

# INTERFACING ICs

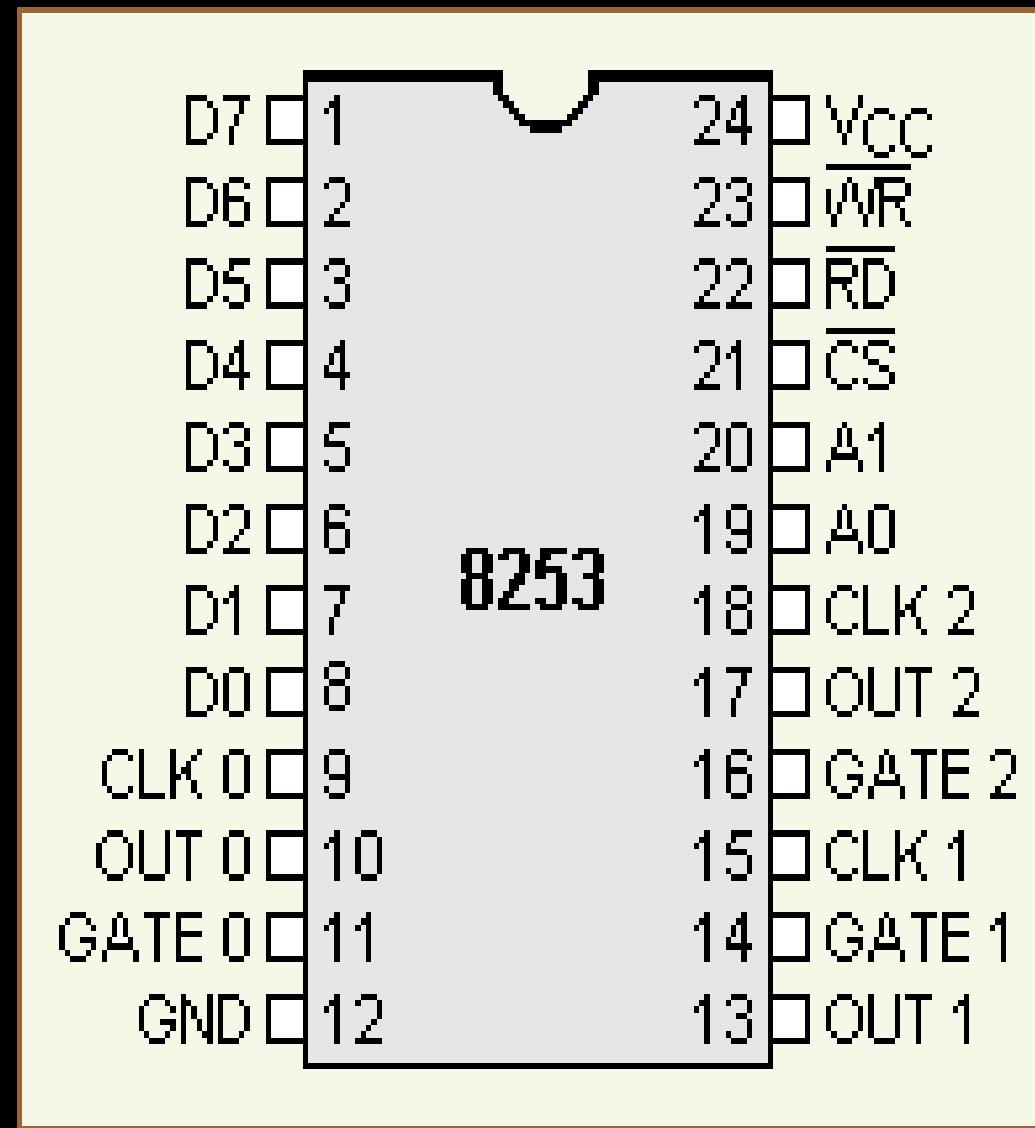
# Topics

- **Interfacing chips**
  - *Programmable Communication Interface – PCI (8251)*
  - *Programmable Interval Timer (8253)*
  - *Programmable Peripheral Interfacing - PPI (8255)*
  - *Programmable DMA controller (8257)*
  - *Programmable Interrupt Controller (8259)*
  - *Programmable Keyboard Display Interface (8279)*

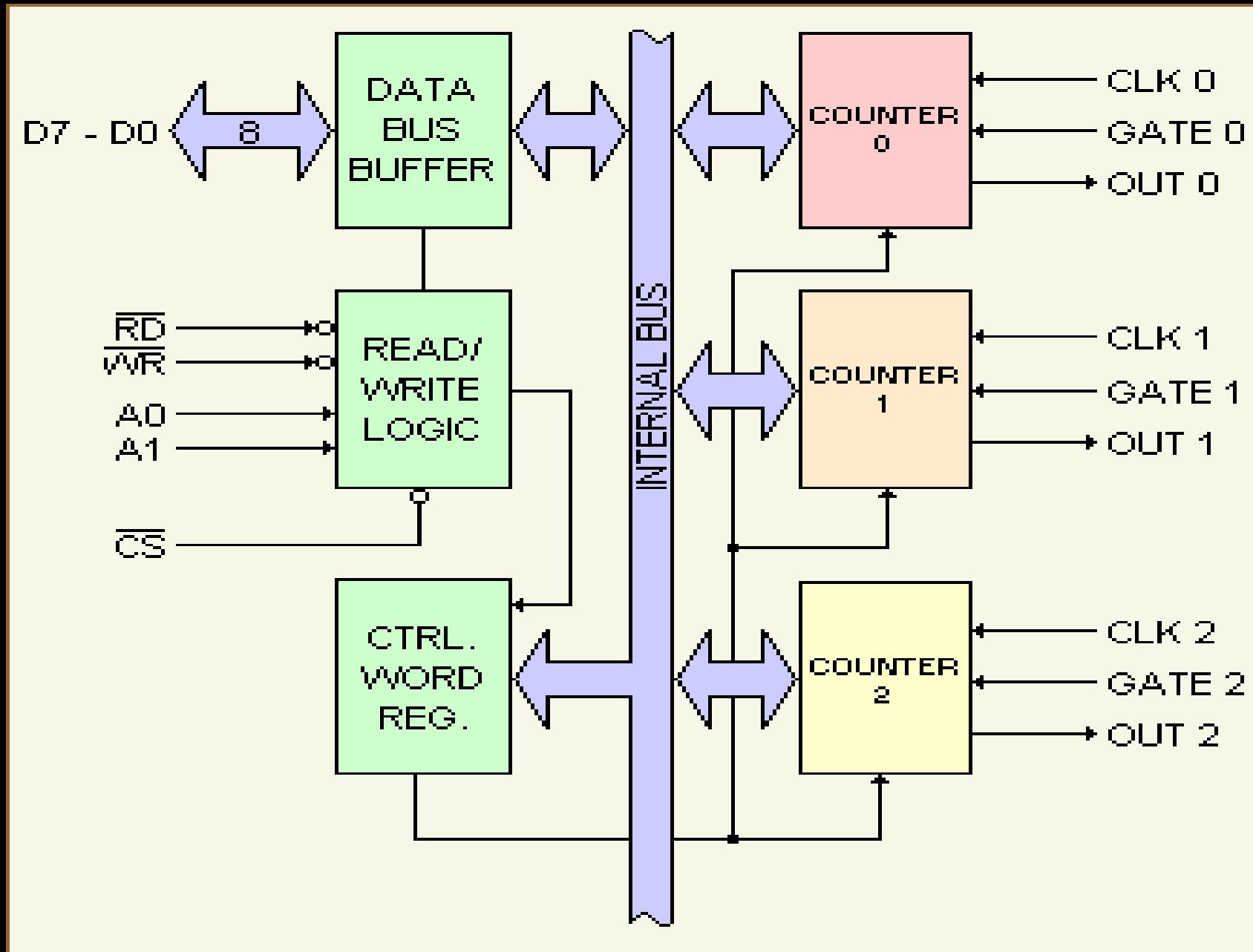
**8253/8254**

**TIMER**

# 8253 Pin Diagram



# 8253 Block Diagram



# Pin Description

- **Clock:** This is the clock input for the counter. The counter is 16 bits.
  - The maximum clock frequency is 1 / 380 nanoseconds or 2.6 megahertz. The minimum clock frequency is DC or static operation.
- **Out:** This single output line is the signal that is the final programmed output of the device.
  - Actual operation of the out line depends on how the device has been programmed.
- **Gate:** This input can act as a gate for the clock input line, or it can act as a start pulse, depending on the programmed mode of the counter.

# Counter Features

- Each counter is **identical**, and each consists of a **16-bit**, pre-settable, **down** counter.
- Each is **fully independent** and can be easily read by the CPU.
- When the counter is read, the data within the counter **will not be disturbed**.
- This allows the system or your own program to **monitor** the counter's value at any time, without disrupting the overall function of the 8253.

# Counter Selection

	$\overline{RD}$	$\overline{WR}$	A0	A1	function
COUNTER 0	1	0	0	0	Load counter 0
	0	1	0	0	Read counter 0
COUNTER 1	1	0	0	1	Load counter 1
	0	1	0	1	Read counter 1
COUNTER 2	1	0	1	0	Load counter 2
	0	1	1	0	Read counter 2
MODE WORD or CONTROL WORD	1	0	1	1	Write mode word
--	0	1	1	1	No-operation

# Control Word Register

- This internal register is used to write information to, **prior to using the device**.
- This register is addressed when **A0** and **A1** inputs are logical 1's.
- The data in the register **controls the operation mode** and the selection of either binary or BCD counting format.
- The register can **only** be written to.
- You **can't read** information from the register.

# Control Word Format

CONTROL BYTE D7 - D0											
D7	D6	D5	D4	D3	D2	D1	D0				
SC1	SC0	RL1	RL0	M2	M1	M0	BCP				
D7 SC1	D6 SC0	D5 RL1	D4 RL0	R	D3 M2	D2 M1	D1 M0	M	D0	counts down in	
0	0	0	0	Co th	0	0	0	rr	0	binary	
0	1			cc lat	0	0	1		1	BCD	al count
1	0			Re	x	1	0		mode 1: programmable one-shot		
1	1			Re	x	1	0		mode 2: rate generator		
				Re	x	1	1		mode 3: square wave generator		
				Re	1	0	0		mode 4: software triggered strobe		
				th	1	0	1		mode 5: hardware triggered strobe		

Once a counter is set up, it will remain that way until it is changed by another control word.

# Different uses of the 8253 gate input pin

Signal Status	Low or going low	Rising	High
Mode			
0	Disables counting	--	Enables counting
1	--	1) Initiates counting 2) Resets output after next clock	--
2	1) Disables counting 2) Sets output immediately high	1) Reloads counter 2) Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	--	Enables counting
5	--	Initiates counting	--

This table shows the different uses of the 8253 gate input pin.

Each mode of operation for the counter has a different use for the GATE input pin.

# Timer Modes - Mode 0

- **Interrupt on Terminal Count**
- The counter will be programmed to an initial value and afterwards **counts down** at a rate equal to the input clock frequency(8 MHz).
- When the count is equal to 0, the **OUT pin** will be a logical 1.
- The output will stay a logical 1 **until the counter is reloaded** with a new value or the same value or **until a mode word is written** to the device.
- Once the counter starts counting down, the **GATE input can disable the internal counting** by setting the GATE to a logical 0.

# Timer Modes - Mode 1

- **Programmable One-Shot**
- In mode 1, the device can be setup to give an **output pulse** that is an integer number of clock pulses.
- The one-shot is **triggered** on the rising edge of the GATE input.
- If the trigger occurs during the pulse output, the 8253 will be **retriggered** again.

# Timer Modes - Mode 2

- **Rate Generator**
- The counter that is programmed for mode 2 becomes a "**divide by n**" counter.
- The **OUT pin** of the counter goes to low for one input clock period.
- The time between the pulses of going low is dependent on the **present count in the counter's register**.

# Timer Modes - Mode 2

- For example, suppose to get an output frequency of **1,000 Hz**, the period would be  **$1 / 1,000 \text{ s} = 1 \text{ ms}$  or  $1,000 \mu\text{s}$** .
- If an input clock of **1 MHz** were applied to the clock input of the counter #0, then the counter #0 would need to be programmed to  $1000 \mu\text{s}$ .
- This could be done in **decimal or in BCD**. (The period of an input clock of 1 MHz is  $1 / 1,000,000 = 1 \mu\text{s}$ .)
- The formula is:  **$n=f_i/f_{out}$** , where  $f_i$  = input clock frequency,  $f_{out}$  = output frequency,  $n$  = value to be loaded.

# Timer Modes - Mode 3

- **Square Wave Generator**
- Mode 3 is similar to the mode 2 except that the output will be high **for half the period** and low **for half**.
- If the count is odd, the output will be high for  $(n+1)/2$  and low for  $(n-1)/2$  counts.

# Timer Modes - Mode 4

- **Software Triggered Strobe**
- In this mode the programmer can set up the counter to give an **output timeout** starting when the register is loaded.
- **On the terminal count**, when the counter equals to 0, the output will go to a logical 0 for one clock period and then returns to a logical 1.
- Firstly, when the mode is set, the **output will be a logical 1**.

# Timer Modes - Mode 5

- **Hardware Triggered Strobe**
- In this mode **the rising edge of the trigger input** will start the counting of the counter.
- The **output goes low for one clock** at the terminal count.
- The counter is **retriggerable**, thus meaning that if the trigger input is taken low and then high during a count sequence, the sequence will start over.
- When the external trigger input goes to a logical 1, **the timer will start to time out**.
- If the **external trigger occurs again**, prior to the time completing a full timeout, the timer will retrigger.

**82555**  
**PPI**

# Introduction

- PPI – Programmable Peripheral Interface
- It is an I/O port chip used for interfacing I/O devices with microprocessor
- Very commonly used peripheral chip
- Knowledge of 8255 essential for students in the Microprocessors lab for Interfacing experiments

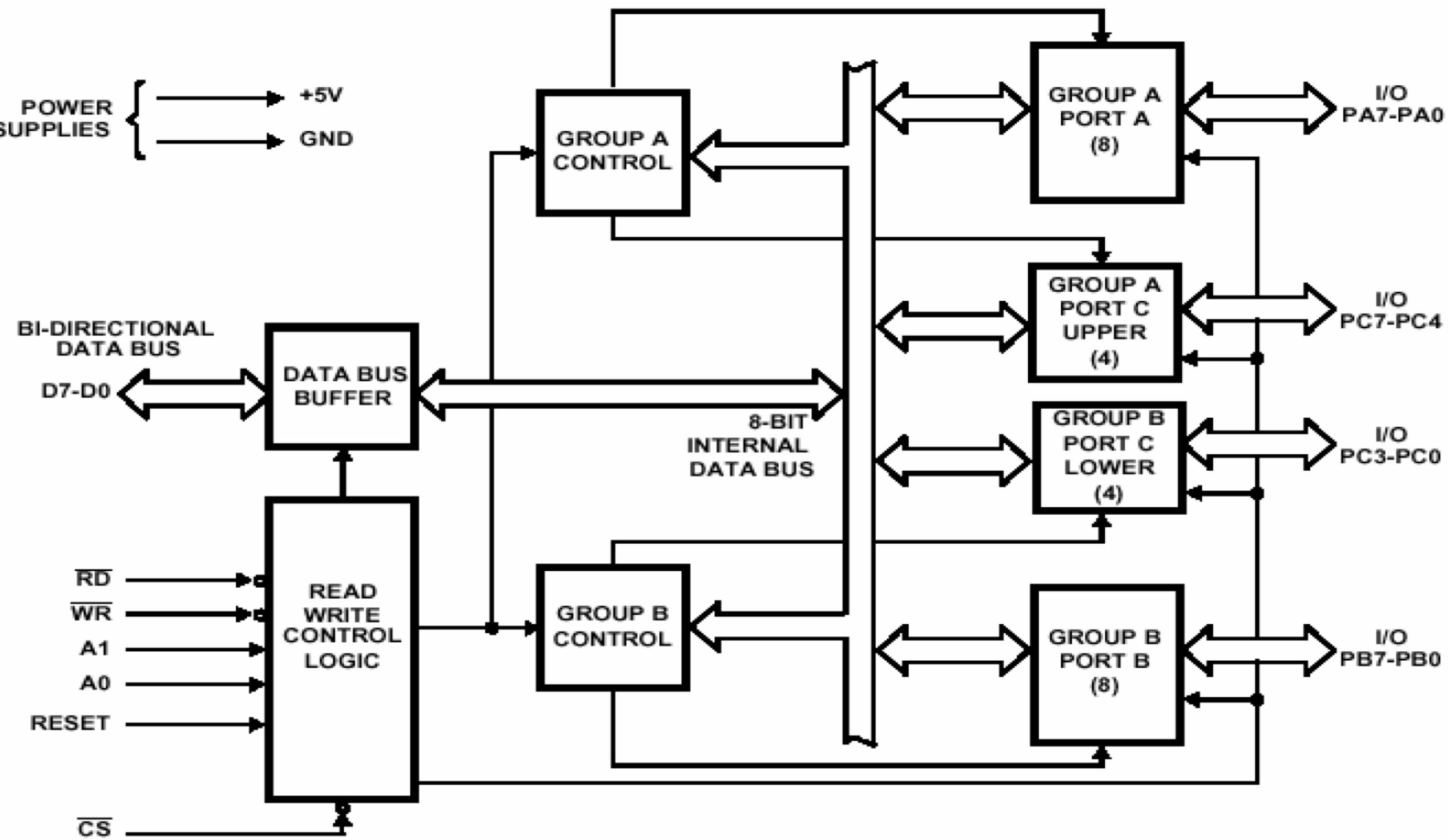
# 8255 Ports

- 8255 PPI has three 8-bit ports.

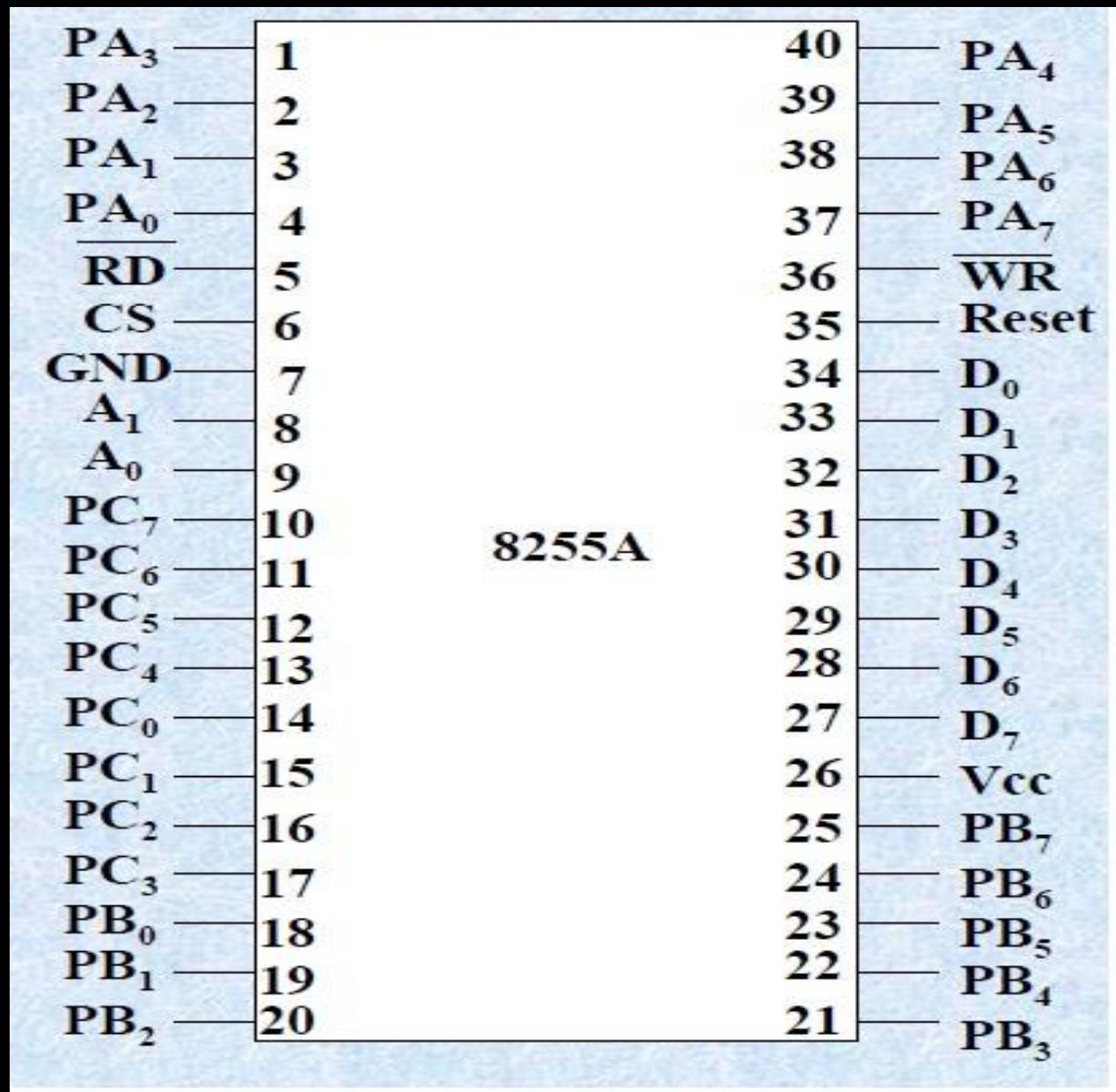
- Port A (PA)
- Port B (PB)
- Port C (PC)

A1	A0	Selected port
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control port

- Port C composed of two independent 4-bit ports: **PC7-4 (PC Upper)** and **PC3-0 (PC Lower)**
- **Port A, Port B, Port C and Control port will have the addresses as 7CH, 7DH, 7EH, and 7FH respectively.**



# Pin Diagram



# Pin Description

- **PA7-PA0** : These are eight port A lines that acts as either latched output or buffered input lines depending upon the control word loaded into the control word register.
- **PC7-PC4** : Upper nibble of port C lines. They may act as either output latches or input buffers lines.  
*This port also can be used for generation of handshake lines in mode 1 or mode 2.*
- **PC3-PC0** : These are the lower port C lines, other details are the same as PC7-PC4 lines.
- **PB0-PB7** : These are the eight port B lines which are used as latched output lines or buffered input lines in the same way as port A.

