

# 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

1 D	esign Problem	2
2 S	olution	2
3 M	leasurement Assumptions	2
4 M	laster/Target Latency	3
5 N	umber of PCI Cycles Required to Transfer n Words of Data	8
6 To	otal Throughput	15
7 Tu	urn-Around Penalty	23
8 R	eferences	28
	List of Figures	
Figure	1 Read Latency Timing Diagram	3
Figure	2 Write Latency Timing Diagram	8
Figure	Number of PCI Cycles to Read 5 Words Timing Diagram	9
Figure	,	
Figure		16
Figure	6 Write Throughput Timing Diagram	20
Figure	7 Read/Read Turn-Around Penalty	24

Trademarks are the property of their respective owners.



#### **List of Tables**

Table 1	Read Latency (Measured in PCI Clock Cycles)	4
Table 2	Number of PCI Cycles to Read n Words of Data	9
Table 3	Number of PCI Cycles to Write n Words of Data	14
Table 4	Total Throughput for Reads (Measured in MB/s)	16
Table 5	Total Throughput for Writes (Measured in MB/s)	21
Table 6	Turn-Around Penalty (Measured in PCI Clock Cycles)	25

### 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.

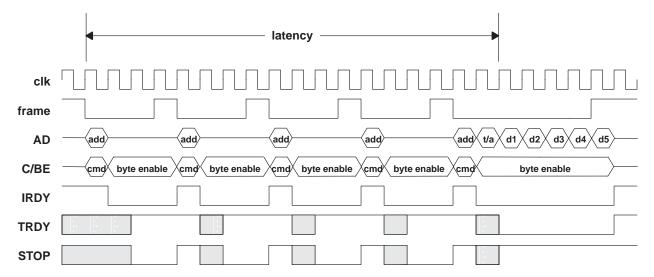


Figure 1. Read Latency Timing Diagram



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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Latency
			8			18
	10.050	22	16			18
		33	64			18
			1024			18
	L2 256K	33	26			
		66	16			26
		00	64			26
			1024			26
				122	64	22
				133	32	22
			0	100	64	22
500				100	32	22
500				122	64	22
			40	133	32	22
			16	100	64	22
	ENNIER (CDDANA)	22		100	32	22
	EMIFA (SDRAM)	33		422	64	22
			64	133	32	22
			04	100	64	22
				100	32	22
				122	64	22
			1024	133	32	22
			1024	100	64	22
				100	32	22



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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Latency
				400	64	30
			8	133	32	30
			8	100	64	30
				100	32	34
				422	64	30
			16	133	32	30
		16	30			
500	EMIEA (CDDAM)	66		100	32	34
500	EMIFA (SDRAM)	00		422	64	30
			64	133	32	30
			64	400	64	30
				100	32	34
				133	64	30
			4024	133	32	30
			1024	100	64	30
				100	32	34
			8			18
		22	16			18
		33	64			18
600	L2 256k		1024			18
600	L2 250K		8			22
		66	16			22
		00	64			22
		66	1024			22



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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Latency
				422	64	18
			8	133	32	22
			8	400	64	22
				100	32	22
				400	64	18
			40	133	32	22
			22			
		00		100	32	22
		33		400	64	18
				133	32	22
			64	400	64	22
				100	32	22
				400	64	18
			4004	133	32	22
			1024	100	64	22
000					32	22
600	EMIFA (SDRAM)		8	133	64	26
					32	30
				400	64	30
				100	32	34
				400	64	26
			40	133	32	30
			16	400	64	30
		00		100	32	34
		66		400	64	26
			0.4	133	32	30
			64	400	64	30
				100	32	34
				400	64	26
			4004	133	32	30
			1024	400	64	30
				100	32	34



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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Latency
			8			14
			22	16		
		33	64			14
	1.2.2564		1024			14
	B 16	22				
		66	16			22
		00	64			22
			1024			22
			0	122	64	18
			0	133	32	22
			16	122	64	18
720		22	16	133	32	22
720		33	64	122	64	18
			04	133	32	22
			4004	122	64	18
	EMIEA (SDDAM)		1024	133	32	22
	EMIFA (SDRAM)		0	122	64	26
			0	133	32	30
			16	122	64	26
		66	10	133	32	30
		00	64	122	64	26
			04	133	32	30
			1024	133	64	26
			1024	133	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.



