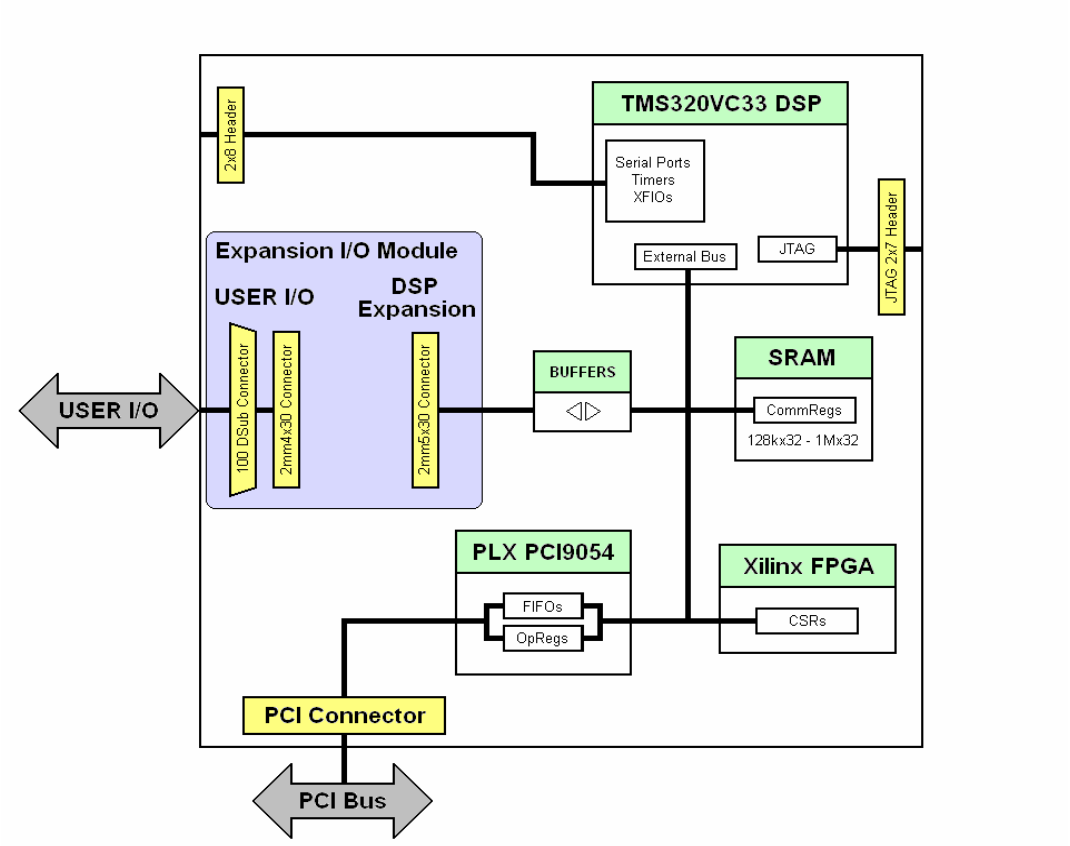
Table of Contents

1.0 System Overview

The SI-C33DSP-PCI from Sheldon Instruments is a powerful Digital Signal Processor (DSP) card for your PC equipped with a PCI bus. It is based on Texas Instruments' 150Mhz TMS320VC33, 32 bit floating point DSP, and can transform your PC into an ultra high performance development system and DSP accelerator. A full line of software development tools are available from Sheldon Instruments and TI, which include compilers, assemblers, linkers, as well as a real-time source debugger.

Key features include:

- 150MFLOP peak performance, 32 bit floating point precision.
- 128Kx32 words of one (1) wait state Dual Access SRAM Memory, expandable to 1Mx32 words.
- Full bi-directional PCI initiated bus mastering, with 132MB/sec peak transfer rate.
- Memory mapped host communications port.
- Software development tools from Sheldon Instruments includes QuVIEW, QuBASE and the SI-DDKs; as well compatibility with TI and third party tools.
- Windows and Linux 32/64 bit drivers and sample application support.
- Expansion connectors for prototyping, analog & digital I/O daughtercards.
- JTAG port for in system development and debugging.



2.0 Hardware Support

The SI-C33DSP includes expansion connectors allowing for custom designs, or for attaching 'off the shelf' multifunction I/O modules from Sheldon Instruments. Sheldon Instruments offers several daughter modules for multichannel analog and digital I/O, including 4 to 64 channels of 16 bit ADCs and DACs.

3.0 Software Support

The SI-C33DSP is available with extensive development tools from Sheldon Instruments and TI.

For quick turnkey development, Sheldon Instruments offers QuVIEW and QuBASE, which are a set of DSP-resident libraries for real time performance that greatly accelerate data acquisition, signal processing, and control applications. QuVIEW is a real time accelerator for LabVIEW, and QuBASE a real time accelerator for Visual Basic. A full range of examples and tutors are provided to demonstrate their ease of use and breadth of functionality and capabilities. QuBASE runs under Windows , while QuVIEW also runs under Linux.

Typical benchmarks for a 150Mhz C33 processor include the computation of a 1024 point Radix-2 complex FFT at 400us.

When purchased as a DSP evaluation board, Sheldon Instruments also includes free sample DSP and Windows or Linux device driver source code to accompany TI's development environment. The DSP source code illustrates full communication modes, and the Windows or Linux device driver source code includes the complete SI-DDK, along with COFF file loader utilities.

4.0 Hardware Description

The SI-C33DSP consists of a TMS320VC33 DSP, Asynchronous SRAM, Xilinx FPGA for glue logic, and host communications via the PCI bus using PLX's PCI9054.

4.1 System Resources

The PCI Base Addresses (BADDR) are automatically assigned by the host at boot time, with the requested resources as outlined in the following table:

PCI BADDRn	**Range** (Bytes)	**Resource**		**Host Offset** (Byte Boundary)	**DSP CE**	**DSP Base Address** (DWord Boundary)
0	256	PLX OpRegs	Memory Mapped	0x00-0xFF	3	0xFF,0000-0xFF,004F
1			I/O Mapped (n/a)			
2	8M	SRAM Bank0 (Comm Regs: 0xC0-0xCB)		0x00,0000-0x1F,FFFF	0	0x00,0000-0x3F,FFFF (Comm Regs: 0x30-0x3B)
		SRAM Bank1		0x20,0000-0x3F,FFFF	1	0x40,0000-0x7F,FFFF
		Expansion Module		0x78,0000-0x7B,FFFF	3	0xFE,0000-0xFE,FFFF
		FPGA: CSRs x2		0x7C,0000-0x7C,0007	n/a	n/a
3	128K	n/a		n/a	n/a	n/a
n/a	64K-2M	Add-on Init Buffer (DSP Side Only)		n/a	3	0xFE,0000-0xFE,FFFF

4.2 System Clocking

The clock for the system is generated from the DSP's crystal. Using the DSP's internal PLL, the base clock of 15Mhz is multiplied five (5) times to yield a resultant clock of 75Mhz. This resultant clock is the foundation for the overall card clocking and timing, and is sourced from the DSP's H1 output pin. Please refer to the C33 data sheet for more detailed information.

The external crystal is used as the clock source to the PLL, which in turn is used to generate the various required base frequencies used throughout the board. In order to minimize loading on those frequencies that are used by multiple devices, a clock distribution device is used as a buffer and stabilizer.

Two different values are utilized: 1) 37.5Mhz fed into the PLX bridge device, the Xilinx FPGA, as well as the expansion card interface, 2) 75Mhz fed into the Xilinx FPGA.

4.3 PCI Interface Hardware

The PLX's PCI9054 IC is set to operate in "C" mode, with the address and data bus demultiplexed. This allows the PCI9054 data and address bus be directly tied to the DSP address and data bus, while allowing the glue logic to solely arbitrate the control signals.

4.4 Memory Interface

The DSP runs with a single wait state on its expansion bus. This wait state is necessary due to the C33's limited memory interface capabilities, which only allow for standard asynchronous/synchronous SRAM interface. Designing a hardware based solution with this limitation requires the use of sub-nanosecond logic which is practically nonexistent. Despite this problem, the SI-C33DSP's design may be capable of running with no wait states when sub-nanosecond logic and faster memory are ever made available. On the other hand, the C33's on-chip, 0 wait state 32Kx32 memory makes up for this limitation as many DSP programs can be made to run in this memory block.

In order maximize performance and avoid delays caused by glue logic decoding, the SRAM memory devices are directly tied to the C33's PAGE pins. Up to two memory banks may be inserted occupying the space assigned to the C33's PAGE0 and PAGE1. The size of each bank of SRAM is 128kx32 or 512kx32. Therefore, the maximum memory can be 1Megx32, composed of a 512kx32 bank at PAGE0 and another bank of 512kx32 at PAGE1. The C33 requires sub 6nsec asynchronous SRAM memory to operate with 0 wait states. To account for propagation delays throughout the board, a 4nsec memory is needed to safely run the DSP with 0 wait states. Unfortunately, the fastest commercially available asynchronous memory is 8nsec. The DSP must run with 1 wait state for stable operation. It is possible to run with 0 wait state if H1 clock is slowed to 50 MHz using 8nsec asynchronous SRAM.

4.5 DSP and PCI9054 External Interface

The FPGA is used for the control logic to perform all communication arbitration between the DSP and PLX's PCI9054. Since the DSP uses a memory mapped interface while the PCI9054 uses a burst mode synchronous interface, all control signals are translated between the DSP and PCI9054.

Because the PCI9054, the C33, and the memory's address and data busses are tied together, the PCI9054 can directly access all devices and memory through the FPGA's control logic, irrespective if the DSP is in a reset state or not. The FPGA asserts the DSP's HOLD request line, which places the DSP signals into a high impedance state. The PCI9054 is then able to access all devices on the card, which include the external DSP memory, as well as the hardware expansion connector designed to hold a custom daughter card.

***NOTE**: The C33's internal memory and registers cannot be accessed directly by the PCI9054. Please refer to the C33 and PCI9054 data sheets and communication modes section below for details on how to access the internal memory and registers.*

5.0 System Initialization

When the system is first powered, the FPGA is automatically configured from its SPROM. During configuration, all FPGA I/O pins are in a high impedance state. In order to avoid spurious accesses, many signals need to be tied high or low for deterministic behavior, as listed below. This configuration guarantees that the DSP is in a reset/inactive state with its PLL functioning (PLL's output is the source of the overall system clock), while the PCI9054, memory and expansion space (daughtercard) are left unaccessed.

C33 Signals

- Pull ups
  - PAGE0, PAGE1, PAGE2, PAGE3
  - R/W, HOLD,
  - INT0, INT1, INT2, INT3
  - RSV0, RSV1
  - SHZ
  - EMU0, EMU1
- Pull downs
  - RESET

PLX Signals

- Pull ups
  - ADS, BLAST, READY, WAIT, LWR
  - BIGEND, LSERR, BTERM, DE,
  - LA2 through LA31
- Pull downs
  - EEDIO, EESK, EECS, EEPRE
  - TEST
  - LBE0, LBE1, LBE2, LBE3
  - BREQi
  - LHOLD, LHOLDA

After the FPGA is configured, its control signals are activated depending upon the source of the access, either the DSP or the PCI9054. When the PCI9054 is not accessing the DSP and/or the DSP is not running (in reset), the control signals to the PCI9054 are active.