# Pin Description

- **RD** : This is the input line driven by the microprocessor and should be low to indicate read operation to 8255.
- **WR** : This is an input line driven by the microprocessor. A low on this line indicates write operation.
- **CS** : This is a chip select line. If this line goes low, it enables the 8255 to respond to RD and WR signals, otherwise RD and WR signal are neglected.
- **A1-A0** : These are the address input lines and are driven by the microprocessor.
- **RESET** : The 8255 is placed into its reset state if this input line is logical 1. All peripheral ports are set to the input mode.

<b>RD</b>	<b>WR</b>	<b>CS</b>	<b>A<sub>1</sub></b>	<b>A<sub>0</sub></b>	<b>Input (Read) cycle</b>
0	1	0	0	0	Port A to Data bus
0	1	0	0	1	Port B to Data bus
0	1	0	1	0	Port C to Data bus
0	1	0	1	1	CWR to Data bus

<b>RD</b>	<b>WR</b>	<b>CS</b>	<b>A<sub>1</sub></b>	<b>A<sub>0</sub></b>	<b>Output (Write) cycle</b>
1	0	0	0	0	Data bus to Port A
1	0	0	0	1	Data bus to Port B
1	0	0	1	0	Data bus to Port C
1	0	0	1	1	Data bus to CWR

<b>RD</b>	<b>WR</b>	<b>CS</b>	<b>A<sub>1</sub></b>	<b>A<sub>0</sub></b>	<b>Function</b>
X	X	1	X	X	Data bus tristated
1	1	0	X	X	Data bus tristated

## Control Word Register

# Programming 8255

- 8255 has three operation modes: *mode 0, mode 1, and mode 2*

*Mode 0 - Simple Input or Output mode*

*Mode 1 - Input or Output with Handshake mode*

*Mode 2 - Bidirectional Data Transfer mode*

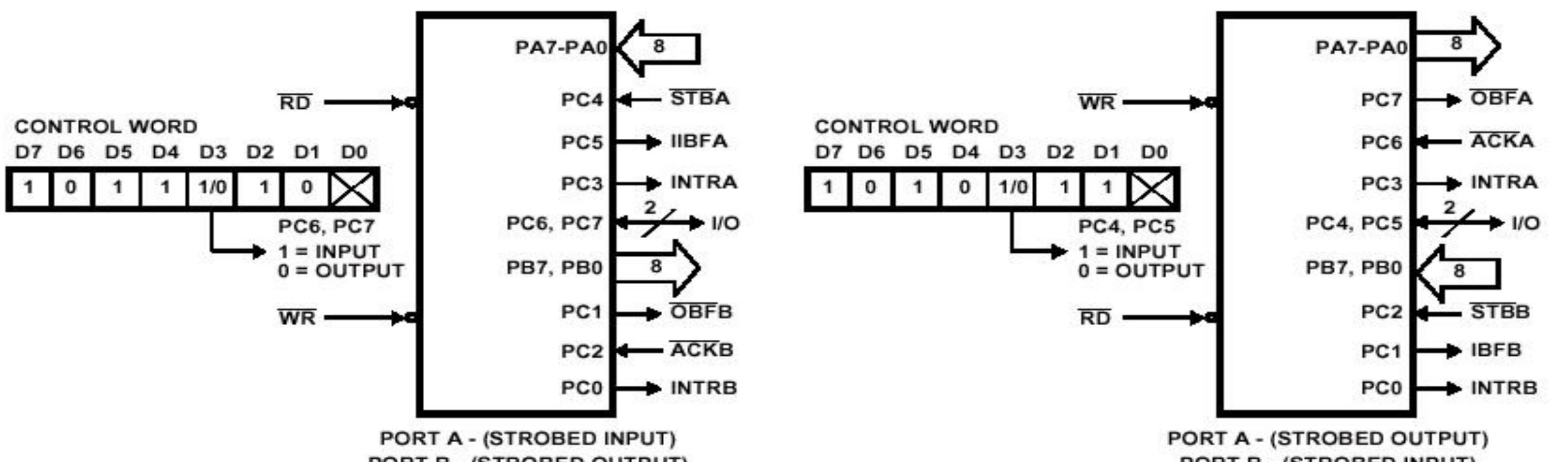
# Mode 0 - Simple Input or Output

- In this mode, ports **A**, **B** are used as **two simple 8-bit I/O ports** & port **C** as **two independent 4-bit ports**.
- **Each port** can be programmed to function as simply an input port or an output port.
- The **input/output features** in Mode 0 are as follows.
  1. *Outputs are latched.*
  2. *Inputs are not latched.*
  3. *Ports don't have handshake or interrupt capability.*

# Mode 1 - Input or Output with Handshake

- In this mode, **handshake signals are exchanged** between the MPU and peripherals prior to data transfer.
- The **features** of the mode include the following:
  1. Two ports (**A** and **B**) function as 8-bit I/O ports.  
They can be configured as either as input or output ports.
  2. Each port uses **three lines from port C as handshake signals**.  
The remaining two lines of Port C can be used for simple I/O operations.
  3. Input and Output data are latched.
  4. Interrupt logic is supported.

# Mode 1 - Input or Output with Handshake



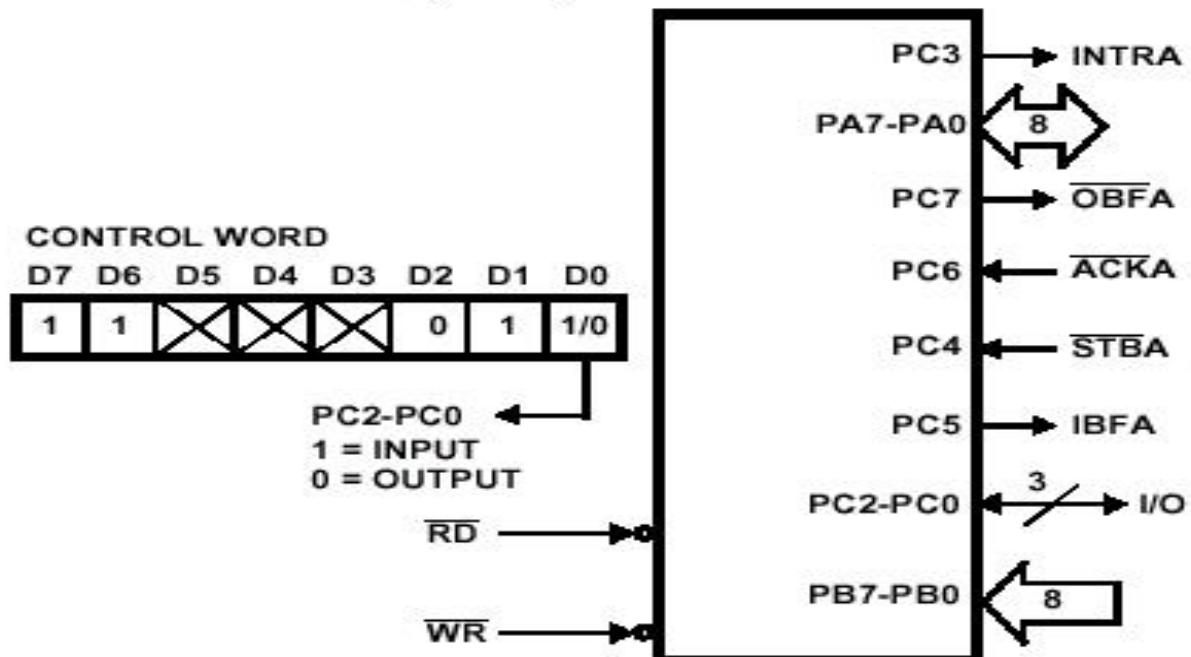
Combinations of Mode 1: Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

FIGURE 10. COMBINATIONS OF MODE 1

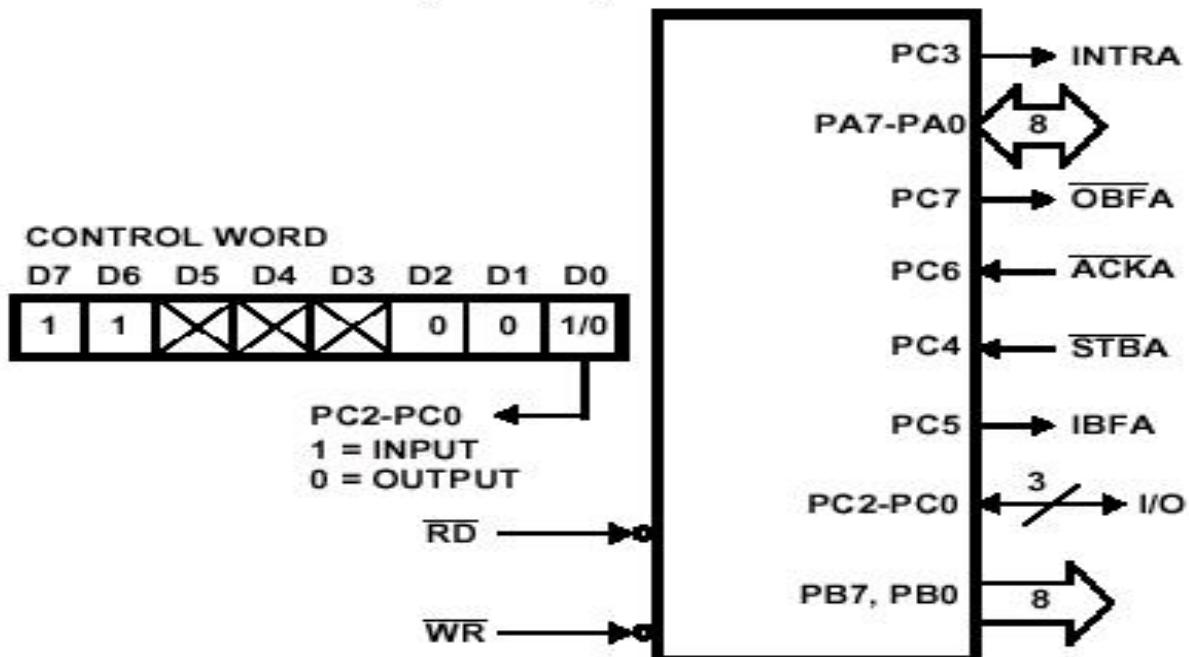
# Mode 2 - Bidirectional Data Transfer

- This mode is used primarily in applications such as **data transfer between two computers**.
- In this mode, **Port A** can be configured as the bidirectional port, **Port B** either in Mode 0 or Mode 1.
- **Port A uses five signals from Port C as handshake signals for data transfer.**
- The remaining three signals from **Port C** can be used either as simple I/O or as handshake for port B.

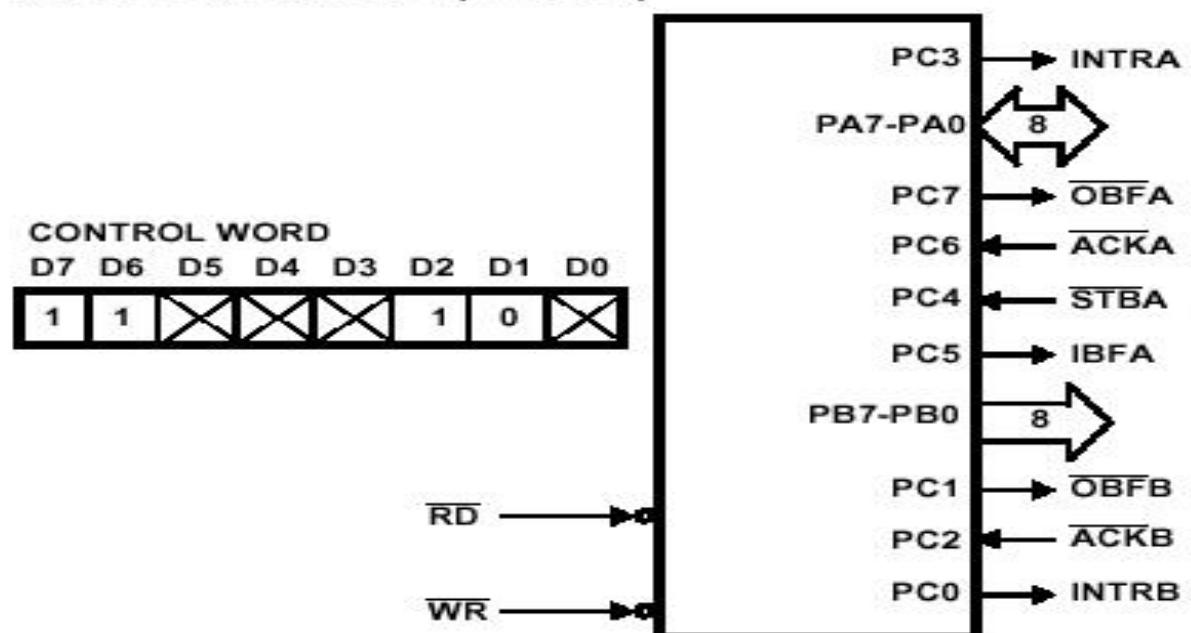
MODE 2 AND MODE 0 (INPUT)



MODE 2 AND MODE 0 (OUTPUT)



MODE 2 AND MODE 1 (OUTPUT)



MODE 2 AND MODE 1 (INPUT)

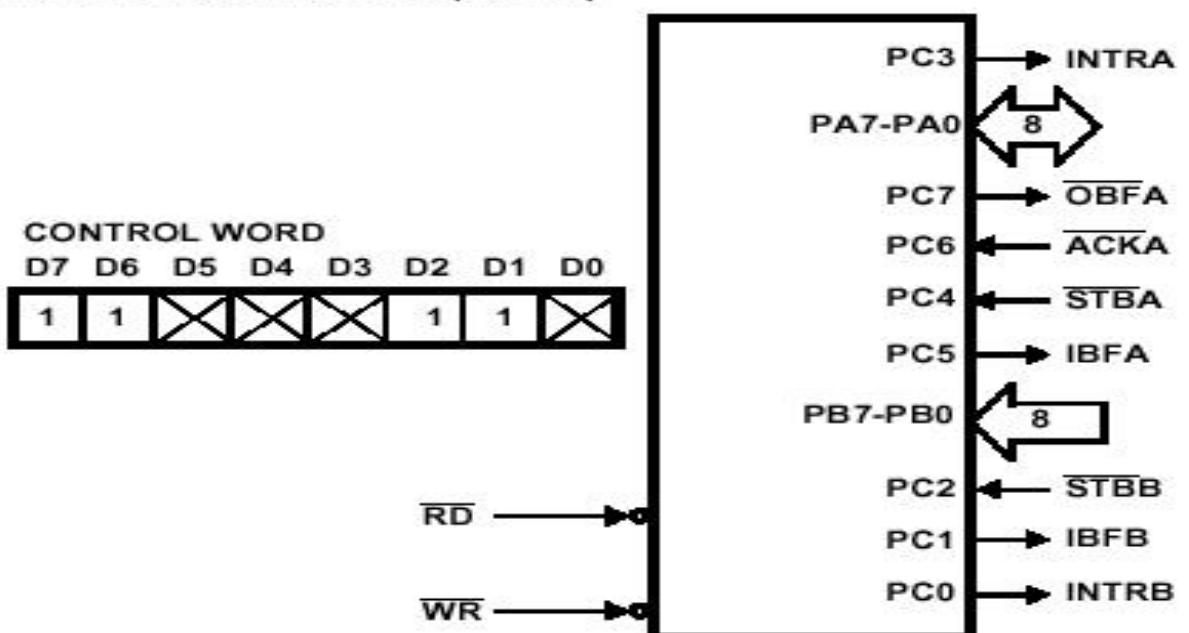


FIGURE 14. MODE 2 COMBINATIONS

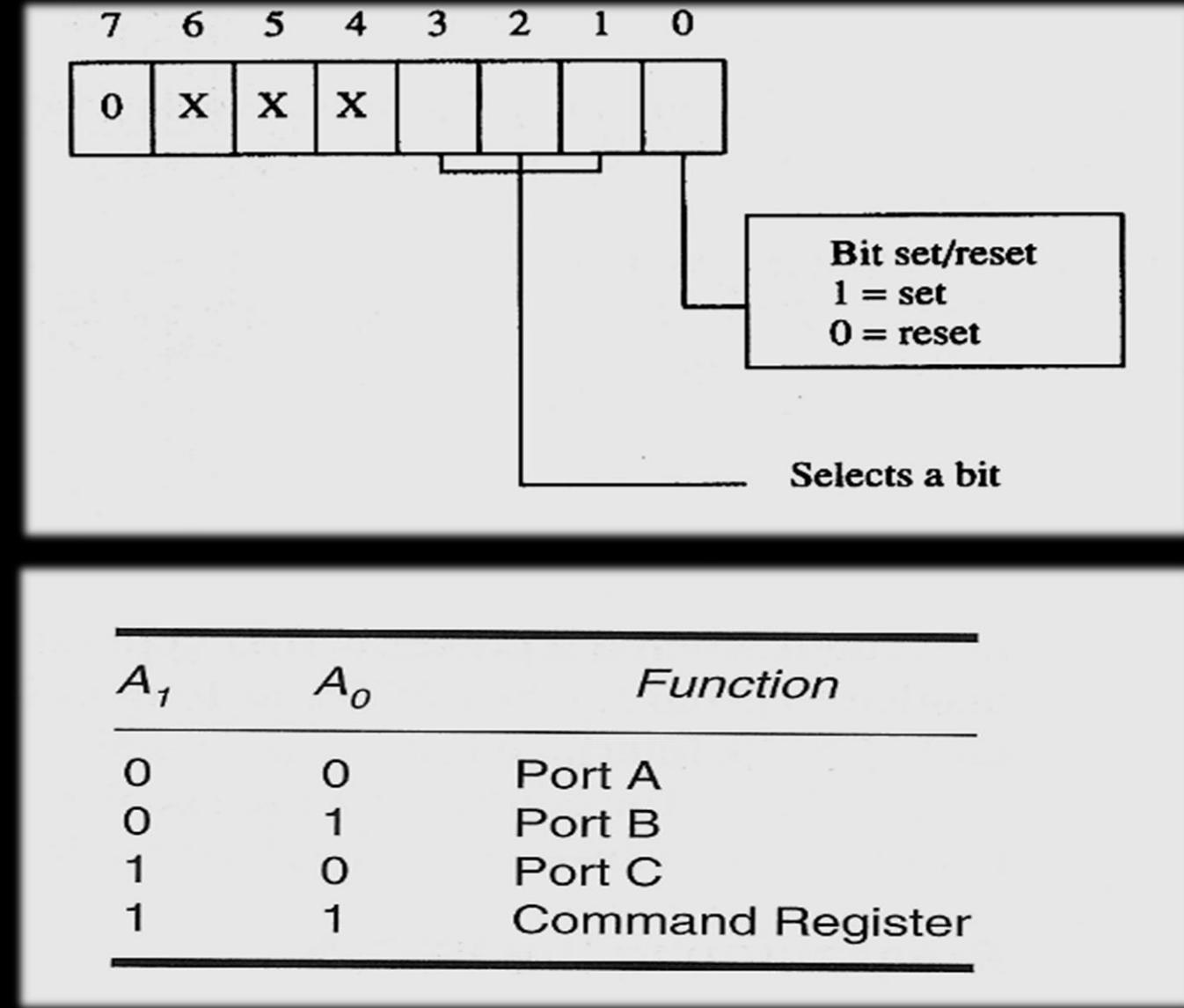
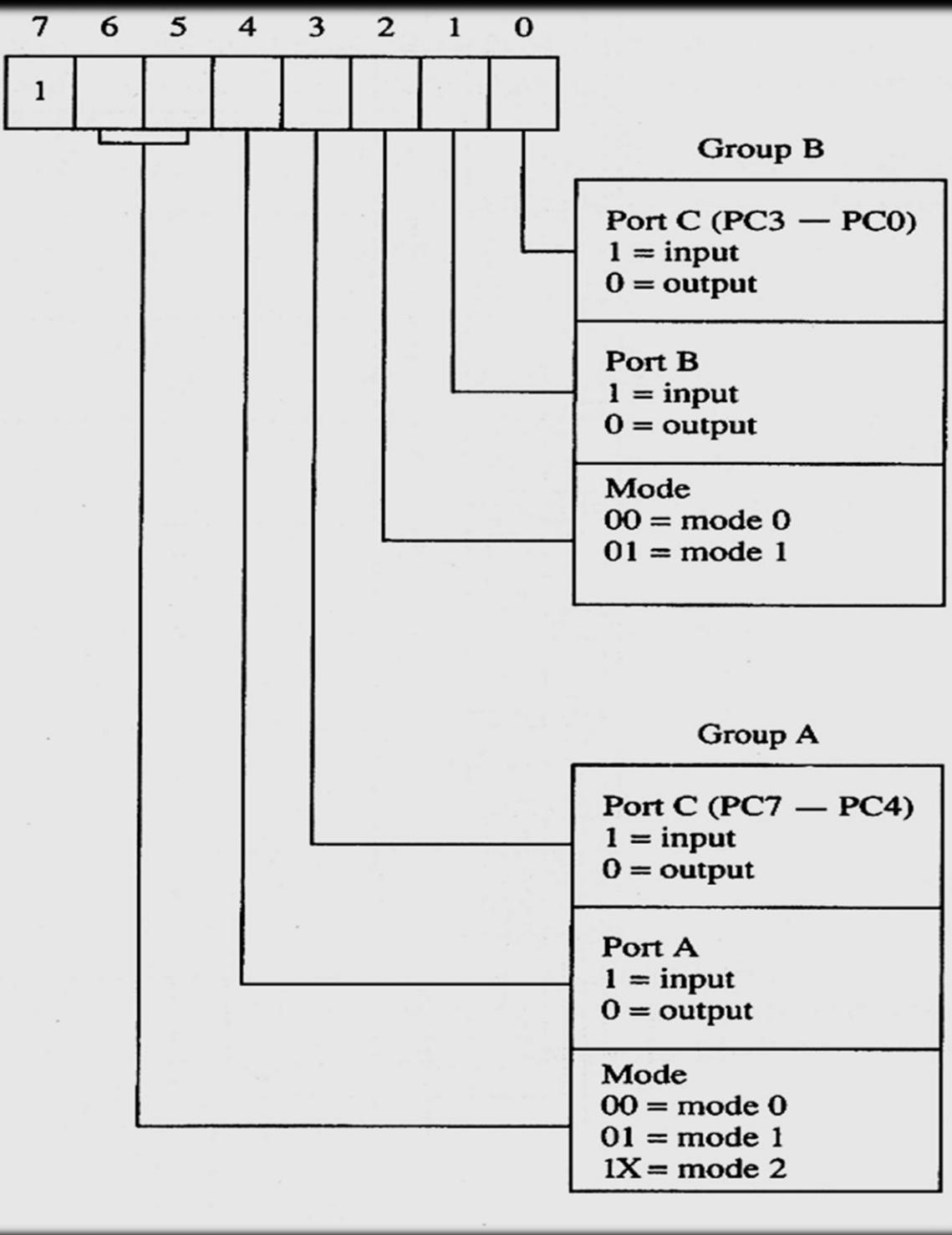
# 8255 Modes Summary

