# Microprocessors and Microcontrollers (EE-231)

## Lecture-17

# Main Objectives

- Direct Memory Access
  - 8237 DMA controller
- Bus interface

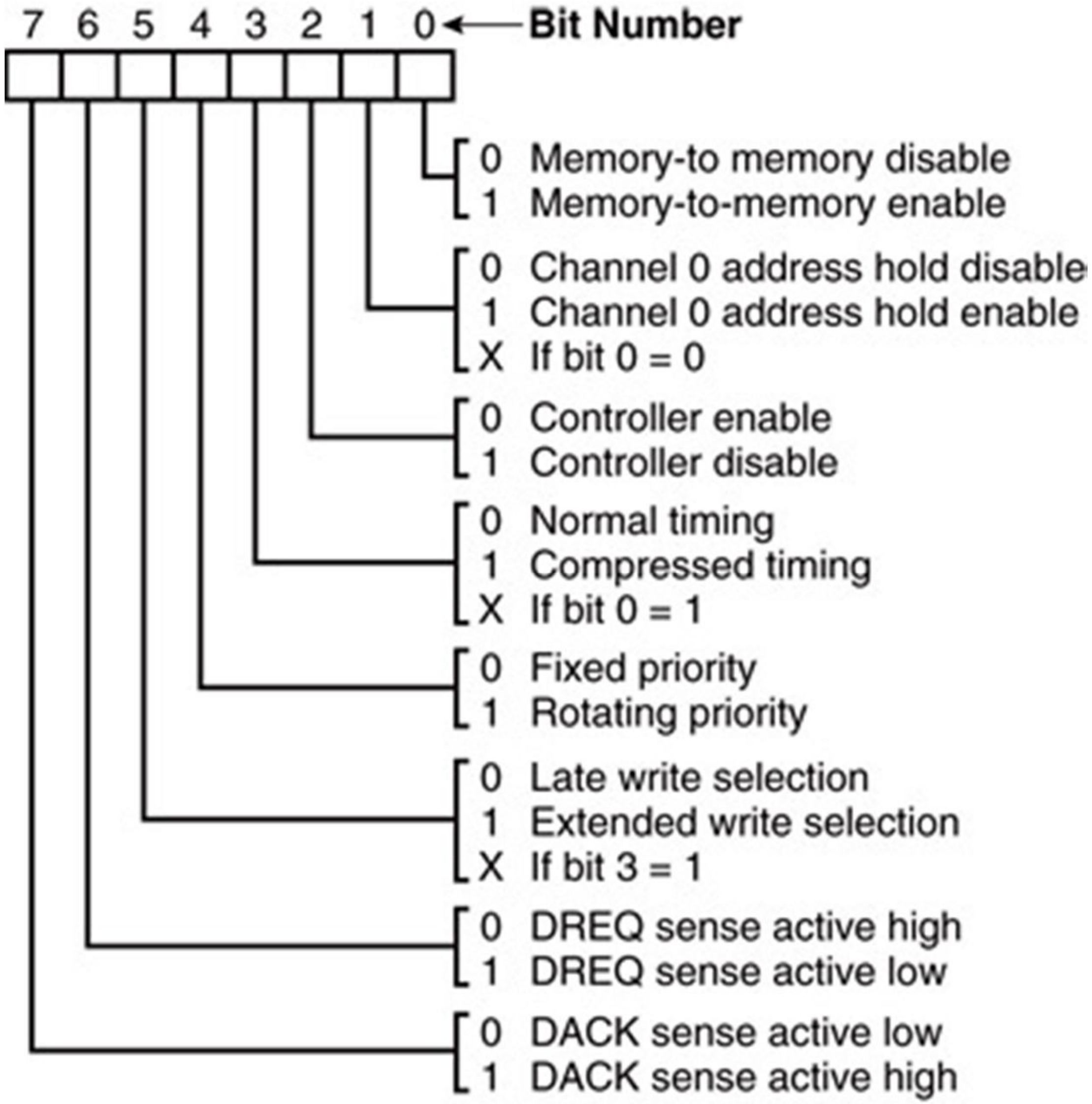
# 8237 Internal Registers

- 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
- 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

# 8237 Internal Registers

- **BA & 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
- **CR**
- The command register programs the operation of the 8237 DMA controller.
- Memory to memory transfer:
- 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 **while** DMA channel 1 holds the destination address

# 8237A-5 command register.

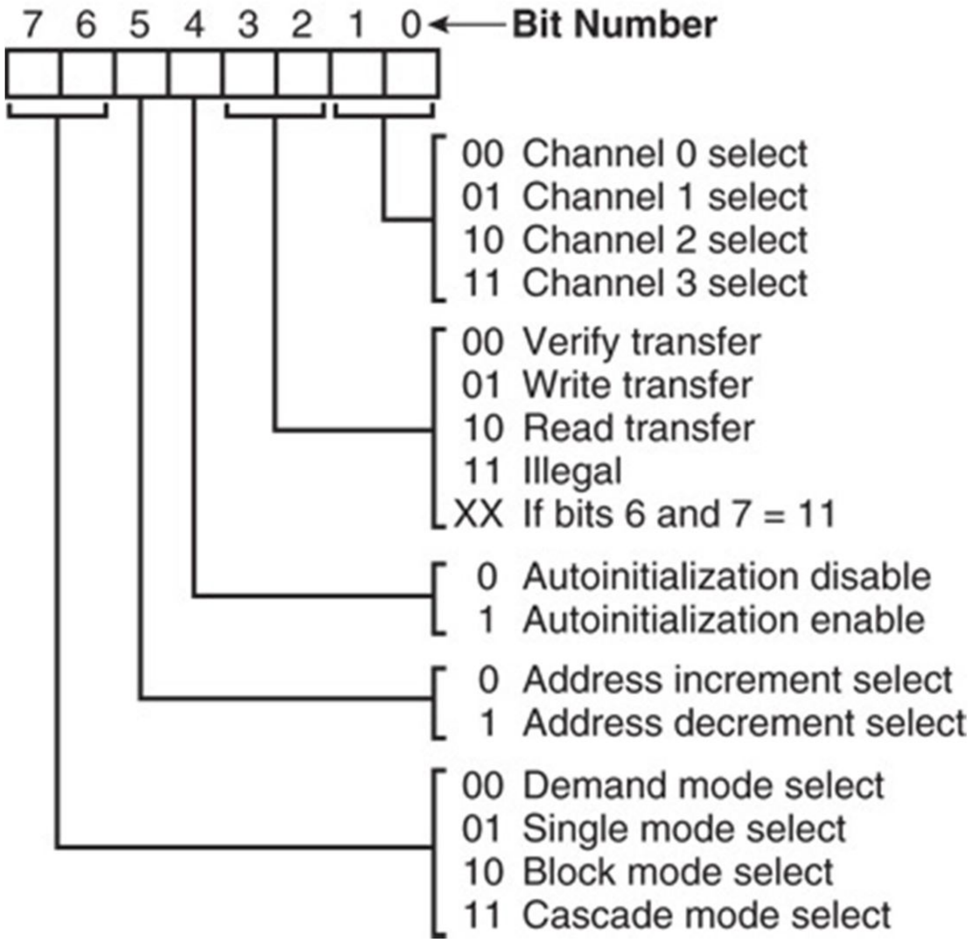


7 6 5 4 3 2 1 0 ← Bit Number

- Bit 0:
  - 0 Memory-to memory disable
  - 1 Memory-to-memory enable
- Bit 1:
  - 0 Channel 0 address hold disable
  - 1 Channel 0 address hold enable
  - X If bit 0 = 0
- Bit 2:
  - 0 Controller enable
  - 1 Controller disable
- Bit 3:
  - 0 Normal timing
  - 1 Compressed timing
  - X If bit 0 = 1
- Bit 4:
  - 0 Fixed priority
  - 1 Rotating priority
- Bit 5:
  - 0 Late write selection
  - 1 Extended write selection
  - X If bit 3 = 1
- Bit 6:
  - 0 DREQ sense active high
  - 1 DREQ sense active low
- Bit 7:
  - 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

