Direct Memory Access

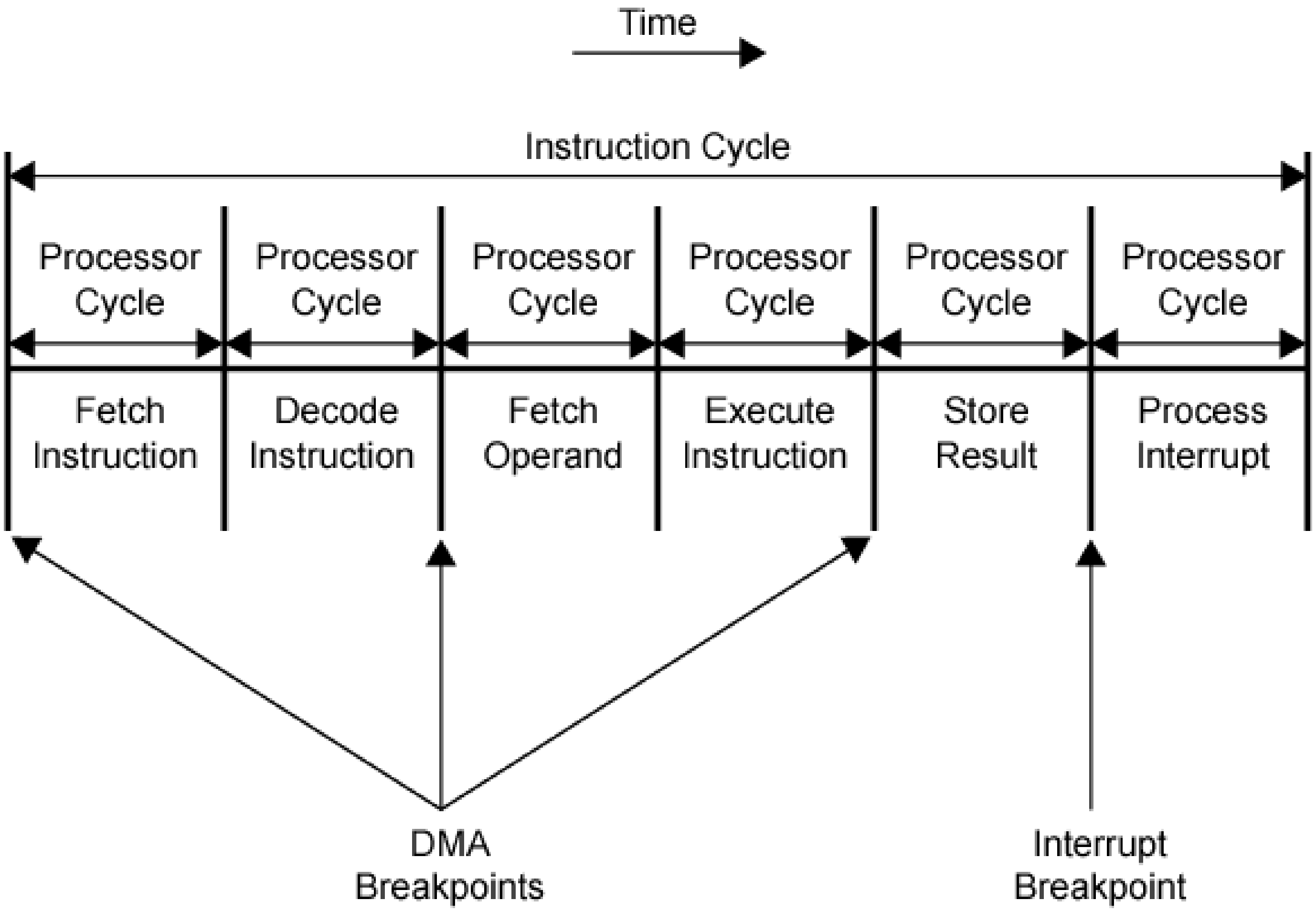
8237

- DMA or Direct Memory Access Controller is an external device that controls the transfer of data between I/O device and memory without the involvement of the processor. It holds the ability to directly access the main memory for read or write operation.
- DMA controller was designed by **Intel**, to have the fastest data transfer rate with less processor utilization.

- The microprocessor first fetches the instruction and then decodes it, then further execute it. But individually if the processor is performing all the task inside the system then it unnecessarily keeps the processor busy all the time.
- To enhance the performance of the processor, an external device is used that can manage data transfer operation between peripherals and memory with least CPU utilization.

- It is nothing but hardware controlled data transfer, where the address and control signals required for transferring the data is generated by an external device.
- This method of data transfer is known as direct memory access and the external device used for this purpose is known as DMA controller.

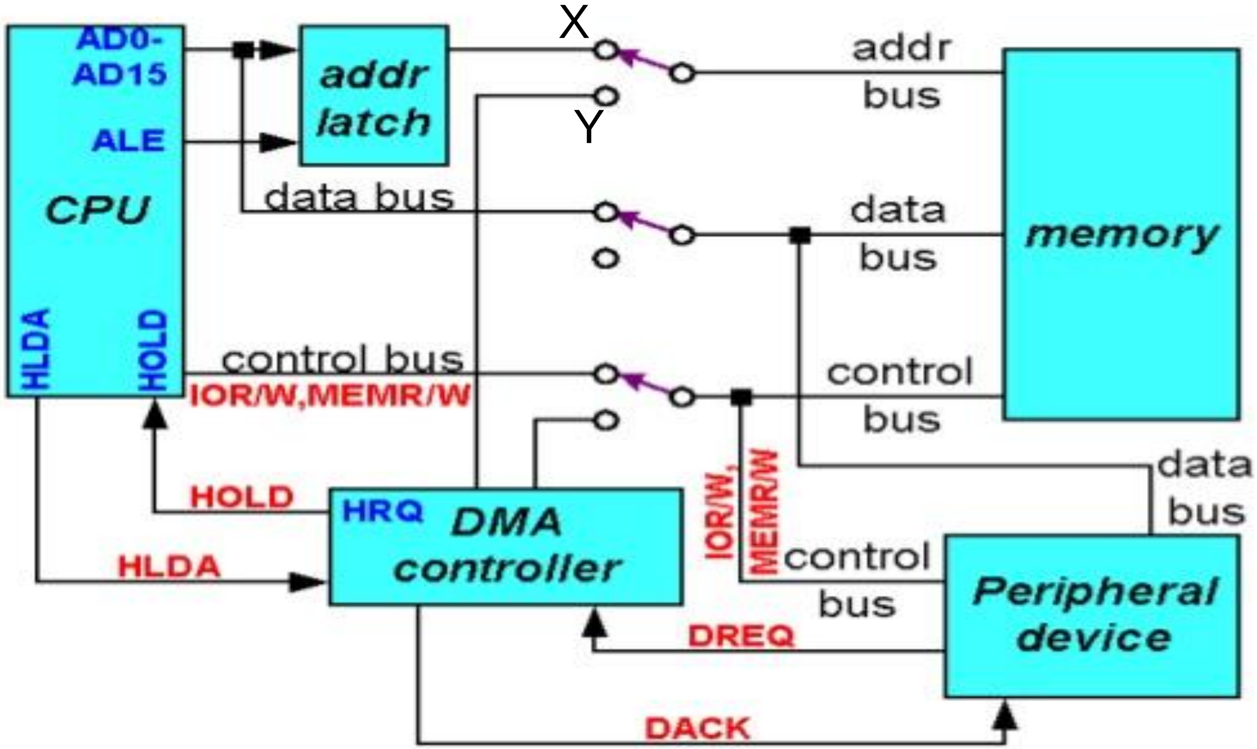
DMA and Interrupt Breakpoints During an Instruction Cycle



Operation of DMA Controller in Microprocessor

- In the **idle cycle** of the system, initially when the system gets on, then the processor has control over the system buses, as the switch are connected with the X position.
- in this position, the buses form the connection between main memory and peripherals through the processor. So, in this position, the processor performs the execution of the instruction.

