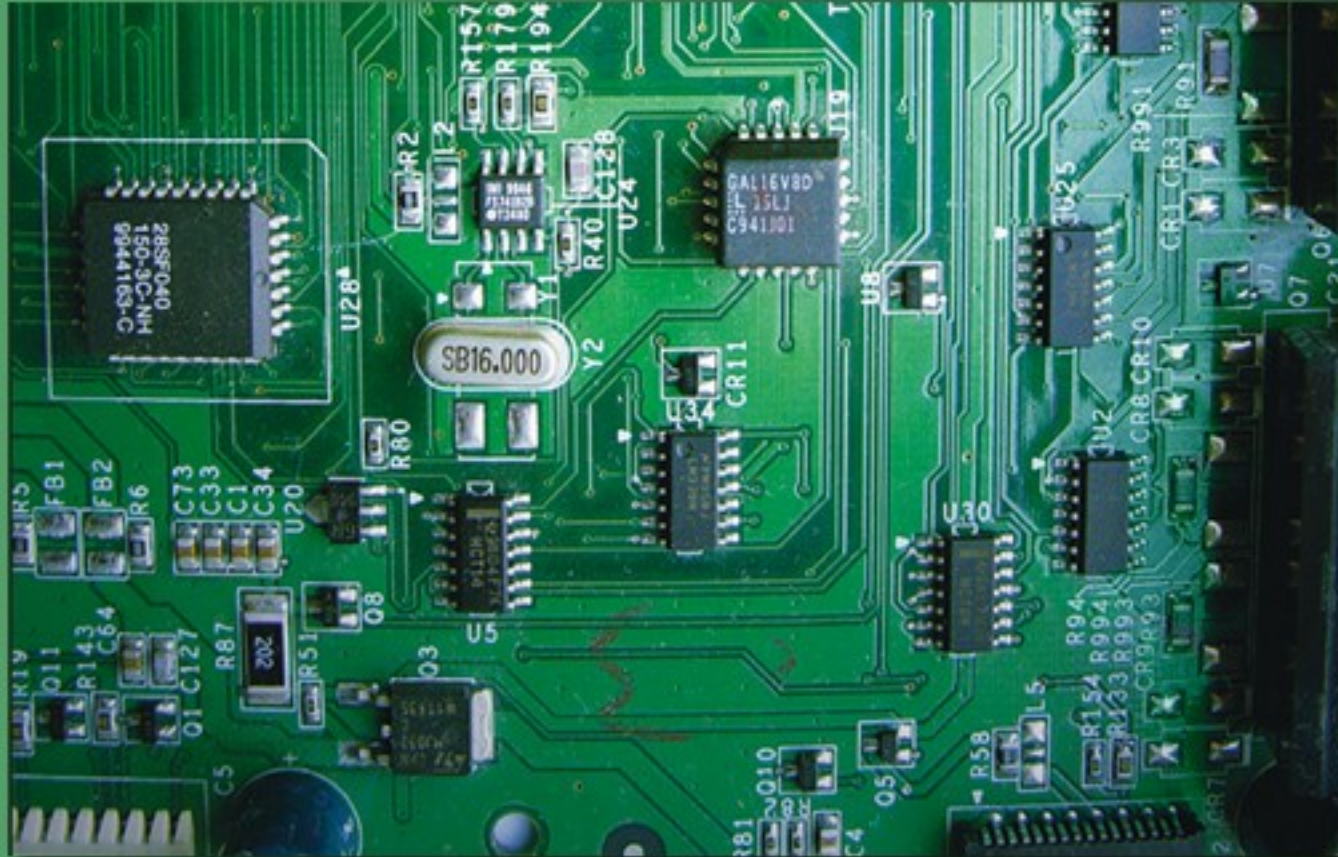


# The Intel Microprocessors

8086/8088, 80186/80188, 80286, 80386, 80486 Pentium, Pentium Pro Processor, Pentium II, Pentium 4, and Core2 with 64-bit Extensions

## Architecture, Programming, and Interfacing



EIGHTH EDITION

Barry B. Brey

PEARSON

## Chapter 13: Direct Memory Access and DMA-Controlled I/O

# Introduction

- The DMA I/O technique provides **direct access** to the **memory** while the microprocessor is temporarily disabled.
- This chapter also explains the operation of disk memory systems and video systems that are often **DMA-processed**.
- Disk memory includes floppy, fixed, and optical disk storage. Video systems include digital and analog monitors.

# Chapter Objectives

Upon completion of this chapter, you will be able to:

- Describe a DMA transfer.
- Explain the operation of the HOLD and HLDA direct memory access control signals.
- Explain the function of the 8237 DMA controller when used for DMA transfers.
- Program the 8237 to accomplish DMA transfers.

# Chapter Objectives

(*cont.*)

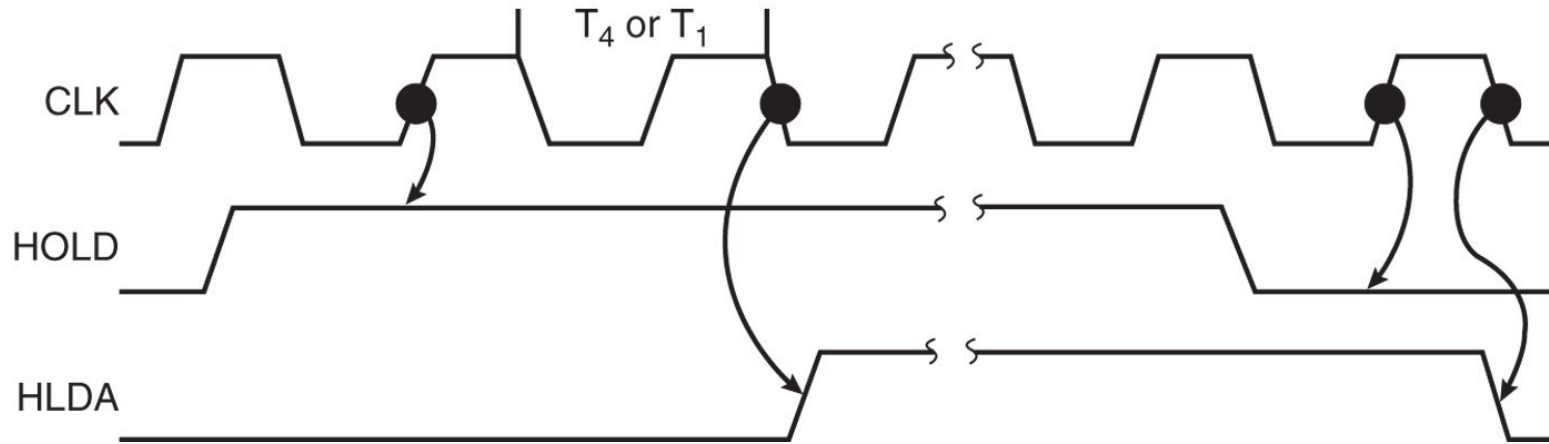
Upon completion of this chapter, you will be able to:

- Describe the **disk standards** found in personal computer systems.
- Describe the various **video interface standards** found in the personal computer.

