TMS320C6000 Expansion Bus Interface to PCI Bus Through PLX PCI9080 Bridge Chip

Zoran Nikolic *Digital Signal Processing Solutions*

ABSTRACT

This application report describes how to interface the Texas Instruments (TI) TMS320C6000™ digital signal processor (DSP) expansion bus to the peripheral component interconnect (PCI) bus using the PCI9080 bridge chip from PLX Technologies. This document contains:

- A block diagram of the interface
- Information required to configure the PCI9080
- Timing diagrams illustrating the interface functionality

Note:
The information presented in this application report has been verified using VHDL simulation.

Contents

List of Figures

List of Tables

1 Interface to the PCI9080

Because the TMS320C6000 DSP is not 5-V tolerant, some voltage protection is required to ensure that the PLX chip does not damage the input/outputs. Figure 1 shows the interface between the expansion bus and the PCI9080. This block diagram also serves as an example of using the C6000™ (C6202 used in this figure) expansion bus in the mixed-signal environment (3 V/5 V). Table 1 lists the PCI PLX9080 to expansion bus pin connections.

C6000 is a trademark of Texas Instruments.

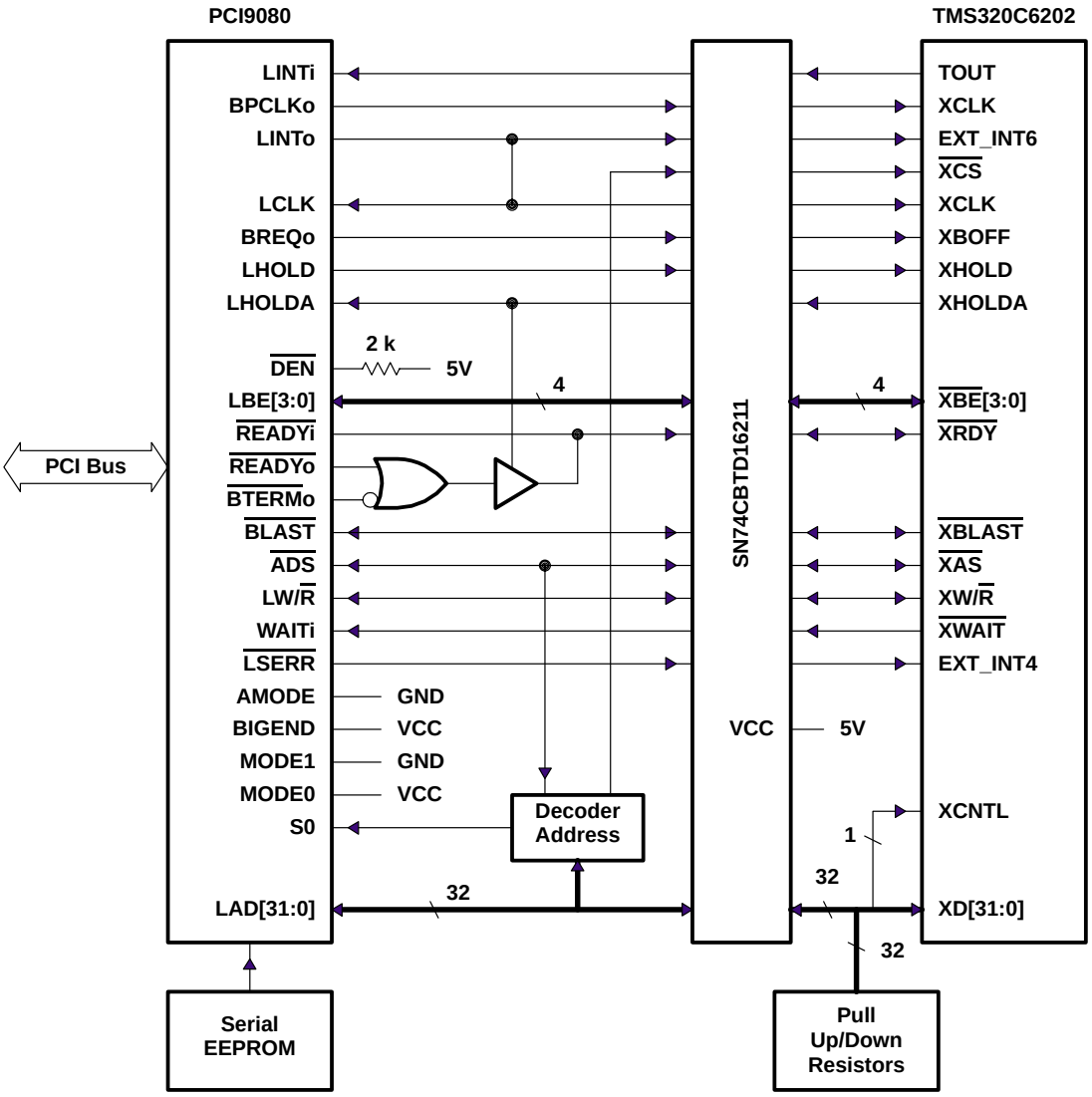

Figure 1. Expansion Bus to PCI Bus Interface Using the PCI9080

Table 1. PLX PCI9080 to Expansion Bus Pin Connections

Expansion Bus Pin	PCI9080 Pin	Comments
XCNTL	A[2]	Address bit of PCI9080 is used as a control signal.
XW/$\overline{R}$	LW/$\overline{R}$	Indicates a read or write access
XD[31:0]	LAD[31:0]	32-bit address/data bus
$\overline{XAS}$	$\overline{ADS}$	Address strobe indicates the transfer of a physical address
$\overline{XCS}$	Address lines are decoded and synchronized using $\overline{ADS}$.	Because the PCI9080 does not have an output chip select line, it is necessary to use some decode logic to generate XCS- (the XBISA and XBD registers of the DSP are mapped at address locations 0x00400004 and 0x00400000 respectively)
$\overline{XBLAST}$	$\overline{BLAST}$	Signal driven by the current bus master to indicate the last transfer in a bus access
XBOFF	BREQo	Asserted to indicate that the PCI9080 requires the bus (if enabled through PCI9080 configuration registers)
EXT_$\overline{INT5}$	LINTo	Any external interrupt can be chosen.
XHOLD	LHOLD	Bus arbitration signal. If the internal bus arbiter of the DSP is enabled, LHOLD can be connected to XHOLD. If the arbiter is disabled, the external arbiter is needed.
XHOLDA	LHOLDA	Bus arbitration signal. If the internal bus arbiter of the DSP is enabled, LHOLDA can be connected to XHOLDA. If the arbiter is disabled, the external arbiter is needed.
XCLK	LCKL	Local clock input. These pins can be connected to BPCLKo of the PCI9080.
$\overline{XBE}$[3:0]	$\overline{LBE}$[3:0]	Expansion bus uses this value on writes only.
$\overline{XRDY}$	READYo, READYi	Ready signal. $\overline{BTERMo}$ signal gates READYo If $\overline{XRDY}$ is an input then: $\overline{XRDY}$ = (BTERMo # $\overline{READYo}$) Please see the *Master/Target Abort* section for details.
EXT_INT4	LSERR_	This signal is used to get the system out of master or target abort situations.
XWAIT_	WAITi	Signal indicating that master is not ready
TOUT	$\overline{LINTi}$	General-purpose output pin (TOUT) is used to generate an interrupt to PCI9080
-	MODE1 = GND MODE0 = Vcc	Bus mode. This combination selects i960Jx bus mode.
-	AMODE = GND	Determines how S[2:0] are used to access the PCI9080 internal registers.
-	DEN = Vcc	DEN_ must be pulled up through a 2-kΩ resistor externally for PCI 9080 to respond to any local bus cycle in Jx mode.
-	S0	Since AMODE = GND, S0 is used as a chip select for PCI9080 (starting from address 0x00300000).
-	BIGEND = Vcc	Big-endian select if BIGEND = GND

