Application Report
SPRA883 – December 2002

OMAP System DMA Throughput Analysis

Bill Winderweedle *DSP Catalog and EEE, Device Applications*

ABSTRACT

The OMAPTM architecture devices contain a nine-channel system DMA, useable for block transfers of data without intervention by the TIARM925T MPU or the TMSC55x DSP. This application report describes optimization techniques for improving throughput. Examples are given for using external memories such as SDRAM and SRAM for transferring data to and from internal memory. Internal memory transfers are also discussed.

Contents

List of Figures

List of Tables

1 Introduction

The OMAP architecture devices contain a nine-channel system DMA, useable for block transfers of data without intervention by the ARM9TDMI MPU or the C55x DSP. This application report analyzes the throughput between some of the possible sources and destinations using the SDMA. Optimization techniques for improving throughput are discussed.

2 System Setup

The following were used in the course of creating this applications report:

- A test board with external Samsung K4S561632C-TL75 256M-bit SDRAM and 512KB Samsung K6T4016U3C-TB70 SRAM
- OMAP Code Composer Studio v2.0 for TIARM925 MPU and C55X DSP
- Tektronix TLA714 Logic Analyzer
- POMAP5910CGZG

The OMAP5910 MPU and DSP clocks were set up for 150MHz operation. The OMAP5910 Traffic Controller (TC) clock was set up for 75MHz operation. The system DMA (SDMA) and internal SRAM (IMIF) run at the same speed as the TC. The slow external memory interface (EMIFS) connected to asynchronous SRAM was clocked at 37.5MHz, with 3 waitstates (WS) for reads and 1 WS for writes, with a write enable length (WELEN) of 1. The fast external memory interface (EMIFF) connected to SDRAM was clocked at 75MHz with a CAS latency of 2. The MPUI division factor (DIVF) for APIF_HNSTROBE was set to 2. All clock domains, peripherals, and SDMA channels unnecessary for performing the throughput analysis were disabled. The clock related settings are summarized in Table 1.

Table 1. Clock Setup

DSP	MPU	MPUI	TC/IMIF/SDMA	EMIFF		EMIFS			
150MHz	150MHz	DIVF 2	75MHz	75MHz	CAS 2	37.5MHz	WELEN 1	3WS Rd	1WS Wr

A combination of ARM assembly and unoptimized C code was used for performing the system DMA transfers on a single channel. The ARM code executed from EMIFS SRAM. Test setup and results tables were stored in internal IMIF. The MPUI port was configured for shared access mode. The DSP executed in a polling loop within SARAM0, located at DSP starting address 0x10000. Iterations of tests were run both with and without ARM I-cache enabled.

The external latency time was measured using a GPIO pin as a logic analyzer trigger. After the GPIO pin was driven high in the test code, the dma channel was directly enabled. No internal or external sync events were used. Latency was not measured where both the source and destination for the transfer were contained within IMIF or DSP internal memory accessed through MPUI. The EMIFS addresses and data busses were observed on the logic analyzer for the cycle of the code fetch where the DMACCR register was set. The latency was derived in TC cycles from the EMIFS code fetch cycle. Therefore, some overhead is included.

Average throughput was measured using an ARM timer peripheral running at 6 MHz. The timer was enabled in code before the logic analyzer was enabled and disabled, after a polling loop detected the end of the transfer. The timer values were scaled to derive the TC cycle count. A code segment used to enable and measure the throughput is shown in Figure 1.

```
// Enable ARM timers
      RunTimersOneTwoThree();

// Set MPUIO 3 high, use this for logic analyzer trigger
  ARMIO_SET_GPIO_BIT(ARMIO_HIGH,3);
// Enable DMA channel 0
      DMA_EnableChannel(DMA_CHANNEL_0);
      // Polling Channel 0 Status registers bit ’FrameEnd’
      while((DMA_ACC(NO_GLOB_REG,
DMA_CHANNEL_0,DMA_CSR)&(0x0001<<SYSDMA_DMA_CSR_CHX_FRAME_POS))==0);
// Disable ARM timers
     StopTimersOneTwoThree();
```

Figure 1. Code Segment for Logic Analyzer Trigger, DMA Enable, and Transfer End Polling Loop

2.1 Peripheral Registers Setup

TC was configured as follows:

- EMIFS_PRIO, EMIFF_PRIO, and IMIF_PRIO settings
  - 31:16 reserved
  - 15:12 LB host set to 000b for single transfers, not used in this analysis
  - 11:8 System DMA set to 0000b and 1111b for 1 and 16 consecutive transfers while other ports are waiting. The larger number provides optimal throughput at the expense of the other ports.
  - 7 reserved
  - 6:4 DSP (CPU or DMA) set to 000b for single transfers, not used in this analysis
  - 3 reserved
  - 2:0 ARM set to 000b for single transfers, ARM EMIFS code fetch (EMIFS_PRIO) affected the transfers in this analysis. IMIF result table writes were not factored into this analysis because this occurred at the end of the test, after the system DMA transfer was completed.
- EMIFS_CONFIG_REG = 0x00000012
  - Boot mode was set for CS3 space at 0x00000000
- EMIFS_CS3_CONFIG = 0x00001139
  - WELEN=1, WRWST=1, RDWST=3, FLCLKDIV=TC/2
- EMIFF_SDRAM_CONFIG = 0x101D4F4
  - SDF1 frequency range, auto-refresh enabled and calculated from SDRAM data sheet for minimum, 16-bit bus, 256 M bits, 4 banks

- EMIFF_MRS = 0x00000027
  - CAS latency 2, full-page burst length
- TIMEOUT1 = TIMEOUT2 = TIMEOUT3 = 0x00000000
- ENDIANISM = 0x00000000
  - Little-endian

SDMA was configured for channel 0 operation as follows:

- DMA_GCR = 0x0000000C
  - Clock autogating on, free
- DMA_CSDP = various
  - No burst and 16-bit, no burst and 32-bit, source and destination burst 4 and 16/32-bit
- DMA_CCR = 0xA060
  - Source and destination single index, entire frame is transferred
- DMA_CICR = 0x0009
  - Frame and timeout interrupts enabled
- DMA_CSSA_L = various
- DMA_CSSA_U = various
- DMA_CDSA_L = various
- DMA_CDSA_U = various
- DMA_CEN = 0x4e20 and 0x2710
  - 20,000 and 10,000 elements
- DMA_CFN = 1
  - Single frame
- DMA_CFI = 1
  - Single frame index
- DMA_CEI = 1
  - Single element index

3 Throughput Analysis Examples

Throughput examples were generated for transfers between all combinations of EMIFF, EMIFS, IMIF, and MPUI. Transfers of 10Kx32-bits and 20Kx16-bits were done *without* burst; transfers of 10Kx32-bits and 20Kx16-bits were done *with* burst. MPUI burst is not supported, but transfers with a different type of source or destination were examined. Table 2 summarizes the types of transfers and the ARM memory map source and destination addresses.

