The x86 PC

assembly language, design, and interfacing

fifth edition

Prentice Hall

Dec Hex Bin
15 F 00001111

ORG ; FIFTEEN

DIRECT MEMORY ACCESS AND DMA CHANNELS IN x86 PC

The x86 PC

assembly language, design, and interfacing

fifth edition

MUHAMMAD ALI MAZIDI
JANICE GILLISPIE MAZIDI
DANNY CAUSEY

OBJECTIVES

this chapter enables the student to:

- Describe the concept of DMA, direct memory accessing.
- List the pins of the 8237A DMA chip and describe their functions.
- Explain how bus arbitration is achieved between DMA and the CPU.
- Explain how the channels of the 8237 are use in the PC.
- List the DMA signals in the ISA bus.

15.1: CONCEPT OF DMA

- There is often need to transfer a many bytes between memory & peripherals like disk drives.
  - Using the microprocessor to transfer the data is too slow.
    - Data must be fetched to the CPU, then sent to its destination.
- The Intel 8237 DMAC (direct memory access controller) chip functions to bypass the CPU.
  - It provides a direct connection between peripherals and memory, transferring the data as fast as possible.
    - Where 8237 can transfer a byte between a peripheral & memory in 4 clocks, the 8088 would take 39 clocks.

15.1: CONCEPT OF DMA
bus sharing

- The data bus, address bus, or control bus can be used either by the main x86 CPU or the 8237 DMA.
  - Since x86 has primary control, it must give permission to DMA to use them.
- When DMA needs the buses, it sends a HOLD signal to the CPU, and the CPU responds with a HLDA (hold acknowledge) signal.
  - Indicating the DMA can use the buses.

15.1: CONCEPT OF DMA
bus sharing

- While DMA uses the buses, the CPU is idle, and when the CPU uses the bus, DMA is sitting idle.
  - After DMA finishes, it makes HOLD go *low* & the CPU will regain control over the buses

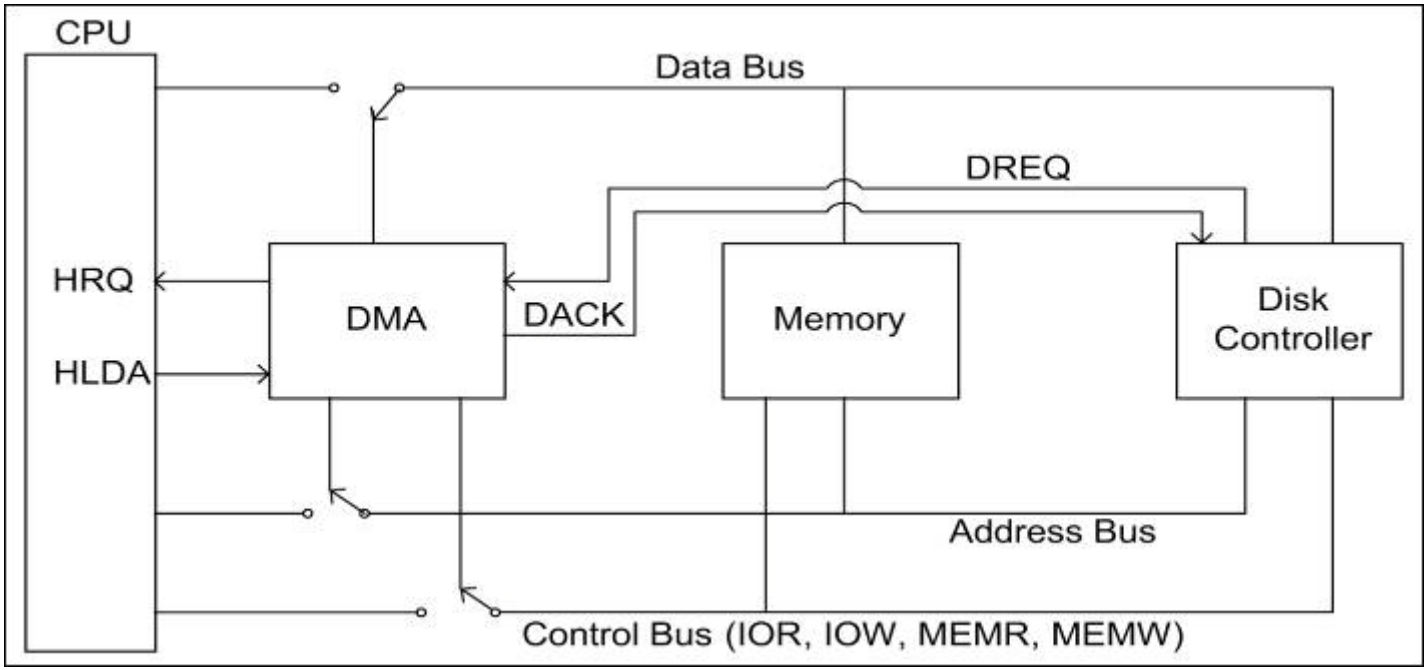


Fig. 15-1
DMA Usage of System Bus

15.1: CONCEPT OF DMA
steps involved in a DMA transfer

- DMA can only *transfer* information.
  - It cannot decode and execute instructions.
- When the CPU receives a HOLD request from DMA, it finishes the present bus cycle (but not necessarily the present instruction) before it hands over control of the buses to the DMA.
- To transfer a block of data from memory to I/O, DMA must know:
  - The address of the beginning of the data block. (address of the first byte of data)
  - The number of bytes (count) it needs to transfer.

15.1: CONCEPT OF DMA
steps involved in a DMA transfer

- DMA Transfer Steps:
  - 1. A peripheral device (like the disk controller) will request DMA service by pulling DREQ (DMA request) *high*.
  - 2. DMA puts a *high* on its HRQ (hold request), signaling the CPU through its HOLD pin that it needs to the buses.
  - 3. The CPU finishes the present bus cycle & responds to DMA by putting *high* on HLDA (hold acknowledge).
    - Telling the 8237 DMA it can use the buses to perform its task.
    - HOLD must remain *active-high* while DMA performs its task.
  - 4. DMA will activate DACK (DMA acknowledge), which tells the peripheral device it will start to transfer the data.

15.1: CONCEPT OF DMA
steps involved in a DMA transfer

- DMA Transfer Steps:
  - 5. DMA starts to transfer data from memory to the I/O peripheral by putting the address of the first byte of the block on the address bus and activating MEMR.
    - Reading the byte from memory into the data bus; it then activates IOW to write the data to the peripheral.
    - DMA decrements the counter, increments the address pointer & repeats the process until the count reaches zero.
  - 6. After the DMA has finished, it will deactivate HRQ, signaling the CPU that it can regain control over its buses.

15.2: 8237 DMA CHIP PROGRAMMING

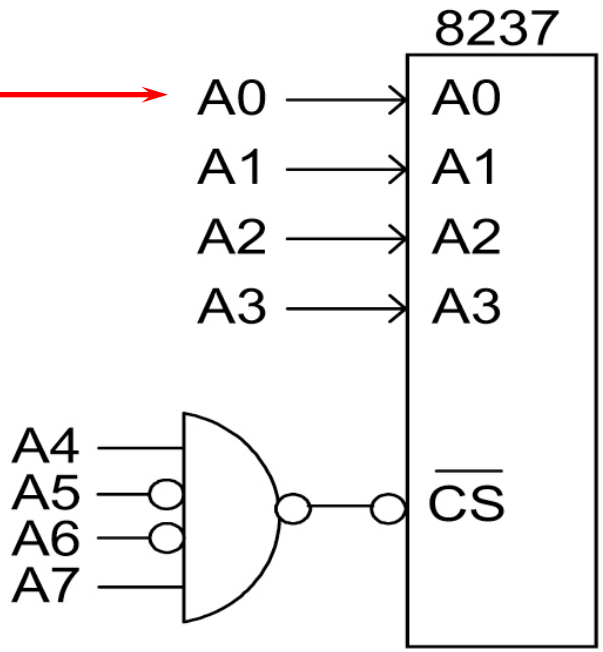
- The 40-pin Intel 8237 DMA controller chip has four data transfer channels, each used for one device.
  - Only one device at a time can use the DMA.
- With every channel are two associated signals:
  - **DREQ** (DMA request) - an input to DMA from the peripheral device.
  - **DACK** (DMA acknowledge) - an output signal from the 8237 going to the peripheral device.
- HOLD & HLDA connect to HOLD & HLDA of x86.
  - Four channels from four different devices can request bus use, but DMA decides who gets control based on how its priority register has been programmed.

15.2: 8237 DMA CHIP PROGRAMMING

- Every channel of the 8237 DMA must be initialized separately for the address of the data block and the count (the size of the block) before it can be used.
  - After initialization, each channel can be enabled and controlled with the use of a control word.
- Many modes of operation can be programmed into the 8237's internal registers.
  - Accessed by four address pins, A0–A3.
    - Along with the CS (chip select) pin.

15.2: 8237 DMA CHIP PROGRAMMING

Internal 8237 register addresses for each channel, and how they are generated.



Example 15-1

Find the port addresses for the base address and word count of each channel of the 8237 for Figure 15-2 (CS is activated by A7–A4 = 1001 binary).

Solution:

From Table 15-1, one can get the addresses found in Table 15-3.

15.2: 8237 DMA CHIP PROGRAMMING

Internal 8237 register addresses for each channel, and how they are generated.

8237 Internal Addresses for Writing Transfer Addresses and Counts

CH	Register	R/W	$\overline{CS}$	$\overline{IOR}$	$\overline{IOW}$	A3	A2	A1	A0
0	Base and current address	Write	0	1	0	0	0	0	0
	Current address	Read	0	0	1	0	0	0	0
1									

Table 15-3: 8237 Address Selection for Example 15-1

Binary address A7 A6 A5 A4	Binary address A3 A2 A1 A0	Hex Address	Function	Read/ Write
1 0 0 1	0 0 0 0	90	CHAN0 memory address register	R/W
1 0 0 1	0 0 0 1	91	CHAN0 count register	R/W
	0 1 0	92	CHAN1 memory address register	R/W
	0 1 1	93	CHAN1 count register	R/W
	1 0 0	94	CHAN2 memory address register	R/W
	1 0 1	95	CHAN2 count register	R/W
1 0 0 1	0 1 1 0	96	CHAN3 memory address register	R/W

See the tables on pages 405 - 406 of your textbook.