The SN74CBTD16211 bus switch is a 24-bit-wide “straight-through” switch with inputs on one side and outputs on the opposite side of the package. The basic bus switch is an NMOS device with the substrate connected to ground, thus allowing bi-directional data flow through the channel. To make this device into a voltage translator, the 5-V Vcc is lowered to 4.3 V through an internal diode. The 3.3-V bus is established because the NMOS bus switch requires a gate-to-source voltage of 1 V to conduct.

The PCI9080 configuration registers are mapped starting from address 0x00300000 in the expansion bus address space. The address decoder (shown in Figure 1) asserts the control signal S0 of the PCI9080, if the DSP is accessing one of the PCI bridge configuration registers. In this system, the XBISA and XBD registers of the DSP are mapped at address locations 0x00400004 and 0x00400000, respectively.

In this interface, the internal bus arbitration of the DSP chip was enabled.

The pull-up/pull-down resistors shown in Figure 1 are used to configure the DSP during reset. The TMS320C6000 DSP reset configuration is presented in Table 3.

2 Configuration

PCI9080 configuration registers (Table 2) can be programmed by an optional serial EEPROM and/or by a local processor. Three local address spaces, -Space 0, Space 1, and expansion ROM-, are accessible from the PCI bus. Each is defined by a set of three registers:

- Local address range (LAS0RR, LAS1RR, and/or EPROMRR). In this design the range is set to 1 MB.
- Local base address (LAS0BA, LAS1BA and/or EPROMBA).
- PCI base address (PCIBAR2, PCIBAR3 and/or PCIERBAR).

A fourth register, the bus region descriptor register (LBRD0, LBRD1) for PCI to local accesses, defines the local bus characteristics for both regions.

Range specifies which PCI address bits to use to decode a PCI access to local bus space. Each bit corresponds to a PCI address bit. Bit 31 corresponds to address bit 31. You should write 1 to all bits that must be included in decode and 0 to all others. Bits in the remap register (remap PCI-to-local addresses) replace the PCI address bits used in decode as the local address bits. PCI reset software determines how much address space is required by writing a value of all ones (1) to a PCI base address register and then reading back the value. PCI9080 returns zeroes in ‘don’t care’ address bits, effectively specifying the address space required. The PCI software then maps the local address space into the PCI address space by programming the PCI base address register.

PCI9080 supports the direct access of the PCI bus by the expansion bus. Master mode must be enabled in the PCI command register. Five registers are used to define local to PCI access:

- Local range register for direct master to PCI (DMRR). In this design for direct master accesses, the range is defined as 1 MB.
- Local base address register for direct master to PCI memory (DMLBAM). In this design, the local base address for direct master to PCI memory is set to 0x00100000.
- Local base address register for direct master to PCI I/O CFG (DMLBAI). In this design, the local base address direct master to PCI I/O CFG is set to 0x00200000.
- PCI configuration address register for direct master to PCI IO/CFG (DMCFGA)
- PCI base address (DMPBAM). In this design, all master transfers are remapped into the PCI address space starting from address 0x00500000.

The range register specifies the local address bits to use to decode a local to PCI access. The local processor can perform only memory access. Therefore, the local base address for direct master to PCI memory register is used to decode an access to PCI memory space. The local base address for direct master to PCI IO/CFG register is used to decode an access to PCI IO space or PCI bus configuration cycle access.