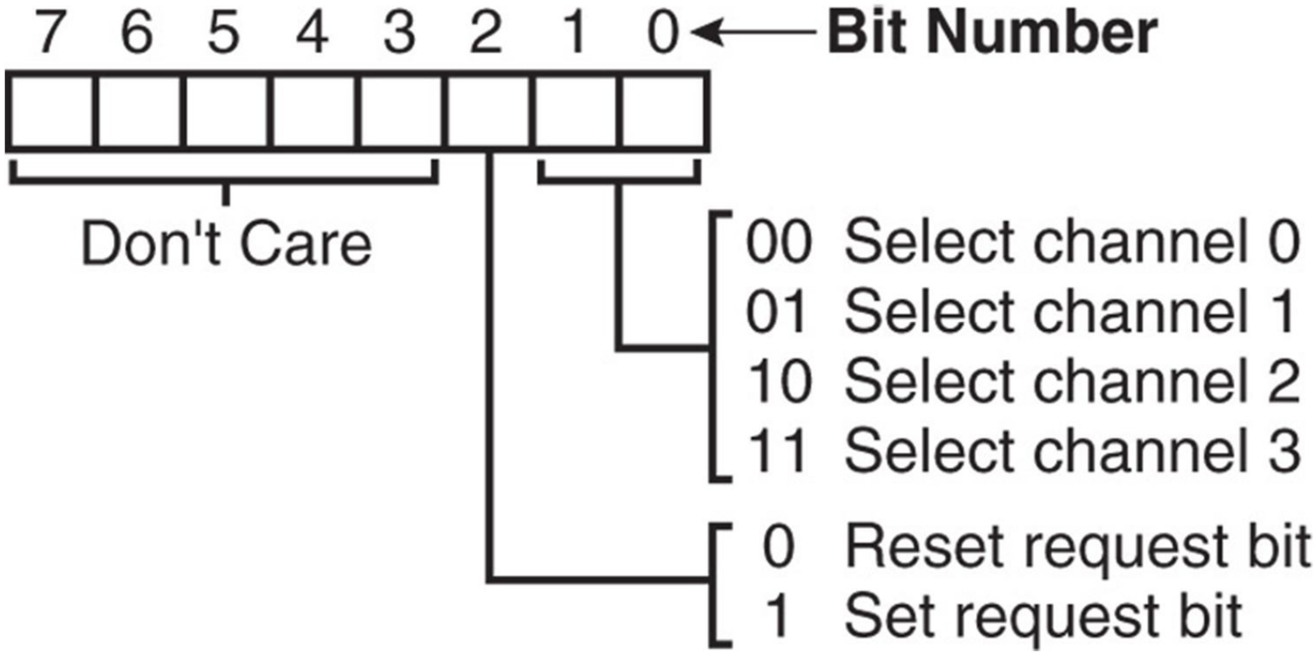


# Three Types of DMA Mode

- Demand mode
  - transfers data until DREQ becomes inactive, or when EOP pin is asserted.
- Single mode
  - releases HOLD after each byte of data is transferred
  - If DREQ is active, DMAC requests a DMA transfer to microprocessor
- Block mode
  - automatically transfers the number of bytes indicated by the count register for the channel

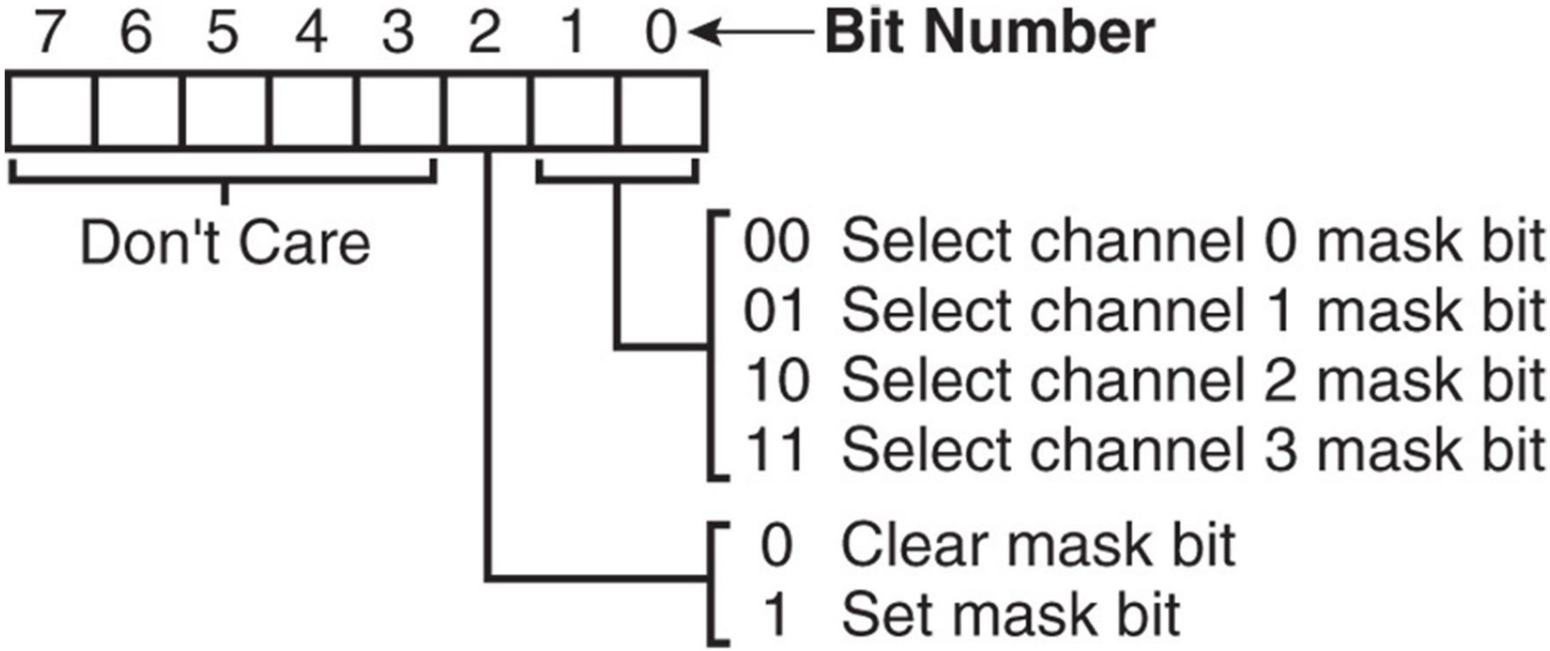
# 8237 Internal Registers

- **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



# 8237 Internal Registers

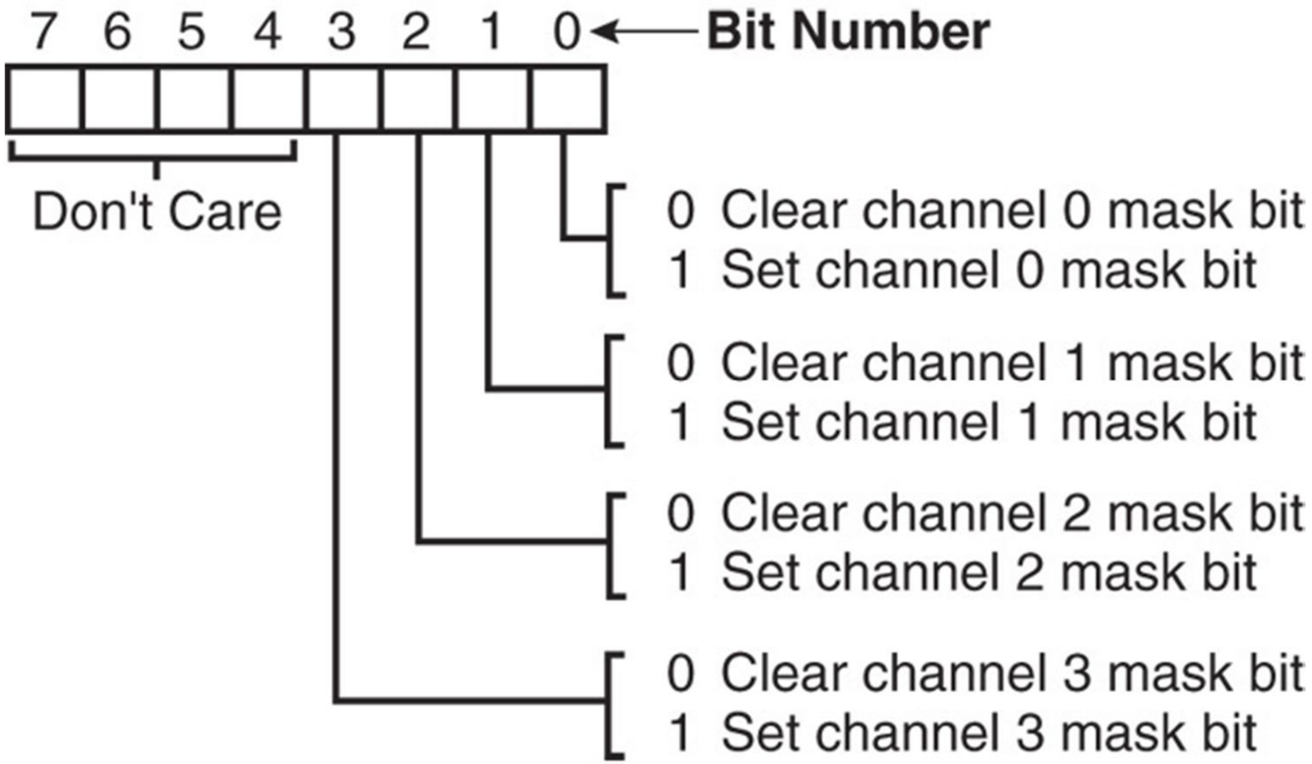
- **MRSR**
- The mask register set/reset sets or clears the channel mask.
- if the mask is set, the channel is disabled.
- The RESET signal sets all channel masks to disable them