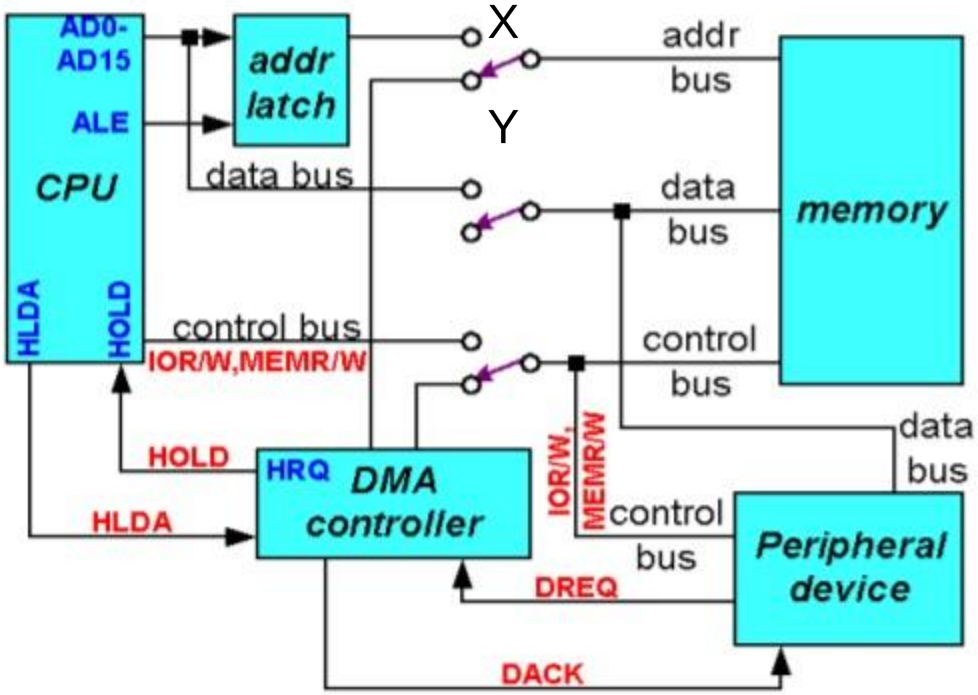
CPU having the control over the bus:



- When need arises to read the data from the disk. Then the microprocessor sends an instruction to the disk controller about the read operation of that particular data.
- On fetching the required data, the disk controller (peripheral device) sends DMA request, i.e., DRQ signal to the DMA controller. This DRQ signal shows that the device directly wants to transfer the data to the memory without disturbing the processor.

- on receiving the DRQ signal, HOLD request i.e., HRQ signal is sent by the DMA controller to the microprocessor.
- The HRQ signal shows the interest of the DMA controller to have access to system buses. So, on receiving HOLD request, the CPU tristates its buses in order to grant the control to the DMA controller.
- Once the processor frees the buses, then it sends the HLDA signal to the DMA controller. And on receiving HLDA signal, the control over the buses is given to the DMA controller as the switch position now changes from X to Y.
- So, gaining control over the buses, the **active cycle** of the DMA gets enabled. Thus now it sends the acknowledge signal DACK to the disk controller that shows that it is now ready for the transfer of data.

When DMA operates:



Basic Process of DMA-Min mode

- The **HOLD** and **HLDA** pins are used to receive and acknowledge the hold request respectively
- Normally the CPU has full control of the system bus.
- In a DMA operation, the peripheral takes over bus control temporarily.

Basic Process of DMA-Max mode

The **RQ/GT1** and **RQ/GT0** pins are used to issue DMA request and receive acknowledge signals.

- Sequence of events of a typical DMA process:

1. Peripheral asserts one of the request pins, e.g. RQ/GT1 or RQ/GT0 (RQ/GT0 has higher priority)
2. 8086 completes its current bus cycle and enters into a HOLD state.
3. 8086 grants the right of bus control by asserting a grant signal via the same pin as the request signal.
4. DMA operation starts.
5. Upon completion of the DMA operation, the peripheral asserts the request/grant pin again to relinquish bus control.

A DMA controller interfaces with several peripherals that may request DMA.

- The controller decides the priority of simultaneous DMA requests communicates with the peripheral and the CPU, and provides memory addresses for data transfer.

- DMA controller commonly used with 8086 is the 8257/8237 programmable device.

- The 8257/8237 is a 4-channel device.

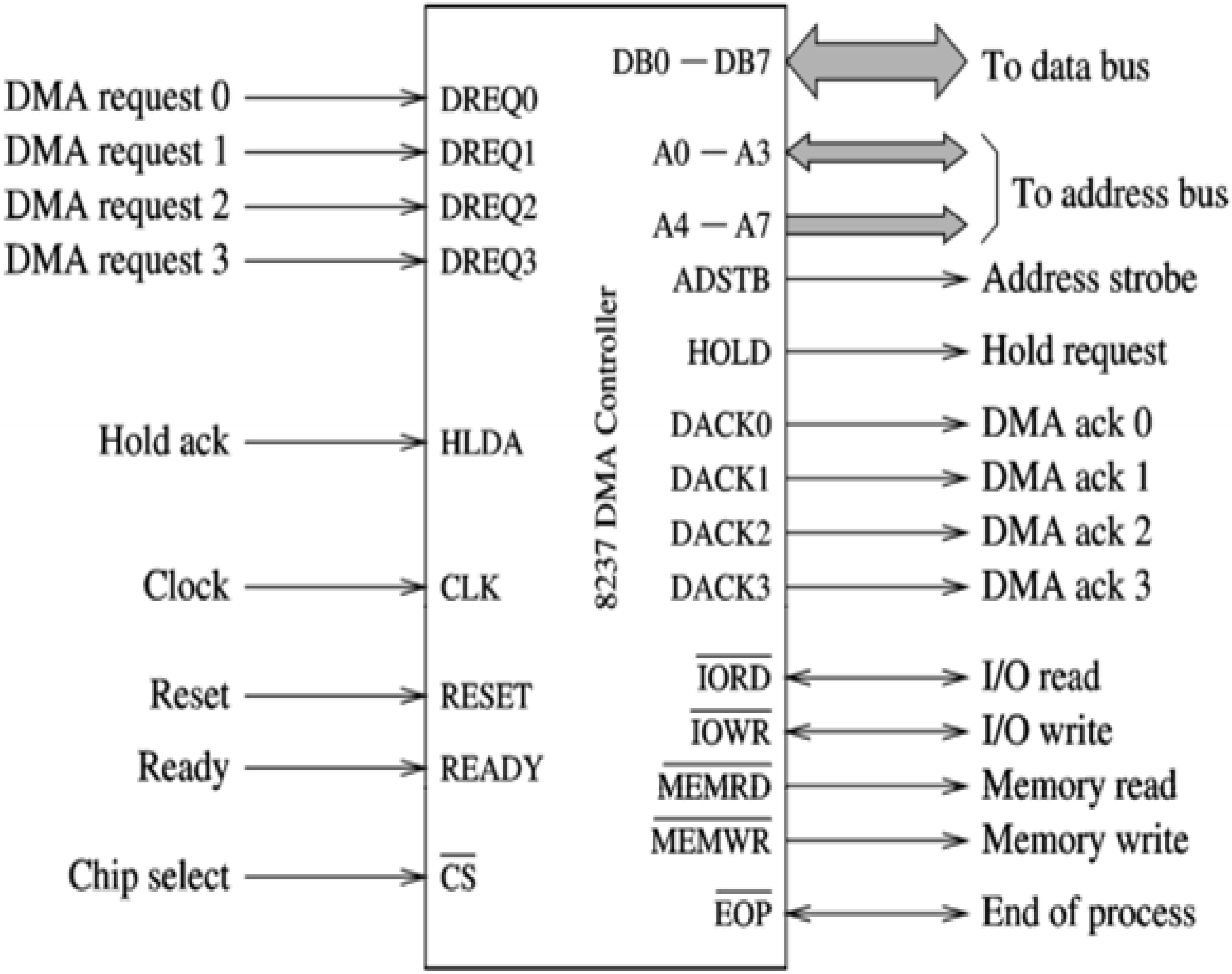
Each channel is dedicated to a specific peripheral device and capable of addressing 64 K bytes section of memory.