Table 2. PCI9080 Configuration Registers

Register	Name Description	Address (Local)	Value	Description
PCICR	PCI command register	0x04	0x0000 0004	Bit(2) = 1 Enable master mode.
PCILTR	PCI latency timer	0x0D	0x0000 00FF	Bit(7:0) = 0xFF Maximum Latency
PCIERBAR	PCI expansion ROM base reg.	0x30	0x0000 0000	Bit(0) = 0x0 Disable XROM.
LAS0RR	Local address space 0, range register for PCI to local bus	0x 80	0xFFF0 0000	Local address space 0 is mapped into memory space. Range is 1 MB.
LAS0BA	Local address space 0 local address (remap) register	0x 84	0x0040 0000	Enable space 0. Local memory address (remap) is 0x0040 000X.
MARBR	Mode arbitration register	0x 88	0x002A05FF	Bit(7:0) = 0xFF Max. local bus latency Bit(15:8) = 0x05 Local bus pause timer Bit(16) = 0 Disable local bus latency timer bit(17) = 1 Enable local bus pause timer Bit(20:19) = 01 DMA channel 0 always has priority. Bit(21) = 1 Release bus when local bus w/FIFO full
LBRD0	Local address space 0 region descriptor reg.	0x 98	0x4943 00C3	Bit(1:0) = 11 Memory space 0 local bus width 32 bits Bit(6) = 1 Memory space 0 ready input enable Bit(7) = 1 Memory space 0 BTERM input enable Bit(24) = 1 Memory space 0 burst enable Bit(27) = 1 De-assert TRDY when direct slave write FIFO is full. Bit(31:28) = 0100 The value (multiplied by 8) of the PCI target retry delay clocks (only pertains to direct slave when Bit(27) = 1)
DMRR	Local range register for direct master for PCI	0x 9C	0xFFF0 0000	Bit(31:20) = 0xFFF Reserve 1 MB for PCI mastering.
DMLBAM	Local Bus base register for direct master to PCI memory	0x A0	0x0010 0000	Bit(31:16) = 0x0010 Base address for mastering PCI memory
DMLBAI	Local base register for direct master to PCI IO/CFG	0x A4	0x0020 0000	Bit(31:16) = 0x0020 Base address for mastering PCI I/O

Register	Name Description	Address (Local)	Value	Description
DMPBAM	PCI base register (re-map) for direct master to PCI memory	0x A8	0x0050 1013	Bit(0) = 1 Enable PCI mem mastering Bit(1) = 1 Enable PCI IO mastering Bit(12,3) = 10 (or 11) Prefetch up to 8 (or 16) LWords from the PCI bus. Bit(4) = 1 Keep PCI when r/FIFO full Bit(9) = 0 Start PCI access right away. Bit(31:16) = 0x0050 Base address in PCI address space for direct mastering
INTCSR	Interrupt control/status register	0x E8	0x0005 0001	Bit(0) = 1 Enable local bus LSERR when PCI bus target abort or master abort status bit is set in PCISCR. Bit(1) = 0 Disable local bus LSERR when PCI parity error occurs during a PCI 9080 master transfer. Bit(16) = 1 Local interrupt output enable Bit(18) = 1 Local DMA channel 0 interrupt enable
CNTRL	Init control register	0xEC	0x1001 767E	Bit(31) = 1 Writing one to this bit wakes up the PCI9080 from reset.

Register	Name Description	Address (Local)	Value	Description
DMAMODE0	DMA channel 0 mode register	0x 100	0x000007C3	Bit(1:0) = 11 32-bits transfers Bit(5:2) = 0 Zero internal wait states) Bit(6) = 1 Enables READY input Bit(7) = 1 Enables BTERM input Bit(8) = 1 Enables local burst Bit(9) = 1 Enables DMA chaining Bit(10) = 1 Enables interrupt when done Bit(11) = 0 Inc. local address Bit(12) = 0 DMA demand mode disabled Bit(13) = 0 Write/invalidate disabled Bit(14) = 0 EOT disabled Bit(15) = 0 Send BLAST to terminate transfer Bit(16) = 0 Byte count is not cleared if descriptor is in Local memory when DMA completes transfer Bit(17) = 0 DMA interrupt routed to local bus

The TMS320C6000 DSP boot configuration is presented in Table 3.

Table 3. TMS320C6000 Boot Configuration via Pull-Up/Pull-Down Resistors on XD[31:0]

Field	Description
BLPOL	Determines polarity of /XBLAST signal BLPOL = 0, XBLAST is active low.
RWPOL	Determines polarity of expansion bus read/write signal RWPOL = 0, XW/R_
HMOD	Host mode (status in XB HPIC) HMOD = 1, external host interface is in synchronous master/slave mode.
XARB	Expansion bus arbiter (status in XBGC) XARB = 1, internal expansion bus arbiter is enabled.
FMOD	FIFO mode (status in XBGC)
LEND	Little-endian mode LEND = 1, system operates in little-endian mode
BootMode[4:0]	Dictates the boot mode of the device, including host port boot, ROM boot, memory map selection. (For a complete list of boot-modes, see the *TMS320C6000 Peripherals Reference Guide*.)

The Timers chapter in the *TMS320C6000 Peripheral Reference Guide* (SPRU190) describes how to configure the DSP's timer pins to operate as general-purpose user input outputs. (Tout pin is used to generate an interrupt to the PCI bridge.)

3 Master/Target Abort

If a PCI master abort or target abort is encountered during a transfer, the PCI9080 asserts the READYo, along with BTERMo for the first cycle only. Because data on the bus is irrelevant at the time, the READYo is enabled only when BTERMo is de-asserted (see the block diagram shown in Figure 1.

If a PCI master abort or target abort is encountered, the PCI9080 asserts $\overline{\text{LSERR}}$, if enabled. Because the $\overline{\text{LSERR}}$ is connected to the external interrupt of the DSP, the interrupt handler takes the appropriate application-specific action. It clears the abort bits in the PCI status configuration register of the PCI9080 (to clear the $\overline{\text{LSERR}}$ interrupt and re-enable direct master transfer). Prior to clearing the bits in the PCI configuration register of the PCI9080, the interrupt service routine must abort stalled transfer by writing STOP='00' into the XBHC register.

4 Deadlock

A deadlock situation can occur when a master on the PCI bus wants to access the PCI9080 local bus (expansion bus) at the same time the DSP (expansion bus master) wants to access the PCI bus.