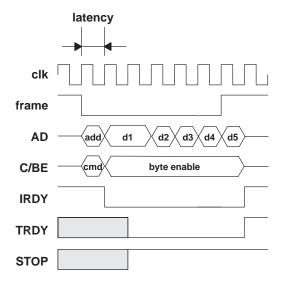


Figure 2. Write Latency Timing Diagram

### 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.



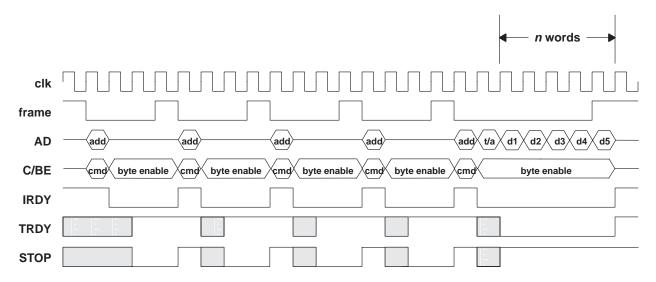


Figure 3. Number of PCI Cycles to Read 5 Words Timing Diagram

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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
			8			8
		22	16			16
		33	64			64
500	1.0.050		1024			1024
500	L2 256k		8			8
			16			16
		66	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
				400	64	8
				133	32	8
			8	400	64	8
				100	32	8
				400	64	16
			40	133	32	16
			16	400	64	16
		20		$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	16	
		33		400	64	64
			0.4	133	64	
			64	400	64	64
				100	32	64
				400	32 64 32 64 32 64 32 64 32 64	1024
			4004	133	32	1024
			1024	400	64	1024
<b>500</b>				100	32	1210
500	EMIFA (SDRAM)			422	64	8
			0	133	32	8
				100	64	8
				100	32	8
				422	32 64 64 64 32 64 64 102 32 102 64 102 32 121 64 8 32 8 64 8 32 8 64 16 32 16 64 152 32 221 64 213	16
			16	133	32	16
			16	100	64	16
		66		100	32	16
		00		122	64	8 16 16 16 16 64 64 64 64 1024 1024 1024 1210 8 8 8 8 16 16 16
			64	133	32	221
			04	100	64	213
				100	32	245
				122	64	2809
			1024	133	32	3660
			1024	100	64	3268
				$ \begin{array}{r}                                     $	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
			8			8
		00	16			16
		33	64			64
	1.0.050		1024			1024
	L2 256k		8			8
		66	16			16
		00	64			64
			1024			1024
				133	64	8
			8		32	8
			0	100	64	8 16 64 1024 8 16 64 1024 8
600					32	8
000				133	64	16
			16		32	16
			10	100	64	16 64 1024 8 16 64 1024 8 8 8 8 8 16 16 16 16 16 4 64 64 64 1024 1024 1024
	EMIFA (SDRAM)	33			32	16
	EIVIIFA (SDRAIVI)	33		133	64	16 64 1024 8 8 8 8 16 16 16 16 4 64 64 64 1024
			64		32	
			04	100	64	64
					32	16 64 64 64 64
				133	64	1024
			1024		32	1024
			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
				400	64	8
				133	32	8
			8	100	64	8
				100	32	8
600 EMIFA (SDRA				133	64	16
			16	133	32	16
			16	100	64	16
600	ENJIEN (SDDANI)	66		100	32	16
000	EIVIIFA (SDRAIVI)	00		133	64	158
			64	133	32	8 8 8 8 16 16 16 16 158 146 150 253 2730 3049 2886 4809 8 16 64 1024 8 16 64
			04	100	64	
				100	32	253
				133	64	150 253 2730 3049
			1024	133	32	3049
			1024	100	64	2886
				100	32	4809
			8			8
		33	16			16
		33	64			64
720	L2 256k		1024			1024
120	LZ 250K		8			8
		66	16			16
		00	64			64
			1024			1024