- Port A can work in Mode 0, Mode 1, or Mode 2
- Port B can work in Mode 0, or Mode 1
- Port C can work in Mode 0 only, if at all
- Port A, Port B and Port C can work in Mode 0
- Port A and Port B can work in Mode 1
- Only Port A can work in Mode 2

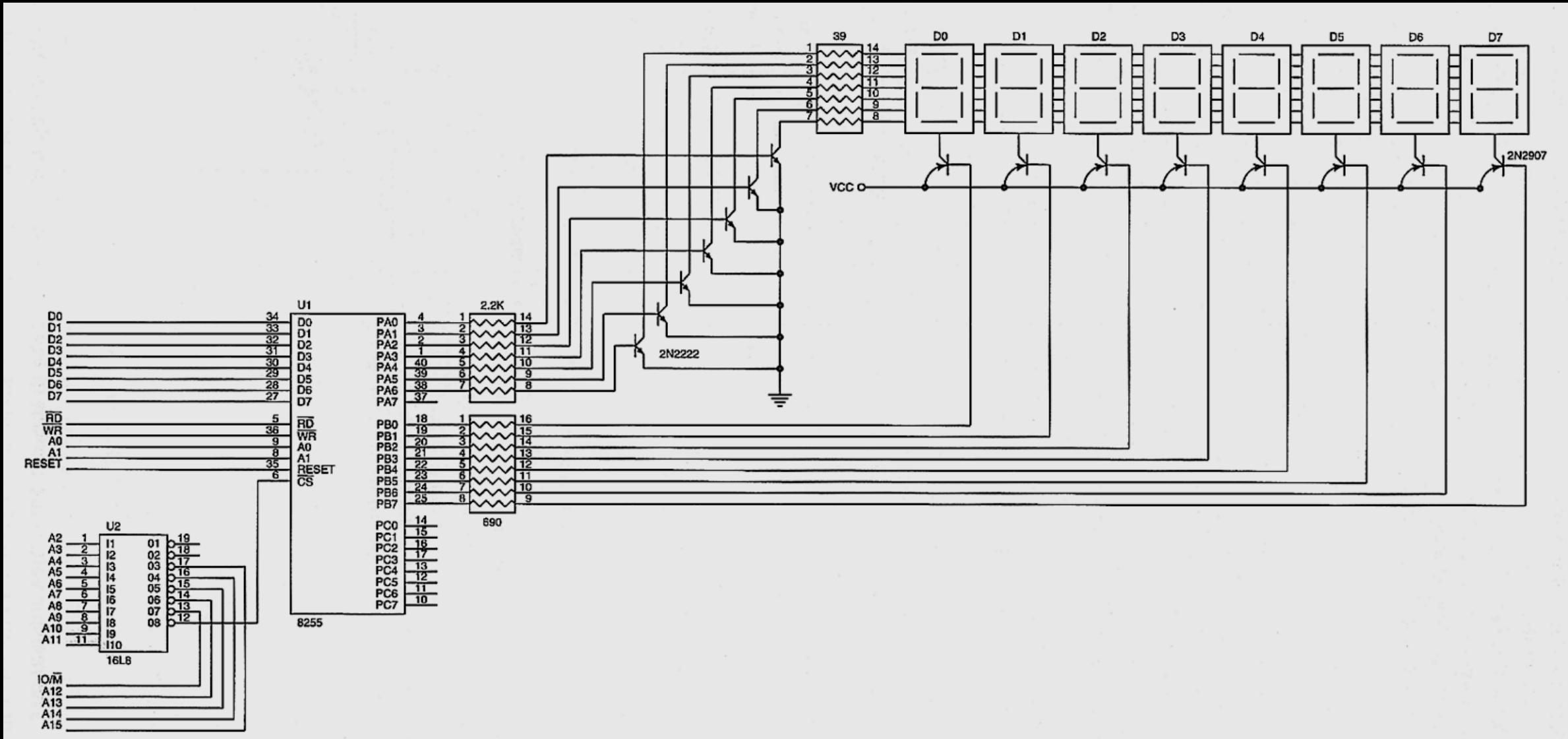
# 8255 Control Words

- There are 2 control words in 8255.
  1. Mode Definition (MD) Control word and
  2. Bit Set / Reset (BSR) Control Word
- MD control word configures the ports of 8255 as input or output in Mode 0, 1, or 2.
- PCBSR control word is used to set to 1 or reset to 0 any one selected bit of Port C

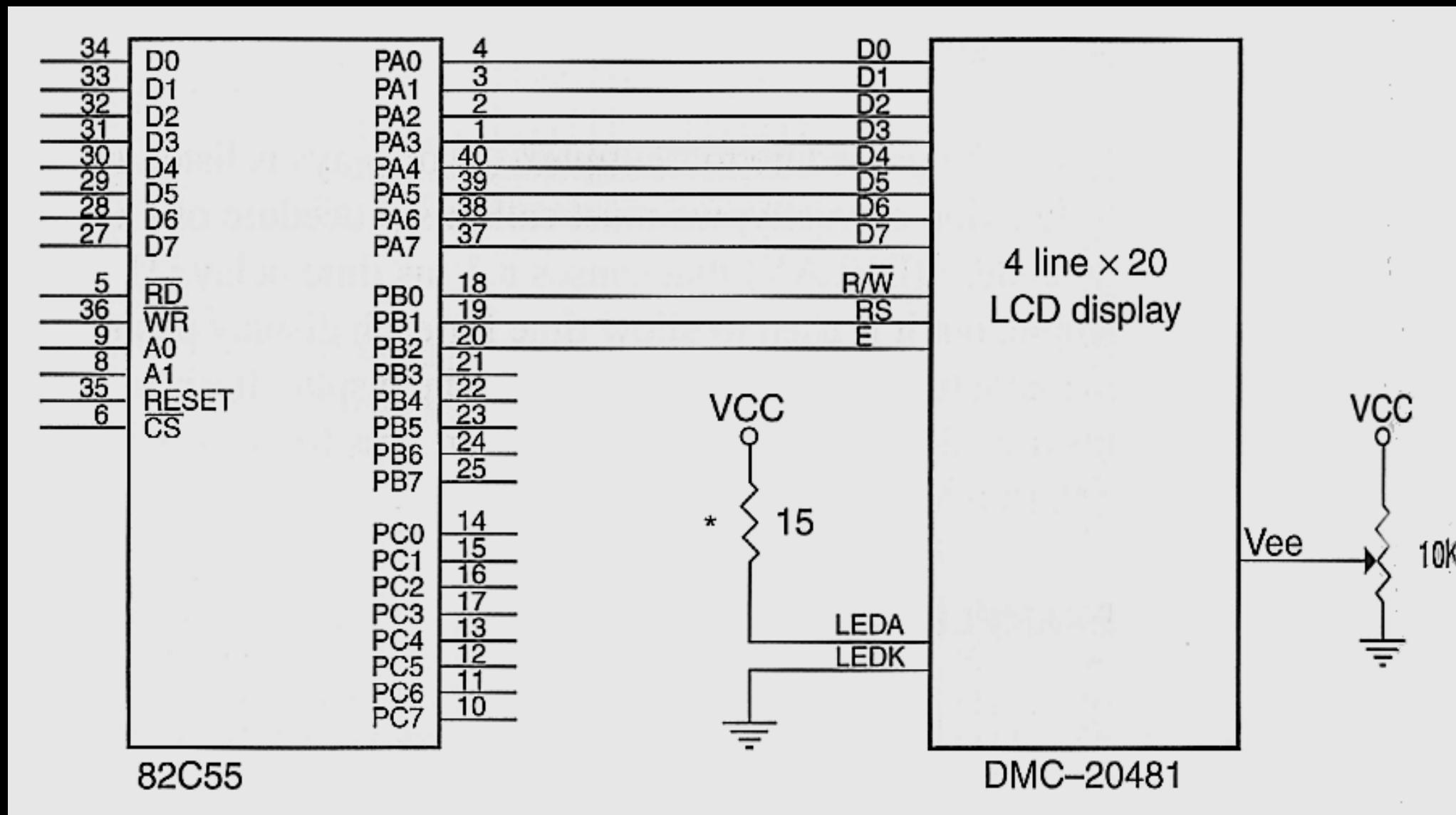
# 8255 Control words



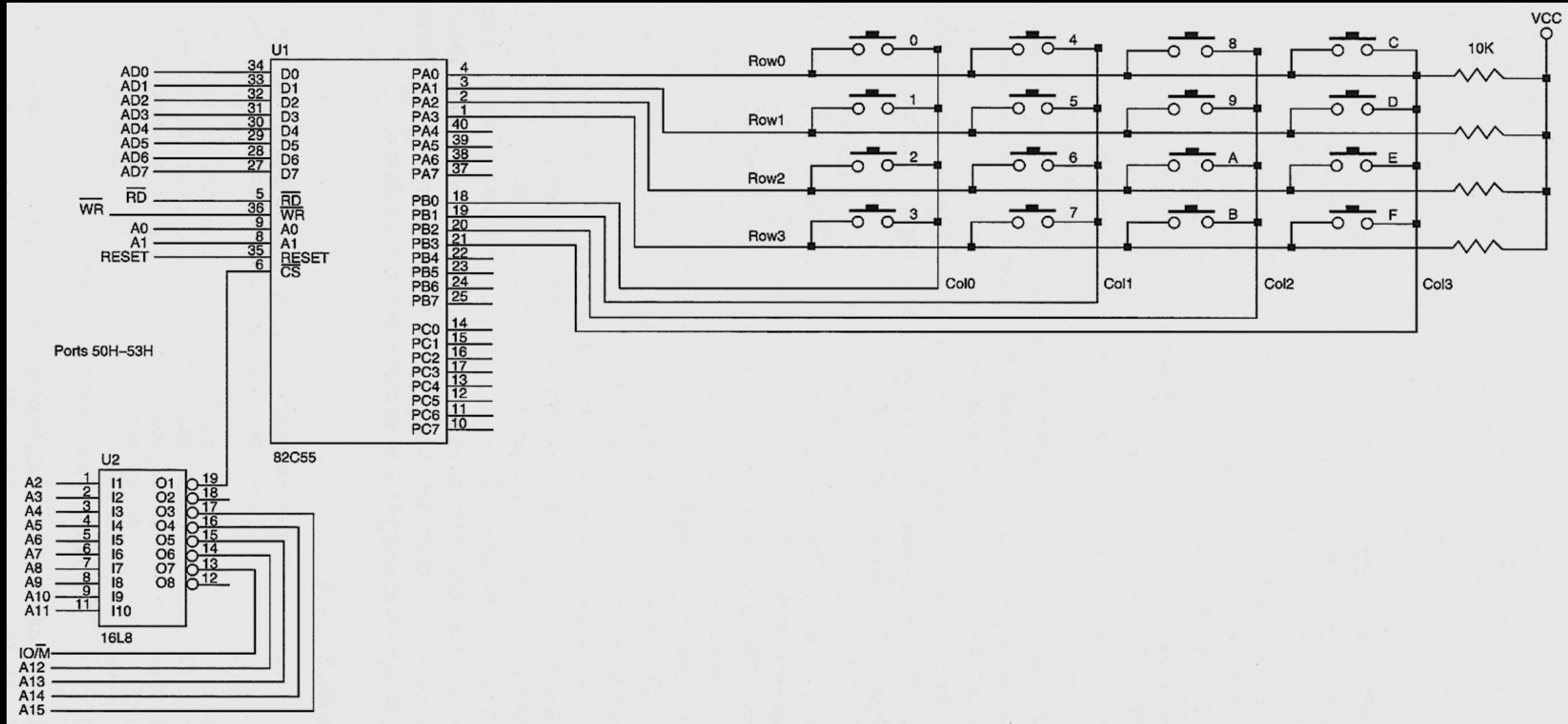
# 8 Digit LED Display Example



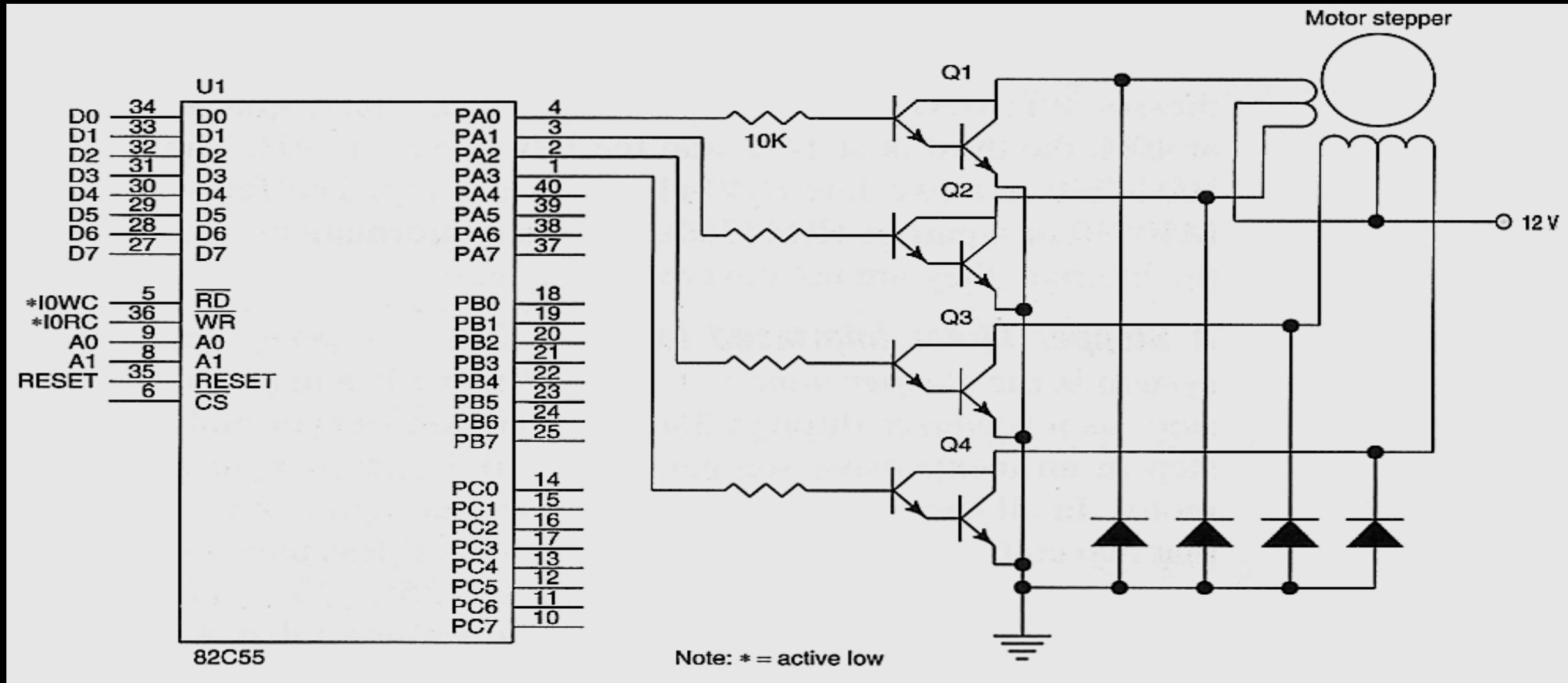
# LCD Display Example



# Keyboard Interface Example



# Stepper Motor Interface Example



**8257/8237**

**DMA Controller**

# Direct memory access

- Direct memory access (DMA) is a process in which an external device takes over the control of system bus from the CPU.
- DMA is for high-speed data transfer from/to mass storage peripherals, e.g. harddisk drive, magnetic tape, CD-ROM, and sometimes video controllers.
- The basic idea of DMA is to transfer blocks of data directly between memory and peripherals.
- The data don't go through the microprocessor but the data bus is occupied.

