Application Report
SPRA965 − October 2003

TMS320C64x DSP Peripheral Component Interconnect (PCI) Performance

Stéphane Smith *C6x Device Applications*

ABSTRACT

This application report describes the number of cycles required to perform a given peripheral component interconnect (PCI) data transfer based on a variety of permutations of burst length, CPU speed, EMIF speed, etc.

The PCI bus, created by Intel in 1992, enables fast accesses between PCI adapters, system memory and external memory. To insure throughput near or at the processor's native bus speed, data transactions are performed as burst transfers. In addition, the PCI architecture implements many features to provide simultaneous connectivity between multiple devices. Due to the nature of burst transfers and PCI bus arbitration, variations in hardware settings can drastically affect the throughput across the PCI bus.

This document provides data sheets of possible TMS320C64xx hardware configurations, and their effects on PCI throughput performance. More specifically, transfer latency, the number of PCI cycles required to transfer *n* words of data, overall throughput given *n* word bursts, and turn-around penalty will be examined.

Contents

List of Figures

List of Tables

1 Design Problem

How do various hardware permutations affect the peripheral component interconnect (PCI) throughput on TMS320C64x digital signal processors?

2 Solution

PCI devices access each other, system memory, and external memory through burst transfers. A burst transfer is characterized by having an initialization or address phase followed by two or more data phases. During the address phase, the master passes a starting address and a transaction type. Various transaction types include I/O read, I/O write, memory read, memory write, configuration read, configuration write, etc. The following phases are data phases until the master signals the last data element. Various configurations influence the performance of PCI burst transfers. The configurations under examination include:

- CPU speed
- PCI speed
- Transfer Source/Destination
- EMIF speed
- EMIF width
- Burst Length

These hardware variations affect transfer latency, the number of PCI cycles required to transfer *n* words of data, overall throughput given *n* word bursts, and turn-around penalty.

3 Measurement Assumptions

The PCI performance measurements were taken with the following assumptions:

- There is no CPU, EMIF, EDMA or PCI activity other than what is required to perform simple PCI transfers. All measurements were taken with ideal system traffic; actual throughput for specific applications will vary.
- The DSP is functioning as a target device. (When in master mode, the target should always be ready without latency.)
- PCI latency timer is set to its default value of 0x0.
- PCI Min_Gnt (Minimum Grant) field is set to its default value of 0x0.
- PCI Max_Lat (Maximum Latency) field is set to the default value of 0x0.
- Unless otherwise stated, EMIF is connected to SDRAM.

4 Master/Target Latency

Initiator and target latency is the amount of time from when the master starts the transaction to when the target is ready to transfer the first data item. In order to prevent devices from monopolizing the PCI bus, the PCI bus specification implements the first data phase rule. According to specification, the target is limited to 16 PCI clock cycles to complete the first data transfer. Similarly, the master is limited to a maximum of 8 PCI clock cycles. If for any reason, the device cannot meet these requirements, the target must issue a retry. The retry, indicated by deasserting *TRDY* and *DEVSEL,* while asserting *STOP*, terminates the transaction prematurely, thereby freeing the PCI bus for use by other devices. After a minimum of two clock cycles, the initiator may reattempt to transfer data.

In general, master/target latency is a function of:

- How fast the master can transfer data
- Access time for the target device

In these tests, the master is requesting a read, therefore the target device must prefetch data. During this prefetching stage, the target will continually issue a retry until it's ready to stream data.

Figure 1 displays the latency where:

- *add.* represents the target address.
- *cmd* is the command used to determine the type of transaction to be performed.
- *t/a* represents the turn-around cycle required by reads to hand off control of the AD bus to the target.
- *d#* represent the data items being transferred.

The latency measurement begins at the start of the transaction. This is followed by a series of retries, until data is finally ready to be transferred.

