INTERRUPTS

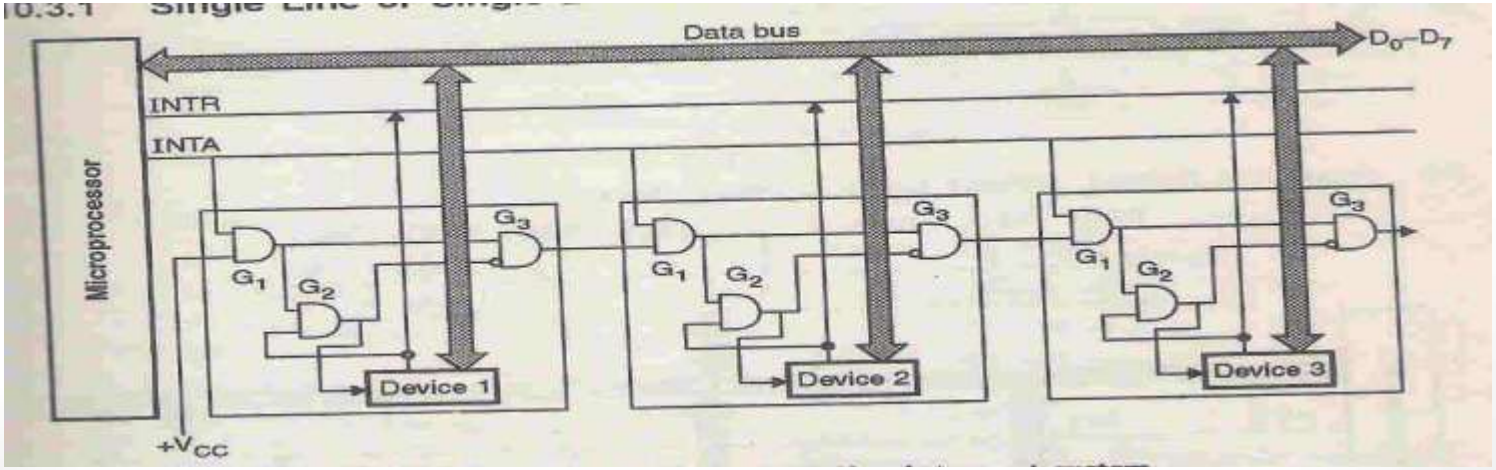
RUNGTA COLLEGE OF ENGG AND TECH.,BHILAI
DEPT OF ELECTRONICS AND TELE. ENGG

Interrupt system

- Single level interrupt system
- Multiple level interrupt System

Single level Interrupt System

1. CPU provide only one interrupt request line
2. MP ignore all other interrupt system during execution of any interrupt service routine
3. Here the problem to identify the and giving the priority to interrupting device can be solve using software and hardware polling



Multiple level Interrupt system

1. Here the interrupt control logic provide multiple interrupt line and pass one of them as per the priority of interrupting device
2. MP can accept other interrupt during execution of any interrupt service routine

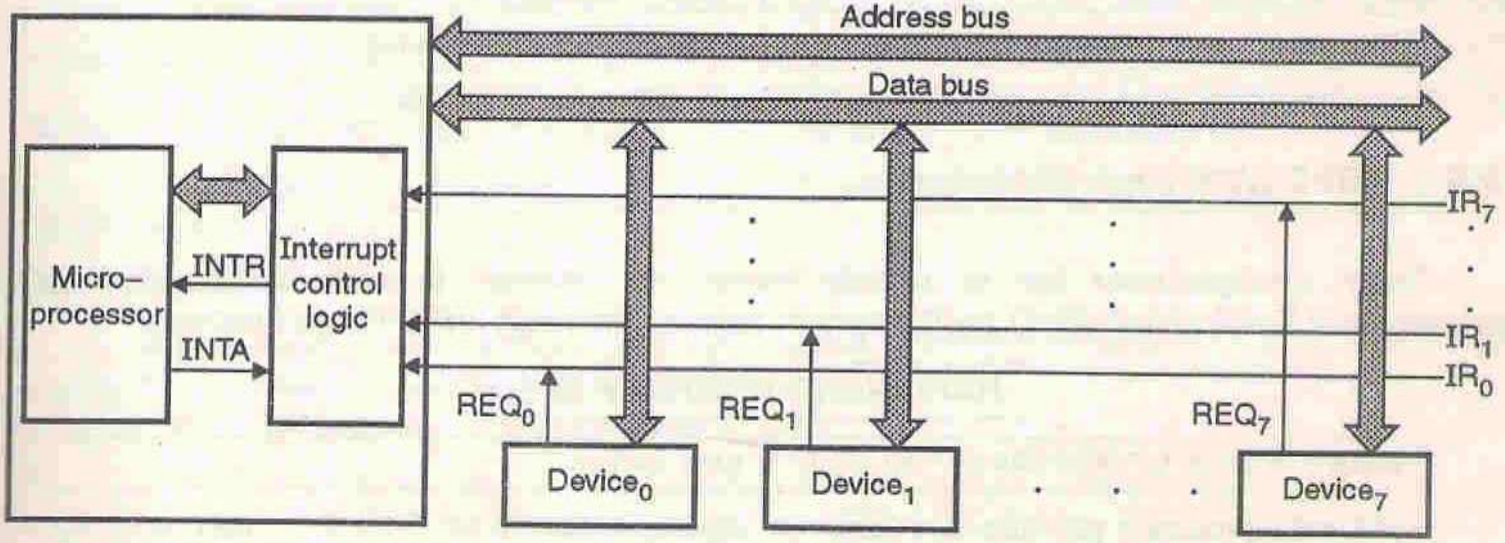


Fig. 10.6 : Multiline or multilevel interrupt system

INTERRUPTS

MP 8085 interrupt structure

Hardware interrupt

1.trap 2. RST7.5 3. RST6.5, RST5.5 INTR

Software interrupt

RST 0 to RST 7

TRAP : -

•It is nonmaskable edged and level triggered request

•It is used for emergency purpose

•MP doesn’t execute any INTA cycle to read the interrupt information

•MP execute idle machine cycle to acknowledge this interrupt during this cycle RST 4.5 is executed and bring the control on 0024 H

