Mikroişlemci Sistemleri

#9_2 – Kesmeler 2

YTÜ-CE

Ders-11 Konular

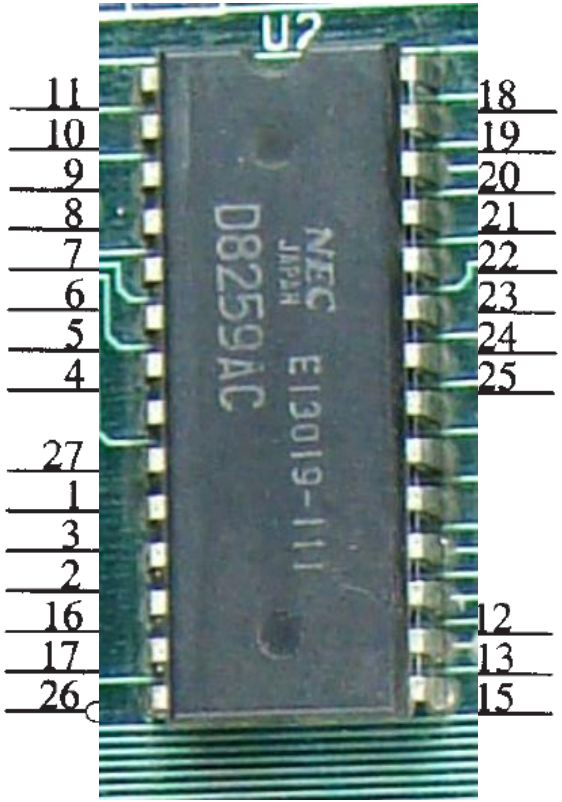
- 8259A
  - Uç Tanımları
  - İç Yapı
  - Ayarlama
    - ICWs
    - OCWs
- 8259A Örneği

8259A Programmable Interrupt Controller

- 8259A programlanabilir kesme kontrol devresidir.
- Kesme isteklerine öncelik atayabilir (priority encoding)
- Tek başına 8 farklı kesme vektörü sağlar
- Master (1)/Slave (8) formunda 64 farklı kesme vektörü sağlar

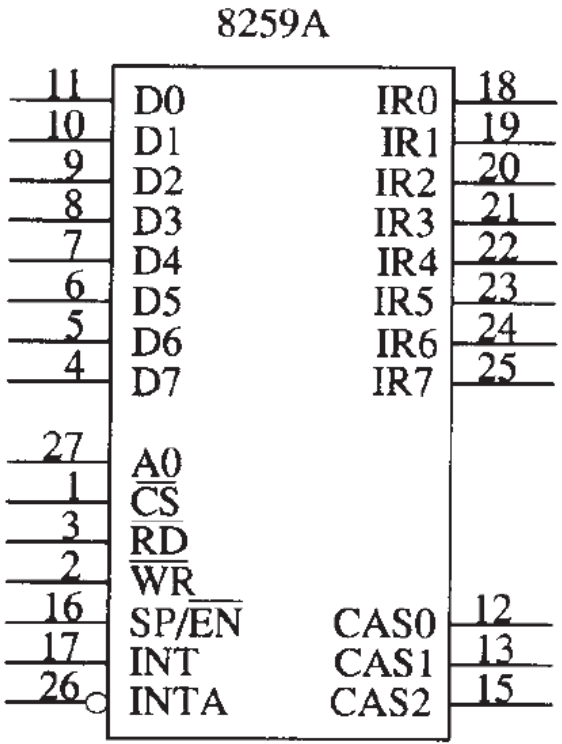
8259A PIC

- D0-D7: Data pinleri
- IR0-IR7: Kesme istek pinleri
- $\overline{WR}, \overline{RD}, \overline{CS}$
- INT: μP INTR ucuna
- $\overline{INTA}$: μP $\overline{INTA}$ ucundan
- A0: Adres ucu
- CAS2-CAS0: Kaskat seçim uçları
- $SP/\overline{EN}$: Slave program/enable buffer



