8259

Programmable Interrupt Controller

Interrupts in Microcomputer System

- Microcomputer system design requires that I.O devices such as
  i. keyboards,
  ii. displays,
  iii. sensors and
  iv. other components

receive servicing in an efficient manner so that large amounts of the total system tasks can be assumed by the µcomputer with little or no effect on throughput

Method

i. Polled approach: General Method

ii. Interrupt

General Method :Polled Method

- In this method
  i. a **large portion of the main program is looping through this continuous polling cycle**
  ii. has a **serious detrimental effect on system throughput**,
  iii. thus **limiting the tasks** that could be assumed by the microcomputer and
  iv. **reducing the cost effectiveness** of using such devices

A More Reliable Method : Interrupt

- A **more desirable method** would be one that would allow the microprocessor to execute its main program and
- **only stop to service peripheral devices when it is told to do so** by the device itself.

A More Reliable Method : Interrupt

- In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device.

- Once this servicing is complete, however, the processor would resume exactly where it left off

Need for 8259

- 8086 Processor has only 2 hardware interrupts.
- Consider an application where a number of I/O devices connected with CPU desire to transfer data using interrupt driven data transfer mode.
- In this process more number of interrupt pins are required.
- In these **multiple interrupt systems the processor will have to take care of priorities.**

What is PIC & Why should we opt for it ?

- The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment.
  i. It **accepts requests** from the peripheral equipment,
  ii. **determines** which of the incoming requests is of the highest importance (**priority**),
  iii. ascertains whether the incoming request has a higher priority value than the level currently being serviced, and
  iv. **issues an interrupt to the CPU** based on this determination.

8259A PIC- FEATURES

- manages 8 levels of interrupts.
- Can be cascaded in a Master Slave configuration to handle upto 64 levels of interrupts.
- Can identify the interrupting device
- Provides a 8 bit type no.

8259A PIC- FEATURES

- Resolve 8 levels of interrupt priorities in variety of modes.
- Mask each interrupt request individually.
- Read the status of pending interrupts, in-service interrupts and masked interrupts.

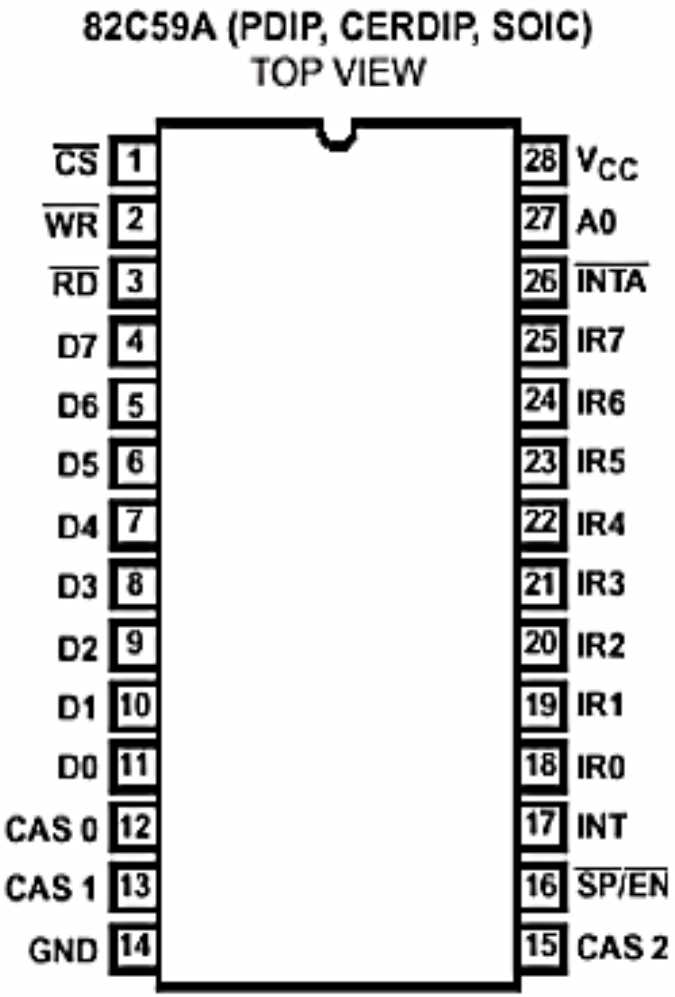
8259A PIC- FEATURES

- Be set up to accept either the level triggered or the edge triggered interrupt request.
- It can be operated in various priority modes.
- The operating modes & masks can be dynamically changed by software at the time of program execution