# Basic process of DMA – Minimum 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.

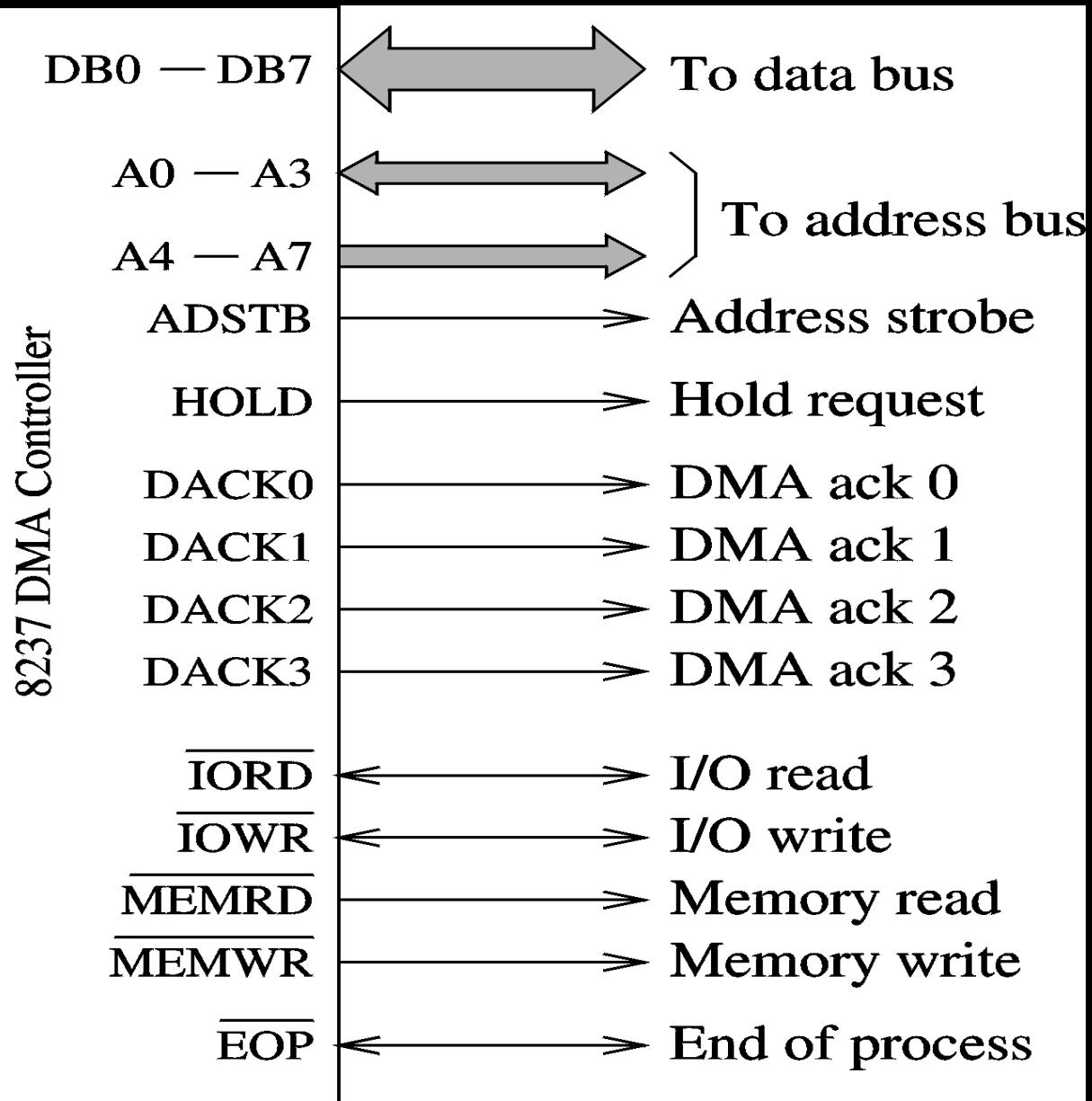
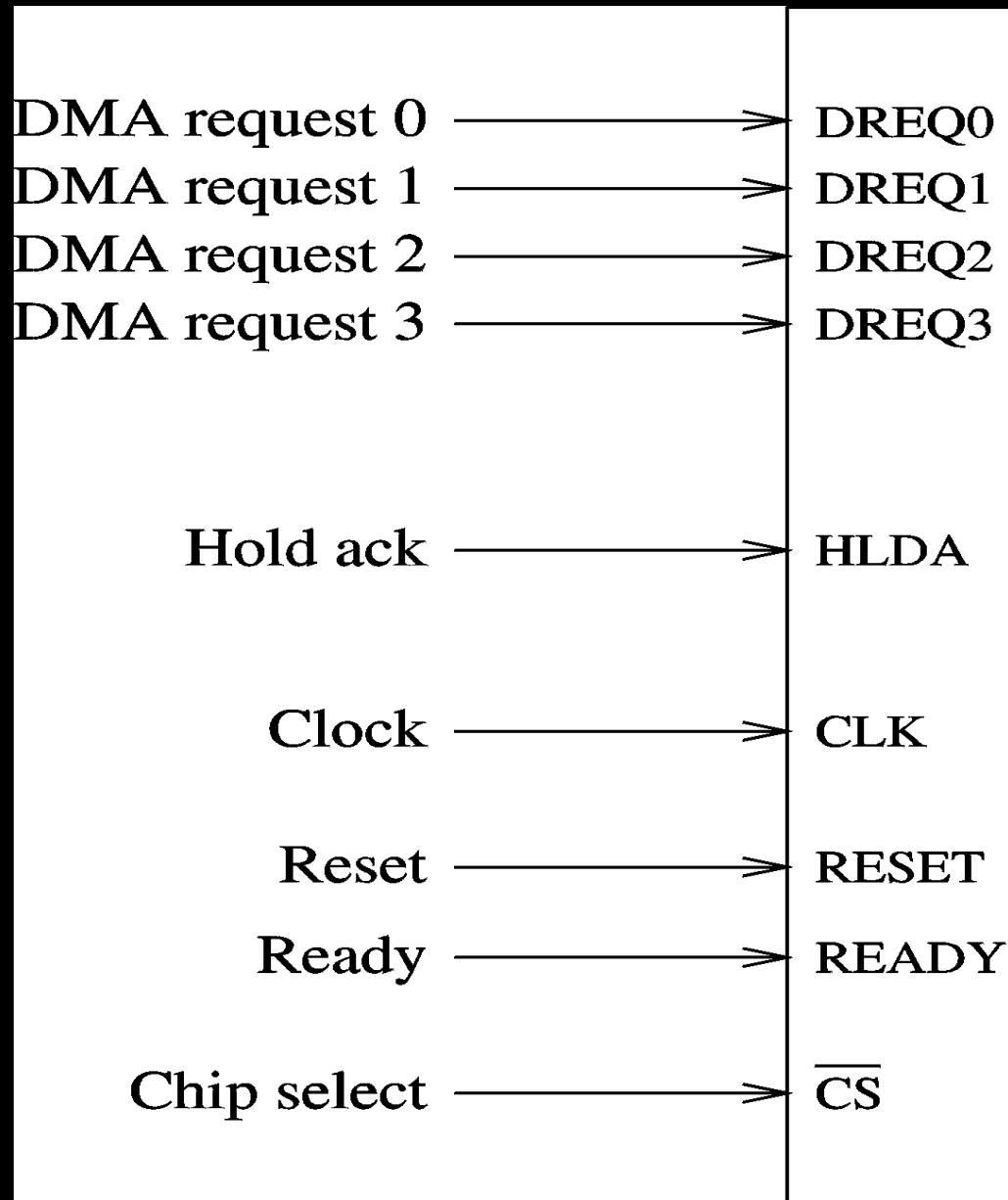
# Basic process of DMA – Maximum Mode

- 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.*

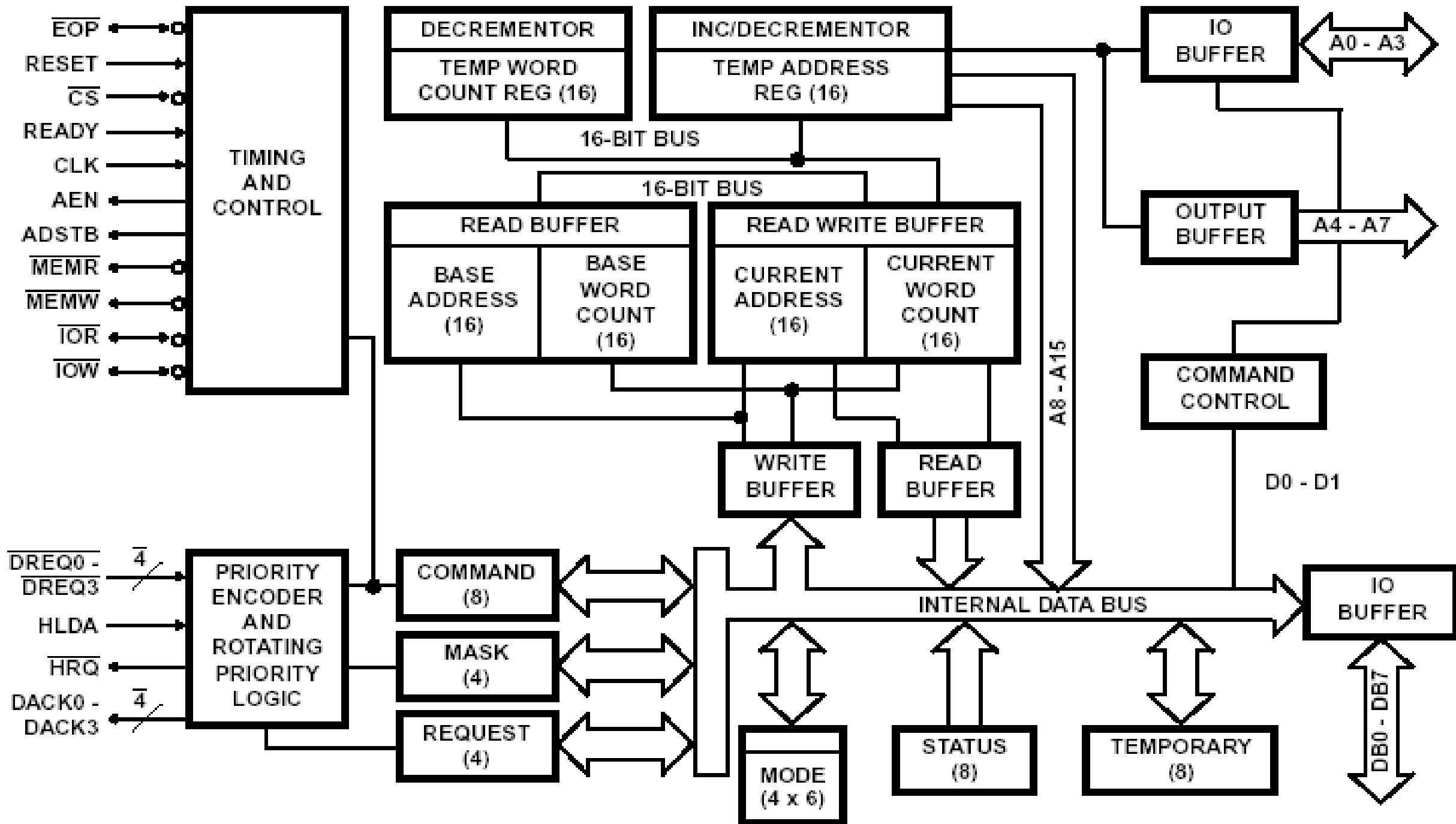
# DMA controller

- 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.

# 8237 - DMA Controller



# Block Diagram



# **8237 Registers**

- 1. Current address register**
- 2. Current word register**
- 3. Command register**
- 4. Mode register**
- 5. Request register**
- 6. Mask register**
- 7. Status register**
- 8. Temporary register**

# 8237 Registers

## 1. Current address register

- One 16-bit register for each channel
- Holds address for the current DMA transfer

## 2. Current word register

- Keeps the byte count
- Generates terminal count (TC) signal when the count goes from zero to FFFFH

## 3. Command register

- Used to program 8257

# 8237 Registers

## 4. Mode register

- Each channel can be programmed to
  - Read or write
  - Autoincrement or autodecrement the address
  - Autoinitialize the channel

## 5. Request register

- For software-initiated DMA

## 6. Mask register

- Used to disable a specific channel

## 7. Status register

## 8. Temporary register

- Used for memory-to-memory transfers

# Types of data transfer

- 8237 supports **four** types of data transfer

## 1. Single cycle transfer

- Only single transfer takes place
- Useful for slow devices

## 2. Block transfer mode

---

- Transfers data until TC is generated or external EOP signal is received

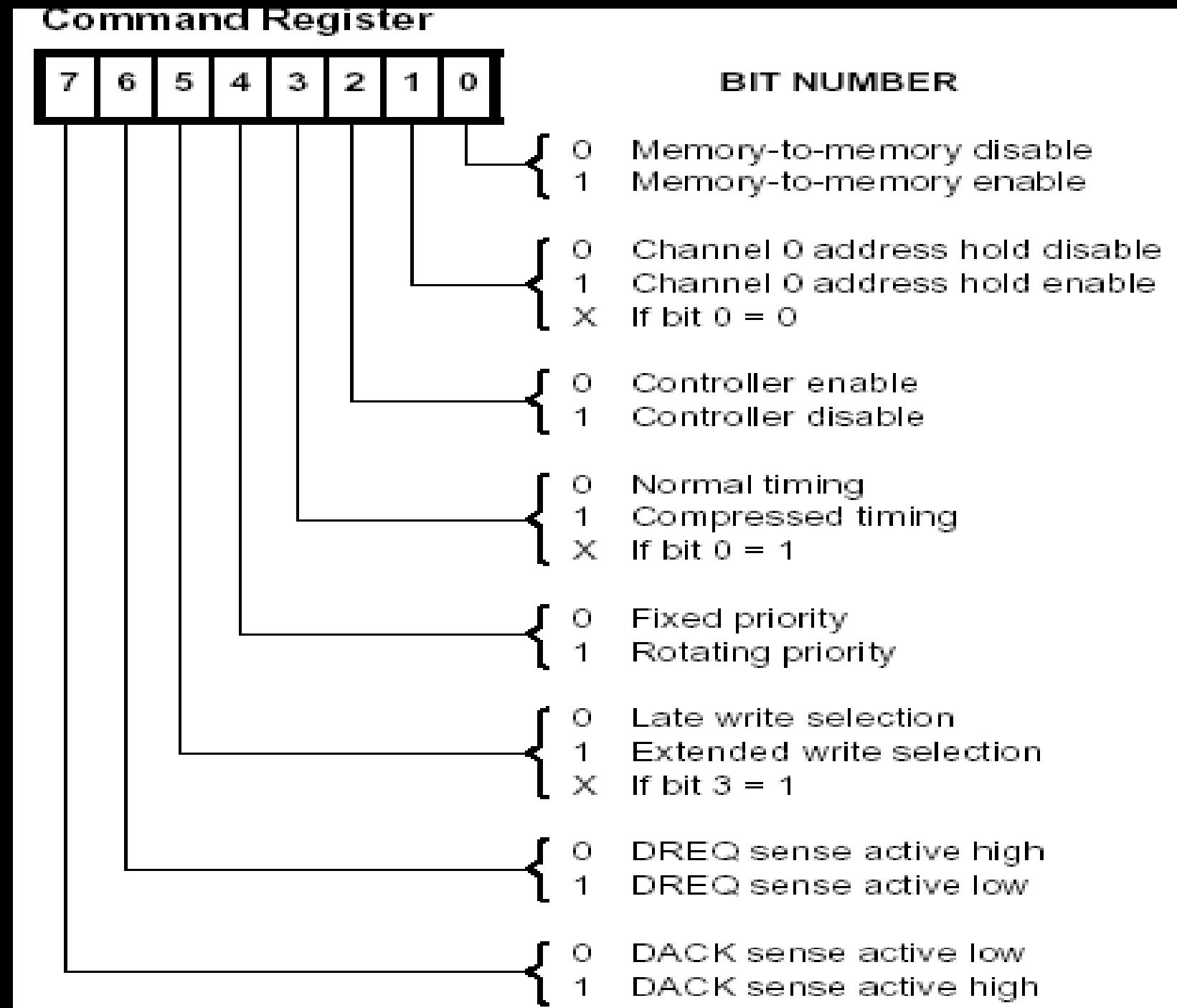
## 3. Demand transfer mode

- Similar to the block transfer mode
- In addition to TC and EOP, transfer can be terminated by deactivating DREQ signal