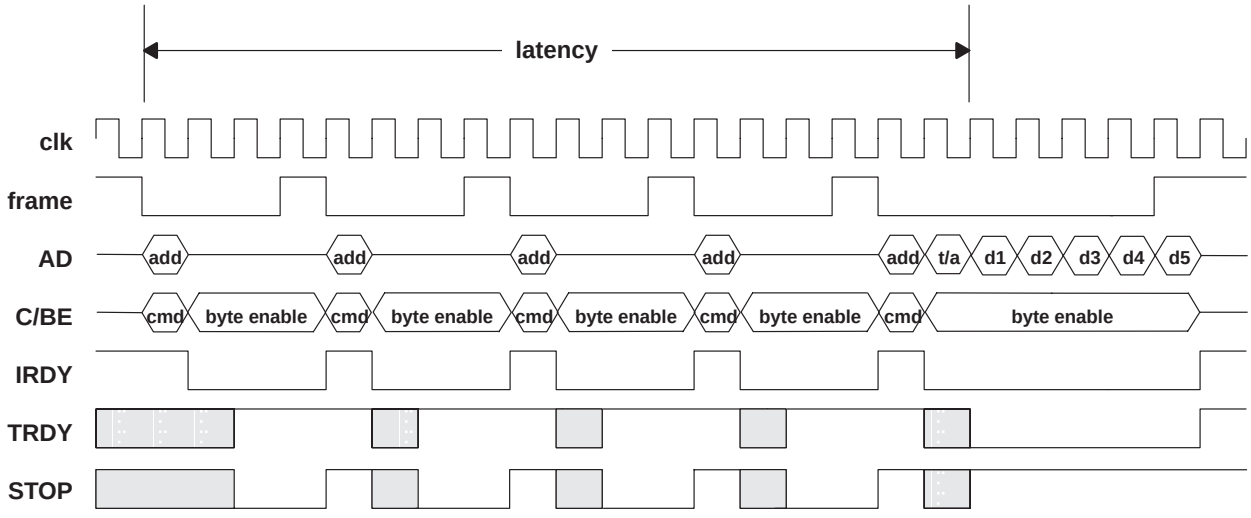

Figure 1. Read Latency Timing Diagram

Table 1. Read Latency (Measured in PCI Clock Cycles)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Latency
500	L2 256k	33	8			18
			16			18
			64			18
			1024			18
		66	8			26
			16			26
			64			26
			1024			26
	EMIFA (SDRAM)	33	8	133	64	22
					32	22
				100	64	22
					32	22
			16	133	64	22
					32	22
				100	64	22
					32	22
			64	133	64	22
					32	22
				100	64	22
					32	22
			1024	133	64	22
					32	22
				100	64	22
					32	22

Table 1. Read Latency (Measured in PCI Clock Cycles) (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Latency
500	EMIFA (SDRAM)	66	8	133	64	30
					32	30
				100	64	30
					32	34
			16	133	64	30
					32	30
				100	64	30
					32	34
			64	133	64	30
					32	30
				100	64	30
					32	34
			1024	133	64	30
					32	30
				100	64	30
					32	34
600	L2 256k	33	8			18
			16			18
			64			18
			1024			18
		66	8			22
			16			22
			64			22
			1024			22

Table 1. Read Latency (Measured in PCI Clock Cycles) (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Latency
600	EMIFA (SDRAM)	33	8	133	64	18
					32	22
				100	64	22
					32	22
			16	133	64	18
					32	22
				100	64	22
					32	22
			64	133	64	18
					32	22
				100	64	22
					32	22
			1024	133	64	18
					32	22
				100	64	22
					32	22
		66	8	133	64	26
					32	30
				100	64	30
					32	34
			16	133	64	26
					32	30
				100	64	30
					32	34
			64	133	64	26
					32	30
				100	64	30
					32	34
			1024	133	64	26
					32	30
				100	64	30
					32	34

Table 1. Read Latency (Measured in PCI Clock Cycles) (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Latency
720	L2 256k	33	8			14
			16			14
			64			14
			1024			14
		66	8			22
			16			22
			64			22
			1024			22
	EMIFA (SDRAM)	33	8	133	64	18
					32	22
			16	133	64	18
					32	22
			64	133	64	18
					32	22
			1024	133	64	18
					32	22
		66	8	133	64	26
					32	30
			16	133	64	26
					32	30
			64	133	64	26
					32	30
			1024	133	64	26
					32	30

Figure 2 displays the latency where:

- *add.* represents the target address.
- *cmd* is the command used to determine the type of transaction to be performed.
- *d#* represent the data items being transferred.