•It has highest priority among all interrupt

•**RST 7.5**

•It is maskable edged triggered interrupt requested line

•MP doesn’t execute any INTA cycle to read the interrupt information

•MP execute idle machine cycle to acknowledge this interrupt during this cycle RST 7.5 is executed and bring the control on 003c H

•It has highest priority among all maskable interrupt

•Disabled by SIM and DI

RST 6.5 & RST 5.5

•This are level triggered interrupt

•MP doesn’t execute any INTA cycle to read the interrupt information

•MP execute idle machine cycle to acknowledge this interrupt during this cycle RST 6.5/ RST 5.5 is executed and bring the control on 0034/002C H

•It is maskable interrupt

INTR

•It is level triggerd interrupt **request line**

•MP execute INTA cycle to read the interrupt information from interrupting device

•Starting address depends on interrupt information

•Not affected by SIM

•Enable and disable by EI & DI

Software interrupt

•It is software generated interrupt

•It can’t be maskable

•Has highest priority among all

•Here jump instruction must be stored in to corresponding location to transfer MP controle to user defined ISR addr

- Trap RST 7.5,6.5,5.5 take only one M/C to acknowledge request
- Response time is very low
- Nos of memory location for each ISR is very low hence JUMP inst must be use to transfer the MP's control from predetermined location to user defined location

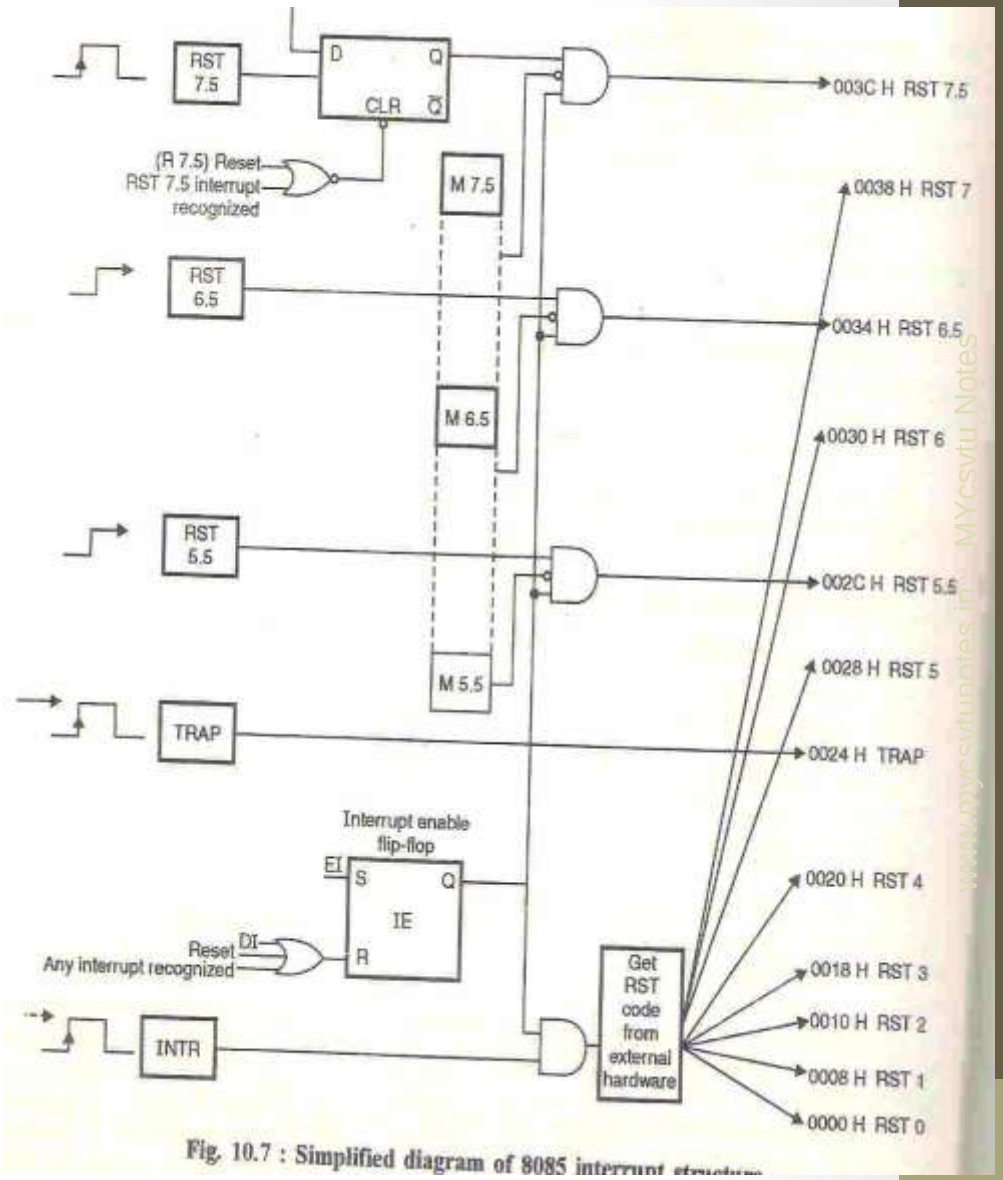


Fig. 10.7 : Simplified diagram of 8085 interrupt structure

Interrupt Structure

, for RST N, instruction is predefined. Following table depicts the same :