8259A PIC- FEATURES

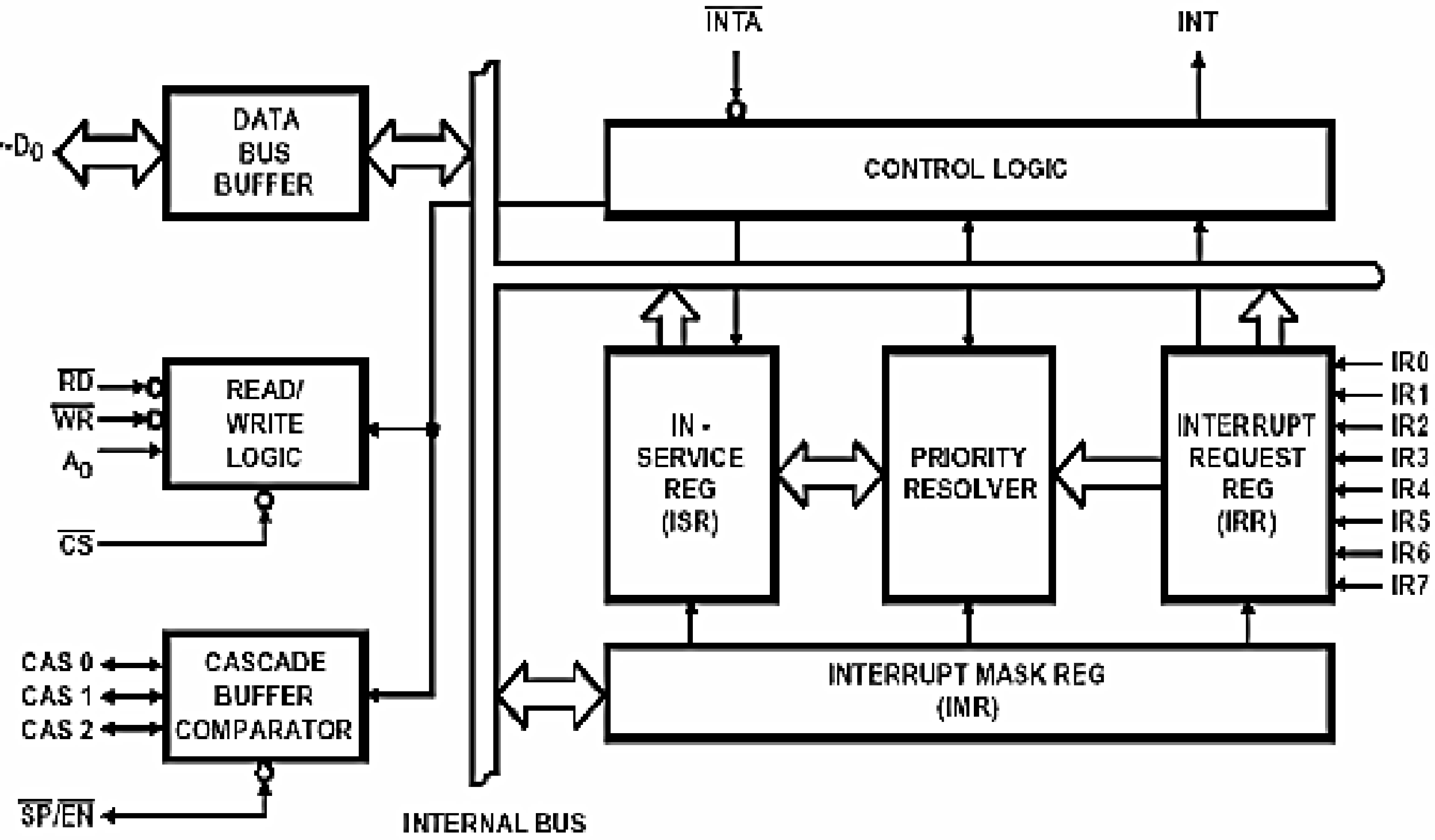
- Be expanded to 64 priority levels by cascading additional 8259As.
- Compatible with 8-bit as well as 16-bit processors.

8259A PIC- PIN DIGRAM



PIN	DESCRIPTION
D7 - D0	Data Bus (Bidirectional)
$\overline{RD}$	Read Input
$\overline{WR}$	Write Input
A0	Command Select Address
$\overline{CS}$	Chip Select
CAS 2 - CAS 0	Cascade Lines
$\overline{SP/EN}$	Slave Program Input Enable
INT	Interrupt Output
$\overline{INTA}$	Interrupt Acknowledge Input
IR0 - IR7	Interrupt Request Inputs

8259A PIC- BLOCK DIAGRAM



8259A PIC- BLOCK DIAGRAM

It includes 8 blocks.

i. Control logic
ii. Read/Write logic
iii. Data bus buffer
iv. Three registers (IRR,ISR and IMR)
v. Priority resolver
vi. Cascade Buffer

