Direct Memory Access Controller (Type 1)

© Copyright IBM Corp. 1990

Figures

© Copyright IBM Corp. 1990

Description

The Direct Memory Access (DMA) controller allows I/O devices to transfer data directly to and from memory. This frees the system microprocessor of I/O tasks, resulting in a higher throughput.

The DMA controller is software programmable. The system microprocessor can address the DMA controller and read or modify the internal registers to define the various DMA modes, transfer addresses, transfer counts, channel masks, and page registers.

The functions of the DMA controller can be grouped into two categories: program mode and DMA transfer mode.

In the program mode, the system microprocessor accesses the DMA controller within the specific address range. These addresses are identified in Figure 1 on page 3. In this mode, the DMA registers can be read from or written to.

In the DMA mode, the DMA controller performs the data transfer. The transfer is initiated when a DMA slave has won the arbitration bus and the DMA controller has been programmed to service the winning request in process. Data transfers can be a single-byte transfer, or multiple-byte transfers (burst).

Deactivation of CD CHRDY by a device can extend accesses for slower I/O or memory devices.

The DMA controller supports the following:

- Register and program compatibility with the IBM Personal Computer AT® DMA channels (8237 compatible mode)
- 16MB (MB equals 1,048,576 bytes) 24-bit address capability for memory and 64KB (KB equals 1024 bytes) 16-bit address capability for I/O
- Eight independent DMA channels capable of transferring data between memory and I/O devices
- DMA operation with a separate read and write cycle for each transfer operation
- Channel programmable for byte or word transfer
- Extended operations:

Personal Computer AT is a registered trademark of the International Business Machines Corporation.

  - Extended program control
  - Extended Mode register
- 8- and 16-bit DMA slaves only
- Programmable arbitration levels for two channels.

DMA Controller Operations

The DMA controller does two types of operations:

- Data transfers between memory and I/O devices
- Read verifications.

Data Transfers between Memory and I/O Devices

The DMA controller performs serial transfers for all read and write operations. These transfers can be between memory and I/O on any channel. Data is read from a device and latched in the DMA controller before it is written back to a second device. The memory address needs to be specified only for a DMA data transfer. A programmable 16-bit I/O address can be provided during the I/O portion of the transfer as a programmable option. If the programmable 16-bit I/O address is not selected, the I/O address is forced to hex 0000 during the I/O transfer.

Read Verifications

The DMA controller can do a memory-read operation without a transfer. The address and the count are updated, and the terminal count is provided.

DMA I/O Address Map

Address (Hex)	Description	Bit Description	Byte Pointer
0000	Channel 0, Memory Address Register	00-15	Used
0001	Channel 0, Transfer Count Register	00-15	Used
0002	Channel 1, Memory Address Register	00-15	Used
0003	Channel 1, Transfer Count Register	00-15	Used
0004	Channel 2, Memory Address Register	00-15	Used
0005	Channel 2, Transfer Count Register	00-15	Used
0006	Channel 3, Memory Address Register	00-15	Used
0007	Channel 3, Transfer Count Register	00-15	Used
0008	Channel 0-3, Status Register	00-07	
000A	Channel 0-3, Mask Register (Set/Reset)	00-02	
000B	Channel 0-3, Mode Register (Write)	00-07	
000C	Clear Byte Pointer (Write)	N/A	
000D	DMA Controller Reset (Write)	N/A	
000E	Channel 0-3, Clear Mask Register (Write)	N/A	
000F	Channel 0-3, Write Mask Register	00-03	
0018	Extended Function Register (Write)	00-07	
001A	Extended Function Execute	00-07	Used *
0081	Channel 2, Page Table Address Register **	00-07	
0082	Channel 3, Page Table Address Register **	00-07	
0083	Channel 1, Page Table Address Register **	00-07	
0087	Channel 0, Page Table Address Register **	00-07	
0089	Channel 6, Page Table Address Register **	00-07	
008A	Channel 7, Page Table Address Register **	00-07	
008B	Channel 5, Page Table Address Register **	00-07	
008F	Channel 4, Page Table Address Register **	00-07	
00C0	Channel 4, Memory Address Register	00-15	Used
00C2	Channel 4, Transfer Count Register	00-15	Used
00C4	Channel 5, Memory Address Register	00-15	Used
00C6	Channel 5, Transfer Count Register	00-15	Used
00C8	Channel 6, Memory Address Register	00-15	Used
00CA	Channel 6, Transfer Count Register	00-15	Used
00CC	Channel 7, Memory Address Register	00-15	Used
00CE	Channel 7, Transfer Count Register	00-15	Used
00D0	Channel 4-7, Status Register	00-07	
00D4	Channel 4-7, Mask Register (Set/Reset)	00-02	
00D6	Channel 4-7, Mode Register (Write)	00-07	
00D8	Clear Byte Pointer (Write)	N/A	
00DA	DMA Controller Reset (Write)	N/A	
00DC	Channel 4-7, Clear Mask Register (Write)	N/A	
00DE	Channel 4-7, Write Mask Register	00-03	