Table 2. Transfer Sources, Destinations, Types, ARM Addresses

	Transfer Destination			
Transfer Source	**IMIF**	**EMIFS**	**EMIFF**	**MPUI**
IMIF	Src:0x20002000	Src:0x20002000	Src:0x20002000	Src:0x20002000
	Dest:0x20007000	Dest:0x00040000	Dest:0x10002000	Dest:0xE0014000
	20Kx16 no burst	20Kx16 no burst	20Kx16 no burst	20Kx16 no burst
	10Kx32 no burst	10Kx32 no burst	10Kx32 no burst	10Kx32 no burst
	20Kx16 burst 4	20Kx16 burst 4	20Kx16 burst 4	20Kx16 burst 4
	10Kx32 burst 4	10Kx32 burst 4	10Kx32 burst 4	10Kx32 burst 4
EMIFS	Src:0x00000000	Src:0x00000000	Src:0x00000000	Src:0x00000000
	Dest:0x20002000	Dest:0x00040000	Dest:0x1000C000	Dest:0xE0014000
	20Kx16 no burst	20Kx16 no burst	20Kx16 no burst	20Kx16 no burst
	10Kx32 no burst	10Kx32 no burst	10Kx32 no burst	10Kx32 no burst
	20Kx16 burst 4	20Kx16 burst 4	20Kx16 burst 4	20Kx16 burst 4
	10Kx32 burst 4	10Kx32 burst 4	10Kx32 burst 4	10Kx32 burst 4
EMIFF	Src:0x10002000	Src:0x10002000	Src:0x10002000	Src:0x10002000
	Dest:0x20002000	Dest:0x00040000	Dest:0x1000C000	Dest:0xE0014000
	20Kx16 no burst	20Kx16 no burst	20Kx16 no burst	20Kx16 no burst
	10Kx32 no burst	10Kx32 no burst	10Kx32 no burst	10Kx32 no burst
	20Kx16 burst 4	20Kx16 burst 4	20Kx16 burst 4	20Kx16 burst 4
MPUI	Src:0xE0000000	Src:0xE0000000	Src:0xE0000000	Src:0xE0000000
	Dest:0x20002000	Dest:0x00040000	Dest:0x10002000	Dest:0xE0015000
	20Kx16 no burst	20Kx16 no burst	20Kx16 no burst	20Kx16 no burst
	10Kx32 no burst	10Kx32 no burst	10Kx32 no burst	10Kx32 no burst
	20Kx16 burst 4	20Kx16 burst 4	20Kx16 burst 4	Burst not supported
	10Kx32 burst 4	10Kx32 burst 4	10Kx32 burst 4	

NOTE: MPUI burst is not supported. All other memories used in this analysis are supported by System DMA bursts.

Logic analyzer data was captured to detect the latency for the first read of an external dma transfer source and the first write of an external dma destination. Latency was not measured for internal memory sources and destinations. Throughput data was derived for the average number of 16-bit words per TC cycle. A transfer of a single element was measured and subtracted from the large block transfers to remove overhead. Throughput was measured with ARM I-cache enabled/disabled and the priority for dma at a minimum 0 and maximum 15.

In the following sections, examples are given for the various transfers between external and internal memories.

3.1 SDMA Transfers with EMIFS Destination

System DMA transfers with EMIFS as the destination and various sources are shown in Table 3, through Table 6.

Table 3. IMIF Transfers to EMIFS: Latency, Throughput

		Latency (TC cycles)		**Average Throughput (TC cycles per 16-bit word)**		
IMIF Source	Src:0x20002000	First read		I-cache disabled		I-cache enabled
EMIFS Destination	Dest:0x20007000		First write	DMA Priority 0	DMA Priority 15	DMA priority 15
	20Kx16 no burst	Internal	121.4	35.0	14.0	14.0
	10Kx32 no burst	Internal	121.4	22.5	12.0	12.0
	20Kx16 burst 4	Internal	121.4	13.1	10.5	10.5
	10Kx32 burst 4	Internal	121.4	13.1	10.5	10.5

Since code is being fetched from EMIFS by the MPU, this negatively impacts the system DMA writes. Use of dma priority 15, while keeping the MPU at dma priority 0, results in greater than 2x of improvement in average throughput. However, the cache does not provide improvement in this case, because the code segment used for the transfer is relatively small. Bursting improves throughput for all cases of I-cache enabled/disabled or priority settings.

Table 4. EMIFS Transfers to EMIFS: Latency, Throughput

		Latency (TC cycles)		**Average Throughput (TC cycles per 16-bit word)**		
EMIFS Source	Src:0x00000000	First read		I-cache disabled		I-cache enabled
EMIFS Destination	Dest:0x20002000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	114.9	167.1	72.0	30.0	30.0
	10Kx32 no burst	114.9	212.1	46.0	25.0	25.0
	20Kx16 burst 4	114.9	227.9	26.5	24.2	24.2
	10Kx32 burst 4	114.9	227.9	26.5	24.2	24.2

Similar improvements for priority, and bursting hold true for EMIFS to EMIFS.

Table 5. EMIFF Transfers to EMIFS: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
EMIFF Source	Src:0x10002000	First read		I-cache disabled		I-cache enabled
EMIFS Destination	Dest:0x20002000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	96.9	121.4	35.0	14.0	14.0
	10Kx32 no burst	96.9	121.4	22.5	12.0	12.0
	20Kx16 burst 4	96.9	121.4	13.1	10.6	10.6
	10Kx32 burst 4	96.9	121.4	13.1	10.6	10.6

The throughput data for EMIFF to EMIFS transfers is nearly identical to IMIF to EMIFS transfers. The EMIFS writes are the dominating factor of the transfer.

Table 6. MPUI Transfers to EMIFS: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
MPUI Source	Src:0xE0000000	First read		I-cache disabled		I-cache enabled
EMIFS Destination	Dest:0x20002000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	Internal	121.4	35.0	14.0	14.0
	10Kx32 no burst	Internal	147.6	22.5	13.0	13.0
	20Kx16 burst 4	Internal	224.9	13.1	13.1	13.1
	10Kx32 burst 4	Internal	224.9	13.6	13.6	13.6

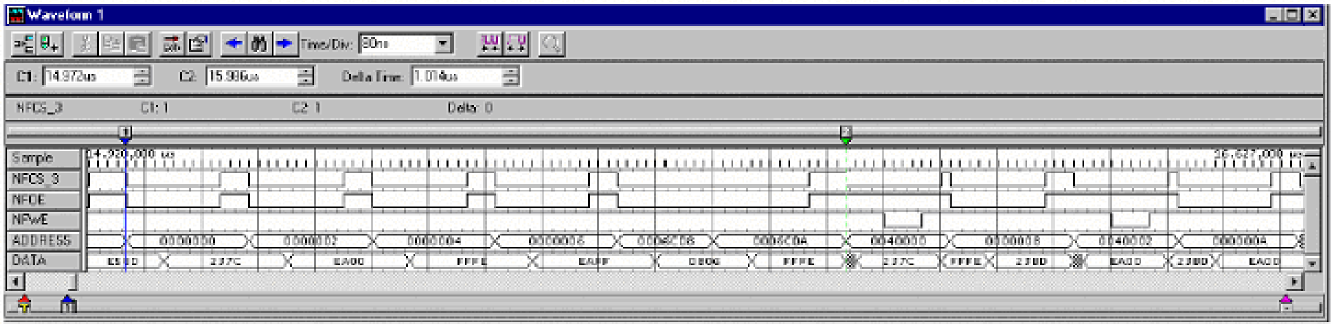
NOTE: MPUI bursts are not supported. Bursting is used on destination EMIFS only.

MPUI transfers to EMIFS continue the trend of the others. MPUI reads are the limiting factor in this example. Burst reads of the MPUI are not supported. This is reflected in the slightly worse throughput, with bursting set for the destination EMIFS.

3.1.1 Logic Analyzer Images of EMIFS to EMIFS System DMA Transfers

Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, and Figure 7 are images captured using a logic analyzer with the system DMA configured for EMIFS to EMIFS transfers in various modes.

For the capture image figure from a logic analyzer in Figure 2, and others to follow, the sample ticks are at the same rate as the TC clock (75 MHz). Beginning at cursor 1, four consecutive SDMA non-burst 16-bit reads of external asynchronous SRAM are followed by a 32-bit CPU fetch. At cursor 2, a SDMA write follows the CPU fetch and the captured sequence ends with another SDMA read. As configured in this example, the SDMA EMIFS read cycles are 10 TC clocks and the SDMA EMIFS write cycles are 10 TC clocks. The CPU EMIFS 32-bit fetch cycle shown is 20 TC cycles long. The EMIFS memory space select signal NFCS_3 is inactive for 3 TC clocks after a SDMA read, 1 TC clock after a SDMA write, and 4 TC clocks after a CPU fetch. For more details on the EMIFS cycle behavior, see Table C–1 and Table C–2 in Appendix C.


