Technical Notes on using Analog Devices' DSP components and development tools
Phone: (800) ANALOG-D, FAX: (781) 461-3010, EMAIL: dsp.support@analog.com, FTP: ftp.analog.com, WEB: www.analog.com/dsp

Interfacing SHARC 2106x DSPs to PLX 9080 PCI Bridge Chips

Introduction

PCI is currently the most common local bus standard for PC systems, but it also is very common in embedded systems. In a typical system, one or more DSPs are the main signal processing components of a PCI card, which is controlled by the main processor of the system. While in some PCI based systems, the DSP cards include all of their own I/O, on many others, significant quantities of data are passed over the PCI bus.

PCI is a back-plane with multiple PCI controllers. Some microprocessors contain PCI controllers within them, while other PCI controllers are bridges, designed to interface PCI to a local bus. This, segments the system into a series of local buses. Each local bus can contain local processors and peripherals. While the PCI interface is common to all systems, the local bus interfaces depend on the devices on the bus. Most bridge chips are designed to interface to several different types of microprocessors. Since no bridges directly interface to a SHARC, some glue logic is needed between the SHARC and the bridge chip.

PLX is a leading vendor of PCI solutions. The PLX 9080 is commonly used with the SHARC, an example being the Bittware Snaggletooth PCI board. It supports a 5V local bus. For systems using the 3.3V SHARCs, the PLX 9054 supports a superset of the 9080 capabilities, and supports a 3.3V local bus system. This note will discuss the hardware interface of the SHARC to the PLX 9080 or 9054.

Interface Description

Since the External Port of a SHARC does not match the Local Bus of the PCI 9080, there is need for some glue logic. The complexity of this glue logic depends on how complex the local bus is, and on how much performance is needed over the bus.

SHARC External Port

The external port of the SHARC can be either a local bus master or slave. When a SHARC is bus master, it drives the address and control lines, and the data lines on writes. When a SHARC is bus slave, either to another SHARC or to the host processor, it responds to the control lines, and drives the data lines on reads. The SHARC external port includes the logic to arbitrate for the bus among multiple SHARCs, as well as to yield the bus upon a host bus request. The SHARC addressing scheme is word based, in this case 32 bit words. We will treat the external port as interfacing to 32 bit wide external memory or a 32 bit wide host, only. Table 1 describes the pins of the SHARC external port.

Pin	Name	Type	Description
HBR	Host Bus Request	I/A	Must be asserted by a host processor to request control of the ADSP-2106x's external bus. When HBR is asserted in a multiprocessing system, the ADSP-2106x that is bus master will relinquish the bus and assert HBG. To relinquish the bus, the ADSP-2106x places the address, data, select, and strobe lines in a high-impedance state. HBR has priority over all ADSP-2106x bus requests (BR1-6) in a multiprocessing system.
HBG	Host Bus Grant.	I/O	Acknowledges an HBR bus request, indicating that the host processor may take control of the external bus. HBG is asserted (held low) by the ADSP-2106x until HBR is released. In a multiprocessing system, HBG is output

			by theADSP-2106x bus master and is monitored by all others.
CS	Chip Select	I/A.	Asserted by host processor to select the ADSP 2106x.
REDY	Host Bus Acknowledge	(o/d) O	The ADSP-2106x deasserts REDY (low) to add wait states to an asynchronous access of its internal memory or IOP registers by a host. Open-drain output (o/d) by default; can be programmed in ADREDY bit of SYSCON register to be active drive (a/d). REDY will only be output if the CS and HBR inputs are asserted.
SBTS	Suspend Bus Tristate	I/S	External devices can assert SBTS (low) to place the external bus address, data, selects, and strobes in a high-impedance state for the following cycle. If the ADSP-2106x attempts to access external memory while SBTS is asserted, the processor will halt and the memory access will not be completed until SBTS is deasserted. SBTS should only be used to recover from PAGE faults or host processor/ADSP-2106x deadlock.

Table 1 SHARC External Port Pins

The SHARC external port is designed to provide 0 wait state access to external memory, so the address and data busses are not multiplexed. On a write, address and data busses are both driven on the same cycle, and on a read, the address lines are driven at the start of the cycle and the read lines are sampled at the end. Both reads and writes can be slowed down by inserting wait states or by deasserting ACK to indicate that the SHARC is not yet ready. The SHARC external port can support either synchronous or asynchronous accesses. It contains no signals to support burst accesses.