To prevent deadlocks while the expansion bus is performing a master transfer, the expansion bus has the XBOFF signal. When asserted, XBOFF suspends the current access and causes the expansion bus to release ownership. When asserted the BREQo pin of the PCI9080 indicates a possible deadlock condition. The BREQo signal is connected to the XBOFF signal of the expansion bus.

The backoff is only recognized by the DSP during active master transfers when XRDY indicates a not-ready status and the PCI9080 is requesting the expansion bus (XHOLD= 1). The backoff request is not serviced until all current master transfers are completed internally. This allows read data to be flushed out of the pipeline. The XBOFF signal is not recognized during expansion-bus I/O-port transfers.

The DSP automatically tries to restart the transfer interrupted by a backoff from the point where the interruption took place. The transfer restart is completely transparent to the user.

5 Timing Verification

To verify proper operation, four functions have been examined:

- PCI9080 DMA channel write to the expansion bus
- PCI9080 DMA channel read from the expansion bus
- Expansion bus read from the PCI9080
- Expansion bus write to the PCI9080

In each instance, timing requirements were compared for each of the devices. The results are shown in the following tables and timing diagrams.

The interface was functionally verified using VHDL simulation (Synopsys PCI9080 Foundry Model and Synopsys PCI Target SourceModel were used in the test bench). The diagrams presented in Figure 2 through Figure 5 are outputs from the simulation. The clock ratio between the operating frequency of the TMS320C6000 DSP and the XCLKIN frequency is set to 5.

The PCI bus was parked on the PCI bridge because the bridge was the only device with mastering capability in the test bench.

Figure 2 presents a case where the DSP initiates a 24-word burst read from the PCI bus. The expansion bus, the PCI9080, and the PCI bus signals are shown in the figure. Note that because the PCI pre-fetch is enabled in the DMPBAM register, the PCI9080 pre-fetches 16 words from the PCI bus (the pre-fetch takes place roughly between 42.8 μs and 43.45 μs).

Figure 3 presents a case where the DSP initiates a 24-word burst write to the PCI bus. The expansion bus, the PCI9080, and the PCI bus signals are shown in the figure. The PCI bus signals are shown at the bottom of the figure.

Figure 4 presents a case where the PCI9080 DMA (direct memory access) moves data from the DSP to the PCI bus (in this case the PCI bridge is a master of the expansion bus). To move data, the PCI9080 is programmed to perform a two part chained DMA transfer (the descriptors for a chained DMA transfer must reside in the PCI bus address space). To perform data transfer, the PCI bridge must set the XBISA register of the DSP first. Therefore, the first part of the chained DMA transfer sets the XBISA register. After the XBISA is set, the descriptor for the second part of the chained DMA transfer is read from the PCI bus (activity on the PCI bus is between 45 μs and 46 μs). The PCI 9080 then performs data transfer from the expansion to the PCI bus.

Figure 5 presents a case where the PCI9080 DMA moves data from the PCI bus to the DSP (in this case the PCI bridge is a master of the expansion bus). To move data, the PCI9080 is programmed to perform a two-part chained DMA transfer (the descriptors for a chained DMA transfer must reside in the PCI bus address space). To perform data transfer, the PCI bridge must set the XBISA register of the DSP first. Therefore, the first part of the chained DMA transfer sets the XBISA register. After the XBISA is set, the descriptor for the second part of the chained DMA transfer is read from the PCI bus (activity on the PCI bus is between 52.25 μs and 53.25 μs). The PCI 9080 then performs the second part of chained DMA transfer. Data is first moved from the PCI bus into the internal FIFO of the PCI bridge. The PCI9080 then initiates a burst transfer on the expansion bus and moves the data to the DSP.

The simulations were performed using a TMS320C6202. However, the signals should be similar for all TMS320C6000 devices.