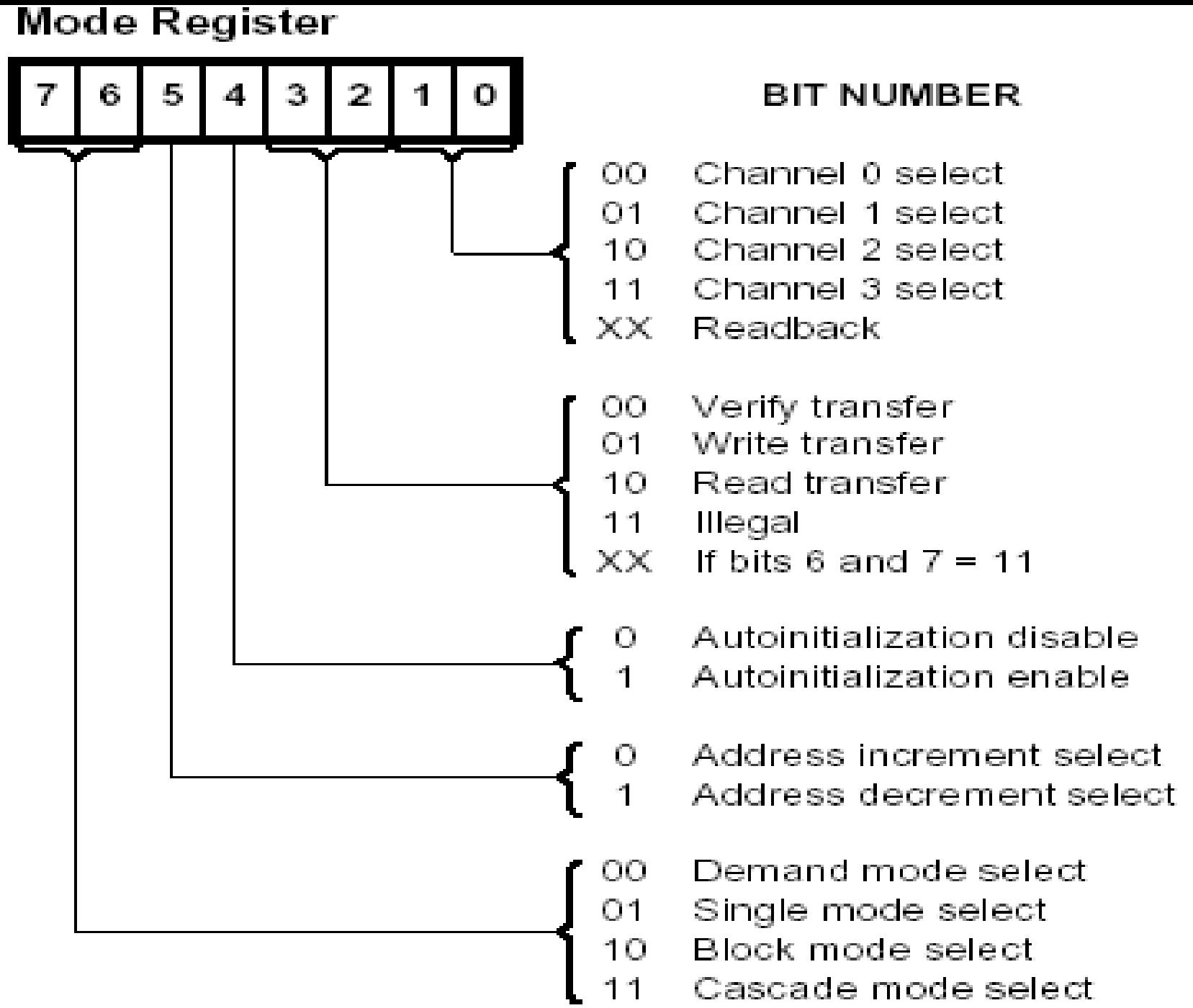
## 4. Cascade mode

- Useful to expand the number channels beyond four

# Command Register

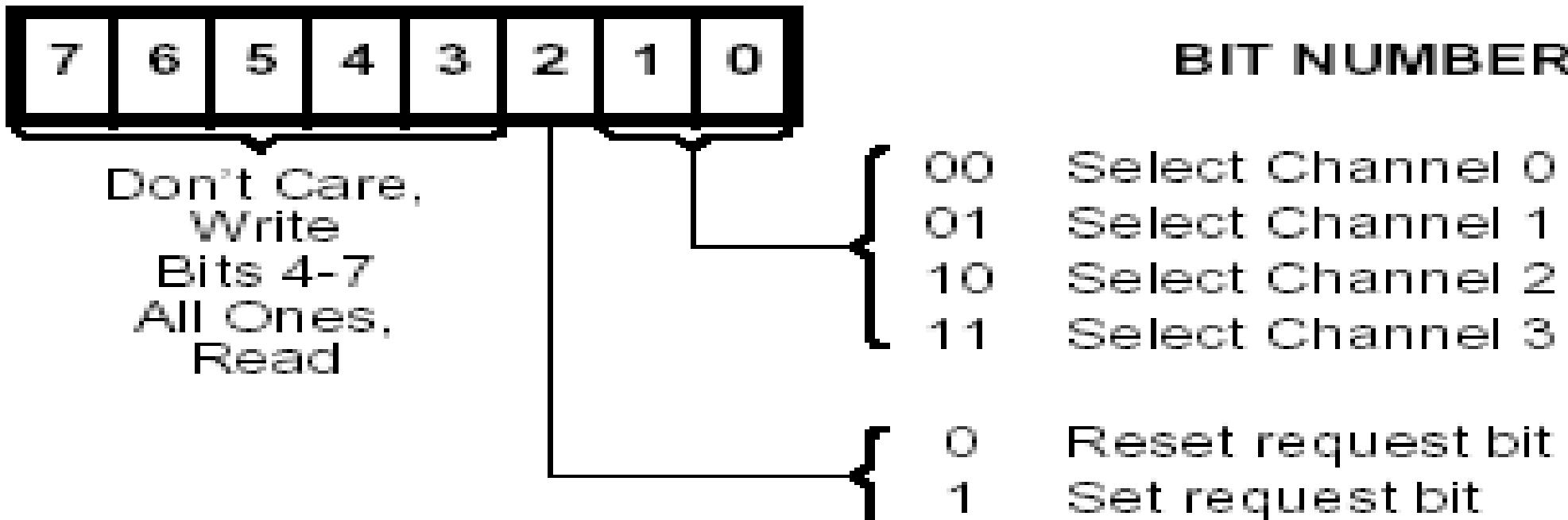


# Mode Register

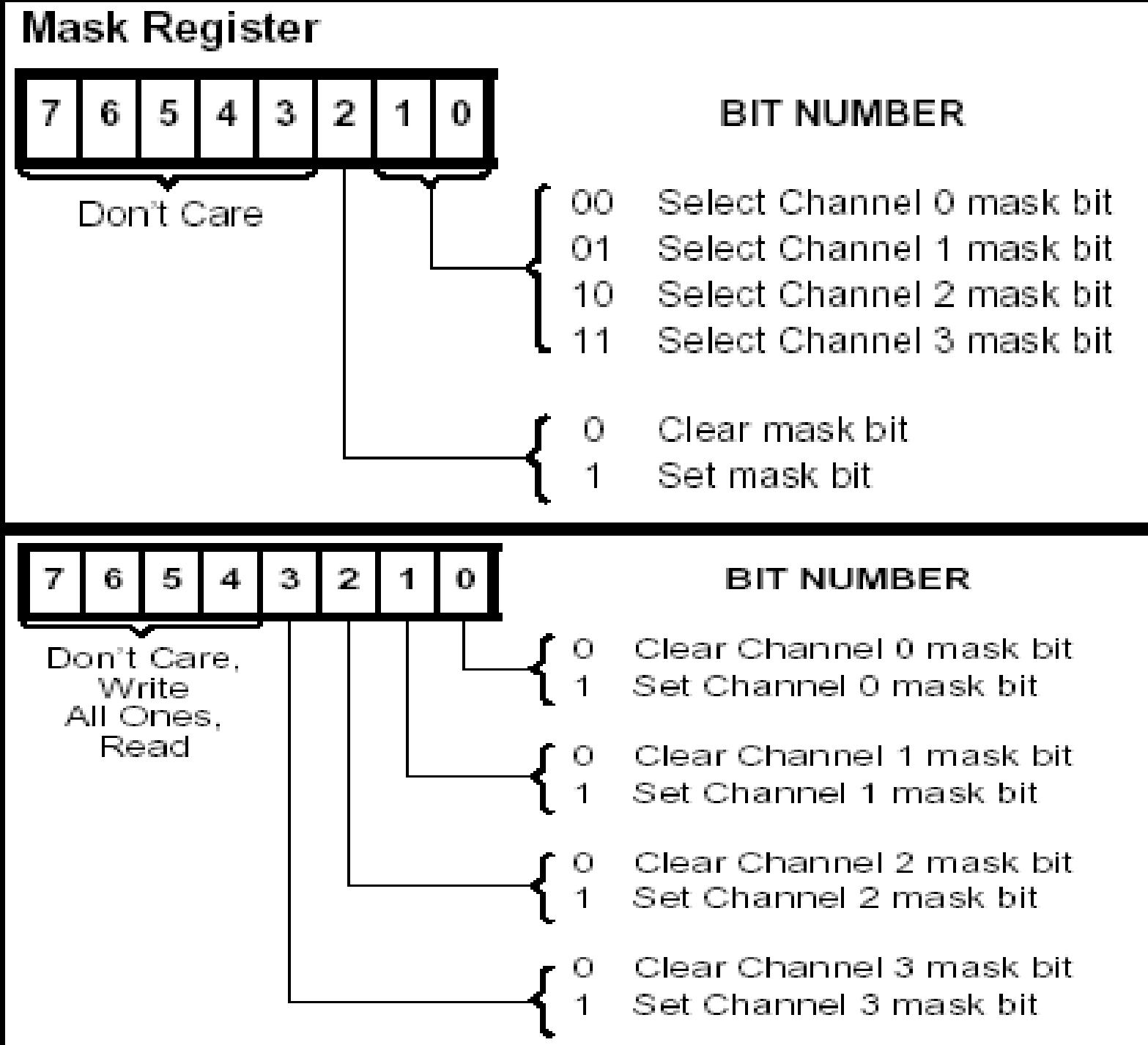


# Request Register

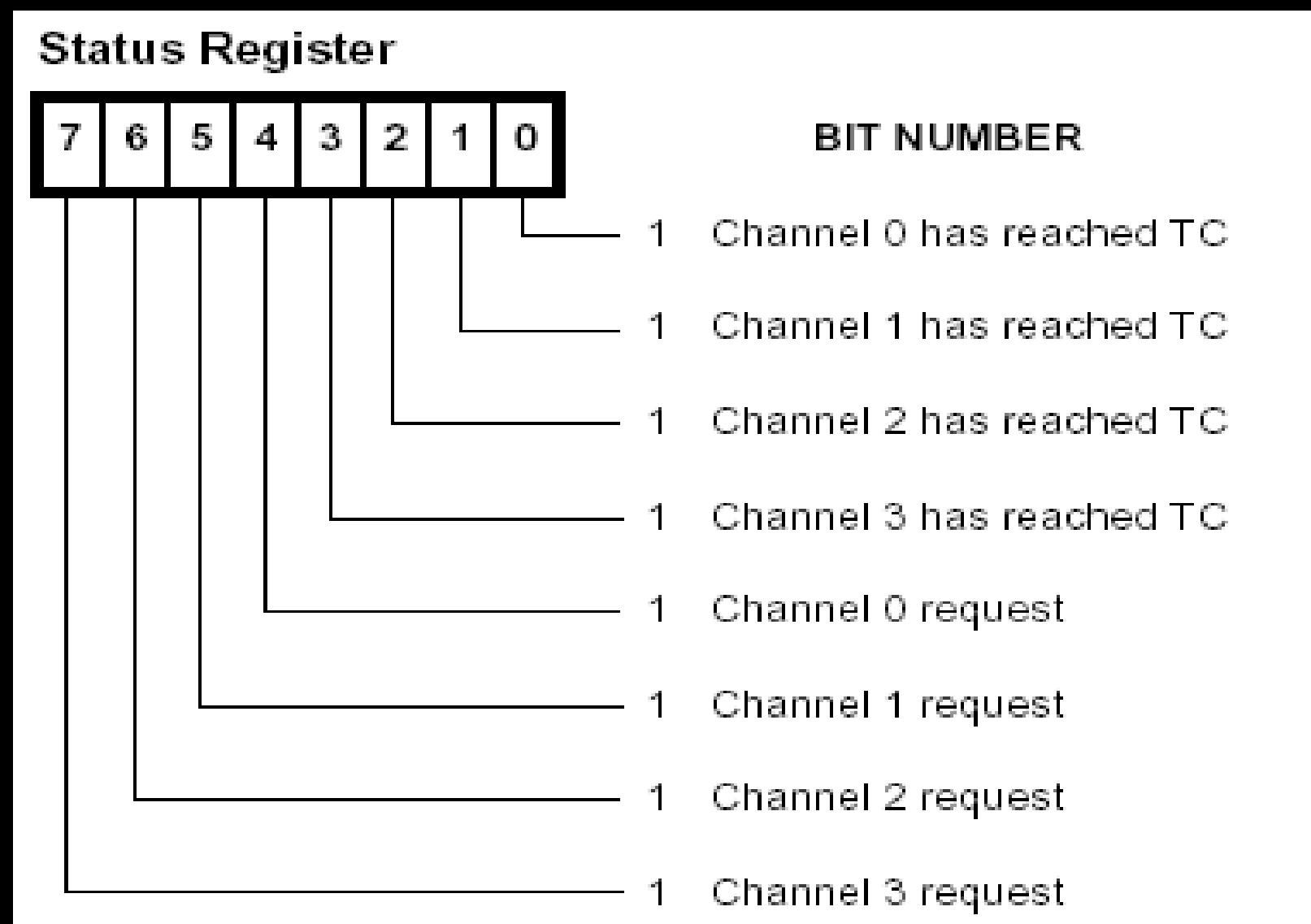
Request Register



# Mask Register



# Status Register



**82599**

**Programmable Interrupt Controller**

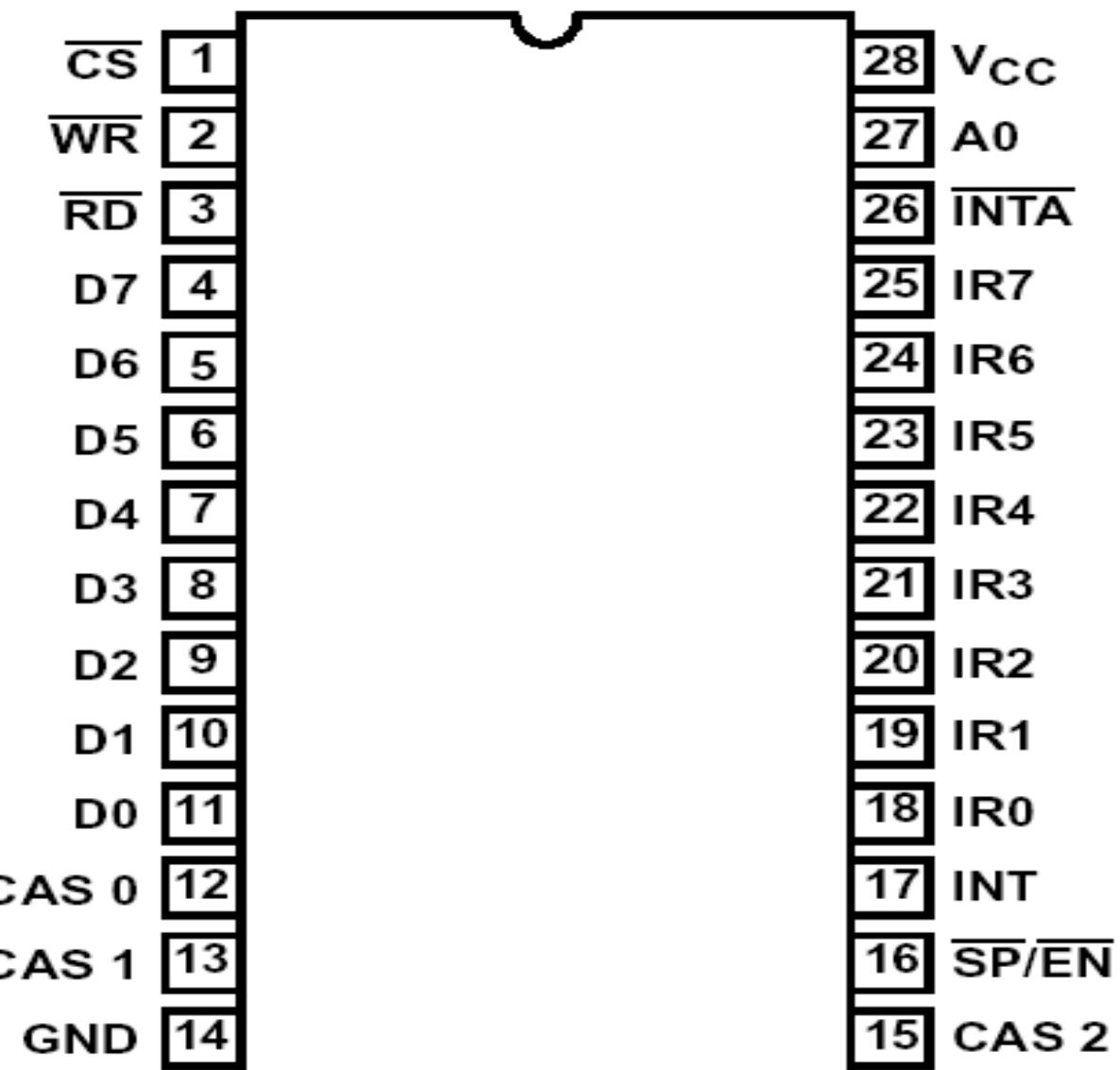
# 8259 Features

- 8259 is Programmable Interrupt Controller (PIC)
- It is a tool for managing the interrupt requests.
- 8259 is a very flexible peripheral controller chip:
  - PIC can deal with up to 64 interrupt inputs
  - interrupts can be masked individually.
  - various priority schemes can also programmed.

# 8259 Pin Diagram

## **82C59A (PDIP, CERDIP)**

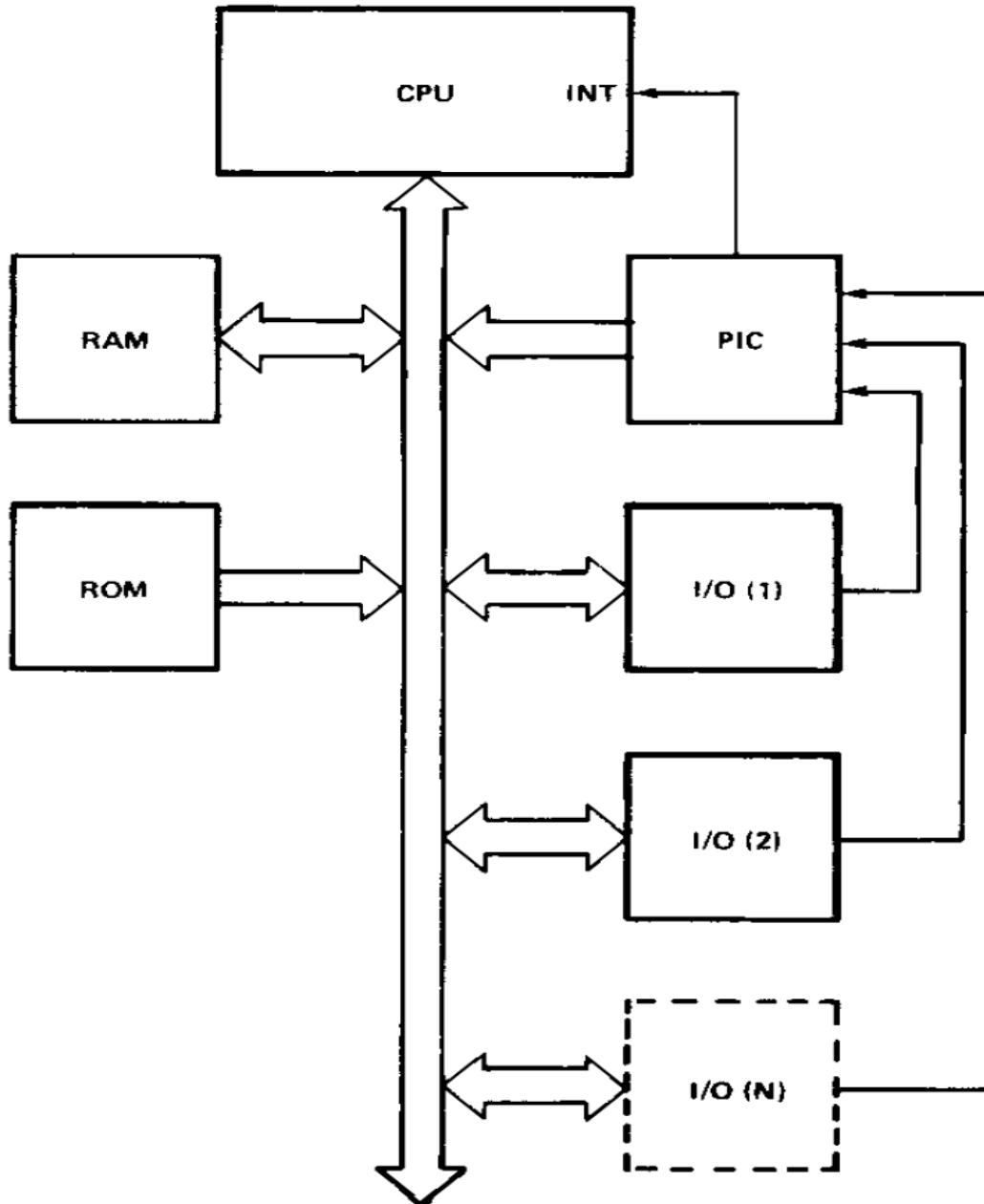
TOP VIEW



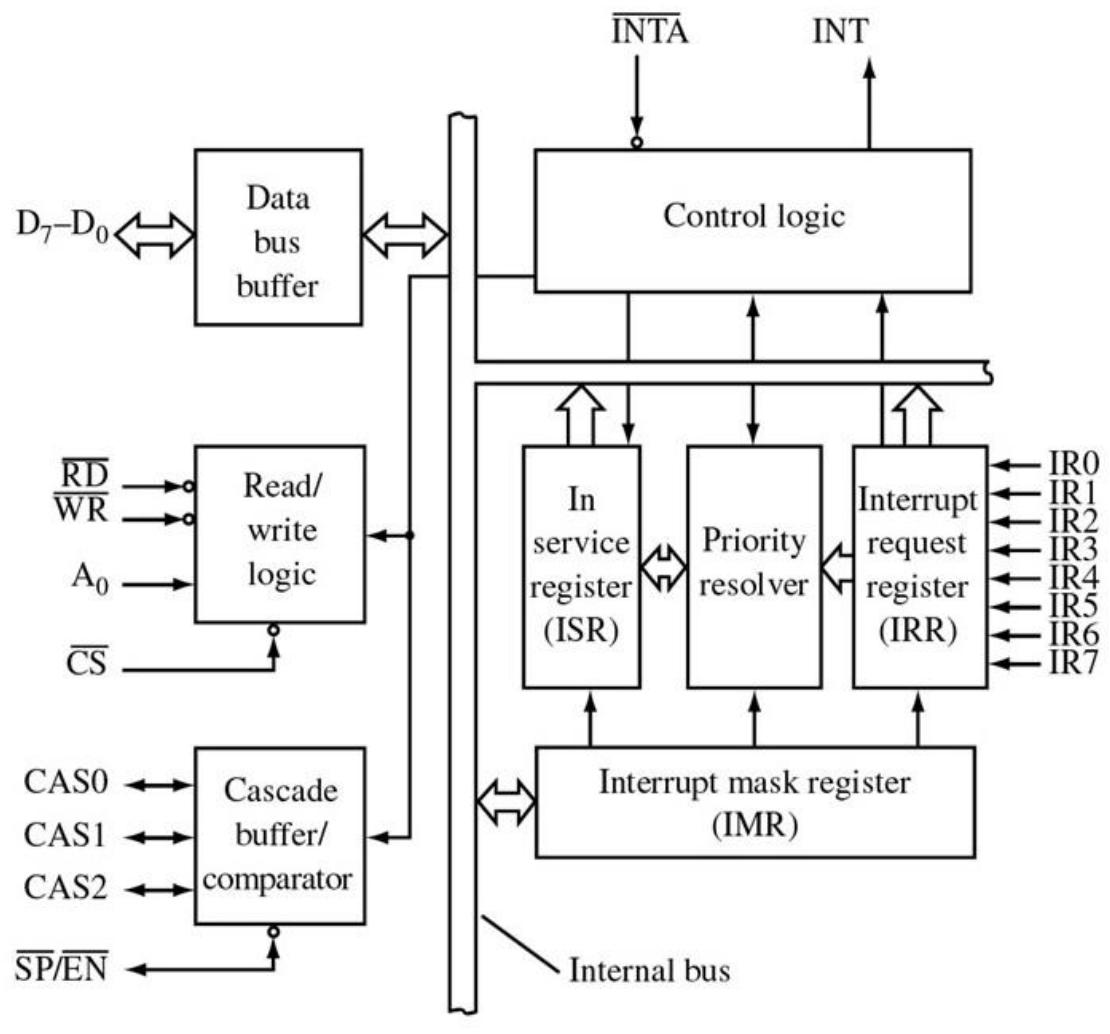
# Pin Details

Symbol	Pin No.	Type	Name and Function
SP/EN	16	I/O	<b>SLAVE PROGRAM/ENABLE BUFFER:</b> This is a dual function pin. When in the Buffered Mode it can be used as an output to control buffer transceivers (EN). When not in the buffered mode it is used as an input to designate a master (SP = 1) or slave (SP = 0).
INT	17	O	<b>INTERRUPT:</b> This pin goes high whenever a valid interrupt request is asserted. It is used to interrupt the CPU, thus it is connected to the CPU's interrupt pin.
IR <sub>0</sub> –IR <sub>7</sub>	18–25	I	<b>INTERRUPT REQUESTS:</b> Asynchronous inputs. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
INTA	26	I	<b>INTERRUPT ACKNOWLEDGE:</b> This pin is used to enable 8259A interrupt-vector data onto the data bus by a sequence of interrupt acknowledge pulses issued by the CPU.
A <sub>0</sub>	27	I	<b>AO ADDRESS LINE:</b> This pin acts in conjunction with the CS, WR, and RD pins. It is used by the 8259A to decipher various Command Words the CPU writes and status the CPU wishes to read. It is typically connected to the CPU A0 address line (A1 for 8086, 8088).

# Interfacing 8259



# Block Diagram



# Working of 8259

1. One or more of the **INTERRUPT REQUEST** lines (IR0 – IR7) are raised high, setting the corresponding IRR bit(s).
2. The 8259A evaluates these requests, and sends an **INT** to the **CPU**, if appropriate.
3. The CPU acknowledges the INT and responds with an **INTA\*** pulse.
4. Upon receiving an **INTA\*** from the CPU group, the **highest priority ISR bit is set and the corresponding IRR bit is reset**.

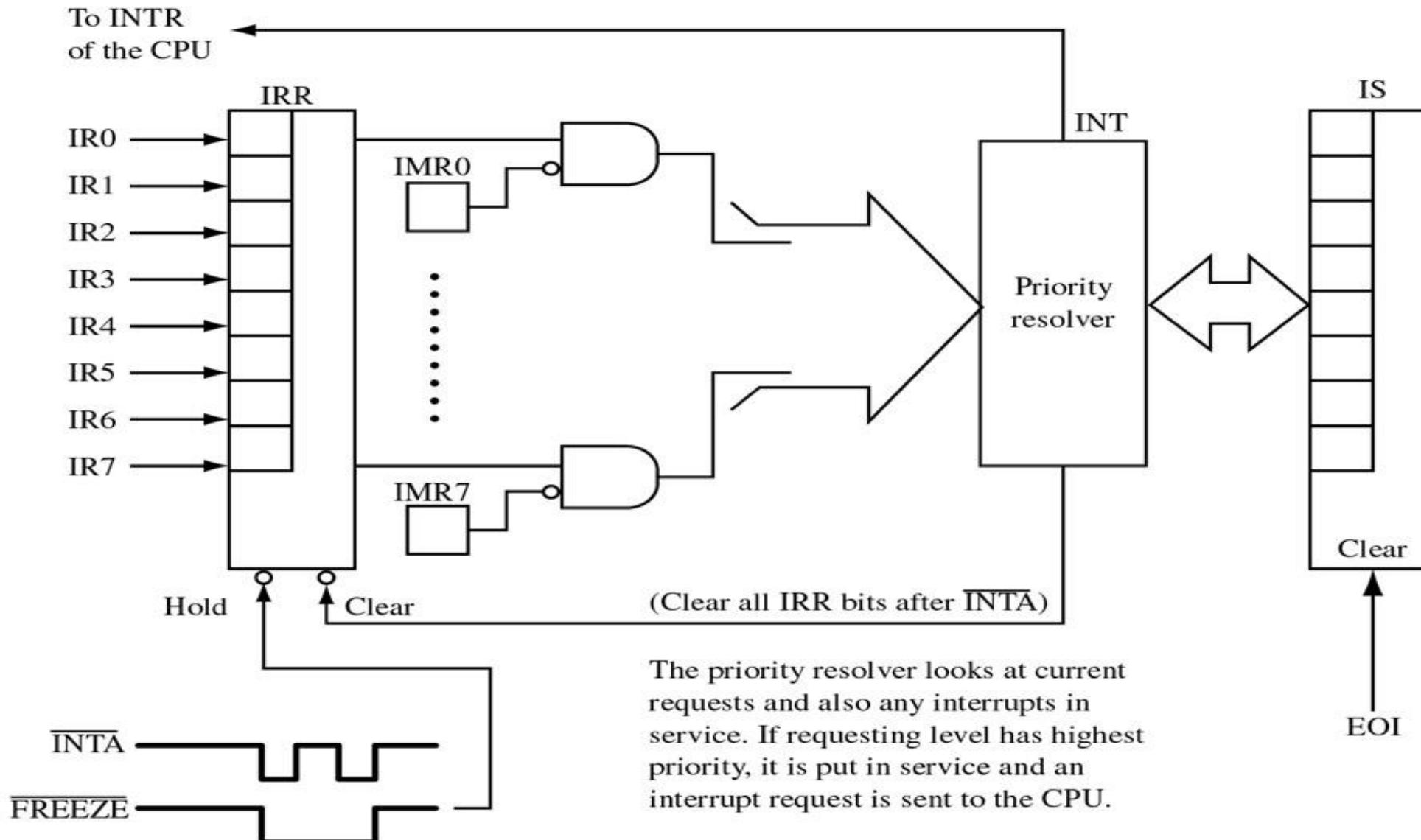
# Working of 8259

5. Then 8086 will send **one more INTA pulse** to 8259.
  - On this second interrupt acknowledge cycle, 8259 will send an interrupt vector byte of data to the CPU, which is a pointer of the interrupt to be processed.
5. This **completes** the interrupt cycle.
6. The **ISR bit is reset** at the end of the **3<sup>rd</sup> INTA pulse**.