6.0 Communication Modes

The SI-C33DSP-PCI card supports multiple, bi-directional communication modes between its DSP/PCI9054 and the host computer. From the perspective of the host computer, the SI-C33DSP-PCI is either a slave or the bus master. When the SI-C33DSP-PCI card is the bus master, either the PCI9054 transfers data using its own DMA engine, or the DSP performs the data transfer using programmed IO or its own DMA engine.

NOTE:
Both passive and active communication modes are supported. Passive communications do NOT require DSP intervention, while active communications require the DSP to be actively running code in order to interpret the command phase and subsequently allow for the data phase to take place.

Passive communications are essentially performed on the hardware level, with all handshaking taking place with arbitration logic. Active communications also make use of software handshaking.

***NOTE**: Please refer to the SICommModes.rtf document for more details.*

All active communication transfers between the host and the DSP consist of two phases:
1) The Command Phase. Details of the specific type of transfer to take place are loaded to the Communication Registers (CommRegs), which reside on the DSP's external SRAM.
2) The Data Transfer Phase. The data in question is actually transferred, using the CommRegs as a temporary depository.

To clarify the nomenclature, all read and write accesses are referred from the host/PCI bus' point of view. Data writes occur when the data is transferred from the host to the DSP external memory space, while data reads occur when the data is transferred from the DSP memory space to the host.

The effective passive communication modes of the SI-C33DSP-PCI are summarized below:

1) [Target][N/A][BB].
- Accesses are performed by the host using programmed I/O, while the DSP's status is irrelevant.
- The CommRegs are NOT used.

2) [BM][N/A][BB].
- Accesses to the host are performed using the PCI9054's DMA engine as the bus master, while the DSP's status is irrelevant.
- The CommRegs are NOT used.

The effective active communication modes of the SI-C33DSP-PCI are summarized below:

1) [Target][IO][Sync-Flg, BP-DReg].

- Both the host and the DSP perform accesses using programmed I/O.
- The CommRegs are used for both the command and data transfer phases. This mode is also referred to as "hostpolling" because the CommRegs serve as a temporary depository for each data point transferred
- Synchronization between the host and the DSP is performed with both polling the Flag register.

2) [AI-BM][IO][Async, BB, Int.]

- Accesses to the host are performed using the DSP as the bus master, with the DSP performing accesses using programmed I/O.
- The CommRegs are only used for the command phase, while the data is directly transferred to host memory mapped in the DSP's memory space.
- Synchronization between the DSP and the host memory is implemented with glue logic which effectively performs hardware wait states when one or the other is not accessible. The DSP's READY line is deasserted thus holding the bus, and no external DSP interrupts are used.
- The DSP alerts the host that a transaction is complete by sending it an interrupt through one of the PLX's doorbell registers.

3) [AI-BM][DMA][Async, BB, Int.]

- Accesses to the host are performed using the DSP as the bus master, with the DSP performing accesses using its DMA engine in Asynchronous mode.
- The CommRegs are only used for the command phase, while the data is directly transferred to host memory mapped in the DSP's memory space.
- Synchronization between the DSP and the host memory is implemented with glue logic which effectively performs hardware wait states when one or the other is not accessible. The DSP's READY line is deasserted thus holding the bus, and no external DSP interrupts are used.
- The DSP alerts the host that a transaction is complete by sending it an interrupt through one of the PLX's doorbell registers.

***NOTE**: Please refer to the SICommModes.rtf document for more details.*

6.1 PCI Slave: Target Mode Access

Target accesses are the simplest form of communication, with the SI-C33DSP-PCI card acting as the PCI slave to the host computer.

In this mode, the host performs direct accesses over the PCI bus to all of the DSP's external devices, which includes all of the external SRAM as well as the expansion daughter card. All handshaking between the DSP and the PCI9054 is implemented in hardware, therefore it is not relevant that the DSP be inactive (in reset) or active running code.

While the DSP is inactive or in reset, its bus is in a high impedance state thereby avoiding bus contention with the host. It is at this time the host downloads the DSP's COFF file. While the DSP is active running code, the DSP is put into a high-impedance state and held until the host completes the transaction. Once completed, the DSP is released and resumes code execution.

Although this method is simple and free from software dependency, it requires constant host processor intervention per access.

***NOTE**: The C33's internal memory and registers cannot be accessed directly by the PCI9054 in this communication mode. Please refer to the C33 data sheet and communication modes below for details on how to access its internal memory and registers.*

6.2 PCI Bus Master: PCI9054 DMA Access

While the PCI9054 is the PCI bus master, its DMA engine performs data transfers in block mode thereby exempting the host from intervention yielding increased host CPU bandwidth for other tasks.

On the PCI9054's local side however, the control logic behaves in an identical manner as target accesses since all handshaking is performed in hardware, and therefore it is not relevant that the DSP be inactive (in reset) or active running code.

In practicality however, this mode is most useful while the DSP is active running code. Like target accesses, while the DSP is active running code, the DSP is put into a high-impedance state and held until the host completes the transaction. Once completed, the DSP is released and resumes code execution.

Unlike target access, multiple data exchanges occur during this time. The maximum number of data accesses is determined by the PCI9054 register for bursting, namely DMASIZ0. By default, a maximum of 8 data cycles are allowed to be bursted, thus disallowing the PCI9054 to monopolize the DSP bus.

***NOTE**: The C33's internal memory and registers cannot be accessed directly by the*

PCI9054 in this communication mode. Please refer to the C33 data sheet and communication modes section of the PCI9054 for details on how to access the internal memory and registers.

6.3 PCI Bus Master: Add-on DSP programmed I/O or DSP DMA Accesses

In the PCI Initiator/Local Master modes, the PCI9054's local bus' DSP processor becomes the PCI Bus Master. The DSP performs accesses directly to host memory, which can be ideal in applications requiring very efficient use of shared system resources.

Even though the DSP is the PCI bus master, it can perform the data transfer using standard I/O instructions, or the DSP's DMA controller may be configured to perform the data transfer. In either case, the PCI9054 is configured to the same PCI Initiator mode and thus behaves identically.

In order for the DSP to perform direct accesses to host memory, a combination of hardware and software handshaking must take place between the DSP and the host. The handshaking sequence is as follows:

1. The host downloads a list of parameters to the DSP's local SRAM, using target mode.
2. The host alerts the DSP of a new transfer request by generating an interrupt to the DSP, namely INT0, again using target mode.
3. Once interrupted, a DSP ISR will examine the parameters to determine the type of transfer it must perform, and then it proceeds with the transfer.
4. Once the DSP completes the transfer, it must alert the host by either 1) causing a PCI interrupt by accessing the PCI9054's Local-to-PCI doorbell register, namely the L2PDBELL register; or 2) it sets a flag which is polled by the host.
5. In turn, the host closes out the transaction by either 1) performing its own ISR in response to the DSP to host interrupt, or 2) acknowledging the flag register.

Please refer to the DSP and host level source code for further details.

***NOTE**: Since the PCI Initiator/Local Master modes require handshaking between the PCI9054 and the DSP, they are therefore the only methods to access internal C33 registers.*

6.4 Communication Registers

The Communication Registers reside inside of the DSP's SRAM Bank 0 memory, mapped as shown below:

***NOTE**: For DSP programmed I/O and DSP DMA accesses to occur, a protocol involving software and hardware handshaking is implemented. Therefore, it is imperative that the DSP is active running code, which requires downloading the DSP code (COFF file contents) before the DSP is able to perform as the PCI bus master.*

***NOTE**: Please refer to the SICommModes.rtf document for more details.*

Register	Name	Host BADDR2+Offset (Byte Boundary)	DSP Address (DWord Boundary)
CommReg0	Mode/Control	0xC0	0x30
CommReg1	Count (DWords)	0xC4	0x31
CommReg2	DSP Source Address	0xC8	0x32
CommReg3	DSP Destination Address	0xCC	0x33
CommReg4	Flag/Status	0xD0	0x34
CommReg5	Data	0xD4	0x35
CommReg[6:7]	n/a	0xD8-0xDC	0x36-0x37
CommReg8	Host->DSP Message Flag/Status	0xE0	0x38
CommReg9	Host->DSP Message Register	0xE4	0x39
CommReg10	DSP->Host Message Flag/Status	0xE8	0x3A
CommReg11	DSP->Host Message Register	0xEC	0x3B

***CommReg0:** Communication Mode/Control Register.*
Written by host processor, defines the type of transfer to take place. Written by the host using target mode.

***CommReg1**: Count Register in DWords.*
Written by host processor, defines the size or number of data values to transfer.

***CommReg2**: Source Address Register.*
Written by host processor, defines the source address within the DSP's memory range, of the data to be transferred.

***CommReg3**: Destination Address Register.*
Written by host processor, defines the destination address within the DSP's memory range, of the data to be transferred.

***CommReg4**: Flag/Status Register.*
Accessed by both the host and the DSP, serves as a status indicator for synchronizing both the host and DSP processors.

***CommReg5**: Data Register.*
Optional register used only for "Host Polling" transfer modes. Accessed by both the host and the DSP, and serves as an intermediate depository for the current data value being transferred.

NOTE: *Only valid for those transfers requiring both processors to poll one another. Otherwise, the data is transferred through a FIFO buffer inside of the PCI bridge device.*

CommReg[7:6]*: Undefined.*
Unreserved, open for general usage.

CommReg8*: Host->DSP Message Flag/Status Register.*
Accessed by both the host and the DSP, serves as a status indicator for synchronizing both processors when the Host 'initiator' sends a message to the DSP 'recipient'. The Host will write a '0x1' value to indicate that a message has been sent, and will expect the DSP 'recipient' to clear or write a '0x0' value to indicate that it has completed the message processing.
NOTE: *The Host 'initiator' also simultaneously updates the DSP INTO register defined below so as to cause an interrupt to the DSP 'recipient'.*

CommReg9*: Host->DSP Message Register.*
A user defined value written by the Host when the Host 'initiator' is to pass a user defined message to the DSP 'recipient'.

CommReg10*: DSP->Host Message Flag/Status Register.*
Accessed by both the host and the DSP, serves as a status indicator for synchronizing both processors when the DSP 'initiator' sends a message to the Host 'recipient'. The DSP will write a '0x1' value to indicate that a message has been sent, and will expect the Host 'recipient' to clear or write a '0x0' value to indicate that it has completed the message processing.
NOTE: *The DSP 'initiator' also simultaneously updates the PLX's Doorbell register defined below so as to cause an interrupt to the Host 'recipient'. The driver automatically and immediately clears the Doorbell register after it detects the interrupt.*

CommReg11*: DSP->Host Message Register.*
A user defined value written by the DSP when the DSP 'initiator' is to pass a user defined message to the Host 'recipient'.

NOTE: *Though optional, when running the sample C33INI.OUT COFF file, there are the following heartbeats:*

Register	**Description**	**Host BADDR2+Offset (Byte Boundary)**	**DSP Address (DWord Boundary)**
Main Heartbeat	Always generated by DSP	0x7C	0x1F
Heartbeat for DMA, communications	Only generated by DSP with host-DSP communications.	0x80	0x20
Secondary Heartbeat	Optionally generated by DSP with Addon Initiated communications.	0x90	0x24

Interrupts are used in conjunction with the Communication Registers, where an 'initiator' interrupts a 'recipient' in order to alert it of a pending transaction.