# 13–1 BASIC DMA OPERATION

- Two control signals are used to request and acknowledge a **direct memory access** (DMA) transfer in the microprocessor-based system.
  - the **HOLD pin** is an **input** used to **request** a DMA action
  - the **HLDA pin** is an **output** that **acknowledges** the DMA action
- Figure 13–1 shows the timing that is typically found on these **two DMA control pins**.

**Figure 13–1** HOLD and HLDA timing for the microprocessor.



- HOLD is sampled in any clocking cycle
- when the processor recognizes the hold, it stops executing software and enters hold cycles
- HOLD input has **higher priority** than INTR or NMI
- the only microprocessor pin that has a higher priority than a HOLD is the RESET pin

- **HLDA** becomes active to indicate the processor has placed its **buses** at **high-impedance state**.
  - as can be seen in the timing diagram, there are a few clock cycles between the time that **HOLD** changes and until **HLDA** changes
- HLDA output is a signal to the requesting device that the processor has relinquished control of its memory and I/O space.
  - one could call HOLD input a DMA request input and HLDA output a DMA grant signal



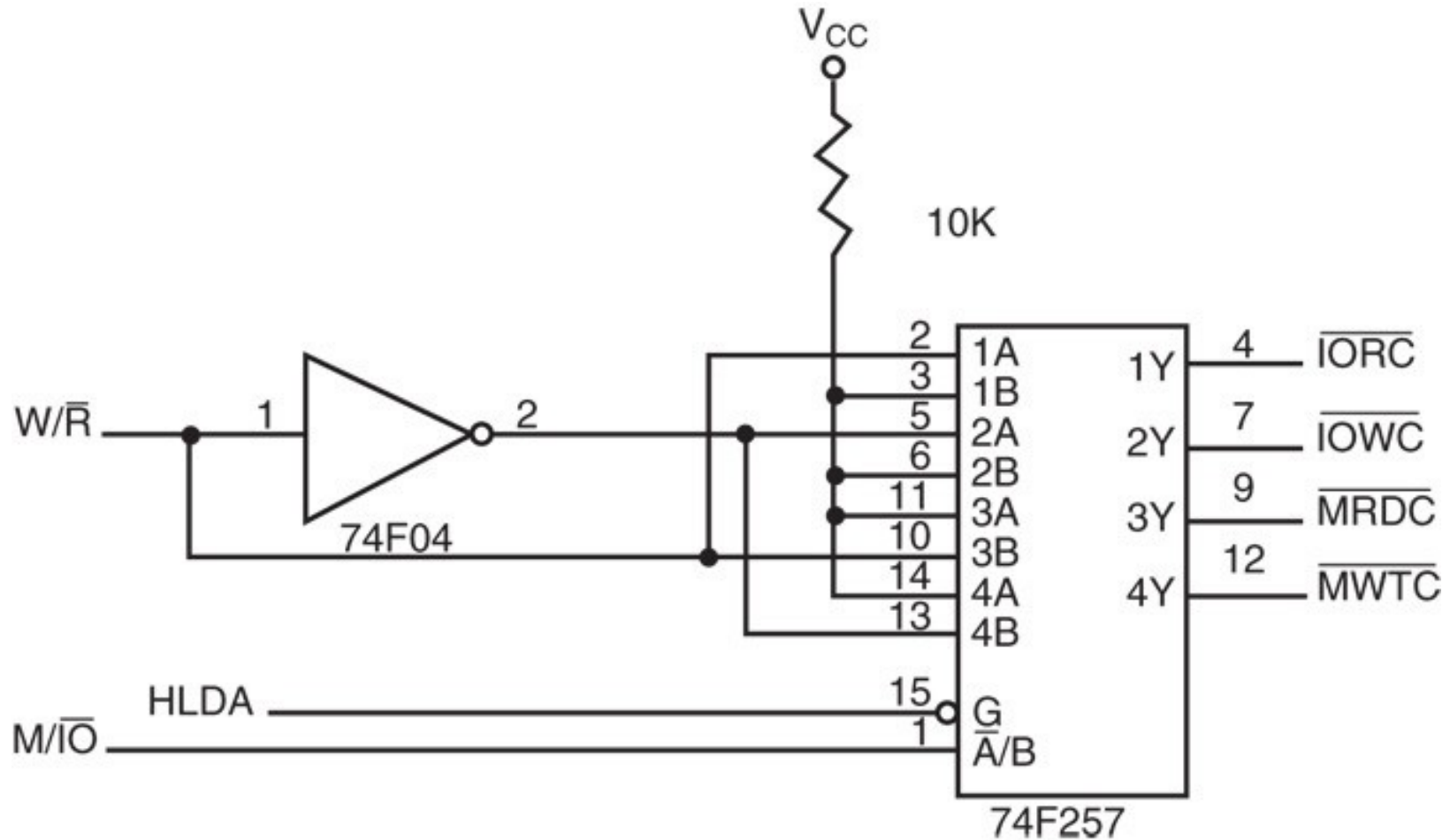
# Basic DMA Definitions

- Direct memory accesses normally occur between an I/O device and memory without the use of the microprocessor.
  - a **DMA read** transfers data from the memory to the I/O device
  - A **DMA write** transfers data from an I/O device to memory
- **Memory & I/O** are controlled **simultaneously**.
  - which is why the system contains separate memory and I/O control signals
  - This special control bus structure allows DMA transfer



- A **DMA read** causes the  $\overline{\text{MRDC}}$  and  $\overline{\text{IOWC}}$  signals to activate simultaneously.
  - transferring data from memory to the I/O device
- A **DMA write** causes the  $\overline{\text{MWTC}}$  and  $\overline{\text{IORC}}$  signals to both activate.
- These control signals are available to all microprocessors except the 8088/8086.
- 8086/8088 require a **controller** or **circuit** such as shown in Fig 13–2 for control bus signal generation.
- The **DMA controller** provides memory with its address, and controller signal ( $\overline{\text{DACK}}$ ) selects the I/O device during the transfer.