Features of 8237

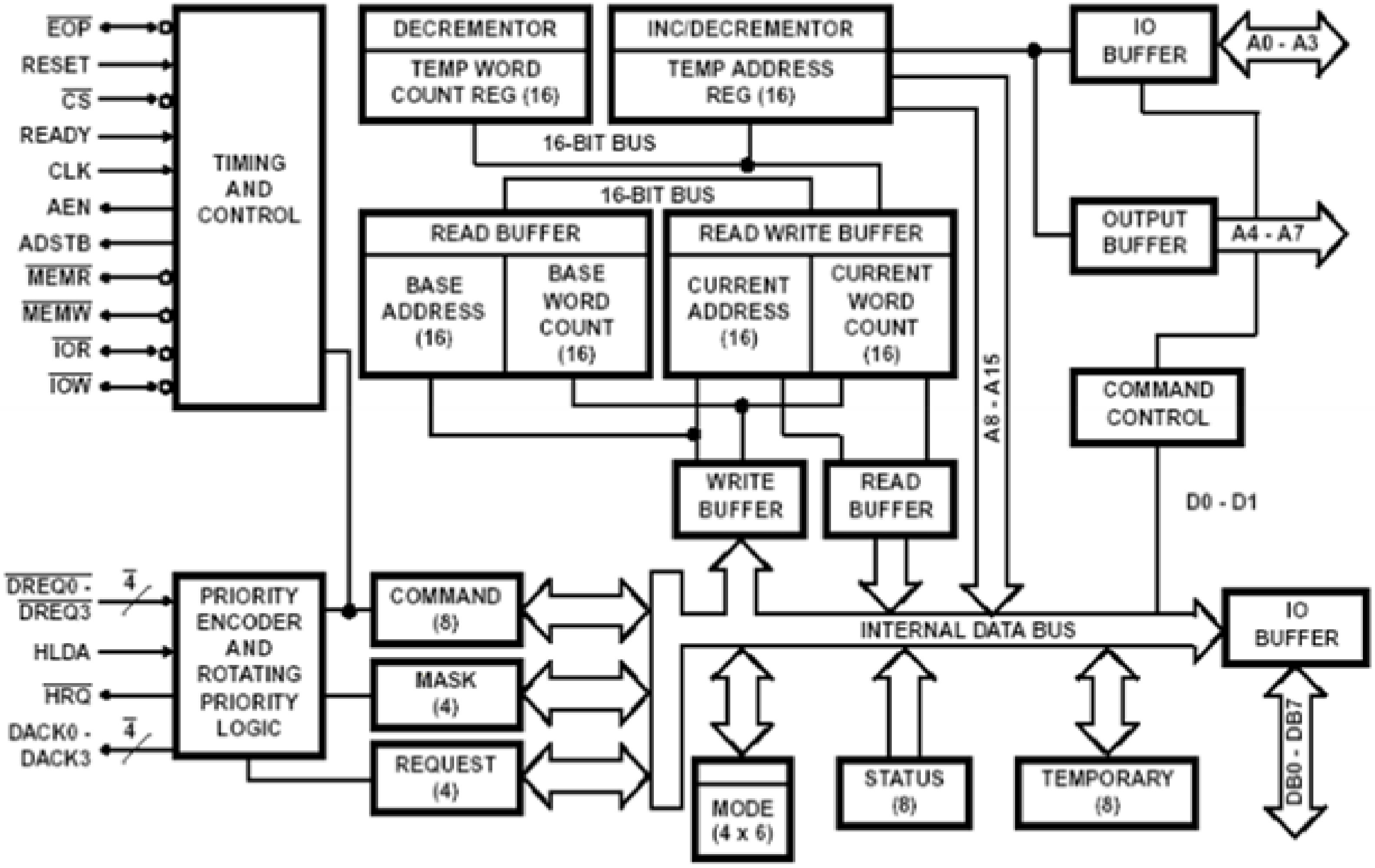
1. It provides various modes of DMA.
2. It provide on chip 4 independent channel. the no. of channel can be increase by cascading.
3. Each channel can be use auto initialize mode.
4. It can transfer data between memory to memory.
5. Address of memory is either increment or decrement.
6. Clock frequency 3Mhz.
7. Data transfer rate 1.6 Mbps/sec.

9. Directly expendable to any no of channel by cascading.
10. It provide EOP line that is used for terminate DMA operation. This signal can be generated by external h/w.
11. DMA can be requested by setting an appropriate bit of request register.
12. Independent control for DREQ and DACK. These signal can be initialize by active high or low.
13. It provide compressed timing to improve throughput. it can compress the transfer time to two (2s)

8237 - DMA Controller



Block Diagram



8237 Pin Definitions

CLK

- **Clock** input is connected to the system clock signal as long as that signal is 5 MHz or less.
  - in the 8086/8088 system, the clock must be inverted for the proper operation of the 8237

- **Chip select** enables 8237 for programming.
- The CS’ pin is normally connected to the output of a decoder.
- The decoder does not use the 8086/8088 control signal IO/M’(M/IO’) because it contains the new memory and I/O control signals (MEMR,’ MEMW,’ IOR’ and IOW’).

8237 Pin Definitions

RESET

- The **reset** pin clears the command, status, request, and temporary registers.
- It also clears the first/last flip-flop and sets the mask register.
  - this input primes the 8237 so it is disabled until programmed otherwise

8237 Pin Definitions

READY

- A logic 0 on the **ready** input causes the 8237 to enter wait states for slower memory components.

HLDA

- A **hold acknowledge** signals 8237 that the microprocessor has relinquished control of the address, data, and control buses.

8237 Pin Definitions

$DREQ_0$–$DREQ_3$

- **DMA request inputs** are used to request a transfer for each of the four DMA channels.
  - the polarity of these inputs is programmable, so they are either active-high or active-low inputs

DB_0–DB_7

- **Data bus** pins are connected to the processor data bus connections and used during the programming of the DMA controller.

8237 Pin Definitions

$\overline{\text{IOR}}$

- **I/O read** is a bidirectional pin used during programming and during a DMA write cycle.

$\overline{\text{IOW}}$

- **I/O write** is a bidirectional pin used during programming and during a DMA read cycle.

8237 Pin Definitions

$\overline{\text{EOP}}$

- **End-of-process** is a bidirectional signal used as an input to terminate a DMA process or as an output to signal the end of the DMA transfer.
  - often used to interrupt a DMA transfer at the end of a DMA cycle

A0-A3

- These **address pins** select an internal register during programming and provide part of the DMA transfer address during a DMA action.
  - address pins are outputs that provide part of the DMA transfer address during a DMA action

8237 Pin Definitions

HRQ

- **Hold request** is an output that connects to the HOLD input of the microprocessor in order to request a DMA transfer.

DACK0-DACK3

- **DMA channel acknowledge** outputs acknowledge a channel DMA request.
- These outputs are programmable as either active-high or active-low signals.
  - DACK outputs are often used to select the DMA- controlled I/O device during the DMA transfer.

8237 Pin Definitions

AEN

- **Address enable** signal enables the DMA address latch connected to the DB_7–DB_0 pins on the 8237.
  - also used to disable any buffers in the system connected to the microprocessor

ADSTB

Address strobe functions as ALE, except it is used by the DMA controller to latch address bits A_{15}–A_8 during the DMA transfer

8237 Pin Definitions

MEMR’

- **Memory read** is an output that causes memory to read data during a DMA read cycle.

MEMW’

- **Memory write** is an output that causes memory to write data during a DMA write cycle.

8237 Internal Registers

CAR

- The **current address register** holds a 16-bit memory address used for the DMA transfer.
  - each channel has its own current address register for this purpose
- When a byte of data is transferred during a DMA operation, CAR is either incremented or decremented.
  - depending on how it is programmed

8237 Internal Registers

CWCR

- The **current word count register** programs a channel for the number of bytes (up to 64K) transferred during a DMA action.
- The number loaded into this register is one less than the number of bytes transferred.
  - for example, if a 10 is loaded to CWCR, then 11 bytes are transferred during the DMA action

8237 Internal Registers

BA and BWC

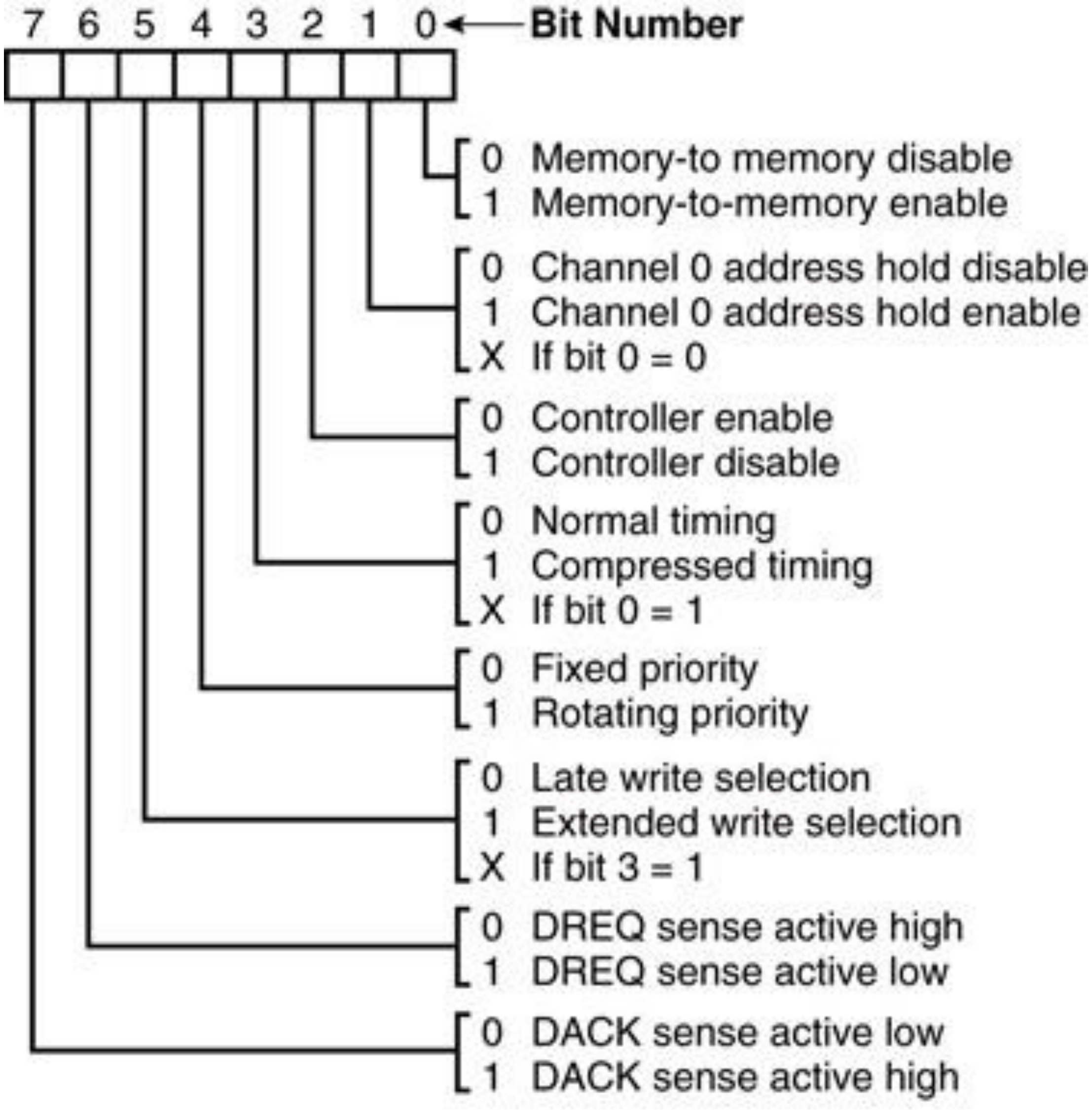
- The **base address** (BA) and **base word count** (BWC) registers are used when auto-initialization is selected for a channel.
- In auto-initialization mode, these registers are used to reload the CAR and CWCR after the DMA action is completed.
  - allows the same count and address to be used to transfer data from the same memory area