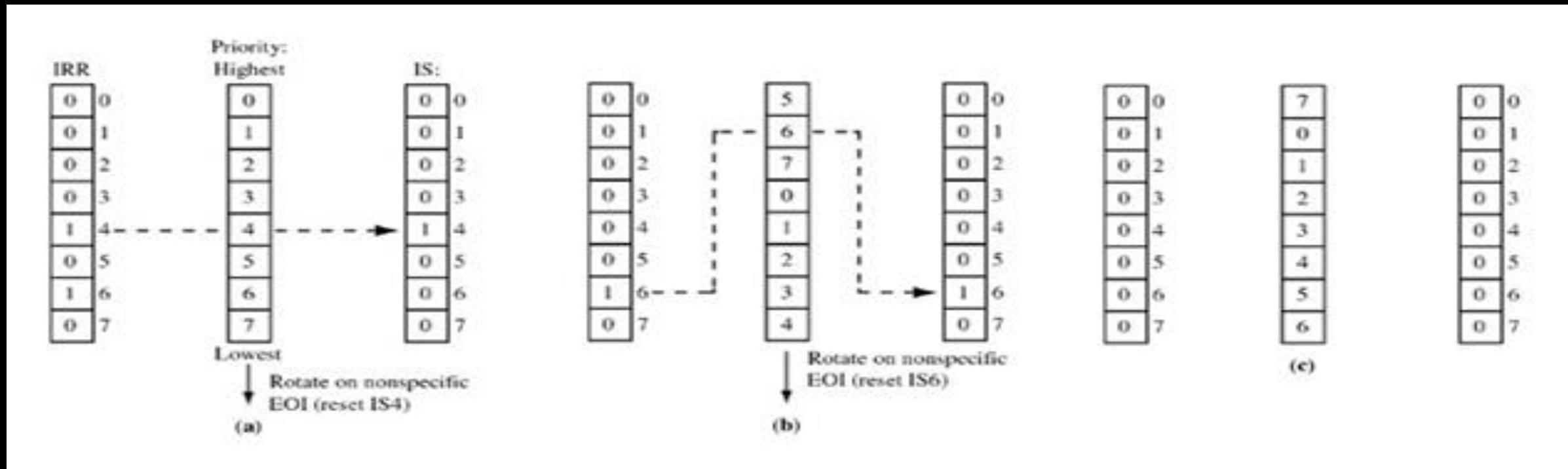
# Interrupt vector byte

	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

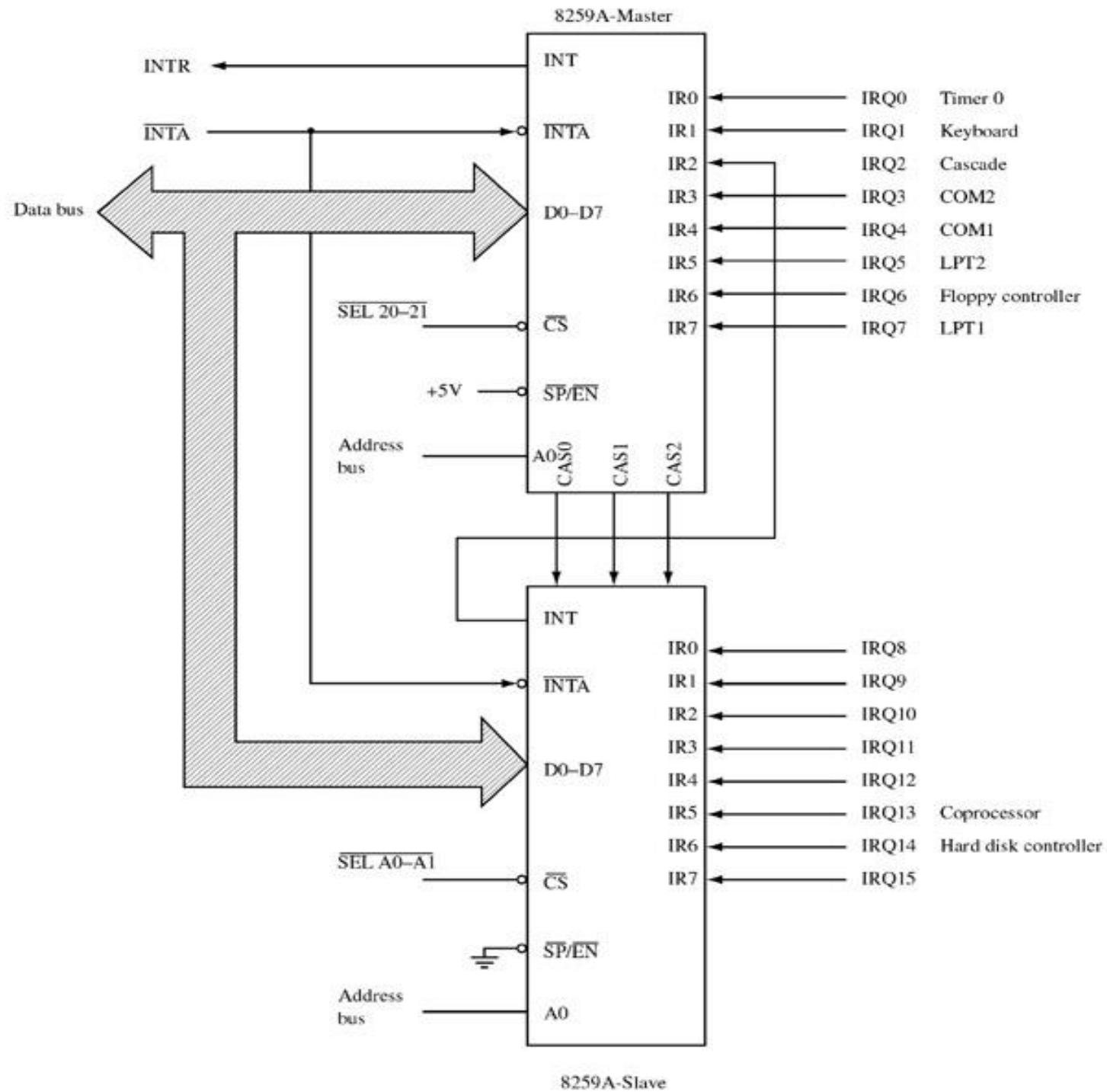
# Work flow inside 8259



# 8259 Priority Resolver



- Simultaneous interrupt requests arrive on IR4 and IR6. IR4 has highest priority and its IS bit is set as the IR4 service routine is put in service.
- The IR4 service routine issues a rotate-on-nonspecific-EOI command, resetting IS4 and assigning it lowest priority. IR6 is now placed in service.
- The IR6 service routine issues a rotate-on-nonspecific-EOI command, resetting IS6 and assigning it lowest priority.



# 8259 Command Words

- There are 2 Command Words in 8259.

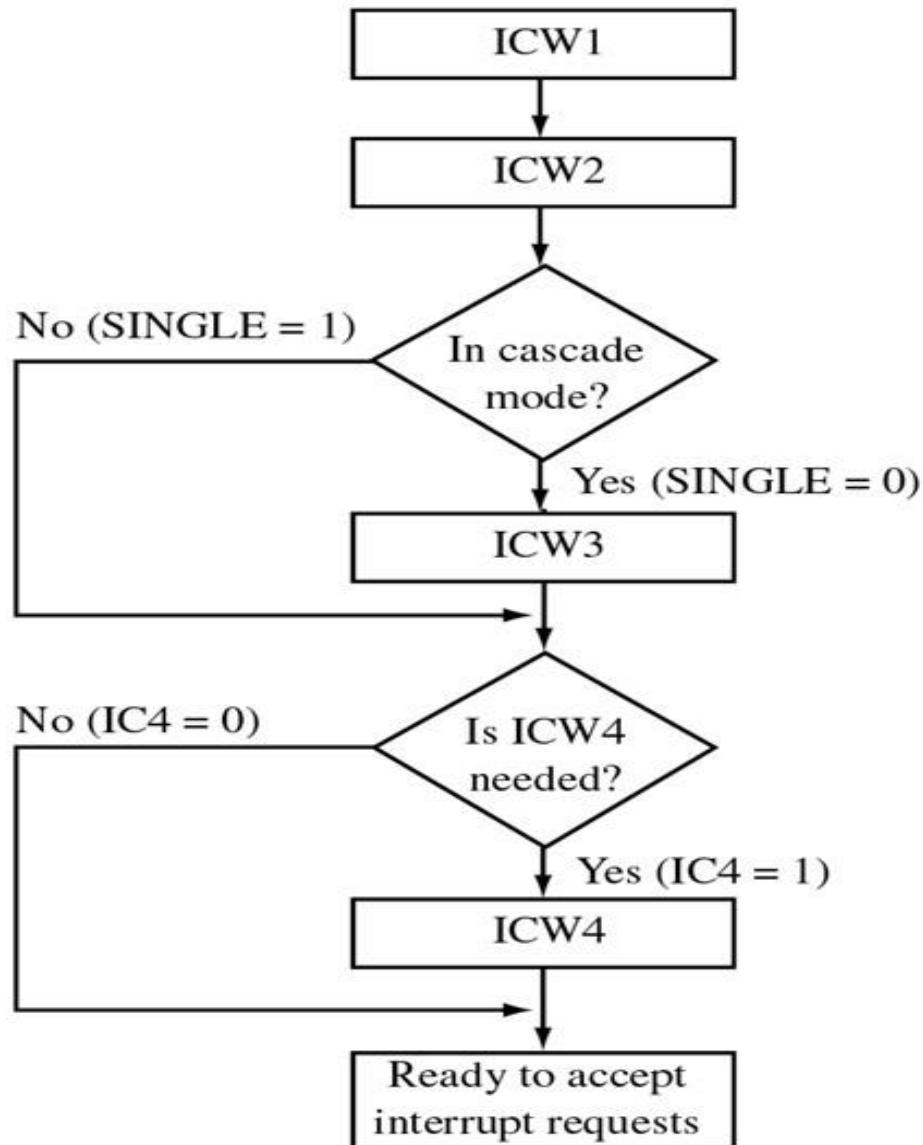
**1. Initialization Command Words (ICWs):** Before normal operation can begin, each 82C59A in the system must be brought to a starting point using these command words.

- There are 4 ICWs in 8259.

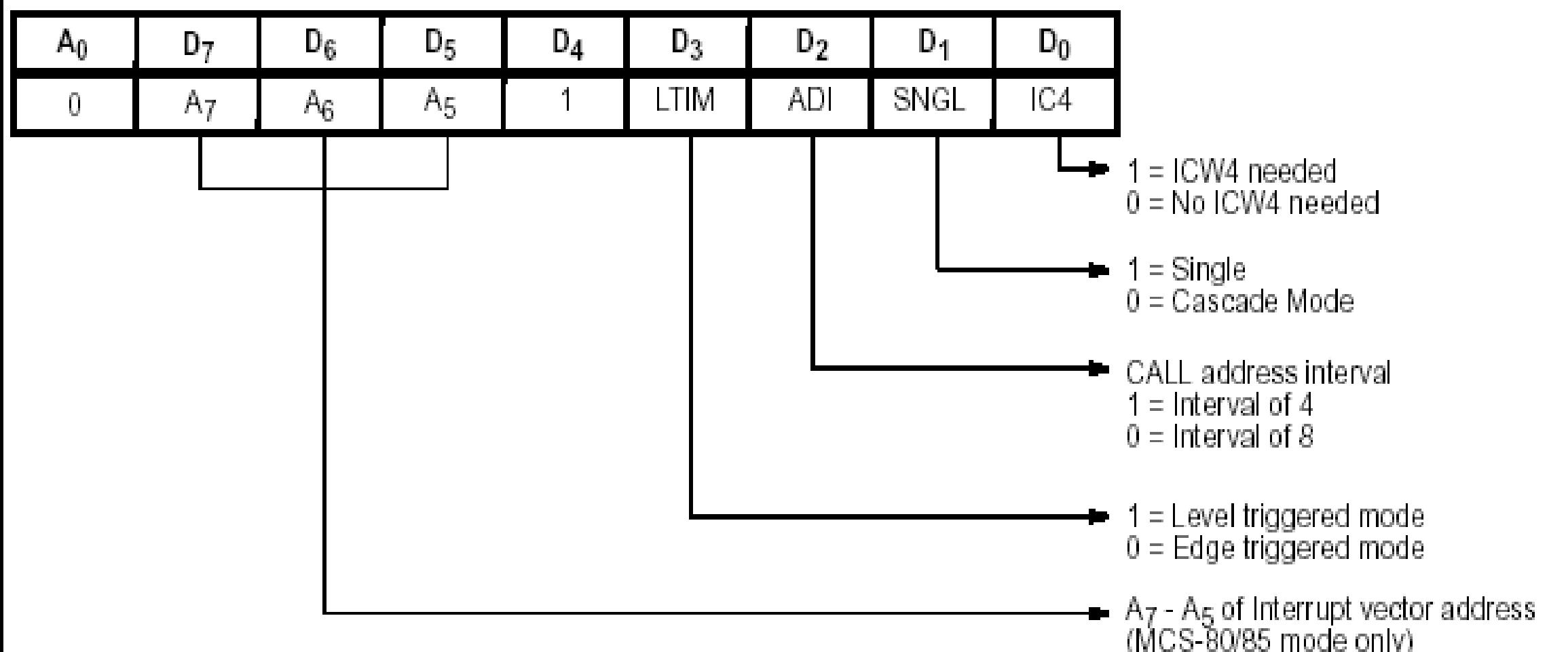
**2. Operation Command Words (OCWs):** These are the command words which command the 82C59A to operate in various interrupt modes.

- There are 3 OCWs in 8259

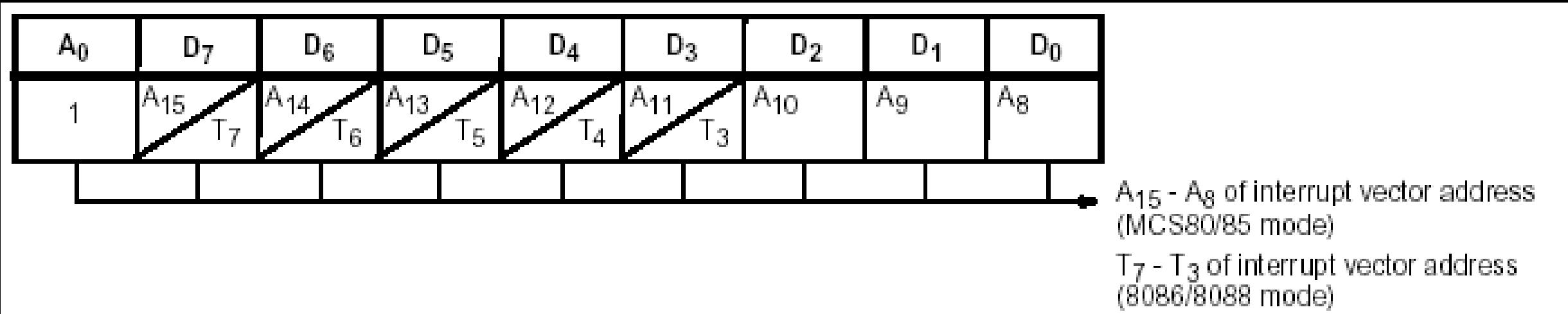
# 8259A initialization sequence



# ICW1 Format



# ICW2 Format



	D7	D6	D5	D4	D3	D2	D1	D0
IR7	T7	T6	T5	T4	T3	1	1	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR0	T7	T6	T5	T4	T3	0	0	0

# ICW3 Format

- This word is read only when there is more than one 8259 in the system and cascading is used, in which case SNGL = 0 in ICW1.

**ICW3 (Master device)**

A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	S <sub>7</sub>	S <sub>6</sub>	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>



This register is treated as a mask, with 1's indicating the IRQ channels connected to master/slave 8259As.

0 = IR Input has a slave  
1 = IR Input does not have a slave

**ICW3 (SLAVE DEVICE)**

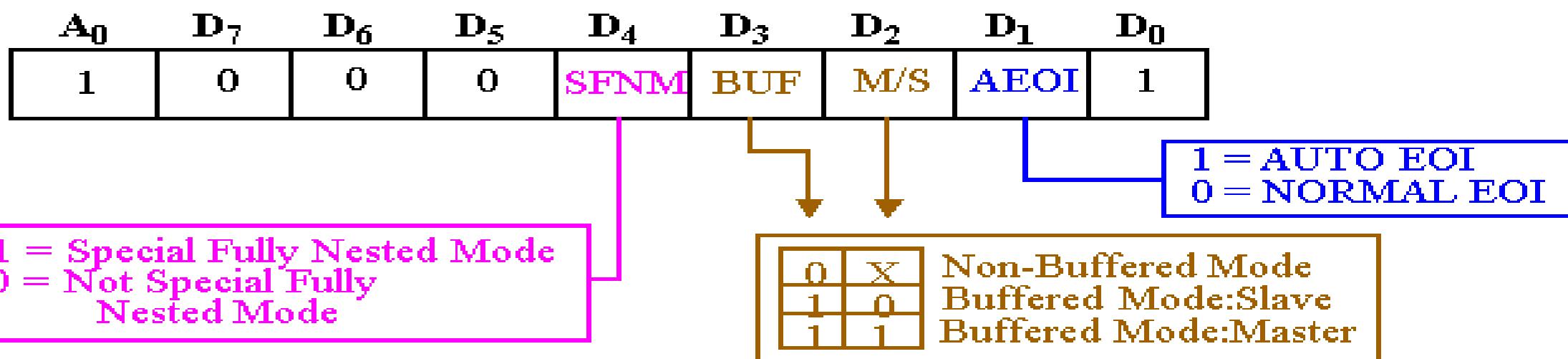
A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
1	0	0	0	0	0	ID <sub>2</sub>	ID <sub>1</sub>	ID <sub>0</sub>

**SLAVE ID (NOTE)**

0	1	2	3	4	5	6	7
0	1	0	1	0	1	0	1
0	0	1	1	0	0	1	1
0	0	0	0	1	1	1	1

# ICW4 Format

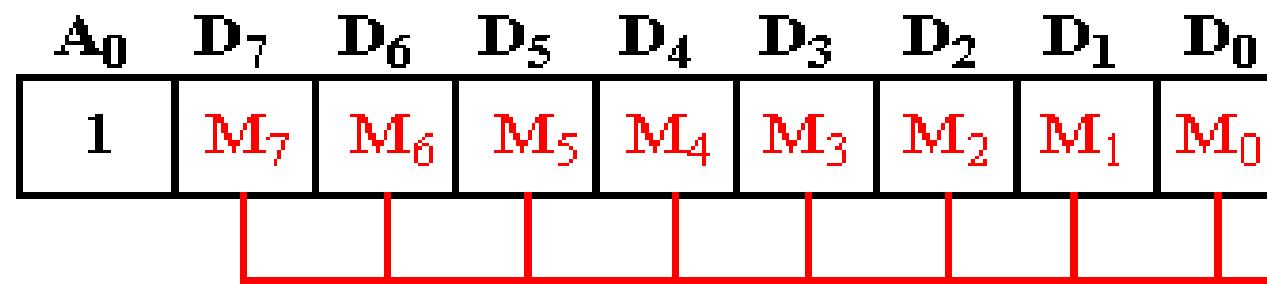
- SFNM: If SFNM = 1, the special fully nested mode is programmed.
- BUF: If BUF = 1, the buffered mode is programmed. In buffered mode,  $\overline{SP}/\overline{EN}$  becomes an enable output and the master/slave determination is by M/S.
- M/S: If buffered mode is selected: M/S = 1 means the 82C59A is programmed to be a master, M/S = 0 means the 82C59A is programmed to be a slave. If BUF = 0, M/S has no function.
- AEOI: If AEOI = 1, the automatic end of interrupt mode is programmed.
- $\mu$ PM: Microprocessor mode:  $\mu$ PM = 0 sets the 82C59A for 8080/85 system operation,  $\mu$ PM = 1 sets the 82C59A for 80C86/88/286 system operation.