**Figure 13–2** A circuit that **generates system control signals** in a DMA environment.



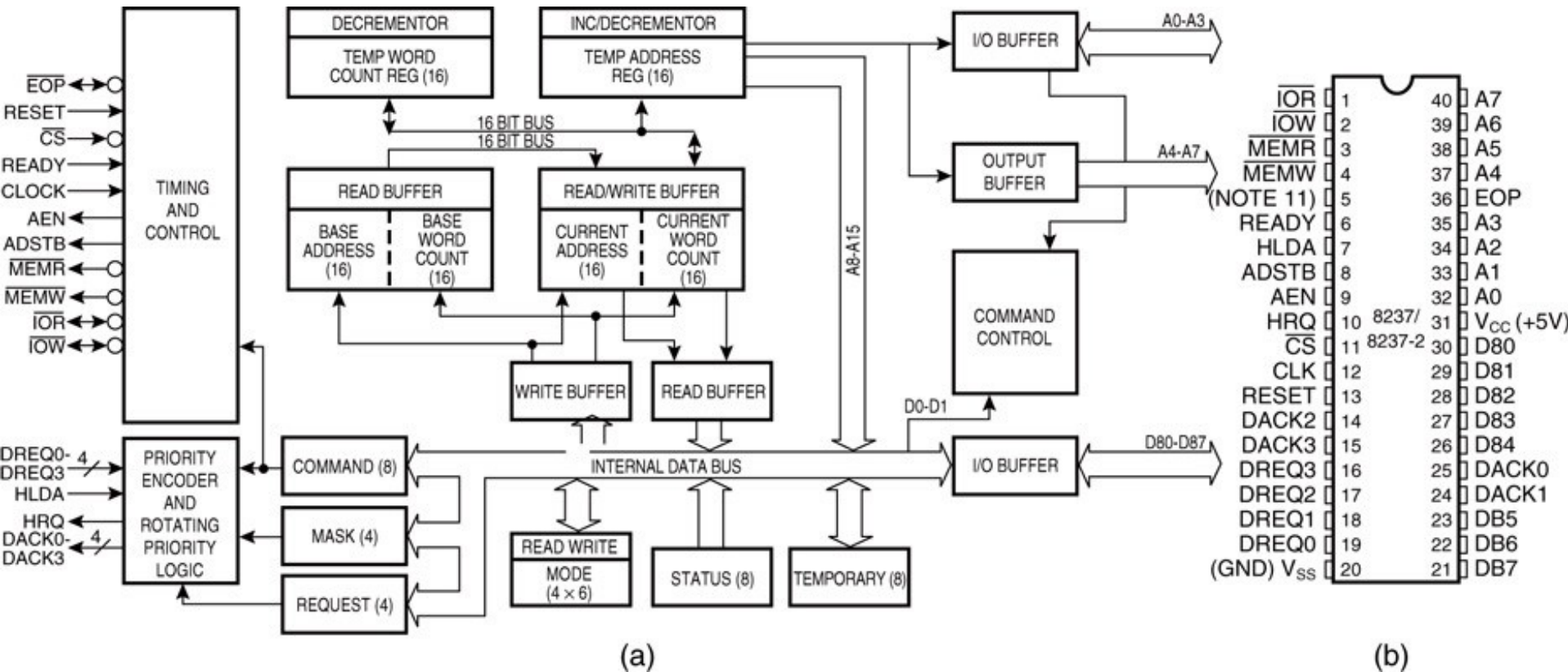
- Data transfer **speed** is determined by speed of the memory device or a DMA controller.
  - if memory speed is **50 ns**, DMA transfers occur at rates up to 1/50 ns or **20 M bytes** per second
  - if the DMA controller functions at a maximum rate of 15 MHz with 50 ns memory, maximum transfer rate is 15 MHz because the DMA controller is slower than the memory
- In many cases, the DMA controller **slows** the speed of the system when transfers occur.

- The switch to serial data transfers in modern systems has made DMA is less important.
- The serial **PCI Express** bus transfers data at rates exceeding DMA transfers.
- The **SATA** (**serial ATA**) interface for disk drives uses serial transfers at the rate of **300 Mbps**
  - and has replaced DMA transfers for hard disks
- Serial transfers on main-boards between components using can approach **20 Gbps** for the **PCI Express** connection.

# 13–2 THE 8237 DMA CONTROLLER

- The 8237 **supplies** memory & I/O with control signals and memory address information during the DMA transfer.
  - actually a special-purpose microprocessor whose job is high-speed data transfer between memory and I/O
- Figure 13–3 shows the pin-out and block diagram of the 8237 **programmable** DMA controller.

**Figure 13–3** The 8237A-5 programmable DMA controller. (a) Block diagram and (b) pin-out. (Courtesy of Intel Corporation.)



- 8237 is not a discrete component in modern microprocessor-based systems.
  - it appears within many system controller **chip sets**
- 8237 is a **four-channel device compatible** with 8086/8088, adequate for small systems.
  - **expandable** to any number of DMA channel inputs
- 8237 is capable of DMA transfers at rates up to **1.6M bytes per second**.
  - each channel is capable of addressing a **full 64K-byte section** of memory and transfer up to 64K bytes with a single programming



# 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

# 8237 Pin Definitions

$\overline{\text{CS}}$

- **Chip select** enables 8237 for programming.
- The  $\overline{\text{CS}}$  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<sub>0</sub>–DREQ<sub>3</sub>

- **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<sub>0</sub>–DB<sub>7</sub>

- **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

## 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



# 8237 Pin Definitions

$A_0 - A_3$

- 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.