Waveform 1

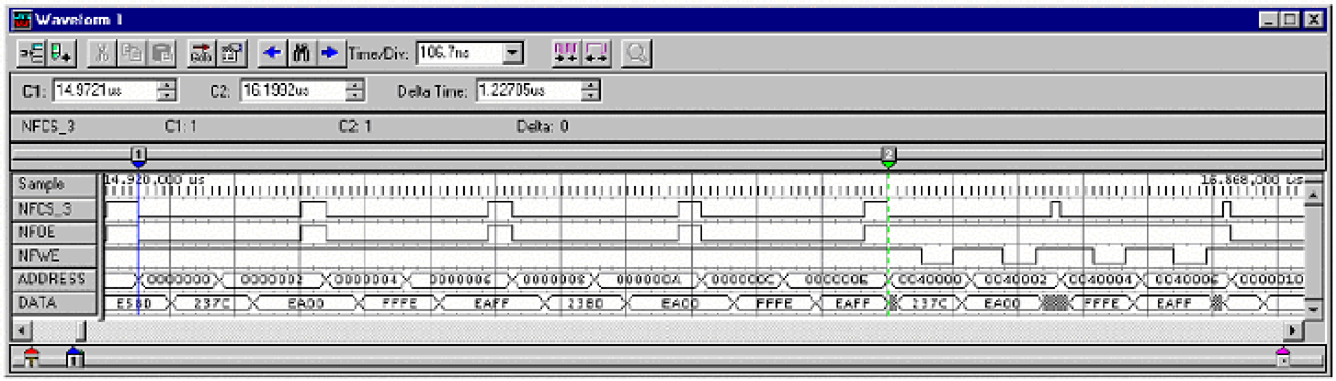
Time/Div: 80ns

C1: 14.972us C2: 15.986us Delta Time: 1.014us

NFCS_3 C1: 1 C2: 1 Delta: 0

Figure 2. System DMA EMIFS Reads and Writes (No Burst, 16-bit)

Similar to the 32-bit CPU fetch seen in Figure 2, for 32-bit reads and writes shown in Figure 3, the throughput is improved by the removal of the inactive TC cycles penalty between the consecutive 16-bit accesses which make up the complete 32-bits. The inactive TC clocks after SDMA EMIFS 32-bit reads and writes are the same for as for 16-bit mode.


Waveform 1

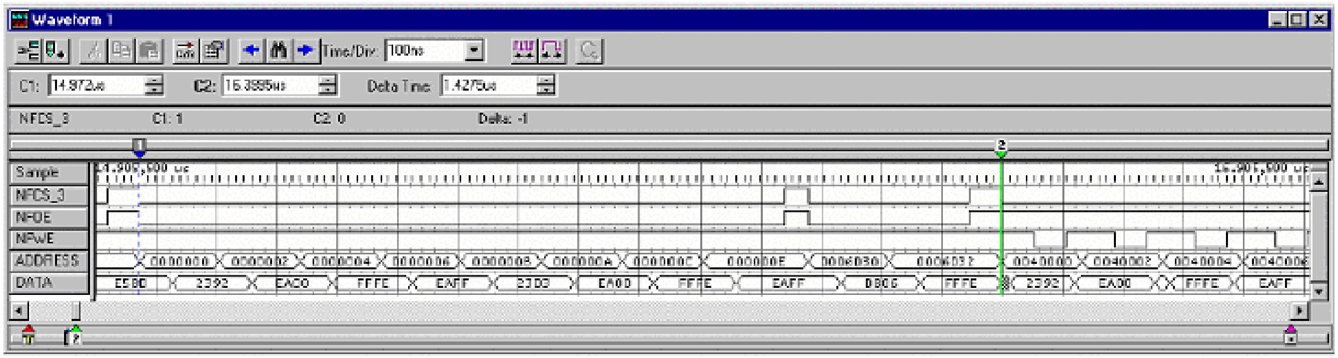
Time/Div: 106.7ns

C1: 14.9721us C2: 16.1992us Delta Time: 1.22705us

NFCS_3 C1: 1 C2: 1 Delta: 0

Figure 3. System DMA EMIFS Reads and Writes (No Burst, 32-bit)

At cursor 1, 2 consecutive System DMA EMIFS read bursts of 4 by 16-bit are shown in Figure 4. The EMIFS interface combines these into a single 8 by 16-bit burst read to the external SRAM. Thus, avoiding the wasted cycles between the single 16-bit or 32-reads in the previous examples. The entire set of bursts completes in 10 TC cycles per 16-bit word (80 TC cycles total) with a following inactive /CS for 3 TC cycles, seen previously. A MPU fetch occurs after the burst read. Following the fetch and its trailing inactive /CS for 4 TC cycles, at cursor 2, a System DMA EMIFS burst write begins writing the data from the read.



Waveform 1

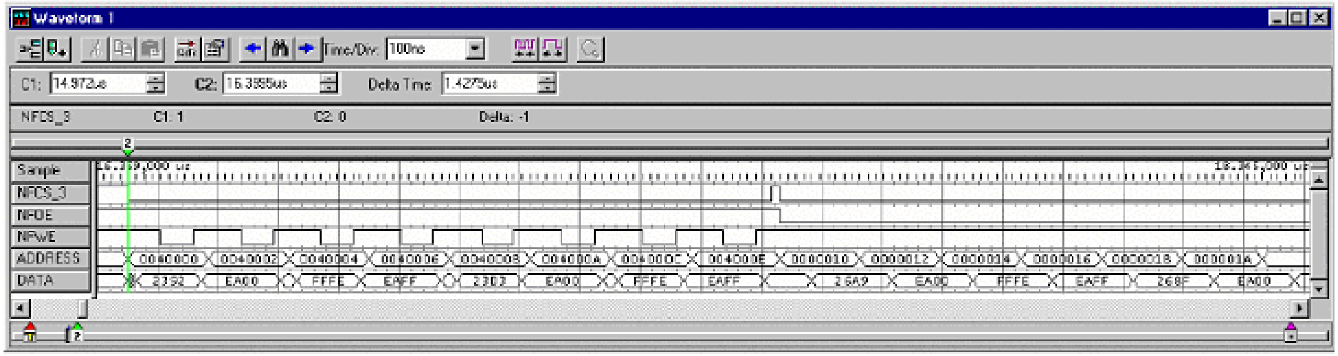
Time/Div: 100ns

C1: 14.972us C2: 16.3995us Delta Time: 1.4275us

NFCS_3 C1: 1 C2: 0 Delta: -1

Figure 4. System DMA EMIFS Read (Burst of 4 by 16-bit)

Two consecutive system DMA EMIFS burst writes of 4 by 16-bit are shown in Figure 5. Similar to the burst read of the same mode, the EMIFS implements this as a single 8 by 16 bit burst. Another set of 2 burst reads begins at the completion of the burst writes.