Register	Name	Host Offset (Byte Boundary)	DSP Address (DWord Boundary)	Interrupt Direction
CSR1	DSP INT0	BADDR2 + 0x7C,0004 Internal FPGA Register.	n/a	Host->DSP
PLX Doorbell	L2PDBELL	BADDR0 + 0x64 Internal PLX Register.	0xFF,0039	DSP->Host

***Control/Status Register 1 (CSR1)**: DSP INT0.*
When the host performs a target write to this location, the onboard control logic generates a pulse on the C33 INT0 interrupt line, which alerts the C33 that either a data transfer or a message is to be performed.

***NOTE**: The DSP INTO register is used as an additional communication register as it facilitates synchronizing Host->DSP initiated events such as all data transfers and Host->DSP messages. Please refer to the SICommModes.rtf document for more details.*

***PLX Doorbell**: L2PDBELL.*
Once the DSP completes the transfer, it must alert the host by causing a PCI interrupt by accessing the PCI9054's Local-to-PCI doorbell register, namely the L2PDBELL register. The value written by the DSP will vary depending if the DSP has completed a data transfer (Add-on Init mode) or a new message to the Host is pending.

***NOTE**: The PLX's Doorbell register is used as an additional communication register as it facilitates synchronizing DSP->Host initiated events such as 'Add-on Initiated' data transfers and DSP->Host messages. Please refer to the SICommModes.rtf document for more details.*

7.0 Control/Status Registers

There are two Control and Status registers that reside inside the FPGA, and are not accessible by the DSP as shown below:

Register	Name	Host BADDR2 +Offset (Byte Boundary)	Bit Position	Description
CSR0	DSP RESET	0x7C,0000	0	Resets DSP 0 = DSP Reset, inactive. 1 = DSP Set, active to run code.
			31:1	n/a
CSR1	DSP INT0	0x7C,0004	31:0	Decoded register where a host write generates pulse to DSP INT0 line.

Control/Status Register 0 (CSR0)*: DSP RESET.*
In order to inactivate the DSP, it must be placed in an idle or 'Reset' state by the host. While in reset, all of the DSP address and data lines are placed in a high impedance state. Once the DSP is removed from reset, it automatically fetches the Reset Vector in memory location 0x0. Please refer to the C33 data sheet for further details. The RESET register is accessed by the host in target mode.

NOTE*: The RESET VECTOR and the RESET REGISTER are completely separate and not to be confused. The RESET VECTOR contains the pointer to the first instruction of executable DSP code, while the RESET REGISTER is only accessible by the host so as to impose a reset condition on the DSP.*

Control/Status Register 1 (CSR1)*: DSP INT0.*
When the host performs a target write to this location, the onboard control logic generates a pulse on the C33 INT0 interrupt line, which alerts the C33 that either a data transfer or a message is to be performed.

8.0 COFF File Downloads

In order for the DSP to be active, a host based application that parses through the contents of a COFF file (Common Object File Format) creates an image that must be downloaded to the DSP's boot memory. The COFF file contains sections with the actual code to be run on the C33 DSP. Since the DSP may be in an unknown state, the COFF file contents must be written to the DSP memory using target mode accesses.

The following sequence of events must occur for a successful COFF load over the PCI:

1. Place the DSP in reset by writing a "0" value to the RESET REGISTER.
2. While maintaining the DSP in the reset state, download an image extracted from the COFF file contents.
3. Remove the DSP reset by writing a "1" to the RESET REGISTER.
4. The DSP starts execution from its reset vector.

These steps are clearly outlined in the source code of the supplied utilities.

9.0 Host Address Mapping

This section describes the DSP's memory mapping as seen from the host via the PCI9054. The DSP's external space is directly mapped into the PCI9054's BADDR2 region (PLX User Space 0), with a total depth of 8M bytes, or 2Mx32 words.

The C33's address bus is 24 bits wide, evenly divided into four (4) subregions or pages of 4x32 words, yielding a total depth of 16Mx32 words. Please consult the C33 data sheet for more details. The criteria was to have each one of the C33's four 4Mx32 subregions or pages contiguously mapped into four 2M byte or 512Kx32 word subregions as seen from the host.

When the host is accessing the DSP's memory space, the following offsets are used:

Range (Bytes)	DSP Page (External Space)	Resource	Host BADDR2 + Offset (Byte Boundary)	DSP Base Address Range (DWord Boundary)
2M	0	DSP SRAM Bank 0 (Comm Regs: 0xC0-0xCB)	0x00,0000-0x1F,FFFF	0x00,0000-0x3F,FFFF (Comm Regs: 0x30-0x3B) (unused above 0x08,0000)
2M	1	DSP SRAM Bank 1 (Optional)	0x20,0000-0x3F,FFFF	0x40,0000-0x7F,FFFF (unused above 0x48,0000)
2M	2 (unused)	n/a	0x40,0000-0x5F,FFFF (unused)	0x80,0000-0xBF,FFFF (unused)
2M	3	unused or unavailable range	0x60,0000-0x7F,FFFF	0xC0,0000-0xFD,FFFF (Refer to DSP Mapping)
		Expansion Module	0x78,0000-0x7B,FFFF	0xFE,0000-0xFE,FFFF
		FPGA: CSRs[0:1]	0x7C,0000-0x7C,0007	n/a

DSP memory space on Page 0 (Byte boundary):
0x000000 to 0x1FFFFF (512Kx32).

DSP memory space on Page 1 (Byte boundary):
0x200000 to 0x3FFFFF (512Kx32)

Page 2 pin inactive (not used):
0x400000 to 0x5FFFFF (512Kx32)

DSP memory space on Page 3 (Byte boundary):
0x600000 to 0x7FFFFF (512Kx32)

Detail for DSP Page 3 as seen from host:
Daughtercard Expansion connector (Byte boundary):
0x780000 to 0x7BFFFF (64Kx32)

DSP Reset register (Byte boundary):
0x7C0000 (1x32)

DSP Interrupt 0 (Byte boundary):

0x7C0004 (1x32)

The actual decoding only uses a few bits, with the host side regions mirrored throughout the DSP's memory range. The address decoding is implemented as shown below in binary format, with "x" reflecting don't cares values:

PAGE0:	0000 0xxx xxxx xxxx xxxx xxxx
PAGE1:	0000 1xxx xxxx xxxx xxxx xxxx
PAGE2:	0001 0xxx xxxx xxxx xxxx xxxx
PAGE3:	0001 1xxx xxxx xxxx xxxx xxxx

Page 3 detail:

Daughtercard expansion:	0001 1xx0 xxxx xxxx xxxx xxxx
Reset Register:	0001 1111 xxxx xxxx xxxx xxx0
IINT0 Register:	0001 1111 xxxx xxxx xxxx xxx1

***NOTE**: Each subregion mapped from the host side is mirrored eight (8) times into each corresponding subregion or page on the DSP side. Mirroring occurs since the 3 most significant address bits on the host side are ignored. The host side only reserves depths of 512Kx32 words, compared to the C33's page depths of 4Mx32 words. This was done to limit the BADDR2 depth to never exceed the maximum physical SRAM bank of 512Kx32 placed on each of the first two pages on the C33.*

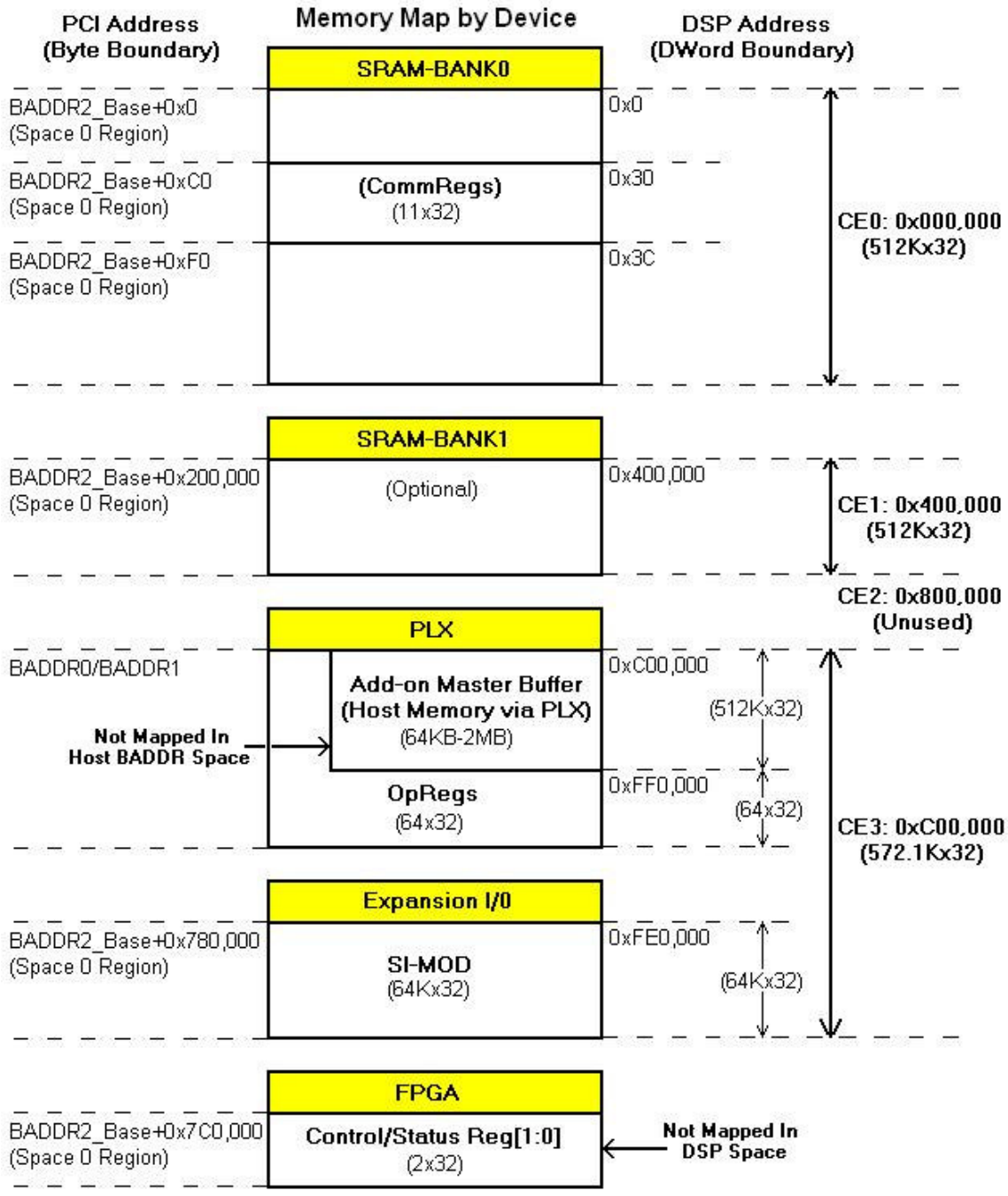
10.0 DSP Address Mapping

When the DSP is running, its external memory map is as follows:

