AMCC
APPLIED MICRO CIRCUITS CORPORATION

Part Number 405GP

Revision 1.01 – November 30, 2004

405GP

Preliminary Application Note

Determining PowerPC® 405GP and 405CR DMA Performance for Transfers Between Peripherals and SDRAM

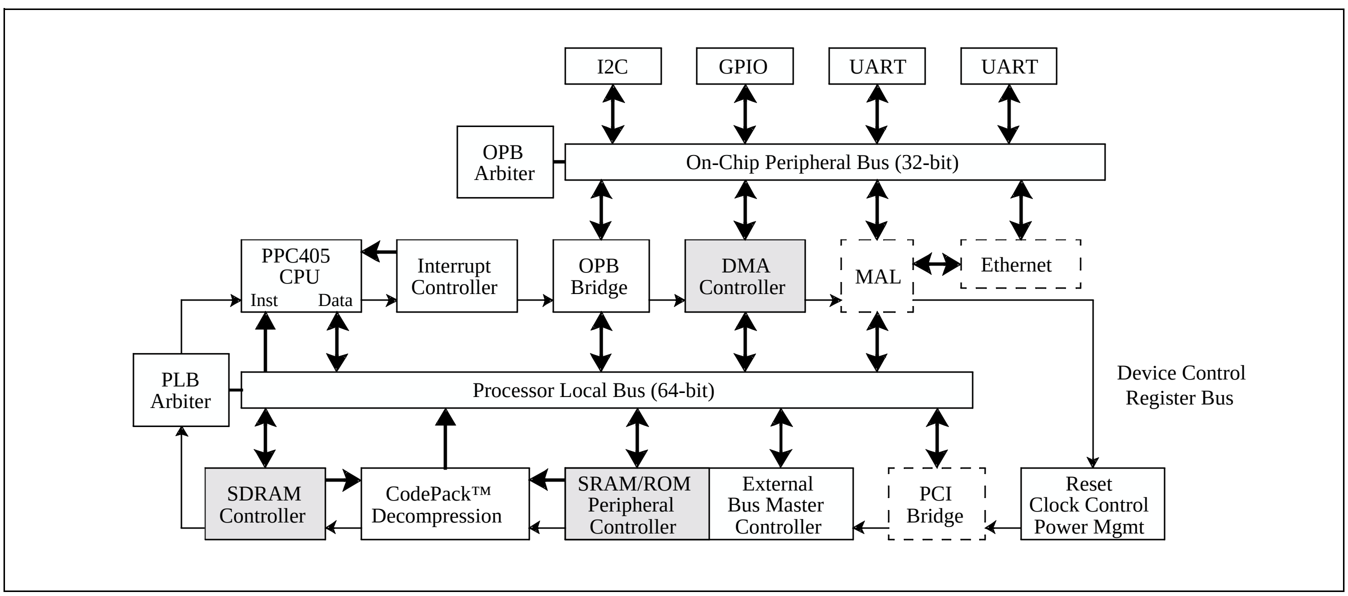
SCOPE

The highly integrated PowerPC 405GP and 405CR embedded controllers consist of a PowerPC processor core and peripherals including a DMA and SDRAM controller. This application note provides a method for estimating the data transfer rate between an external DMA peripheral and SDRAM memory.

As illustrated in Figure 1, the 405GP embedded controller consists of logical units interconnected via three buses: the Processor Local Bus (PLB), the On-Chip Peripheral Bus, and the Device Control Register Bus. These buses form the IBM CoreConnect™ architecture and enable more efficient design and integration of system-on-a-chip devices. By designing and testing macros to CoreConnect specifications, product development cycle times be can reduced and macro interoperability is guaranteed by design.

The 405CR differs from the 405GP in that it does not include the PCI Bridge, MAL, and Ethernet controllers. For both of these chips the SDRAM, DMA, and peripheral controllers reside on the high speed PLB. While the DMA and SDRAM controllers operate at the PLB clock rate, the peripheral controller is configurable for speeds of 1/2, 1/3, 1/4, or 1/5 of the PLB rate. Many factors, including these different clock ratios and configurable data buffering within the DMA controller affect the DMA transfer rate between SDRAM and DMA peripherals. Through several tables and simple equations this application note provides a means of accurately estimating the DMA performance for particular configurations.

Figure 1. PPC405GP Embedded Controller Block Diagram

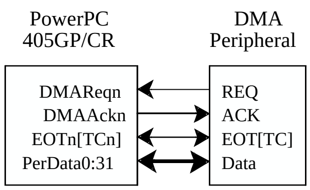


PERIPHERAL MODE DMA TRANSFERS

While the DMA controller supports data transfers from memory-to-memory or between memory and peripheral, this application note applies only to the latter. In the context of the DMA controller, a peripheral transfer is defined as one where the DMA acknowledge signal, DMAAckn, serves as the data strobe for the peripheral side of the transfer.