Instruction	Opcode	Address of ISR
RST 0	11000111 = C7	0000H (8 × 0) = 0000H
RST 1	11001111 = CF	0008H (8 × 1) = 0008H
RST 2	11010111 = D7	0010H (8 × 2) = 0010H
RST 3	11011111 = DF	0018H (8 × 3) = 0018H
RST 4	11100111 = E7	0020H (8 × 4) = 0020H
RST 5	11101111 = EF	0028H (8 × 5) = 0028H
RST 6	11110111 = F7	0030H (8 × 6) = 0030H
RST 7	11111111 = FF	0038H (8 × 7) = 0038H

e difference between two successive locations is only 8 bytes. Hence jump inst

Difference between two successive location is 8 byte hence if the ISR is greater then 8byte so JUMP instruction must be used to stored in corresponding location to transfer MP's control corresponding location

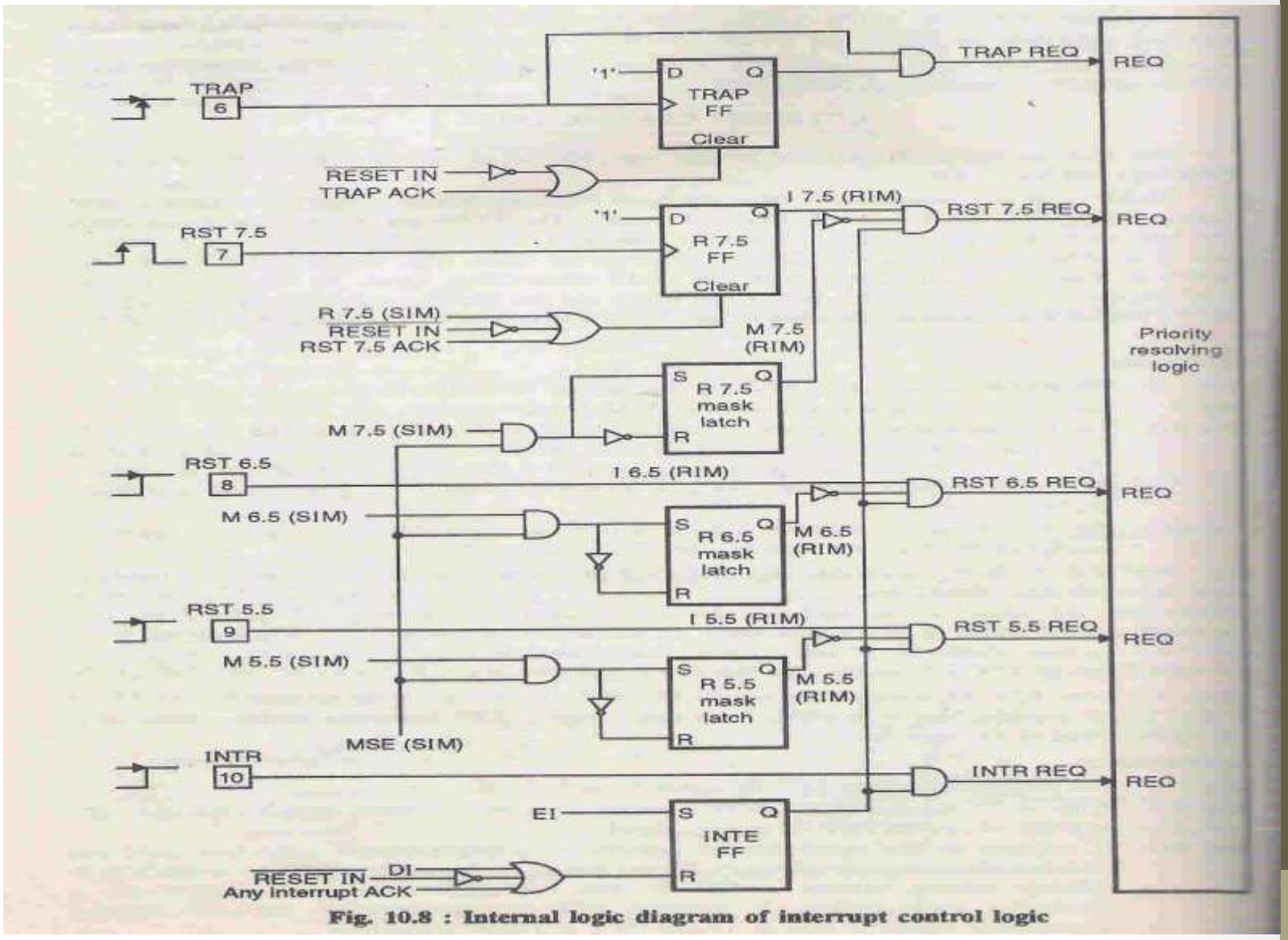


Fig. 10.8 : Internal logic diagram of interrupt control logic

Internal logic control of interrupt control logic

EI –ENABLE INTERRUPT

•It is used to enable all the maskable interrupt

•It sets an INTE f/f of interrupt control logic

•It is single byte instruction takes only one M/c

•It is also used to enable maskable interrupt during execution of ISR

•It does not effect on trap\

DI- DISABLE INTERRUPT

• It is used to disable all maskable interrupt

•It reset an INTE f/f of interrupt control logic

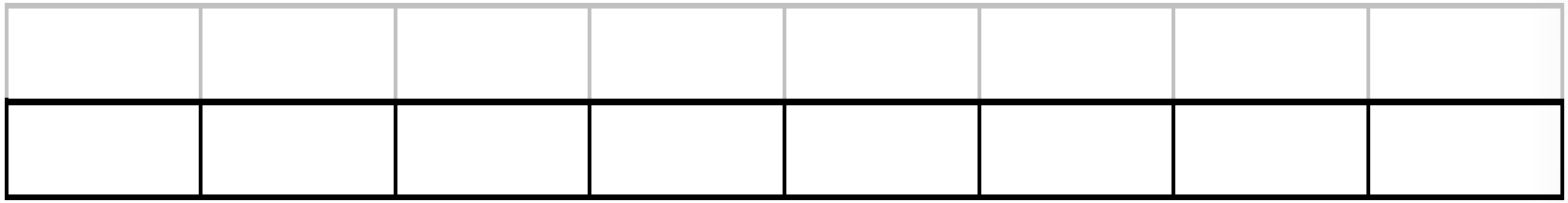
•It is single byte instruction takes only one M/c

•It is also used to prevent a critical part of program from maskable interrupt

•It does not effect on trap

SIM :SET INTERRUPT MASK

- It is one byte instruction used to enable all maskable interrupt
- It does not affect on TRAP and INTR
- It is also used in serial data transmission
- It transfer SIM format from accumulator to control logic
- It also transfer serial data bit D7 to the SOD pin



SOD: - serial output data

SDE : - serial data enable (1 enable , 0 disable)

R 7.5: - reset R 7.5 F/F (1 reset F/F , 0 no effect on F/F)

M7.5,M6.5,M5.5: - mask RST (1 mask or disable ,0 unmask)

MSE: - mask set enable (1 -d2,d1,d0 bit are effective 0 – bits are ignored)

RIM : READ INTERRUPT MASK