8259A PIC- BLOCK DIAGRAM

This section consists of

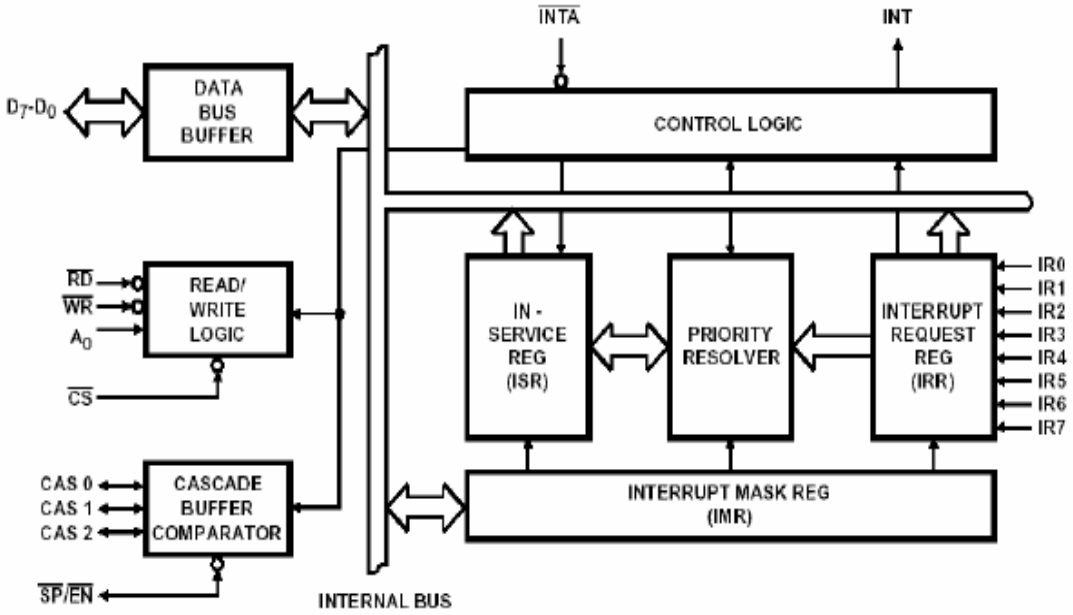
i. IRR (Interrupt Request Register)

ii. ISR (In-Service Register)