# Operation Command Words (OCWs)

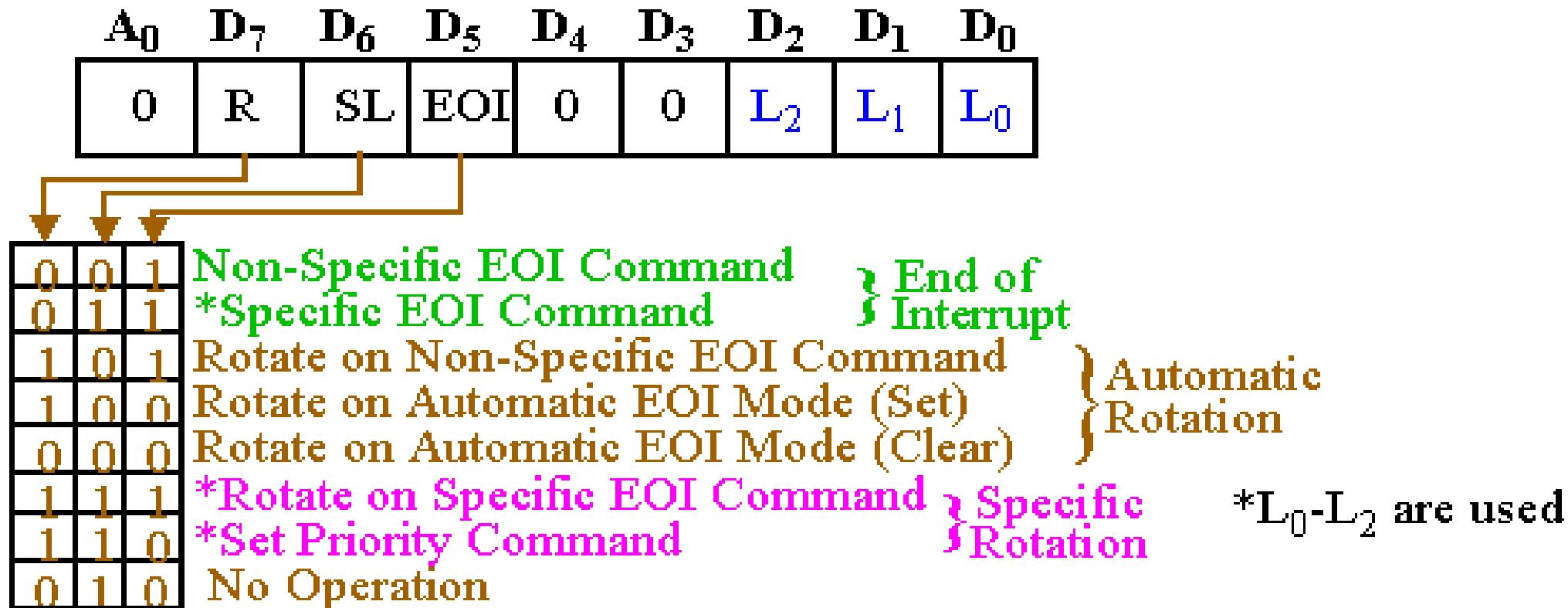
- After the **Initialization Command Words (ICWs)** are programmed into the 8259A, the chip is ready to accept interrupt requests at its input lines.
- However, during the 8259A operation, a selection of algorithms can command the **8259A to operate in various modes** through the Operation Command Words (OCWs).

# OCW1 Format



**INTERRUPT MASK**  
0 = Mask Reset  
1 = Mask Set

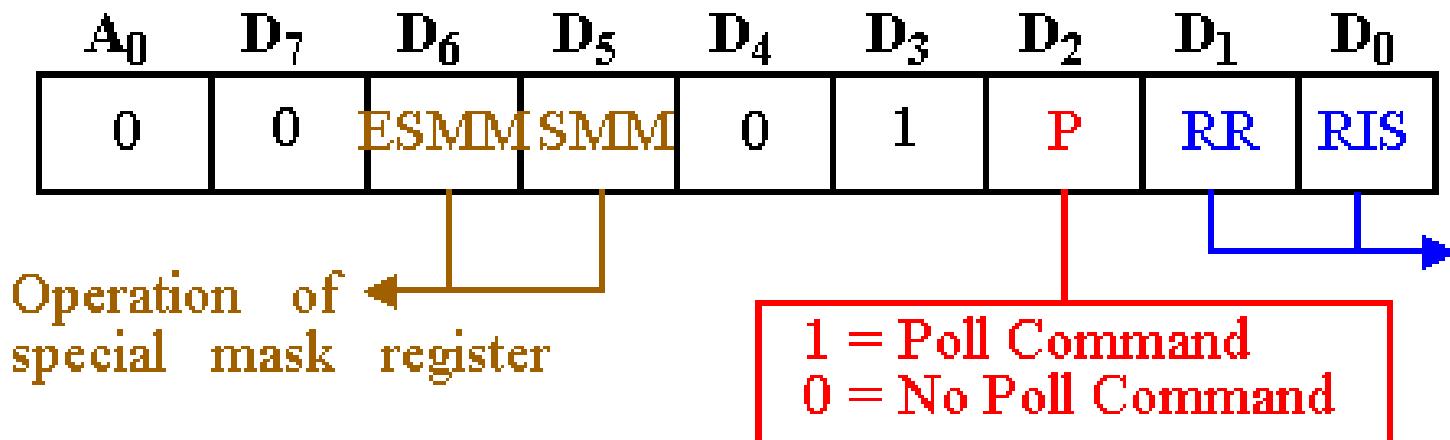
# OCW2 Format



**R, SL, EOI:** These three bits control the Rotate and End of Interrupt modes and combinations of the two.

**L2, L1, L0:** These bits determine the interrupt level acted upon when the SL bit is active.

# OCW3 Format



Indicates which status register, IRR or ISR, is to be read.

**ESMM** - Enable Special Mask Mode. When this bit is set to 1 it enables the SMM bit to set or reset the Special Mask Mode. When ESMM = 0, the SMM bit becomes a “don’t care”.

**SMM** - Special Mask Mode. If ESMM = 1 and SMM = 1, the 82C59A will enter Special Mask Mode. If ESMM = 1 and SMM = 0, the 82C59A will revert to normal mask mode. When ESMM = 0, SMM has no effect.

# 8259 Working Modes

- There are 4 different modes for 8259.
  1. Fully nested mode.
  2. Rotating priority mode.
  3. Special mask mode.
  4. Polled mode.

# Fully nested mode

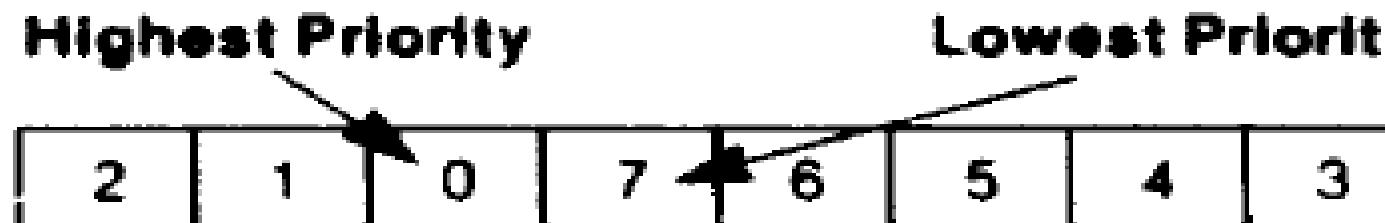
- This mode is entered after initialization unless another mode is programmed.
- The interrupt requests are ordered in priority from 0 through 7 (0 highest).
- When an interrupt is acknowledged the highest priority request is determined and its vector placed on the bus.
- Additionally, a bit of the Interrupt Service register (ISO-7) is set.
- This bit remains set until the microprocessor issues an End of Interrupt (EOI) command immediately before returning from the service routine
- If AEOI (Automatic End of Interrupt) bit is set, until the trailing edge of the last INTA.

**After Rotate** (IR4 was serviced, all other priorities rotated correspondingly)

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
0	1	0	0	0	0	0	0

“IS” Status

231468–20



Priority Status

231468–21

# Special mask mode

- Some applications may require an interrupt service routine to dynamically alter the system priority structure during its execution under software control.
- For example, the routine may wish to inhibit lower priority requests for a portion of its execution but enable some of them for another portion.
- That is where the Special Mask Mode comes in.
- In the special Mask Mode, when a mask bit is set in OCW1, it inhibits further interrupts at that level and enables interrupts from all other levels (lower as well as higher) that are not masked.
- Thus, any interrupts may be selectively enabled by loading the mask register.

# Polled mode

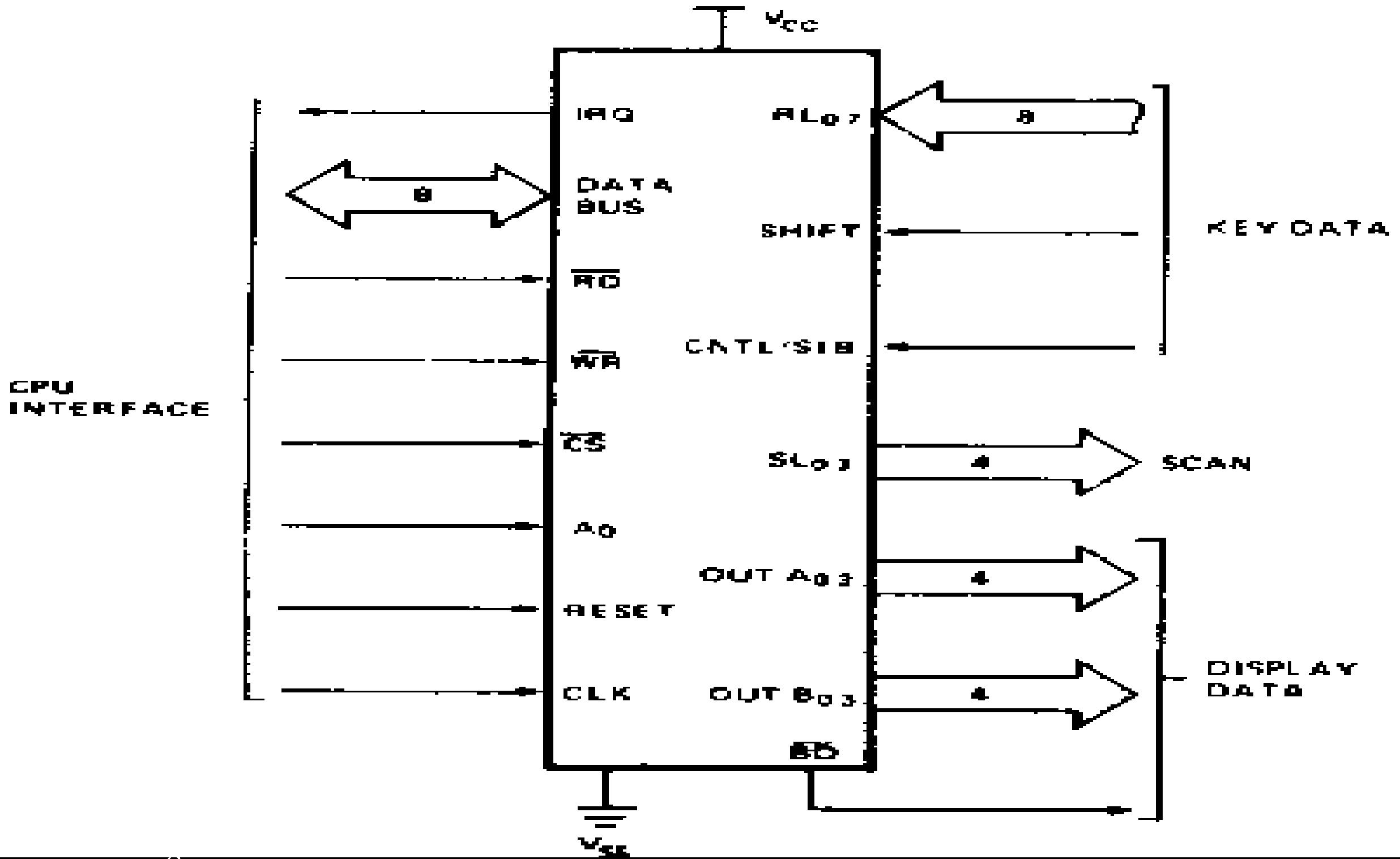
- In Polled mode the INT output functions as it normally does.
- The microprocessor should ignore this output.
- This can be accomplished either by not connecting the INT output or by masking interrupts within the microprocessor, thereby disabling its interrupt input.
- Service to devices is achieved by software using a Poll command.
- The Poll command is issued by setting P = 1 in OCW3.

82799

Keyboard Display Interface

# Features of 8279

- Scans and encodes up to a 64-key keyboard.
- Controls up to a 16-digit numerical display.
- Keyboard has a built-in FIFO 8 character buffer to store the keyboard Entries and an Interrupt signal with each Entry.
- The display is controlled from an internal 16x8 RAM that stores the coded display information.



# Pin Details

- **A0** : Selects data (0) or control/status (1) for reads and writes between microprocessor and 8279.
- **BD** : Output that blanks the displays.
- **CLK** : Used internally for timing. Maximum is 3 MHz
- **CN/ST** : Control/strobe, connected to the control key on the keyboard.
- **CS** : Chip select that enables programming, reading the keyboard, etc.
- **DB7-DB0** : Consists of bidirectional pins that connect to data bus on micro.

# Pin Details

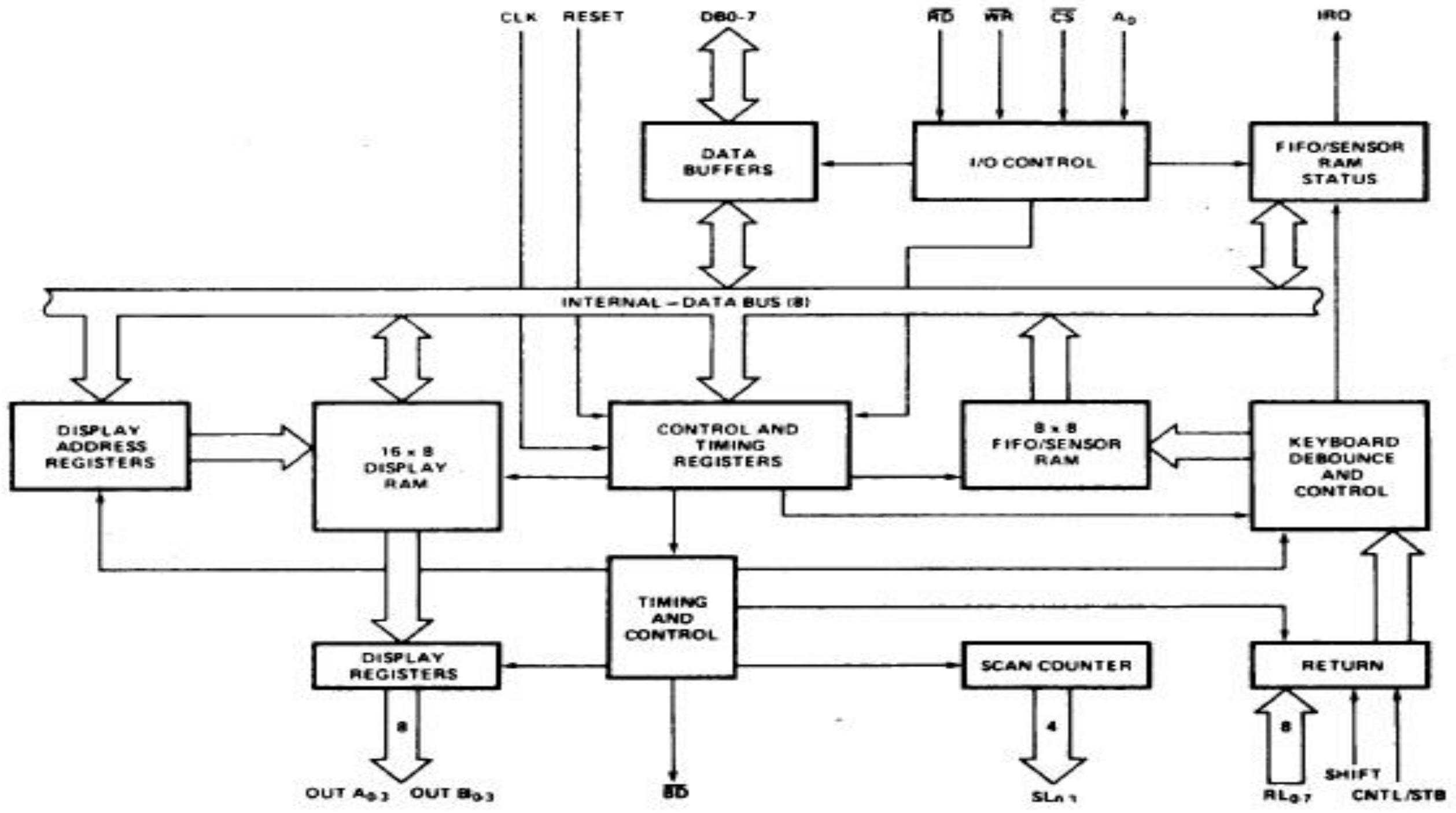
- **IRQ** : Interrupt request, becomes 1 when a key is pressed, data is available.
- **OUT A3-A0/B3-B0** : Outputs that sends data to the most significant/least significant nibble of display.
- **RD(WR)** : Connects to micro's IORC or RD signal, reads data/status registers.
- **RESET** : Connects to system RESET.
- **RL7-RL0** : Return lines are inputs used to sense key depression in the keyboard matrix.
- **Shift** : Shift connects to Shift key on keyboard.
- **SL3-SL0** : Scan line outputs scan both the keyboard and displays.

## **Input Modes**

- Scanned Keyboard—with encoded ( $8 \times 8$  key keyboard) or decoded ( $4 \times 8$  key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key roll-over.
- Scanned Sensor Matrix—with encoded ( $8 \times 8$  matrix switches) or decoded ( $4 \times 8$  matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.
- Strobed Input—Data on return lines during control line strobe is transferred to FIFO.

## **Output Modes**

- 8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit ( $B_0 = D_0, A_3 = D_7$ ).
- Right entry or left entry display formats.



# **Block Diagram**

Mainly 4 Sections:

**1.KeyBoard Section**

**2.Scan Section**

**3.Display Section**

**8086 Interfacing ICs** **4. I/O Interface Section**

# Keyboard Section

- RL0 – RL7 connected to 8 columns of keyboard
- 2 modes: 2-key lockout  
N key Rollover
- FIFO RAM with 8 registers to store 8 keyboard entries and each read in the order of their entries.
- Sent IRQ signal when FIFO is not empty

# Scan Section

- It has a Scan Counter and 4 Scan Lines
- Connected to 4 – to – 16 Decoder to generate 16 scan lines
- Scan lines can be connected to rows of Matrix Keyboard and Digit drivers of Display

# Display Section

- 8 Output lines divided into 2 groups:  
**A0 – A3 and B0 – B3**
- Can be used as 8 or 4/4
- BDline for display Blanking
- 16 X 8 Display RAM

# $\mu$ p Interface Section

- Data bus : DB0 – DB7
- One IRQ Line
- Six interface lines and A0
- A0 = 1 ; signals are control/status word

# **8279 COMMAND WORDS**

## **(AO = 1)**

1. Keyboard/Display Mode
2. Program Clock
3. Read FIFO/Sensor RAM
4. Read Display RAM
5. Write Display RAM
6. Display write inhibit / Blanking
7. Clear
8. End Interrupt/Error mode set

# D7,D6,D5 Configurations

D7	D6	D5	Function	Purpose
0	0	0	Mode set	Selects the number of display positions, type of key scan...
0	0	1	Clock	Programs internal clk, sets scan and debounce times.
0	1	0	Read FIFO	Selects type of FIFO read and address of the read.
0	1	1	Read Display	Selects type of display read and address of the read.
1	0	0	Write Display	Selects type of write and the address of the write.
1	0	1	Display write inhibit	Allows half-bytes to be blanked.
1	1	0	Clear	Clears the display or FIFO
1	1	1	End interrupt	Clears the IRQ signal to the microprocessor.

## Keyboard/Display Mode Set

	MSB		LSB
Code:	0	0	0 D D K K K

Where DD is the Display Mode and KKK is the Keyboard Mode.

### DD

- 0 0 8 8-bit character display—Left entry
- 0 1 16 8-bit character display—Left entry\*
- 1 0 8 8-bit character display—Right entry
- 1 1 16 8-bit character display—Right entry

**KKK**

- |       |                                       |
|-------|---------------------------------------|
| 0 0 0 | Encoded Scan Keyboard—2 Key Lock-out* |
| 0 0 1 | Decoded Scan Keyboard—2-Key Lock-out  |
| 0 1 0 | Encoded Scan Keyboard—N-Key Roll-over |
| 0 1 1 | Decoded Scan Keyboard—N-Key Roll-over |
| 1 0 0 | Encoded Scan Sensor Matrix            |
| 1 0 1 | Decoded Scan Sensor Matrix            |
| 1 1 0 | Strobed Input, Encoded Display Scan   |
| 1 1 1 | Strobed Input, Decoded Display Scan   |

## Program Clock

Code:

0	0	1	P	P	P	P	P
---	---	---	---	---	---	---	---

All timing and multiplexing signals for the 8279 are generated by an internal prescaler. This prescaler divides the external clock (pin 3) by a programmable integer. Bits PPPPP determine the value of this integer which ranges from 2 to 31.