- It is single byte instruction used to check the status of all maskable interrupt
- It also transfer serial data bit from SID line to D7 bit of accumulator
- It does not provide status of trap & INTR
- This instruction transfer the content of interrupt control logic to accumulator
- It is used to check status of pending interrupt

							0
							5

SID:-serial data bit

I7.5,I6.5,I5.5:-RST bit is pending(1 pending ,0 not effected)

IE:-Interrupt enable (1 INTE F/F is SET , 0 F/F reset)

M7.5,M6.5,M5.5:-Mask RST (1 Masked or disable , 0 unmasked)

INTERRUPT CONTROLE LOGIC OPERATIONS

Trap

•PPI generate trap to set trap F/F

•In response to this MP completes current instruction and execute ideal cycle during this cycle MP calculate starting addr of ISR 0024H

•Than MP reset trap and INTE F/F and MP execute 2 M/C to store PC content to stack memory

RST 7.5

•PPI activate RST 7.5 to set RST 7.5 F/F

•If R 7.5 ids set and INTE F/F set and mask 7.5 F/F reset logic activates an RST 7.5 request

•In response to this MP completes current instruction and execute ideal cycle during this cycle MP calculate starting addr of ISR 003CH

•MP execute 2 M/C to store PC content to stack memory

RST 6.5 AND RST 5.5 Same as RST 7.5

INTR : -

• the PPI activates INTR signal

•In response to this INTE F/F is set

•MP completes current instruction cycle and execute one interrupt acknowledgement cycle (INTA)for call instruction from external hardware

8259 Programmable Interrupt controller

Features:

•8 levels of interrupts.

•Can be cascaded in master-slave configuration to handle 64 levels of interrupts.

•Internal priority resolve.

•Fixed priority mode and rotating priority mode.

•Individually maskable interrupts.

•Modes and masks can be changed dynamically.

•Accepts IRQ, determines priority, checks whether incoming priority > current level being serviced, issues interrupt signal.

•In 8085 mode, provides 3 byte CALL instruction. In 8086 mode, provides 8 bit vector number.

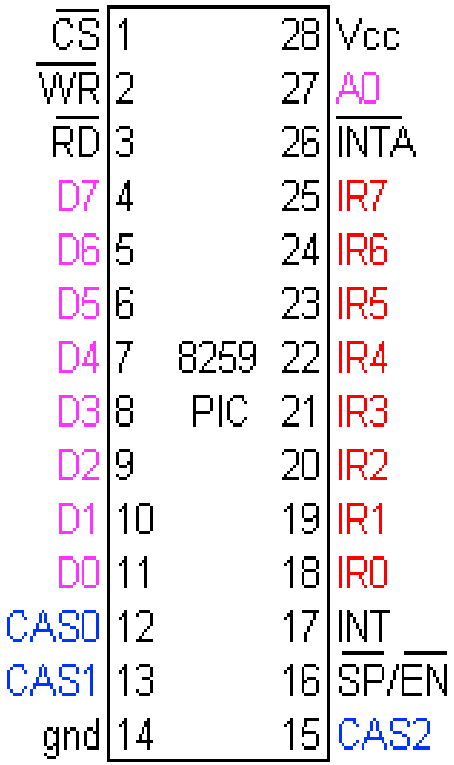
•Polled and vectored mode.