Figure 2 illustrates the typical method for attaching a peripheral mode DMA device to a 405GP or 405CR. Observe that no address or chip select is required.

Figure 2. Peripheral Mode DMA Wiring



Although the DMA interface supports asynchronous transfers, all timing is defined with respect to PerClk, the External Bus Controller (EBC) clock. Given the ratio between the SDRAM clock and PerClk, the CAS latency for the SDRAM memory, and the programming in the DMA Channel Control Register, DMA0_CRn, it is possible to determine the maximum transfer rate for a DMA channel. Table 1 lists the fields in DMA0_CRn that affect the DMA transfer rate.

Table 1. DMA Channel Control Register Fields Affecting Peripheral Mode DMA Performance

DMA0_CRn Field	Usage	Value	Description
TD	Transfer Direction	0	Transfer is from memory to peripheral
		1	Transfer is from peripheral to memory
PW	Peripheral Width	0b00	Byte (8-bits)
		0b01	Halfword (16-bits)
		0b10	Word (32-bits)
BEN	Buffer Enable	0	Disable 32-byte DMA buffer
		1	Enable 32-byte DMA buffer
PSC	Peripheral Setup Cycles	0-3	Number of PerClk cycles the peripheral bus is idle before DMAAckn becomes active
PWC	Peripheral Wait Cycles	0-63	DMAAckn is active for 1+PWC PerClk cycles
PHC	Peripheral Hold Cycles	0-7	Number of PerClk cycles the peripheral bus is idle after DMAAckn is inactive
PF	Memory Read Prefetch	0b00	Prefetch 1 doubleword (64-bits)
		0b01	Prefetch 2 doublewords
		0b10	Prefetch 4 doublewords

Please note that transfer rates calculated by the method presented in this application note represent an upper limit on performance since:

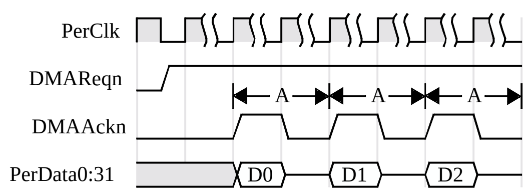
- It is assumed that the DMA peripheral continuously drives its DMA request line and that the channel is not interrupted by a higher priority channel.
- The effects of SDRAM refreshes and page misses are excluded.
- It is further assumed that there is no non-DMA activity on the peripheral bus and that PLB transactions between other masters and slaves do not delay the movement of data between the DMA controller and SDRAM memory.

Since the preceding conditions are not present in most systems, designers should apply an appropriate derating factor for their application.

TRANSFER RATE WITH DMA BUFFER DISABLED

The DMA controller includes a 32-byte buffer that serves to improve performance by reducing the number of discrete memory accesses. With the buffer disabled the DMA controller issues a separate memory operation to the SDRAM controller for each individual DMA data item. As Figure 3 illustrates, the timing on the DMA interface[1] is constant between these data items.

Figure 3. Peripheral Mode DMA Transfer Timings when DMA0_CRn[BEN]=0



1.Although DMAReqn and DMAAckn are shown as active high, their polarity is programmable via the DMA0_POL register. For additional details, see either the PPC405GP or PPC405CR User’s Manual.

The number of PerClk cycles from one DMA acknowledge to the next, A, depends on the Peripheral Setup, Peripheral Wait and Peripheral Hold values programmed in the DMA Channel Control Register plus the time for the SDRAM access:

A DMA0_CRn[PSC] 1 DMA0_CRn[PWC] +( )DMA0_CRn[PHC] + Value from Table 2.

Table 2. Additional Cycles of Delay when DMA0_CRn[BEN]=0

Transfer Direction DMA0_CRn[TD]	SDRAM CAS Latency SDRAM0_TR[CASL]	Hold Cycles DMA0_CRn[PHC]	Additional PerClk Cycles at SDRAM:EBC Clock Ratio of			
			2:1	3:1	4:1	5:1
Memory to Peripheral	2	0	10	7	5	4
		>0	9	6	5	4
	3	0	11	7	6	5
		>0	10	7	5	4
Peripheral to Memory	—	0	7	5	4	3
		1	7	4	3	3
		>1	6	4	3	3

As an example, consider a system with the following parameters:

- SDRAM at 100 MHz with a CAS latency of 2 or 3 cycles
- External peripheral bus clocked at 50 MHz (SDRAM:EBC ratio of 2:1)
- DMA channel configured for peripheral to memory transfers with zero setup cycles, zero wait states, 1 hold cycle, and DMA buffering disabled.