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

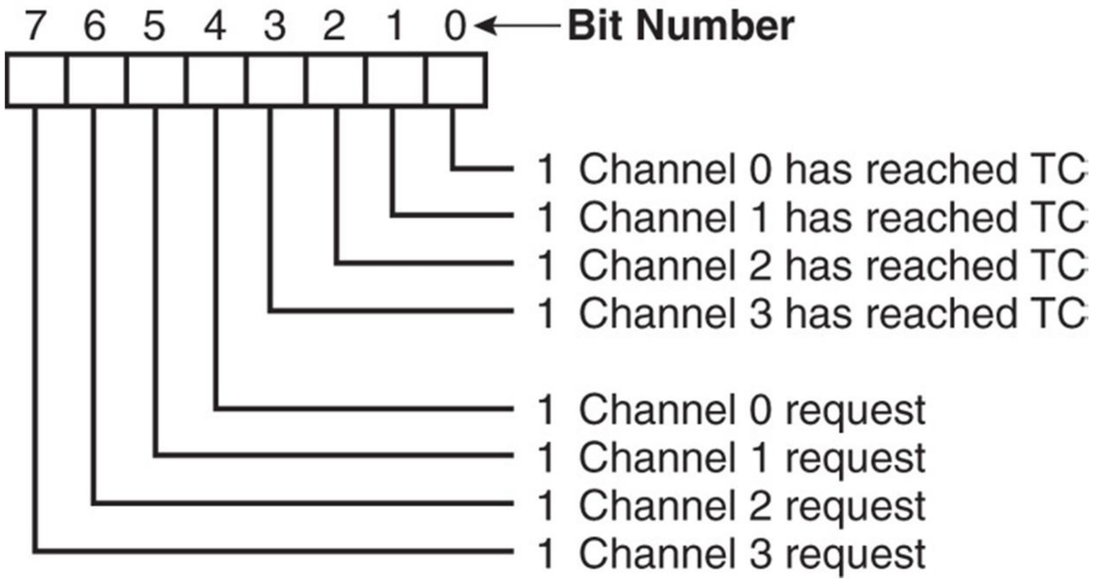
# 8237 Internal Registers

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



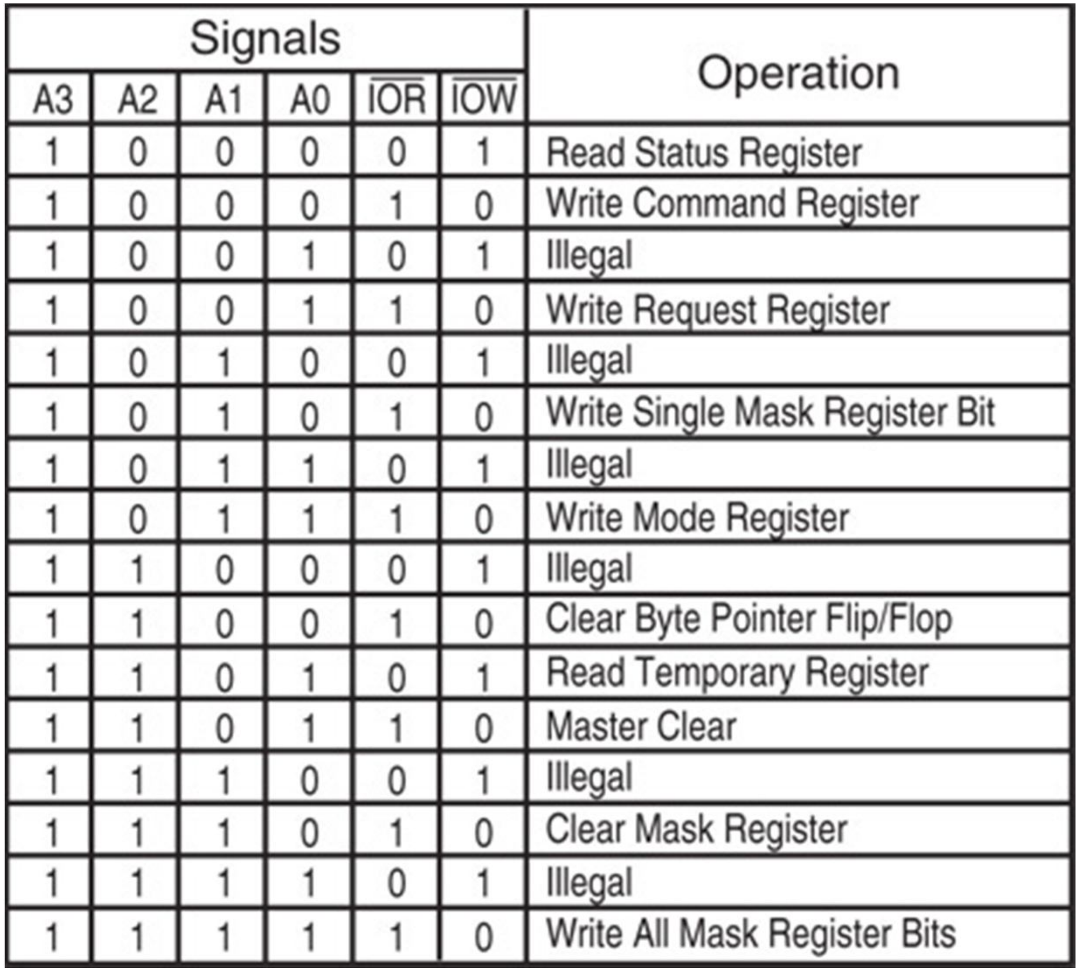
# 8237 Internal Registers

- **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


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

# Software Commands

- Following are the software commands to govern the operation of the DMA



| 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 |

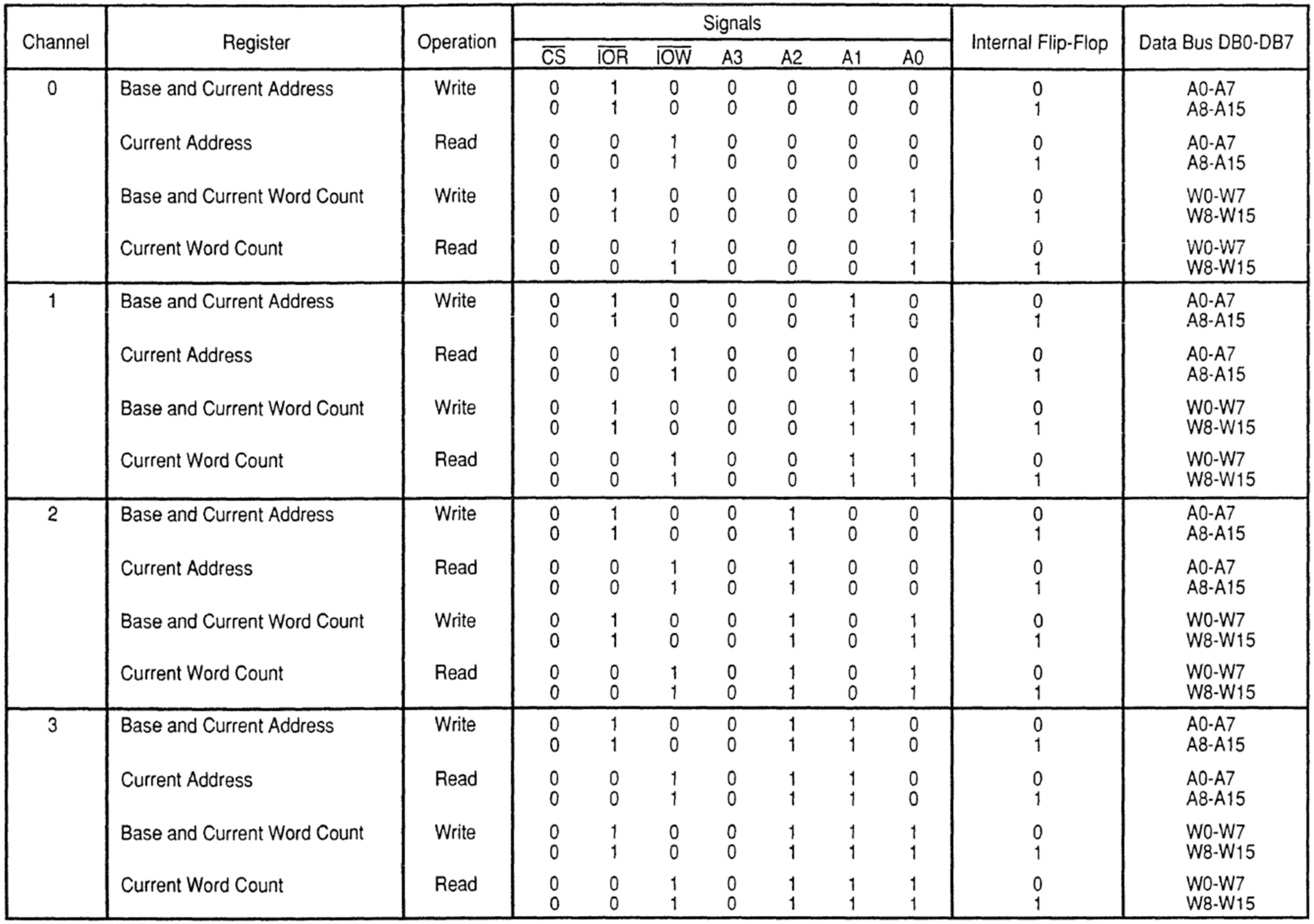
# Software Commands

- **(1)Master Clear**
- Acts exactly the same as the RESET signal to the 8237.
  - as with the RESET signal, this command disables all channels
- **(2)Clear mask register**
- Enables all four DMA channels.
- **(3)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 (on next slide) 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.