For write transfers, data is ready immediately; therefore, the latency is always one PCI cycle.

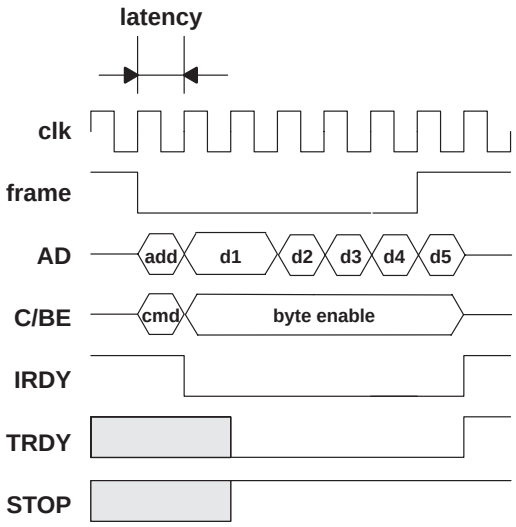

Figure 2. Write Latency Timing Diagram

5 Number of PCI Cycles Required to Transfer *n* Words of Data

Before a device transfers data through the PCI bus, it must prefetch data into either its read-ahead or write buffers. The device may then connect to and burst data across the PCI bus. As the data in the buffer depletes, the device must simultaneously prefetch more data while continually transferring information. If the time required to prefetch data exceeds the time to transfer data, the buffer will eventually completely empty and the device will be forced to disconnect from its target. Once more data are available in the buffer, the initiator may reattempt to complete the transaction.

Figure 3 displays the number of cycles to transfer five words where:

- *add.* represents the target address.
- *cmd* is the command used to determine the type of transaction to be performed.
- *t/a* represents the turn around cycle required by reads to hand off control of the AD bus to the target.
- *d#* represent the data items being transferred.

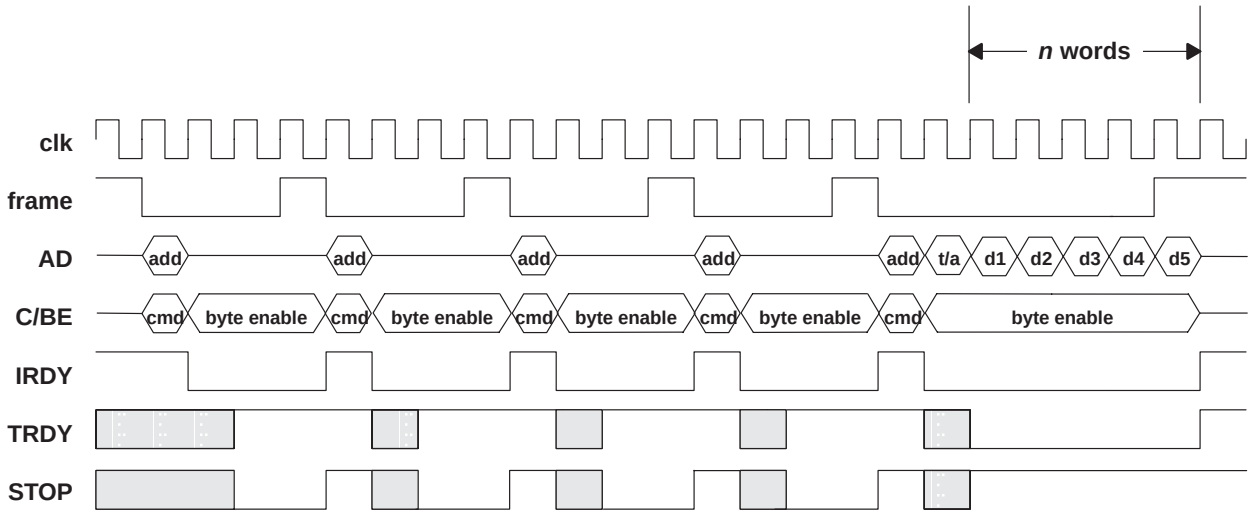

Figure 3. Number of PCI Cycles to Read 5 Words Timing Diagram

Table 2. Number of PCI Cycles to Read *n* Words of Data

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
500	L2 256k	33	8			8
			16			16
			64			64
			1024			1024
		66	8			8
			16			16
			64			64
			1024			1479