Example 15-1

Find the port addresses for the base address and word count of each channel of the 8237 for Figure 15-2 (CS is activated by A7–A4 = 1001 binary).

Solution:

From Table 15-1, one can get the addresses found in Table 15-3.

15.2: 8237 DMA CHIP PROGRAMMING

- Two sets of information needed to program a channel of the 8237 DMA to transfer data are:
  - The address of the first byte of data to be transferred. (base address)
  - How many bytes of data are to be transferred. (word count)
- For set 1, the channel's memory address register must be programmed.
  - The 8237 memory address register 16 bits, and the data bus is 8 bits.
    - One byte at a time, consecutively, is sent in to the same port address.

15.2: 8237 DMA CHIP PROGRAMMING

- For set 2, the channel count register is programmed.
  - The count can go as high as FFFFH.

Since the count register is 16 bits and the DMA data bus is only 8 bits, it takes two consecutive writes to program that register, as shown in Example 15-2.

See the entire example on page 406 of your textbook.

Example 15-2

Assume that channel 2 of the DMA in Example 15-1 is to transfer a 2K (2048) byte block of data from memory locations starting at 53400H. Use the port addresses of Example 15-1 for the DMA to program the memory address register and count register of channel 2.

Solution:

The port addresses for the channel 2 memory address register and count register in Example 15-1 are 94H and 95H, respectively. The initialization will look as follows:

```
MOV   AX,3400H    ;load lower 4 digits of start address
OUT   94H,AL      ;send out the low byte of the address
MOV   AL,AH       ;
OUT   94H,AL      ;send out the high byte of the address
MOV   AX,2048     ;load block size into AX
OUT   95H,AL      ;send out the low byte of the count
MOV   AL,AH       ;
OUT   95H,AL      ;send out the high byte of the count
```

The contents of the memory address and count registers can be read in the same manner (low byte first, then high byte) to monitor these registers at any time. From looking at the above program one might ask, since the system address bus is 20 bits and the memory address is 53400H, why does this program use 16-bit addresses? This is a limitation of the 8237 DMA. In the 8237, not only is the register holding the address of the block 16 bits, but in addition there are only 16 address pins that carry the addresses. The x86 PC solves this problem by using an external *n*-bit register to hold the upper bits of the addresses.

15.2: 8237 DMA CHIP PROGRAMMING
internal control registers

- One set of control/command registers is used by all 8237 channels.
  - These registers are shown in Table 15-1 on page 405.
  - To understand how to access those registers, review Example 15-3 on page 407.

15.2: 8237 DMA CHIP PROGRAMMING
internal control registers

Table 15-2: 8237 Internal Addresses for Commands/Status

A3	A2	A1	A0	$\overline{IOR}$	$\overline{IOW}$	Operation
1	0	0	0	0	1	Read status regis
			0	1	0	Write command r
			1	0	1	Illegal
1	0	0	1	1	0	Write request reg

See pages 405 - 407 of your textbook.

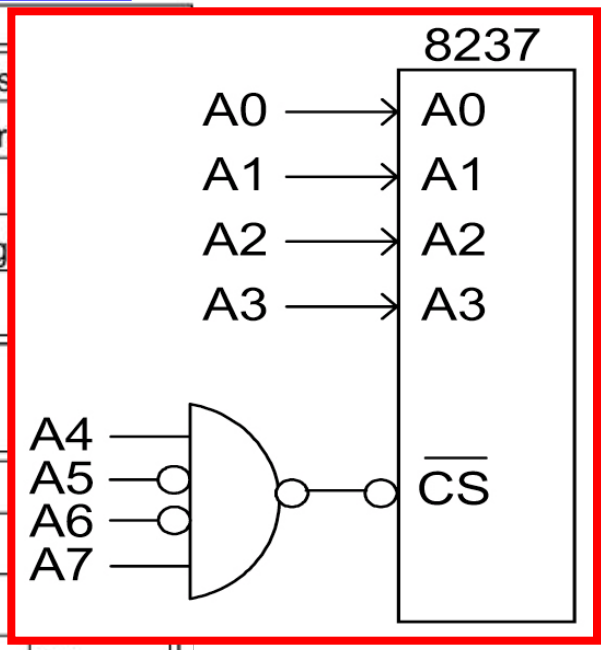


Table 15-4: Address Selection for Example 15-3

Binary address								Hex	Register Name	
A7	A6	A5	A4	A3	A2	A1	A0			
1	0	0	1	1	0	0	0	98	Status/command register	
1	0	0	1	1	0	0	1	99	Request register	
1	0	0	1	1	0	1	0	9A	Single mask register bit	
1	0	0	1	1	0	1	1	9B	Mode register	W

Example 15-3

Use the circuit in Example 15-1 to find the address of the 8237 DMA control registers.

Solution:

Using Table 15-2 and substituting for A7–A3 gives the information in Table 15-4.

15.2: 8237 DMA CHIP PROGRAMMING
command register

- An 8-bit register used for controlling the operation of the 8237.
  - It must be programmed (written into) by the CPU.
  - It is cleared by the RESET signal from the CPU.
    - Or the master clear instruction of the DMA.

15.2: 8237 DMA CHIP PROGRAMMING
command register

8237 is capable of transferring data…

From a peripheral device to memory. (reading from disk)

From memory to a peripheral device (writing a file to disk)

From memory to memory. (Shadow RAM)

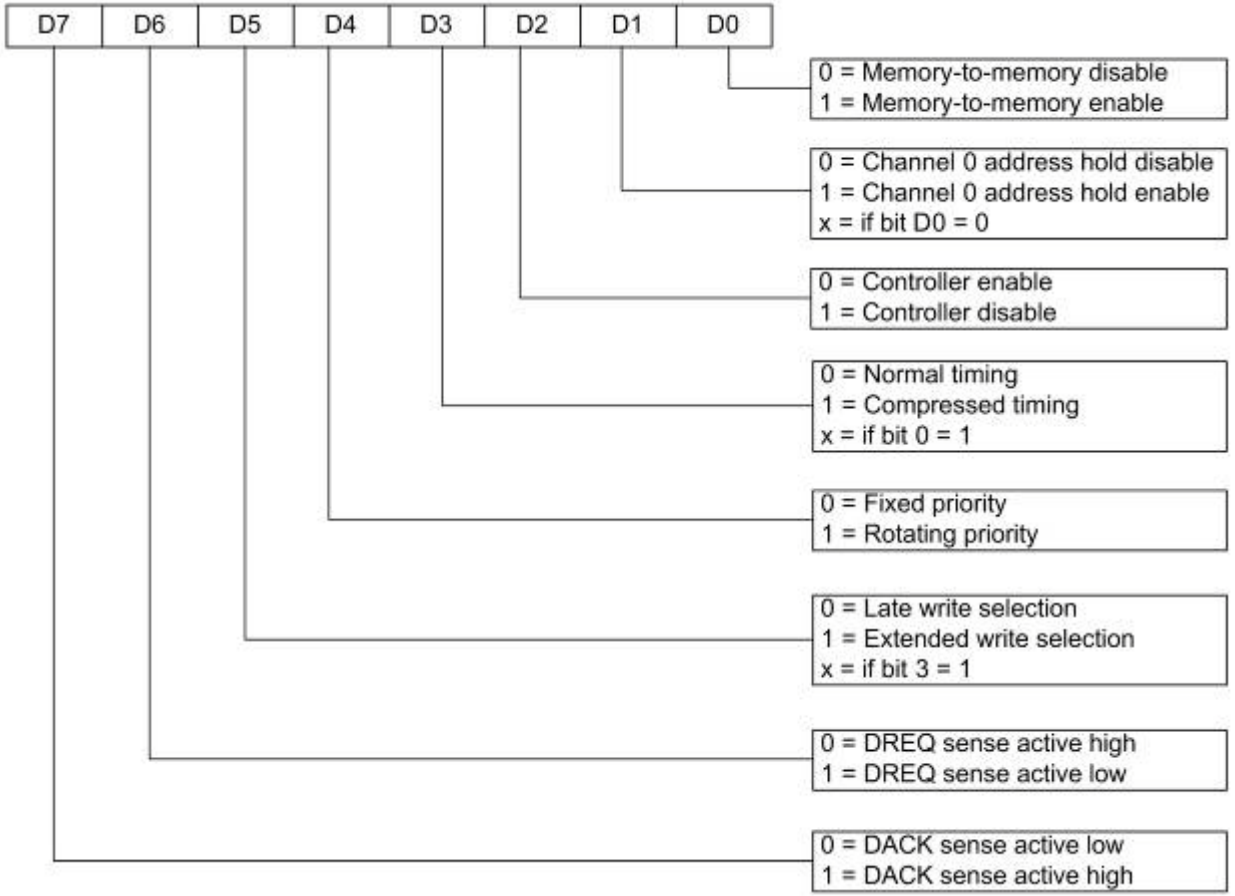


Fig. 15-3 8237 Command Register Format

15.2: 8237 DMA CHIP PROGRAMMING
command register bits

- **D0** - option to use only channels 0 & 1 for transfer of a block of data from memory to memory.
  - Channel 0 is the source; channel 1 the destination.
- **D1** - used only when the memory-to-memory option is enabled and can be used to disable the memory incrementation/decrementation of channel 0.
  - In order to write a fixed value into a block of memory.
- **D2** - used to enable or disable DMA.
- **D3** - option to choose between normal memory cycle of 4 clock pulses, and compressed timing.

15.2: 8237 DMA CHIP PROGRAMMING
command register bits

- **D4** - option of using the four channels on fixed
- **D5** - allows time for the write signal to be extended for slow devices.
- **D6** - option of programming DREQ activation level.
  - It can be an *active-high* or *active-low* signal.
- **D7** - option of programming DACK activation level.
  - It can be an *active-high* or *active-low* signal.

15.2: 8237 DMA CHIP PROGRAMMING
command register bits

- The command byte is issued to this register through port address X8H.
  - Where X is the combination provided to activate CS.

Example 15-4

Program the command register of the 8237 in Example 15-3 for the following options: no memory-to-memory transfer, normal timing, fixed priority, late write, and DREQ and DACK both active-high.