Note: * Used only during extended functions, see "Extended Commands" on page 10. ** Upper byte of memory address register.

Figure 1. DMA I/O Address Map

Byte Pointer

A byte pointer gives 8-bit ports access to consecutive bytes of registers greater than 8 bits. For program I/O, the registers that use it are the Memory Address registers (3 bytes), the Transfer Count registers (2 bytes), and the I/O Address registers (2 bytes). Interrupts should be masked off while programming DMA controller operations.

DMA Registers

All system microprocessor access to the DMA controller must be 8-bit I/O instructions. The following figure lists the names and sizes of the DMA registers.

Register	Size (Bits)	Quantity of Registers	Allocation
Memory Address	24	8	1 per Channel
I/O Address	16	8	1 per Channel
Transfer Count	16	8	1 per Channel
Temporary Holding	16	1	All Channels
Mask	4	2	1 for Channels 7 - 4 1 for Channels 3 - 0
Arbus	4	2	1 for Channel 4 1 for Channel 0
Mode	8	8	1 per Channel
Status	8	2	1 for Channels 7 - 4 1 for Channels 3 - 0
Function	8	1	All Channels
Refresh	9	1	Independent of DMA

Figure 2. DMA Registers

Memory Address Register

Each channel has a 24-bit Memory Address register, which is loaded by the system microprocessor. The Mode register determines whether the address is incremented or decremented. The Mode register can be read by the system microprocessor in successive I/O byte operations. To read this register, the microprocessor must use the extended DMA commands.

I/O Address Register

Each channel has a 16-bit I/O Address register, which is loaded by the system microprocessor. The bits in this register do not change during DMA transfers. This register can be read by the system microprocessor in successive I/O byte operations. To read this register, the microprocessor must use the extended DMA commands.

Typically, a DMA slave is selected for DMA transfers by a decode of the arbitration level, status (-S0 exclusively ORed with -S1), and M/-IO. In this case, the respective I/O address register must have a value of 0.

A DMA slave can be selected based on a decode of the address rather than the arbitration level. In this case, the respective I/O address register must have the proper I/O address value.

Transfer Count Register

Each channel has a 16-bit Transfer Count register, which is loaded by the system microprocessor. The transfer count determines how many transfers the DMA channel will execute before reaching the terminal count. The number of transfers is always 1 more than the count specifies. If the count is 0, the DMA controller does one transfer. This register can be read by the system microprocessor in successive I/O byte operations. To read this register, the system microprocessor can use only the extended DMA commands.

Temporary Holding Register

This 16-bit register holds the intermediate value for the serial DMA transfer taking place. A DMA operation requires the data to be held in the register before it is written back. This register is not accessible by the system microprocessor.