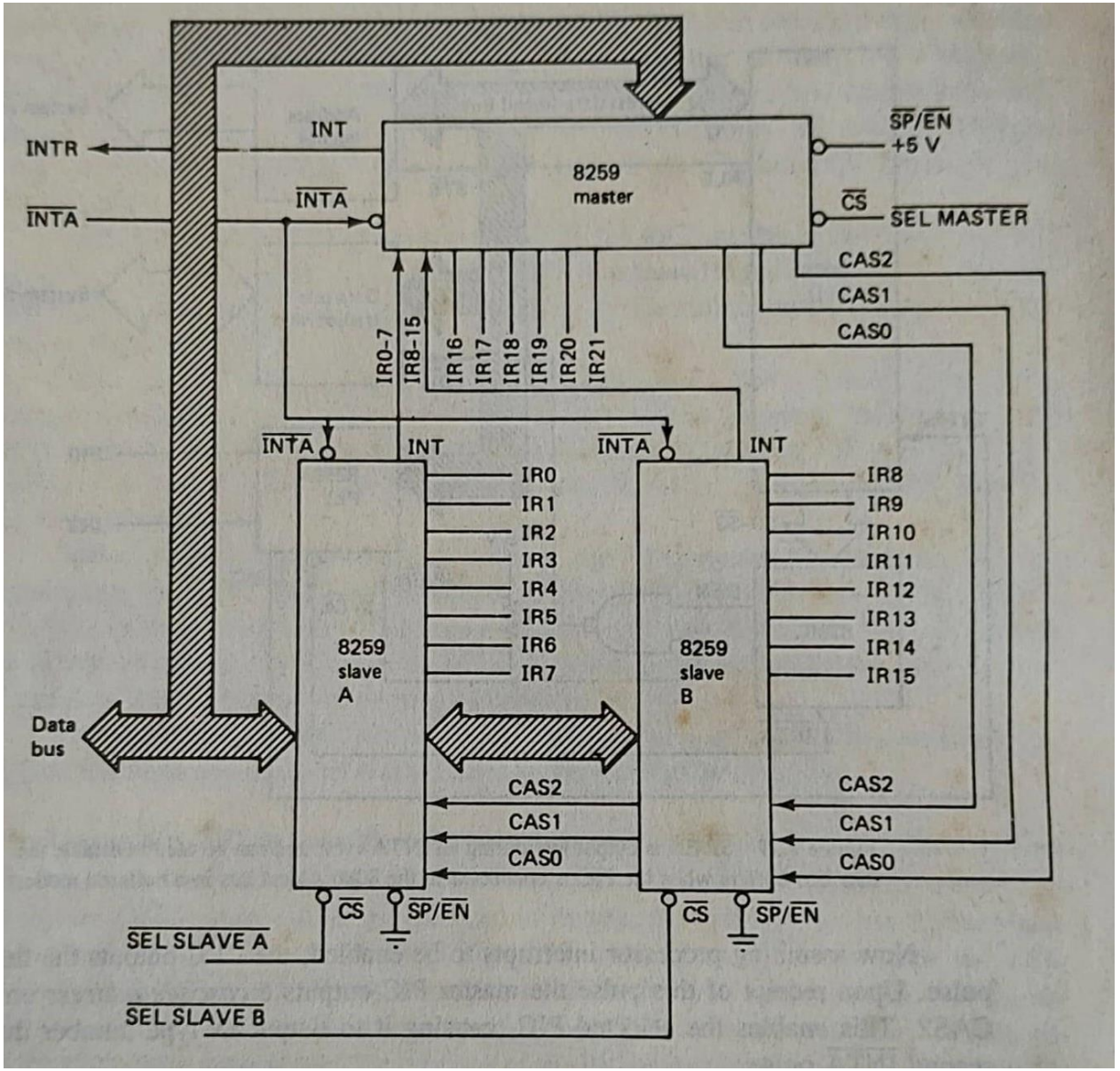
iii. Priority Resolver

iv. IMR (Interrupt Mask Register)

v. Control logic block



Cascading of PIC



8259A PIC – PRIORITY MODES

8259A PIC – PRIORITY MODES

PRIORITY MODES:

i. Fully Nested Mode

ii. Special Fully Nested Mode

iii. Rotating Priority Mode

iv. Special Masked Mode

8259A PIC – PRIORITY MODES

FULLY NESTED MODE:

- **General purpose** mode.
- **Default priority mode** - it continues to operate in this mode until it is changed through OCWs
- All IRs are arranged from highest to lowest.
- **IR0→ Highest IR7→Lowest**

8259A PIC – PRIORITY MODES

FULLY NESTED MODE:

- When the interrupt is **acknowledged**, it **sets the corresponding bit of ISReg**
- This bit will **prevent all interrupts** of the **same or lower level**, but it will accept all higher priority requests.

8259A PIC – PRIORITY MODES

FULLY NESTED MODE:

- The bit in ISReg **will remain set until EOI** command is issued by the µP at the end of ISR(Interrupt Service Routine).
- If the **AEOI** bit is set, the **bit in the ISReg resets at the trailing edge of last $\overline{\text{INTA}}$**

8259-INTERRUPT PROCESS IN FULLY NESTED MODE

i. **Prioritizes** the IR inputs : IR0 has highest & IR7 has lowest priority

ii. **For eg**:

   i. If an interrupt on IR3 is being serviced (IS3 = 1) &

   ii. a request occurs on IR2,

   iii. the controller will issue an interrupt request because IR2 has higher priority

8259-INTERRUPT PROCESS IN FULLY NESTED MODE

iv. For eg:

i. If an interrupt on IR3 is being serviced (IS3 = 1) &

ii. a request occurs on IR4,

iii. the controller will not issue an interrupt request because IR4 has lower priority

8259-INTERRUPT PROCESS IN FULLY NESTED MODE

But the **IR2 request will not be acknowledge unless the processor has set IF** within the IR3 service routine

8259-INTERRUPT PROCESS IN FULLY NESTED MODE

v. In all operating modes, the **ISReg bit corresponding to the active routine must be reset** to allow other lower priority interrupts to be acknowledged

vi. This can be done manually by issuing a special **nonspecific EOI instruction** to the controller just before IRET

8259-INTERRUPT PROCESS IN FULLY NESTED MODE

v. Alternatively, the controller can be programmed to perform this **nonspecific EOI** **automatically** when the second INTA pulse occurs

8259A- OPERATING MODES

SPECIAL FULLY NESTED MODE:

- Used in case of **larger system for the master 8259** in a cascaded configuration, and the has to be programmed in the master using ICW4
- Its **priority structure is fixed** & is same as FNM (IR0→ Highest IR7→Lowest)

8259A- OPERATING MODES

SPECIAL FULLY NESTED MODE:

- In SFNM, the master **can recognize a higher priority interrupt from a slave** whose **another interrupt is currently being serviced**

8259A- OPERATING MODES

SPECIAL FULLY NESTED MODE:

- In this mode,
  i. when an **interrupt request** from a certain slave is **in service**,
  ii. this **slave can further send requests** to the master,
  iii. if the requesting device connected to the slave has **higher priority** than the one being currently served.