Solution:

From Figure 15-3, the command byte would be 1000 0000 = 80H and the program is

```
MOV   AL,80H ;load the command byte into AL
OUT   98H,AL      ;issue the command byte to port 98H
```

15.2: 8237 DMA CHIP PROGRAMMING
command register bits

- The command byte is issued to this register through port address X8H.
  - Where X is the combination provided to activate CS

Example 15-5

Assume that the CPU is doing some very critical processing and that the 8237 DMA should be disabled. Use the ports in Example 15-3 to show the program.

Solution:

To disable the 8237, send 0000 0100 = 04H to the command register as follows:

```
MOV   AL,04H
OUT   98H,AL
```

15.2: 8237 DMA CHIP PROGRAMMING
status register

- An 8-bit register that can only be read by the CPU via the same port address as the command register.
  - Often referred to as RO (read only).
  - This register contains various information about the operating state of the four channels.

15.2: 8237 DMA CHIP PROGRAMMING
status register

The lower four bits, **D0**–**D3**, indicate if channels 0–3 have reached their TC. (terminal count).

TC is set high when the count register has decremented to zero.

The upper four bits, **D4**–**D7**, keep count of pending DMA requests.

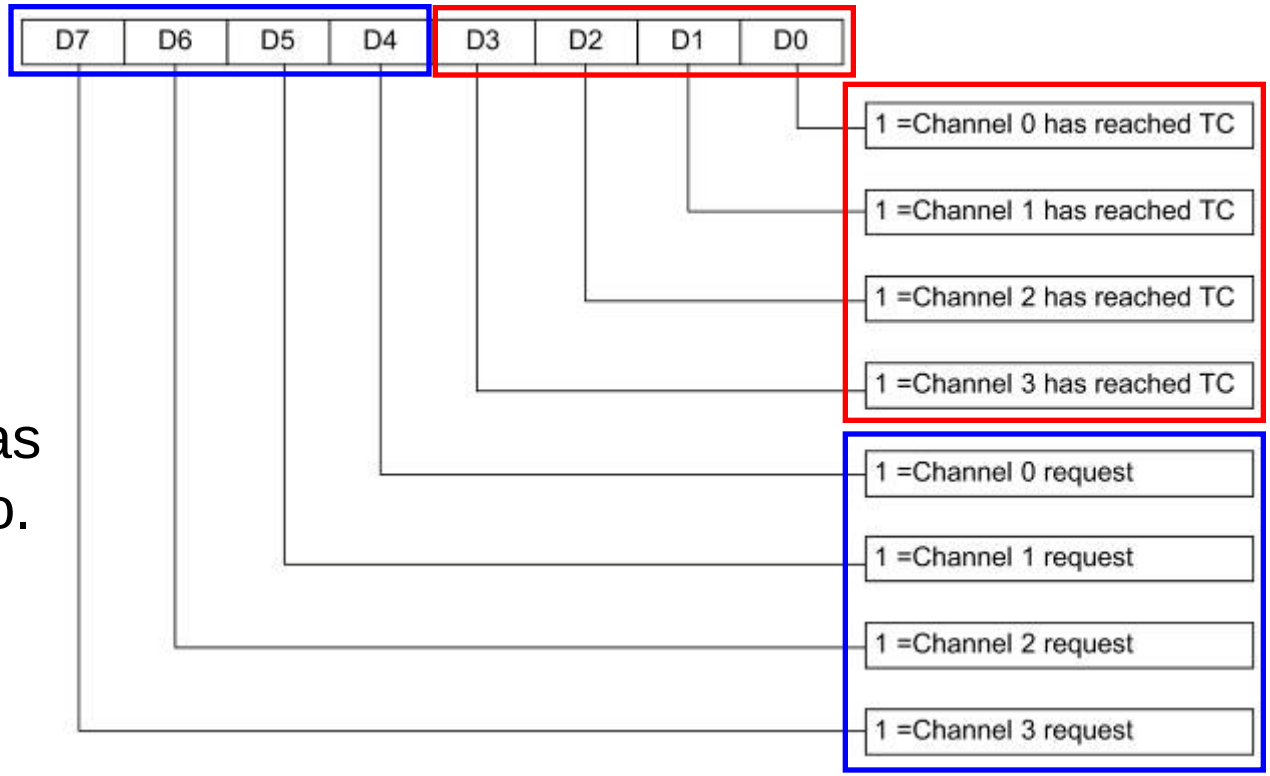


Fig. 15-4 8237 Status Register Format

15.2: 8237 DMA CHIP PROGRAMMING
mode register

- This register can only be written to by the CPU through port address XBH.
  - Where X is the address combination for CS activation.
- The lower two bits, D0/D1, are for channel selection.
  - The other 6 bits select various operation modes to be used for the channel selected by bits D0 and D1.
  - Examples 15-6 & 15-7 show programming of the mode register.

15.2: 8237 DMA CHIP PROGRAMMING
mode register

D2 & **D3** specify data transfer mode.

D4 is used for autoinitialization.

D5 gives the option to increment or decrement the address.

D6 & **D7** determine which of four mode in which the 8237 is used.

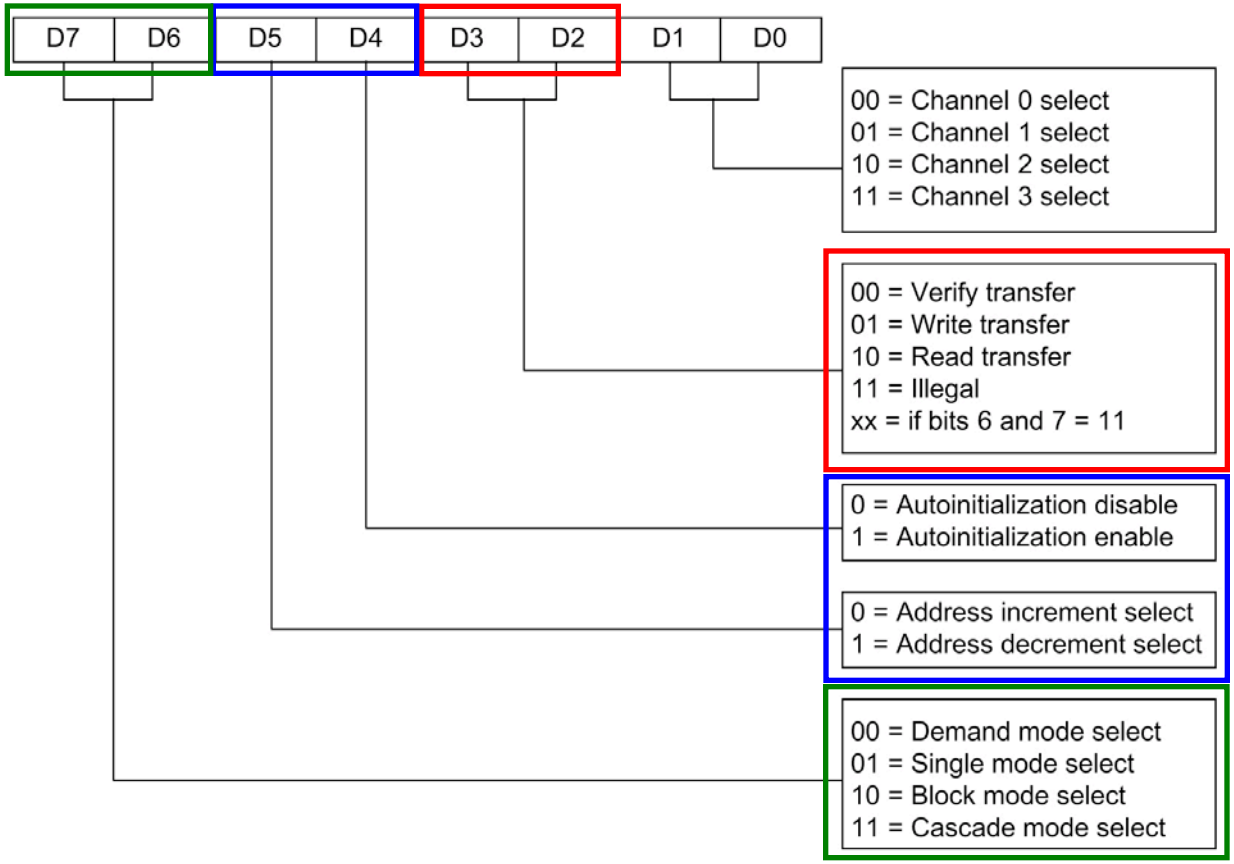


Fig. 15-5 8237 Mode Register Format

15.2: 8237 DMA CHIP PROGRAMMING
mode register – D6 & D7 mode options

- **Demand mode** - transfer of data continues until DREQ is deactivated or terminal count is reached.
- **Block mode** - same as demand, except DREQ can be deactivated after the DMA cycle starts & data transfer continues until TC state is reached.
- **Single mode** - if DREQ is held active, DMA transfers one byte of data, then allows x86 to gain control of the system bus by deactivating its HRQ for one bus cycle.
- **Cascade mode** - several DMAs can be cascaded to expand the number of DREQs to more than 4.

Programming of the 8237 mode register

Example 15-6

Program the 8237's mode register of Example 15-3 to select channel 2 to transfer from memory to I/O using autoinitialization, address increment, and single-byte transfer.

Solution:

From Figure 15-5, with these options, the mode register must have 01011010 = 5AH. The port address for the command register is 9BH, which results in

```
MOV    AL,5AH
OUT    9BH,AL
```

Example 15-7

Program the 8237 of Example 15-3 to enable channel 2.