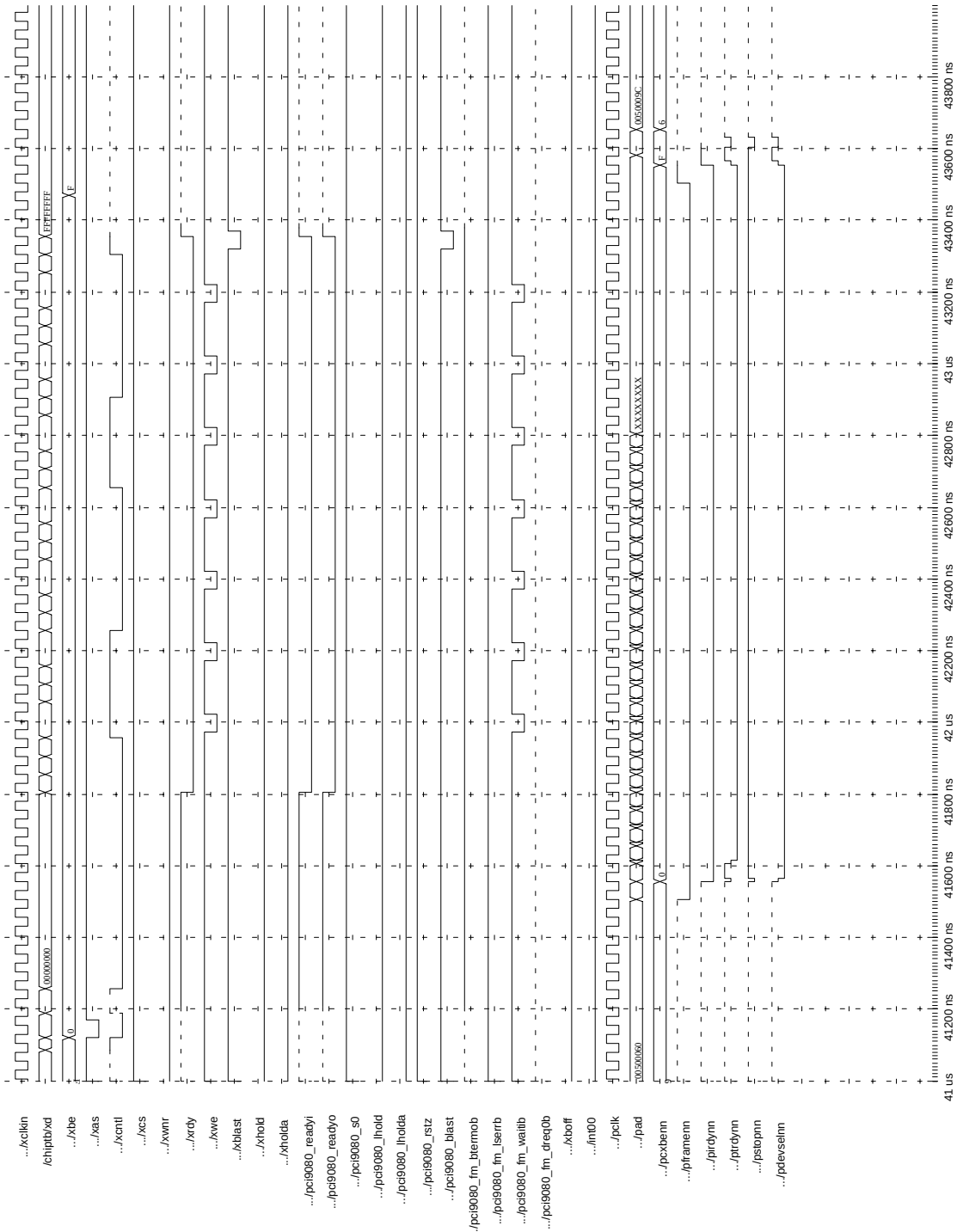

Figure 2. Expansion Bus Burst Read From the PCI Bus Using PCI9080

Figure 3. Expansion Bus Burst Write From the PCI Bus Using PCI9080

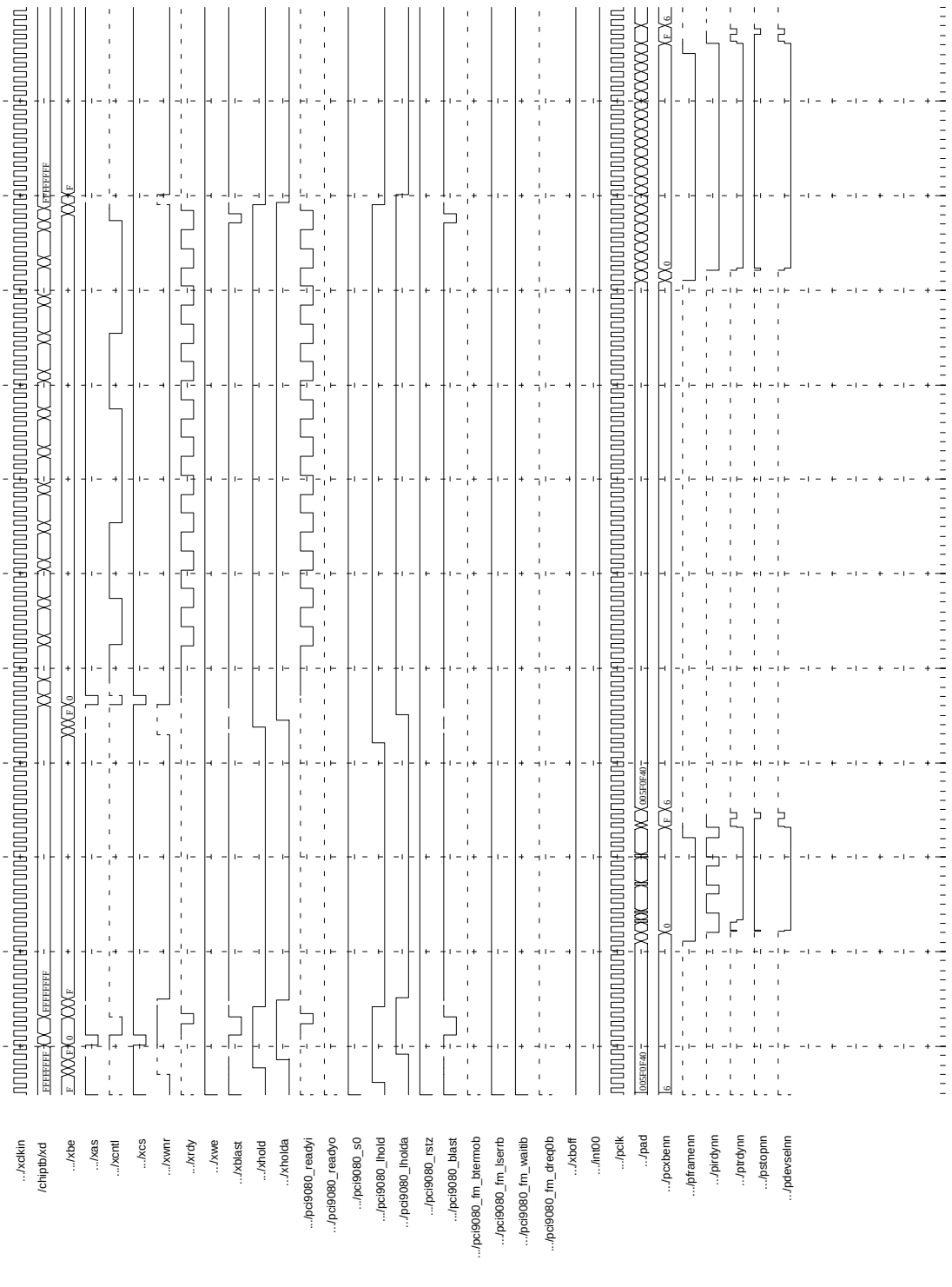

Figure 4. PCI9080 DMA Channel Moves a Burst of Data From the Expansion Bus to the PCI Bus