Table 2. Number of PCI Cycles to Read *n* Words of Data (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
500	EMIFA (SDRAM)	33	8	133	64	8
					32	8
				100	64	8
					32	8
			16	133	64	16
					32	16
				100	64	16
					32	16
			64	133	64	64
					32	64
				100	64	64
					32	64
			1024	133	64	1024
					32	1024
				100	64	1024
					32	1210
		66	8	133	64	8
					32	8
				100	64	8
					32	8
			16	133	64	16
					32	16
				100	64	16
					32	16
			64	133	64	154
					32	221
				100	64	213
					32	245
			1024	133	64	2809
					32	3660
				100	64	3268
					32	4584

Table 2. Number of PCI Cycles to Read *n* Words of Data (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
600	L2 256k	33	8			8
			16			16
			64			64
			1024			1024
		66	8			8
			16			16
			64			64
			1024			1024
	EMIFA (SDRAM)	33	8	133	64	8
					32	8
				100	64	8
					32	8
			16	133	64	16
					32	16
				100	64	16
					32	16
			64	133	64	64
					32	64
				100	64	64
					32	64
			1024	133	64	1024
					32	1024
				100	64	1024
					32	1303

Table 2. Number of PCI Cycles to Read *n* Words of Data (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
600	EMIFA (SDRAM)	66	8	133	64	8
					32	8
				100	64	8
					32	8
			16	133	64	16
					32	16
				100	64	16
					32	16
			64	133	64	158
					32	146
				100	64	150
					32	253
			1024	133	64	2730
					32	3049
				100	64	2886
					32	4809
720	L2 256k	33	8			8
			16			16
			64			64
			1024			1024
		66	8			8
			16			16
			64			64
			1024			1024

Table 2. Number of PCI Cycles to Read *n* Words of Data (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
720	EMIFA (SDRAM)	33	8	133	64	8
					32	8
			16	133	64	16
					32	16
			64	133	64	64
					32	64
			1024	133	64	1024
					32	1024
		66	8	133	64	8
					32	8
			16	133	64	16
					32	16
			64	133	64	154
					32	142
			1024	133	64	2534
					32	3245

Figure 4 displays the number of cycles to transfer five words where:

- *add.* represents the target address.
- *cmd* is the command used to determine the type of transaction to be performed.
- *d#* represent the data items being transferred.

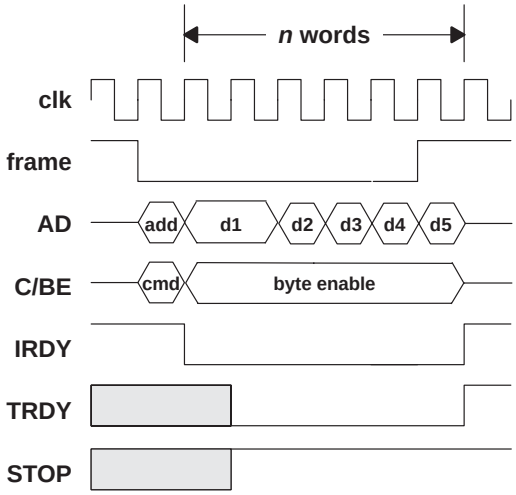

Figure 4. Number of PCI cycles to write 5 words Timing Diagram

Table 3. Number of PCI Cycles to Write *n* Words of Data

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
500	L2 256k	33	8			9
			1024			1025
		66	8			9
			1024			1265
	EMIFA (SDRAM)	33	8	133	64	9
					32	9
				100	64	9
					32	9
			1024	133	64	1025
					32	1025
				100	64	1025
					32	1025
		66	8	133	64	9
					32	9
				100	64	9
					32	9
			1024	133	64	1417
					32	1417
				100	64	1421
					32	1421
600	L2 256k	33	8			9
			1024			1025
		66	8			9
			1024			1025
	EMIFA (SDRAM)	33	8	133	64	9
					32	9
				100	64	9
					32	9
			1024	133	64	1025
					32	1025
				100	64	1025
					32	1025