# 8237A-5 DMA channel I/O port addresses.



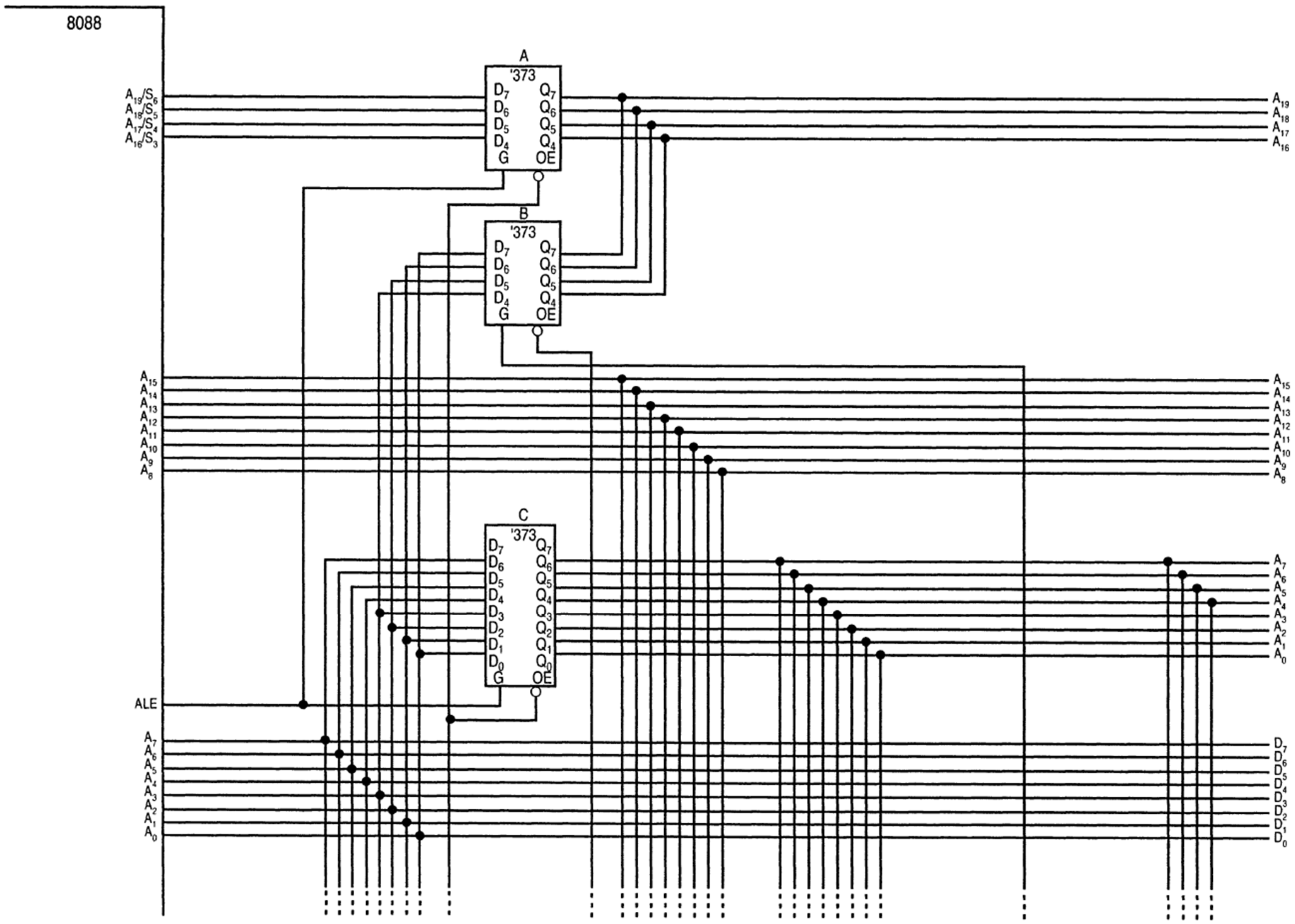
| Channel | Register | Operation | Signals | | | | | | | Internal Flip-Flop | Data Bus DB0-DB7 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | $\overline{CS}$ | $\overline{IOR}$ | $\overline{IOW}$ | A3 | A2 | A1 | A0 | | |
| 0 | Base and Current Address | Write | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 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 |
`

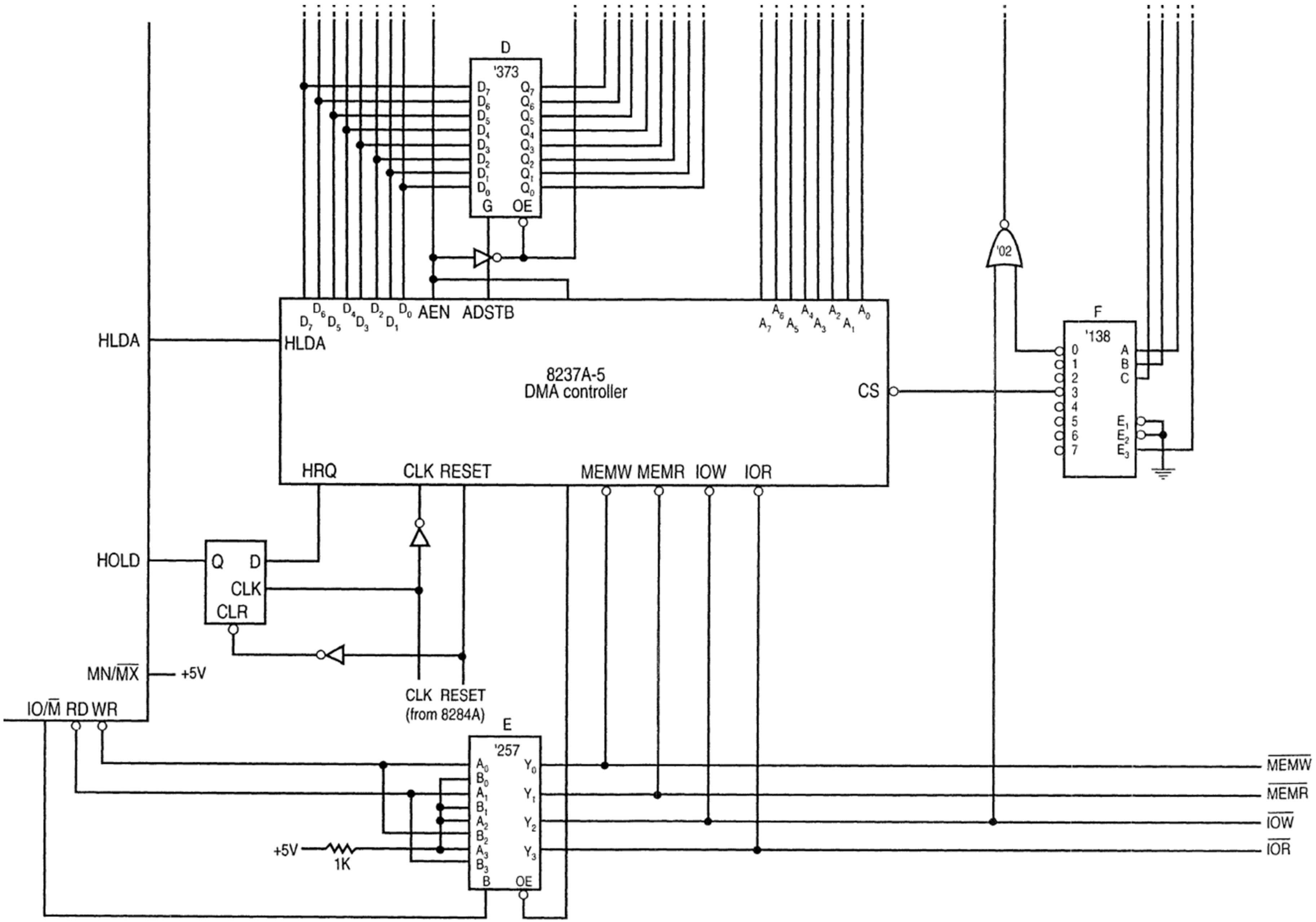
# Basic Steps to Program

- 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 $A_7$–$A_0$

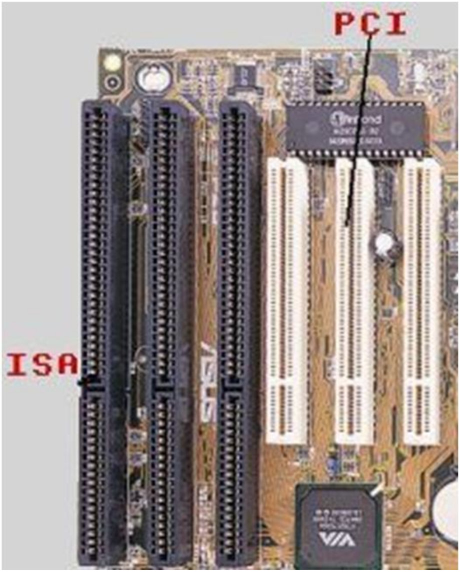

8088
A
'373
B
'373
C
'373
D7 D6 D5 D4 G OE
Q7 Q6 Q5 Q4
D3 D2 D1 D0
Q3 Q2 Q1 Q0
A19/S6
A18/S5
A17/S4
A16/S3
A15 A14 A13 A12 A11 A10 A9 A8
ALE
A7 A6 A5 A4 A3 A2 A1 A0
A19 A18 A17 A16
D7 D6 D5 D4 D3 D2 D1 D0


D
'373
$D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$
$Q_7$ $Q_6$ $Q_5$ $Q_4$ $Q_3$ $Q_2$ $Q_1$ $Q_0$
G OE
'02
$D_7$ $D_6$ $D_5$ $D_4$ $D_3$ $D_2$ $D_1$ $D_0$ AEN ADSTB
$A_7$ $A_6$ $A_5$ $A_4$ $A_3$ $A_2$ $A_1$ $A_0$
HLDA
HLDA
8237A-5
DMA controller
CS
F
'138
0 1 2 3 4 5 6 7
A B C
$E_1$ $E_2$ $E_3$
HRQ CLK RESET
MEMW MEMR IOW IOR
HOLD
Q D
CLK
CLR
MN/$\overline{MX}$ +5V
CLK RESET
(from 8284A)
IO/$\overline{M}$ RD WR
E
'257
$A_0$ $B_0$ $A_1$ $B_1$ $A_2$ $B_2$ $A_3$ $B_3$
B OE
$Y_0$ $Y_1$ $Y_2$ $Y_3$
+5V
1K
$\overline{MEMW}$
$\overline{MEMR}$
$\overline{IOW}$
$\overline{IOR}$

- 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 conntrol signals.