8237 Internal Registers

CR

- The **command register** programs the operation of the 8237 DMA controller.
- The register uses bit position 0 to select the memory-to-memory DMA transfer mode.
  - memory-to-memory DMA transfers use DMA channel 0 to hold the source address
  - DMA channel 1 holds the destination address
- Similar to operation of a MOVSB instruction.

Figure 13–4 8237A-5 command register. (Courtesy of Intel Corporation.)



0 Memory-to memory disable
1 Memory-to-memory enable

0 Channel 0 address hold disable
1 Channel 0 address hold enable
X If bit 0 = 0

0 Controller enable
1 Controller disable

0 Normal timing
1 Compressed timing
X If bit 0 = 1

0 Fixed priority
1 Rotating priority

0 Late write selection
1 Extended write selection
X If bit 3 = 1

0 DREQ sense active high
1 DREQ sense active low

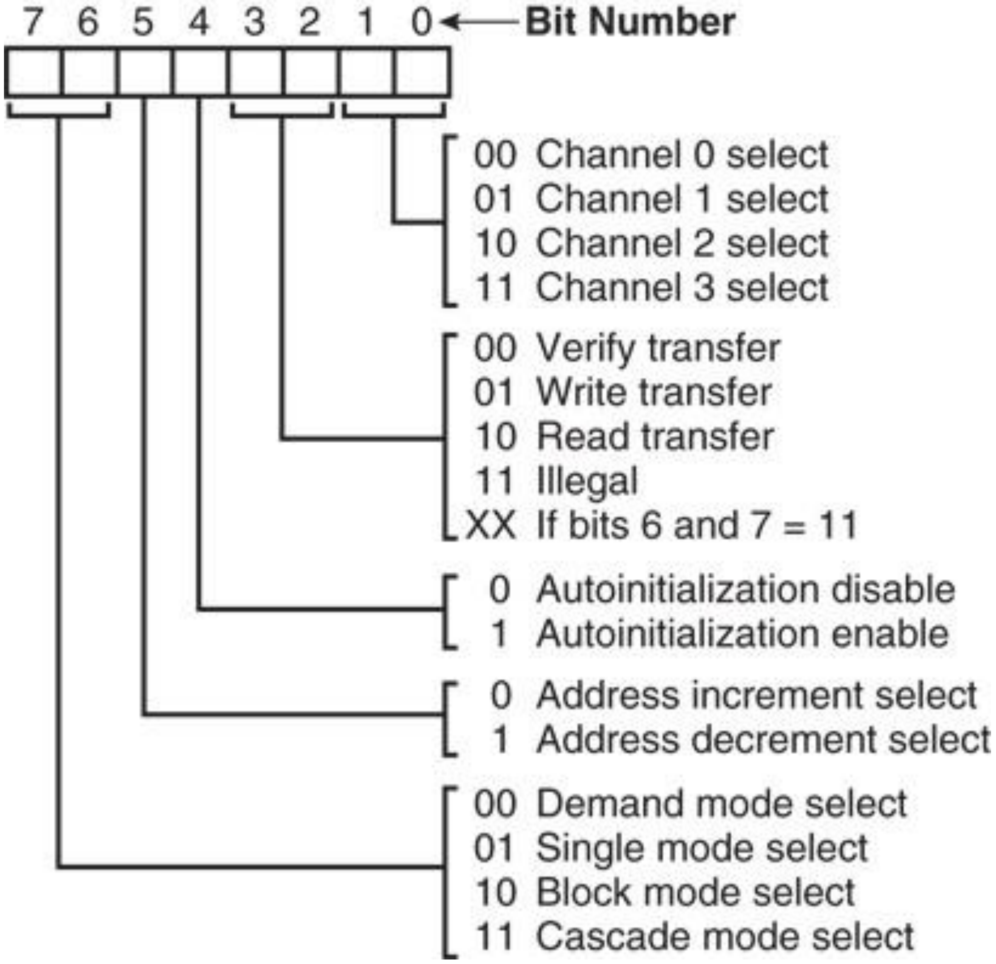
0 DACK sense active low
1 DACK sense active high

8237 Internal Registers

MR

- The **mode register** programs the mode of operation for a channel.
- Each channel has its own mode register as selected by bit positions 1 and 0.
  - remaining bits of the mode register select operation, auto-initialization, increment/decrement, and mode for the channel

Figure 13–5 8237A-5 mode register. (Courtesy of Intel Corporation.)