Table 3. Number of PCI Cycles to Write *n* Words of Data (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
600	EMIFA (SDRAM)	66	8	133	64	9
					32	9
				100	64	9
					32	9
			1024	133	64	1245
					32	1245
				100	64	1245
					32	1237
720	L2 256k	33	8			9
			1024			1025
		66	8			9
			1024			1025
	EMIFA (SDRAM)	33	8	133	64	9
					32	9
			1024	133	64	1025
					32	1025
		66	8	133	64	9
					32	9
			1024	133	64	1025
					32	1025

6 Total Throughput

The total throughput is defined as the amount of data transferred per unit time. Total PCI throughput is measured from the start of the transaction until the last data item has been transferred. The equation for calculating total throughput is:

$$TotalThroughput = \frac{(\#words)(4)}{(plck)(latency + xfer)} \text{ [bytes/s]}$$

Where:

#words is the number of words of data transferred.

pclk is the PCI clock period (typically 30ns for a 33MHz clock or 15ns for a 66MHz clock).

latency is the number of cycles between when the master starts the transaction to when the target is ready to transfer the first data item.

xfer is the number of cycles required to transfer n words of data.

Figure 5 displays the total-throughput of a PCI transfer where:

- *add.* represents the target address.
- *cmd* is the command used to determine the type of transaction to be performed.
- *t/a* represents the turnaround cycle required by reads to hand off control of the AD bus to the target.
- *d#* represent the data items being transferred.

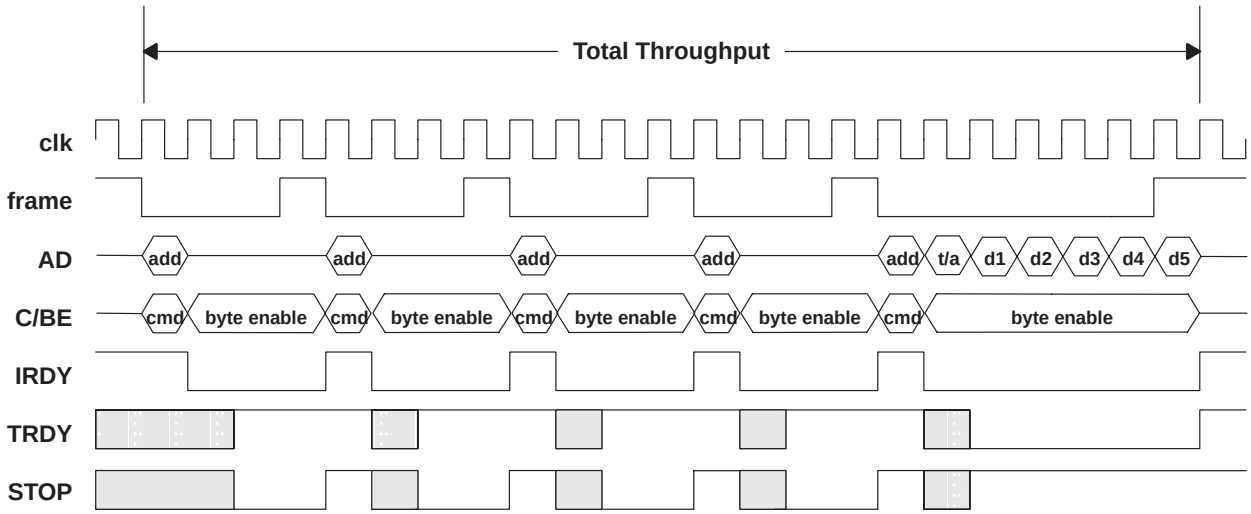

Figure 5. Read Throughput Timing Diagram

Table 4. Total Throughput for Reads (Measured in MB/s)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
500	L2 256k	33	8			41
			16			62.7
			64			104.1
			1024			131
		66	8			62.7
			16			101.6
			64			189.6
			1024			181.4