8259A- OPERATING MODES

SPECIAL FULLY NESTED MODE:

- In this mode, the master **interrupts the CPU only when the interrupting device has the highest priority or the same priority** than the one currently being served.
- In normal mode, **other requests than the one being served are masked.**

8259A- OPERATING MODES

ROTATING PRIORITY MODE:

- 2 rotating priority mode
  i. Automatic Rotation
  ii. Specific Rotation.

8259A- OPERATING MODES

ROTATING PRIORITY MODE: Automatic Rotation

i. It is preferred when **several interrupt sources are of equal priority**

ii. In this mode, a **device after being serviced becomes the lowest priority** and the **consecutive next interrupt becomes highest priority.**

iii. All other **priorities rotate subsequently**.

8259A- OPERATING MODES

Before Rotate (IR4 the highest priority requiring service) :-

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0	
0	1	0	1	0	0	0	0	IS STATUS
7	6	5	4	3	2	1	0	Priority Status

After Rotate (IR4 the highest priority requiring service) :-

IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0	
0	1	0	0	0	0	0	0	IS STATUS
2	1	0	7	6	5	4	3	Priority Status

8259A- OPERATING MODES

ROTATING PRIORITY MODE: Specific Rotation

i. It is also rotating priority, but the **user can set any IR level for lowest priority**, & thus **fix all other priorities.**

ii. In this mode, lowest priority can be assigned to any interrupt input (IR0 to IR7) **by specific rotation command.**

8259A- OPERATING MODES

ROTATING PRIORITY MODE: Specific Rotation

iii. The programmer can change priorities

- by programming the bottom priority and
- Thus fixing all other priorities

i.e., if IR5 is programmed as the bottom priority device, then IR6 will have the highest one.

8259A- OPERATING MODES

ROTATING PRIORITY MODE: Specific Rotation

- The Set Priority command is issued in **OCW2**
- In this mode **internal status is updated by software control during OCW2.**

8259A- OPERATING MODES

ROTATING PRIORITY MODE: Specific Rotation

- It is **independent of the End of Interrupt (EOI)** command (also executed by OCW2).
- Priority changes can be executed during an EOI command by using the **Rotate on Specific EOI command in OCW2.**

8259A- OPERATING MODES

SPECIAL MASK MODE

- 8259 **disables interrupts lower or equal** to the interrupt, which is currently in service
- If an **interrupt is in service, then the corresponding bit in the ISReg is set** & the lowest priority interrupts are inhibited
- In SMM 8259 **permits interrupts of all levels** (lower or higher) except the one currently in service

8259A- OPERATING MODES

POLL MODE

- Here the INT line of 8259 is disabled

The µP gives poll command to the 8259 using OCW3

In return, 8259 provides the poll word to the µP

The poll word indicates the highest priority interrupt, which requires service

Thereafter the µP services the interrupt.

8259A- OPERATING MODES

POLL MODE

- It is preferred when
  - Subroutine is common for several interrupt levels.
  - To expand no of interrupt levels more than 64.

8259A- OPERATING MODES

POLL MODE

- Poll word

D7	D6	D5	D4	D3	D2	D1	D0
I	-	-	-	-		w2	w1	w0

W0 - W2: Binary code of the highest priority level requesting service.

I : Equal to ``1'' if there is an interrupt.

8259A- OPERATING MODES

BUFFERED MODE

- In this mode SP/$\overline{EN}$ becomes low during $\overline{INTA}$ cycle
- 8259A sends an enable signal on $\overline{SP/EN}$ to enable the buffers.
- In this mode, whenever the 8259A's data bus outputs are enabled, the SP/EN output becomes active.

8259A- OPERATING MODES

BUFFERED MODE

- When the 8259A is used in a large system ---→ the cascading mode is used.
- This modification forces the use of software programming to determine whether the 8259A is a master or a slave.
- **Bit 3 in ICW4 programs the buffered mode, and bit 2 in ICW4 determines whether it is a master or a slave**.

8259A- OPERATING MODES

END OF INTERRUPT (EOI)

- After the completion of an interrupt service, the corresponding ISR bits needs to be reset to update the information in the ISR. This is called EOI command.
- It can be issued in three formats
  i. Non Specific EOI Command
  ii. Specific EOI Command
  iii. Automatic EOI

8259A- OPERATING MODES

END OF INTERRUPT (EOI)

- When this command is sent to 8259A, it resets the highest priority ISR bit.
- This command specifies which ISR bit is to reset.

8259A- OPERATING MODES

i. Non Specific EOI Command

- When this command is sent to 8259A, it resets the highest priority ISR bit.

ii. Specific EOI Command

- This command specifies which ISR bit is to reset.