Physical Range Out of 4Mx32 Available (DWords)	DSP Page (External Space)	Resource	DSP Base Address Range (DWord Boundary)
512K	0	DSP SRAM Bank 0 (Comm Regs: 0xC0-0xCB)	0x00,0000-0x3F,FFFF (Comm Regs: 0x30-0x3B) (unused above 0x08,0000)
512K	1	DSP SRAM Bank 1 (Optional)	0x40,0000-0x7F,FFFF (unused above 0x48,0000)
512K	2 (unused)	unused	0x80,0000-0xBF,FFFF (unused)
4M decoded Only 572.1K used.	3	Add-on Init Buffer, Expansion Module, PLX OpRegs	0xC0,0000-0xFF,FFFF

DSP Page 3 detail:

Resource	DSP Base Address Range (DWord Boundary)
Add-on Init Buffer (512Kx32) (Mapped into Host Memory)	0xC0,0000-0xC7,FFFF
Unused (3.5Mx32)	0xC8,0000-0xFD,FFFF
Expansion Module (64Kx32)	0xFE,0000-0xFE,FFFF
PLX OpRegs (64x32)	0xFF,0000-0xFF,004F
Unused (63.9Kx32)	0xFF,0050-0xFF,FFFF



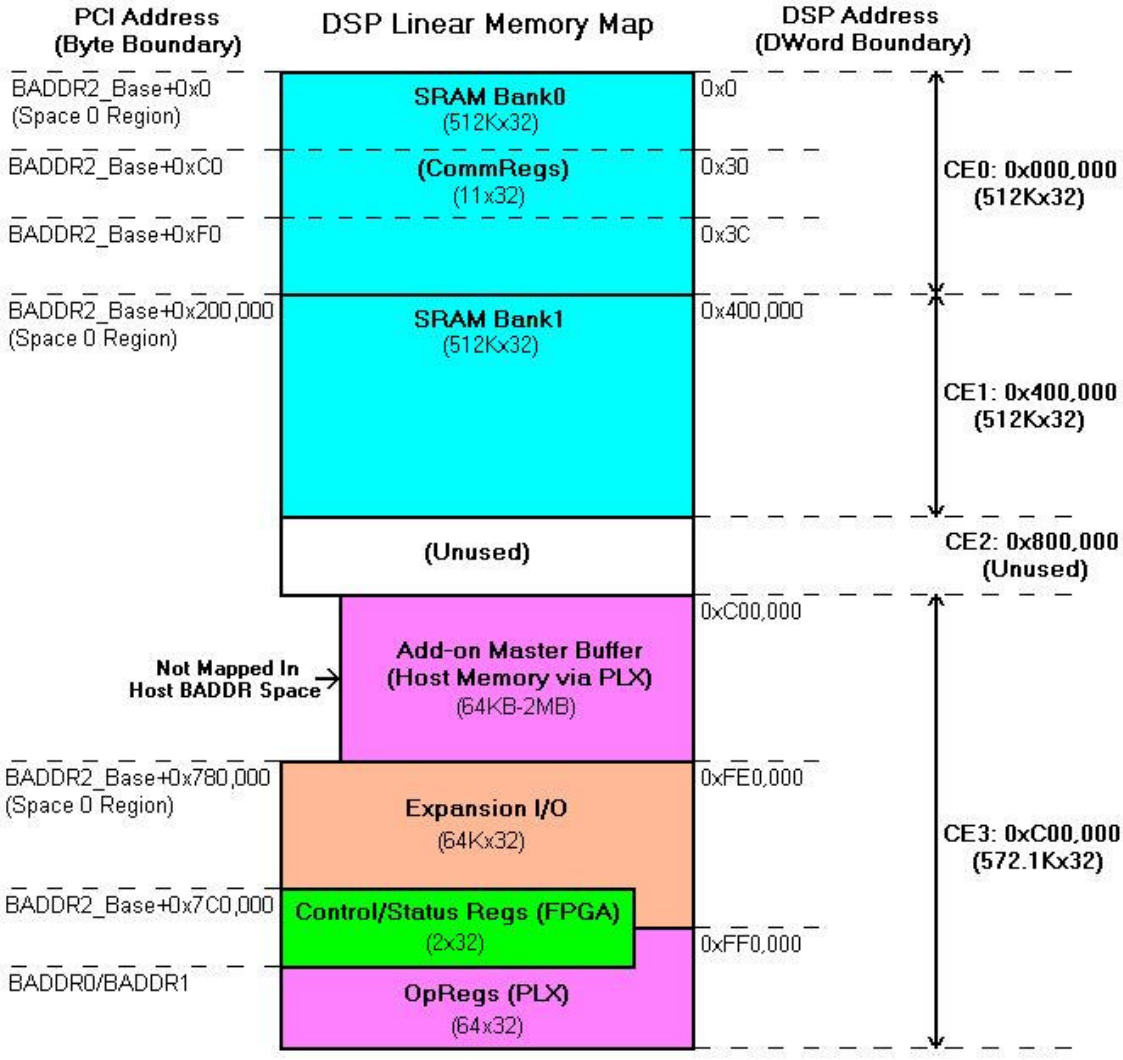
The actual decoding only uses a few bits, with the host side regions mirrored throughout the DSP's memory range. The address decoding is implemented as shown below in binary format, with "x" reflecting don't cares values:

DSP PAGE0:	00xx xxxx xxxx xxxx xxxx xxxx
DSP PAGE1:	01xx xxxx xxxx xxxx xxxx xxxx
DSP PAGE2 (Unused):	10xx xxxx xxxx xxxx xxxx xxxx
DSP PAGE3:	11xx xxxx xxxx xxxx xxxx xxxx

DSP Page 3 detail:

DSP access to host memory: 110x xxxx xxxx xxxx xxxx xxxx
PCI9054 opregs (including doorbell):1111 1111 xxxx xxxx xxxx xxxx
Daughtercard expansion connector: 1111 1110 xxxx xxxx xxxx xxxx

***NOTE**: Each subregion mapped from the host side is mirrored eight (8) times into each corresponding subregion or page on the DSP side. Mirroring occurs since the 3 most significant address bits on the host side are ignored. The host side only reserves depths of 512Kx32 words, compared to the C33's page depths of 4Mx32 words. This was done to limit the BADDR2 depth to never exceed the maximum physical SRAM bank of 512Kx32 placed on each of the first two pages on the C33.*



11.0 Address Conversion Performed by Host Application Software

In order to reduce the confusion caused by different addresses accessing the same hardware, the supplied utilities perform address conversions in software. Addresses are converted to DSP addresses when host side (PCI9054) addresses are used. The following table shows the conversion:

Address Conversions			
PAGEn Access	**Parameters Passed From Software**	**Actual Value Used by PCI9054 for Host Accesses**	**Actual Value Used by DSP**
0	0x00,0000 to 0x3F,FFFF	0x00,0000 to 0x07,FFFF	0x00,0000 to 0x3F,FFFF
1	0x40,0000 to 0x7F,FFFF	0x08,0000 to 0x0F,FFFF	0x40,0000 to 0x7F,FFFF
2	0x80,0000 to 0xBF,FFFF	0x10,0000 to 0x17,FFFF	0x80,0000 to 0xBF,FFFF
3	0xC0,0000 to 0xFF,FFFF	0x18,0000 to 0x1F,FFFF	0xC0,0000 to 0xFF,FFFF

NOTE: The host target and busmaster accesses can have two locations that show identical values. For example,

Access to 0xFE,0300 is the same as access to 0x1E,0300 (expansion card)
Access to 0xFF,0000 is the same as access to 0x1F,0000 (DSP Reset register)
Access to 0xFF,0001 is the same as access to 0x1F,0001 (DSP IINT0 register)
Access to 0x08,0000 is the same as access to 0x40,0000

The DSP I/O or DSP DMA accesses to host memory do not have this conversion, nor does the C33 on-chip memory and register.

12.0 Mirroring on Page 0 and Page 1 Memory Ranges

Each subregion mapped from the host side is mirrored eight (8) times into each corresponding subregion or page on the DSP side. Mirroring occurs since the 3 most significant address bits on the host side are ignored, only the lower 19 address bits are physically tied to SRAM. Any host access outside of the valid memory address is mirrored throughout the DSP address space within the page.

The host side only reserves depths of 512Kx32 words, compared to the C33's page depths of 4Mx32 words. This was done to limit the BADDR2 (Local Side Space 0) depth to never exceed the maximum physical SRAM bank of 512Kx32 placed on each of the first two pages on the C33.

For example, in the case that 128Kx32 SRAM is present on DSP Page 0, the following accesses are the same:

0x00,0000 to 0x01,FFFF
0x02,0000 to 0x03,FFFF
0x04,0000 to 0x05,FFFF
0x06,0000 to 0x07,FFFF
0x08,0000 to 0x09,FFFF
0x0A,0000 to 0x0B,FFFF
0x0C,0000 to 0x0D,FFFF
0x0E,0000 to 0x0F,FFFF
0x10,0000 to 0x11,FFFF
.
.
.
0x3E,0000 to 0x3F,FFFF

For 128Kx32 SRAM present on DSP Page 1, the following accesses are the same:

0x40,0000 to 0x41,FFFF
0x42,0000 to 0x43,FFFF
0x44,0000 to 0x45,FFFF
0x46,0000 to 0x47,FFFF
0x48,0000 to 0x49,FFFF
0x4A,0000 to 0x4B,FFFF
0x4C,0000 to 0x4D,FFFF
0x4E,0000 to 0x4F,FFFF
0x50,0000 to 0x51,FFFF
.
.
.
0x7E,0000 to 0x7F,FFFF

In the case that 512Kx32 SRAM is present on DSP Page 0, the following accesses are the same:

0x00,0000 to 0x07,FFFF
0x08,0000 to 0x0F,FFFF
0x10,0000 to 0x17,FFFF
0x18,0000 to 0x1F,FFFF
0x20,0000 to 0x27,FFFF
0x28,0000 to 0x2F,FFFF
0x30,0000 to 0x37,FFFF
0x38,0000 to 0x3F,FFFF

For 512Kx32 SRAM present on DSP Page 1, the following accesses are the same:

0x40,0000 to 0x47,FFFF
0x48,0000 to 0x4F,FFFF
0x50,0000 to 0x57,FFFF
0x58,0000 to 0x5F,FFFF
0x60,0000 to 0x67,FFFF
0x68,0000 to 0x6F,FFFF
0x70,0000 to 0x77,FFFF
0x78,0000 to 0x7F,FFFF

The advantage of mirroring is the contiguous appearance on the host side of the first two banks of external memory. For example, a board with 512Kx32 SRAM on DSP Page 0 and 512Kx32 SRAM on DSP Page 1 could be treated as having a single 1Mx32 SRAM block of memory with the following mapping:

Host/PCI9054 Mapping	DSP Mapping
0x00,0000 to 0x0F,FFFF	0x38,0000 to 47,FFFF

Host BADDR2 + Offset (Byte Boundary)		**SRAM Size 512Kx32**	**DSP Base Address Range (DWord Boundary)**	
Single 4Mx8 (1Mx32) block	**0x00,0000-0x1F,FFFF** (No Mirroring from Host)	**SRAM Bank 0**	**0x00,0000-0x07,FFFF**	**Single 512Kx32 block**
		First Mirrored SRAM Bank 0	0x08,0000-0x0F,FFFF	
		..	..	
		Last Mirrored SRAM Bank 0	*0x38,0000-0x3F,FFFF*	**Single 1Mx32 block**
	0x20,0000-0x3F,FFFF (No Mirroring from Host)	**SRAM Bank 1 (Optional)**	**0x40,0000-0x47,FFFF**	
		First Mirrored SRAM Bank 1	0x48,0000-0x4F,FFFF	
		..	..	
		Last Mirrored SRAM Bank 1	0x78,0000-0x7F,FFFF	