The external port also includes some FIFOs to buffer data when a SHARC is written to. This enables the throughput of a SHARC to be greater on writes than on reads. Also, due to the internal structure of the SHARC, the response to I/O buffer reads is quicker than that to memory reads. As a result, the highest throughput is achieved by using a SHARC DMA channel than by doing individual memory transfers. Both of these should be controlled by system software to achieve the maximum throughput.

9080 Local Bus

The 9080 local bus interface is designed to support several different types of interfaces. The one used to interface to the SHARC is C mode: 32 bit address, 32 bit data, not multiplexed. The external port assumes a byte addressing scheme, and provides individual byte enables. The 9080 local bus interface does not perform bus arbitration on its own, but includes signals HOLD and HOLDA to request that another arbiter grant it the bus. It is designed for synchronous pipelined burst access, so it drives the address bus for one cycle, then the databus in the following cycle, with the next address on the address bus. The BLAST signal indicates that the last address has been presented on the bus. The READY lines are used to add delay to transfers.

Pin	Name	Type	Description
ADS#	Address Strobe	I/O TS	Indicates a valid address and start of a new Bus access. Asserted for first cloc of a Bus access.
BLAST#	Burst Last 1	I/O TS	Signal driven by current Local Bus Master to indicate last transferin a Bus access.
BTERM#	Burst Terminate	I	For processors that burst up to four Lwords. If Bterm is disabled through the PCI 9080 Configuration registers, the PCI 9080 also bursts up to four Lwords. If enabled, the PCI 9080 continues to burst until a BTERM# input is asserted. BTERM# is a ready input that breaks up a Burst cycle and causes another Address cycle to occur. Used in conjunction with the PCI 9080 programmable wait state generator.
DEN#	Data Enable	O S	Used in conjunction with DT/R# to provide control for data transceivers attached to Local Bus.
DT/R#	Data Transmit/Receive	O TS	Used in conjunction with DEN# to provide control for data transceivers attached to Local Bus. When asserted, signal indicates the PCI 9080 receives data.

LW/R#	Write/Read	I/O TS	Asserted low for reads and high for writes.
LLOCK#	Bus Lock	I	Indicates an atomic operation that may require multiple transactions to complete. Used by the PCI 9080 for Direct Local access to PCI Bus.
LA[31:2]	Address Bus	I/O TS	Carries upper 30 bits of physical address bus. During bursts, LA[31:2] increment to indicate successive Data cycles.
LD[31:0]	Data Bus	I/O TS	Carries 32-, 16-, or 8-bit data quantities depending on bus width configuration.
LBE[3:0]#	Byte Enables	I/O TS	For a 32-bit bus, the four byte enables indicate which of the four bytes are active during a Data cycle: BE3# Byte Enable 3—LD[31:24] BE2# Byte Enable 2—LD[23:16] BE1# Byte Enable 1—LD[15:8] BE0# Byte Enable 0—LD[7:0]
LCLK	Local Processor Clock	I	Local clock input.
LHOLD	Hold Request	O TP	Asserted to request use of Local Bus. The Local Bus arbiter asserts LHOLDA when control is granted.
LHOLDA	Hold Acknowledge	I	Asserted by Local Bus arbiter when control is granted in response to LHOLD. The bus should not be granted to the PCI 9080 unless requested by LHOLD.
LRESETo#	Local Bus Reset Out	O TP	Asserted when the PCI 9080 chip is reset. Used to drive RESET# input of Local processor.
READYi#	Ready In	I	When the PCI 9080 is a Bus Master, indicates that Read data on bus is valid or that a Write Data transfer is complete. Used in conjunction with the PCI 9080 programmable wait state generator.
READYo#	Ready Out	O DTS	When a Local Bus access is made to the PCI 9080, indicates Read data on bus is valid or a Write Data transfer is complete. READYo# can be connected to READYi#.
EOT0#	End of Transfer for DMA Ch 0	I	Terminates current DMA Ch 0 transfer.
EOT1#	End of Transfer for DMA Ch 1	I	Terminates current DMA Ch 1 transfer.

Table 2 9080 Local Bus Pins (Mode C)

Basic Local Bus Operation

In a basic configuration, the goal is to enable the DSP to communicate over the PCI bus, not to achieve optimum bandwidth. For many systems, in which the DSP functions autonomously with limited intervention by a host processor, this goal is sufficient. At a minimum, the local bus must support: single reads and writes from the DSP to the PCI bridge (and the rest of the PCI system), single reads and writes from the PCI bridge to the DSP, and block writes from the PCI bridge (maximum of 4 32-bit words).