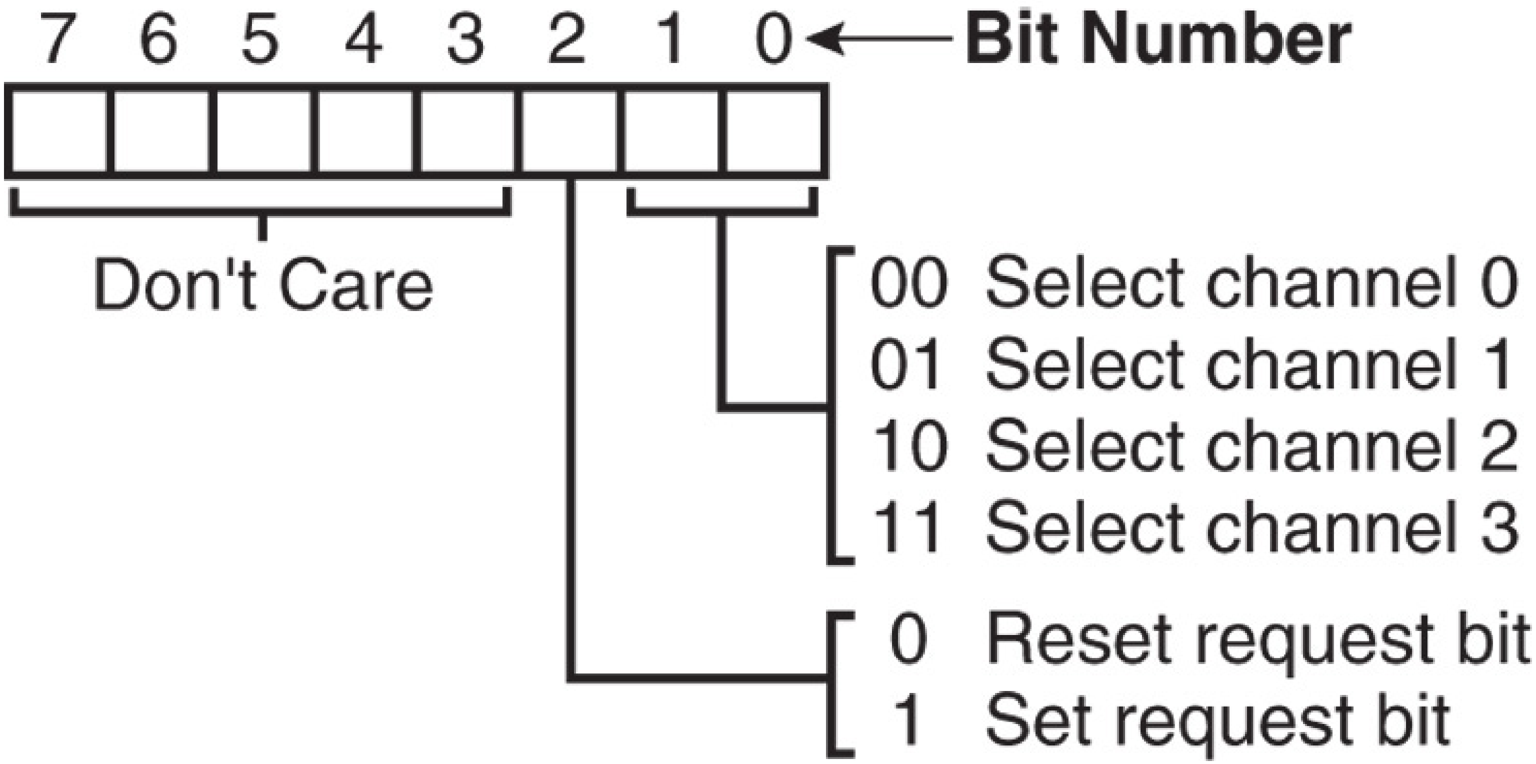


8237 Internal Registers

BR

- The **bus request register** is used to request a DMA transfer via software.
  - very useful in memory-to-memory transfers, where an external signal is not available to begin the DMA transfer

Figure 13–6 8237A-5 request register. (Courtesy of Intel Corporation.)

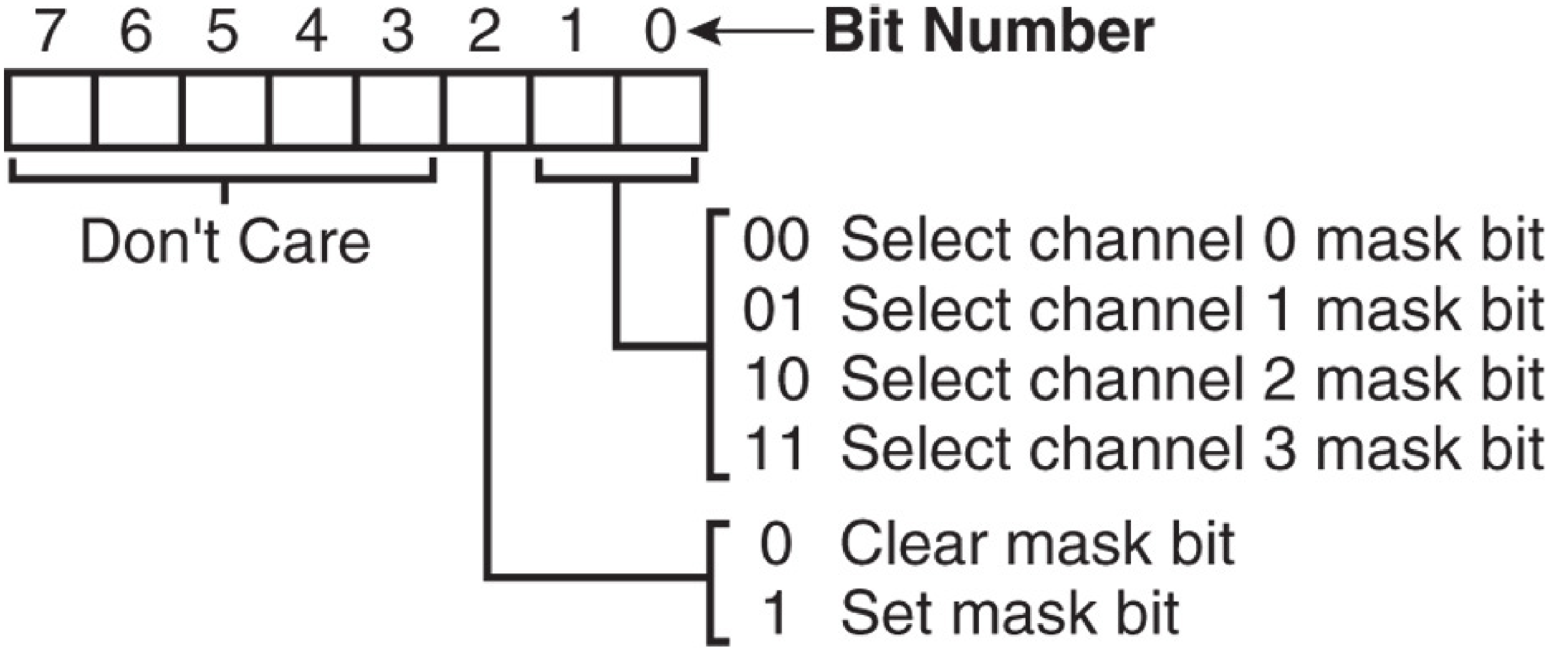


8237 Internal Registers

MRSR

- The **mask register set/reset** sets or clears the channel mask.
  - if the mask is set, the channel is disabled
  - the RESET signal sets all channel masks to disable them

Figure 13–7 8237A-5 mask register set/reset mode. (Courtesy of Intel Corporation.)

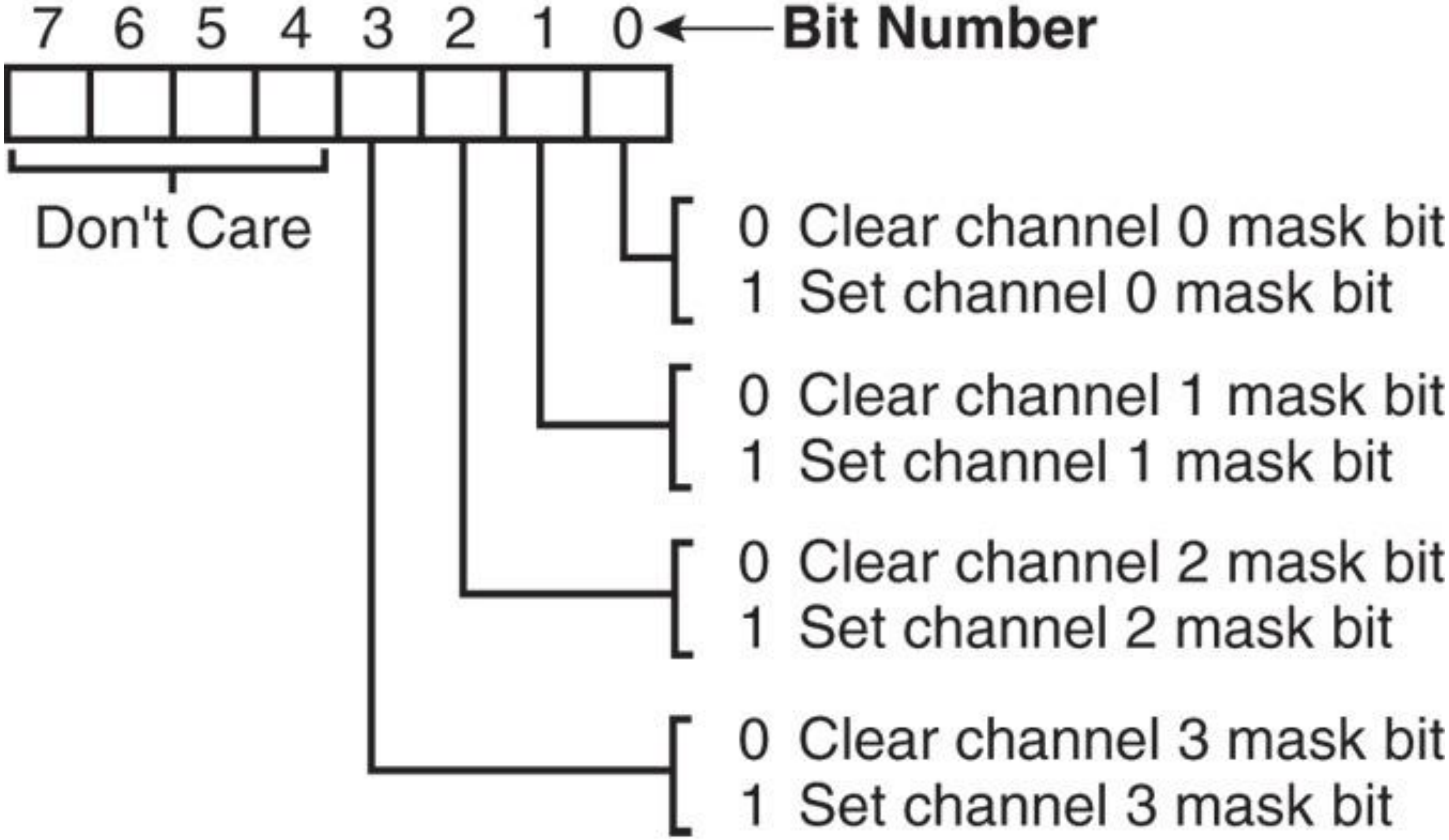

7 6 5 4 3 2 1 0
Bit Number
Don't Care
00 Select channel 0 mask bit
01 Select channel 1 mask bit
10 Select channel 2 mask bit
11 Select channel 3 mask bit
0 Clear mask bit
1 Set mask bit

8237 Internal Registers

MSR

- The **mask register** clears or sets all of the masks with one command instead of individual channels, as with the MRSR.