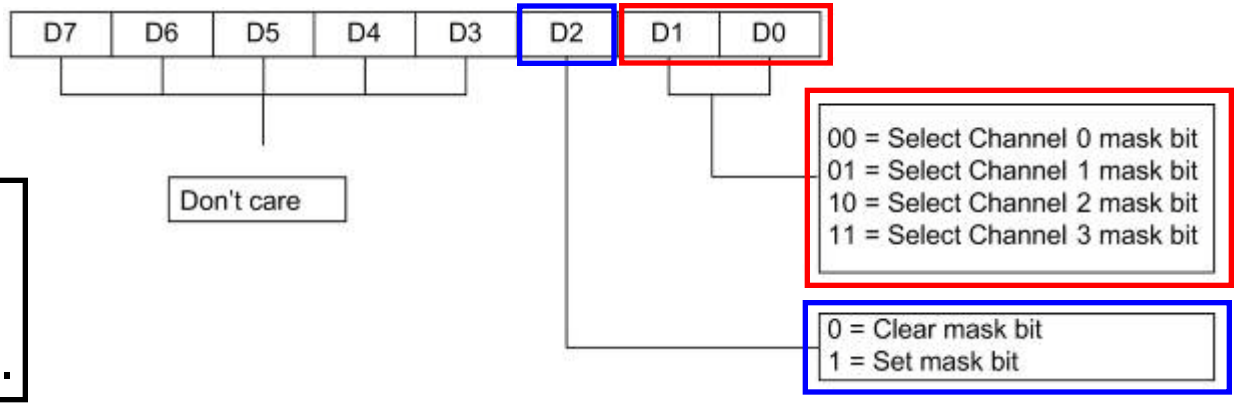
Solution:

From Figure 15-6, the value for the single mask register to enable (unmask) channel 2 is 0000 0010 = 02H and is sent to port 9AH as follows:

```
MOV    AL,02
OUT    9AH,AL
```

15.2: 8237 DMA CHIP PROGRAMMING
single mask register

- This 8-bit register can only be written to by the CPU through port address XA hex, where X is for CS.
  - **D0** & **D1** select the channel.
  - **D2** clears or sets the mask bit for that channel
    - DREQ input of a specific channel can be *masked* (disabled) or *unmasked* (enabled)..



Only one channel can be masked or unmasked at a time.

Fig. 15-6 8237 Single Mask Register Format

15.2: 8237 DMA CHIP PROGRAMMING
all mask register

- Similar to the single mask register, except that all four channels can be masked or unmasked with one write operation.
  - This register can only be written to by the CPU through the port address XFH, where X is for CS activation.

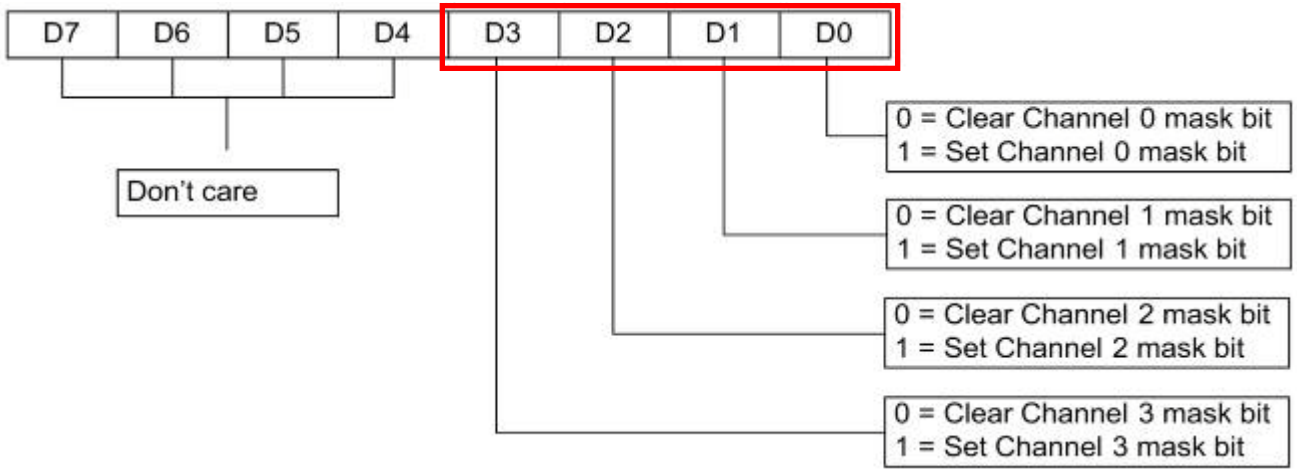


Fig. 15-7 8237 All Mask Register Format

15.2: 8237 DMA CHIP PROGRAMMING
master clear/temporary register

- Written to by the CPU through port address XDH.
  - Where X is for CS activation.
- The byte sent to this register does not matter since it simply clears the status, command, request, and mask registers and forces the DMA to the idle cycle.
  - The same as activating the hardware RESET of the 8237.

15.2: 8237 DMA CHIP PROGRAMMING
clear mask register

- Written to by the CPU only through port address XEH, where X is for CS.
  - Function is to clear the mask bits of all four channels, enabling them to accept the DMA request through the DREQs.
    - Bit patterns written to it do not matter.

15.3: 8237 DMA INTERFACING IN THE IBM PC

- The 8237 DMA has eight addresses, A0–A7.
  - A bidirectional address bus is formed by **A0**–**A3**, which sends addresses to the 8237 to select one of the 16 possible registers.

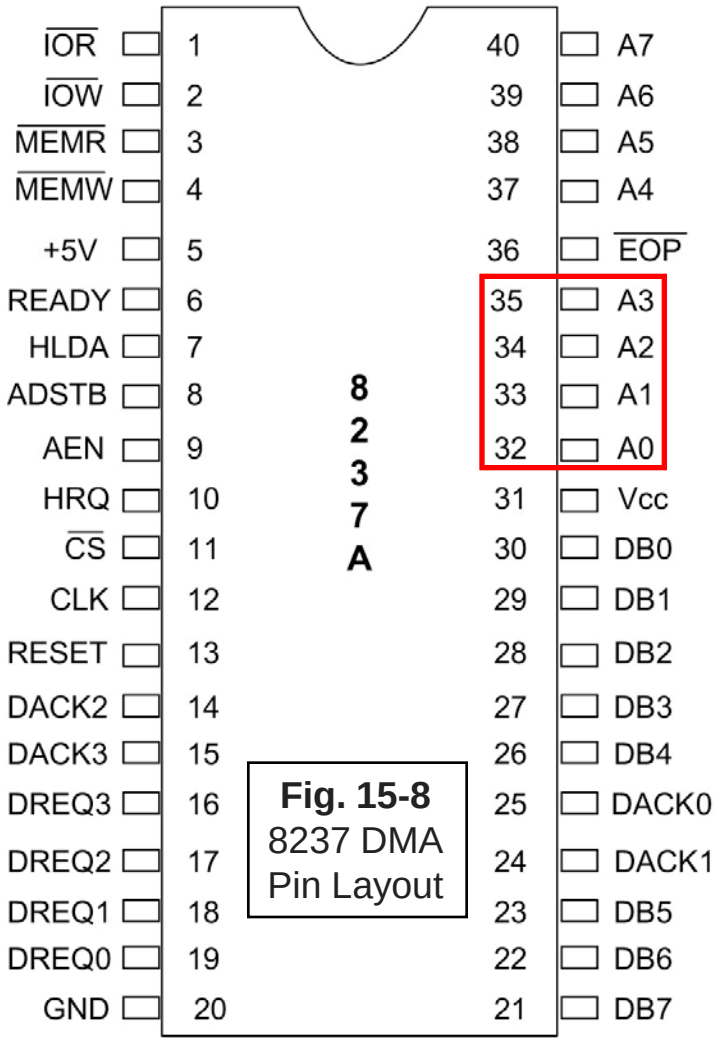


Fig. 15-8
8237 DMA Pin Layout

15.3: 8237 DMA INTERFACING IN THE IBM PC

In the IBM PC, chip select is activated by **Y0** of the 74LS138.

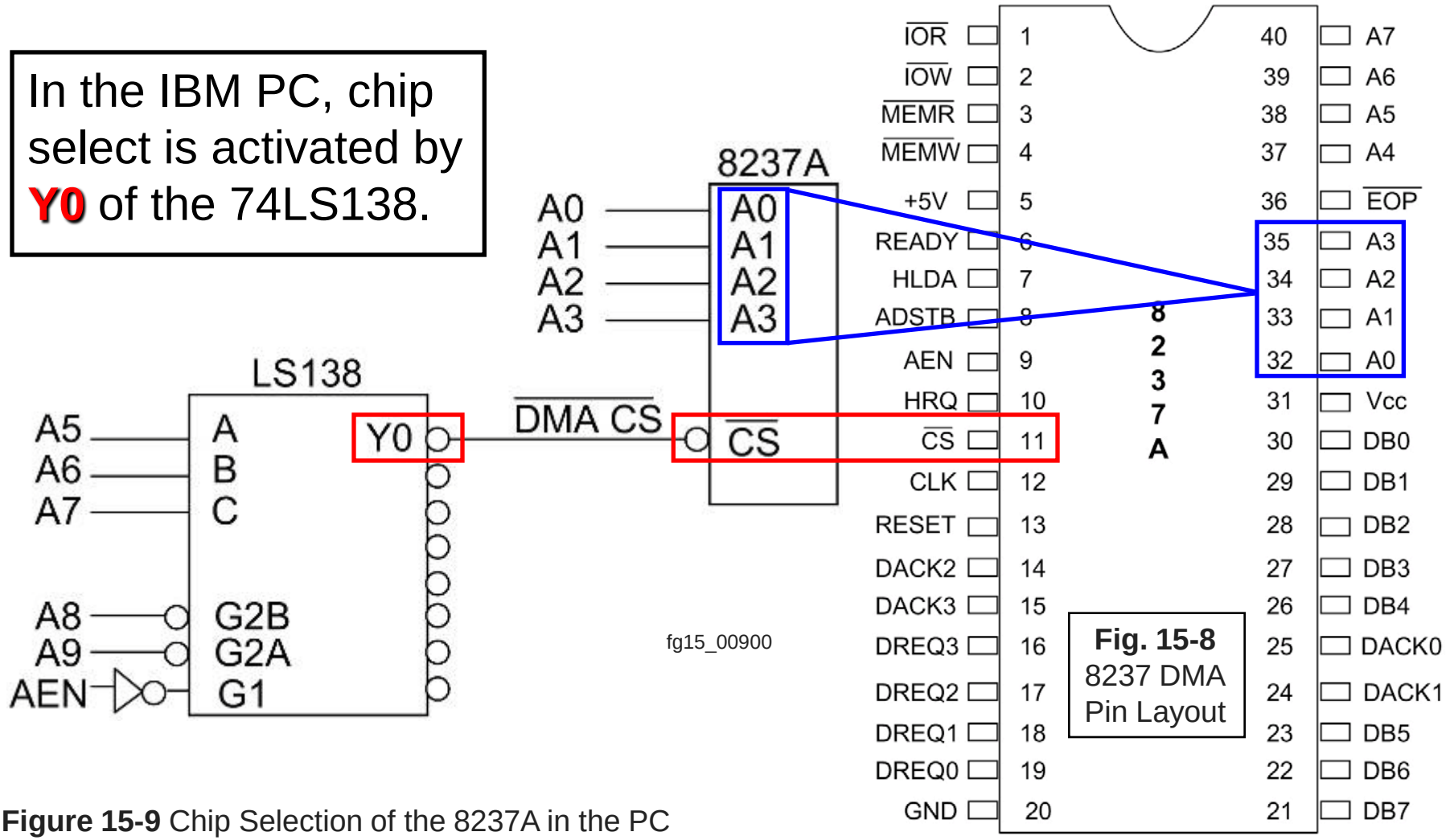


Fig. 15-8 8237 DMA Pin Layout

Figure 15-9 Chip Selection of the 8237A in the PC

15.3: 8237 DMA INTERFACING IN THE IBM PC

- Address selection of the registers inside the 8237.
  - Summarized, assuming zero for each x.