Waveform 1

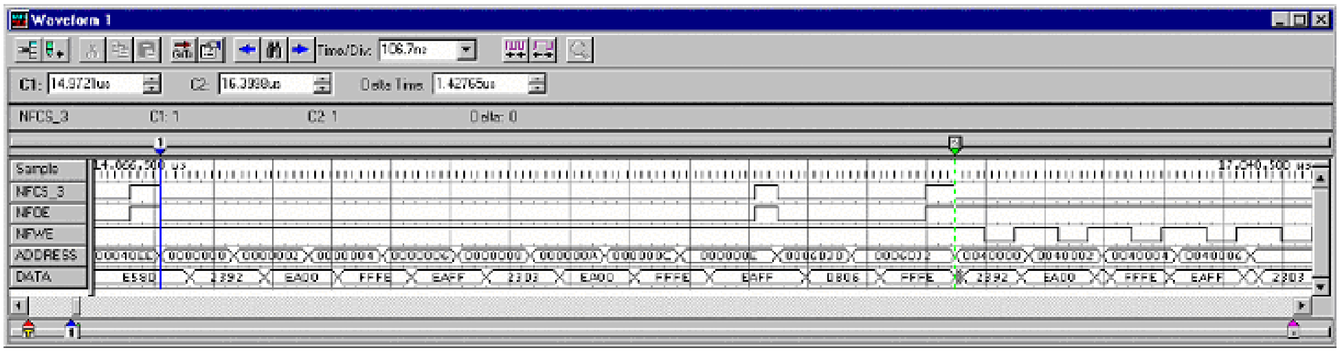
Time/Div: 100ns

C1: 14.972us C2: 16.3995us Delta Time: 1.4275us

NFCS_3 C1: 1 C2: 0 Delta: -1

Figure 5. System DMA EMIFS Write (Burst of 4 by 16-bit)

Figure 6 shows a single System DMA EMIFS burst read of 4 by 32-bit implemented as a single 8 by 16-bit burst read by the EMIFS. Externally, the behavior is the same as for the 2 consecutive 4 by 16-bit bursts. However, internally throughput is improved by the wider bus usage. A MPU fetch and the beginning of a burst write follow the burst read.



Waveform 1

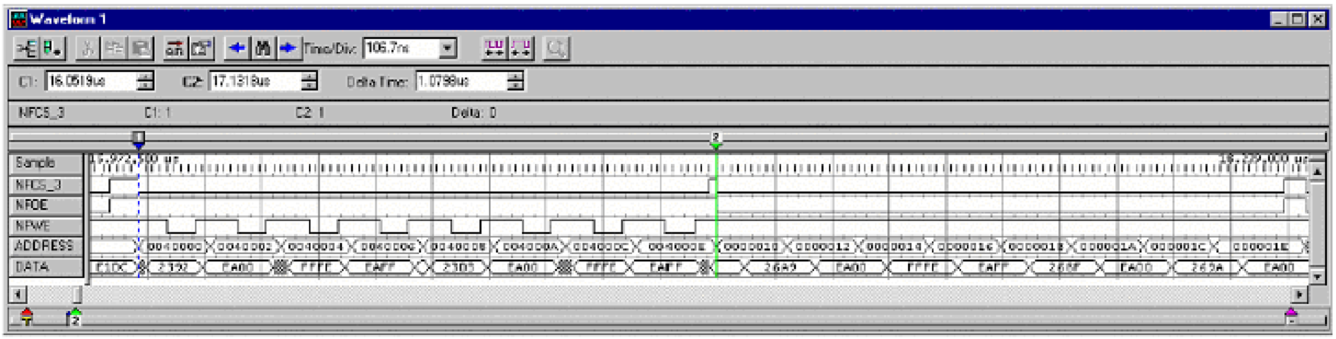
Time/Div: 106.7ns

C1: 14.9721us C2: 16.3998us Delta Time: 1.42765us

NFCS_3 C1: 1 C2: 1 Delta: 0

Figure 6. System DMA EMIFS Read (Burst of 4 by 32-bit)

A single System DMA EMIFS burst write of 4 by 32-bit is shown in Figure 7. This is implemented as a single 8 by 16-bit burst write by the EMIFS. Again, the behavior is similar to the 2 consecutive 4 by 16-bit variety, but takes advantage of the internal 32-bit bus width.



Waveform 1

Time/Div: 106.7ns

C1: 16.0519us C2: 17.1318us Delta Time: 1.0798us

NFCS_3 C1: 1 C2: 1 Delta: 0

Figure 7. System DMA EMIFS Write (Burst of 4 by 32-bit)

3.2 SDMA Transfers with EMIFF Destination

System DMA transfers with EMIFF SDRAM as the destination and various sources are shown in Table 7, Table 8, Table 9, and Table 10.

Table 7. IMIF Transfers to EMIFF: Latency, Throughput

		Latency (TC cycles)		**Average Throughput (TC cycles per 16-bit word)**		
IMIF Source	Src:0x20002000	First read		I-cache disabled		I-cache enabled
EMIFF Destination	Dest:0x10002000		First write	DMA Priority 0	DMA Priority 15	DMA priority 15
	20Kx16 no burst	Internal	101.9	4.1	4.1	4.1
	10Kx32 no burst	Internal	101.9	2.0	2.0	2.0
	20Kx16 burst 4	Internal	104.9	2.0	2.0	2.0
	10Kx32 burst 4	Internal	104.9	2.0	2.0	2.0

The use of 32-bit and/or bursting for IMIF to EMIFF transfers shows at least 2X improvement over non-burst 16-bit transfers. The initial latency for the first write increases slightly with the word size and the use of bursting.

Table 8. EMIFS Transfers to EMIFF: Latency, Throughput

		Latency (TC cycles)		**Average Throughput (TC cycles per 16-bit word)**		
EMIFS Source	Src:0x00000000	First read		I-cache disabled		I-cache enabled
EMIFF Destination	Dest:0x1000C000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	114.9	131.1	37.0	16.0	16.0
	10Kx32 no burst	114.9	140.9	23.5	13.0	13.0
	20Kx16 burst 4	114.9	200.9	13.4	13.4	13.4
	10Kx32 burst 4	114.9	200.9	13.4	13.4	13.4

The latency improvement trend continues for 32-bit, bursting, and dma priority. The initial write increases more significantly with word size and bursting for an EMIFS source than for an IMIF source.

Table 9. EMIFF Transfers to EMIFF: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
EMIFF Source	Src:0x10002000	First read		I-cache disabled		I-cache enabled
EMIFF Destination	Dest:0x1000C000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	96.9	125.1	17.1	17.1	17.1
	10Kx32 no burst	96.9	129.6	9.5	9.5	9.6
	20Kx16 burst 4	96.9	116.9	4.3	4.3	4.3
	10Kx32 burst 4	96.9	116.9	4.3	4.3	4.3

A 4X improvement is seen with the use of bursting. The first write is worse for a larger word size, but improves with bursting enabled.

Table 10. MPUI Transfers to EMIFF: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
MPUI Source	Src:0xE0000000	First read		I-cache disabled		I-cache enabled
EMIFF Destination	Dest:0x10002000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	Internal	112.4	13.0	13.0	13.0
	10Kx32 no burst	Internal	125.1	13.0	13.0	13.0
	20Kx16 burst 4	Internal	203.1	13.0	13.0	13.0
	10Kx32 burst 4	Internal	203.1	13.6	13.6	13.6

NOTE: MPUI bursts are not supported. Bursting is used on destination EMIFF only.