8259A PIC

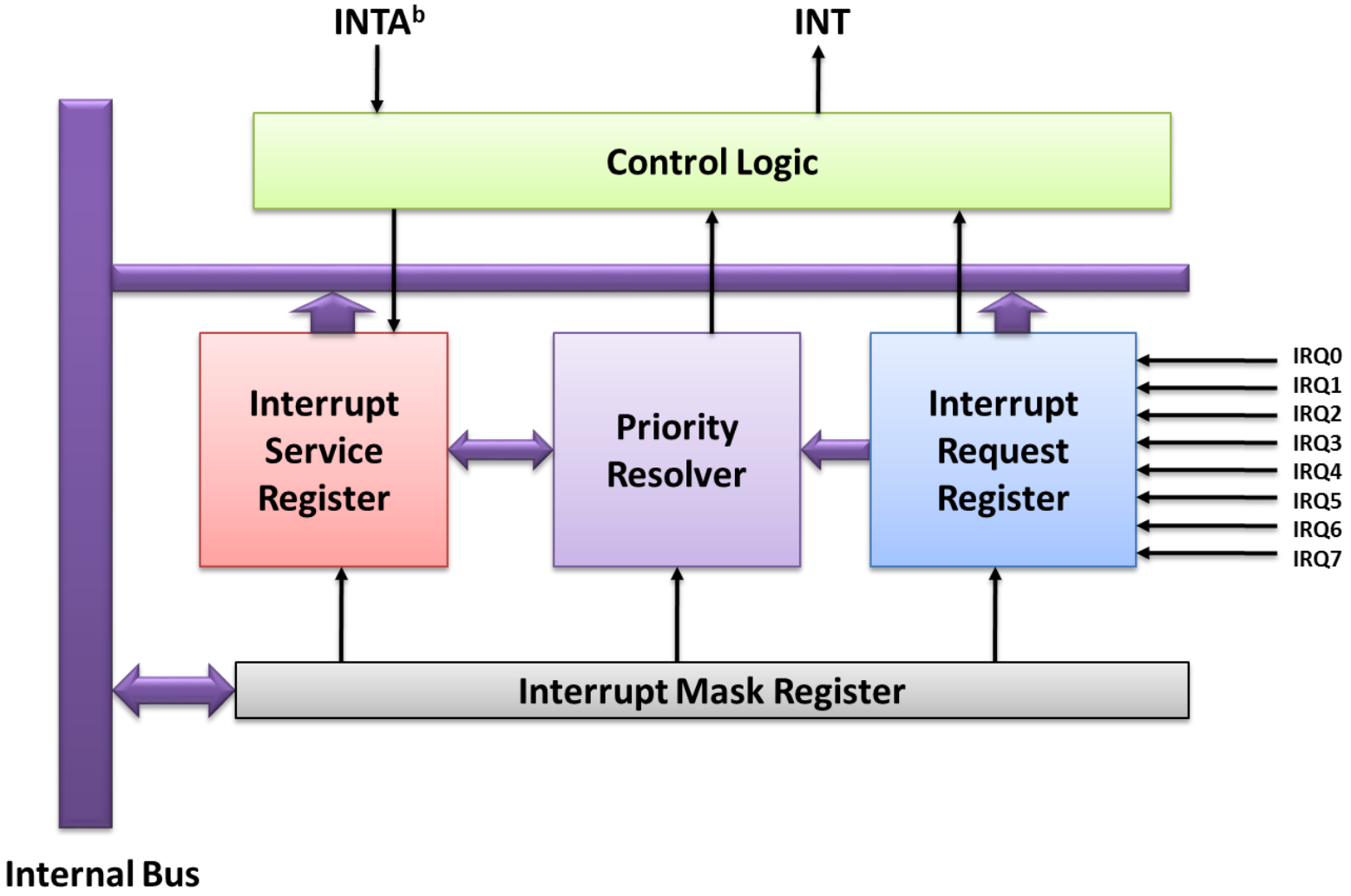
- D0-D7: Data pinleri
- IR0-IR7: Kesme istek pinleri
- $\overline{WR}, \overline{RD}, \overline{CS}$
- INT: μP INTR ucuna
- $\overline{INTA}$: μP $\overline{INTA}$ ucundan
- A0: Adres ucu
- CAS2-CAS0: Kaskat seçim uçları
- $SP/\overline{EN}$: Slave program/enable buffer