Host BADDR2 + Offset (Byte Boundary)		**SRAM Size 128Kx32**	**DSP Base Address Range (DWord Boundary)**	
Single 512Kx8 (128Kx32) block	**0x00,0000-0x07,FFFF**	**SRAM Bank 0**	**0x00,0000-0x01,FFFF**	**Single 128Kx32 block**
	0x08,0000-0x0F,FFFF	First Mirrored SRAM Bank 0	0x02,0000-0x03,FFFF	
	0x10,0000-0x17,FFFF	..	..	
Single 1Mx8 (256Kx32) block	*0x18,0000-0x1F,FFFF*	*Last Mirrored SRAM Bank 0*	*0x3E,0000-0x3F,FFFF*	**Single 256Kx32 block**
	0x20,0000-0x27,FFFF	**SRAM Bank 1 (Optional)**	**0x40,0000-0x41,FFFF**	
	0x28,0000-0x2F,FFFF	First Mirrored SRAM Bank 1	0x42,0000-0x43,FFFF	
	0x30,0000-0x37,FFFF	..	..	
	0x38,0000-0x3F,FFFF	Last Mirrored SRAM Bank 1	0x7E,0000-0x7F,FFFF	

13.0 JTAG Debugging

When power is first applied to the DSP card, the JTAG port is limited to boundary scan operations. However, for code debugging several of the following steps must be observed:

Step 1: Load Onboard Logic with FPGA Loader Utility.
Logic must be present inside of the FPGA for proper system operation. By default, no external FPGA SPROM is supplied and therefore the FPGA must be loaded using software. If an external FPGA SPROM is present, the FPGA is automatically loaded and hence no utility is required.

NOTE: *Please refer to the FPGALoad documentation for more details.*

Step 2: Launch Code Composer Studio.
After launching CCS, invoke the 'Debug>>Run Free' command.

NOTE: *It is imperative to invoke this step before attempting any PCI accesses to the DSP card while the JTAG device is simultaneously attached, otherwise host system will lock up.*

Step 3: Activate the DSP by COFFLoading with the First Instance via the PCI Bus.
In order for the DSP to run in the emulation mode, it must be activated by deasserting the 'Reset' line, which is achieved by a COFF load from the host. The COFF file with the minimum required initialization code is fetched from the bootMEM by the DSP when it is first activated or removed from 'Reset', meanwhile the emulation mode is automatically set by internal DSP hardware. The bootMEM contents are typically loaded from the host before the DSP is activated using using any SI utilities (QuX or SISAMPLEs).

NOTE: *For PCI COFF loads, the DSP project must exclude JTAG and debug support, as those supplied by SI or one created using SI foundation projects such as 'C33ini' or 'c_proj', where the resultant loadable COFF file is to be created in 'Release' mode.*

Step 4: COFF Load the DSP with the Second Instance From CCS Via the JTAG Port.
Within CCS, invoke the 'File>>Load Program' command.

NOTE:
1) For JTAG COFF loads, the DSP project must include JTAG and debug support, as supplied in the supplied sample foundation project 'c_proj_jtag', where the resultant loadable COFF file is to be created in 'Debug' mode.
2) Make sure that Code Composer Studio has been properly configured for your JTAG debugger and the C33 chipset.

14.0 Custom Hardware Interface

The SI-C33DSP includes expansion connectors allowing for custom designs, or for attaching 'off the shelf' multifunction I/O modules from Sheldon Instruments. Sheldon Instruments offers several daughter modules for multichannel analog and digital I/O, including 4 to 64 channels of 16 bit ADCs and DACs.

14.1 Logic Analyzer Support

Since there are many logic analyzers from different vendors, no logic analyzer files are provided with the hardware. The hardware contains various test points where the logic analyzer can be used as a debugging tool. Using these test points, the logic analyzer can be used to capture the traffic between hardware components. For example, all DSP address and data lines, along with control lines PAGE, STRB, and H1 can be monitored to allow the logic analyzer to function as real-time disassembler. These signals can also be used as a debugging tool for custom daughtercard development.

In order to use these test points, wires must be soldered onto the test points. Each test point's dimensions are 0.026" outer diameter with 0.012" holes. Typical wire wrap wires, such as Kynar (TM), can be used. These wires should be as short as possible, typically not to exceed 6".

The following DSP signals are available for monitoring:

DSP_A[15..0]	: Lower 16 bits of DSP address
DSP_A[23..16]	: Upper 8 bits of DSP address
DATA[15..0]	: Lower 16 bits of shared data (same for DSP, PLX, Daughtercard)
DATA[31..16]	: Upper 16 bits of shared data (same for DSP, PLX, Daughtercard)
HOLD	: DSP Hold
HOLDA	: DSP Hold acknowledge
DSP_STRB	: DSP Strobe
DSP_PAGE	: DSP Page
DSP_H1 (X_CLK1)	: DSP H1 clock
DSP_RST	: DSP Reset
DSP_RDY	: DSP Ready
DSP_H3 (X_CLK0)	: DSP H3 clock
DSP_INT (X_INT)	: DSP Interrupts
DSP_IACK	: DSP Interrupt Acknowledge
DSP_RW	: DSP Read/Write
XF0, XF1	: DSP Flag
TIMER[1:0]	: DSP Timer Clock
CLKR0, CLKX0	: DSP serial port signals
FSR0, FSX0	: DSP serial port signals
DR0, DX0	: DSP serial port signals

The following PLX signals are available for monitoring:

LA[15..2]	: Lower 14 bits of PLX address
LA[31..16]	: Upper 16 bits of PLX address
DATA[15..0]	: Lower 16 bits of shared data (same for DSP, PLX, Daughtercard)
DATA[31..16]	: Upper 16 bits of shared data (same for DSP, PLX, Daughtercard)
LHOLD	: PLX Hold
LHOLDA	: PLX Hold acknowledge
LCLK	: PLX Clock
ADS	: PLX Control signal
CCS	: PLX Control signal
BLAST	: PLX Control signal
READY	: PLX Control signal
LRESET	: PLX Control signal
BIGEND	: PLX Control signal
LSERR	: PLX Control signal
DP	: PLX Control signal
USERo	: PLX Control signal
USERi	: PLX Control signal
LINT	: PLX Control signal
BREQi	: PLX Control signal
BREQo	: PLX Control signal

15.0 SI-C33DSP-PCI Expansion Hardware Interface for Rev2/3

The SI-C33DSP includes expansion connectors allowing for custom designs, or for attaching 'off the shelf' multifunction I/O modules from Sheldon Instruments. Sheldon Instruments offers several daughter modules for multichannel analog and digital I/O, including 4 to 64 channels of 16 bit ADCs and DACs.

15.1 DSP Bus Connector: 120 Pin 2mm to C33 Bus

A 2mm, 4x30 pin connector (120 contacts total) is used to interface the custom daughtercard to the DSP's data, address, and control busses. 64Kx32 words of the DSP's range is decoded for use by the custom daughtercard.

NOTE: *The memory mapping for the DSPs are as follows:*

a) C3x: 0xFE0000 to 0xFEFFFF, 32 bit addressing. Please refer to the SI-C3xDSP and TMS320C3x User Manuals for further details.

b) C67x: 0x80380000 to 0x803BFFFF, 8 bit addressing. Please refer to the SI-C6x and TMS320C67x User Manuals for further details.

DSP Carrier Pinout:

D.30. **GND**	C.30. **GND**	B.30. **GND**	A.30. **-**
D.29. **+3.3V**	C.29. **+3.3V**	B.29. **+3.3V**	A.29. -
D.28. **+5V**	C.28. **+1.8V**	B.28. **+5V**	A.28. -
D.27. **+12V/+15V**	C.27. RSV2	B.27. **+12V/+15V**	A.27. -
D.26. **-12V/-15V**	C.26. **-5V**	B.26. **-12V/-15V**	A.26. -
D.25. **GND**	C.25. **GND**	B.25. **GND**	A.25. -
D.24. FPGA_DATA	C.24. FPGA_INIT	B.24. A15	A.24. -
D.23. FPGA_CLK	C.23. FPGA_PGRMn	B.23. A14	A.23. **-**
D.22. FPGA_DONE	C.22. **GND**	B.22. A13	A.22. -
D.21. X_CLK1	C.21. X_CLK0	B.21. A12	A.21. **-**
D.20. D15	C.20. **GND**	B.20. D31	A.20. -
D.19. D14	C.19. RSV1	B.19. D30	A.19. -
D.18. D13	C.18. X_INT1	B.18. D29	A.18. -
D.17. D12	C.17. X_INT0	B.17. D28	A.17. -
D.16. **GND**	C.16. **GND**	B.16. **GND**	A.16. -
D.15. D11	C.15. A11	B.15. D27	A.15. -
D.14. D10	C.14. A10	B.14. D26	A.14. -
D.13. D9	C.13. A9	B.13. D25	A.13. -
D.12. D8	C.12. A8	B.12. D24	A.12. -
D.11. D7	C.11. A7	B.11. D23	A.11. -
D.10. D6	C.10. A6	B.10. D22	A.10. -
D.9. D5	C.9. A5	B.9. D21	A.9. -
D.8. D4	C.8. A4	B.8. D20	A.8. -
D.7. D3	C.7. A3	B.7. D19	A.7. -
D.6. D2	C.6. A2	B.6. D18	A.6. -
D.5. D1	C.5. A1	B.5. D17	A.5. -
D.4. D0	C.4. A0	B.4. D16	A.4. -
D.3. R/Wn	C.3. X_CS0n	B.3. RSV0	A.3. -
D.2. X_RDYn	C.2. **+5V**	B.2. **+3.3V**	A.2. -
D.1. **GND**	C.1. **GND**	B.1. **GND**	A.1. **-**

15.2 User I/O Connector: 120 Pin 2mm To 100 Pin D-Sub

A 100 pin half pitch (0.050"), Series III DSUB connector is used to interface external user defined signals to a single 2mm, 4x30 pin connector (120 contacts total). This connector is linked to the custom expansion card, and no signals from the DSP are routed. Below is the connection diagram.

***NOTE**:*

a) Signals designated as "USERI/On" correspond with pin "n" of the 100 pin DSub connector.

b) The USERI/O connectors are ABSENT on PC104Plus form factor cards. PC104Plus cards use a separate adapter for accessing USERI/O signals.