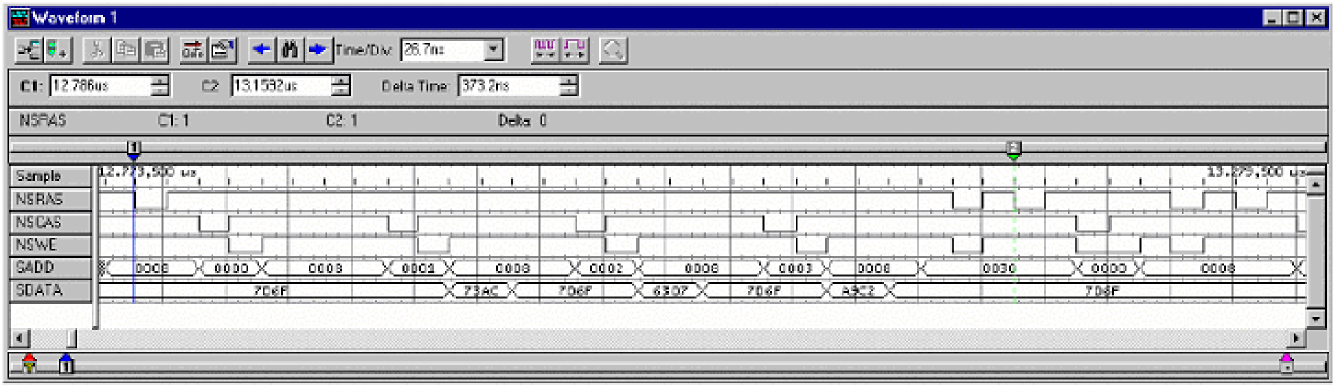
The MPUI reads are the limiting factor in this example. As in the MPUI to EMIFS example, the lack of bursting support on MPUI results in slightly worse throughput for bursting mode set on the EMIFF port.

3.2.1 Logic Analyzer Images of EMIFF to EMIFF System DMA Transfers

Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13, and Figure 14 are images captured using a logic analyzer with the system DMA configured for EMIFF to EMIFF transfers in various modes.

The EMIFF is clocked at the same rate as the TC (75 MHz). Beginning at cursor 1, in Figure 8, 4 consecutive SDMA EMIFF 16-bit non-burst reads are shown. These 4 reads occur on the same row of the SDRAM. The row is opened by the active NSRAS and the closing of the row is done with a precharge, NSRAS and NSWE active on the same TC cycle. Each read is initiated by the active NSCAS. Since only a single read is done on each access, this is considered as a non-burst. However, the EMIFF performs this as a SDRAM burst of 1 with a burst terminate or burst stop, NSWE active, on the TC cycle following the active NSCAS. In this example, the CAS latency is 2. This is shown with the data valid at 2 TC clocks after the NSCAS active.

At cursor 2, in Figure 8, a SDMA EMIFF 16-bit non-burst write follows the closing of the row on the previous reads. Similar to the 16-bit non-burst SDMA EMIFF reads, the EMIFF implements the non-burst as a SDRAM single burst with a burst terminate on the cycle immediately following the write. For more details on EMIFF cycle behavior, refer to Table D–1 in Appendix D.



Waveform 1

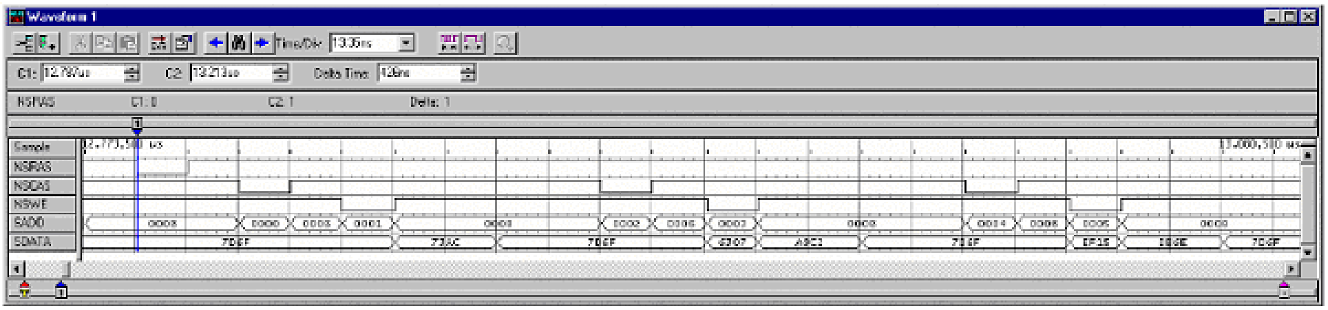
Time/Div: 26.7ns

C1: 12.786us C2: 13.1592us Delta Time: 373.2ns

NSRAS C1: 1 C2: 1 Delta: 0

Figure 8. System DMA EMIFF Reads and Write (No Burst, 16-bit)

Figure 9 shows 3 consecutive SDMA EMIFF 32-bit non-burst reads. These reads are implemented by the EMIFF as SDRAM dual burst 16-bit reads. Thus, throughput is improved by the elimination of TC cycles between the 2 consecutive 16-bit reads, which make up the 32-bit read.



Waveform 1

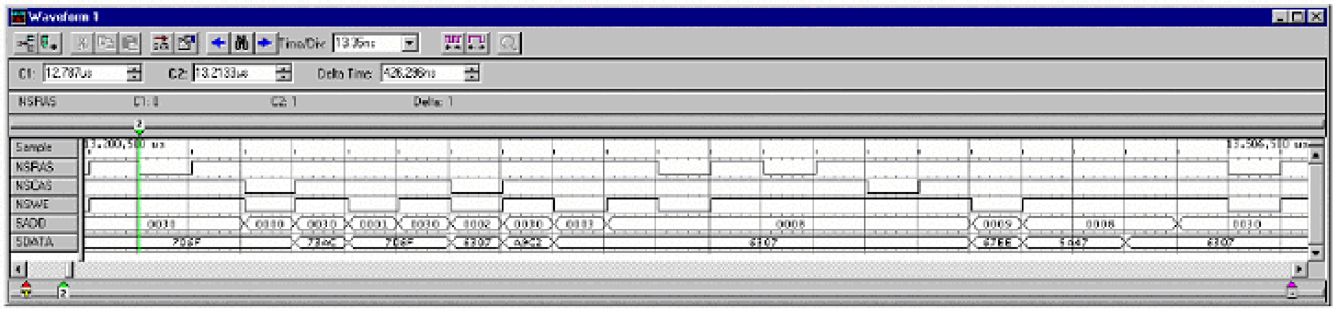
Time/Div: 13.35ns

C1: 12.787us C2: 13.213us Delta Time: 426ns

NSRAS C1: 0 C2: 1 Delta: 1

Figure 9. System DMA EMIFF Reads (No Burst, 32-bit)

At cursor 2, 2 consecutive SDMA EMIFF 32-bit non-burst writes are shown in Figure 10, followed by a single read. Similar to the 32-bit non-burst reads, the writes of the same mode are implemented by the EMIFF as SDRAM dual burst 16-bit writes. Thus, throughput is improved by the elimination of TC cycles between the 2 consecutive 16-bit reads, which make up each 32-bit write.



Waveform 1

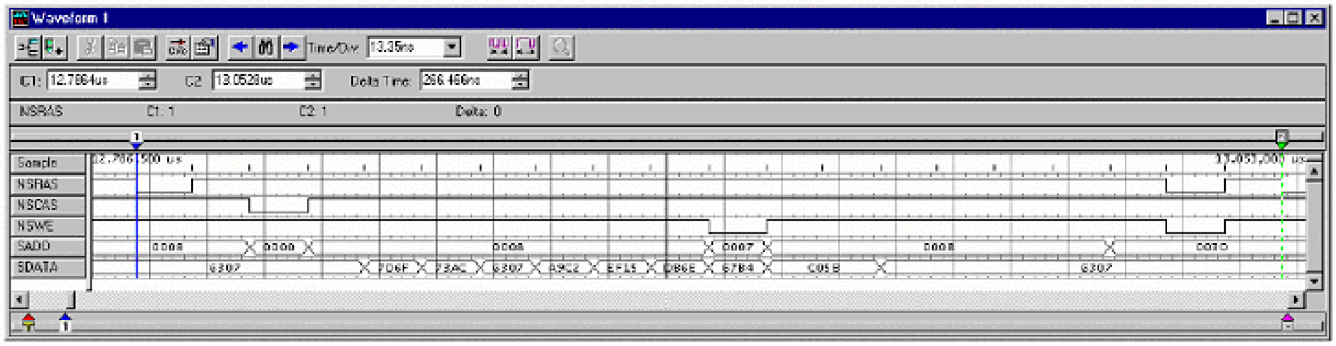
Time/Div: 13.35ns

C1: 12.787us C2: 13.2133us Delta Time: 426.296ns

NSRAS C1: 0 C2: 1 Delta: 1

Figure 10. System DMA EMIFF Writes (No Burst, 32-bit)

In Figure 11, 2 consecutive SDMA EMIFF bursts of 4 by 16-bit are shown. The EMIFF is configured for SDRAM bursts of 8 by 16-bit. This significantly improves the throughput over the previous non-burst examples.