			•	<u></u>		
CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
			8	422	64	8
			0	133	32	8
			40	400	64	16
		20	16	133	32	16
		33	64	133 - 133 - 133 - 133 - 133 -	64	64
			04		32	64
			1024	122	64	1024
700				133	32	1024
720	EMIFA (SDRAM)		0	422	64	8
			8	133	32	8
			4.0	400	64	16
			16	133	32	16
		66	0.4	400	64	154
			64	133	32	8 8 16 16 64 64 1024 1024 8 8 16
			1024	422	64	2534
			1024	133	32	3245

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

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.

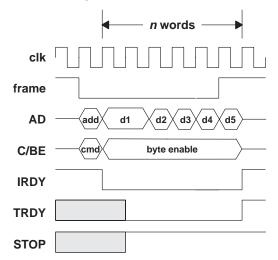


Figure 4. Number of PCI cycles to write 5 words Timing Diagram



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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
			8			9
	1.0.0504	33	1024			1025
500 600	L2 256K	00	8			9 1025 9 1265 9 9 9 9 1025 1025 1025 1025 1025 9 9 9 1417 1421 1421 9 1025 9 1025 9 1025
		00	1024			1265
				122	64	9
500				133	32	9
			0	100	64	9
500		22		133	9	
		33		400	64	1025
500			8 9 1024 102 8 9 1024 126 8 9 1024 126  133 64 9 32 9 100 32 9 100 64 102 32 102 100 32 102 100 32 102 100 32 9 100 32 9 100 32 9 100 32 9 100 32 9 100 32 9 100 32 9 100 32 9 100 32 102 100 32 9 100 32 9 100 64 102 100 64 102 100 8 9 1024 102 100 8 9 1024 102 100 32 9 100 32 9 100 32 9 100 32 9 100 32 9 100 32 9 100 32 9 100 32 9 100 32 9	1025		
500	BMIFA (SDRAM)  133    8	1025				
				100	32	1025
				422	64	9
				133	32	9
			0	100	64	9
		66		100	32	
		00		422	64	1417
			1024	133	32	1417
			1024	100	64	1421
				32 100 32 133 64 100 32 100 32 1100 32 1100 32 1100 32 1100 32 1100 32 1100 32 1100 32 1100 32 1100 32 1100 32 1100 32 1100	1421	
			8			9
	10.050	33	1024			1025
	L2 256K	00	8			9
		00	1024			1025
				400	64	9
000				133	32	9
600			0	100	64	9
		00		100	32	9
	EMIFA (SDKAM)	33		400	64	1025
			4004	133	32	1025
			1024	100	64	1025
					32	1025



Table 3.	Number of PCI Cycles to Write <i>n</i> Words
	of Data (Continued)

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Xfer
				400	64	9
			0	133	32	9
			8	400	64	9
600		00		100	32	9
600	EMIFA (SDRAM)	66	1024	400	64	1245
				133	32	1245
				400	64	1245
				100	32	1237
	L2 256k	33	8			9
		33	1024			1025
		66	8			9
		66	1024			1025
			8	422	64	9
720		33	0	133	32	9
720		33	1024	133	64	1025
			1024	133	32	1025
	EMIFA (SDRAM)		0	422	64	9
		66	8	133	32	9
		66	1024	422	64	1025
			1024	133	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.