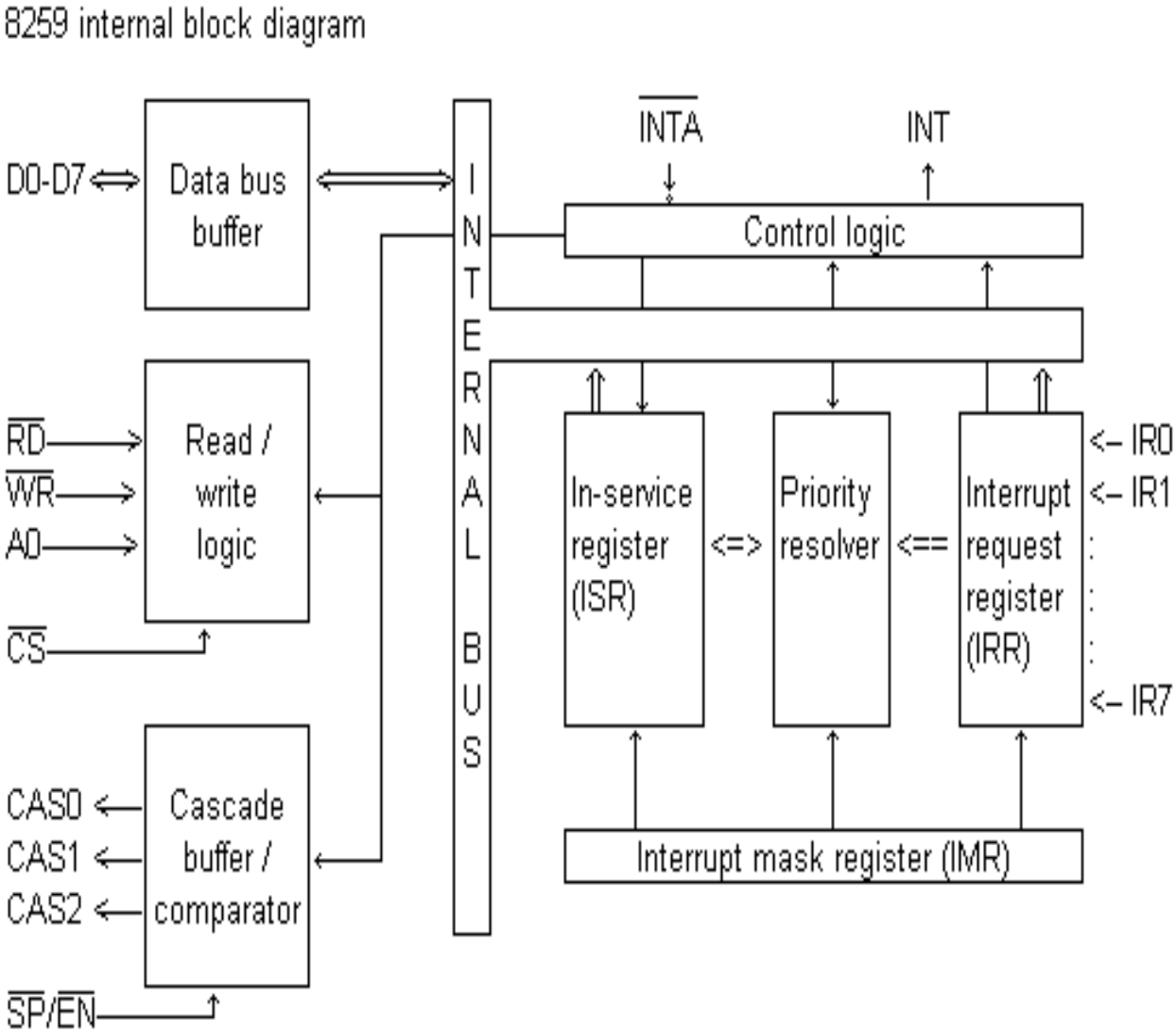
•Starting address of ISR or vector number is programmable.

•No clock required.

Block Diagram of 8259

Pin function





D0-D7	Bi-directional, tristated, buffered data lines. Connected to data bus directly or through buffers
RD-bar	Active low read control
WR-bar	Active low write control
A0	Address input line, used to select control register
CS-bar	Active low chip select
CAS0-2	Bi-directional, 3 bit cascade lines. In master mode, PIC places slave ID no. on these lines. In slave mode, the PIC reads slave ID no. from master on these lines. It may be regarded as slave-select.
SP-bar / EN-bar	Slave program / enable. In non-buffered mode, it is SP-bar input, used to distinguish master/slave PIC. In buffered mode, it is output line used to enable buffers
INT	Interrupt line, connected to INTR of microprocessor
INTA-bar	Interrupt ack, received active low from microprocessor
IR0-7	Asynchronous IRQ input lines, generated by peripherals.

Control logic :-it generates INT signal in response to this MP send INTA bar, it release 3 byte call instruction or one byte vectored addr

IIR :- Interrupt request –it is used to store all the pending request ,MP can read the content of this register by issuing appropriate command word

ISR : In service routine :- it is used to store all the interrupt levels currently being serviced each bit of this register is set by priority resolver and reset by end of interrupt command word ,MP can read the content of this register by issuing appropriate command word

Priority resolver : it determines the priority of IIR

IMR (interrupt mask register): -it is programmable register, it is used to mask out unwanted interrupt

Interrupt sequence (single PIC)

1. One or more of the IR lines goes high.
2. Corresponding IRR bit is set.
3. 8259 evaluates the request and sends INT to CPU.
4. CPU sends INTA-bar.
5. Highest priority ISR is set. IRR is reset.
6. 8259 releases CALL instruction on data bus.
7. CALL causes CPU to initiate two more INTA-bar's.
8. 8259 releases the subroutine address, first low byte then high byte.
9. ISR bit is reset depending on mode.