Figure 13–8 8237A-5 mask register. (Courtesy of Intel Corporation.)



7 6 5 4 3 2 1 0 ← Bit Number

Don't Care

0 Clear channel 0 mask bit
1 Set channel 0 mask bit

0 Clear channel 1 mask bit
1 Set channel 1 mask bit

0 Clear channel 2 mask bit
1 Set channel 2 mask bit

0 Clear channel 3 mask bit
1 Set channel 3 mask bit

8237 Internal Registers

SR

- The **status register** shows status of each DMA channel. The TC bits indicate if the channel has reached its terminal count (transferred all its bytes).
- When the terminal count is reached, the DMA transfer is terminated for most modes of operation.
  - the request bits indicate whether the DREQ input for a given channel is active

Figure 13–9 8237A-5 status register. (Courtesy of Intel Corporation.)

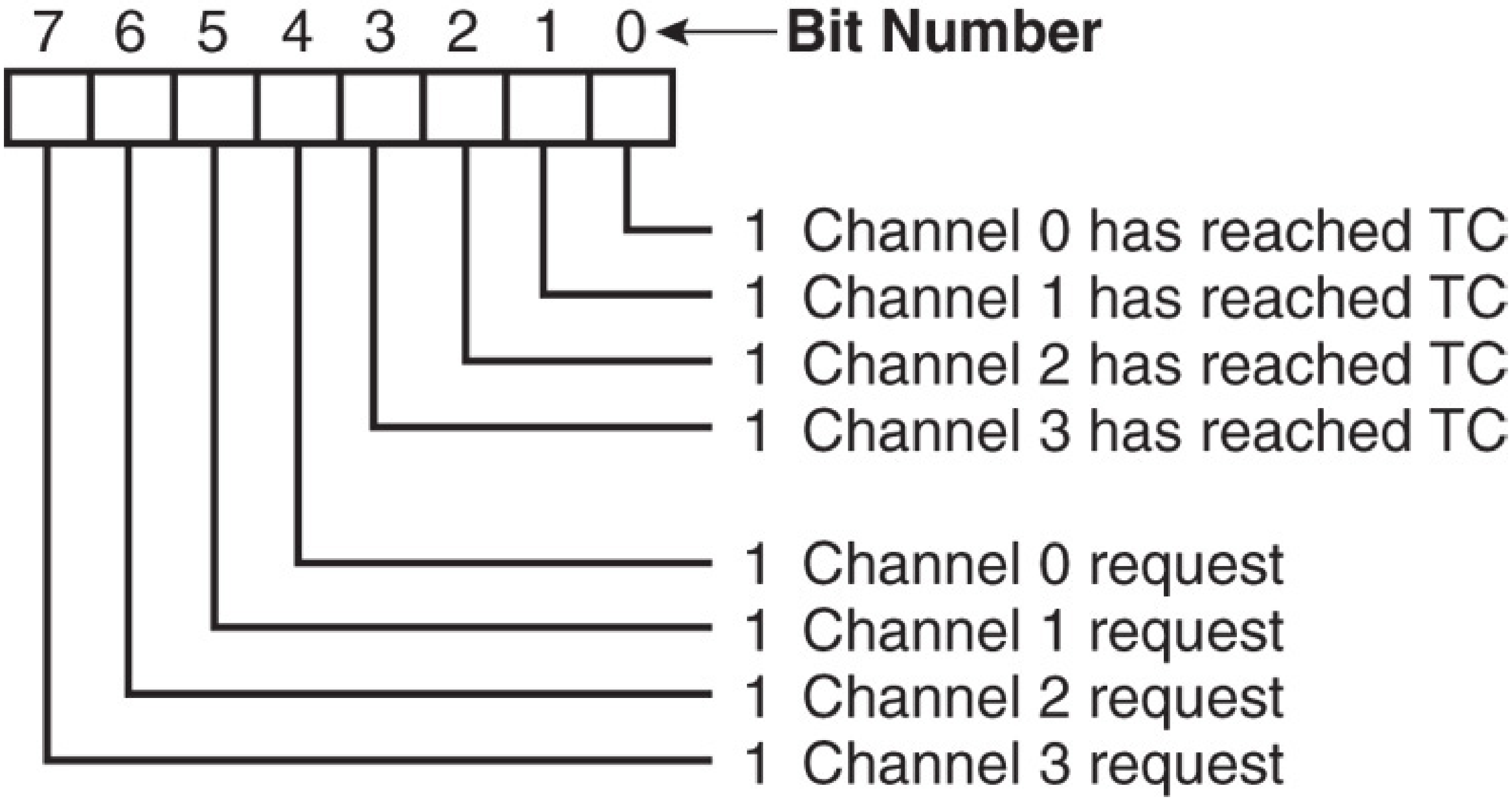
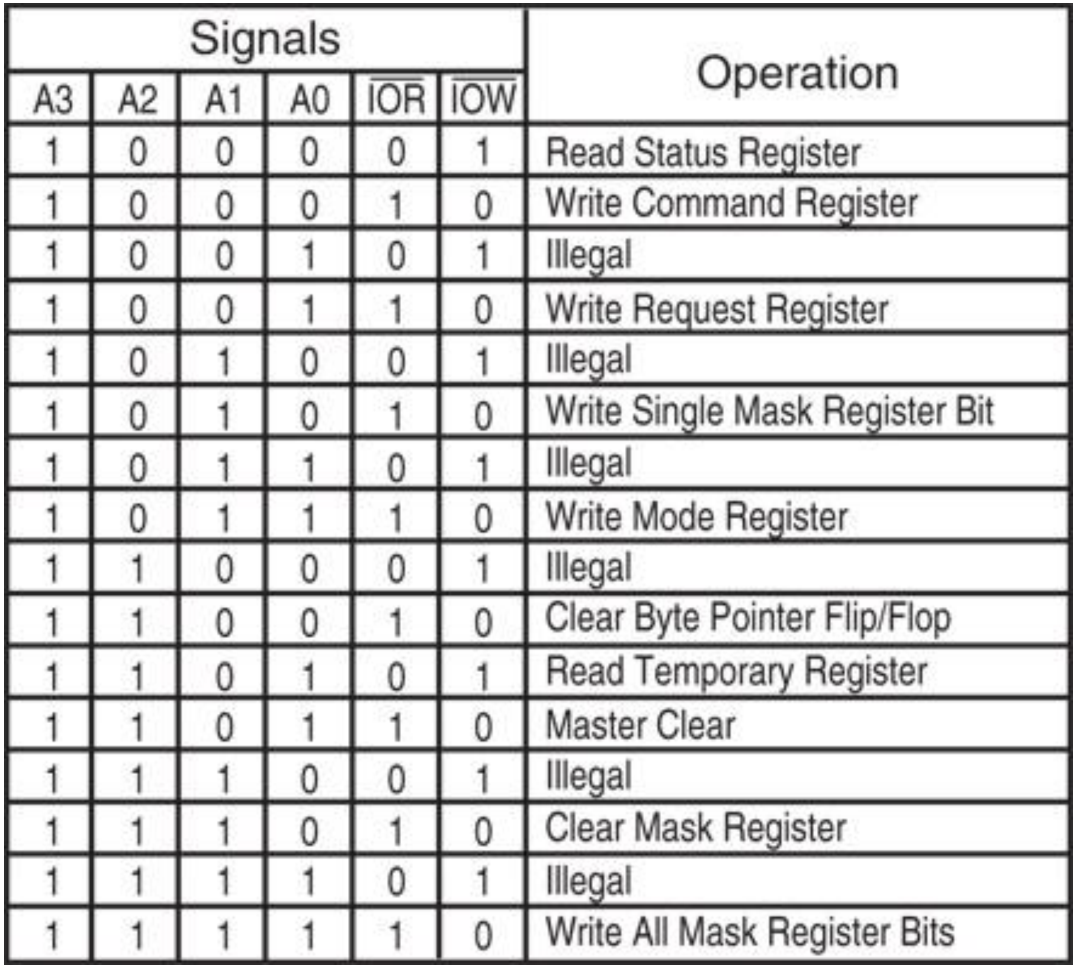

7 6 5 4 3 2 1 0
Bit Number
1 Channel 0 has reached TC
1 Channel 1 has reached TC
1 Channel 2 has reached TC
1 Channel 3 has reached TC
1 Channel 0 request
1 Channel 1 request
1 Channel 2 request
1 Channel 3 request

Figure 13–10 8237A-5 command and control port assignments. (Courtesy of Intel Corporation.)