Port addresses 0–7 are assigned to the four channels

Addresses 08–0F are assigned to control registers commonly used by all channels.

See page 414 of your textbook.

Table 15-5: PC8237 Internal Register Port Addresses

Binary address		Hex		
A7 A6 A5 A4	A3 A2 A1 A0	Address	Function	Read/ Write
0 0 0 x	0 0 0 0	00	CHAN0 memory address register	R/W
0 0 0 x	0 0 0 1	01	CHAN0 count register	R/W
0 0 0 x	0 0 1 0	02	CHAN1 memory address register	R/W
0 0 0 x	0 0 1 1	03	CHAN1 count register	R/W
0 0 0 x	0 1 0 0	04	CHAN2 memory address register	R/W
0 0 0 x	0 1 0 1	05	CHAN2 count register	R/W
0 0 0 x	0 1 1 0	06	CHAN3 memory address register	R/W
0 0 0 x	0 1 1 1	07	CHAN3 count register	R/W
0 0 0 x	1 0 0 0	08	Status/command register	R/W
0 0 0 x	1 0 0 1	09	Request register	W
0 0 0 x	1 0 1 0	0A	Single mask register bit	W
0 0 0 x	1 0 1 1	0B	Mode register	W
0 0 0 x	1 1 0 0	0C	Clear byte pointer	W
0 0 0 x	1 1 0 1	0D	Master clear/temporary register	R/W
0 0 0 x	1 1 1 0	0E	Clear mask register	W
0 0 0 x	1 1 1 1	0F	Mask register bits	W

15.3: 8237 DMA INTERFACING IN THE IBM PC
connections in the IBM PC

- DMA must be capable of transferring data between I/O & memory without interference from the CPU.
  - It must have all required control, data & address buses.

15.3: 8237 DMA INTERFACING IN THE IBM PC
connections in the IBM PC

- 8237 has its own bidirectional data bus, **D0–D7**, which is connected to x86 system bus **D0–D7**.
  - It has four control buses: **IOR**; **IOW**; **MEMR**; **MEMW**.
  - The address bus, **A0–A7**, is only 8 bits.
    - **ADSTB** (address strobe) activates the latch when 8237 provides the upper 8-bit address through the data bus.

AEN = 0 - x86 controls system bus.
AEN = 1 - 8237 DMA controls system bus.

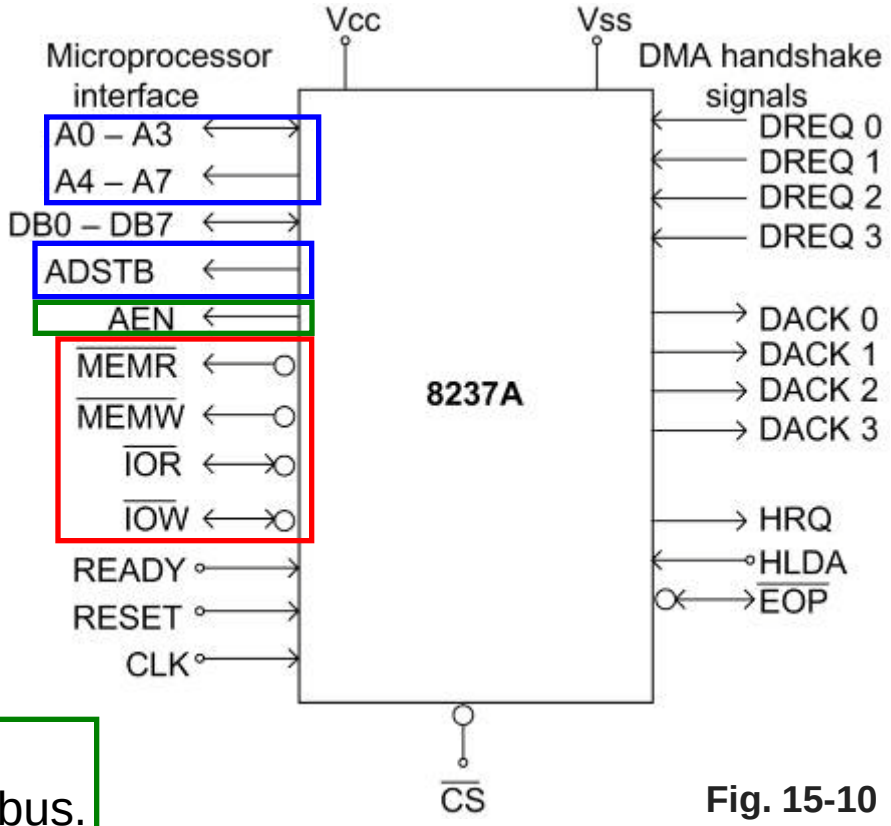
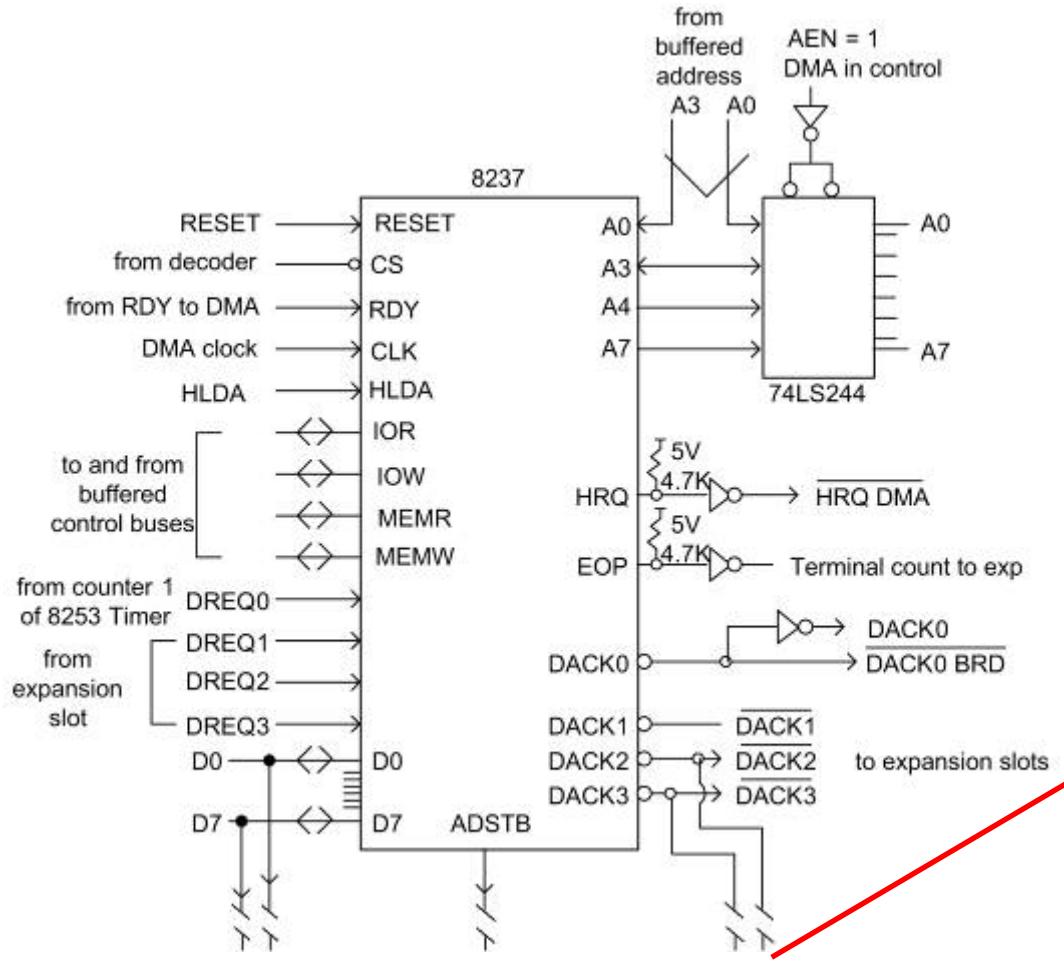
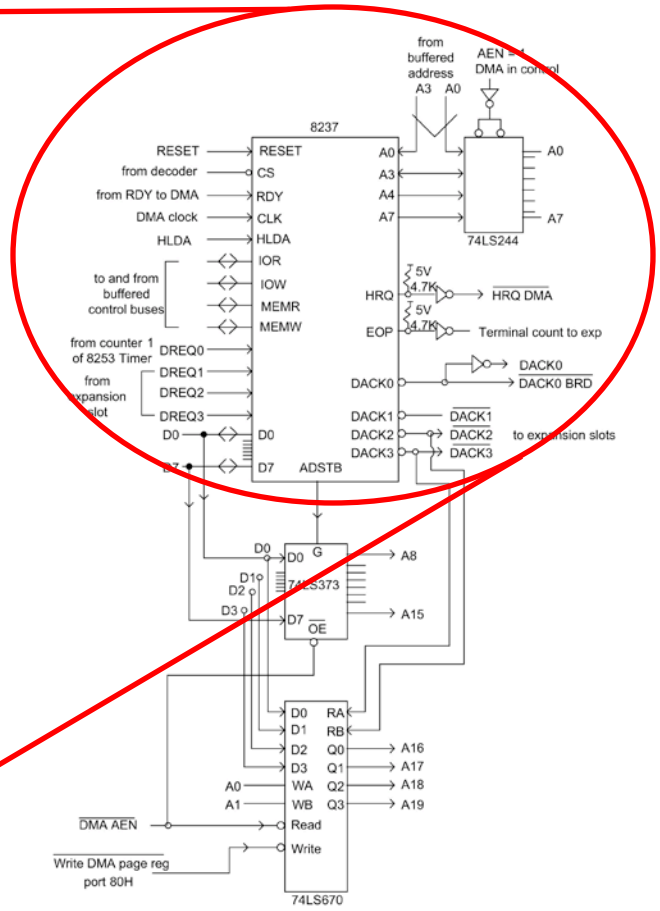


Fig. 15-10
Block Diagram of the 8237A DMA

15.3: 8237 DMA INTERFACING IN THE IBM PC connections in the IBM PC





See the entire circuit on page 415 of your textbook.