User I/O Pinout as seen from the 2MM 4x30 pin connector:

D.30. USERI/O99	C.30. USERI/O100	B.30. USERI/O49	A.30. USERI/O50
D.29. USERI/O97	C.29. USERI/O98	B.29. USERI/O47	A.29. USERI/O48
D.28. USERI/O95	C.28. USERI/O96	B.29. USERI/O45	A.28. USERI/O46
D.27. USERI/O93	C.27. USERI/O94	B.29. USERI/O43	A.27. USERI/O44
D.26. USERI/O91	C.26. USERI/O92	B.29. USERI/O41	A.26. USERI/O42
D.25. -	C.25. -	B.25. -	A.25. -
D.24. USERI/O89	C.24. USERI/O90	B.24. USERI/O39	A.24. USERI/O40
D.23. USERI/O87	C.23. USERI/O88	B.23. USERI/O37	A.23. USERI/O38
D.22. USERI/O85	C.22. USERI/O86	B.22. USERI/O35	A.22. USERI/O36
D.21. USERI/O83	C.21. USERI/O84	B.21. USERI/O33	A.21. USERI/O34
D.20. USERI/O81	C.20. USERI/O82	B.20. USERI/O31	A.20. USERI/O32
D.19. -	C.19. -	B.19. -	A.19. -
D.18. USERI/O79	C.18. USERI/O80	B.18. USERI/O29	A.18. USERI/O30
D.17. USERI/O77	C.17. USERI/O78	B.17. USERI/O27	A.17. USERI/O28
D.16. USERI/O75	C.16. USERI/O76	B.16. USERI/O25	A.16. USERI/O26
D.15. USERI/O73	C.15. USERI/O74	B.15. USERI/O23	A.15. USERI/O24
D.14. USERI/O71	C.14. USERI/O72	B.14. USERI/O21	A.14. USERI/O22
D.13. -	C.13. -	B.13. -	A.13. -
D.12. USERI/O69	C.12. USERI/O70	B.12. USERI/O19	A.12. USERI/O20
D.11. USERI/O67	C.11. USERI/O68	B.11. USERI/O17	A.11. USERI/O18
D.10. USERI/O65	C.10. USERI/O66	B.10. USERI/O15	A.10. USERI/O16
D.9. USERI/O63	C.9. USERI/O64	B.9. USERI/O13	A.9. USERI/O14
D.8. USERI/O61	C.8. USERI/O62	B.8. USERI/O11	A.8. USERI/O12
D.7. -	C.7. -	B.7. -	A.7. -
D.6. USERI/O59	C.6. USERI/O60	B.6. USERI/O9	A.6. USERI/O10
D.5. USERI/O57	C.5. USERI/O58	B.5. USERI/O7	A.5. USERI/O8
D.4. USERI/O55	C.4. USERI/O56	B.4. USERI/O5	A.4. USERI/O6
D.3. USERI/O53	C.3. USERI/O54	B.3. USERI/O3	A.3. USERI/O4
D.2. USERI/O51	C.2. USERI/O52	B.2. USERI/O1	A.2. USERI/O2
D.1. -	C.1. -	B.1. -	A.1. -

User I/O Pinout as seen from the 100 pin DSub connector:

USERI/O100. C30	USERI/O50. A30
USERI/O99. D30	USERI/O49. B30
USERI/O98. C29	USERI/O48. A29
USERI/O97. D29	USERI/O47. B29
USERI/O96. C28	USERI/O46. A28
USERI/O95. D28	USERI/O45. B28
USERI/O94. C27	USERI/O44. A27
USERI/O93. D27	USERI/O43. B27
USERI/O92. C26	USERI/O42. A26
USERI/O91. D26	USERI/O41. B26
USERI/O90. C24	USERI/O40. A24
USERI/O89. D24	USERI/O39. B24
USERI/O88. C23	USERI/O38.A23
USERI/O87. D23	USERI/O37. B23
USERI/O86. C22	USERI/O36. A22
USERI/O85. D22	USERI/O35. B22
USERI/O84. C21	USERI/O34. A21
USERI/O83. D21	USERI/O33. B21
USERI/O82. C20	USERI/O32. A20
USERI/O81. D20	USERI/O31. B20
USERI/O80. C18	USERI/O30. A18
USERI/O79. D18	USERI/O29. B18
USERI/O78. C17	USERI/O28. A17
USERI/O77. D17	USERI/O27. B17
USERI/O76. C16	USERI/O26. A16
USERI/O75. D16	USERI/O25. B16
USERI/O74. C15	USERI/O24. A15
USERI/O73. D15	USERI/O23. B15
USERI/O72. C14	USERI/O22. A14
USERI/O71. D14	USERI/O21. B14
USERI/O70. C12	USERI/O20. A12
USERI/O69. D12	USERI/O19. B12
USERI/O68. C11	USERI/O18. A11
USERI/O67. D11	USERI/O17. B11
USERI/O66. C10	USERI/O16. A10
USERI/O65. D10	USERI/O15. B10
USERI/O64. C9	USERI/O14. A9
USERI/O63. D9	USERI/O13. B9
USERI/O62. C8	USERI/O12. A8
USERI/O61. D8	USERI/O11. B8
USERI/O60. C6	USERI/O10. A6
USERI/O59. D6	USERI/O9. B6
USERI/O58. C5	USERI/O8. A5
USERI/O57. D5	USERI/O7. B5
USERI/O56. C4	USERI/O6. A4
USERI/O55. D4	USERI/O5. B4
USERI/O54. C3	USERI/O4. A3
USERI/O53. D3	USERI/O3. B3
USERI/O52. C2	USERI/O2. A2
USERI/O51. D2	USERI/O1. B2

NOTE: The manufacturer part numbers for the 100 pin, half pitch DSUB socket on the DSP card are as follows: a) Amp - 787169-9, b) Amp - 787170-9, c) Amp - 787362-9, d) Honda - PCS-XE100LFD-HS

15.3 DSP Peripheral Port Connectors

A 2x20 pin (40 contacts), 0.1" pitch connector is used to interface to the C33's serial port, timers, as well as the general purpose IO lines derived from the FPGA:

1. **GND**	2. **GND**
3. DSPSPa2(CLKX0)	4. DSPSPa1(DX0)
5. DSPSPa0(FSX0)	6. DSPSPa5(CLKR0)
7. DSPSPa4(DR0)	8. DSPSPa3(FSR0)
9. -	10. **GND**
11. TIMER0	12. XF0
13. TIMER1	14. XF1
15. **GND**	16. **GND**
17. GPIO0	18. GPIO1
19. GPIO2	20. GPIO3
21. GPIO4	22. GPIO5
23. GPIO6	24. GPIO7
25. GPIO8	26. GPIO9
27. GPIO10	28. GPIO11
29. GPIO12	30. GPIO13
31. GPIO14	32. GPIO15
33. **GND**	34. **GND**
35. -	36. -
37. -	38. -
39. -	40. -

BLUE: DSPSP
PURPLE: DSPTIMERn/XFn
BLACK: GPIO

NOTE: *Only the first 16 lines in a 2x8 pattern are available through the header. The GPIO lines are not operational at the moment.*

15.4 DSP JTAG Port Connector

A 2x7 pin arrangement (13 contacts, with pin 6 omitted to serve as the polarizing pin), 0.1" pitch connector is used to interface to the C3x's JTAG port. Pin 1 is distinguished with a square pad, with pinout as follows:

14. DSP_EMU1	13. DSP_EMU0
12. **GND**	11. TCK
10. **GND**	9. TCK_RTN
8. **GND**	7. TDO
6. No Connect	5. **+3Vdc**
4. **GND**	3. TDI
2. TRESET	1. TMS

16.0 SI-C33DSP-PCI Expansion Hardware Interface for Rev1

The SI-C33DSP includes expansion connectors allowing for custom designs, or for attaching 'off the shelf' multifunction I/O modules from Sheldon Instruments. Sheldon Instruments offers several daughter modules for multichannel analog and digital I/O, including 4 to 64 channels of 16 bit ADCs and DACs.

16.1 DSP Bus Connector: 96 Pin DIN to C33 Bus

A second 96 pin DIN, designated P2, is used to interface the custom daughtercard to the C33's bus. 32Kx32 words are decoded for use by the custom daughtercard, which is mapped into the C33's bus, DWord boundary address space ranging from 0xFE0000 to 0xFEFFFF. Please refer to the TMS320C33 User's Manual for further details.

The following signals still follow the timing outlined in the User Manual, but are regenerated by the FPGA which includes an extra layer of decoding:

X_CLK[1:0]: *37.5Mhz Expansion Clock signals.*
Input signals to expansion card, generated by DSP card. X_CLK0 is the primary clock source, while X_CLK1 is the secondary clock source, both are derived from the DSP's 75Mhz *Hx CLK*.
X_INT[1:0]: *Expansion Card Interrupts to DSP.*
Output signals generated by expansion card. Interrupt signals routed to one of the DSP's external interrupt lines labeled as *EXTINT[3:2]*; with the primary X_INT0 line is routed the DSP's *EXTINT3*, while the secondary X_INT1 line is routed to the DSP's *EXTINT2*. When driven LO, the expansion card is generating an interrupt to the DSP.
X_INTACK: *Interrupt Acknowledge from DSP to Expansion Card.*
Routed to DSP's EXTINTACK line.
R/Wn: *R/Wn, DSP Bus Write Enable signal.*
Input signal to expansion card, generated by DSP card. Routed from DSP's *R/Wn* line. When driven LO, the DSP is performing a write cycle, otherwise a read cycle is assumed.
X_CSn: *Expansion Card Select signal.*
Input signal to expansion card, generated by DSP card. Equivalent to DSP's *CE3* line, with additional logical ANDing with expansion card's decoded address lines. When driven LO, the DSP is performing an access to the expansion card.
X_RDYn: *RDYn, Expansion Card Ready to DSP.*
Output signal generated by expansion card. Routed to DSP's *RDYn* line. When driven HI, hardware wait states are inserted into the DSP's access cycle.
D[31:0]: *D[31:0], Data Bus.*
Bidirectional data bus. Routed from DSP's *D[31:0]* bus.
A[14:0]: *A[14:0], Address Bus.*
Input bus to expansion card, generated by DSP card. Routed from DSP's *A[14:0]* bus.

Below is the pinout:

P2.A1 - +5V	P2.B1 - D0	P2.C1 - +5V
P2.A2 - D1	P2.B2 - A0	P2.C2 - D2
P2.A3 - A1	P2.B3 - D3	P2.C3 - A2
P2.A4 - D4	P2.B4 - A3	P2.C4 - D5
P2.A5 - A4	P2.B5 - D6	P2.C5 - A5
P2.A6 - D7	P2.B6 - A6	P2.C6 - D8
P2.A7 - A7	P2.B7 - D9	P2.C7 - A8
P2.A8 - D10	P2.B8 - A9	P2.C8 - D11
P2.A9 - A10	P2.B9 - D12	P2.C9 - A11
P2.A10 - D13	P2.B10 - A12	P2.C10 - D14
P2.A11 - A13	P2.B11 - D15	P2.C11 - A14
P2.A12 - D30	P2.B12 - MEMSTRBn*	P2.C12 - X_INT1 (C33_INT2n)*
P2.A13 - R/Wn	P2.B13 - D31	P2.C13 - X_INT0 (C33_INT3n)
P2.A14 - D26	P2.B14 - X_INTACK*	P2.C14 - D29
P2.A15 - D25	P2.B15 - XMAP*	P2.C15 - D17
P2.A16 - D21	P2.B16 - D24	P2.C16 - D28
P2.A17 - D18	P2.B17 - D20	P2.C17 - GND
P2.A18 - D23	P2.B18 - GND	P2.C18 - D19
P2.A19 - GND	P2.B19 - D16	P2.C19 - GND
P2.A20 - X_CSn	P2.B20 - +5V	P2.C20 - C33_STRB*
P2.A21 - +5V	P2.B21 - D22	P2.C21 - +5V
P2.A22 - D27	P2.B22 - TIMER1	P2.C22 - XF0
P2.A23 - TIMER0	P2.B23 - N/C	P2.C23 - X_RDYn
P2.A24 - XF1	P2.B24 - GND	P2.C24 - X_CLK0 (C33_H3/2 @ 37.5Mhz)
P2.A25 - X_CLK1 (C33_H1 @ 75Mhz)	P2.B25 - GND	P2.C25 - GND
P2.A26 - GND	P2.B26 - N/C	P2.C26 -
P2.A27 - N/C	P2.B27 - N/C	P2.C27 - GND
P2.A28 - GND	P2.B28 - CLKX0	P2.C28 - CLKR0
P2.A29 - DX0	P2.B29 - DR0	P2.C29 - FSX0
P2.A30 - FSR0	P2.B30 - +12V	P2.C30 - +12V
P2.A31 - +3.3V	P2.B31 - -12V	P2.C31 - -12V
P2.A32 - N/C	P2.B32 - -5V	P2.C32 - -5V

16.2 User I/O Connector: 100 Pin D-Sub to 96 Pin DIN

A 100 pin half pitch (0.050"), Series III DSUB connector, designated J1 (BB1), is used to interface external user defined signals to one of the 96 pin DIN connectors, designated P1. This P1 DIN connector is linked to the custom daughter card. Neither of these connectors contain any signals from the DSP. Below is the connection diagram.