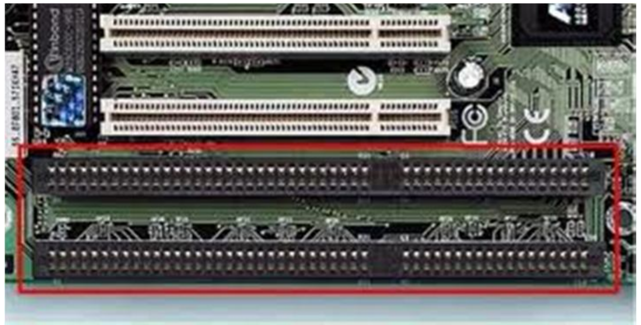
# The ISA BUS

- The **I**ndustry **S**tandard **A**rchitecture, bus has been around since start of the IBM-PC
  - since 1982
- Any card from the very first personal computer will plug in & function in any P4-based system.
  - provided they have an ISA slot
- ISA bus mostly gone from the home PC, but still found in many industrial applications.
  - due to low cost & number of existing cards



# Evolution of the ISA Bus

- Over years, the ISA bus evolved from original 8-bit, to the 16-bit standard found today.
- With the P4, ISA bus started to disappear.
  - a 32-bit version called the EISA bus (**E**xtended ISA) has also largely disappeared
- What remains today is an ISA slot that can accept 8-bit ISA  or 16-bit ISA cards.
- 32-bit printed circuit cards are now PCI bus
  - in some older 80486 systems, VESA cards



# The 8-Bit ISA Bus Output Interface

- The ISA bus connector contains
  - the demultiplexed address bus ($A_{19}$–$A_0$) for the 1M-byte 8088 system
  - the 8-bit data bus ($D_7$–$D_0$)
  - control signals MEMR, MEMW, IOR, and IOW for controlling I/O and any memory placed on the printed circuit card
  - Memory is seldom added to ISA today because ISA cards operate at only 8 MHz.
  - EPROM or flash memory for setup may be on some ISA cards, but never RAM
  - Other signals, useful for I/O interface, are the interrupt request lines IRQ2–IRQ7.

# The 8-Bit ISA Bus Output Interface

- The ISA bus connector contains
  - DMA channel 0–3 control signals are also present on the connector.
  - DMA request inputs are labeled DRQ1–DRQ3 and the DMA acknowledge outputs are labeled DACK0 - DACK3.

# The 8-bit ISA bus.

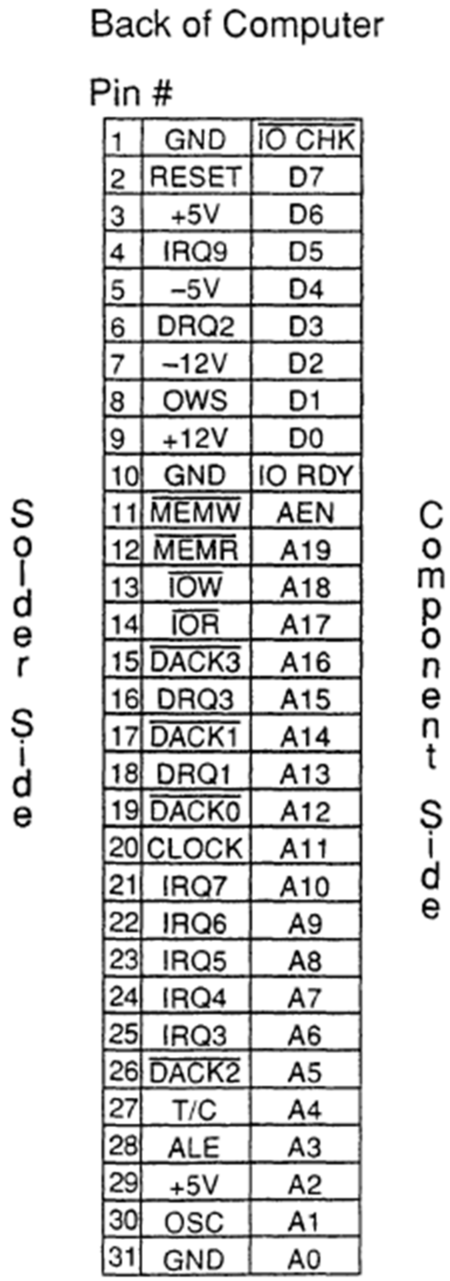
– $IRQ_2$ is redirected to $IRQ_9$ on modern systems, and is so labeled here

– note the $DRQ_0$ input pin is missing,

– early PCs used $DRQ_0$ & the $DACK_0$ output as a refresh signal to refresh DRAM on the ISA card

– today, this output pin contains a 15.2 $\mu$s clock signal used for refreshing DRAM