Waveform 1

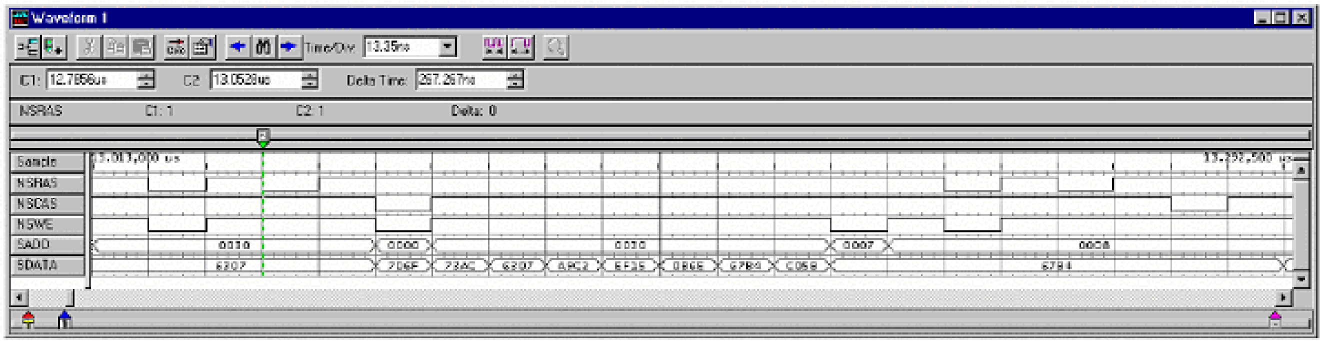
Time/Div: 13.35ns

C1: 12.7864us C2: 13.0528us Delta Time: 266.466ns

NSRAS C1: 1 C2: 1 Delta: 0

Figure 11. System DMA EMIFF Read (Burst of 4 x 16-bit)

Beginning at cursor 2, 2 consecutive SDMA EMIFF 4 by 16-bit write bursts are shown in Figure 12. As in the previous example for the SDRAM reads in the same mode, the EMIFF performs this as a single 8 by 16-bit burst. Again, the throughput is much improved over the non-burst mode.



Waveform 1

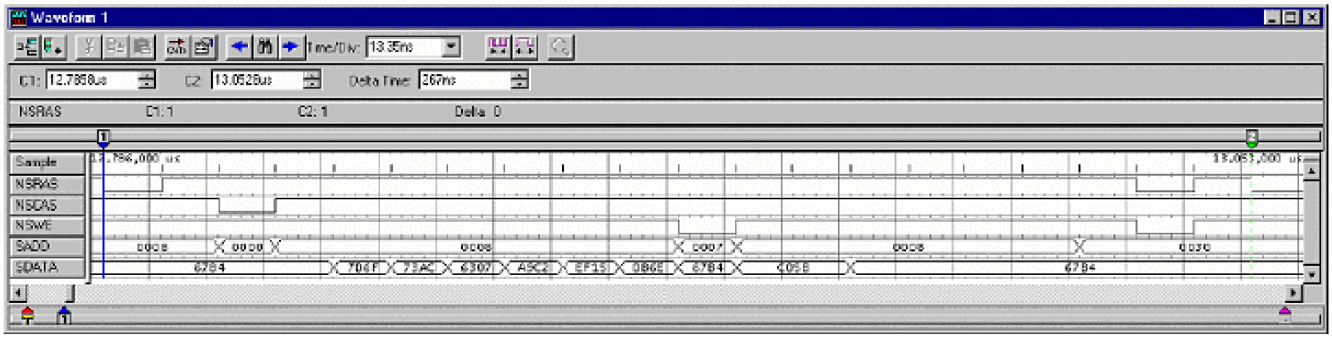
Time/Div: 13.35ns

C1: 12.7856us C2: 13.0528us Delta Time: 267.267ns

NSRAS C1: 1 C2: 1 Delta: 0

Figure 12. System DMA EMIFF Write (Burst of 4 x 16-bit)

A single SDMA EMIFF read burst of 4 by 32-bit is displayed in Figure 13. This is implemented as a SDRAM read burst of 8 by 16-bit. Since the external EMIFF bus is 16 bits, the single 4 by 32-bit burst read is performed in the same number of TC cycles as for the 2 consecutive 4 by 16-bit burst reads.



Waveform 1

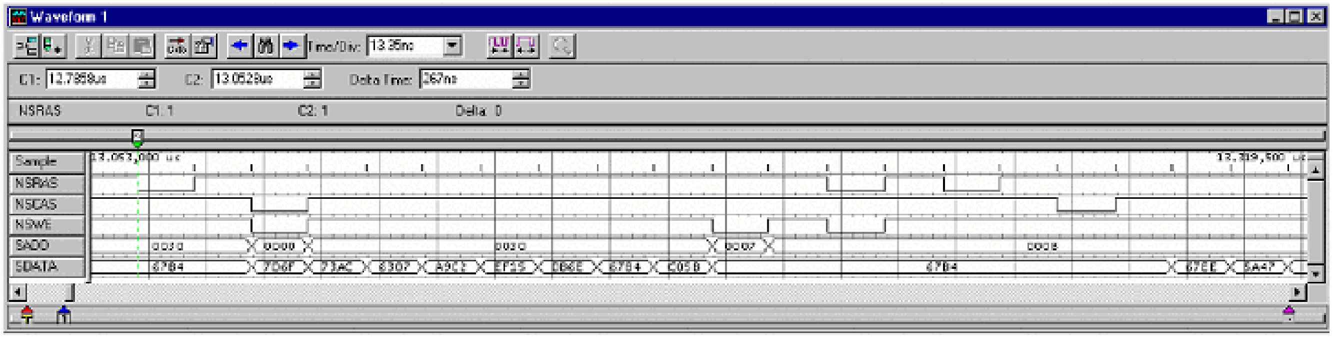
Time/Div: 13.35ns

C1: 12.7858us C2: 13.0528us Delta Time: 267ns

NSRAS C1: 1 C2: 1 Delta 0

Figure 13. System DMA EMIFF Read (Burst of 4 x 32-bit)

At cursor 2, a SDMA EMIFF write burst of 4 by 32-bit is shown in Figure 14, followed by the start of a read cycle. The EMIFF implements this as a SDRAM write burst of 8 by 16-bit.



Waveform 1

Time/Div: 13.35ns

C1: 12.7858us C2: 13.0528us Delta Time: 267ns

NSRAS C1: 1 C2: 1 Delta 0

Figure 14. System DMA EMIFF Write (Burst of 4 x 32-bit)

3.3 SDMA Transfers with IMIF Destination

System DMA transfers with IMIF internal SRAM as the destination and various sources are shown in Table 11, Table 12, Table 13, and Table 14.