# 8237 Pin Definitions

## $\text{DACK}_0$ – $\text{DACK}_3$

- **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

# 8237 Pin Definitions

## 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.

---

## MEMR

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

# 8237 Pin Definitions

---

## 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

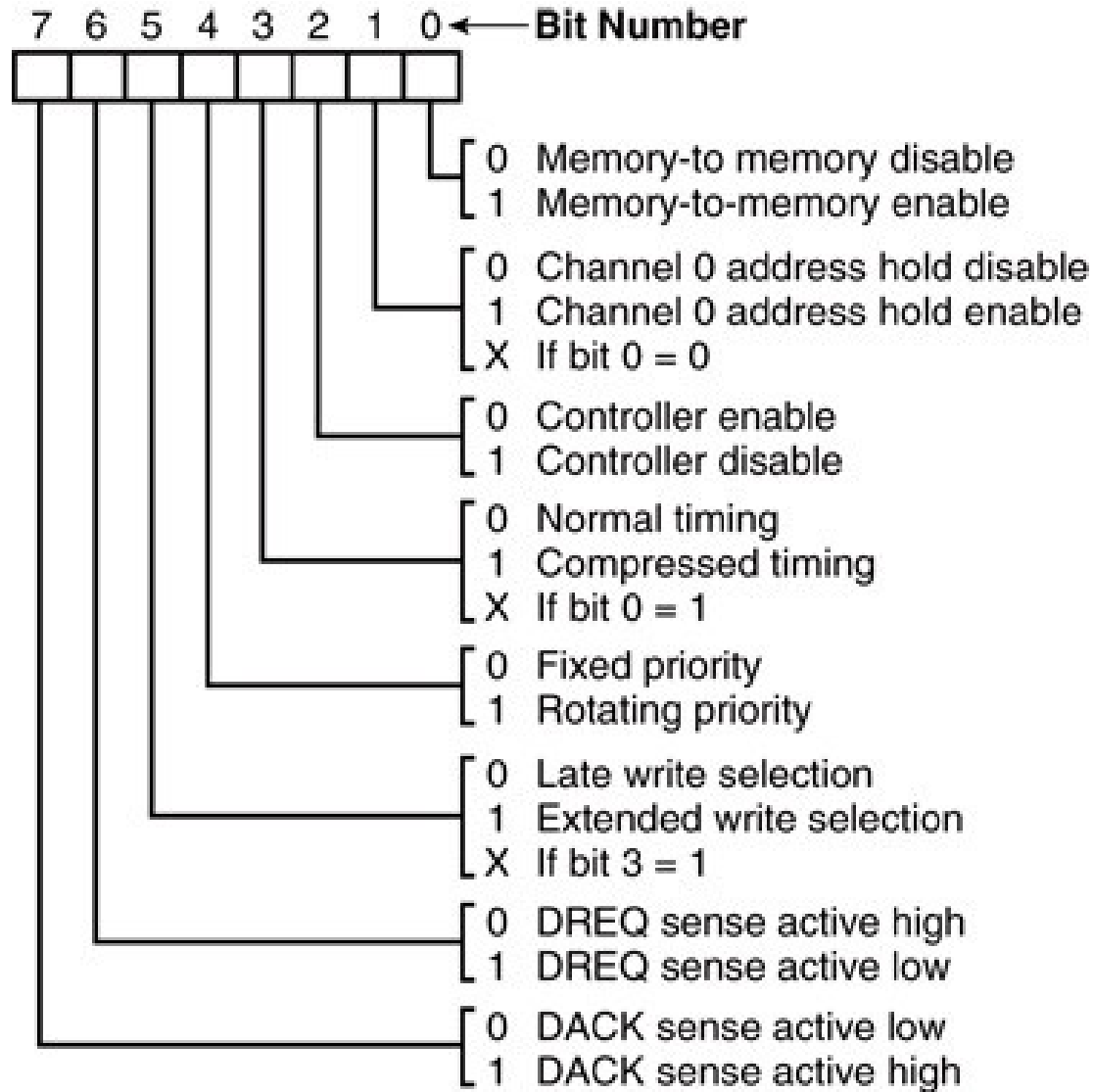
- The **base address** (BA) and **base word count** (BWC) registers are used when **auto-initialization** is selected for a channel.
- In **auto-initialization mode**, these registers are used to reload the CAR and CWCR after the DMA action is completed.
  - allows the same count and address to be used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