Figure15-11a DMA Circuit Connection in the PC

15.3: 8237 DMA INTERFACING IN THE IBM PC
connections in the IBM PC

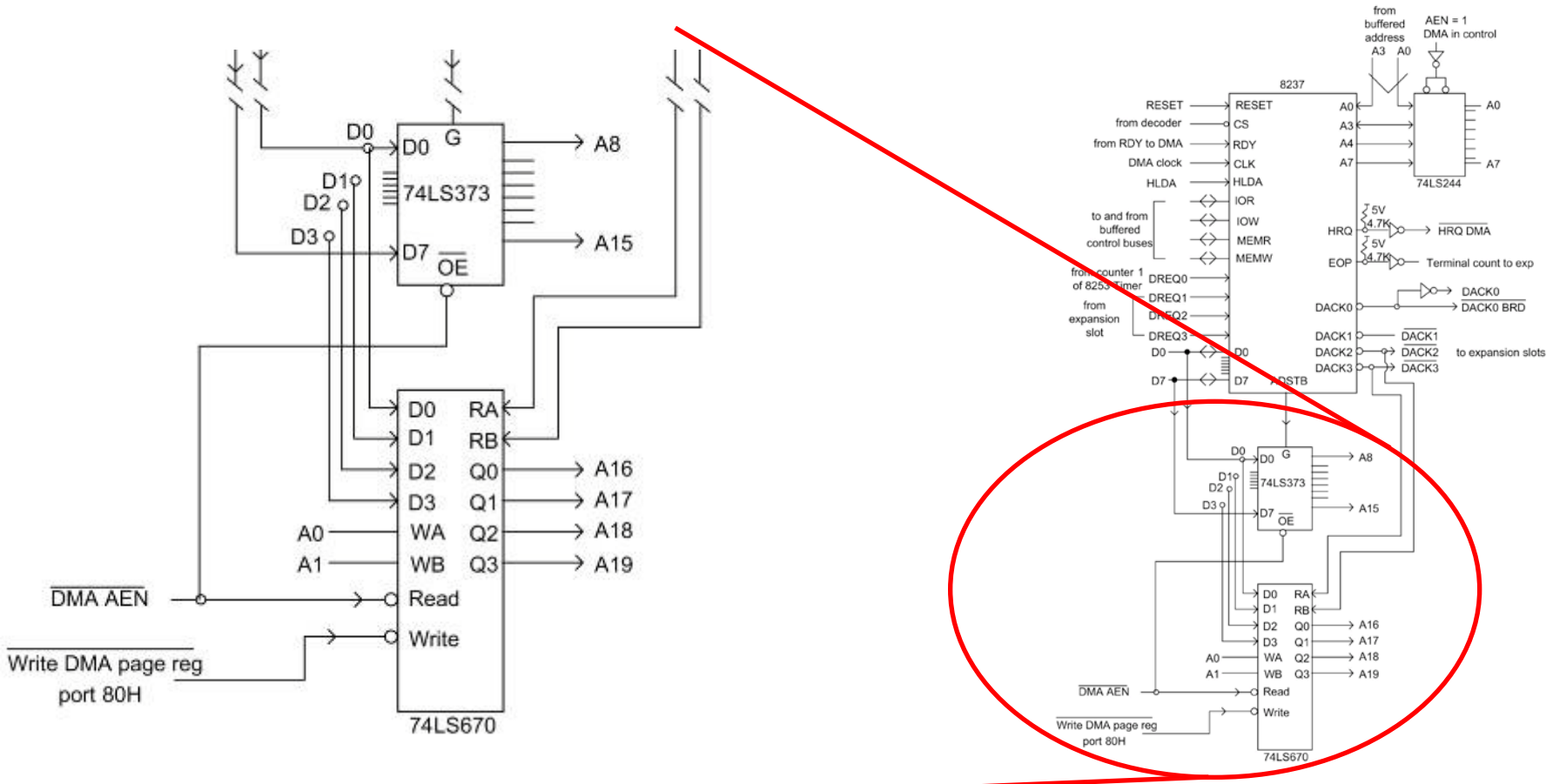


Figure15-11b DMA Circuit Connection in the PC

See the entire circuit on page 415 of your textbook.

15.3: 8237 DMA INTERFACING IN THE IBM PC
connections in the IBM PC

- **RESET** - input coming from the RESET of 8284.
- **CS** is from the 74LS138 decoder.
- **READY** input is from the RDYDMA of the wait state generation circuitry.
  - The purpose is to extend the memory cycle of the DMA.
- **HOLD** and **HLDA** are connected to the pins with the same name on the x86 CPU.
- **EOP** (end of process) is inverted and becomes **TC** (terminal count).
  - Activated whenever the count register of any of the four channels is decremented to zero.

15.3: 8237 DMA INTERFACING IN THE IBM PC
connections in the IBM PC

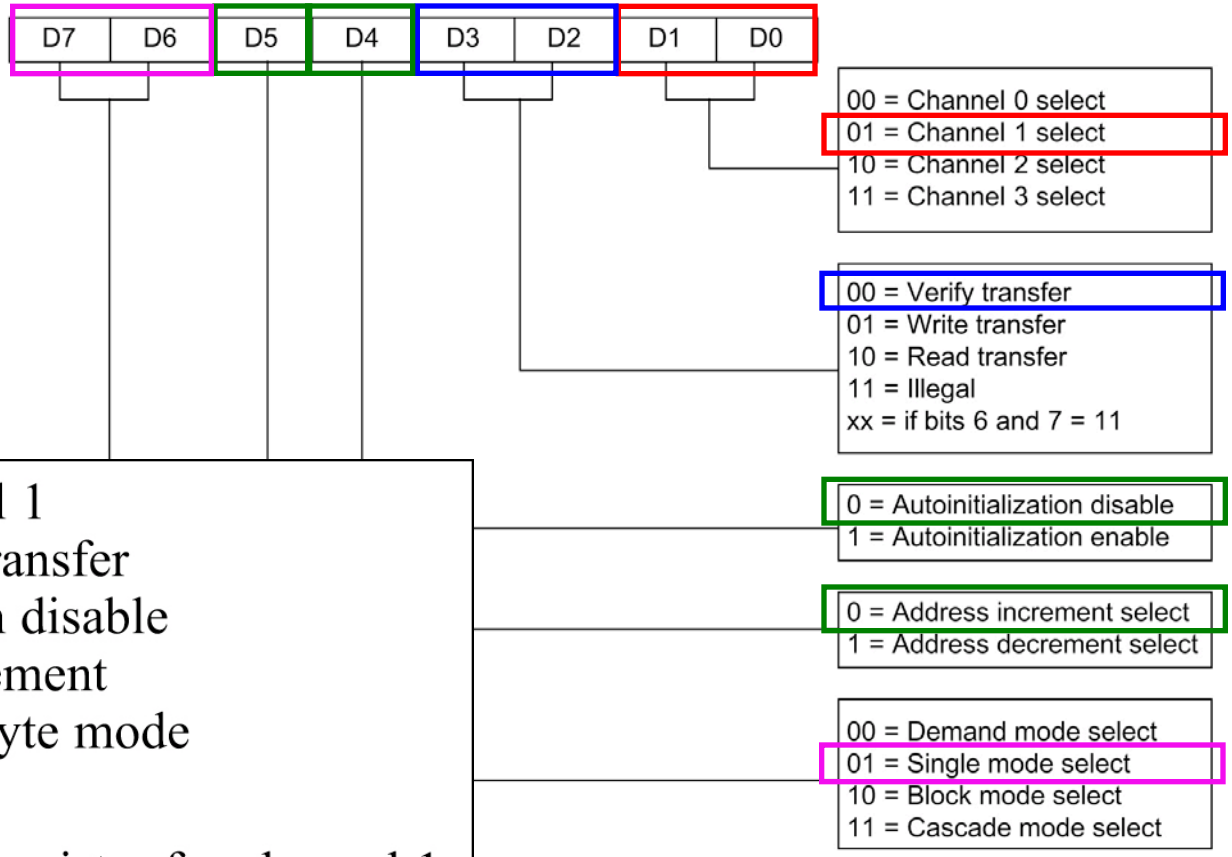
- **DREQ0** & **DACK0** are the signals for channel 0, used for refreshing DRAM.
  - While the DREQ is *active-high*, DACK0 is programmed to be *active-low* by BIOS.
- **DREQ1–DREQ3** & **DACK1–DACK3** are signals for channel 1 to 3, available through the expansion slot.

15.3: 8237 DMA INTERFACING IN THE IBM PC
channel assignment of the 8237 in the IBM PC

- In the original PC, each of the four 8237 channels is assigned in the following fashion:
  - 1. Channel 0 for refreshing DRAM, later abandoned.
  - 2. Channel 1 is unused, but in many implementations it is used for networks.
  - 3. Channel 2 usually is used for the floppy disk controller.
  - 4. Channel 3 normally is used for the hard disk controller.
- Inspecting IBM BIOS shows that 8237 channels 1, 2, and 3 have been initialized by programming the mode register.