Mask Register

Bit	Function
7 - 3	Reserved = 0
2	0 Clear Mask Bit 1 Set Mask Bit
1, 0	00 Select Channel 0 or 4 01 Select Channel 1 or 5 10 Select Channel 2 or 6 11 Select Channel 3 or 7

Figure 3. Set/Clear Single Mask Bit Using 8237 Compatible Mode

Bit	Function
7 - 4	Reserved = 0
3	0 Clear Channel 3 or 7 Mask Bit 1 Set Channel 3 or 7 Mask Bit
2	0 Clear Channel 2 or 6 Mask Bit 1 Set Channel 2 or 6 Mask Bit
1	0 Clear Channel 1 or 5 Mask Bit 1 Set Channel 1 or 5 Mask Bit
0	0 Clear Channel 0 or 4 Mask Bit 1 Set Channel 0 or 4 Mask Bit

Figure 4. DMA Mask Register Write Using 8237 Compatible Mode

Each channel has a corresponding mask bit that, when set, disables the DMA from servicing the requesting device. Each mask bit can be set to 0 or 1 by the system microprocessor. A system reset or DMA Controller Reset command sets all mask bits to 1. A Clear Mask Register command sets mask bits 0 - 3 or mask bits 4 - 7 to 0.

When a device requesting DMA cycles wins the arbitration cycle, and the mask bit is set to 1 on the corresponding channel, the DMA controller does not execute any cycles in its behalf and allows external devices to provide the transfer. If no device responds, the bus times out and causes a nonmaskable interrupt (NMI). This register can be programmed using the 8237 compatible mode commands (used by the IBM Personal Computer AT) or the extended DMA commands.

Mode Register

The Mode register for each channel identifies the type of operation that takes place when that channel transfers data.

Bit	Function
7, 6	Reserved = 0
5, 4	Reserved = 0
3, 2	00 Verify Operation 01 Write Operation 10 Read Operation 11 Reserved
1, 0	Channel Accessed 00 Select Channel 0 or 4 01 Select Channel 1 or 5 10 Select Channel 2 or 6 11 Select Channel 3 or 7

Figure 5. 8237 Compatible Mode Register

The Mode register is programmed by the system microprocessor, and its contents are reformatted and stored internally in the DMA controller. In the 8237 compatible mode, this register can only be written.

Extended Mode Register

Besides the 8237 compatible mode, all channels support an 8-bit Extended Mode register. The Extended Mode register can be programmed and read by the system microprocessor.

The DMA controller supports an Extended Mode register for each channel that can be programmed and read by the system microprocessor. This register is used whenever a DMA channel requests a DMA data transfer.

The DMA channel must be programmed to match the transfer size of the DMA slave on the channel. Bit 6 of this register is used to program the size of the DMA transfer.

Bit	Function
7	Reserved = 0
6	0 = 8-Bit Transfer 1 = 16-Bit Transfer
5	Reserved = 0
4	Reserved = 0
3	0 = Read Memory Transfer 1 = Write Memory Transfer
2	0 = Read Verifications Operation 1 = Data Transfer Operation
1	Reserved = 0
0	0 = I/O Address equals 0000H 1 = Use programmed I/O Address

Figure 6. Extended Mode Register

Status Register

The Status register, which can be read by the system microprocessor, contains information about the status of the devices. This information tells which channels have reached the terminal count and which channels have requested the bus since the last time the register was read.

Bit	Function
7	Channel 3 or 7 Request
6	Channel 2 or 6 Request
5	Channel 1 or 5 Request
4	Channel 0 or 4 Request
3	TC on Channel 3 or 7
2	TC on Channel 2 or 6
1	TC on Channel 1 or 5
0	TC on Channel 0 or 4

Figure 7. Status Register

Bits 3 through 0 in each Status register are set every time a terminal count is reached by a corresponding channel. Bits 7 through 4 are set when a corresponding arbitration level has controlled the bus. All bits are cleared by a system reset or following a system microprocessor Status Read command. This register can be read using the 8237 commands or extended DMA commands.