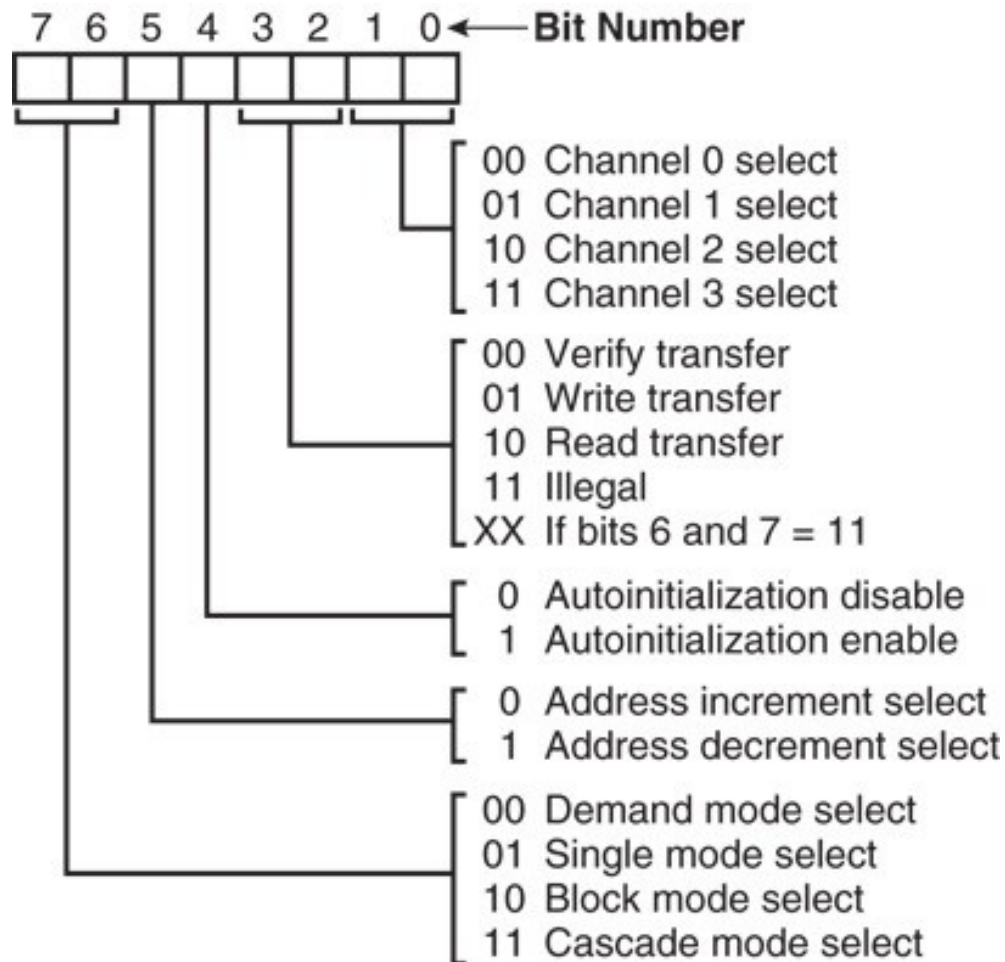


# 8237 Internal Registers

## 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.)



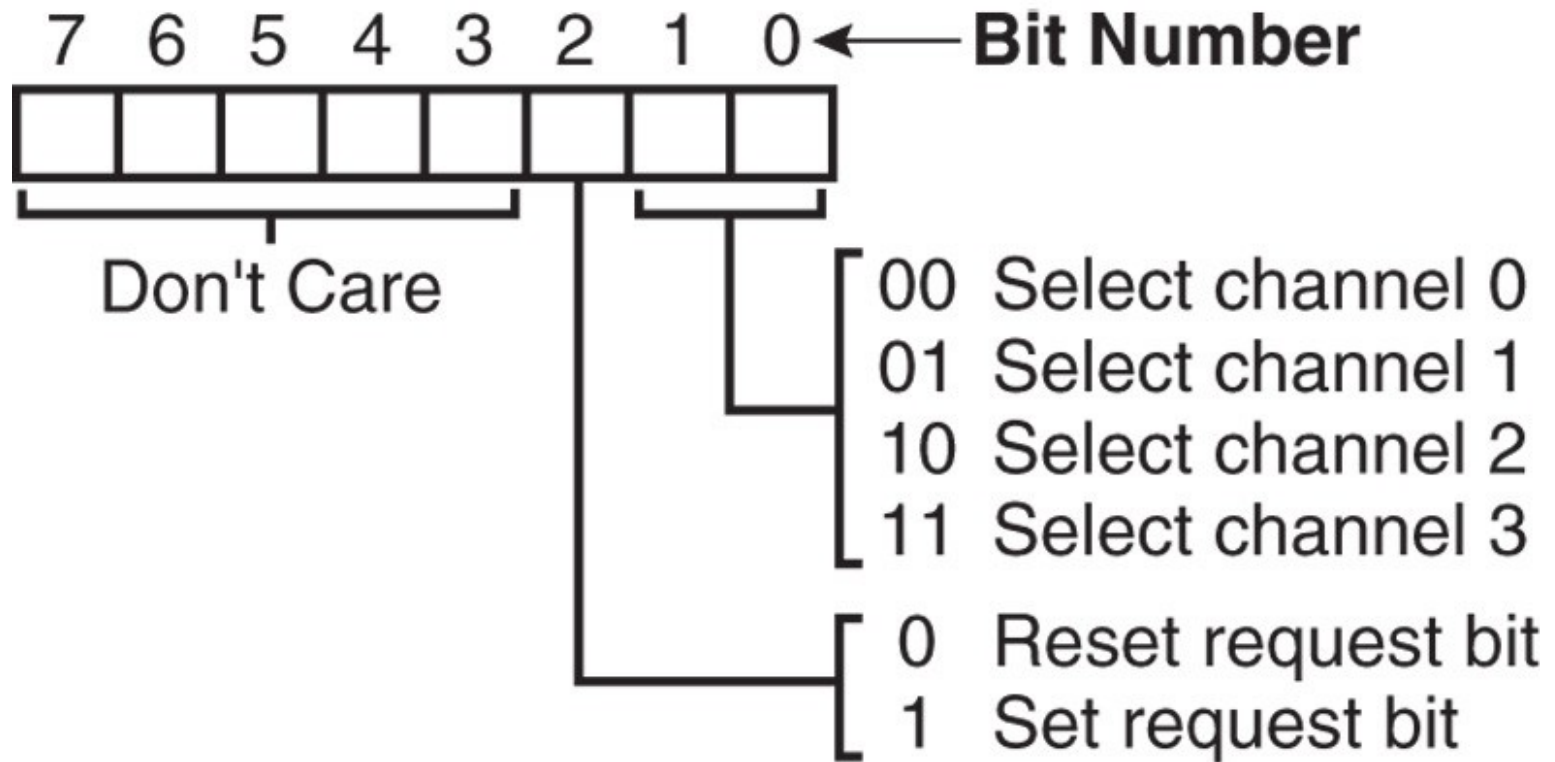
# 8237 Internal Registers

## 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.)

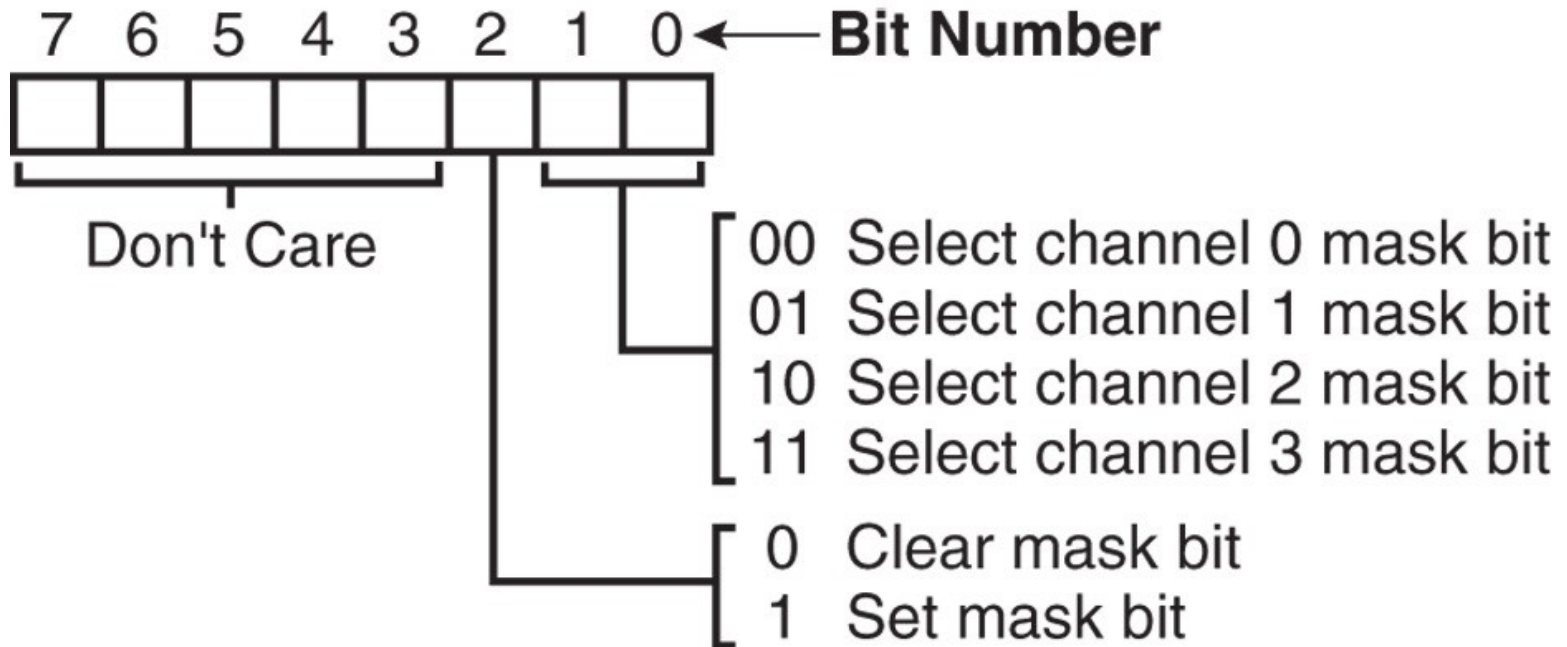


# 8237 Internal Registers

## 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.)

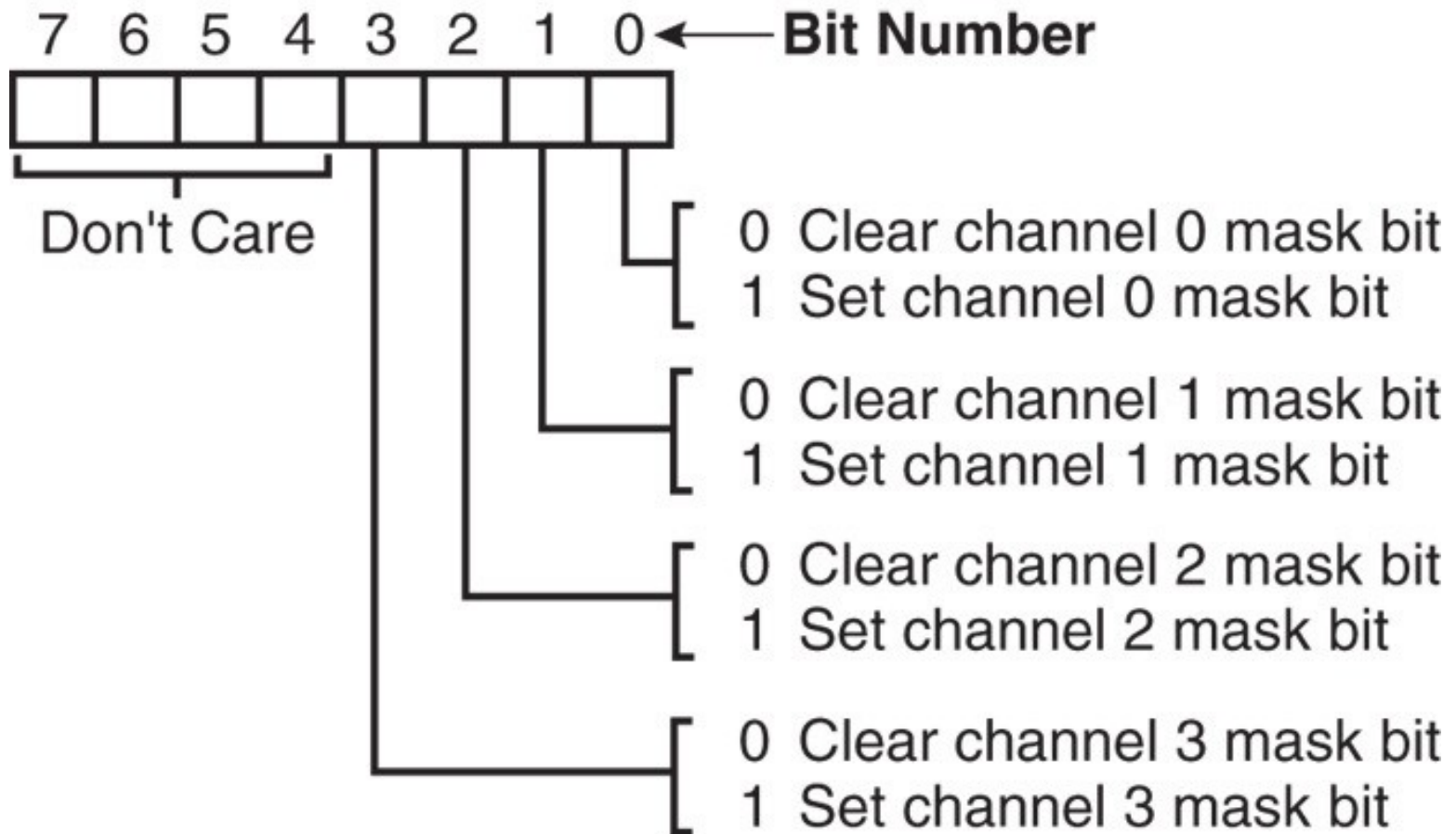


# 8237 Internal Registers

## 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.)

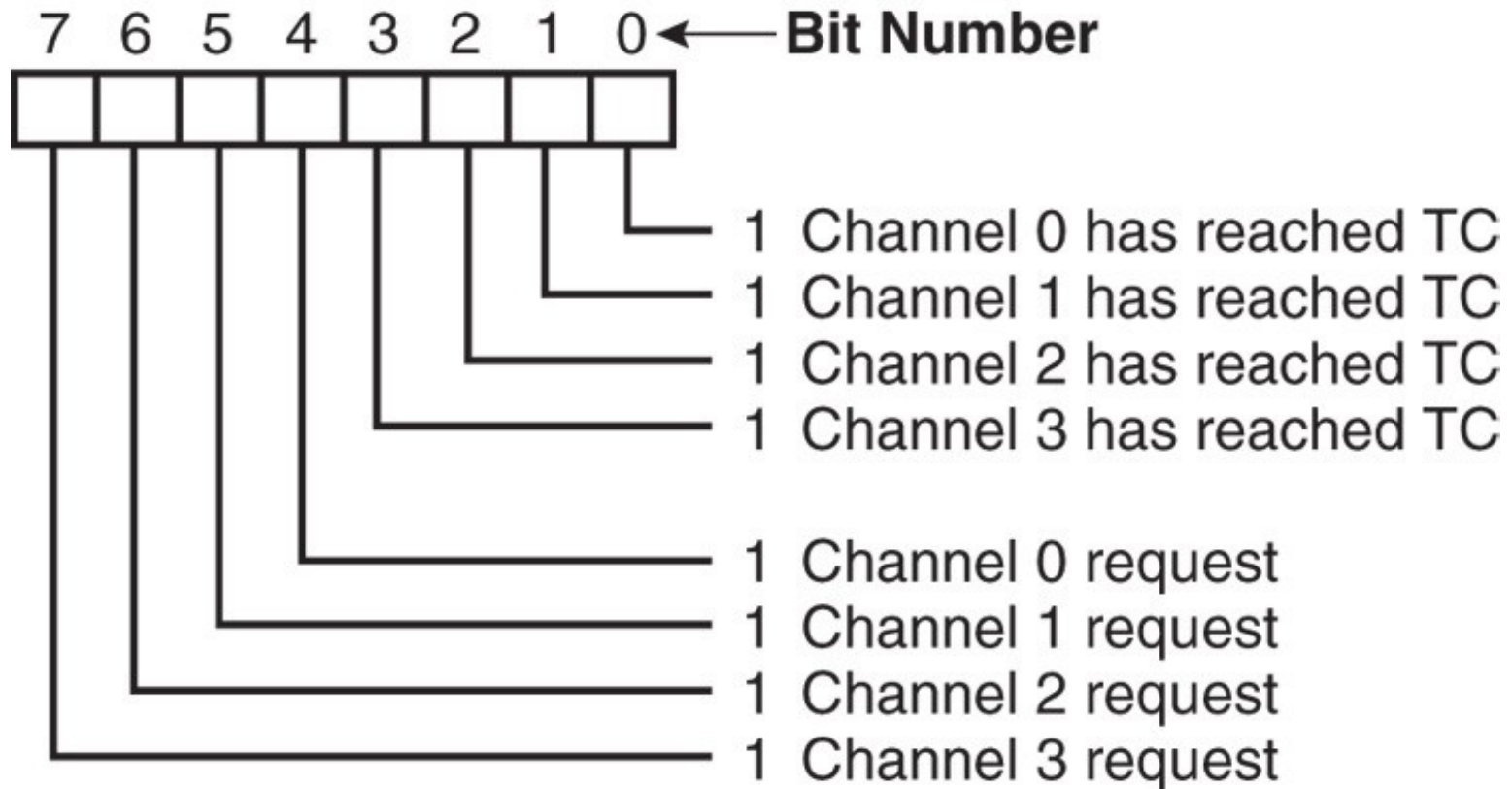


# 8237 Internal Registers

## 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.)



# Software Commands

- Three software commands are used to control the operation of the 8237.
- These commands do not have a binary bit pattern, as do various control registers within the 8237.