15.2: 8237 DMA CHIP PROGRAMMING
mode register

The mode register for channel 1, which must be sent to port address 0BH.



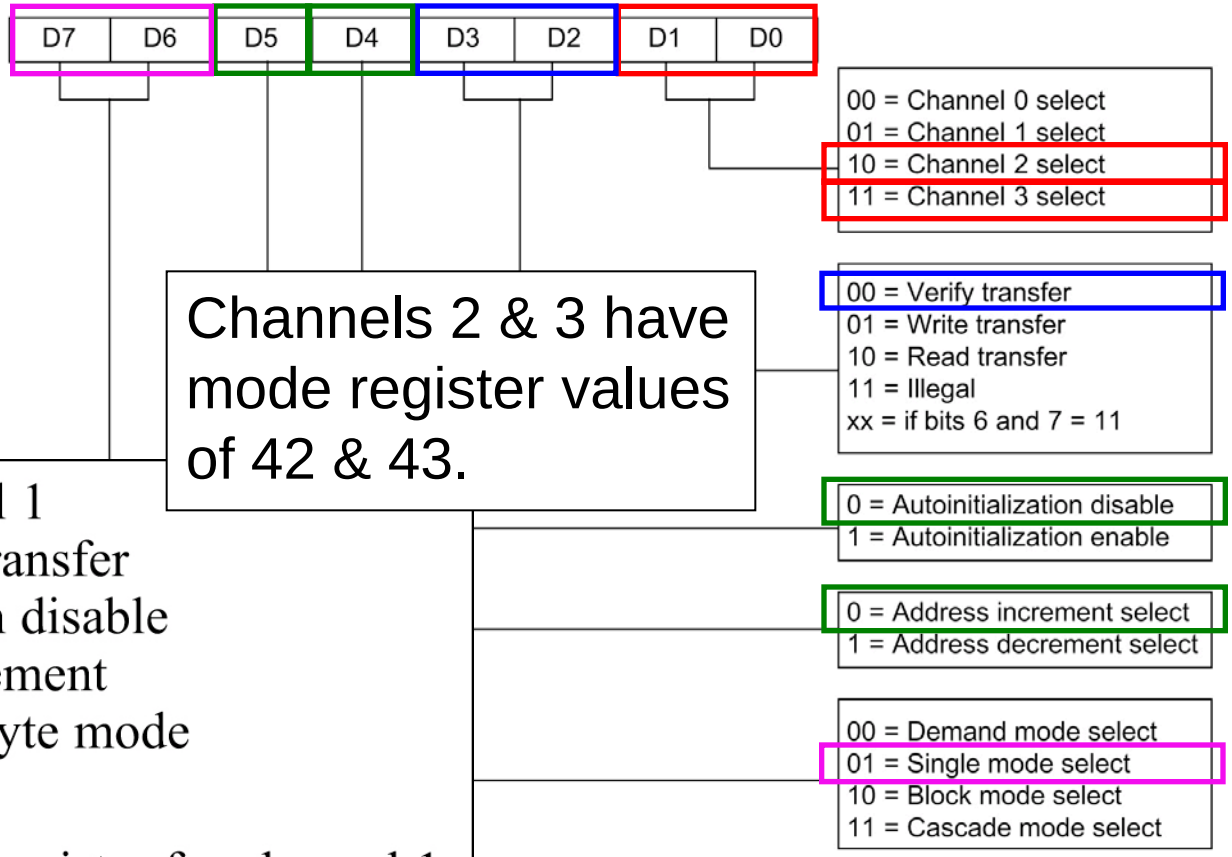
D1,D0 = 01 for channel 1
D3,D2 = 00 for verify transfer
D4 = 0 autoinitialization disable
D5 = 0 for address increment
D7,D6 = 01 for single byte mode

D7 D0
0100 0001 = 41H mode register for channel 1

Fig. 15-5 8237 Mode Register Format

15.2: 8237 DMA CHIP PROGRAMMING
mode register

For channels 2 & 3, the mode register value is the same except that **D0** & **D1** change to 10 & 11.



D1,D0 = 01 for channel 1
D3,D2 = 00 for verify transfer
D4 = 0 autoinitialization disable
D5 = 0 for address increment
D7,D6 = 01 for single byte mode

D7 D0
0100 0001 = 41H mode register for channel 1

Fig. 15-5 8237 Mode Register Format

15.4: DMA IN x86 PCs

- The original IBM PC had only three DMA channels available through the expansion slot.
  - These channels were designed for 8-bit data transfer.

To expand capability, x86 designers added the **second 8237** and made it a 16-bit data transfer DMA.

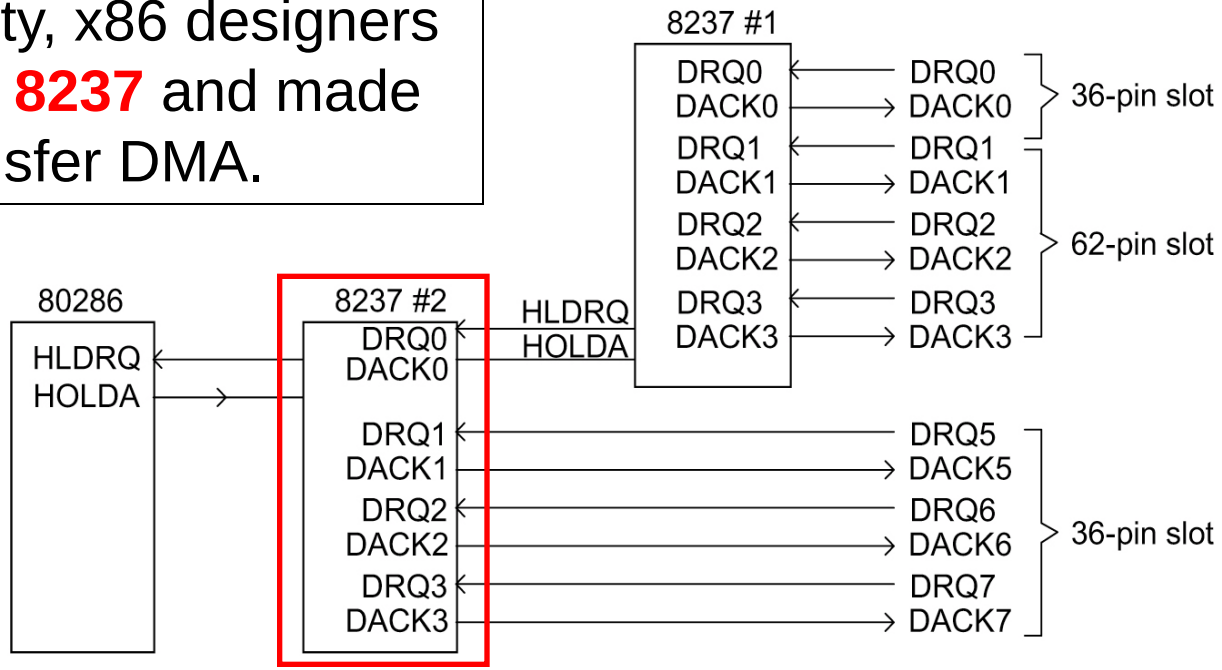


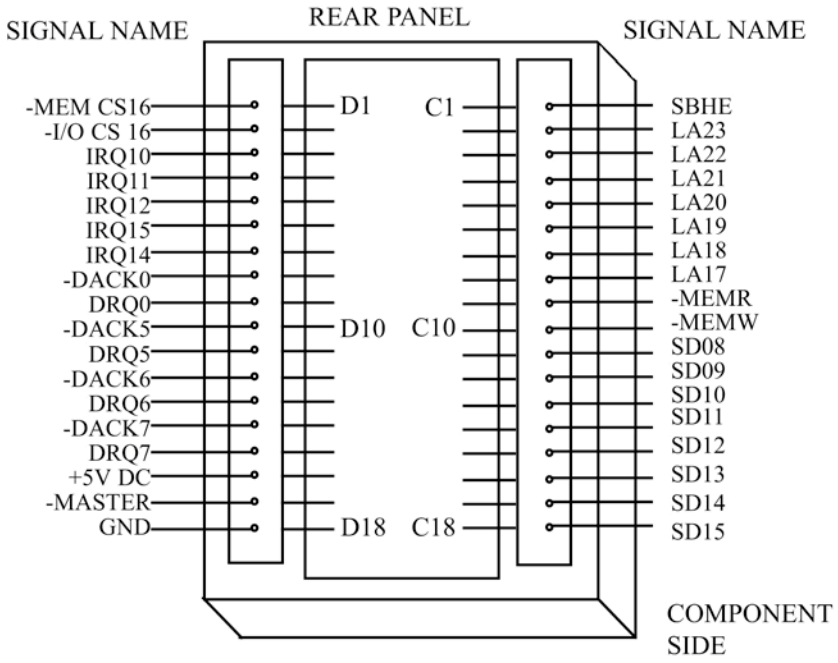
Fig. 15-12
80286 (and higher) PCDMA

15.4: DMA IN x86 PCs
8237 DMA #1

- **8237 DMA #1** - DREQ1, DREQ2 & DREQ3 of DMA #1 are available through the expansion slot.
  - 8-bit transfer between 8-bit I/O & 16 MB memory range.
    - Ports assigned to DMA#1 are the same as in the PC.
  - Channel 0 is available through the ISA expansion slot, signaled by DREQ0 and DACK0, accessed though the 36 edge of the ISA.
    - Channels 0, 1, 2, and 3 can be used only for data transfer between 8-bit I/O and system memory.
    - 2. As the count register is a 16-bit register, each of channels 0, 1, 2, and 3 can transfer up to a 64K-byte block of data.
    - 3. Each channel, 0, 1, 2, or 3, can transfer data in 64K-byte blocks throughout the 16M system memory address space.

15.4: DMA IN x86 PCs

x86 PCs abandoned refreshing DRAM using DMA channel 0, replacing it with DRAM refresher circuitry.



