Technical Notes on using Analog Devices' DSP components and development tools
Contact our technical support by phone: (800) ANALOG-D or e-mail: dsp.support@analog.com
Or visit our on-line resources http://www.analog.com/dsp and http://www.analog.com/dsp/EZAnswers

ADSP-BF535 Blackfin® Processor PCI Interface Performance

Contributed by Jorge Manguane *September 3, 2003*

Introduction

This Engineer to Engineer Note will briefly discuss the performance of the ADSP-BF535 Blackfin® Processor PCI interface. In general, maximum PCI performance is achieved during burst transfers, where a single *address phase* (one clock cycle at the beginning of the transaction where the address and transfer type are output on the AD bus and on the C/BE lines, respectively) is followed by multiple *data phases*. It is therefore easy to see how single data transfers may yield lower throughput. We will look at the ramifications of each of these transfer types, namely, burst and single word transfers.

To validate this analysis the Eagle-35 and the Hawk-35 boards were used as a system host and PCI device, respectively. Both boards have the ADSP-BF535 Blackfin® Processor as the main processor and are sold and supported by Momentum Data Systems, MDS. Also, a VMETRO PCI bus analyzer/exerciser was used to both monitor the PCI traffic along with its metrics and to exercise the bus.

It should be noted that a PCI-to-PCI bridge is used on the Hawk-35 board to interface the BF535 PCI core interface to the PCI bus signals. The main purpose of this bridge is to perform automatic voltage detection and translation in order to allow the board to be plugged into either a 3.3 Volt or 5 Volt system.

On the BF535, a dedicated bus is available on chip to allow an external bus master to transfer data directly to/from internal (L2 memory) or external memory spaces without involving the processor core.

Burst transfers

When the BF535 is the bus master, burst transfers are accomplished through Direct Memory Access (DMA), in particular through the processor's Memory DMA (MemDMA) engine. For PCI transfers, MemDMA allows memory to memory DMA transfers between PCI memory space and either internal L2 or SDRAM memory.

The MemDMA engine's burst size is 8 words, which means that after every 8 words transferred the PCI interface issues the next PCI address. This situation is illustrated in the PCI trace shown in Figure 1, where every address phase (Start) is followed by eight data words. A new Start (address phase) transfer follows, and the process goes on until all data has been transferred. Not every transaction (address phase, data phase) will be the same (i.e., 8 data words for each address phase) because at some point the target may issue "retries." These retries will force the PCI bus master to reissue a new address. Figure 2 shows the PCI bus statistics including the *Data Transfers per Transaction* , which indicate an average number of data words transferred during each burst access.