  - a simple output to the correct port number enables the software command
- Fig 13–10 shows I/O port assignments that access all registers and the software commands.



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

Signals						Operation
A3	A2	A1	A0	$\overline{\text{IOR}}$	$\overline{\text{IOW}}$	
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 13–11 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 13–11** 8237A-5 DMA channel I/O port addresses. (Courtesy of Intel Corporation.)

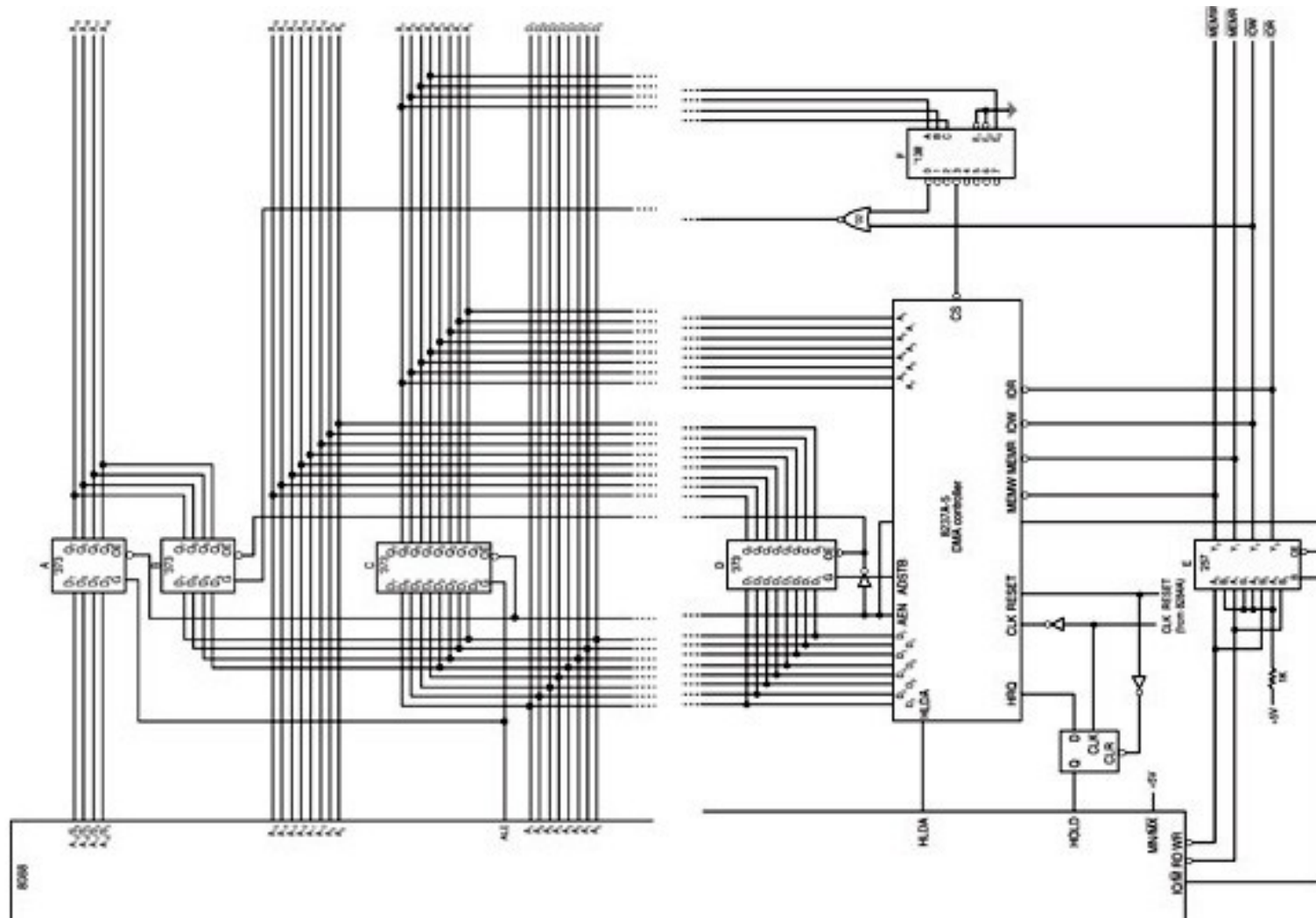
Channel	Register	Operation	Signals							Internal Flip-Flop	Data Bus DB0-DB7
			CS	IOR	IOW	A3	A2	A1	A0		
0	Base and Current Address	Write	0	1	0	0	0	0	0	0	A0-A7 A8-A15
			0	1	0	0	0	0	0	1	
	Current Address	Read	0	0	1	0	0	0	0	0	A0-A7 A8-A15
			0	0	1	0	0	0	0	1	
	Base and Current Word Count	Write	0	1	0	0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

- 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

# The 8237 Connected to the 80X86

- The address enable (AEN) output of 8237 controls the output pins of the latches and outputs of the 74LS257 (E).
  - during normal operation ( $AEN=0$ ), latches A & C and the multiplexer (E) provide address bus bits  $A_{19}-A_{16}$  and  $A_7-A_0$
- See Figure 13-12.

**Figure 13–12** Complete 8088 minimum mode DMA system.





- The multiplexer provides the system control signals as long as the 80X86 is in control of the system.
  - during a DMA action (AEN=1), latches A & C are disabled along with the multiplexer (E)
  - latches D and B now provide address bits  $A_{19}-A_{16}$  and  $A_{15}-A_8$
- Address bus bits  $A_7-A_0$  are provided directly by the 8237 and contain part of the DMA transfer address.
- The DMA controller provides control signals.