The time from one DMA acknowledge to the next is:

A – 0 + (1 + 0) + 1 + 7 = 9 PerClk cycles.

If the DMA channel is configured to transfer 32-bit words, the transfer rate is:

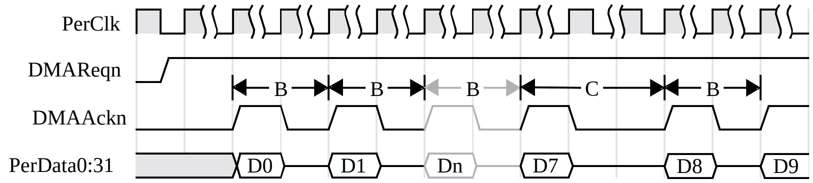
$$\frac{4\text{ bytes}}{\text{transfer}} \bullet \frac{1\text{ transfer}}{9\text{ cycles}} \bullet \frac{50\text{M cycles}}{\text{s}} = 22.22\text{ MB/s.}$$

TRANSFER RATE WITH DMA BUFFER ENABLED

Enabling the DMA 32-byte buffer greatly improves throughput in most systems. Instead of each DMA acknowledge cycle causing a corresponding SDRAM memory access, the buffer serves as an intermediate stopping point for data. In the case of peripheral to memory transfers, up to 32 bytes of data are collected in the buffer and then written to SDRAM in a single transaction. For memory to peripheral transfers the number of 64-bit doublewords loaded into the buffer is programmed through the DMA0_CRn[PF] bit field. For the best transfer rate the DMA controller should be set to prefetch four doublewords from SDRAM. Smaller prefetch counts are recommended only in cases where the DMA channel can be interrupted by another channel of higher priority. This is because the DMA controller has only a single 32-byte buffer that is flushed whenever one channel is interrupted by another. The DMA buffer is also flushed whenever the active channel deasserts its DMAReqn.

Since the DMA controller must periodically access SDRAM, the timing profile is not constant. As shown in Figure 4, there is a longer delay between DMA acknowledges when the memory access occurs.

Figure 4. Peripheral Mode DMA Transfer Timings when DMA0_CRn[BEN]=1



Assume the waveform in Figure 4 is a memory to peripheral transfer. Then, words D0, D1, …, D7 are buffered before being written to SDRAM. The number of PerClk cycles between DMA acknowledges for words D0 through D7 is:

B =DMA0_CRn[PSC] + 1 +DMA0_CRn[PWC] +D MA0_CRn[PHC] +Value from Table 3.

Table 3. Additional Cycles of Delay for DMA Transfers to or from the DMA Buffer

Transfer Direction DMA0_CRn[TD]	Hold Cycles DRAM0_CRn[PHC]	Additional PerClk Cycles at SDRAM:EBC Clock Ratio of			
		2:1	**3:1**	**4:1**	**5:1**
Memory to Peripheral	—	3	2	2	2
Peripheral to Memory	0	4	3	2	2
	>0	3	2	2	1

After the buffer becomes full, or empty for memory to peripheral transfers, the DMA controller must perform an SDRAM memory access. The number of PerClk cycles between DMA acknowledges whenever a memory access occurs is:

C =DMA0_CRn[PSC] + 1 +DMA0_CRn[PWC] +DMA0_CRn[PHC] +Value from Table 4.

For a peripheral to memory transfer, the width of the DMA peripheral determines the number of data items that fit in the 32-byte DMA buffer. For example, if DMA0_CRn[PW]=0b01, the DMA controller reads 16-bit halfwords and stores them in the DMA buffer. After accumulating 16 halfwords the full buffer is written to SDRAM. The following equation accounts for the peripheral width and quantifies the number of cycles to move 32 bytes of data from the DMA peripheral to SDRAM memory:

$(2^{5 - \mathrm{DMA0\_CRn[PW]}} - 1) \bullet B + C$ PerClk cycles.