SPECIFIC EOI COMMAND FORMAT

A0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	1	0	0	L2	L1	L0

ISR Bit is to be reset

0 0 0 → 0th Bit

0 0 1 → 1st Bit

Etc…

8259A- OPERATING MODES

iii. Automatic EOI

- In this mode, no command is necessary.
- During the third interrupt acknowledge cycle, the ISR bit is reset.
- DRAWBACK: The ISR does not have information about which ISR is being serviced. Thus, any IR can interrupt the service routine, irrespective of its priority, if the interrupt enable FF is set.

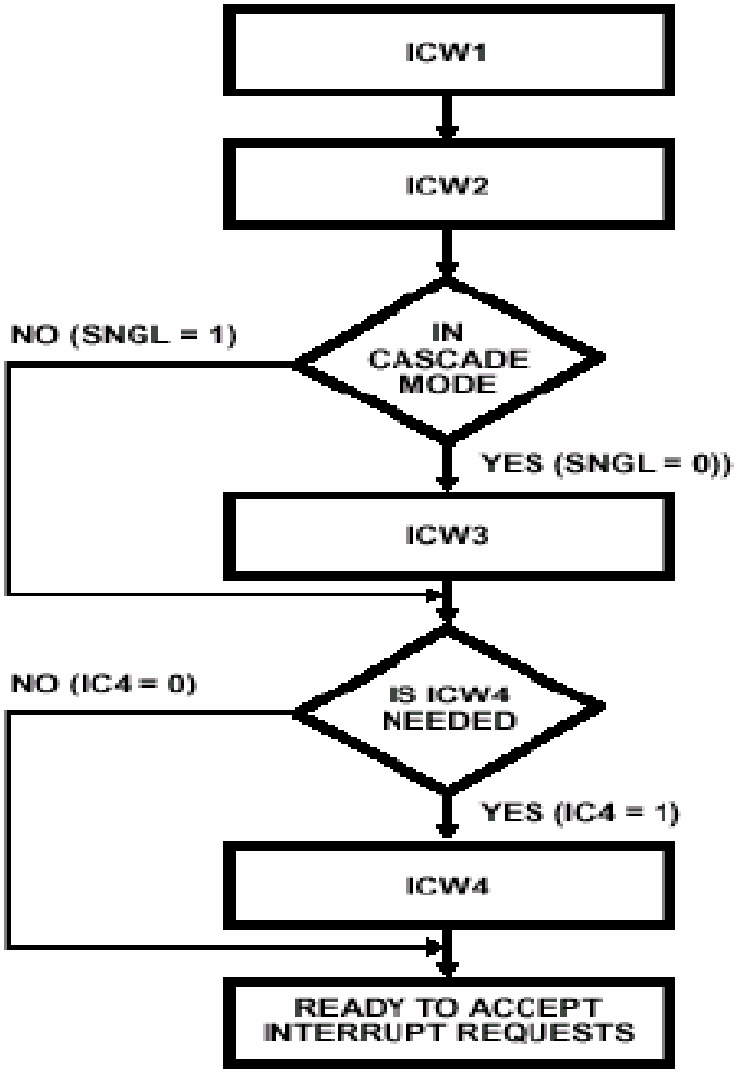
8259A PIC- PROGRAMMING

Programming

8259A PIC- PROGRAMMING

- 8259 can be programmed through a sequence of simple I/O operations
- It accepts 2 types of command words. They are:
  i. Initialization Command Words (ICWs)
  ii. Operation Command Words (OCWs)
- The 8259 can be initialized with 4 ICWs
- The first 2 are compulsory and the other 2 are optional based on the modes being used
- The initialization word should be issued in a given sequence

INITIALIZATION SEQUENCE OF 8259A



ICW1 & ICW2 are Compulsory command Words in the initialization sequence.

ICW3 & ICW4 are Optional.