– remaining pins are for power and RESET



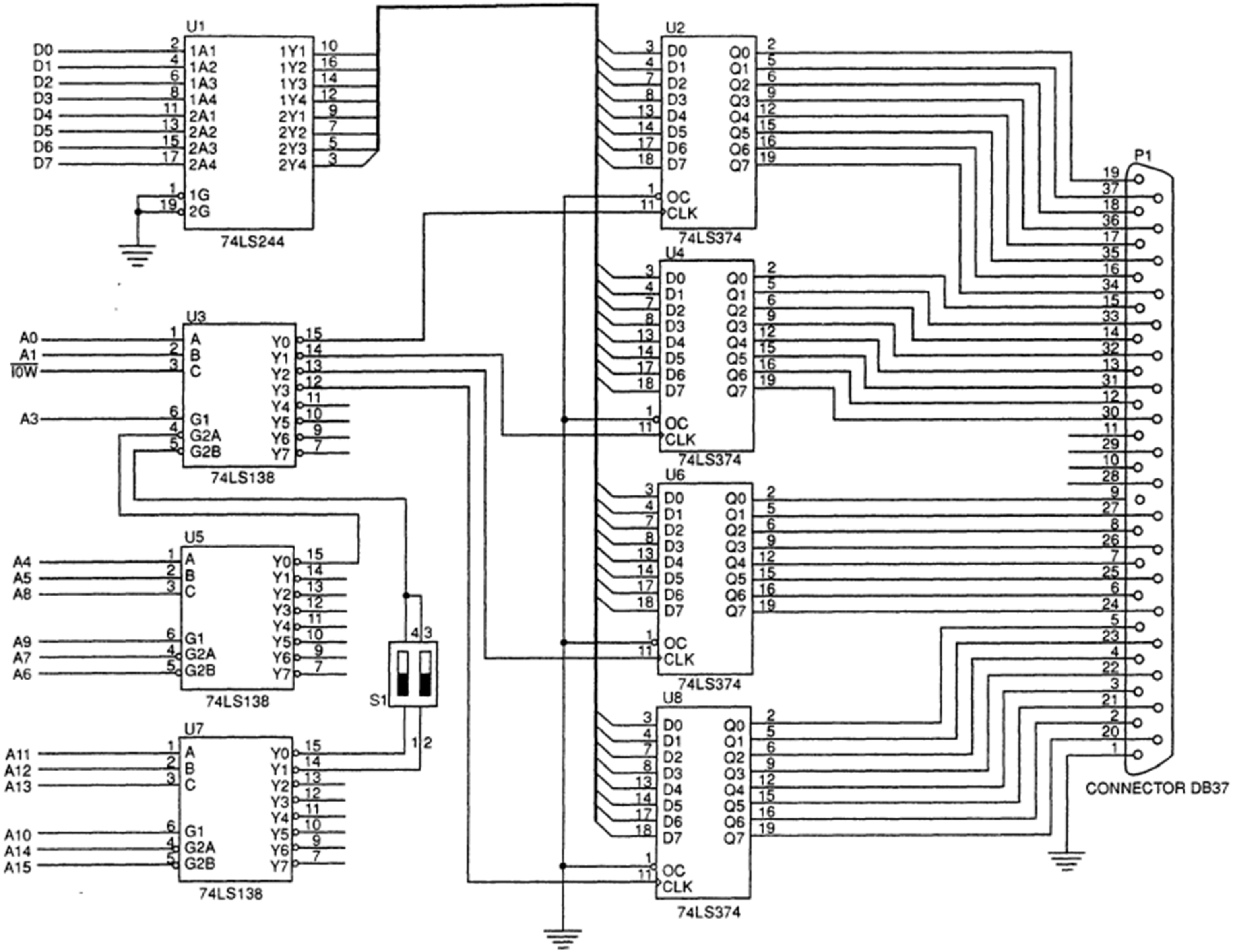
Back of Computer

Solder Side

Component Side

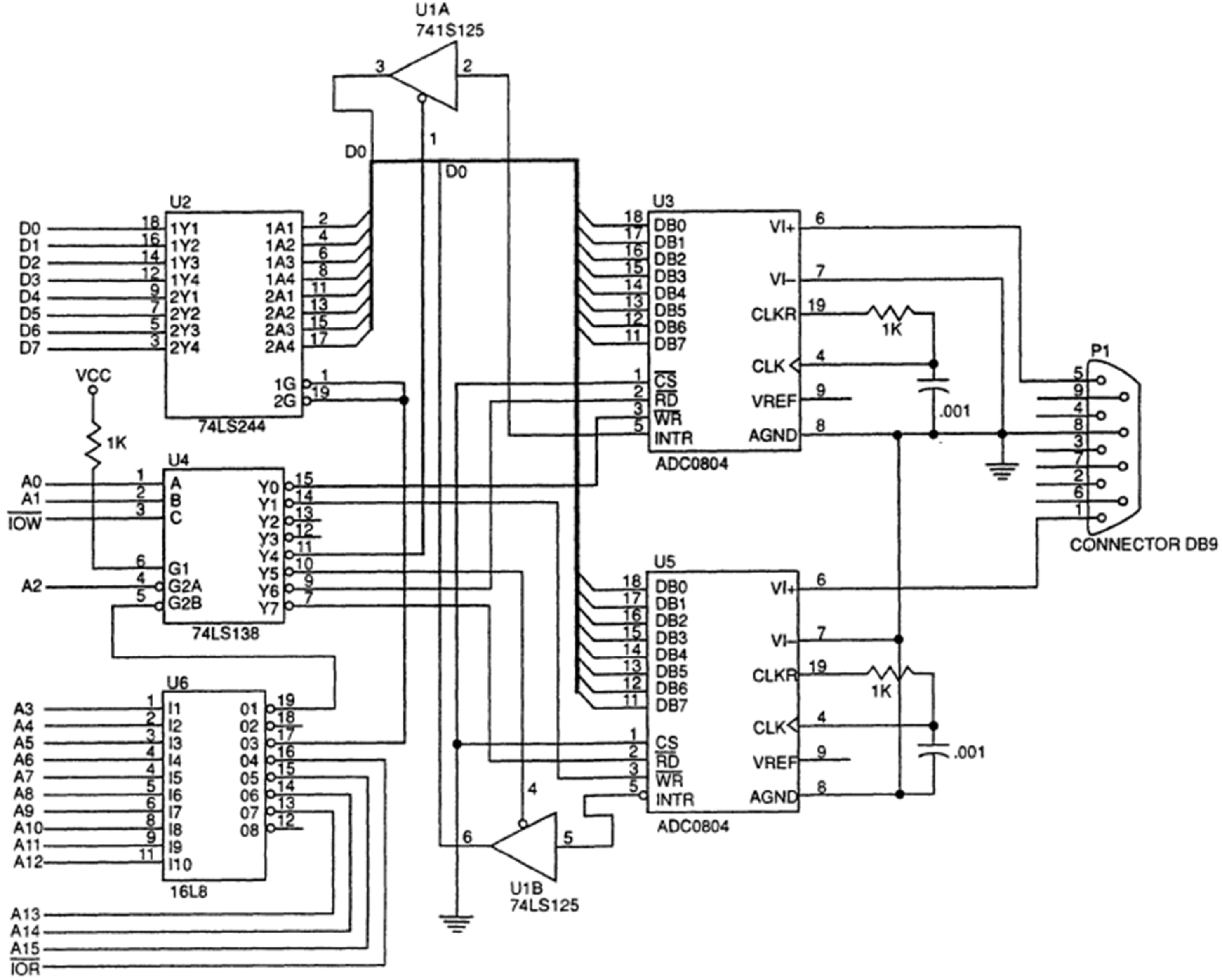
| Pin # | | |
|---|---|---|
| 1 | GND | $\overline{\text{IO CHK}}$ |
| 2 | RESET | D7 |
| 3 | +5V | D6 |
| 4 | IRQ9 | D5 |
| 5 | –5V | D4 |
| 6 | DRQ2 | D3 |
| 7 | –12V | D2 |
| 8 | OWS | D1 |
| 9 | +12V | D0 |
| 10 | GND | IO RDY |
| 11 | $\overline{\text{MEMW}}$ | AEN |
| 12 | $\overline{\text{MEMR}}$ | A19 |
| 13 | $\overline{\text{IOW}}$ | A18 |
| 14 | $\overline{\text{IOR}}$ | A17 |
| 15 | $\overline{\text{DACK3}}$ | A16 |
| 16 | DRQ3 | A15 |
| 17 | $\overline{\text{DACK1}}$ | A14 |
| 18 | DRQ1 | A13 |
| 19 | $\overline{\text{DACK0}}$ | A12 |
| 20 | CLOCK | A11 |
| 21 | IRQ7 | A10 |
| 22 | IRQ6 | A9 |
| 23 | IRQ5 | A8 |
| 24 | IRQ4 | A7 |
| 25 | IRQ3 | A6 |
| 26 | $\overline{\text{DACK2}}$ | A5 |
| 27 | T/C | A4 |
| 28 | ALE | A3 |
| 29 | +5V | A2 |
| 30 | OSC | A1 |
| 31 | GND | A0 |

# The 8-Bit ISA Bus Output Interface  (Generating a 32-bit output)



# The 8-Bit ISA Bus Input Interface

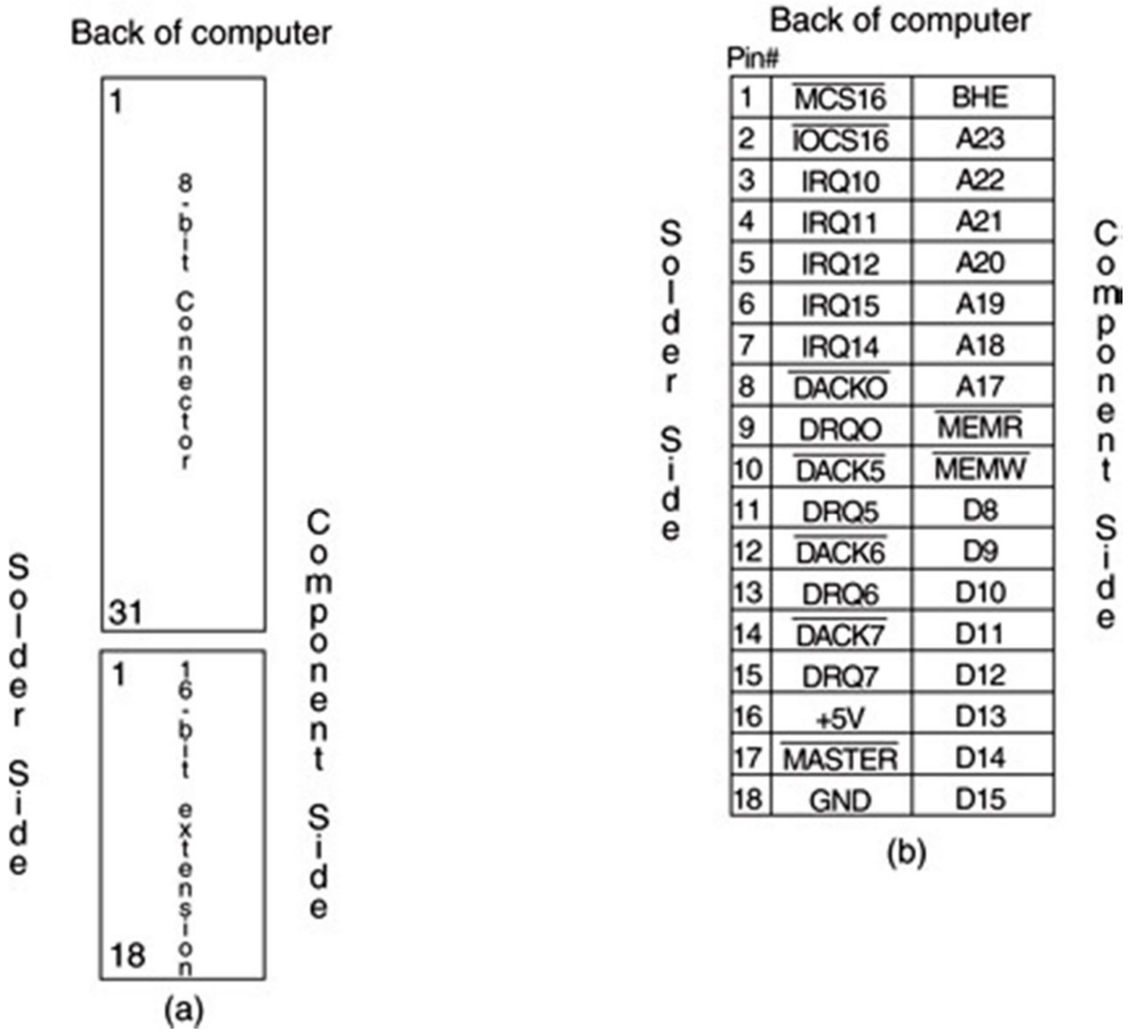
- Figure shows an input interface to the ISA bus, using a pair of ADC804 analog-to-digital converters.
  - – made through a nine-pin $DB_9$ connector
- Decoding I/O port addresses is more complex, as each converter needs:
  - – a write pulse to start a conversion
  - – a read pulse to read the digital data converted
  - – a pulse to enable the selection of the INTR output


U1A
741S125
U1B
74LS125
D0
U2
74LS244
1Y1 1Y2 1Y3 1Y4 2Y1 2Y2 2Y3 2Y4
1A1 1A2 1A3 1A4 2A1 2A2 2A3 2A4
1G 2G
D0 D1 D2 D3 D4 D5 D6 D7
VCC
1K
U4
74LS138
A B C G1 G2A G2B
Y0 Y1 Y2 Y3 Y4 Y5 Y6 Y7
A0 A1 IOW A2
U6
16L8
I1 I2 I3 I4 I5 I6 I7 I8 I9 I10
01 02 03 04 05 06 07 08
A3 A4 A5 A6 A7 A8 A9 A10 A11 A12
A13 A14 A15 IOR
U3
U5
ADC0804
DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7
CS RD WR INTR
VI+ VI– CLKR CLK VREF AGND
1K
.001
P1
CONNECTOR DB9

# The 16-Bit ISA Bus

- The difference between 8- & 16-bit ISA is an extra connector behind the 8-bit connector.
- A 16-bit card contains two edge connectors:
  - – one plugs into the original 8-bit connector
  - – the other plugs into the new 16-bit connector
- Figure shows pin-out and placement of the additional connector in relation to the 8-bit connector.