Single PIC System

085 microprocessor.

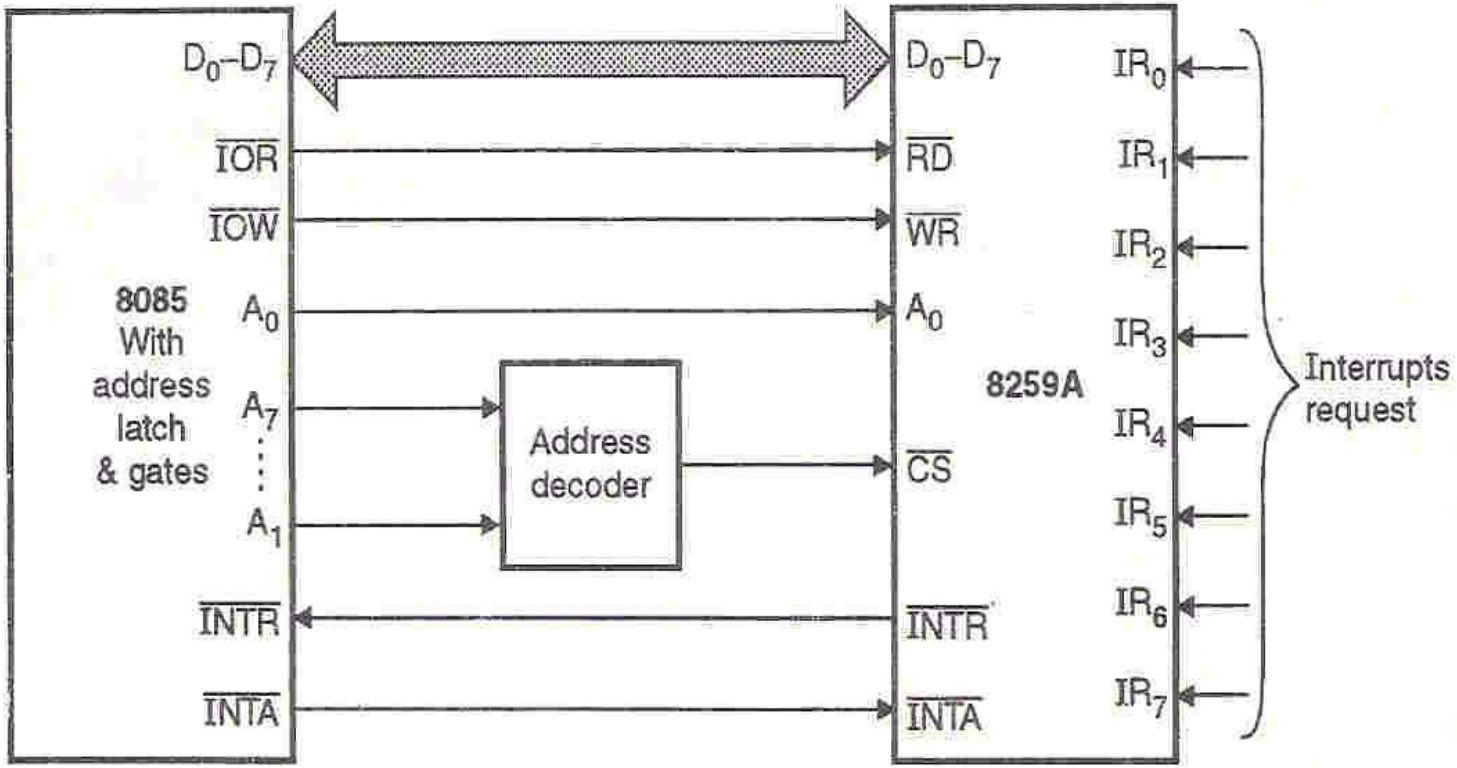


Fig. 10.29 : Single PIC system

NTERRUPT OPERATION ·

Cascaded PICs system (vectored mode)

- 8 pics may be cascaded to act as slave unit ot maser PIC thus total nos of 64 pics may be connected to MP

- In this system INT of salve PIC are connected to corresponding IR pin of master PIC

- Each PIC has its own addr know as slave identification nos

- 3 lines of CAS0 to CAS2 of master PIC are connected to slave pic

- SP of master is connected to Vcc and SP line of slave Pic is Grounded

- Each slave is identified by its slave identification nos through CAS 0 to CAS 2