Kesme İşlemler Sıralaması

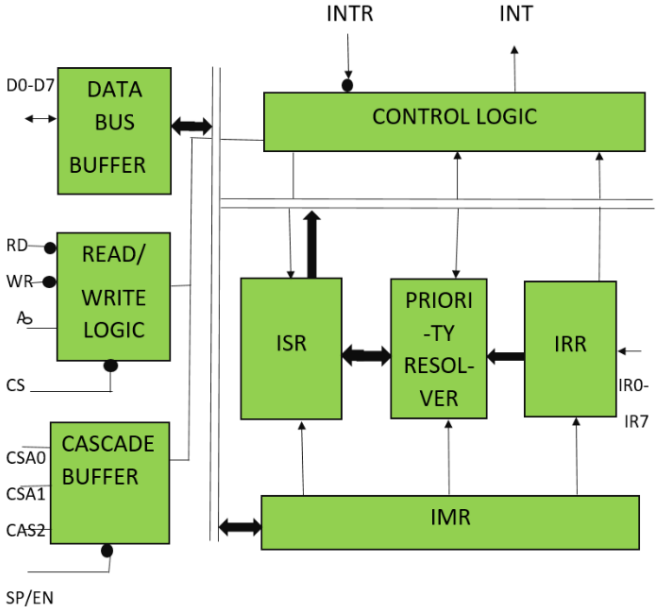
- IR7-0 uçlarından biri veya birden fazlası 1'e çıkar
- 8259 istekleri değerlendirip, CPU'ya INTR gönderir
- CPU, $\overline{INTA}$ ile karşılık verir
- $\overline{INTA}$ 8259A tarafına erişince, en yüksek öncelikli kesmeye ilişkin ISR biti 1, IRR biti 0 yapılır.
- CPU ikinci $\overline{INTA}$ darbesini göderir, 8259A karşılık olarak kesme vektör numarasını veriyoluna koyar.
- AEOI modunda ISR biti 0 yapılır (EOI modunda ise ISR bitini 0 yapmak için uygun bir komut beklenir.)

8259A İç Yapısı



1. **Data bus buffer –**

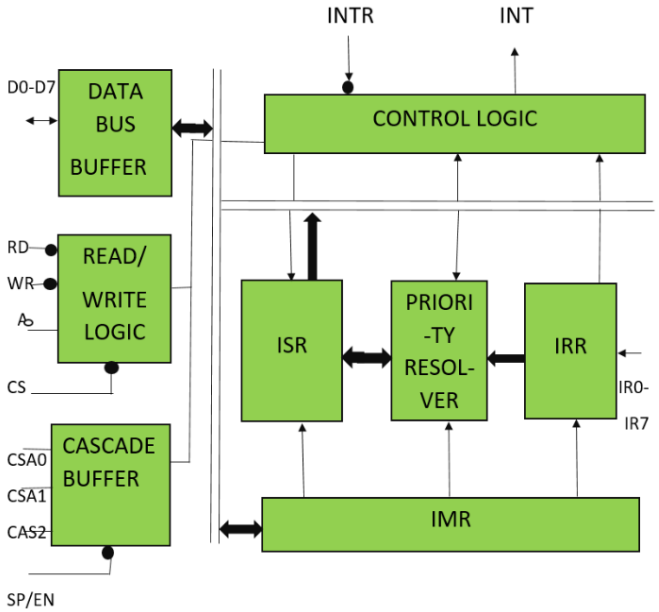
This Block is used as a mediator between 8259 and 8085/8086 microprocessor by acting as a buffer. It takes the control word from the 8085 (let say) microprocessor and transfer it to the control logic of 8259 microprocessor. Also, after selection of Interrupt by 8259 microprocessor, it transfer the opcode of the selected Interrupt and address of the Interrupt service sub routine to the other connected microprocessor. The data bus buffer consists of 8 bits represented as D0-D7 in the block diagram. Thus, shows that a maximum of 8 bits data can be transferred at a time.