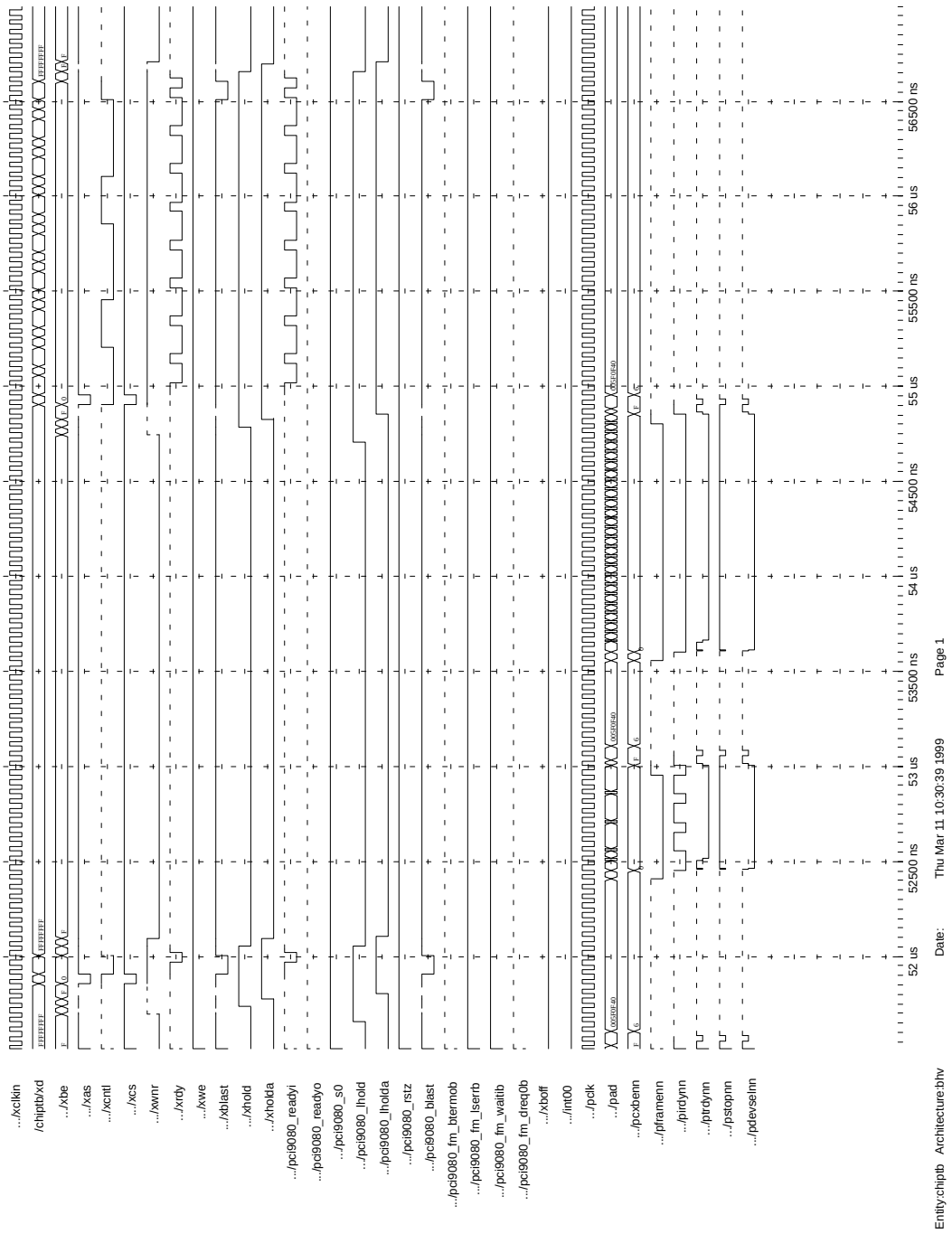

Figure 5. PCI9080 DMA Channel Moves a Burst of Data From the PCI Bus to the Expansion Bus

Table 4 through Table 7 summarize the timing requirements when the local bus is running at the PCI bus clock speed, T_{clk} = 30ns. The timing requirements presented in Table 4 and Table 5 apply when the C6000™ DSP is the local bus master. The timing requirements presented in Table 6 and Table 7 apply when the PCI9080 is the local bus master.

Table 4 through Table 7 show that the timing parameters for both devices are met in the interface of PCI9080 and TMS320C6000. This interface is based on a PCI9080 (local bus is running at 33 MHz) and a TMS320C6000 device at any frequency ranging from 100 MHz-250 MHz. The maximum local bus clock speed that can be achieved in the interface is 34 MHz. The limiting factor for the local bus speed in this design is the set-up time for the LWAITi signal of the PCI9080.

Table 4. Timing Requirements for the PCO9080 (TMS320C6000 Expansion Bus Master)

PCI9080 Symbol	C6000 Symbol	Parameter	PCI 9080 Min (ns)	C6000* Min (ns)
Ts(ADS)	Tcyc-Td(XCKIH-XDV)	Address strobe (ADS) valid before XCLKIN high	6	13.5
Th(ADS)	Td(XCKIH-XDIV)	Address strobe (ADS) valid after XCLKIN high.	1	5
Ts(BLAST)	Tcyc-Td(XCKIH-XBLTV)	Burst last (BLAST) valid before XCLKIN high	6	13.5
Th(BLAST)	Td(XCKIH-XBLTV)	Burst last (BLAST) valid after XCLKIN high.	0	5
Ts(LAD[31:2])	Tcyc-Td(XCKIH-XDV)	Data (LAD[31:2]) valid before XCLKIN high (WRITE)	5	13.5
Th(LAD[31:2])	Td(XCKIH-XDIV)	Data (LAD[31:2]) invalid after XCLKIN high (WRITE).	0	5
Ts(LBE[3:0])	Tcyc-Td(XCKIH-XBEV)	Byte enable (LBE[3:0]) valid before XCLKIN high	7	13.5
Th(LBE[3:0])	Td(XCKIH-XBEV)	Byte enable (LBE[3:0]) valid after XCLKIN high.	0	5
Ts(WAITi)	Tcyc-Td(XCKIH-XWTV)	Read/write (LW/R) valid before XCLKIN high	13	13.5
Th(WAITi)	Td(XCKIH-XWTV)	Read/write (LW/R) valid before XCLKIN high	0	5
Ts(LW/R)	Tcyc-Td(XCKIH-XWRV)	Read/write (LW/R) valid before XCLKIN high	9	13.5
Th(LW/R)	Td(XCKIH-XWRV)	Read/write (LW/R) valid after XCLKIN high.	0	5