DMA Extended Function Register (Hex 0018)

This 8-bit register minimizes I/O address requirements and provides the extended program functions. The system microprocessor loads this register using I/O write operations. See "Extended Commands" on page 10 for more information.

Bit	Function
7 - 4	Program Command (DMA Extended Commands)
3	Reserved = 0
2 - 0	Channel Number (0 through 7)

Figure 8. DMA Extended Function Register (Hex 0018)

Arbus Register

This register is used for virtual DMA operations.

Bit	Function
7 - 4	Reserved
3 - 0	Arbitration Level

Figure 9. Arbus Register

Virtual DMA channel operation permits programming of the arbitration level assignment for channels 0 and 4 using the two 4-bit Arbus registers. These registers enable the system microprocessor to dynamically reassign the arbitration ID value by which the DMA controller responds to bus arbitration for DMA requests. This allows channels 0 and 4 to service devices at any arbitration level. The value of arbitration level hex F is reserved.

DMA Extended Operations

The function register supports an extended set of commands for the DMA channels. The extended command hex 8 programs the Arbus registers; the upper 4 bits of the Extended Function register are set to a value of 8 to select the Arbus register, and the lower 4 bits are set to the channel number (0 or 4). If channel 0 = 1 - 3 or 5 - 7 then channel 0 is active; if channel 0 = 4 then channel 0 is inactive. The system microprocessor uses the following addresses to gain control of the internal DMA registers.

I/O Address (Hex)	Command
0018	Write Extended Function Register
0019	Reserved
001A	Execute Extended Function Register
001B	Reserved

Figure 10. DMA Extended Address Decode

The system microprocessor uses the following steps to write to or read from any of the DMA internal registers:

1. Write to the Extended Function register by executing an I/O Write instruction to address hex 0018, with the proper data to indicate the function and the channel number. The internal byte pointer is always reset to 0 when an I/O write to address hex 0018 is detected.
2. Execute the Extended Function command by doing an I/O Read or I/O Write instruction to address hex 001A. The byte pointer automatically increments and points to the next byte each time port address hex 001A is used. This step is not required for Direct commands because they are executed when the Out command to address hex 0018 is detected.

Extended Commands

The following figure shows the available extended command set contained in the Extended Function register.

Registers/Bits Accessed	Bits	Extended Command (Hex) (7-4 *)	Byte Pointer
I/O Address Register	00-15	0	Used
Reserved		1	
Memory Address Register Write	00-23	2	Used
Memory Address Register Read	00-23	3	Used
Transfer Count Register Write	00-15	4	Used
Transfer Count Register Read	00-15	5	Used
Status Register Read	00-07	6	
Mode Register	00-07	7	
Arbus Register	00-07	8	
Mask Register Set Single Bit **		9	
Mask Register Reset Single Bit **		A	
Reserved		B	
Reserved		C	
Master Clear **		D	
Reserved		E	
Reserved		F	

Note: * Bits 7-4 of the Extended Function Register. ** Direct commands to the Extended Function register

Figure 11. DMA Extended Commands

The following is an example showing the programming of DMA channel 2 using the 8237 compatible mode and the extended mode. In this example, to perform each step, write the data indicated to the corresponding addresses.

Program Step	8237 Compatible Mode Address/Data	Extended Mode Address/Data
Set Channel Mask Bit	(000AH) x6H	(0018H) 92H
Clear Byte Pointer	(000CH) xxH	(0018H) 22H
Write Memory Address	(0004H) xxH	(001AH) xxH
Write Page Table Address	(0081H) xxH	(001AH) xxH
Clear Byte Pointer	(000CH) xxH	(0018H) 42H
Write Register Count	(0005H) xxH	(001AH) xxH
Write Register Count	(0005H) xxH	(001AH) xxH
Write Mode Register	(000BH) xxH	(0018H) 72H (001AH) xxH
Clear Channel 2 Mask Bit	(000AH) x2H	(0018H) A2H

Note: x's represent data.

Figure 12. DMA Channel 2 Programming Example, Extended Commands