Cascaded PIC

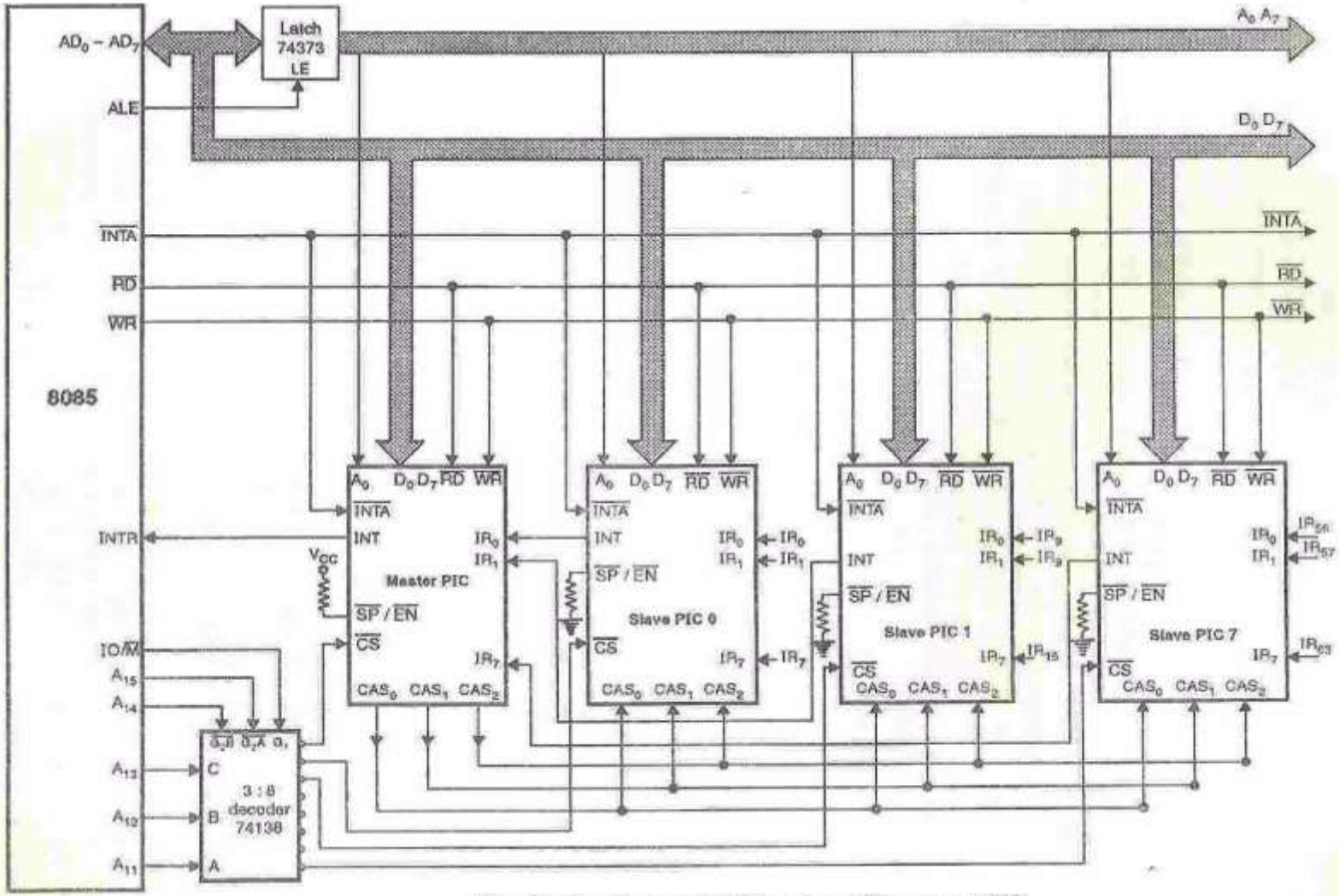


Fig. 10.30 : Cascaded PIC system I/O mapped I/O

Polled PIC system : -

- In this system Mp check status of each PIC
- Nos of PIC that can be connected with MP is very large
- The PPI activate one are more IR this sets the corresponding IIR
- MP issues a poll command to status of 8259
- The MP executes a I/O read cycle to read polled word from PIC
- MP decodes polled word and call subroutine
- If I = 0 the MP issue a poll command to another PIC in this way the MP checks status of all PIC

Polled PIC system

128 × 8 = 1024 peripherals). Fig. 10.31 shows 8259A interface in polled mode.