ref: https://www.geeksforgeeks.org/8259-pic-microprocessor/

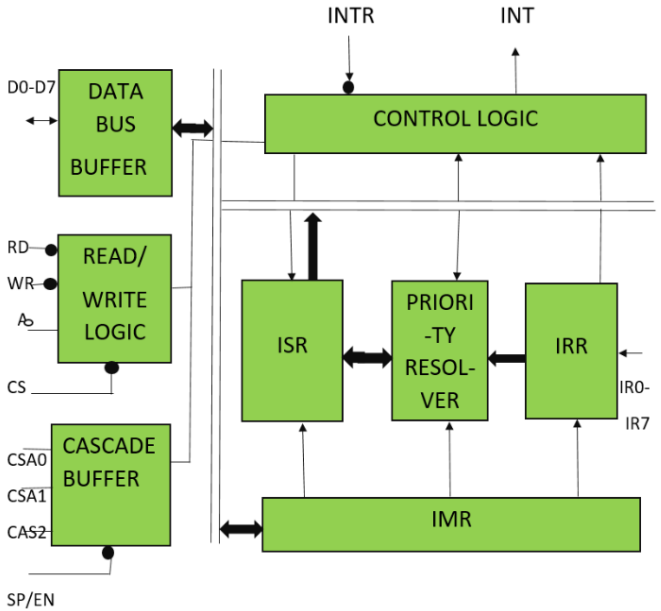
2. **Read/Write logic –**

This block works only when the value of pin CS is low (as this pin is active low). This block is responsible for the flow of data depending upon the inputs of RD and WR. These two pins are active low pins used for read and write operations.



3. **Control logic –**

It is the centre of the microprocessor and controls the functioning of every block. It has pin INTR which is connected with other microprocessor for taking interrupt request and pin INT for giving the output. If 8259 is enabled, and the other microprocessor Interrupt flag is high then this causes the value of the output INT pin high and in this way 8259 responds to the request made by other microprocessor.



4. **Interrupt request register (IRR) –**

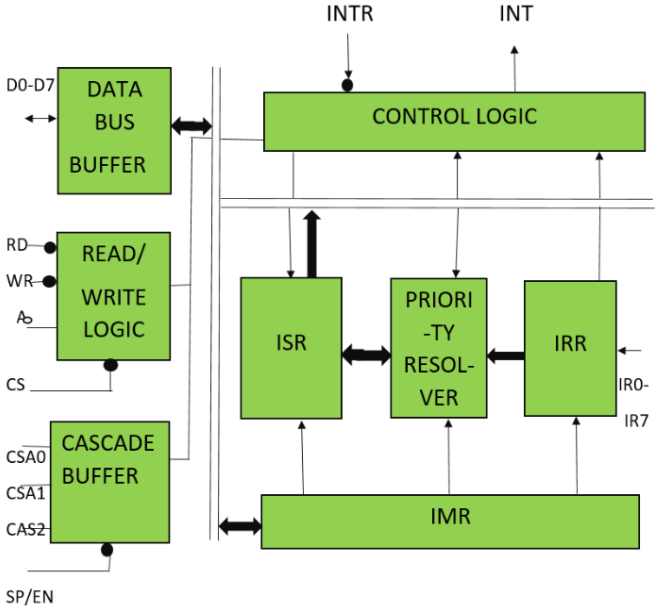
   It stores all the interrupt level which are requesting for Interrupt services.

5. **Interrupt service register (ISR) –**

   It stores the interrupt level which are currently being executed.

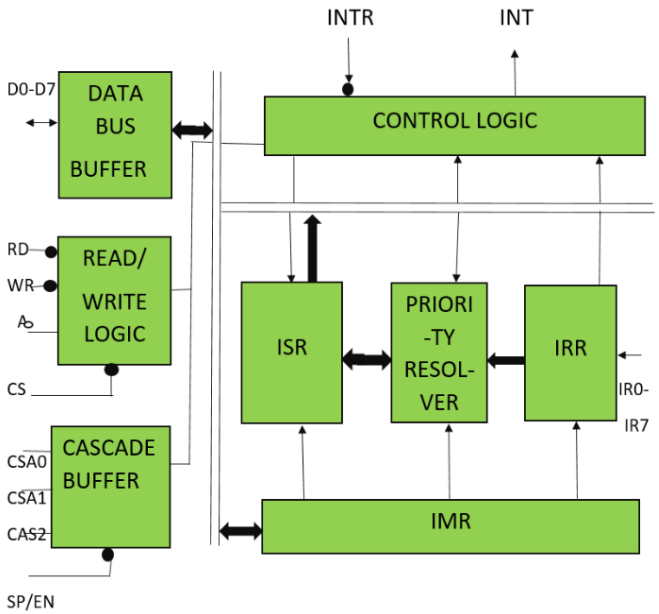
6. **Interrupt mask register (IMR) –**

   It stores the interrupt level which have to be masked by storing the masking bits of the interrupt level.