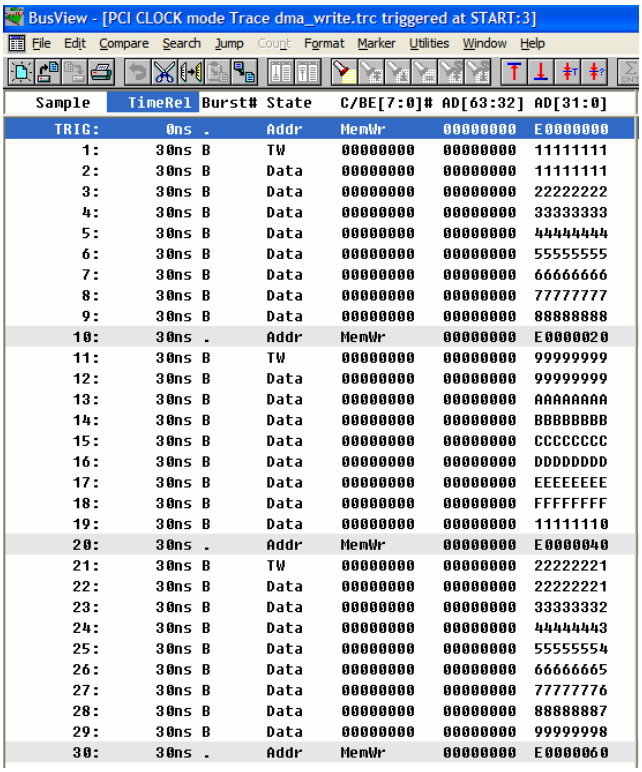

Figure 1. Eagle-35 Burst write trace

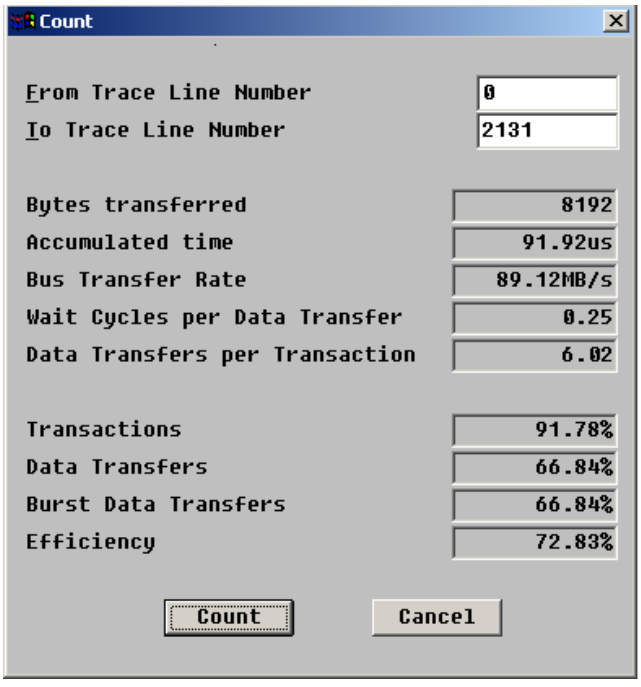

Figure 2. Eagle-35 Burst Write statistics

These metrics were obtained from a DMA write transfer between the Eagle-35 board (as the bus master) and the Hawk-35 board (as the bus slave). The target memory accessed was the on-chip ADSP-BF535 L2 memory. This will be the case for the remainder of the traces shown in this application note. Accesses to SDRAM yielded a slight (insignificant) decrease in throughput – refer to the metrics in Table 1. As it can be seen from Figure 2. Eagle-35 Burst Write statistics, a high throughput of 89.12 MB/s was achieved, as well as an efficiency of 72.83%. Here the efficiency is measured as a ratio of the percentage of data transfers against the percentage of total transactions.

The processors' system clocks, SCLK, on both boards were set to 131 MHz, with the core clocks running at 262 MHz. Since the maximum burst length on the BF535 is 8, SCLK is a major factor that influences PCI throughput. As an example, when SCLK was set to 120 MHz, the observed throughput was 80 MB/s. Because the MemDMA engine operates in the SCLK domain, it is apparent that a higher SCLK allows more data transferred per unit time. Note, however, that 133 MHz is the maximum frequency to which SCLK can be set.

When the BF535 is the target (slave) of a burst transfer, the initiator's burst size will determine the amount of data transferred per transaction. As an example, the burst length of the VMETRO's PCI exerciser can be specified to an arbitrary length. Burst length also influences throughput. As the burst length increases, the number of transactions needed to complete the transfer decreases. Figure 3 shows the Hawk-35 being accessed by the VMETRO's PCI exerciser (read access), and Figure 4 shows the corresponding bus statistics. Here the burst length was set to equal the number of bytes transferred.

PCI TRANSFER mode Trace on Tracer triggered at START

Sample	TimeRel	Wait	Size	Burst	Command	Address	Data	Status
TRIG:	0ns	.	AD32	Start	MemRd	E0000000	....	TdwodTr
1:	840ns	.	AD32	Start	MemRd	E0000000	11111111	OK
2:	30ns	.	AD32	B	MemRd	E0000000	22222222	OK
3:	30ns	.	AD32	B	MemRd	E0000000	33333333	OK
4:	30ns	.	AD32	B	MemRd	E0000000	44444444	OK
5:	30ns	.	AD32	B	MemRd	E0000000	55555555	OK
6:	30ns	.	AD32	B	MemRd	E0000000	66666666	OK
7:	30ns	.	AD32	B	MemRd	E0000000	77777777	OK
8:	30ns	.	AD32	B	MemRd	E0000000	88888888	OK
9:	30ns	.	AD32	B	MemRd	E0000000	99999999	OK
10:	30ns	.	AD32	B	MemRd	E0000000	AAAAAAAA	OK
11:	30ns	.	AD32	B	MemRd	E0000000	BBBBBBBB	OK
12:	30ns	.	AD32	B	MemRd	E0000000	CCCCCCCC	OK
13:	30ns	.	AD32	B	MemRd	E0000000	DDDDDDDD	OK
14:	30ns	.	AD32	B	MemRd	E0000000	EEEEEEEE	OK
15:	30ns	.	AD32	B	MemRd	E0000000	FFFFFFFF	OK
16:	30ns	.	AD32	B	MemRd	E0000000	11111110	OK
17:	30ns	.	AD32	B	MemRd	E0000000	22222221	OK
18:	30ns	.	AD32	B	MemRd	E0000000	33333332	OK
19:	30ns	.	AD32	B	MemRd	E0000000	44444443	OK
20:	30ns	.	AD32	B	MemRd	E0000000	55555554	OK