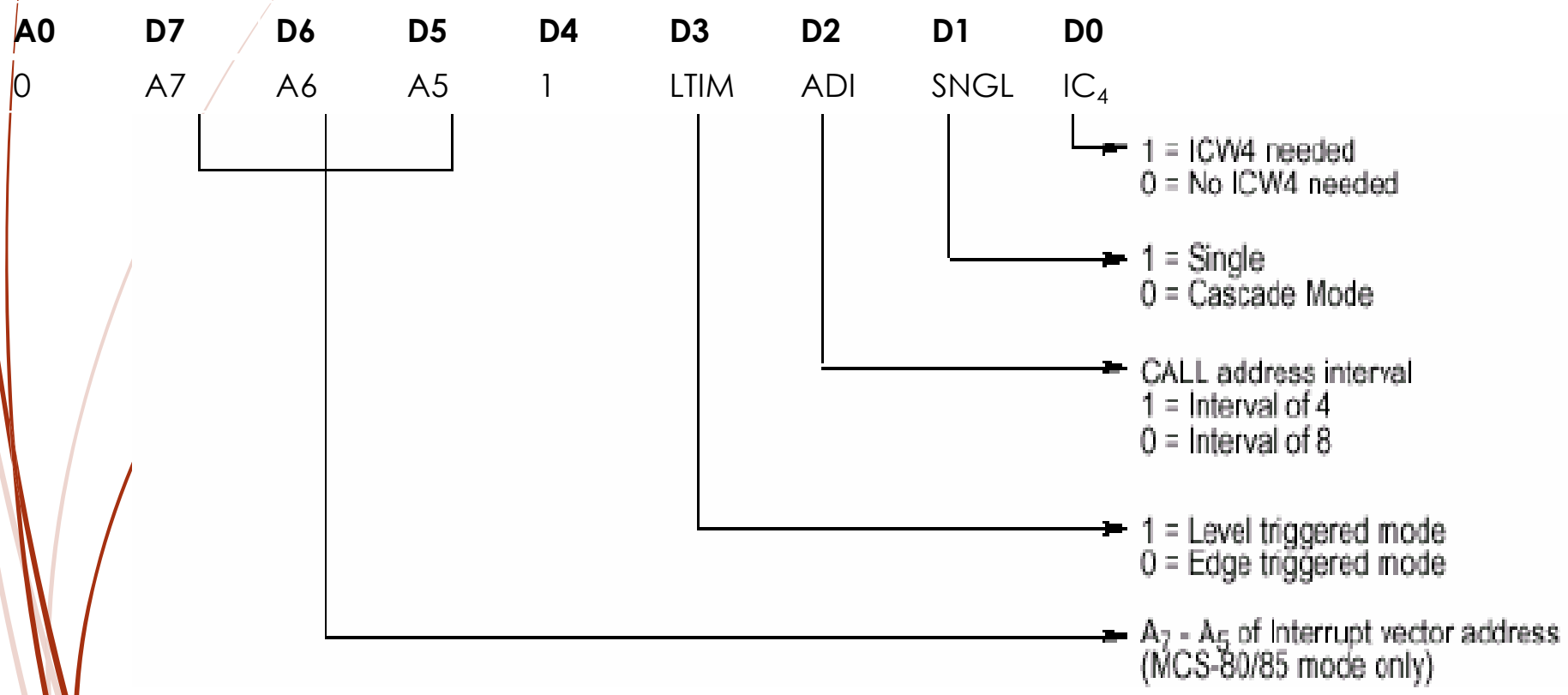
ICW3 is read only when More than one 8259 used in the system (SNGL bit in ICW1 is 0).

8259A PIC- PROGRAMMING

- After initialization 8259 can be set to operate in various modes by using 3 different OCWs
- They need not be issued in specific sequence
- They may be loaded any time after initialization to dynamically alter the priority modes

ICW Format

- ICW 1:



ICW Format (contd.)

- ICW 2:

A0	D0	D1	D2	D3	D4	D5	D6	D7
1	A15/T7	A14/T6	A13/T5	A12/T4	A11/T3	A10	A9	A8

A8 – A15

(VECTOR ADDRESSES in case of MCS 80/85 system)

T3-T7

(VECTOR ADDRESSES in case of MCS 8086/8088 system)

ICW Format (contd.)

- ICW 3: (MASTER MODE)

A0	D0	D1	D2	D3	D4	D5	D6	D7
1	S7	S6	S5	S4	S3	S2	S1	S0

S0 – S7 = 1, IR input has a slave

= 0, IR input does not have a slave

- ICW 3: (SLAVE MODE)

A0	D0	D1	D2	D3	D4	D5	D6	D7
1	0	0	0	0	0	ID2	ID1	ID0

- ID0-2 = Slave IDs

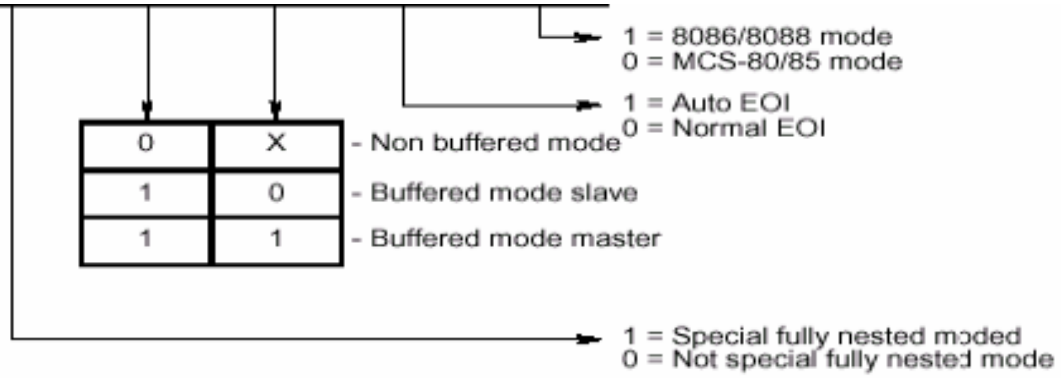
8259A ICW

ID2	ID1	ID0	Slave Pic
0	0	0	Slave on IR0
0	0	1	Slave on IR1
0	1	1	Slave on IR2
0	1	1	Slave on IR3
1	0	0	Slave on IR4
1	0	1	Slave on IR5
1	1	0	Slave on IR6
1	1	1	Slave on IR7