Table 4. Additional Cycles of Delay for DMA Transfers to or from the DMA Buffer

Transfer Direction DMA0_CRn[TD]	Prefetch Count DMA0_CRn[PF]	SDRAM CAS Latency SDRAM0_TR[CASL]	Hold Cycles DMA0_CRn[PHC]	Additional PerClk Cycles at SDRAM:EBC Clock Ratio of			
				2:1	**3:1**	**4:1**	**5:1**
Memory to Peripheral	1	2	0	11	7	6	5
			>0	10	7	5	4
		3	0	11	8	6	5
			>0	10	7	5	4
	2	2	0	12	8	6	5
			>0	11	7	6	5
		3	0	12	8	6	5
			>0	11	8	6	5
	4	2	0	14	9	7	6
			>0	13	9	7	5
		3	0	14	10	7	6
			>0	13	9	7	6
Peripheral to Memory	—	—	0	9	6	5	4
			>0	8	5	4	3

Since the number of 64-bit doublewords prefetched from SDRAM during a memory to peripheral transfer is configurable, only a portion of the 32-byte buffer may be in use. Accounting for the amount of the DMA buffer in use, the number of cycles to load the buffer from SDRAM and transfer the contents to the DMA peripheral is:

$$(2^{3 + DMA0\_CRn[PF]\ DMA0\_CRn[PW]} - + 1) \bullet B + C \text{ PerClk cycles.}$$

To illustrate the performance of a DMA transfer using the 32-byte buffer, consider an application with:

- SDRAM at 100 MHz with a CAS latency of 2 or 3 cycles
- External peripheral bus PerClk of 50 MHz (SDRAM:EBC ratio of 2:1)
- DMA channel configured for word (32-bit) peripheral to memory transfers with no setup cycles, zero wait states, and 1 hold cycle

The time from one DMA acknowledge to the next while writing into the DMA buffer is:

$$B = 0 + (1 + 0) + 1 + 3 = 5 \text{ PerClk cycles.}$$

After the last item is read into the DMA buffer, the time to the next DMA acknowledge is:

$$C = 0 + (1 + 0) + 1 + 8 = 10 \text{ PerClk cycles.}$$

Therefore, the total number of cycles to transfer one buffer, eight words, from peripheral to SDRAM is:

$$(2^{5-2} - 1) \bullet 5 + 10 = 7 \bullet 5 + 10 = 45 \text{ PerClk cycles.}$$

The available bandwidth for this configuration is then:

$$\frac{32 \text{ bytes}}{\text{buffer}} \bullet \frac{1 \text{ buffer}}{45 \text{ cycles}} \bullet \frac{50\text{M cycles}}{\text{s}} = 35.56 \text{ MB/s.}$$

CONCLUSION

The methods presented in this application note provide designers with an algorithm for estimating the performance of peripheral mode DMA transfers that target SDRAM memory. Since the results presented here do not include system-level effects such as other bus activity and SDRAM refreshes an appropriate derating factor should be applied. As an aid to designers, Table 5 lists the DMA transfer rates for a DMA peripheral configured to prefetch four doublewords from SDRAM and with zero setup cycles, zero wait states, and one hold cycle.

Table 5. DMA Transfer Rates (MB/s) for SDRAM at 100 MHz

			Peripheral Transfer Width					
			Byte		Halfword		Word	
			SDRAM:EBC Clock Ratio					
Transfer Direction	**DMA Buffer**	**SDRAM CAS Latency**	**2:1**	**3:1**	**2:1**	**3:1**	**2:1**	**3:1**
Memory to Peripheral	Disabled	2	4.55	4.17	9.09	8.33	18.18	16.67
		3	4.17	3.70	8.33	7.41	16.67	14.81
	Enabled	—	9.41	7.90	17.78	15.02	32.00	27.35
Peripheral to Memory	Disabled	—	5.56	5.56	11.11	11.11	22.22	22.22
	Enabled	—	9.70	8.14	18.82	15.92	35.56	30.48

DOCUMENT REVISION HISTORY

Revision Date	Contents of Modification
November 30, 2004	Converted document to AMCC format. Non of the content has changed.
August 5, 2002	Second revision (02). Converted application note to IBM Microelectronics standard format. Previous versions of this application note included only the buffer load/empty time in the Table 4 values. As shown in Figure 4, the value C also includes the time to transfer the last data element. Corrected the values in Table 4 to account for the overhead required to transfer the last data item and updated Table 5 to reflect these changes. Modified all equations to explicitly account for the one PerClk cycle of DMAAckn active time always present during a DMA peripheral transfer. To offset this change all entries in Table 2, Table 3, and Table 4 were reduced by one PerClk cycle.
August 14, 2000	First revision (1.0). Made minor wording changes so that the application note applies to both the PowerPC 405GP and 405CR. Modified Figure 1 to show functional units not present in the 405CR with dashed borders. Changed the DMA and SDRAM controller register names to match the RISCWatch™ register naming convention adopted for the PowerPC 405 series of embedded controllers. Fixed incorrect data word numbering in Figure 4.

Applied Micro Circuits Corporation
6290 Sequence Dr., San Diego, CA 92121

Phone: (858) 450-9333 — (800) 755-2622 — Fax: (858) 450-9885

http://www.amcc.com