Table 4. Total Throughput for Reads (Measured in MB/s) (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
500	EMIFA (SDRAM)	33	8	133	64	35.6
					32	35.6
				100	64	35.6
					32	35.6
			16	133	64	56.1
					32	56.1
				100	64	56.1
					32	56.1
			64	133	64	99.2
					32	99.2
				100	64	99.2
					32	99.2
			1024	133	64	130.5
					32	130.5
				100	64	130.5
					32	110.8
		66	8	133	64	56.1
					32	56.1
				100	64	56.1
					32	50.8
			16	133	64	92.8
					32	92.8
				100	64	92.8
					32	85.3
			64	133	64	92.8
					32	68
				100	64	70.2
					32	61.2
			1024	133	64	96.2
					32	74.0
				100	64	82.8
					32	59.1

Table 4. Total Throughput for Reads (Measured in MB/s) (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
600	L2 256k	33	8			41
			16			62.7
			64			104.1
			1024			131
		66	8			71.1
			16			112.3
			64			198.4
			1024			261.1
	EMIFA (SDRAM)	33	8	133	64	41
					32	35.6
				100	64	35.6
					32	35.6
			16	133	64	62.7
					32	56.1
				100	64	56.1
					32	56.1
			64	133	64	104.1
					32	99.2
				100	64	99.2
					32	99.2
			1024	133	64	131
					32	130.5
				100	64	130.5
					32	103

Table 4. Total Throughput for Reads (Measured in MB/s) (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
600	EMIFA (SDRAM)	66	8	133	64	62.7
					32	56.1
				100	64	56.1
					32	50.8
			16	133	64	101.6
					32	92.8
				100	64	92.8
					32	85.3
			64	133	64	92.8
					32	97
				100	64	94.8
					32	59.5
			1024	133	64	99.1
					32	88.7
				100	64	93.6
					32	56.4
720	L2 256k	33	8			48.5
			16			71.1
			64			109.4
			1024			131.5
		66	8			71.1
			16			112.3
			64			198.4
			1024			261.1

Table 4. Total Throughput for Reads (Measured in MB/s) (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
720	EMIFA (SDRAM)	33	8	133	64	41
					32	35.6
			16	133	64	62.7
					32	56.1
			64	133	64	104.1
					32	99.2
			1024	133	64	131
					32	130.5
		66	8	133	64	62.7
					32	56.1
			16	133	64	101.6
					32	92.8
			64	133	64	94.8
					32	99.2
			1024	133	64	106.7
					32	83.4

Figure 6 displays the total-throughput of a PCI transfer where:

- *add.* represents the target address.
- *cmd* is the command used to determine the type of transaction to be performed.
- *d#* represent the data items being transferred.

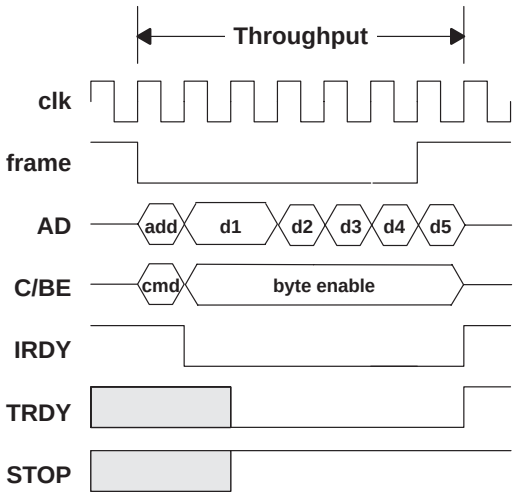

Figure 6. Write Throughput Timing Diagram

Table 5. Total Throughput for Writes (Measured in MB/s)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
500	L2 256k	33	8			106.7
			1024			133.1
		66	8			213.3
			1024			215.7
	EMIFA (SDRAM)	33	8	133	64	106.7
					32	106.7
				100	64	106.7
					32	106.7
			1024	133	64	133.1
					32	133.1
				100	64	133.1
					32	133.1
		66	8	133	64	213.3
					32	213.3
				100	64	213.3
					32	213.3
			1024	133	64	192.6
					32	192.6
				100	64	192
					32	192