Table 11. IMIF Transfers to IMIF: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
IMIF Source	Src:0x20002000	First read		I-cache disabled		I-cache enabled
IMIF Destination	Dest:0x20007000		First write	DMA Priority 0	DMA Priority 15	DMA priority 15
	20Kx16 no burst	Internal	Internal	4.0	4.0	4.0
	10Kx32 no burst	Internal	Internal	2.0	2.0	2.0
	20Kx16 burst 4	Internal	Internal	1.7	1.7	1.7
	10Kx32 burst 4	Internal	Internal	1.7	1.7	1.7

Usage of bursting and/or 32-bits for IMIF to IMIF transfers shows at least 2X improvement over non-burst 16-bit mode.

Table 12. EMIFS Transfers to IMIF: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
EMIFS Source	Src:0x00000000	First read		I-cache disabled		I-cache enabled
IMIF Destination	Dest:0x20002000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	114.9	Internal	37.0	16.0	16.0
	10Kx32 no burst	114.9	Internal	23.5	23.5	13.0
	20Kx16 burst 4	114.9	Internal	13.4	13.4	13.4
	10Kx32 burst 4	114.9	Internal	13.4	13.4	13.4

A nearly 3X latency improvement is given by the use of bursting. The initial reads are independent of word size and bursts for an EMIFS source to IMIF destination.

Table 13. EMIFF Transfers to IMIF: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
EMIFF Source	Src:0x10002000	First read		I-cache disabled		I-cache enabled
IMIF Destination	Dest:0x20002000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	96.9	Internal	6.1	6.1	6.1
	10Kx32 no burst	96.9	Internal	3.5	3.5	3.5
	20Kx16 burst 4	96.9	Internal	2.8	2.8	2.8
	10Kx32 burst 4	96.9	Internal	2.8	2.8	2.8

A greater than 2X improvement is seen with the use of bursting. Again, the initial read latency is independent of word size and bursting.

Table 14. MPUI Transfers to IMIF: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
MPUI Source	Src:0xE0000000	First read		I-cache disabled		I-cache enabled
IMIF Destination	Dest:0x10002000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	Internal	Internal	13.0	13.0	13.0
	10Kx32 no burst	Internal	Internal	13.0	13.0	13.0
	20Kx16 burst 4	Internal	Internal	13.0	13.0	13.0
	10Kx32 burst 4	Internal	Internal	13.6	13.6	13.6

NOTE: MPUI bursts are not supported. Bursting is used on destination IMIF only.

The transfers are dominated by the relatively slow MPUI source. Again, bursting with 32 bits is slightly worse than the other modes, when the source is MPUI.

3.4 SDMA Transfers with MPUI Destination

System DMA transfers with a destination of DSP SARAM through the MPUI and various sources are shown in Table 15, Table 16, Table 17, and Table 18.

Table 15. IMIF Transfers to MPUI: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
IMIF Source	Src:0x20002000	First read		I-cache disabled		I-cache enabled
MPUI Destination			First write	DMA Priority 0	DMA Priority 15	DMA priority 15
	20Kx16 no burst	Internal	Internal	13.0	13.0	13.0
	10Kx32 no burst	Internal	Internal	13.0	13.0	13.0
	20Kx16 burst 4	Internal	Internal	13.0	13.0	13.0
	10Kx32 burst 4	Internal	Internal	13.0	13.0	13.0

NOTE: MPUI bursts are not supported. Bursting is used on the source IMIF only.

IMIF to MPUI transfers are dominated by the MPUI writes.

Table 16. EMIFS Transfers to MPUI: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
EMIFS Source	Src:0x00000000	First read		I-cache disabled		I-cache enabled
MPUI Destination	Dest:0xE0014000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	114.9	Internal	37.0	16.0	16.0
	10Kx32 no burst	114.9	Internal	23.5	13.0	13.0
	20Kx16 burst 4	114.9	Internal	13.4	13.4	13.4
	10Kx32 burst 4	114.9	Internal	13.5	13.5	13.5

NOTE: MPUI bursts are not supported. Bursting is used on the source EMIFS only.

The initial reads are independent of word size and bursts for an EMIFS source to MPUI destination. A nearly 3X latency improvement is given by the use of bursting. Average EMIFS throughput for bursting in 32-bit mode is slightly worse than for 16-bit burst mode.

Table 17. EMIFF Transfers to MPUI: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
EMIFF Source	Src:0x10002000	First read		I-cache disabled		I-cache enabled
MPUI Destination	Dest:0xE0014000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	96.9	Internal	13.0	13.0	13.0
	10Kx32 no burst	96.9	Internal	13.0	13.0	13.0
	20Kx16 burst 4	96.9	Internal	13.0	13.0	13.0
	10Kx32 burst 4	96.9	Internal	13.0	13.0	13.0

NOTE: MPUI bursts are not supported. Bursting is used on the source EMIFF only.

The MPUI writes dominate the transfer throughput. The initial EMIFF read latency is independent of word size and bursting.

Table 18. MPUI Transfers to MPUI: Latency, Throughput

		Latency (TC cycles)		Average Throughput (TC cycles per 16-bit word)		
MPUI Source	Src:0xE0000000	First read		I-cache disabled		I-cache enabled
MPUI Destination	Dest:0xE0015000		First write	DMA Priority 0	DMA Priority 15	DMA Priority 15
	20Kx16 no burst	Internal	Internal	26.0	26.0	26.0
	10Kx32 no burst	Internal	Internal	26.0	26.0	26.0
	Bursting	N/A	N/A	N/A	N/A	N/A

NOTE: MPUI bursts are not supported.

System DMA transfers through MPUI have the same average throughput in either direction. The transfer latency is double that seen for transfers with MPUI as the limiting factor.

4 Conclusions

In general, for the greatest system DMA throughput, bursting should be used whenever possible. Use of I-cache is beneficial for tight code loops running within the same memory as other activity. Increasing the priority levels for a requestor of a particular TC memory can provide better throughput at the expense of other ports accessing the same memory. Furthermore, the choice of a particular memory type will have a major impact to throughput. In order from fastest memory to slowest memory, these are: IMIF, EMIFF, EMIFS, and MPUI (DSP memory).

5 References

1. *OMAP5910 Dual-Core Processor Data Manual* (SPRS197)
2. *OMAP5910 Dual-Core Processor Technical Reference Manual* (SPRU602)
3. Samsung Electronics http://www.samsungelectronics.com