Signals						Operation
A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	
1	0	0	0	0	1	Read Status Register
1	0	0	0	1	0	Write Command Register
1	0	0	1	0	1	Illegal
1	0	0	1	1	0	Write Request Register
1	0	1	0	0	1	Illegal
1	0	1	0	1	0	Write Single Mask Register Bit
1	0	1	1	0	1	Illegal
1	0	1	1	1	0	Write Mode Register
1	1	0	0	0	1	Illegal
1	1	0	0	1	0	Clear Byte Pointer Flip/Flop
1	1	0	1	0	1	Read Temporary Register
1	1	0	1	1	0	Master Clear
1	1	1	0	0	1	Illegal
1	1	1	0	1	0	Clear Mask Register
1	1	1	1	0	1	Illegal
1	1	1	1	1	0	Write All Mask Register Bits

8237 Software Commands

Master clear

- Acts exactly the same as the RESET signal to the 8237.
  - as with the RESET signal, this command disables all channels

Clear mask register

- Enables all four DMA channels.

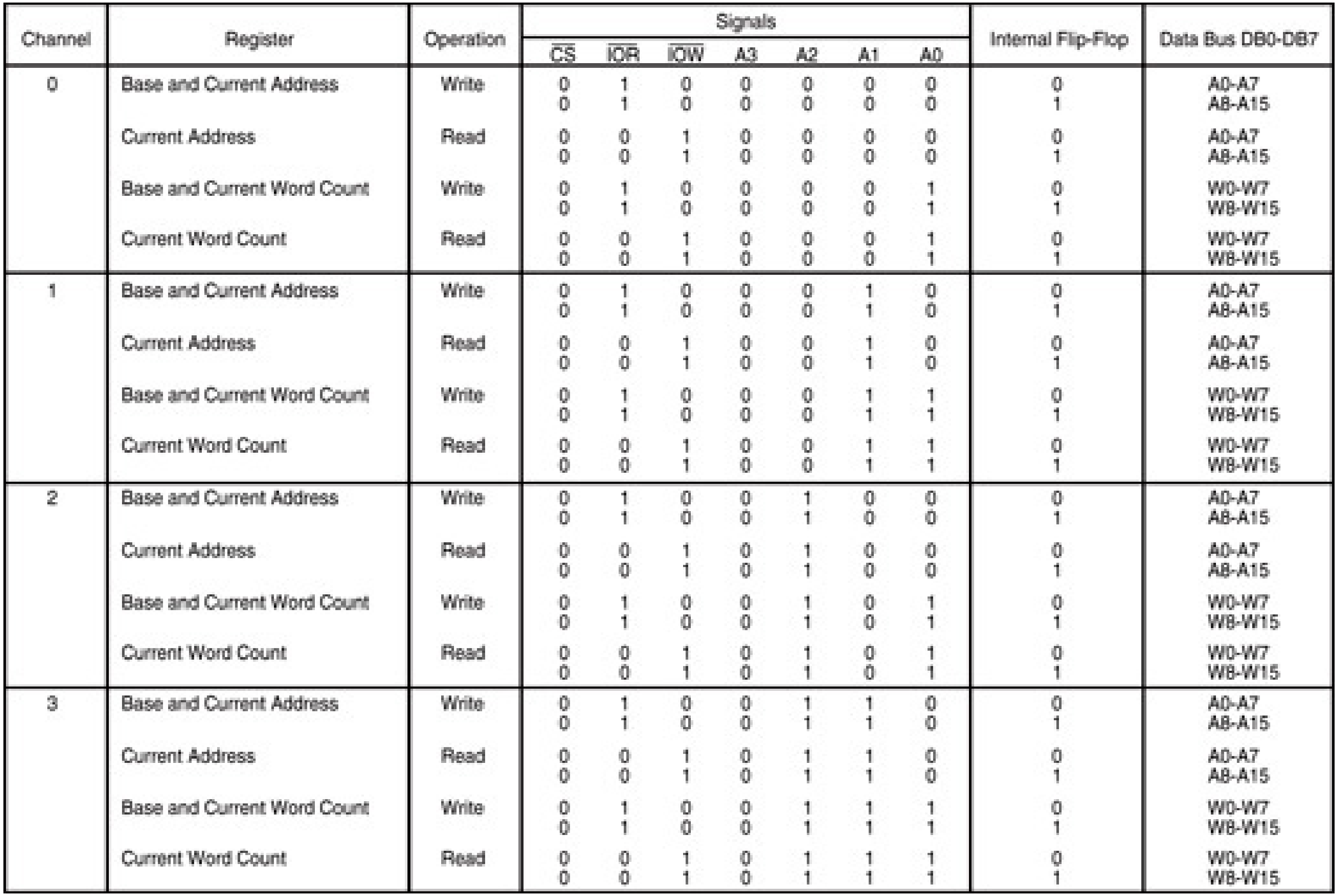
8237 Software Commands
Clear the first/last flip-flop

- Clears the first/last (F/L) flip-flop within 8237.
- The F/L flip-flop selects which byte (low or high order) is read/written in the current address and current count registers.
  - if F/L = 0, the low-order byte is selected
  - if F/L = 1, the high-order byte is selected
- Any read or write to the address or count register automatically toggles the F/L flip-flop.

Programming the Address and Count Registers

- Figure shows I/O port locations for programming the count and address registers for each channel.
- The state of the F/L flip-flop determines whether the LSB or MSB is programmed.
  - if the state is unknown, count and address could be programmed incorrectly
- It is important to disable the DMA channel before address and count are programmed.

Figure 8237A-5 DMA channel I/O port addresses. (Courtesy of Intel Corporation.)



Channel	Register	Operation	Signals							Internal Flip-Flop	Data Bus DB0-DB7
			$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0-A7
			0	1	0	0	0	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	0	0	0	A0-A7
			0	0	1	0	0	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	0	1	0	W0-W7
			0	1	0	0	0	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	0	1	0	W0-W7
			0	0	1	0	0	0	1	1	W8-W15
1	Base and Current Address	Write	0	1	0	0	0	1	0	0	A0-A7
			0	1	0	0	0	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	0	1	0	0	A0-A7
			0	0	1	0	0	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	0	1	1	0	W0-W7
			0	1	0	0	0	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	0	1	1	0	W0-W7
			0	0	1	0	0	1	1	1	W8-W15
2	Base and Current Address	Write	0	1	0	0	1	0	0	0	A0-A7
			0	1	0	0	1	0	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	0	0	0	A0-A7
			0	0	1	0	1	0	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	0	1	0	W0-W7
			0	1	0	0	1	0	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	0	1	0	W0-W7
			0	0	1	0	1	0	1	1	W8-W15
3	Base and Current Address	Write	0	1	0	0	1	1	0	0	A0-A7
			0	1	0	0	1	1	0	1	A8-A15
	Current Address	Read	0	0	1	0	1	1	0	0	A0-A7
			0	0	1	0	1	1	0	1	A8-A15
	Base and Current Word Count	Write	0	1	0	0	1	1	1	0	W0-W7
			0	1	0	0	1	1	1	1	W8-W15
	Current Word Count	Read	0	0	1	0	1	1	1	0	W0-W7
			0	0	1	0	1	1	1	1	W8-W15

- Four steps are required to program the 8237:
  - (1) The F/L flip-flop is cleared using a clear F/L command
  - (2) the channel is disabled
  - (3) LSB & MSB of the address are programmed
  - (4) LSB & MSB of the count are programmed
- Once these four operations are performed, the channel is programmed and ready to use.
  - additional programming is required to select the mode of operation before the channel is enabled and started

Example

- Design the 8237 decoding circuit and the 8237 address line connections so that the 8237 is in the address range 70h-7Fh
- Write a program that starts a block memory-to-memory DMA transfer from memory locations 10000H-13FFFH to 14000H-17FFFH using channel 0 as source and channel 1 as destination.