Table 5. Total Throughput for Writes (Measured in MB/s) (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
600	L2 256k	33	8			106.7
			1024			133.1
		66	8			213.3
			1024			266.1
	EMIFA (SDRAM)	33	8	133	64	106.7
					32	106.7
				100	64	106.7
					32	106.7
			1024	133	64	133.1
					32	133.1
				100	64	133.1
					32	133.1
		66	8	133	64	213.3
					32	213.3
				100	64	213.3
					32	213.3
			1024	133	64	219.2
					32	219.2
				100	64	219.2
					32	220.6
720	L2 256k	33	8			106.7
			1024			133.1
		66	8			213.3
			1024			266.1
	EMIFA (SDRAM)	33	8	133	64	106.7
					32	106.7
			1024	133	64	133.1
					32	133.1
		66	8	133	64	213.3
					32	213.3
			1024	133	64	266.1
					32	266.1

7 Turn-Around Penalty

The turnaround penalty is the result of the overhead associated with back-to-back transfers. During a read, the target device prefetches data and issues retries until it is ready to stream data to its master device. Similarly, for writes, the target device prefetches data into a write buffer before transferring the first data item. When a device transitions from one transaction to another, the read-ahead or write buffers might be either full or partially full, and the device must therefore empty them and reprefetch more data before performing its next burst transfer. The additional time required for flushing FIFOs is the turnaround penalty. Different back-to-back configurations include read/write, read/read, write/read and write/write.

Figure 7 displays a back-to-back read/read transaction where:

- *ad.* is the target address.
- *cm* is the command determining the type of transaction to be performed.
- *t/a* represents the turnaround cycle required by reads to hand off control of the AD bus to the target.
- *d#* represents the data items being transferred.