P1.A1 - USERI/O1 (DB1.1)	P1.B1 - USERI/O51 (DB1.51)	P1.C1 - USERI/O2 (DB1.2)
P1.A2 - USERI/O52 (DB1.52)	P1.B2 - USERI/O3 (DB1.3)	P1.C2 - USERI/O53 (DB1.53)
P1.A3 - USERI/O4 (DB1.4)	P1.B3 - USERI/O54 (DB1.54)	P1.C3 - USERI/O5 (DB1.5)
P1.A4 - USERI/O55 (DB1.55)	P1.B4 - USERI/O6 (DB1.6)	P1.C4 - USERI/O56 (DB1.56)
P1.A5 - USERI/O7 (DB1.7)	P1.B5 - USERI/O57 (DB1.57)	P1.C5 - USERI/O8 (DB1.8)
P1.A6 - USERI/O58 (DB1.58)	P1.B6 - USERI/O9 (DB1.9)	P1.C6 - USERI/O59 (DB1.59)
P1.A7 - USERI/O10 (DB1.10)	P1.B7 - USERI/O60 (DB1.60)	P1.C7 - USERI/O11 (DB1.11)
P1.A8 - USERI/O61 (DB1.61)	P1.B8 - USERI/O12 (DB1.12)	P1.C8 - USERI/O62 (DB1.62)
P1.A9 - USERI/O13 (DB1.13)	P1.B9 - USERI/O63 (DB1.63)	P1.C9 - USERI/O14 (DB1.14)
P1.A10 - USERI/O64 (DB1.64)	P1.B10 - USERI/O15 (DB1.15)	P1.C10 - USERI/O65 (DB1.65)
P1.A11 - USERI/O16 (DB1.16)	P1.B11 - USERI/O66 (DB1.66)	P1.C11 - USERI/O17 (DB1.17)
P1.A12 - USERI/O67(DB1.67)	P1.B12 - USERI/O18 (DB1.18)	P1.C12 - USERI/O68 (DB1.68)
P1.A13 - USERI/O19 (DB1.19)	P1.B13 - USERI/O69 (DB1.69)	P1.C13 - USERI/O20 (DB1.20)
P1.A14 - USERI/O70 (DB1.70)	P1.B14 - USERI/O21 (DB1.21)	P1.C14 - USERI/O71 (DB1.71)
P1.A15 - USERI/O22 (DB1.22)	P1.B15 - USERI/O72 (DB1.72)	P1.C15 - USERI/O24 (DB1.24)
P1.A16 - USERI/O73 (DB1.73)	P1.B16 - USERI/O24 (DB1.24)	P1.C16 - USERI/O74 (DB1.74)
P1.A17 - USERI/O25 (DB1.25)	P1.B17 - USERI/O75 (DB1.75)	P1.C17 - USERI/O27 (DB1.27)
P1.A18 - USERI/O76 (DB1.76)	P1.B18 - USERI/O27 (DB1.27)	P1.C18 - USERI/O77 (DB1.77)
P1.A19 - USERI/O28 (DB1.28)	P1.B19 - USERI/O78 (DB1.78)	P1.C19 - USERI/O30 (DB1.30)
P1.A20 - USERI/O79 (DB1.79)	P1.B20 - USERI/O30 (DB1.30)	P1.C20 - USERI/O80 (DB1.80)
P1.A21 - USERI/O31 (DB1.31)	P1.B21 - USERI/O81 (DB1.81)	P1.C21 - USERI/O33 (DB1.33)
P1.A22 - USERI/O82 (DB1.82)	P1.B22 - USERI/O33 (DB1.33)	P1.C22 - USERI/O83 (DB1.83)
P1.A23 - USERI/O34 (DB1.34)	P1.B23 - USERI/O84 (DB1.84)	P1.C23 - USERI/O36 (DB1.36)
P1.A24 - USERI/O85 (DB1.85)	P1.B24 - USERI/O36 (DB1.36)	P1.C24 - USERI/O86 (DB1.86)
P1.A25 - USERI/O37 (DB1.37)	P1.B25 - USERI/O87 (DB1.87)	P1.C25 - USERI/O39 (DB1.39)
P1.A26 - USERI/O88 (DB1.88)	P1.B26 - USERI/O39 (DB1.39)	P1.C26 - USERI/O89 (DB1.89)
P1.A27 - USERI/O40 (DB1.40)	P1.B27 - USERI/O90 (DB1.90)	P1.C27 - USERI/O42 (DB1.42)
P1.A28 - USERI/O91 (DB1.91)	P1.B28 - USERI/O42 (DB1.42)	P1.C28 - USERI/O92 (DB1.92)
P1.A29 - USERI/O43 (DB1.43)	P1.B29 - USERI/O93 (DB1.93)	P1.C29 - USERI/O44 (DB1.44)
P1.A30 - USERI/O94 (DB1.94)	P1.B30 - USERI/O45 (DB1.45)	P1.C30 - USERI/O95 (DB1.95)
P1.A31 - USERI/O46 (DB1.46)	P1.B31 - USERI/O96 (DB1.96)	P1.C31 - USERI/O47 (DB1.47)
P1.A32 - USERI/O97 (DB1.97)	P1.B32 - USERI/O48 (DB1.48)	P1.C32 - USERI/O98 (DB1.98)

DB1.49 - GND
DB1.50 - +5V
DB1.99 - No Connect
DB1.100 - No Connect

17.0 SI-C33DSP-cPCI/PXI Custom Hardware Interface for Rev1

The SI-C33DSP-cPCI/PXI includes expansion connectors allowing for custom designs, or for attaching 'off the shelf' multifunction I/O modules from Sheldon Instruments. Sheldon Instruments offers several daughter modules for multichannel analog and digital I/O, including 4 to 64 channels of 16 bit ADCs and DACs.

17.1 DSP Bus Connector: 50 Pin D-Sub Pair to C33 Bus

A second pair of 50 pin, half pitch D-sub connectors (100 contacts total), designated P4-P5, is used to interface the custom daughtercard to the C33's data, address, and control busses. 64Kx32 words are decoded for use by the custom daughtercard, which is mapped into the C33's primary bus, address space ranging from 0xFE0000 to 0xFEFFFF. Please refer to the TMS320C3x User's Manual for further details. Below is the pinout:

The following signals still follow the timing outlined in the User Manual, but are regenerated by the FPGA which includes an extra layer of decoding:

X_CLK[1:0]: *37.5Mhz Expansion Clock signals.*
Input signals to expansion card, generated by DSP card. X_CLK0 is the primary clock source, while X_CLK1 is the secondary clock source, both are derived from the DSP's 75Mhz *Hx CLK*.
X_INT[1:0]: *Expansion Card Interrupts to DSP.*
Output signals generated by expansion card. Interrupt signals routed to one of the DSP's external interrupt lines labeled as *EXTINT[3:2]*; with the primary X_INT0 line is routed the DSP's *EXTINT3*, while the secondary X_INT1 line is routed to the DSP's *EXTINT2*. When driven LO, the expansion card is generating an interrupt to the DSP.
X_INTACK: *Interrupt Acknowledge from DSP to Expansion Card.*
Routed to DSP's EXTINTACK line.
R/Wn: *R/Wn, DSP Bus Write Enable signal.*
Input signal to expansion card, generated by DSP card. Routed from DSP's *R/Wn* line. When driven LO, the DSP is performing a write cycle, otherwise a read cycle is assumed.
X_CS0n: *Expansion Card Select signal.*
Input signal to expansion card, generated by DSP card. Equivalent to DSP's *CE3* line, with additional logical ANDing with expansion card's decoded address lines. When driven LO, the DSP is performing an access to the expansion card.
X_RDYn: *RDYn, Expansion Card Ready to DSP.*
Output signal generated by expansion card. Routed to DSP's *RDYn* line. When driven HI, hardware wait states are inserted into the DSP's access cycle.
D[31:0]: *D[31:0], Data Bus.*
Bidirectional data bus. Routed from DSP's *D[31:0]* bus.
A[15:0]: *A[15:0], Address Bus.*
Input bus to expansion card, generated by DSP card. Routed from DSP's *A[15:0]* bus.

P5 DSP Carrier B:

B.25. GND	B.50. GND
B.24. +5V	B.49. +3.3V
B.23. +12V/+15V	B.48. +1.8V
B.22. -12V/-15V	B.47. -
B.21. -5V	B.46. -
B.20. -	B.45. -
B.19. D31	B.44. TIMER1
B.18. D30	B.43. TIMER0
B.17. D29	B.42. XF1
B.16. D28	B.41. XF0
B.15. D27	B.40. X_CLK1 (C33_H1 @ 75Mhz)
B.14. D26	B.39. X_INT1 (C33_INT2)
B.13. D25	B.38. X_INTACKn
B.12. D24	B.37. -
B.11. D23	B.36. -
B.10. D22	B.35. -
B.9. D21	B.34. A15
B.8. D20	B.33. A14
B.7. D19	B.32. A13
B.6. D18	B.31. A12
B.5. D17	B.30. A11
B.4. D16	B.29. A10
B.3. -	B.28. -
B.2. -	B.27. +5V
B.1. GND	B.26. GND

P4 DSP Carrier A:

A.25. GND	A.50. GND
A.24. +5V	A.49. +3.3V
A.23. +12V/+15V	A.48. +1.8V
A.22. -12V/-15V	A.47. -
A.21. -5V	A.46. CLK0
A.20. -	A.45. FSX0
A.19. D15	A.44. DX0
A.18. D14	A.43. CLKR0
A.17. D13	A.42. FSR0
A.16. D12	A.41. DR0
A.15. D11	A.40. X_CLK0 (C33_H3/2 @ 37.5Mhz)
A.14. D10	A.39. X_INT0 (C33_INT3)
A.13. D9	A.38. A9
A.12. D8	A.37. A8
A.11. D7	A.36. A7
A.10. D6	A.35. A6
A.9. D5	A.34. A5
A.8. D4	A.33. A4
A.7. D3	A.32. A3
A.6. D2	A.31. A2
A.5. D1	A.30. A1
A.4. D0	A.29. A0
A.3. R/Wn	A.28. X_CS0n
A.2. X_RDYn	A.27. +5V
A.1. GND	A.26. GND

17.2 User I/O Connector: 100 Pin D-Sub to 50 Pin D-Sub Pair

A 100 pin half pitch (0.050"), Series III DSUB connector, designated P8, is used to interface external user defined signals to a pair of 50 pin, half pitch D-sub connectors (100 contacts total), designated P6-P7. This connector pair is linked to the custom daughter card. Neither of these connectors contain any signals from the DSP. Below is the connection diagram.