8237 Programming Example

```
CLEAR_FF       EQU 7CH ;F/L CLEAR VALUE
CH0_A          EQU 70H ;CHANNEL 0 ADDRESS
CH1_A          EQU 72H ;CHANNEL 1 ADDRESS
CH1_C          EQU 73H ;CHANNEL 1 COUNT
MODE           EQU 7BH ;MODE
CR             EQU 78H ;COMMAND REGISTER
MASKS          EQU 7FH ;MASKS
REQ            EQU 79H ;REQUEST REGISTER
STATUS         EQU 78H ;STATUS REGISTER
   ;ES = segment of source and destination
   ;SI = source address
   ;DI = destination address
   ;CX = count
DMA PROC FAR
   MOV AL, 0
   OUT CLEAR_FF, AL    ;CLEAR F/L FF
   MOV AX, ES          ;PROGRAM SOURCE ADDRESS
   SHL AX, 4           ;SHIFT LEFT SEGMENT
   ADD AX, SI          ;ADD SOURCE OFFSET
   OUT CH0_A, AL       ;CHANNEL 0 ADDRESS PROGRAMMING LSB FIRST
   MOV AL, AH          ;ONLY AL ALLOWED IN IN/OUT INSTRUCTIONS
   OUT CH0_A, AL       ;CHANNEL 0 ADDRESS PROGRAMMING MSB LAST
```

EXAMPLE (CONTINUED)

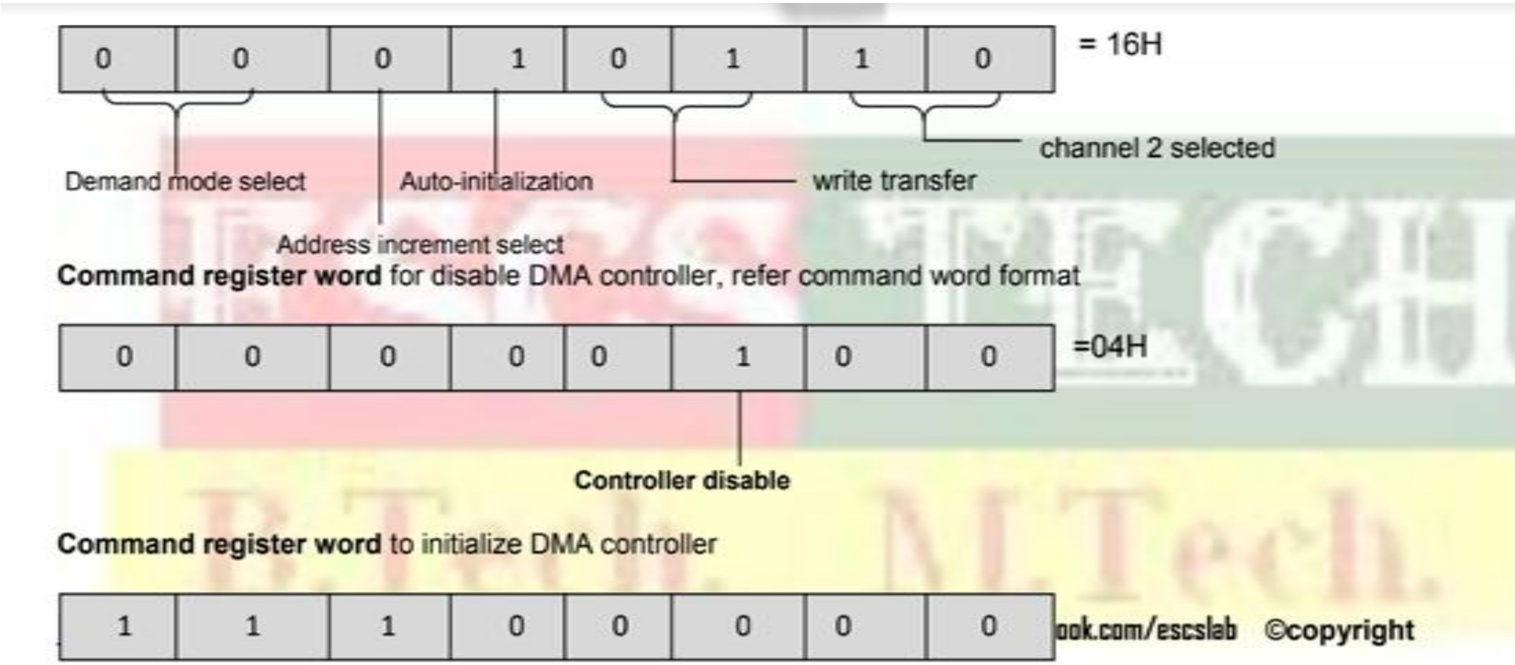
```
MOV AX, ES      ;PROGRAM DESTINATION ADDRESS
SHL AX, 4       ;SHIFT LEFT SEGMENT
ADD AX, DI      ;ADD DESTINATION OFFSET
OUT CH1_A, AL   ;CHANNEL 1 ADDRESS PROGRAMMING LSB FIRST
MOV AL, AH      ;ONLY AL ALLOWED IN IN/OUT INSTRUCTIONS
OUT CH1_A, AL   ;CHANNEL 1 ADDRESS PROGRAMMING MSB FIRST
MOV AX, CX      ;PROGRAM COUNT
DEC AX          ;ADJUST COUNT
OUT CH1_C, AL   ;MOVE TO CHANNEL 1 COUNT
MOV AL, AH
OUT CH1_C, AL
MOV AL, 88H     ;PROGRAM MODE
OUT MODE, AL
MOV AL,3        ;MEMORY-TO-MEMORY TRANSFER
OUT CR, AL
MOV AL, 0EH     ;UNMASK CHANNEL 0
OUT MASKS, AL
MOV AL, 4       ;START DMA TRANSFER BY SETTING REQUEST BIT FOR
                       CHANNEL 0
OUT REQ, AL
```

- Design an interfacing 8237 with the 8086 for mode register address 0BH and initialize the 8237 to meet following specifications:

1. Initialize the channel-2 to transfer 10K bytes from the system memory to the I/O device assigned to Channel-2.
2. The starting address of the data block to be transferred is 2050H and subsequent data bytes have memory address in increasing order.
3. Initialize the 8237 for normal timing, fixed priority, extended write with DREQ and DACK active high.
4. Initialize the 8237 in block mode of the DMA transfer with auto-initialization.

- Solution: Step-1: Interface the 8237 with the 8085 refer figure and assign the addresses of the internal registers as listed below:
- A7 A6 A5 A4 A3 A2 A1 A0
- 0 0 0 0 0 0 0 0 = 00H Channel 0 (CH0) CAR and BA address
- 0 0 0 0 0 0 0 1 = 01H Channel 0 (CH0) CWCR and BC address
- 0 0 0 0 0 0 1 0 = 02H Channel 1 (CH1) CAR and BA address
- 0 0 0 0 0 0 1 1 = 03H Channel 1 (CH1) CWCR and BC address
- 0 0 0 0 0 1 0 0 = 04H Channel 2 (CH2) CAR and BA address
- 0 0 0 0 0 1 0 1 = 05H Channel 2 (CH2) CWCR and BC address
- 0 0 0 0 0 1 1 0 = 06H Channel 3 (CH3) CAR and BA address
- 0 0 0 0 0 1 1 1 = 07H Channel 3 (CH3) CWCR and BC address
- 0 0 0 0 1 0 0 0 = 08H Status/Command Register address
- 0 0 0 0 1 0 0 1 = 09H Request Register (RR) address
- 0 0 0 0 1 0 1 0 = 0AH Individual mask register address
- 0 0 0 0 1 0 1 1 = 0BH Mode register (MR) address
- 0 0 0 0 1 1 0 0 = 0CH clear byte pointer flip-flop
- 0 0 0 0 1 1 0 1 = 0DH Temporary register
- 0 0 0 0 1 1 1 0 = 0EH Clear Mask Register
- 0 0 0 0 1 1 1 1 = 0FH All/common Mask Register

Write command/mode word for different programmable registers as per the given specification.
Mode register word, refer mode word format



0	0	0	1	0	1	1	0	= 16H

Command register word for disable DMA controller, refer command word format

0	0	0	0	0	1	0	0	=04H

Command register word to initialize DMA controller

1	1	1	0	0	0	0	0

- Initialize count register for 10K of byte:
  10x1024=10240 (2800H)
  Write count value one less than the number of byte to be transfer i.e. write 27FFH into Current Count Register.

- Step-3: Write Assembly language program to initialize DMA transfer
- MOV AL,04; load AL with command word:
- **OUT 08, AL ;**write command word in command register to disable DMA controller
- MOV AL,16H ; load AL with Mode word
- OUT 0BH,AL ; write mode word in mode register
- MOV AL, 50H ; load AL with LSB of system memory address
- OUT 04H,AL; write lower address byte into current address register (CAR)
- MOV AL, 20H ; load AL with MSB of memory address
- OUT 04H,AL; write higher address byte into CAR
- MOV AL, FFH ; load AL with LSB of count value
- OUT 05H ,AL; write lower byte of count value into current count register (CCR)
- MOV AL, 27H ; load AL with MSB of count value
- OUT 05H ,AL; write higher byte of count value into CCR
- MOV AL, E0H ; load AL with command word
- OUT 08H ,AL; write command word into commend register to start transfer.
- NOP ; wait for transfer start