ICW Format

- ICW 4:

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	0	0	0	SFNM	BUF	M/S	AEOI	µPM



AEOI mode requires no commands. During the second INTA the ISR bit is reset. The major drawback with this mode is that the ISR doesn't have info on which IR is served. Thus any IR with any priority can *now* Interrupt service routine.

BUF when 1 selects buffer mode. The SP/EN pin becomes an output for the data buffers.
When 0, the SP/EN pin becomes the input for the (MASTER/SLAVE) functionality

M/S is used to set the function of the 8259 when operated in buffered mode.
If M/S is set the 8259 will function as the MASTER.
If cleared will function as SLAVE.

Operation Command Words (OCW)

- OCW1:-

A0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0

Interrupt Mask = 1 Mask Set

= 0 Mask Reset

Operation Command Words (OCW)

- OCW 2:-
- A0 D7 D6 D5 D4 D3 D2 D1 D0

  0 R SL EOI 0 0 L2 L1 L0

  0 0 1 - Non-Specific EOI Command

  0 1 1 - Specific EOI Command

  1 0 1 - Rotate on Non-Specific EOI Command

  1 0 0 - Rotate in automatic EOI mode (Set)

  0 0 0 - Rotate in automatic EOI mode (Clear)

  1 1 1 - Rotate on Specific EOI command

  1 1 0 - Set Priority Command

  0 1 0 - No Opearation
- L0 – L2 = IR Level to be acted upon

Operation Command Words (OCW)

- OCW 3 :-
- A0 D7 D6 D5 D4 D3 D2 D1 D0

0 0 ESMM SMM 0 1 P RR RIS

No Action 0 0

No Action 0 1

Read IR reg. on next RD pulse 1 0

Read IR reg. on next RD pulse 1 1

P =1 Poll Command

=0 No Poll Command

ESMM	SMM	
0	X	No action
1	0	Reset special mask mode
1	1	Set special mask mode

ADDITIONAL FEATURES OF THE 8259A

INTERRUPT TRIGGERING:

- 8259A can accept an interrupt request with either the edge triggered or level triggered mode.
- Mode is determined by initialization instructions.

INTERRUPT STATUS:

- The status of the three interrupt registers (IRR, ISR and IMR) can be read, and this status information can be used to make the interrupt process versatile.

POLL METHOD:

- 8259A can be set up to function in polled environment.
- MPU polls the 8259A rather than each peripheral.

8259A PIC- INTERRUPT OPERATION

- To implement interrupt, the interrupt Enable FF must be enabled by writing EI & STI instruction
- 8259A→ should be initialized by writing control words in the control register.
- 8259 requires two types of control words:

  ICW→ Used to set up proper conditions and specify RST vector address.

  OCW→ Used to perform functions such as masking interrupts, setting up status, read operations etc.

8259A PIC- INTERRUPT OPERATION

After 8259A is initialized, the following sequence of events occurs when one or more interrupt request lines go high.

1. IRR stores the Interrupt requests.
2. Priority Resolver Checks three registers:

   IRR→ for interrupt requests.

   IMR→ for Masking bits.

   ISR→ for the interrupt request being serviced.

It resolves the priority and sets the INT high when appropriate.

8259A PIC- INTERRUPT OPERATION

3. μP acknowledges the interrupt by sending interrupt acknowledge.

4. After $\overline{\text{INTA}}$ is received,
   i. the appropriate priority bit in the ISR is set to indicate which level is being served and
   ii. the corresponding bit in the IRR is reset to that request is accepted.
   iii. 8259 prepares to to send the vector number N to the μP on the data bus

8259A PIC- INTERRUPT OPERATION

5. When µP sends the second $\overline{INTA}$ to 8259.

6. When 8259 receives second $\overline{INTA}$, 8259 sends the 1 byte vector no on the data bus.

7. Now the µP multiplies N x 4 to calculate address in the IVT table

8259A PIC- INTERRUPT OPERATION

8. In AEOI, the ISR bit is reset either automatically (AEOI) or by a command word that must be issued at the end of the service routine (EOI). This option is determined by the ICW.
9. The µP pushes the contents of Flag Register, CS, IP in to the the stack, clears the IF & TF & the program sequence is transferred to the address of the Interrupt Subroutine

AEOI → Automatic End of Interrupt Mode

EOI → End of Interrupt Mode