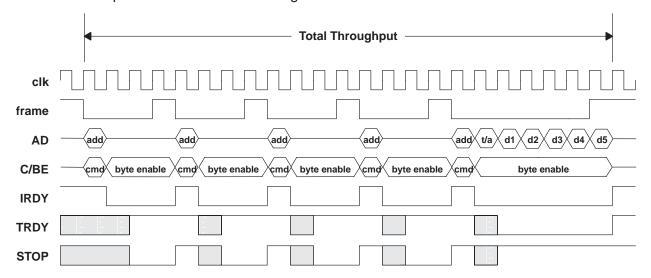


Figure 5. Read Throughput Timing Diagram

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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
		33	8			41
			16			62.7
			64			104.1
500	L2 256k		1024			131
500		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
				400	64	35.6
			0	133	32	35.6
			8	100	64	35.6
				100	32	35.6
				400	64	56.1
			40	133	32	56.1
			16	400	64	56.1
		00		100	32	56.1
		33		400	64	99.2
			0.4	133	32	99.2
			64	400	64	99.2
				100	32	99.2
			1024	400	64	130.5
				133	32	130.5
				400	64	130.5
500	EMIFA			100	32	110.8
500	(SDRAM)		8	133	64	56.1
					32	56.1
				100	64	56.1
					32	50.8
				400	64	92.8
			40	133	32	92.8
			16	400	64	92.8
		00		100	32	85.3
		66		400	64	92.8
			0.4	133	32	68
			64	400	64	70.2
				100	32	61.2
				400	64	96.2
			1024	133	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
			8			41
			16			62.7
		33	64			104.1
	1.0.0501		1024			131
	L2 256k		8			71.1
		00	16			112.3
		66	64			198.4
			1024			261.1
				400	64	41
			8	133	32	35.6
				100	64	35.6
600				100	32	35.6
600			16	133	64	62.7
					32	56.1
				100	64	56.1
	EMIFA	22			32	56.1
	(SDRAM)	33		400	64	104.1
			64	133	32	99.2
			64	400	64	99.2
				100	32	99.2
				422	64	131
			1024	133	32	130.5
				400	64	130.5
				100	32	103



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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
				400	64	62.7
			_	133	32	56.1
			8	400	64	56.1
				100	32	50.8
				133	64	101.6
			16	133	32	92.8
			16	100	64	92.8
600	EMIFA	66		100	32	85.3
600	(SDRAM)	00		133	64	92.8
			64	133	32	97
				100	64	94.8
				100	32	59.5
			1024	133	64	99.1
				133	32	88.7
				100	64	93.6
				100	32	56.4
			8			48.5
		33	16			71.1
		33	64			109.4
720	LO OEGL		1024			131.5
120	L2 256k		8			71.1
		66	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
			8	400	64	41
			8	133	32	35.6
			40	400	64	62.7
		22	16	133	32	56.1
		33	64	422	64	104.1
			64	133	32	99.2
	EMIFA (SDRAM)		1024	133	64	131
700					32	130.5
720			8	133	64	62.7
					32	56.1
				400	64	101.6
		00	16	133	32	92.8
		66	64	400	64	94.8
			64	133	32	99.2
			4004	400	64	106.7
			1024	133	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.

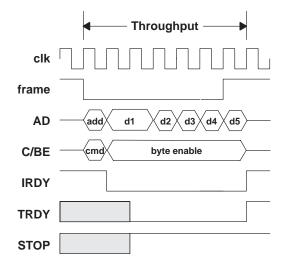


Figure 6. Write Throughput Timing Diagram



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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
		22	8			106.7
	1.0.0501	33	1024			133.1
	L2 256k	00	8			213.3
		66	1024			215.7
				133	64	106.7
			8	133	32	106.7
			8	400	64	106.7
	EMIFA	33		100	32	106.7
			1024	133	64	133.1
500					32	133.1
500				100	64	133.1
					32	133.1
	(SDRAM)			133	64	213.3
			8		32	213.3
			0	100	64	213.3
		00		100	32	213.3
		66		400	64	192.6
			4004	133	32	192.6
			1024	400	64	192
				100	32	192