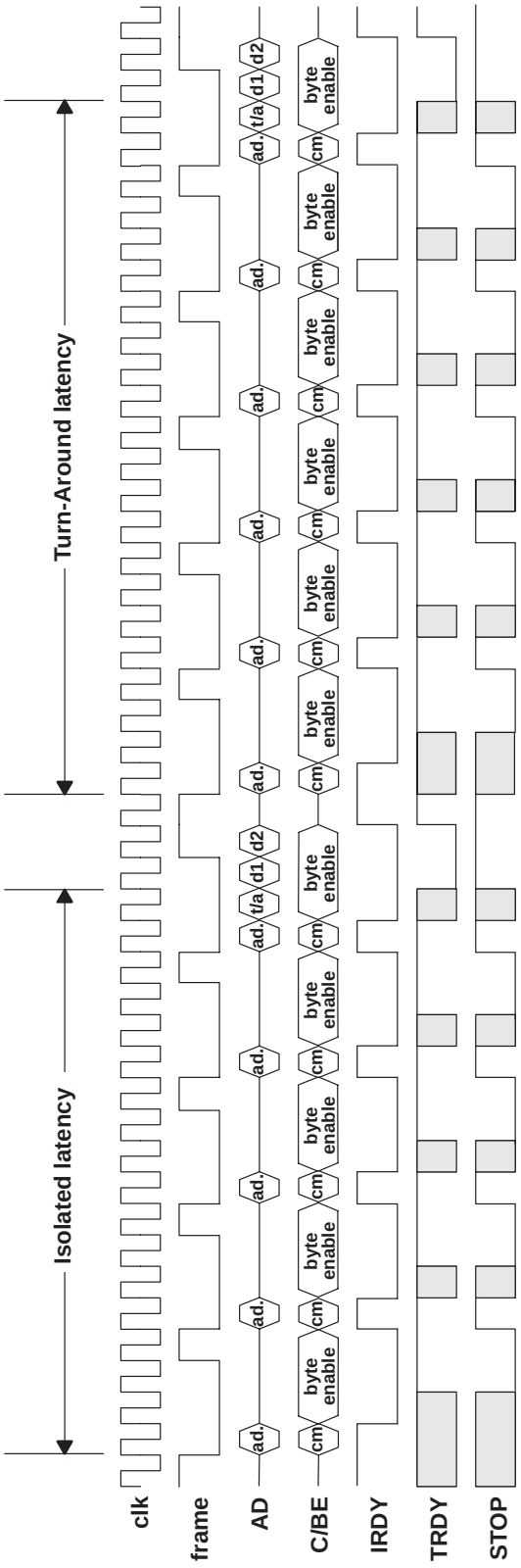

Figure 7. Read/Read Turn-Around Penalty

The equation to calculate the turn-around penalty is as follows:

TurnAroundPenalty = TurnAroundLatency − IsolatedLatency

Table 6. Turn-Around Penalty (Measured in PCI Clock Cycles)

CPU	SRC/DST	PCI	Type	EMIF	EMIF Width	T/A Penalty
500	L2 256k	33	R/W			0
			R/R			4
			W/R			8
			W/W			8
		66	R/W			0
			R/R			4
			W/R			8
			W/W			8
	EMIFA (SDRAM)	33	R/W	133	64	0
					32	0
				100	64	0
					32	0
			R/R	133	64	0
					32	0
				100	64	0
					32	4
			W/R	133	64	8
					32	8
				100	64	8
					32	8
			W/W	133	64	8
					32	8
				100	64	8
					32	8

Table 6. Turn-Around Penalty (Measured in PCI Clock Cycles) (Continued)

CPU	SRC/DST	PCI	Type	EMIF	EMIF Width	T/A Penalty
500	EMIFA (SDRAM)	66	R/W	133	64	0
					32	0
				100	64	0
					32	0
			R/R	133	64	16
					32	20
				100	64	20
					32	20
			W/R	133	64	8
					32	12
				100	64	8
					32	8
			W/W	133	64	8
					32	8
				100	64	8
					32	8
600	L2 256k	33	R/W			0
			R/R			0
			W/R			8
			W/W			8
		66	R/W			0
			R/R			4
			W/R			8
			W/W			8

Table 6. Turn-Around Penalty (Measured in PCI Clock Cycles) (Continued)

CPU	SRC/DST	PCI	Type	EMIF	EMIF Width	T/A Penalty
600	EMIFA (SDRAM)	33	R/W	133	64	0
					32	0
				100	64	0
					32	0
			R/R	133	64	4
					32	0
				100	64	0
					32	4
			W/R	133	64	8
					32	8
				100	64	8
					32	8
			W/W	133	64	8
					32	8
				100	64	8
					32	8
		66	R/W	133	64	0
					32	0
				100	64	0
					32	0
			R/R	133	64	16
					32	16
				100	64	20
					32	24
			W/R	133	64	12
					32	8
				100	64	8
					32	8
			W/W	133	64	8
					32	8
				100	64	8
					32	8

Table 6. Turn-Around Penalty (Measured in PCI Clock Cycles) (Continued)

CPU	SRC/DST	PCI	Type	EMIF	EMIF Width	T/A Penalty
720	L2 256k	33	R/W			0
			R/R			4
			W/R			8
			W/W			8
		66	R/W			0
			R/R			0
			W/R			8
			W/W			8
	EMIFA (SDRAM)	33	R/W	133	64	0
					32	0
			R/R	133	64	4
					32	4
			W/R	133	64	8
					32	12
			W/W	133	64	8
					32	8
		66	R/W	133	64	0
					32	0
			R/R	133	64	12
					32	16
			W/R	133	64	8
					32	8
			W/W	133	64	8
					32	8

8 References

1. *TMS320C6000 Peripherals Reference Guide* (SPRU190)
2. Shanley, Tom and Don Anderson. *PCI System Architecture 4th ed.* Mindshare Inc., 1999.
3. *TMS320C6201/6701 DSP Host Port Interface (HPI) Performance* (SPRA449)

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

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

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

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. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI 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 and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265