P7 User I/O B:

B.25. USERI/O100	B.50. USERI/O50
B.24. USERI/O99	B.49. USERI/O49
B.23. USERI/O98	B.48. USERI/O48
B.22. USERI/O97	B.47. USERI/O47
B.21. USERI/O96	B.46. USERI/O46
B.20. USERI/O95	B.45. USERI/O45
B.19. USERI/O94	B.44. USERI/O44
B.18. USERI/O93	B.43. USERI/O43
B.17. USERI/O92	B.42. USERI/O42
B.16. USERI/O91	B.41. USERI/O41
B.15. USERI/O90	B.40. USERI/O40
B.14. USERI/O89	B.39. USERI/O39
B.13. USERI/O88	B.38. USERI/O38
B.12. USERI/O87	B.37. USERI/O37
B.11. USERI/O86	B.36. USERI/O36
B.10. USERI/O85	B.35. USERI/O35
B.9. USERI/O84	B.34. USERI/O34
B.8. USERI/O83	B.33. USERI/O33
B.7. USERI/O82	B.32. USERI/O32
B.6. USERI/O81	B.31. USERI/O31
B.5. USERI/O80	B.30. USERI/O30
B.4. USERI/O79	B.29. USERI/O29
B.3. USERI/O78	B.28. USERI/O28
B.2. USERI/O77	B.27. USERI/O27
B.1. USERI/O76	B.26. USERI/O26

P6 User I/O A:

A.25. USERI/O75	A.50. USERI/O25
A.24. USERI/O74	A.49. USERI/O24
A.23. USERI/O73	A.48. USERI/O23
A.22. USERI/O72	A.47. USERI/O22
A.21. USERI/O71	A.46. USERI/O21
A.20. USERI/O70	A.45. USERI/O20
A.19. USERI/O69	A.44. USERI/O19
A.18. USERI/O68	A.43. USERI/O18
A.17. USERI/O67	A.42. USERI/O17
A.16. USERI/O66	A.41. USERI/O16
A.15. USERI/O65	A.40. USERI/O15
A.14. USERI/O64	A.39. USERI/O14
A.13. USERI/O63	A.38. USERI/O13
A.12. USERI/O62	A.37. USERI/O12
A.11. USERI/O61	A.36. USERI/O11
A.10. USERI/O60	A.35. USERI/O10
A.9. USERI/O59	A.34. USERI/O9
A.8. USERI/O58	A.33. USERI/O8
A.7. USERI/O57	A.32. USERI/O7
A.6. USERI/O56	A.31. USERI/O6
A.5. USERI/O55	A.30. USERI/O5
A.4. USERI/O54	A.29. USERI/O4
A.3. USERI/O53	A.28. USERI/O3
A.2. USERI/O52	A.27. USERI/O2
A.1. USERI/O51	A.26. USERI/O1

17.3 DSP Serial Port Connector

An ancillary 2x10 header connector, designated P3, is available to directly access the DSP's serial port lines, which are buffered through differential line transceivers. The receiver portion is driven by standard DS34C86 devices, while the transmitter portion is driven by standard DS34C87 devices. All differential lines have common mode 100ohm termination resistors in place. The pinout is as follows:

1. +DX0	2. -DX0
3. +CLKX0	4. -CLKX0
5. +FSX0	6. -FSX0
7. GND	8. GND
9. GND	10. GND
11. GND	12. GND
13. GND	14. GND
15. +FSR0	16. -FSR0
17. +CLKR0	18. -CLKR0
19. +DR0	20. -DR0

17.4 Jumpers

The ancillary serial port has jumpers to configure the source of the clock and frame synchronization lines as described below:

JT1: TxD_DIR. Transmitter port signal source selector.

1-2:Tx_E = CLKX0 and FSX0 are sourced externally and are receivers/inputs on the 2x10 header serial port connector.

2-3:Tx_I* = CLKX0 and FSX0 are sourced internally from the DSP and are transmitters/drivers on the 2x10 header serial port connector; default* mode.

JT2: RxD_DIR. Receiver port signal source selector.

1-2:Rx_I = CLKR0 and FSR0 are sourced internally from the DSP and are transmitters/drivers on the 2x10 header serial port connector.

2-3:Rx_E* = CLKR0 and FSR0 are sourced externally and are receivers/inputs on the 2x10 header serial port connector, default* mode.

18.0 Technical Specifications

Processor for SI-C33DSP:

- TMS320VC33 120-150Mhz DSP.
- Single DMA channel.

Memory:

- 128K x 32 bit words one (1) wait state dual access SRAM, expandable to 1M x 32 bit words on C33's primary bus.

Interface to Host:

- Four 32 bit, bi-directional communications modes between TMS320C33 primary bus and the 9054:
  1) Target access mode.
  2) Block Mode DMA Bus Master mode, using the 9054 as the PCI bus master.
  3) PCI Initiated/Local Master with DSP I/O, using the DSP as the PCI bus master.
  4) PCI Initiated/Local Master with DSP DMA, using the DSP's DMA as the PCI bus master.
- The PCI9054's internal operation registers are mapped into the C33's primary bus, address space starting at 0xFF0000.
- INT0 used by C33 for basic communication and DMA transfer initialization; C33_INT3 (X_INT0) and C33_INT2 (X_INT1) available on DSP Bus expansion connectors.

SI-C33DSP-PCI-Rev2/3 Peripheral Expansion:

- One external 100 pin half pitch DSUB connector, and two 120 pin metric socket connectors:
  a) First 120 pin socket is designated P5, for interfacing custom daughter board to the DSP's bus.
  b) Second 120 pin socket is designated P6, for interfacing external user defined signals to custom daughter board. Linked only to externally accessible 100 pin half pitch DSUB connector, P8.
  c) External 100 pin, half pitch (0.050"), Series III DSUB connector, designated P8, for interfacing external user defined signals to P6. AMP part 787169-9, 787170-9, or 787362-9; Thomas & Betts part HFR100RA29CS1.
- P5 120 pin metric connector decodes 8Kx32 words, mapped into the DSP's primary bus, address space ranging from 0xFE0000 to 0xFE1FFF.
- P5 120 pin metric connector contains the following DSP signals:
  a) Address: A[15:0].
  b) Data: D[31:0].
  c) Control: R/Wn, STRB (X_CSn), C33_INT[3:2] (X_INT[1:0]), IACK (X_INTACK), RDYn (X_RDYn), C33_H3/2 @ 37.5Mhz (X_CLK0), C33_H1 @ 75Mhz (X_CLK1)
  d) +1.8V, +3.3V, +5V, +/-12V and GND.
- One 16 pin header for DSP I/O signals: Serial port 0, XF[0:1], TMCK[0:1].

- One 14 pin header for JTAG port.

SI-C33DSP-PCI-Rev1 Peripheral Expansion:

- One external 100 pin half pitch DSUB connector, and two 96 pin DIN socket connectors:

  a) First 96 pin DIN is designated J2, for interfacing custom daughter board to the DSP's bus.

  b) Second 96 pin DIN designated J3, for interfacing external user defined signals to custom daughter board. Linked only to externally accessible 100 pin half pitch DSUB connector, J1.

  c) External 100 pin, half pitch (0.050"), Series III DSUB connector, designated J1, for interfacing external user defined signals to J2. AMP part 787169-9, 787170-9, or 787362-9; Thomas & Betts part HFR100RA29CS1.
- J3 DIN connector decodes 8Kx32 words, mapped into the C33's primary bus, address space ranging from 0xFE0000 to 0xFE1FFF.
- J3 DIN connector contains the following C33 signals:

  a) Address: A[12:0].

  b) Data: D[31:0].

  c) Control: R/Wn, STRB (X_CSn), C33_INT[3:2] (X_INT[1:0]), IACK (X_INTACK), RDYn (X_RDYn), C33_H3/2 @ 37.5Mhz (X_CLK0), C33_H1 @ 75Mhz (X_CLK1)

  d) I/O lines: Serial port 0, XF[0:1], TMCK[0:1].

  e) +1.8V, +3.3V, +5V, +/-12V and GND.
- One 14 pin header for JTAG port.

SI-C33DSP-cPCI/PXI Peripheral Expansion:

- One external 100 pin half pitch DSUB connector (Hippi style), and four 50 pin half pitch DSUB plug connectors (I-Pack style):

  a) First 50 pin DSUB pair, P4-P5, for interfacing custom daughter board to the DSP's bus.

  b) Second 50 pin DSUB pair, P6-P7, for interfacing external user defined signals to custom daughter board. Linked only to externally accessible 100 pin half pitch DSUB connector, P8.

  c) External 100 pin, half pitch (0.050"), Series III DSUB connector, designated P8, for interfacing external user defined signals to P4-P5. AMP part 787169-9, 787170-9, or 787362-9; Thomas & Betts part HFR100RA29CS1.
- Second DSUB connector pairs, P4-P5, decode 8Kx32 words, mapped into the DSP's primary bus, address space ranging from 0xFE0000 to 0xFE1FFF.
- Second DSUB connector pairs, P4-P5, contain the following DSP signals:

  a) Address: A[12:0].

  b) Data: D[31:0].

  c) Control: R/Wn, STRB (X_CSn), C33_INT[3:2] (X_INT[1:0]), IACK (X_INTACK), RDYn (X_RDYn), C33_H3/2 @ 37.5Mhz (X_CLK0), C33_H1 @ 75Mhz (X_CLK1)

  d) I/O lines: Serial port 0, XF[0:1], TMCK[0:1].

  e) +1.8V, +3.3V, +5V, +/-12V and GND.

- One 14 pin header, P2, for JTAG port.
- One 2x10 pin header, P3, for direct access to DSP's serial bus.

Software:

- Win9x/NT/2000/XP and Linux driver support.
- Extensive QuVIEW DSP-resident libraries for LabVIEW, including examples for real time acquisition, signal processing, and control.
- Extensive QuBASE DSP-resident libraries for Visual Basic, including examples for real time acquisition, signal processing, and control.
- Sample code for COFF loaders, PC <-> DSP communications source code and SI-DDK.
- Compatible with separately purchased TI debuggers, C/C++ compilers, assemblers and linkers.

Physical Dimensions & Electrical Requirements:

- SI-C33DSP-PCI: Half size PCI-bus card measuring 6.4"(L) x 3.9"(H).
- SI-C33DSP-cPCI/PXI: 3U size CompactPCI/PXI bus card measuring 160mm(L) x 100mm(H).
- 0.31lbs or 140 grams
- Supply Voltages: 3.3V for all circuitry, and 5V for expansion bus buffers; 3V expansion buffers may be placed on special request. +/-12V supplies passed on to expansion connector and not used by onboard circuitry.
- 1 watt typical (2 watts maximum) with 128Kx32 words SRAM.