## The 16-bit ISA bus. (a) Both 8- and 16-bit connectors and (b) the pinout of the 16-bit connector.

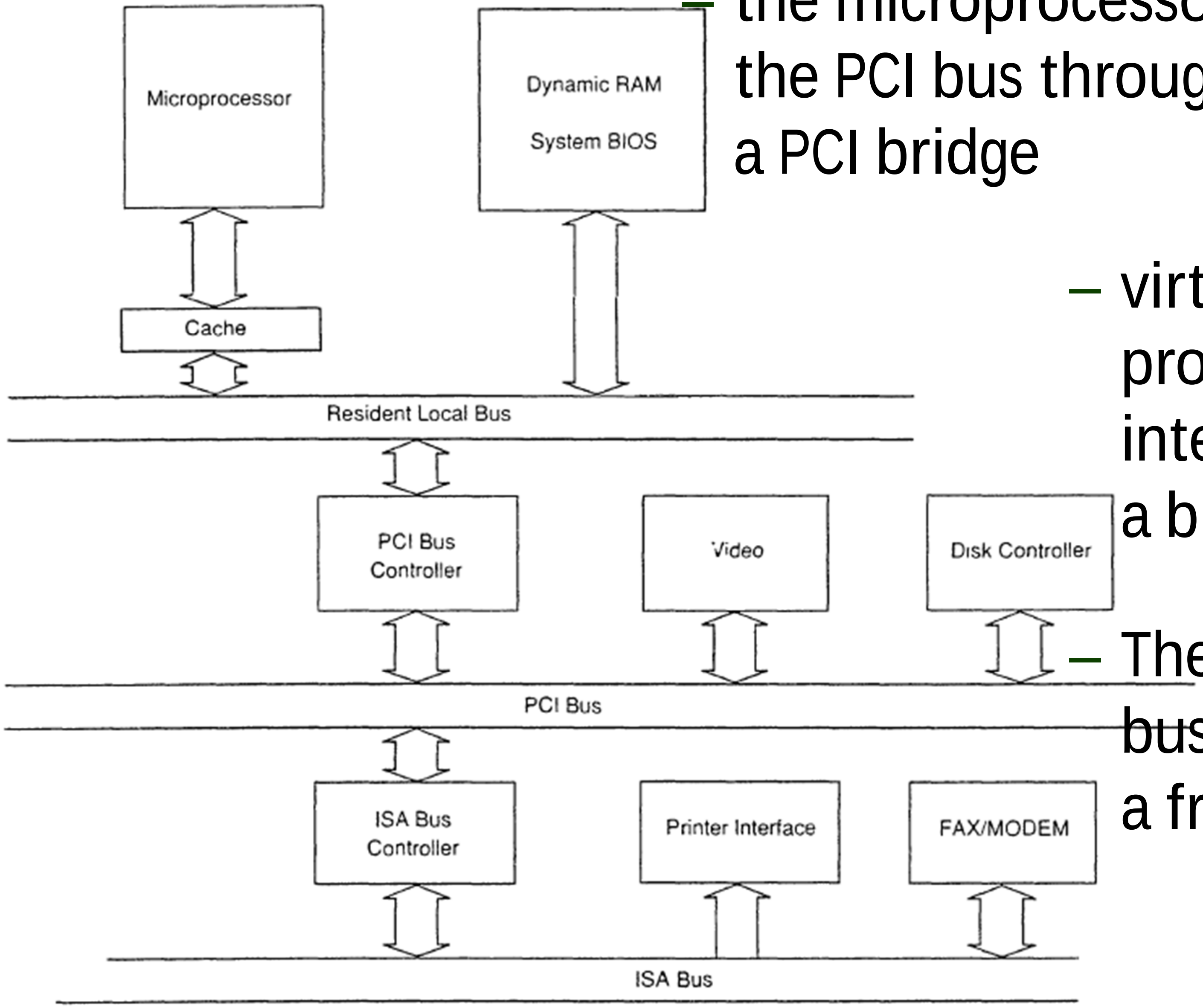


| Pin# | Solder Side | Component Side |
|---|---|---|
| 1 | $\overline{MCS16}$ | BHE |
| 2 | $\overline{IOCS16}$ | A23 |
| 3 | IRQ10 | A22 |
| 4 | IRQ11 | A21 |
| 5 | IRQ12 | A20 |
| 6 | IRQ15 | A19 |
| 7 | IRQ14 | A18 |
| 8 | $\overline{DACKO}$ | A17 |
| 9 | DRQO | $\overline{MEMR}$ |
| 10 | $\overline{DACK5}$ | $\overline{MEMW}$ |
| 11 | DRQ5 | D8 |
| 12 | $\overline{DACK6}$ | D9 |
| 13 | DRQ6 | D10 |
| 14 | $\overline{DACK7}$ | D11 |
| 15 | DRQ7 | D12 |
| 16 | +5V | D13 |
| 17 | $\overline{MASTER}$ | D14 |
| 18 | GND | D15 |

# PERIPHERAL COMPONENT INTERCONNECT (PCI) BUS

- PCI (**peripheral component interconnect**) is virtually the only bus found in new systems.
  - ISA still exists by special order for older cards
- PCI has replaced the VESA local bus.
- PCI has plug-and-play characteristics and ability to function with a 64-bit data bus.
- A PCI interface contains registers, located in a small memory device containing information about the board.
- this allows PC to automatically configure the card
- this provides plug-and-play characteristics to the ISA bus, or any other bus
- Called plug-and-play (PnP), it is the reason PCI has become so popular.

# System block diagram for the PC that contains a PCI bus.



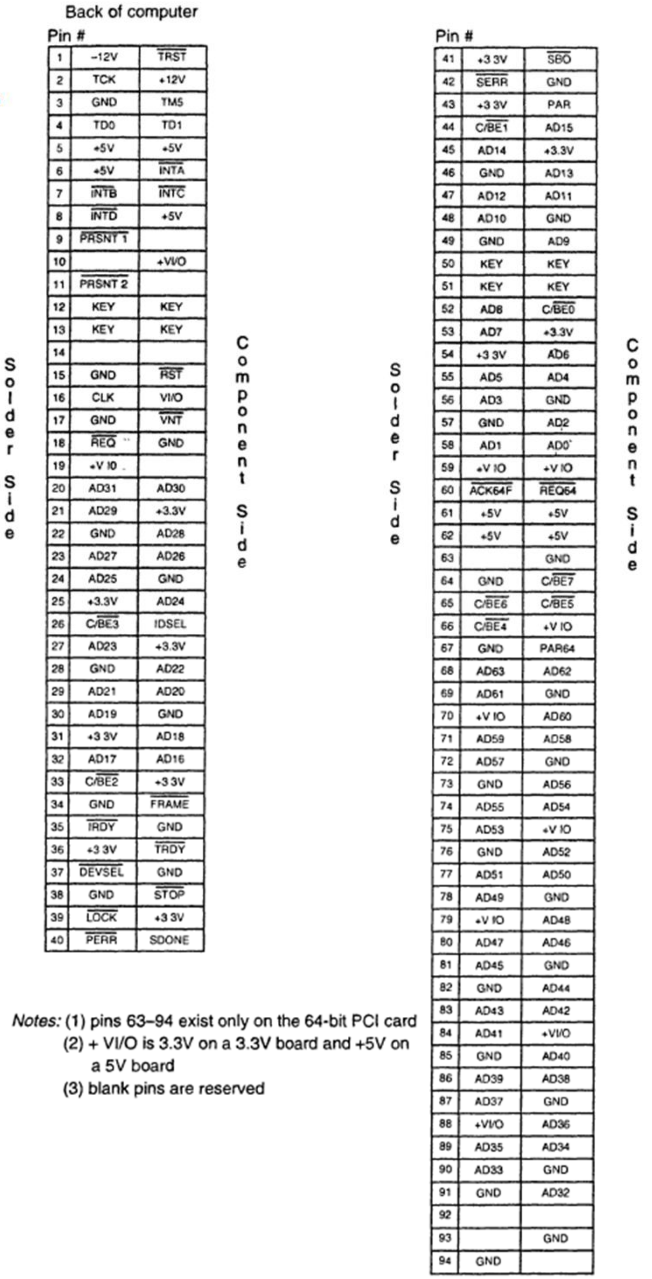
- the microprocessor connects to the PCI bus through an IC called a PCI bridge
- virtually any processor can interface to PCI wit a bridge
- The resident local bus is often called a front side bus

# The PCI Bus Pin-Out

- PCI functions with a 32- or 64-bit data bus and a full 32-bit address bus.
  - address and data buses, labeled $AD_0$–$AD_{63}$ are multiplexed to reduce size of the edge connector
- A 32-bit card has connections 1 through 62, the 64-bit card has all 94 connections.
- The 64-bit card can accommodate a 64-bit address if required at some future point.
- Figure shows the PCI bus pin-out.