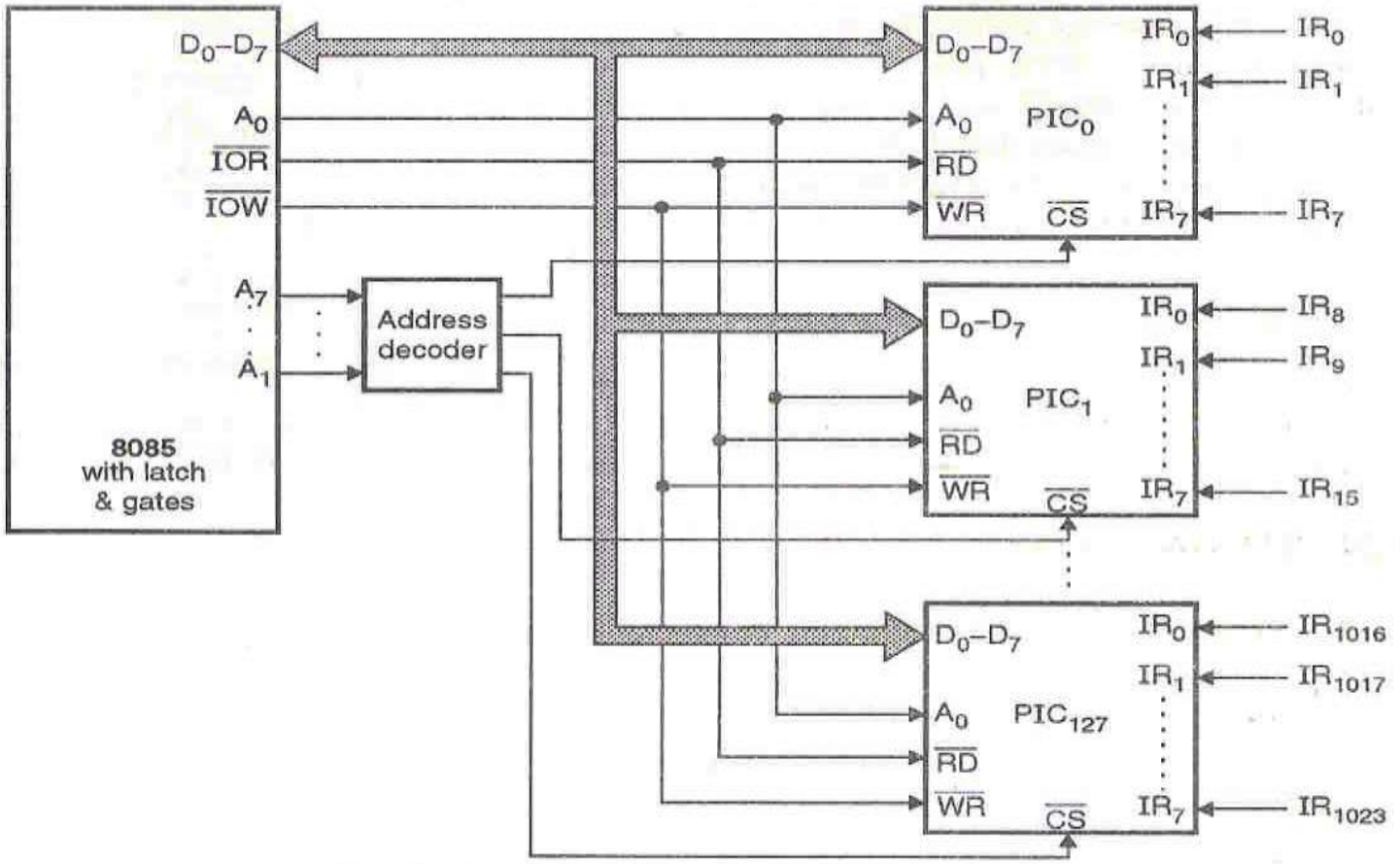
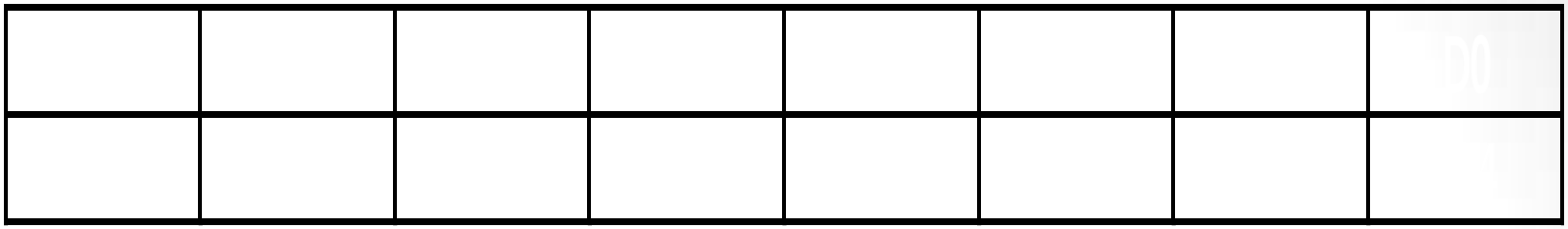


Fig. 10.31 : Polled PIC system (I/O mapped I/O)

Operation :

Initialization of Command Word [ICW 1]



IC4: - This indicate whether ICW4 is required or not(1 ICW 4 is issued , 0 ICW 4 is not issued)

SNGL: - indicate whether PIC is cascaded or not (1 single PIC , 0 cascaded PIC)

ADI : -it determine the spacing between successive interrupt routine (1 space between to ISR is 8 byte, 0 space between to ISR is 4 byte)

LTIM : - (If 1 all IR input level triggered,0 all IR input level triggered)

A5,A6,A7 : - this bits are used to provide 4 or 8 byte spacing between successive ISR (where the A0 to A$ is provided by 8259)

Initialization of Command Word [ICW 2]

							D0
							A8/0

- This bit is used to program higher byte of ISR addr in 8085
- Here A0 line should be 1
- Here D0 to D7 line used to program hihger addr bit of ISR addr
- Where D0 to D3 are provided by 8259 it self to choose IR0 to IR 7

Initialization of Command Word [ICW 3] master format

This is ICW 3 for master PIC here A0 is 1 means this ICW should be store in master PIC ,this tells the master PIC that which IR input is connected to slave

Initialization of Command Word [ICW 3] slave format

							D0
							ID0

- This CWR must be written in slave PIC
- Here ID2,ID1,ID0 bit are used to assign 3 bit slave identification number(000 to 111 for salve0 to slave7)

Initialization of Command Word [ICW 4]

- This CWR is used to initialize the 8259 in different modes
- [D0 = 1 8085 mode ,0 8086 mode][D1 =1normal EOI,0= AutoEOI]
- [D2 :- master/slave][D3 BUFFER mode][SFNM Special fully nested mode]

Operational command word [OCW 1]

•After initialization 8259 is ready to accept IR

•OCW can change the priority,modes of opration,controle and EOI command

•Here OCW 1 is used to mask out un wanted interrupt

Operational command word [OCW 2]

•Provide EOI which Clear appropriate ISR ,Rotate priority in normal and auto EOI mode

Operational command word [OCW 3] This command is used

•To operate 8259 in special mask mode ,polled mode

•To read IIR and ISR