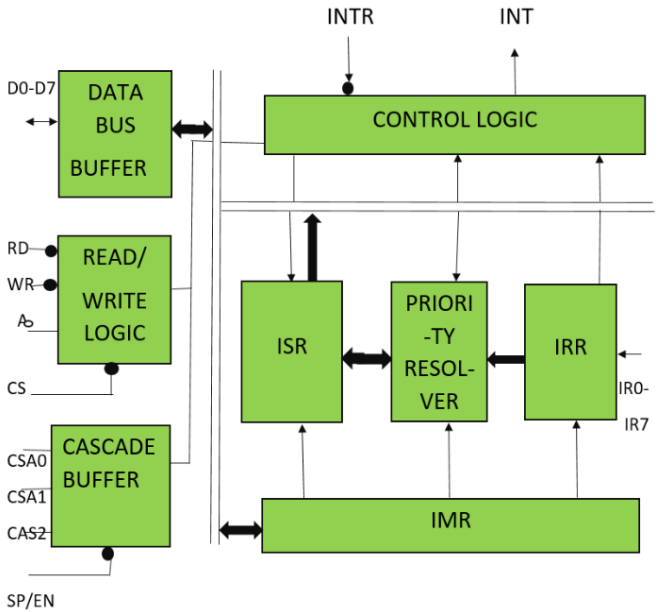
7. **Priority resolver –**

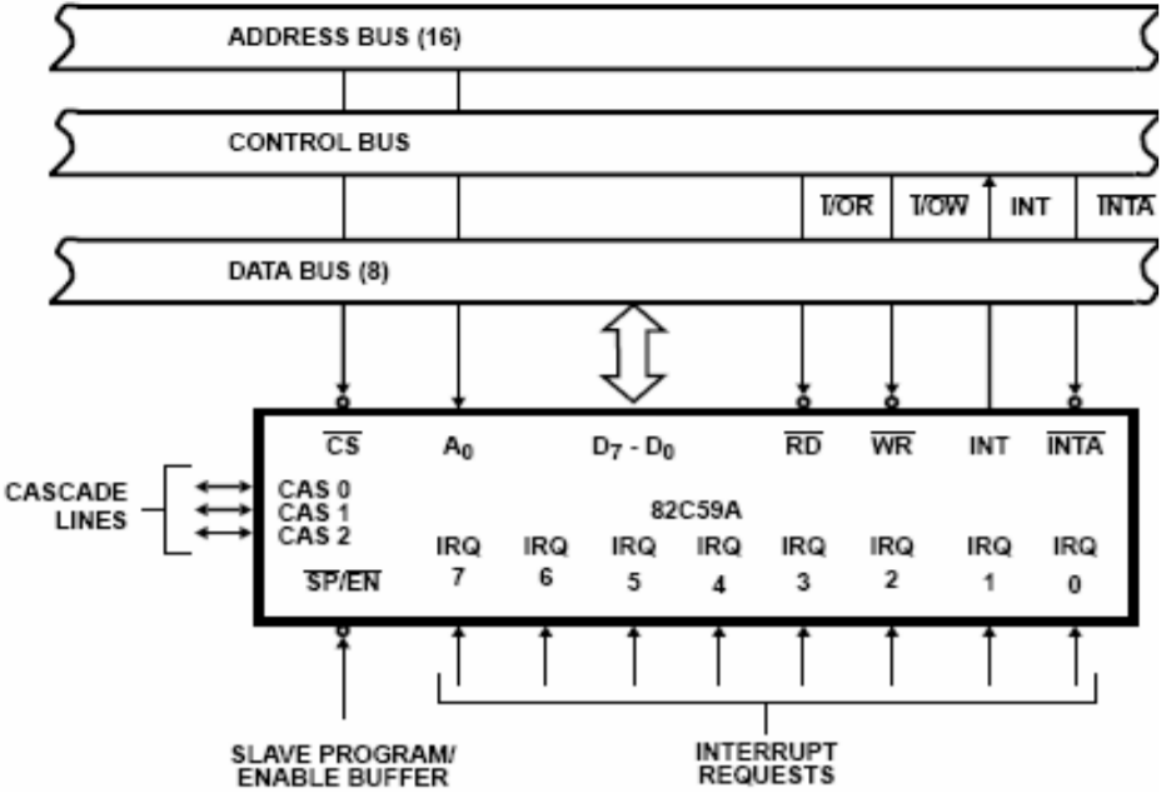
It examines all the three registers and set the priority of interrupts and according to the priority of the interrupts, interrupt with highest priority is set in ISR register. Also, it reset the interrupt level which is already been serviced in IRR.