*C6000 refers to the C6202, C6202B, C6203 and C6204 devices

Table 5. Timing Requirements for TMS320C6000 (TMS320C6000 Expansion Bus Master)

PCI9080 Symbol	C6000 Symbol	Parameter	PCI 9080 Min (ns)	C6000* Min (ns)
Tcyc-Td(READYo)-tPLD	Tsu(XRY-XCKIH)	Ready signal (READYo) valid before XCLKIN high	11	3.5
Td(READYo)+tPLD	Th(XCKIH-XRY)	Ready signal (READYo) valid after XCLKIN high	10	2.8
Tcyc-Td(LAD[31:2])	Tsu(XDV-XCKIH)	Data (XD) valid before XCLKN high (READ)	10	3.5
Td(LAD[31:2])	Th(XCKIH-XDV)	Data (XD) invalid after XCLKIN high (READ)	5	2.8
Tcyc-Td(BREQo)	Tsu(XBFF-XCKIH)	Back-off signal (XBOFF) valid before XCLKIN high	17	3.5
Td(BREQo)	Th(XCKIH-XBFF)	Back-off signal (XBOFF) valid after XCLKIN high	5	2.8

*C6000 refers to the C6202, C6202B, C6203 and C6204 devices

Table 6. Timing Requirements for TMS320C6000 (PCI9080 Expansion Bus Master)

PCI9080 Symbol	C6000 Symbol	Parameter	PCI 9080 Min (ns)	C6000* Min (ns)
Tcyc-Td(LAD[31:2])-tPAL	Tsu(XCSV-XCKIH)	Chip Select (XCS) valid before XCLKIN high	5	3.5
Td(LAD[31:2])	Th(XCKIH -XCS)	Chip Select (XCS) valid after XCLKIN high	10	2.8
Tcyc-Td(ADS)	Tsu(XAS-XCKIH)	Address strobe (XAS) valid before XCLKIN high	15.5	3.5
Td(ADS)	Th(XCKIH-XAS)	Address strobe (XAS) valid after XCLKIN high	14.5	2.8
Tcyc-Td(BLAST)	Tsu(XBLTV-XCKIH)	Burst last (XBLAST) valid before XCLKIN high	14	3.5
Td(BLAST)	Th(XCKIH-XBLTV)	Burst last (XBLAST) valid after XCLKIN high	16	2.8
Tcyc-Td(LAD[31:2])	Tsu(XD-XCKIH)	Data (XD) valid before XCLKIN high (WRITE)	10	3.5
Td(LAD[31:2])	Th(XCKIH-XD)	Data (XD) valid after XCLKIN high (WRITE)	5	2.8
Tcyc-Td(LBE[3:1])	Tsu(XBEV-XCKIH)	Byte enable (XBE[3:0]) valid before XCLKIN high	14	3.5
Td(LBE[3:1])	Th(XCKIH-XBEV)	Byte enable (XBE[3:0]) valid after XCLKIN high	16	2.8
Tcyc-Td(LW/R)	Tsu(XWR-XCKIH)	Read/write (XR/W) valid before XCLKIN high	16	3.5
Td(LW/R)	Th(XCKIH-XWR)	Read/write (XR/W) valid after XCLKIN high	14	2.8

*C6000 refers to the C6202, C6202B, C6203 and C6204 devices

Table 7. Timing Requirements for PCI9080 (PCI9080 Expansion Bus Master)

PCI9080 Symbol	C6000 Symbol	Parameter	PCI 9080 Min (ns)	C6000* Min (ns)
Ts(READYi)	Tcyc-Td(XCKIH-XRDY)	Ready signal (READYo) valid before XCLKIN high	8	13.5
Th(READYi)	Td(XCKIH-XRDY)	Ready signal (READYo) valid after XCLKIN high	1	5
Ts(LAD[31:2])	Tcyc-Td(XCKIH-XDV)	Data (XD) valid before XCLKN high (READ)	5	13.5
Th(LAD[31:2])	Td(XCKIH-XDIV)	Data (XD) invalid after XCLKIN high (READ)	0	5

*C6000 refers to the C6202, C6202B, C6203 and C6204 devices

6 References

1. *TMS320C6000 Peripherals Reference Guide* (SPRU190).
2. *TMS320C6202, TMS320C6202B Fixed-Point Digital Signal Processors* (SPRS104).
3. *TMS320C6203, TMS320C6203C Fixed-Point Digital Signal Processors* (SPRS086).
4. *TMS320C6204 Fixed-Point Digital Signal Processor* (SPRS152).
5. PCI9080 Data Book, PLX Technology, Inc. http://www.plxtech.com.