Bus Arbitration

SHARCs are able to gluelessly arbitrate for the external bus with each other and with a host processor through the external port. The host processor always has the highest priority on the external bus. If the PCI bridge is configured as a host, and only the PCI bridge can perform burst transfers, then there is no need for the interface logic to perform any bus arbitration. LHOLD from the PCI bridge is inverted to drive the cluster bus HBR#, and LHOLDA is driven by the assertion of HBG#. If LHOLDA is asserted, then the PCI bridge is bus master and it is assumed to be driving the local address bus and control signals. If LHOLDA is deasserted, then the SHARC cluster will handle the bus arbitration among SHARCs and the local address bus and control signals are assumed to come from the master SHARC.

Single Reads and Writes from SHARC Bus Master

The SHARC must be configured in software to interface to a 32 bit external bus. That means that for 48 bit external transfers, the DMA controller must be used, with 32-48 packing mode enabled. 32 bit transfers are unpacked, so they can be done either as direct core accesses, or a DMA with no packing. If HBG is not asserted, then the SHARC cluster will be the source of all address and control signals, except for the ACK. ADDR[30:0] of the SHARC cluster is connected to the bridge's LA[31:2] and the byte enables should be all enabled. The SHARC RD# and WR# pins are multiplexed to drive the LW/R# pin of the bridge. On the first clock cycle of the memory access, the ADS# is asserted to the bridge, and the ACK is deasserted to the SHARC (inserting a wait state), since the data won't be available. On a write from the SHARC, ACK will be asserted in the next cycle, indicating because the bridge can accept the data in the cycle after the address. On a read by the SHARC, ACK will be derived from READYo#.

Single Reads and Writes from PLX 9080 as Host

The SHARC must be configured in software to interface to a 32 bit host processor. These are not the same registers used to set up the DMA channels for the external port. If HBG is asserted, the PLX will be the source of all address and control signals, except for the ACK. The PLX will use synchronous acceses as bus master. Again, ADDR[30:0] of the SHARC cluster is connected to the bridge's LA[31:2], also ADDR[31] should be driven to 0 if a SHARC's internal memory or I/O processor is the intended target. Although the bridge could attempt to use the LBE#, the DSP is not capable of byte selection, so they will be ignored. Therefore, only 32 bit transfers should be attempted. The bridge will drive LW/R#, which will be demuxed into RD# and WR# on the SHARC cluster. The SHARC will not respond to ADS#. On writes, the WR# signal should remain active until the REDY signal is asserted from the target, and the READYo# signal is driven by the PLX indicating that the data on the bus is valid. If REDY is holding off the access, it should also assert READYi# to delay completion of the access. On reads, the RD# signal should remain active until the REDY signal is asserted from the target DSP, and the READYo# and ADS# signals from the bridge indicate that valid data is present. The REDY signal should also assert the READYi# to hold the data on the local bus.

Burst Writes from PLX 9080

The easiest way to support burst writes does not achieve maximum data transfer. This basically alternates address and data cycles on the bus. This is done by deasserting READYi# each time ADS is driven high on the bus, even if no other circumstance calls for it. Other than that, it is the same as a regular PLX write cycle above.

Advanced Local Bus Operation

In a given application, the basic operation above may be sufficient, but if higher performance is required, the interface must support burst reads and writes from both the PLX and the SHARC. Since the SHARC has a very powerful DMA controller onboard, the PLX DMA controller does not need to be supported. Supporting burst mode from the PLX achieving one cycle per word throughput requires the ability for the interface logic to turn the pipelined address bus (the address is presented a cycle before it is needed) into the SHARCs one address, one access, one cycle format. Also, for maximum performance, the EPB should be used, which have a constant address. This will require programming overhead on the 9080. Similarly, supporting burst mode from the SHARC requires turning the SHARC address and data cycle into a pipelined one, and generating BLAST by adding a counter to the external circuitry. This counter will also have to be programmed when setting up a DMA transfer. Details of this high-speed interface will be explored in a later version of this document.

Conclusion

This paper has discussed many of the issues involved in interfacing a SHARC to a PLX 9080 or 9054. The details of a simple interface have been described, and the issues required to achieve better performance are discussed.