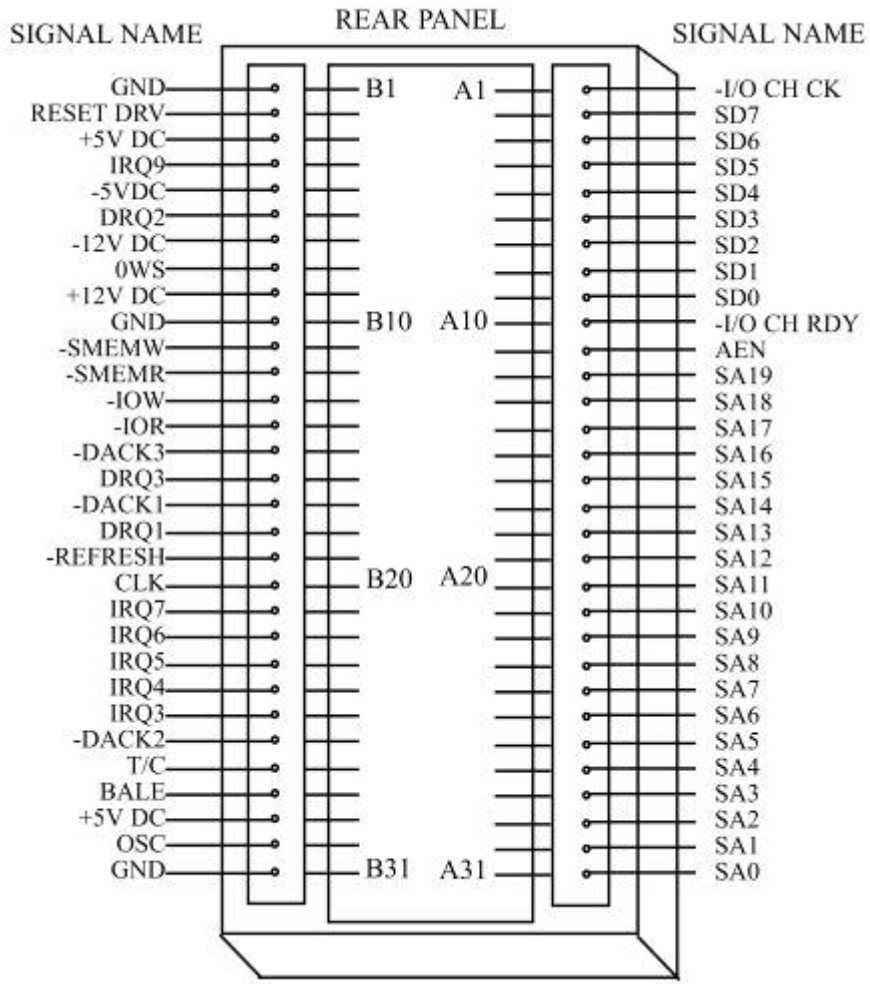


Fig. 15-13 ISA (IBM PC) Bus Slot Signals

15.4: DMA IN x86 PCs
8237 DMA #2

- The **second 8237** is connected as master (level 1) and its channel 0 is used for cascading of **DMA1**

The other three channels of this DMA are available through the expansion slot (36 edge) under DREQ5 & DACK5, DREQ6 & DACK6, and DREQ7 & DACK7.

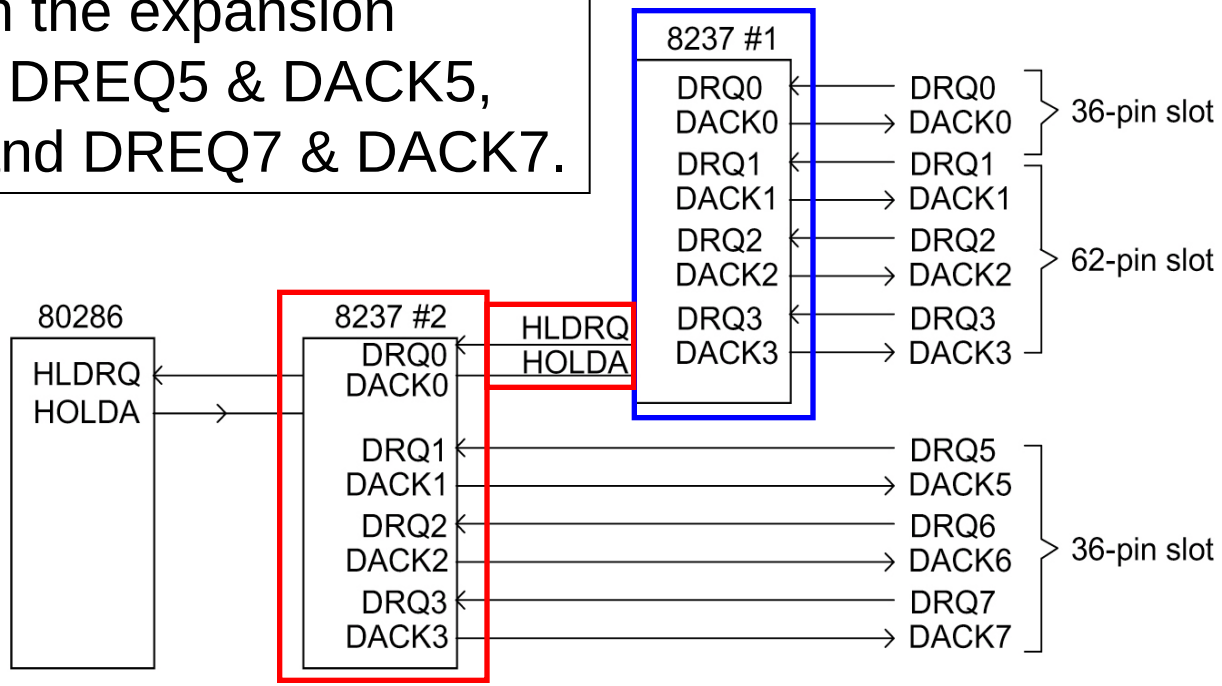


Fig. 15-12
80286 (and higher) PCDMA

15.4: DMA IN x86 PCs
8237 DMA #2

- Channels 5, 6 & 7 of DMA #2 are used for 16-bit data transfer between the 16 MB memory address space & I/O peripherals which support 16-bit data.
- The number of 16-bit (2-byte) words transferred is in the 16-bit count register, of channels 5, 6 & 7.
  - Each channel can transfer up to 65,536 words.
    - 128K bytes between I/O and memory.
  - Channels 5, 6 & 7 transfer data in blocks of a maximum size of 128K bytes.
- The memory address for a DMA memory transfer must be on an even-byte address boundary.

15.4: DMA IN x86 PCs
8237 DMA #2

- Since channels 5, 6 & 7 cannot transfer data on an odd-byte boundary, A0 & **BHE** are both forced to 0.

DMA #2 can be accessed (programmed) by another master from the expansion slot using the **MASTER** input signal on the 36-pin part of the ISA bus.

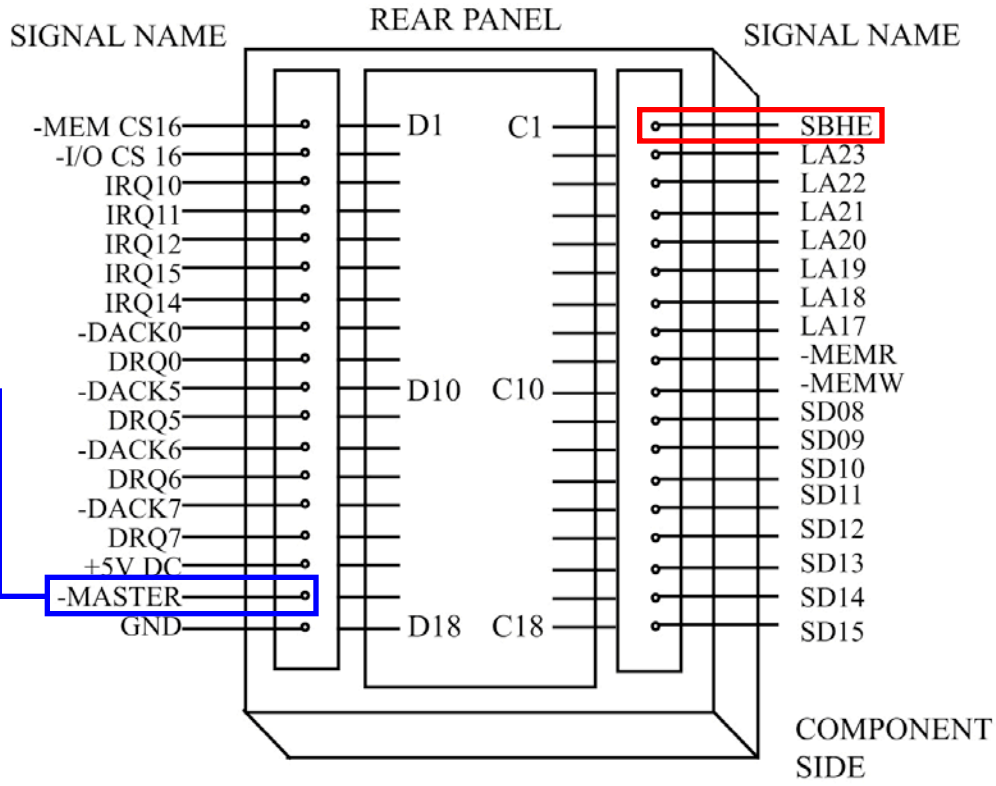


Fig. 15-13 ISA (IBM PC) Bus Slot Signals

15.4: DMA IN x86 PCs
DMA channel priority

- Of the seven x86 ISA expansion slot DMA channels, channel 0 has highest & channel 7 lowest priority.
  - BIOS programs both DMAs to have channel 0 as the highest priority.

The master DMA (8237 #2) has channel 0 as the highest priority.

Since the slave 8237 #1 is connected to it, channels 0 through 3 have higher priority than channels 5, 6, and 7.

x86 ISA DMA Channel Priority	
channel 0	Highest priority
channel 1	
channel 2	
channel 3	
channel 5	
channel 6	
channel 7	Lowest priority

The x86 PC

assembly language, design, and interfacing

fifth edition

Prentice Hall

Dec Hex Bin
15 F 00001111

ENDS ; FIFTEEN

The x86 PC

assembly language, design, and interfacing

fifth edition

MUHAMMAD ALI MAZIDI
JANICE GILLISPIE MAZIDI
DANNY CAUSEY