Appendix A OMAP Block Diagram

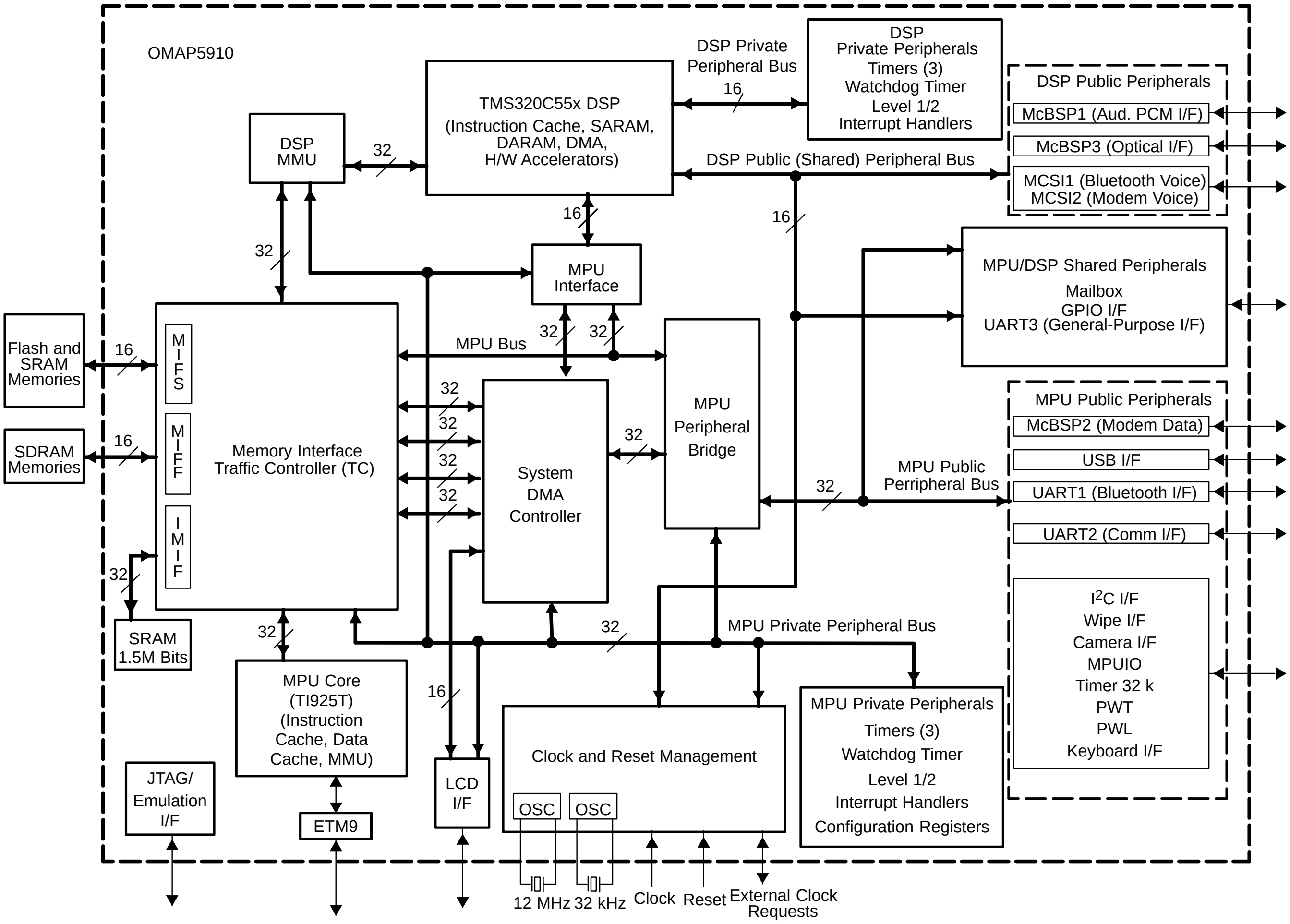


Figure A–1. OMAP Block Diagram

Appendix B MPU Memory Map

Table B–1. MPU Memory Map

Address Range	On-Chip	External Interface
0x0000 0000 0x01FF FFFF		FLASH CS0 32 Mbytes
0x0200 0000 0x03FF FFFF	reserved	
0x0400 0000 0x05FF FFFF		FLASH CS1 32 Mbytes
0x0600 0000 0x07FF FFFF	reserved	
0x0800 0000 0x09FF FFFF		FLASH CS2 32 Mbytes
0x0A00 0000 0x0BFF FFFF	reserved	
0x0C00 0000 0x0DFF FFFF		FLASH CS3 32 Mbytes
0x0E00 0000 0x0FFF FFFF	reserved	
0x1000 0000 0x13FF FFFF		SDRAM 64Mbytes
0x1400 0000 0x1FFF FFFF	reserved	
0x2000 0000 0x2002 FFFF	Internal SRAM 192Kbytes	
0x2003 0000 0x2FFF FFFF	reserved	
0x3000 0000 0x7FFF FFFF		Local bus space for USB host
0x8000 0000 0xDFFF FFFF	reserved	
0xE000 0000 0xE0FF FFFF	DSP public memory space (accessible by MPUI) 16Mbytes	
0xE100 0000 0xEFFF FFFF	DSP public peripherals (accessible by MPUI)	
0xFFFB 0000 0xFFFC FFFF	MPU public, MPU/DSP shared peripherals	
0xFFFD 0000 0xFFFE FFFF	MPU private peripherals	
0xFFFF 0000 0xFFFF FFFF	reserved	

Appendix C EMIFS Behavior

Table C–1. EMIFS Cycle Behavior with ARM I/D-Caches Disabled, EMIFS_PRIO = 0x00000000, and EMIFS_CS3_CONFIG.FCLKDIV=TC/2

EMIFS Cycle Behavior		
Consecutive Operations		**Minimum TC Cycles Between $\overline{\text{CS}}$ Active**
1st	**2nd**	
System DMA Read (16-bit, 32-bit, or burst)	System DMA Write (16-bit, 32-bit, or burst)	3
System DMA Read (16-bit, 32-bit, or burst)	ARM Program Fetch or Data Read (32-bit)	3
System DMA Write (16-bit, 32-bit or burst)	System DMA Read (16-bit, 32-bit, or burst)	1
System DMA Write (32-bit or burst)	ARM Program Fetch or Data Read (32-bit)	1
System DMA Read (32-bit or burst)	Consecutive 16-bit words within first operation	0
System DMA Write (32-bit or burst)	Consecutive 16-bit words within first operation	0
ARM Program Fetch or Data Read (32-bit)	System DMA Read (16-bit, 32-bit, or burst)	4
ARM Program Fetch or Data Read (32-bit)	System DMA Write (16-bit, 32-bit, or burst)	4

Table C–2. EMIFS /CS Active Widths for Asynchronous Reads/Writes

FCLKDIV	$\overline{\text{CS}}$ Active Width Read (TC Cycles)	$\overline{\text{CS}}$ Active Width Write (TC Cycles)
divide by 1	1*(RDWST+1)+1	1*(WRWST+WELEN+1)+2
divide by 2	2*(RDWST+1)+2	2*(WRWST+WELEN+1)+4
divide by 4	4*(RDWST+1)+4	4*(WRWST+WELEN+1)+8
divide by 6	6*(RDWST+1)+6	6*(WRWST+WELEN+1)+12

For this application report, the following apply:

- FCLKDIV = /2, RDWST = 3, WRWST = 1, WELEN = 1
- $\overline{\text{CS}}$ Active Width Read = 2*(RDWST+1)+2 = 2*(3+1)+2 = 10 TC cycles
- $\overline{\text{CS}}$ Active Width Write = 2*(WRWST+WELEN+1)+4 = 2*(1+1+1)+4 = 10 TC cycles

Appendix D EMIFF Behavior

Table D–1. EMIFF Behavior for Single and Burst Reads/Writes (TC Cycles)

	t_{RCD}	$t_{RAS}–t_{RCD}$		t_{RP}	t_{RRD}			
System DMA Data Type	**$\overline{\text{RAS}}$ to $\overline{\text{CAS}}$**	**$\overline{\text{CAS}}$ to Burst Terminate**	**Burst Terminate to Precharge**	**Precharge to Next $\overline{\text{RAS}}$**	**Total TC Cycles**	**TC Cycles per 16-bit Word**	**Precharge to Auto-Refresh**	**Auto-Refresh to $\overline{\text{RAS}}$**
16-bit read	2	1	5	2	10	10	4	5
32-bit read	2	2	5	2	11	5.5	4	5
16-bit read burst of 4	2	8 (EMIFF bursts 8x16)	8	2	20	2.5 (for transfers in multiples of 8x16)	4	5
32-bit read burst of 4	2	8 (EMIFF bursts 8x16)	8	2	20	2.5	4	5
16-bit write	2	1	5	2	10	10	4	5
32-bit write	2	2	5	2	11	5.5	4	5
16-bit write burst of 4	2	8 (EMIFF bursts 8x16)	2	2	14	1.75 (for transfers in multiples of 8x16)	4	5
32-bit write burst of 4	2	8 (EMIFF bursts 8x16)	2	2	14	1.75	4	5

Mailing Address:

Texas Instruments
Post Office Box 655303
Dallas, Texas 75265