Figure 3. Exerciser Burst Reads trace

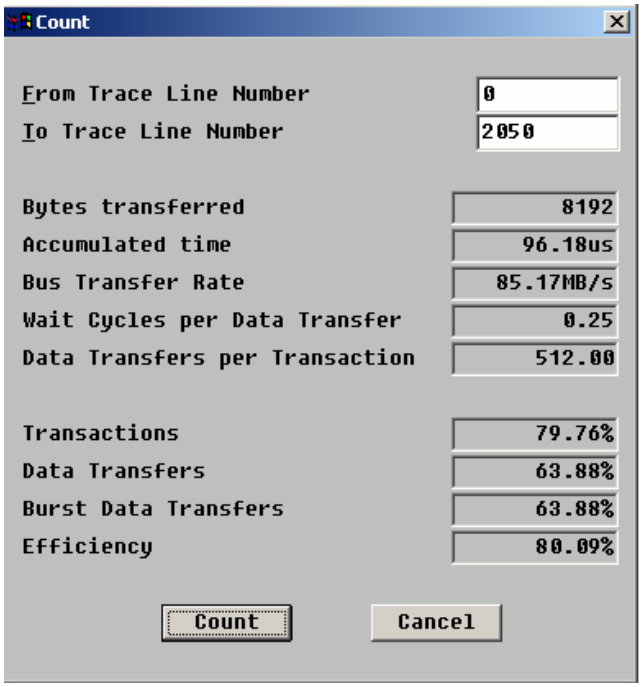

Figure 4. PCI Exerciser Burst Read statistics

Single Word Accesses

For the case of single word accesses, each transaction consists of one address phase followed by one data word in the data phase. Writes are posted, meaning that the transaction is buffered at an intermediate agent (e.g., a bridge from one bus to another -- transaction FIFOs on the BF535) and completes at the source before actually completing at the destination. The BF535 transaction FIFO is 4 transactions deep. Writes are posted as long as there's space in both the transaction and transmit data FIFOs (8 x 32-bit words deep). The posting of writes allows the BF535 PCI interface to reach the theoretical maximum throughput of 44 MB/s for single-word write accesses.

BusView - [PCI CLOCK mode Trace core_write.trc triggered at START:4]

File Edit Compare Search Jump Count Format Marker Utilities Window Help

Sample	TimeRel	Burst#	State	C/BE[7:0]#	AD[63:32]	AD[31:0]
TRIG:	0ns	.	Addr	MemWr	00000000	E0000FFC
1:	30ns	.	TW	00000000	00000000	11111111
2:	30ns	.	Data	00000000	00000000	11111111
3:	30ns	.	Addr	MemWr	00000000	E0001000
4:	30ns	.	TW	00000000	00000000	22222222
5:	30ns	.	Data	00000000	00000000	22222222
6:	30ns	.	Addr	MemWr	00000000	E0001004
7:	30ns	.	TW	00000000	00000000	33333333
8:	30ns	.	Data	00000000	00000000	33333333
9:	30ns	.	Addr	MemWr	00000000	E0001008
10:	30ns	.	TW	00000000	00000000	44444444
11:	30ns	.	Data	00000000	00000000	44444444
12:	30ns	.	Addr	MemWr	00000000	E000100C
13:	30ns	.	TW	00000000	00000000	55555555
14:	30ns	.	Data	00000000	00000000	55555555
15:	30ns	.	Addr	MemWr	00000000	E0001010
16:	30ns	.	TW	00000000	00000000	66666666
17:	30ns	.	Data	00000000	00000000	66666666
18:	30ns	.	Addr	MemWr	00000000	E0001014
19:	30ns	.	TW	00000000	00000000	77777777
20:	30ns	.	Data	00000000	00000000	77777777
21:	30ns	.	Addr	MemWr	00000000	E0001018
22:	30ns	.	TW	00000000	00000000	88888888
23:	30ns	.	Data	00000000	00000000	88888888
24:	30ns	.	Addr	MemWr	00000000	E000101C
25:	30ns	.	TW	00000000	00000000	99999999
26:	30ns	.	Data	00000000	00000000	99999999
27:	30ns	.	Addr	MemWr	00000000	E0001020
28:	30ns	.	TW	00000000	00000000	AAAAAAAA
29:	30ns	.	Data	00000000	00000000	AAAAAAAA
30:	30ns	.	Addr	MemWr	00000000	E0001024