## The pin-out of the PCI bus.

- PCI is most often used for I./O interface to the microprocessor
- memory could be interfaced, but with a Pentium, would operate at 33 MHz, half the speed of the Pentium resident local
- PCI 2.1 operates at 66 MHz, and 33 MHz for older interface cards
- P4 systems use 200 MHz bus speed (often listed as 800 MHz)
- there is no planned modification to the PCI bus speed yet



Back of computer

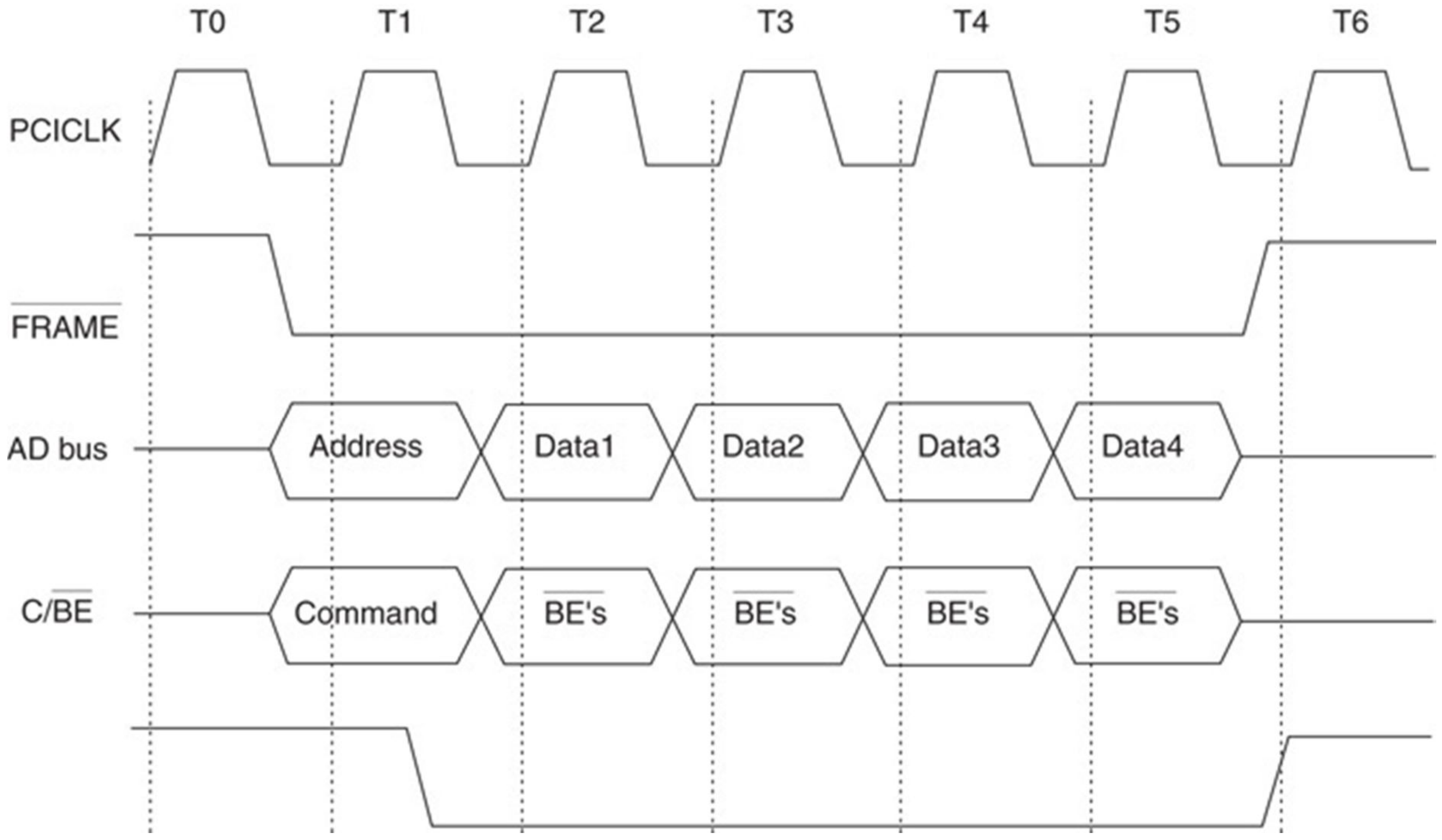
| Pin # | Solder Side | Component Side |
|---|---|---|
| 1 | −12V | $\overline{\text{TRST}}$ |
| 2 | TCK | +12V |
| 3 | GND | TMS |
| 4 | TDO | TD1 |
| 5 | +5V | +5V |
| 6 | +5V | $\overline{\text{INTA}}$ |
| 7 | $\overline{\text{INTB}}$ | $\overline{\text{INTC}}$ |
| 8 | $\overline{\text{INTD}}$ | +5V |
| 9 | $\overline{\text{PRSNT 1}}$ | |
| 10 | | +VI/O |
| 11 | $\overline{\text{PRSNT 2}}$ | |
| 12 | KEY | KEY |
| 13 | KEY | KEY |
| 14 | | |
| 15 | GND | $\overline{\text{RST}}$ |
| 16 | CLK | VI/O |
| 17 | GND | $\overline{\text{VNT}}$ |
| 18 | $\overline{\text{REQ}}$ | GND |
| 19 | +V IO | |
| 20 | AD31 | AD30 |
| 21 | AD29 | +3.3V |
| 22 | GND | AD28 |
| 23 | AD27 | AD26 |
| 24 | AD25 | GND |
| 25 | +3.3V | AD24 |
| 26 | C/$\overline{\text{BE3}}$ | IDSEL |
| 27 | AD23 | +3.3V |
| 28 | GND | AD22 |
| 29 | AD21 | AD20 |
| 30 | AD19 | GND |
| 31 | +3 3V | AD18 |
| 32 | AD17 | AD16 |
| 33 | C/$\overline{\text{BE2}}$ | +3 3V |
| 34 | GND | $\overline{\text{FRAME}}$ |
| 35 | $\overline{\text{IRDY}}$ | GND |
| 36 | +3 3V | $\overline{\text{TRDY}}$ |
| 37 | $\overline{\text{DEVSEL}}$ | GND |
| 38 | GND | $\overline{\text{STOP}}$ |
| 39 | $\overline{\text{LOCK}}$ | +3 3V |
| 40 | $\overline{\text{PERR}}$ | SDONE |
| 41 | +3 3V | $\overline{\text{SBO}}$ |
| 42 | $\overline{\text{SERR}}$ | GND |
| 43 | +3 3V | PAR |
| 44 | C/$\overline{\text{BE1}}$ | AD15 |
| 45 | AD14 | +3.3V |
| 46 | GND | AD13 |
| 47 | AD12 | AD11 |
| 48 | AD10 | GND |
| 49 | GND | AD9 |
| 50 | KEY | KEY |
| 51 | KEY | KEY |
| 52 | AD8 | C/$\overline{\text{BE0}}$ |
| 53 | AD7 | +3.3V |
| 54 | +3 3V | AD6 |
| 55 | AD5 | AD4 |
| 56 | AD3 | GND |
| 57 | GND | AD2 |
| 58 | AD1 | AD0 |
| 59 | +V IO | +V IO |
| 60 | $\overline{\text{ACK64F}}$ | $\overline{\text{REQ64}}$ |
| 61 | +5V | +5V |
| 62 | +5V | +5V |
| 63 | | GND |
| 64 | GND | C/$\overline{\text{BE7}}$ |
| 65 | C/$\overline{\text{BE6}}$ | C/$\overline{\text{BE5}}$ |
| 66 | C/$\overline{\text{BE4}}$ | +V IO |
| 67 | GND | PAR64 |
| 68 | AD63 | AD62 |
| 69 | AD61 | GND |
| 70 | +V IO | AD60 |
| 71 | AD59 | AD58 |
| 72 | AD57 | GND |
| 73 | GND | AD56 |
| 74 | AD55 | AD54 |
| 75 | AD53 | +V IO |
| 76 | GND | AD52 |
| 77 | AD51 | AD50 |
| 78 | AD49 | GND |
| 79 | +V IO | AD48 |
| 80 | AD47 | AD46 |
| 81 | AD45 | GND |
| 82 | GND | AD44 |
| 83 | AD43 | AD42 |
| 84 | AD41 | +VI/O |
| 85 | GND | AD40 |
| 86 | AD39 | AD38 |
| 87 | AD37 | GND |
| 88 | +VI/O | AD36 |
| 89 | AD35 | AD34 |
| 90 | AD33 | GND |
| 91 | GND | AD32 |
| 92 | | |
| 93 | | GND |
| 94 | GND | |

*Notes:* (1) pins 63–94 exist only on the 64-bit PCI card
(2) + VI/O is 3.3V on a 3.3V board and +5V on a 5V board
(3) blank pins are reserved

# The PCI Address/Data Connections

- The PCI address appears on $AD_0$–$AD_{31}$ and is multiplexed with data.
  - – some systems have a 64-bit data bus using $AD_{32}$–$AD_{63}$ for data transfer only
  - – these pins can be used for extending the address to 64 bits
- Figure shows the PCI bus timing diagram
  - – which shows the address multiplexed with data and control signals used for multiplexing

The basic timing for the PCI bus system. Note that this transfers either four 32-bit numbers (32-bit PCI) or four 64-bit numbers (64-bit PCI).



# Configuration Space

- PCI contains a 256-byte memory to allow the PC to interrogate the PCI interface.
  - this feature allows the system to automatically configure itself for the PCI plug-board
  - Microsoft calls this plug-and-play (PnP)
- The first 64 bytes contain information about the PCI interface.
- The first 32-bit doubleword contains the unit ID code and the vendor ID code.

# PCI Express Bus

- The PCI Express transfers data in serial at 2.5 GHz
- 250 MBps to 8 GBps for PCI Express interfaces
  - standard PCI delivers data at about 133 MBps
- Each serial connection on the PCI Express bus is called a **lane**.
  - slots on the main board are single lane slots with a total transfer speed of 1 GBps
- A PCI Express video card connector currently has 16 lanes with a transfer speed of 4 GBps.

- The standard allows up to 32 lanes.
  - at present the widest is the 16 lanes video card
- Most main boards contain four single lane slots for peripherals and one 16 lane slot for the video card.
  - a few newer boards contain two 16 lane slots
- PCI Express 2 bus was released in late 2007.
  - transfer speed from 250 MBps to 500 MBps, twice that of the PCI Express
- PCI is replacing most current video cards on the AGP port with the PCI Express bus.