# Memory-to-Memory Transfer with the 8237

- Memory-to-memory transfer is much more powerful than the automatically repeated MOVSB instruction.
  - most modern chip sets do not support the memory-to-memory feature
- 8237 requires only  $2.0 \mu\text{s}$  per byte, which is over twice as fast as a software data transfer.
- This is not true if an 80386, 80846, or Pentium is in use in the system.

# ***Sample Memory-to-Memory DMA Transfer***

- Suppose contents of memory locations 10000H–13FFFH are to be transferred to locations 14000H–17FFFH.
  - accomplished with a repeated string move instruction or with the DMA controller
- Example 13–1 shows the software required to initialize the 8237 and program latch B in Figure 13–12 for this DMA transfer.

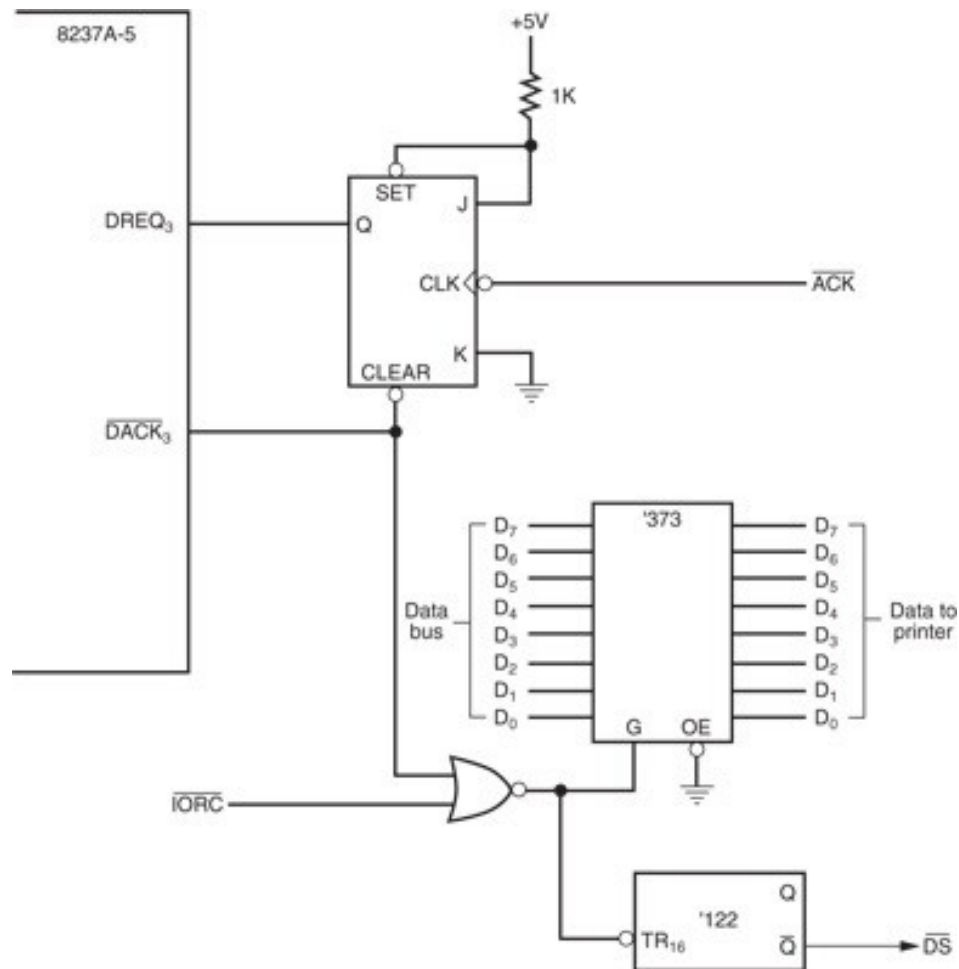
# ***Sample Memory Fill Using the 8237***

- To fill an area of memory with the same data, the channel 0 source register is programmed to point to the same address throughout the transfer.
  - accomplished with the channel 0 hold mode
- The controller copies the contents of this single memory location to an entire block of memory addressed by channel 1.
- This has many useful applications.

# DMA-Processed Printer Interface

- Fig 13–13 illustrates the hardware added to Fig13–12 for a DMA-controlled printer interface.
  - software to control this interface is simple as only the address of the data and number of characters to be printed are programmed
- Once programmed, the channel is enabled, and the DMA action transfers a byte at a time to the printer interface.
  - each time a printer ~~ACK~~ signal is received

**Figure 13–13** DMA-processed printer interface.



# SUMMARY

- The HOLD input is used to request a DMA action, and the HLDA output signals that the hold is in effect.
- When a logic 1 is placed on the HOLD input, the micro-processor (1) stops executing the program; (2) places its address, data, and control bus at their high-impedance state; and (3) signals that the hold is in effect by placing a logic 1 on the HLDA pin.

# SUMMARY

*(cont.)*

- A DMA read operation transfers data from a memory location to an external I/O device.
- A DMA write operation transfers data from an I/O device into the memory.
- Also available is a memory-to-memory transfer that allows data to be transferred between two memory locations by using DMA techniques.



# SUMMARY

(*cont.*)

- The 8237 direct memory access (DMA) controller is a four-channel device that can be expanded to include an additional channel of DMA.
- Disk memory comes in the form of floppy disk storage that is found as 3-1/2" micro-floppy disks.

# SUMMARY

(*cont.*)

- Floppy disk memory data are stored using NRZ (non-return to zero) recording.
- This method saturates the disk with one polarity of magnetic energy for a logic 1 and the opposite polarity for a logic 0.
- In either case, the magnetic field never returns to 0.
- This technique eliminates the need for a separate erase head.

# SUMMARY

(*cont.*)

- The MFM scheme records a data pulse for a logic 1, no data or clock for the first logic 0 of a string of zeros, and a clock pulse for the second and subsequent logic 0 in a string of zeros.
- The RLL scheme encodes data so that 50% more information can be packed onto the same disk area. Most modern disk memory systems use the RLL encoding scheme.

# SUMMARY

(*cont.*)

- Video monitors are either TTL or analog.
- The TTL monitor uses two discrete voltage levels of 0 V and 5.0 V.
- The analog monitor uses an infinite number of voltage levels between 0.0 V and 0.7 V.
- The analog monitor can display an infinite number of video levels, while the TTL monitor is limited to two video levels.

# SUMMARY

(*cont.*)

- The color TTL monitor displays 16 different colors.
- This is accomplished through three video signals (red, green, and blue) and an intensity input.
- The analog color monitor can display an infinite number of colors through its three video inputs.

# SUMMARY

- In practice, the most common form of color analog display system (VGA) can display 16 M different colors.
- The video standards found today include VGA (640 ´ 480), SVGA (800 ´ 600), and EVGA or XVGA (1024 ´ 768).
- In all three cases, the video information can be 16M colors.