Figure 5. Single word writes trace. Eagle-35 (master) to Hawk-35 (target)

On the other hand, reads are not posted, and for every transaction there's at least one extra cycle added between the address and data phases – this cycle is referred to as the "turnaround cycle," after which the target drives the data on the PCI Address/Data bus. In addition, both the target and the master may introduce wait states, or the target may disconnect the transaction with no data being transferred, which would greatly reduce the PCI throughput for single read accesses.

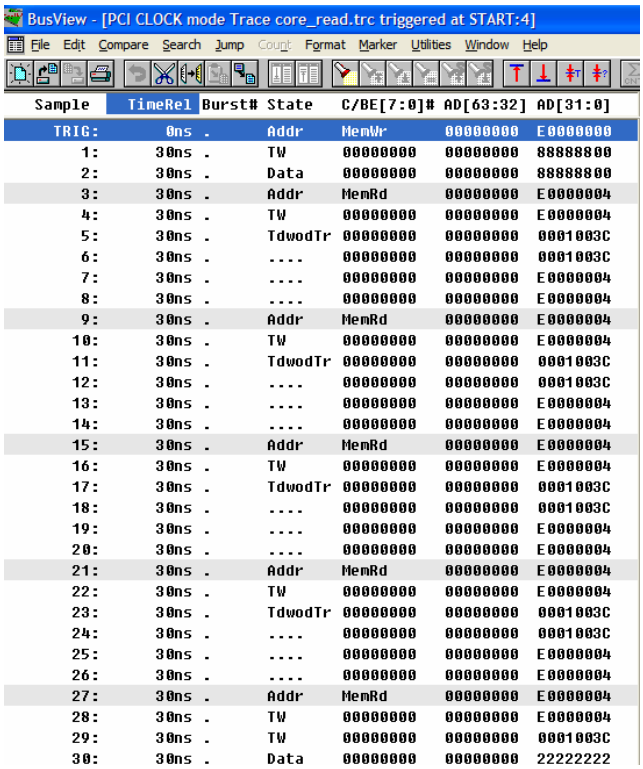

Figure 6. Single word reads trace. Eagle-35 (master) to Hawk-35 (target)

The following tables summarizes the ADSP-BF535 PCI performance metrics:

Access Type	Target memory	Throughput	Efficiency
Burst Writes	L2	89.12 MB/s	72.83 %
	SDRAM	88.26 MB/s	72.44%
Burst Reads	L2	26.04 MB/s	40 %
	SDRAM	25.97 MB/s	39.91%
Sequential Single Writes	L2	44.42 MB/s	33.33%
	SDRAM	44.42 MB/s	33.33
Sequential Single Reads	L2	4.16 MB/s	9.37%
	SDRAM	3.55 MB/s	8%

Table 1. Transfer between Eagle-35 (PCI master) and Hawk-35 (PCI device) both with an SCLK of 131 MHz.

Access Type	Efficiency	Throughput
Burst Reads	80.09 %	85.17 MB/s
Burst Writes	95.48 %	57.96 MB/s

Table 2. Transfer between the VMETRO's PCI exerciser (PCI master) and the Hawk-35 board (PCI device)

Conclusion

This Engineer to Engineer note provides some PCI benchmarks and performance considerations that should be helpful in determining the usage of the ADSP-BF535 PCI interface appropriate to the architecture of a given PCI agent. For instance, it may make sense to have the ADSP-BF535 PCI interface perform a write to a PCI agent rather than have the PCI agent read the ADSP-BF535 PCI memory. This may, however, not be necessary if the reading agent has a rather large burst length, as was the case above for the VMETRO PCI exerciser.

References

[1] ADSP-BF535 Blackfin® Processor Hardware Reference Revision 2.0, April 2003 Analog Devices, Inc.

[2] PCI Hardware and Reference, 5th Edition, Solari & Willse.

[3] PCI System Architecture, 4th Edition, Mindshare, Inc

[4] PCI Local Bus Specification Revision 2.2 December 18, 1998

Document History

Version	Description
September 3, 2003 by J Manguane	Initial Final Release
August 4, 2003 by J Manguane	Initial Draft Release