## **Read FIFO/Sensor RAM**

Code:

0	1	0	AI	X	A	A	A
---	---	---	----	---	---	---	---

X = Don't Care

The CPU sets the 8279 for a read of the FIFO/Sensor RAM by first writing this command. In the Scan Keyboard Mode, the Auto-Increment flag (AI) and the RAM address bits (AAA) are irrelevant. The 8279 will automatically drive the data bus for each subsequent read ( $A_0 = 0$ ) in the same sequence in which the data first entered the FIFO. All subsequent reads will be from the FIFO until another command is issued.

In the Sensor Matrix Mode, the RAM address bits AAA select one of the 8 rows of the Sensor RAM. If the AI flag is set ( $AI = 1$ ), each successive read will be from the subsequent row of the sensor RAM.

## Read Display RAM

Code:

0	1	1	A1	A	A	A	A	A
---	---	---	----	---	---	---	---	---

The CPU sets up the 8279 for a read of the Display RAM by first writing this command. The address bits AAAA select one of the 16 rows of the Display RAM. If the A1 flag is set ( $A1 = 1$ ), this row address will be incremented after each following read *or write* to the Display RAM. Since the same counter is used for both reading and writing, this command sets the next read or *write* address and the sense of the Auto-Increment mode for both operations.

## Write Display RAM

Code:

1	0	0	AI	A	A	A	A
---	---	---	----	---	---	---	---

The CPU sets up the 8279 for a write to the Display RAM by first writing this command. After writing the command with  $A_0 = 1$ , all subsequent writes with  $A_0 = 0$  will be to the Display RAM. The addressing and Auto-Increment functions are identical to those for the Read Display RAM. However, this command does not affect the source of subsequent Data Reads; the CPU will read from whichever RAM (Display or FIFO/Sensor) which was last specified. If, indeed, the Display RAM was last specified, the Write Display RAM will, nevertheless, change the next Read location.

## Display Write Inhibit/Blanking

	A	B	A	B
Code:	1	0	1	X

The IW Bits can be used to mask nibble A and nibble B in applications requiring separate 4-bit display ports. By setting the IW flag (IW = 1) for one of the ports, the port becomes marked so that entries to the Display RAM from the CPU do not affect that port.

If the user wishes to blank the display, the BL flags are available for each nibble. The last Clear command issued determines the code to be used as a "blank." This code defaults to all zeros after a reset. Note that both BL flags must be set to blank a display formatted with a single 8-bit port.

## Clear

Code:

1	1	0	$C_D$	$C_D$	$C_D$	$C_F$	$C_A$
---	---	---	-------	-------	-------	-------	-------

The  $C_D$  bits are available in this command to clear all rows of the Display RAM to a selectable blanking code as follows:

$C_D$	$C_D$	$C_D$	
0	X		All Zeros (X = Don't Care)
1	0		AB = Hex 20 (0010 0000)
1	1		All Ones

Enable clear display when  $= 1$  (or by  $C_A = 1$ )

If the  $C_F$  bit is asserted ( $C_F = 1$ ), the FIFO status is cleared and the interrupt output line is reset. Also, the Sensor RAM pointer is set to row 0.

$C_A$ , the Clear All bit, has the combined effect of  $C_D$  and  $C_F$ ; it uses the  $C_D$  clearing code on the Display RAM and also clears FIFO status. Furthermore, it resynchronizes the internal timing chain.

## **End Interrupt/Error Mode Set**

Code:

1	1	1	E	X	X	X	X
---	---	---	---	---	---	---	---

x = Don't care

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM.

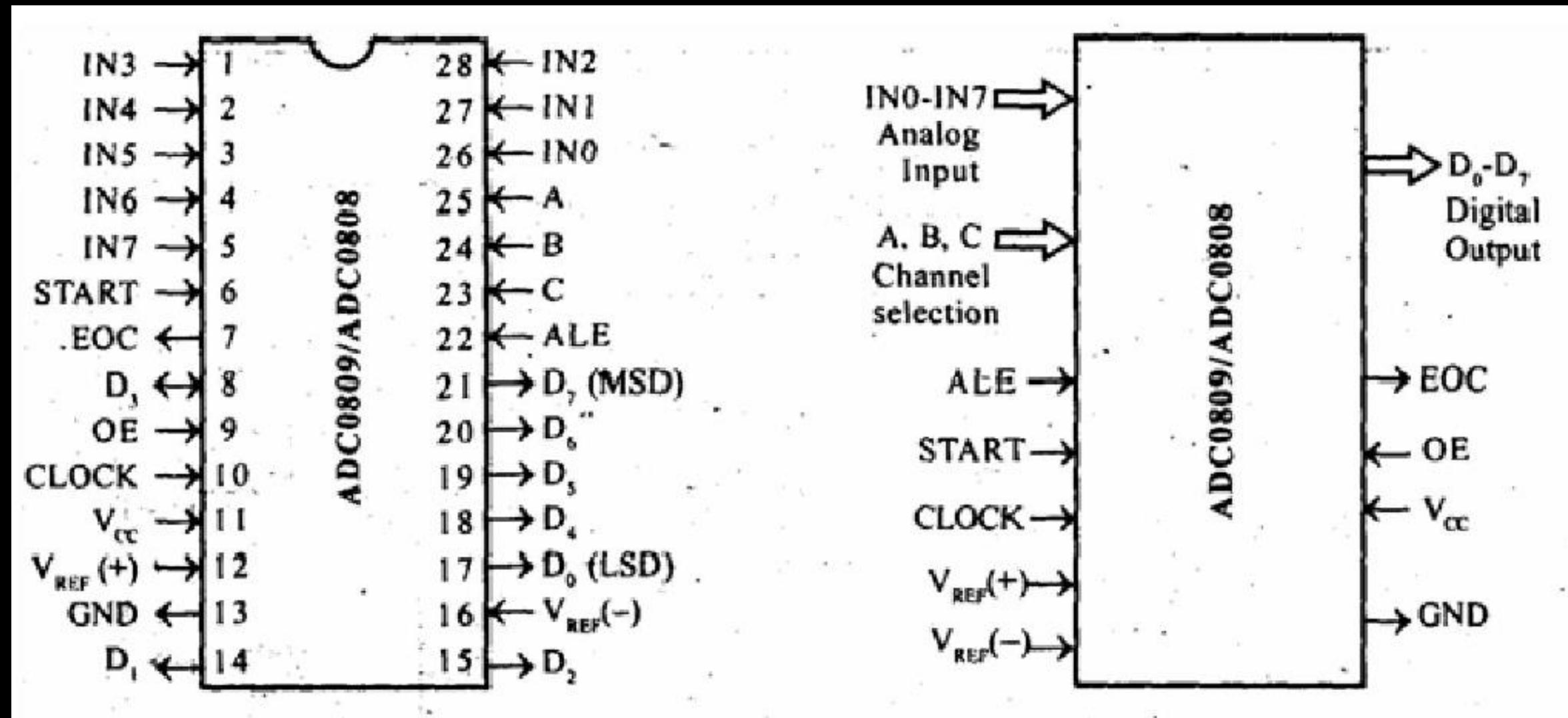
For the N-key rollover mode—if the E bit is programmed to “1” the chip will operate in the special Error mode.

# ADC

## ADC 0809

- The ADC0809 is an 8-bit successive approximation type ADC with inbuilt 8-channel multiplexer.
- The ADC0809 is suitable for interface with 8086 microprocessor.
- The ADC0809 is available as a 28 pin IC in DIP (Dual Inline Package).
- The ADC0809 has a total unadjusted error of  $\pm 1$  LSD (Least Significant Digit).
- The ADC0808 is also same as ADC0809 except the error. The total unadjusted error in ADC0808 is  $\pm 1/2$  LSD.

# ADC



Signals	Description
IN0-IN7	Eight single ended analog input to ADC.
A, B, C	3-bit binary input to select one of the eight analog signals for conversion at any one time.
ALE	Address latch enable. Used to latch the 3-bit address input to an internal latch.
START	Start of conversion pulse input. To start ADC process this signal should be asserted <b>high</b> and then <b>low</b> . This signal should remain <b>high</b> for atleast 100ns.
CLOCK	Clock input and the frequency of clock can be in the range of 10 kHz to 1280 kHz. Typical clock input is 640 kHz.
$V_{REF}(+), V_{REF}(-)$	Reference voltage input. The positive reference voltage can be less than or equal to $V_{cc}$ and the negative reference voltage can be greater than or equal to ground.
D <sub>0</sub> -D <sub>7</sub>	The 8-bit digital output. The reference voltages will decide the mapping of analog input to digital data.
EOC	End of conversion. This signal is asserted <b>high</b> by the ADC to indicate the end of conversion process and it can be used as interrupt signal to processor.
OE	Output buffer Enable. This signal is used to read the digital data from output buffer after a valid EOC.
$V_{cc}$	Power supply, +5V
GND	Power supply ground, 0V

# ADC

## INTERNAL BLOCK DIAGRAM & WORKING OF ADC0809/ADC0808

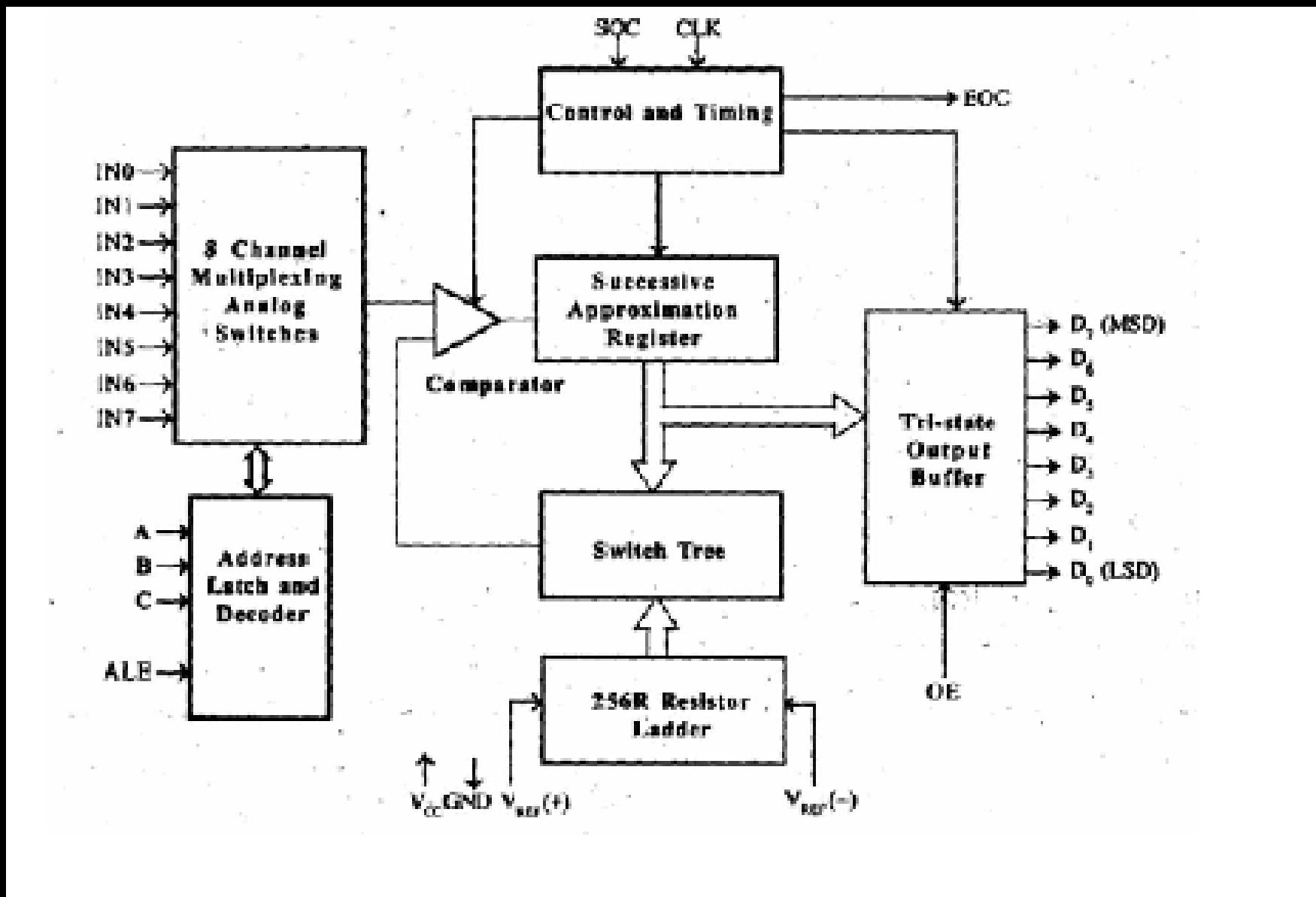
The internal block diagram of ADC0809/ADC0808 is,

The various functional blocks of ADC are 8-channel multiplexer, comparator, 256R resistor ladder, switch tree, successive approximation register, output buffer, address latch and decoder.

- The 8-channel multiplexer can accept eight analog inputs in the range of 0 to 5V and allow one by one for conversion depending on the 3-bit address input. The channel selection logic is,

Address Input			Selected Channel
C	B	A	
0	0	0	IN0
0	0	1	IN1
0	1	0	IN2
0	1	1	IN3
1	0	0	IN4
1	0	1	IN5
1	1	0	IN6
1	1	1	IN7

# ADC



# ADC

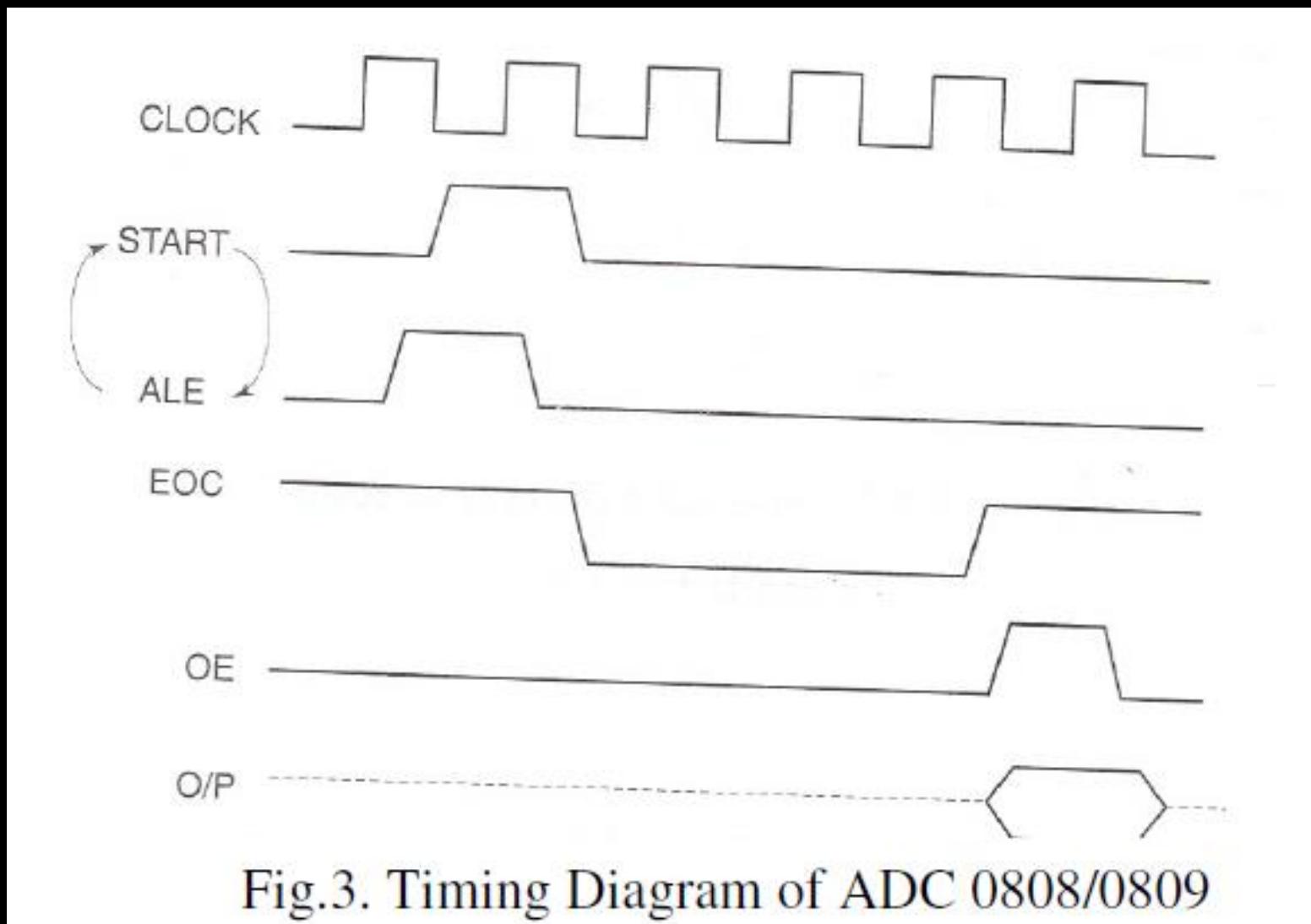


Fig.3. Timing Diagram of ADC 0808/0809

# ADC interfacing diagram

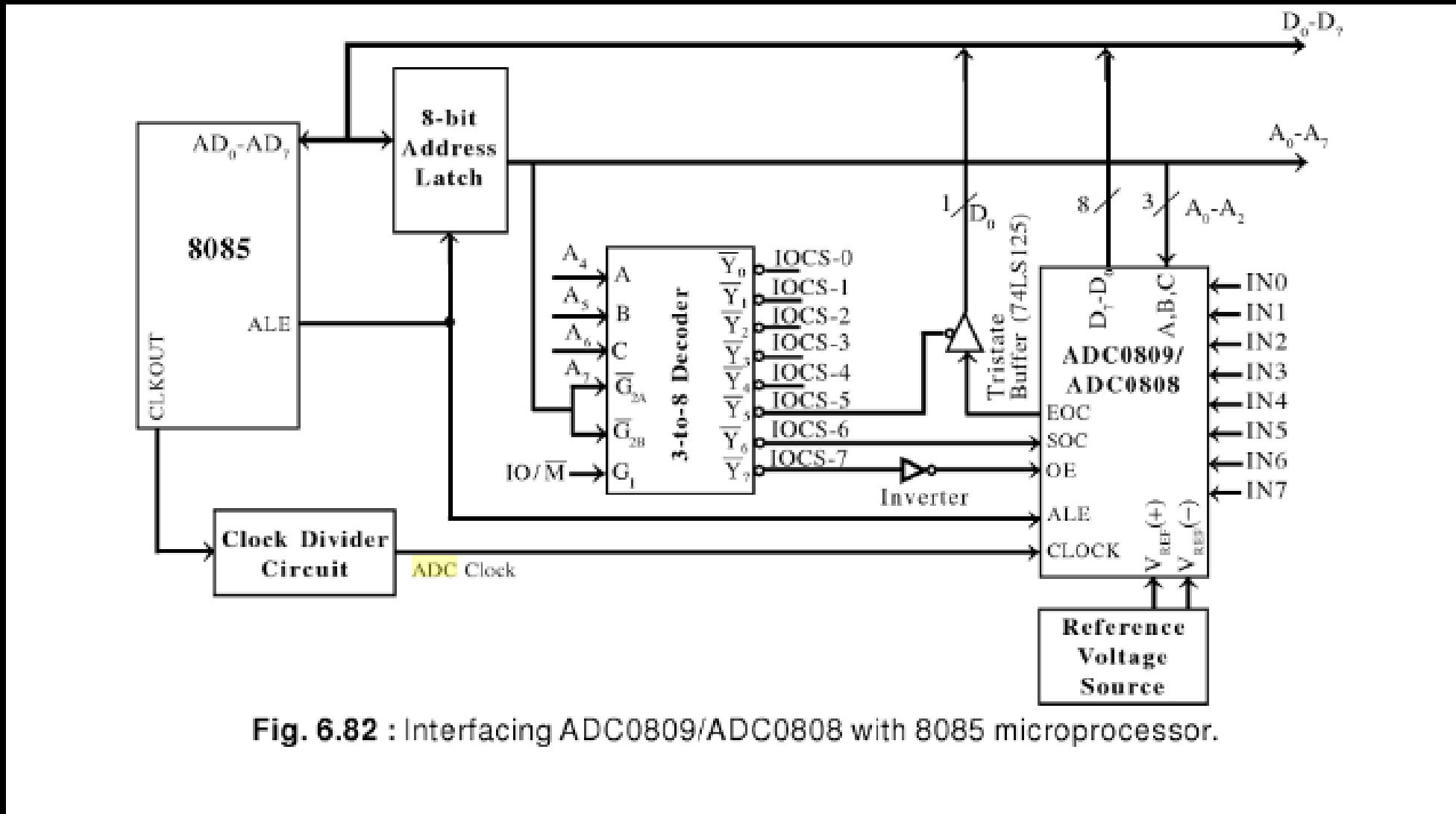


Fig. 6.82 : Interfacing ADC0809/ADC0808 with 8085 microprocessor.