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

CPU	SRC/DST	PCI	Burst Length	EMIF	EMIF Width	Throughput
		00	8			106.7
	L2 256k	33	1024			133.1
		00	8			213.3
		66	1024			266.1
				400	64	106.7
			0	133	32	106.7
			8	400	64	106.7
		00		100	32	106.7
		33		400	64	133.1
000			4004	133	32	133.1
600			1024	400	64	133.1
	EMIFA			100	32	133.1
	(SDRAM)	66		400	64	213.3
			0	133	32	213.3
			8	100	64	213.3
				100	32	213.3
				133	64	219.2
			1024		32	219.2
				100	64	219.2
				100	32	220.6
		22	8			106.7
	L2 256k	33	1024			133.1
	L2 200K	66	8			213.3
		66	1024			266.1
			0	400	64	106.7
720		22	8	133	32	106.7
720		33	4024	122	64	133.1
	EMIFA		1024	133	32	133.1
	(SDRAM)		8	133	64	213.3
		66	Ŏ	133	32	213.3
			1004	122	64	266.1
			1024	133	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.



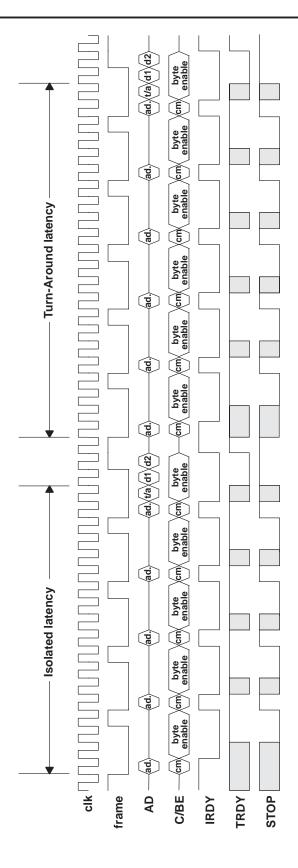


Figure 7. Read/Read Turn-Around Penalty

24



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	Туре	EMIF	EMIF Width	T/A Penalty
			R/W			0
		33	R/R			4
			W/R			8
	1.0.0504		W/W			8
	L2 256k		R/W			0
		00	R/R			4
		66	W/R			8
			W/W			8
				400	64	0
			R/W	133	32	0
				100	64	0
<b>500</b>					32	0
500			R/R	133	64	0
				133	32	0
				400	64	0
	ENNEA (CDDANA)	33		100	32	4
	EMIFA (SDRAM)	33		400	64	8
			W/R	133	32	8
			VV/K	100	64	8
				100	32	8
				122	64	8
			10/04/	133	32	8
			W/W	100	64	8
				100	32	8



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

CPU	SRC/DST	PCI	Туре	EMIF	EMIF Width	T/A Penalty
				422	64	0
			R/W	133	32	0
			K/VV	100	64	0
				100	32	0
				133	64	16
			R/R	133	32	20
			K/K	400	64	20
500	ENTER (CDDANA)	00		100	32	20
500	EMIFA (SDRAM)	66		400	64	8
			W/R	133	32	12
				100	64	8
					32	8
			W/W	133	64	8
					32	8
				100	64	8
				100	32	8
			R/W			0
		33	R/R			0
		33	W/R			8
600	L2 256k		W/W			8
000	LZ ZOOK		R/W			0
		66	R/R			4
		66	W/R			8
			W/W			8



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

CPU	SRC/DST	PCI	Туре	EMIF	EMIF Width	T/A Penalty
				133	64	0
			R/W	133	32	0
			I K/VV	400	64	0
				100	32	0
				400	64	4
			D/D	133	32	0
			R/R	400	64	0
		00		100	32	4
		33		400	64	8
			\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	133	32	8
			W/R	400	64	8
				100	32	8
			W/W	400	64	8
				133	32	8
				100	64	8
000	E141E4 (ODD 414)				32	8
600	EMIFA (SDRAM)		R/W	400	64	0
				133	32	0
				400	64	0
				100	32	0
				400	64	16
			D/D	133	32	16
			R/R	400	64	20
		00		100	32	24
		66		400	64	12
			\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	133	32	8
			W/R	400	64	8
				100	32	8
				400	64	8
				133	32	8
			W/W		64	8
				100	32	8



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

CPU	SRC/DST	PCI	Туре	EMIF	EMIF Width	T/A Penalty
	L2 256k	33	R/W			0
720			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

Copyright © 2003, Texas Instruments Incorporated