Appendix A PCI9080 Timing Requirements

Table A–1. PCI Timing Parameters (Local Inputs)

Characteristic	Setup Min (ns)	Hold Max (ns)
READYi	8	1
ADS	6	1
LBE[3:0]	7	0
BIGEND	4	0
BLAST	6	0
BREQi	7	0
BTERM	7	1
DP[3:0]	4	0
DREQ[1:0]	3	1
EOT0	7	1
EOT1	1	1
LA[31:0]	5	0
LLOCK	4	0
LAD	5	0
LD[31:0]	5	0
LHOLDA	7	2
LINTi	7	0
LW/R	9	0
WAITi	13	0
USERi	4	0
S[2:0]	1	2

Table A–2. PCI Timing Parameters (Local Outputs)

Characteristic	Output valid	
	Min (ns)	Max (ns)
ADS	5	14.5
WAITo	5	18
USERO	5	11
READYo	5	14
PCHK	5	12
LW/R	5	14
LSERR	5	12
LLOCKo	5	12
LINTo	5	13
LHOLD	5	13
LDSHOLD	5	12
LD[31:0]	5	20
LBE[3:0]	5	16
LABS[3:1]	5	12
LA[31:2]	5	20
DT/R	5	14
DP[3:0]	5	20
DMPAF	5	17
DEN	5	13
BLAST	5	16
DACK[1:0]	5	14
BTERMo	5	15
BREQo	5	13

The timing requirements in Table A–1 and Table A–2 are provided for quick reference only. For detailed description, notes, and restrictions, please see the *PCI9080 Data Book*.

Appendix B TMS320C6000 Timing Parameters

Table B–1. TMS320C6202, C6202B, C6203, C6204 Timing Parameters (External Device is a Master)

Characteristic	Symbol	Min (ns)	Max (ns)
Setup time, XCS valid before XCLKIN high	Tsu(XCSV-XCKIH)	3.5	
Hold time, XCS valid after XCLKIN high	Th(XCKIH-XCSV)	2.8	
Setup time, XAS valid before XCLKIN high	Tsu(XASV-XCKIH)	3.5	
Hold time, XAS valid after XCLKIN high	Th(XCKIH-XASV)	2.8	
Setup time, XCNTL valid before XCLKIN high	Tsu(XCTL-XCKIH)	3.5	
Hold time, XCNTL valid after XCLKIN high	Th(XCKIH-XCTL)	2.8	
Setup time, XWR valid before XCLKIN high	Tsu(XWR-XCKIH)	3.5	
Hold time, XWR valid after XCLKIN high	Th(XCKIH-XWR)	2.8	
Setup time, XBLAST valid before XCLKIN high	Tsu(XBLTV-XCKIH)	3.5	
Hold time, XBLAST valid after XCLKIN high	Th(XCKIH-XBLTV)	2.8	
Setup time, XBE valid before XCLKIN high	Tsu(XBEV-XCKIH)	3.5	
Hold time, XBE valid after XCLKIN high	Th(XCKIH-XBEV)	2.8	
Setup time, XD valid before XCLKIN high	Tsu(XD-XCKIH)	3.5	
Hold time, XD valid after XCLKIN high	Th(XCKIH-XD)	2.8	
Delay time, XCLKIN high to XD low impedance	Td(XCKIH-XDLZ)	0	
Delay time, XCLKIN high to XD valid	Td(XCKIH-XDV)		16.5
Delay time, XCLKIN high to XD invalid	Td(XCKIH-XDIV)	5	
Delay time, XCLKIN high XD high impedance	Td(XCKIH-XDHZ)		4P*
Delay time, XCLKIN high to XRDY valid	Td(XCKIH-XRY)	5	16.5

* P = 1/CPU clock frequency in ns.

Table B–2. TMS320C6202, C6202B, C6203, C6204 Timing Parameters (TMS320C620x is a Master)

Characteristic	Symbol	Min (ns)	Max (ns)
Setup time, XD valid before XCLKIN high	Tsu(XDV-XCKIH)	3.5	
Hold time, XD valid after XCLKIN high	Th(XCKIH-XDV)	2.8	
Setup time, XRDY valid before XCLKIN high	Tsu(XRY-XCKIH)	3.5	
Hold time, XRDY valid after XCLKIN high	Th(XCKIH-XRY)	2.8	
Setup time, XBOFF valid before XCLKIN high	Tsu(XBFF-XCKIH)	3.5	
Hold time, XBOFF valid after XCLKIN high	Th(XCKIH-XBFF)	2.8	
Delay time, XCLKIN high to XAS valid	Td(XCKIH-XAS)	5	16.5
Delay time, XCLKIN high to XWR valid	Td(XCKIH-XWR)	5	16.5
Delay time, XCLKIN high to XBLAST valid	Td(XCKIH-XBLTV)	5	16.5
Delay time, XCLKIN high to XBE valid	Td(XCKIH-XBEV)	5	16.5
Delay time, XCLKIN high to XD low impedance	Td(XCKIH-XDLZ)	0	
Delay time, XCLKIN high to XD valid	Td(XCKIH-XDV)		16.5
Delay time, XCLKIN high to XD invalid	Td(XCKIH-XDIV)	5	
Delay time, XCLKIN high to XD high impedance	Td(XCKIH-XDHZ)		4P
Delay time, XCLKIN high to XWE/XWAIT valid	Td(XCKIH-XWTV)	5	16.5

* P = 1/CPU clock frequency in ns.

The timing requirements in Table B–1 through Table B–2 are provided for quick reference only. For detailed description, notes, and restrictions, please see the corresponding *Fixed-Point Digital Signal Processor* data sheet.

IMPORTANT NOTICE

Texas Instruments and its subsidiaries (TI) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement, and limitation of liability.

TI warrants performance of its products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Customers are responsible for their applications using TI components.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance or customer product design. TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such products or services might be or are used. TI's publication of information regarding any third party's products or services does not constitute TI's approval, license, warranty or endorsement thereof.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations and notices. Representation or reproduction of this information with alteration voids all warranties provided for an associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Resale of TI's products or services with *statements different from or beyond the parameters* stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service, is an unfair and deceptive business practice, and TI is not responsible nor liable for any such use.

Also see: Standard Terms and Conditions of Sale for Semiconductor Products. www.ti.com/sc/docs/stdterms.htm

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265