8. **Cascade buffer –**

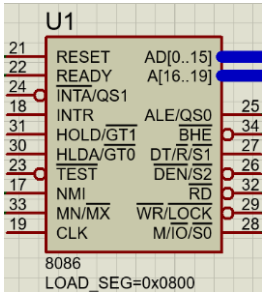
To increase the Interrupt handling capability, we can further cascade more number of pins by using cascade buffer. So, during increment of interrupt capability, CSA lines are used to control multiple interrupt structure.

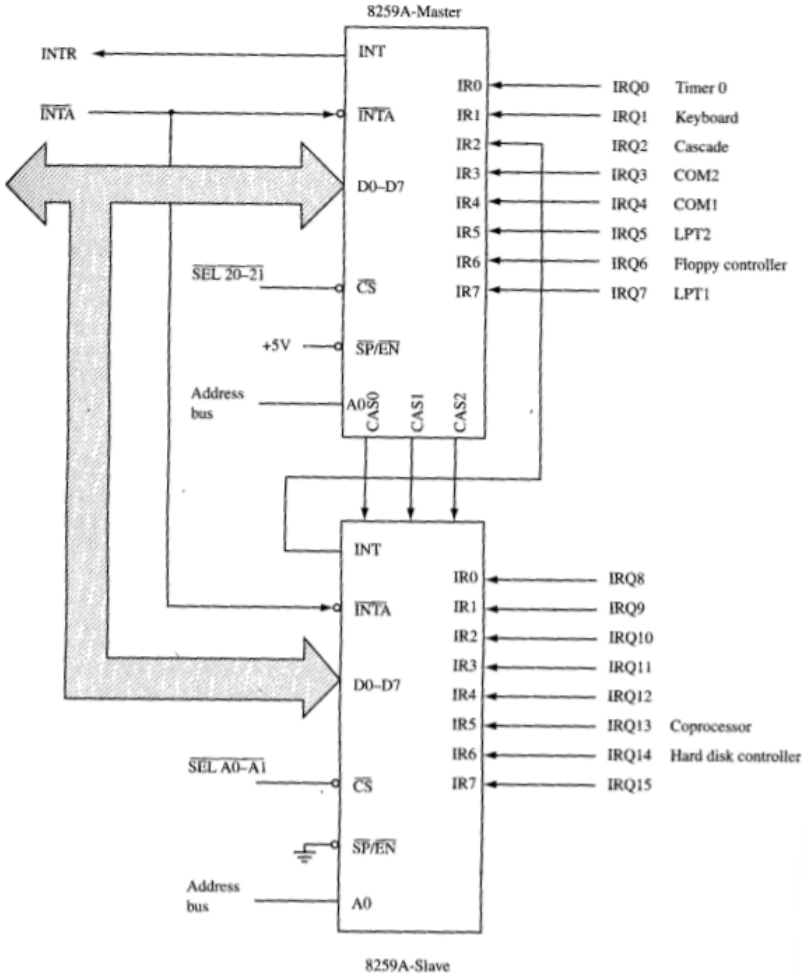




82C59A STANDARD SYSTEM BUS INTERFACE

ref: http://www.ee.hacettepe.edu.tr/~alkar/EE212/week12[2006].pdf

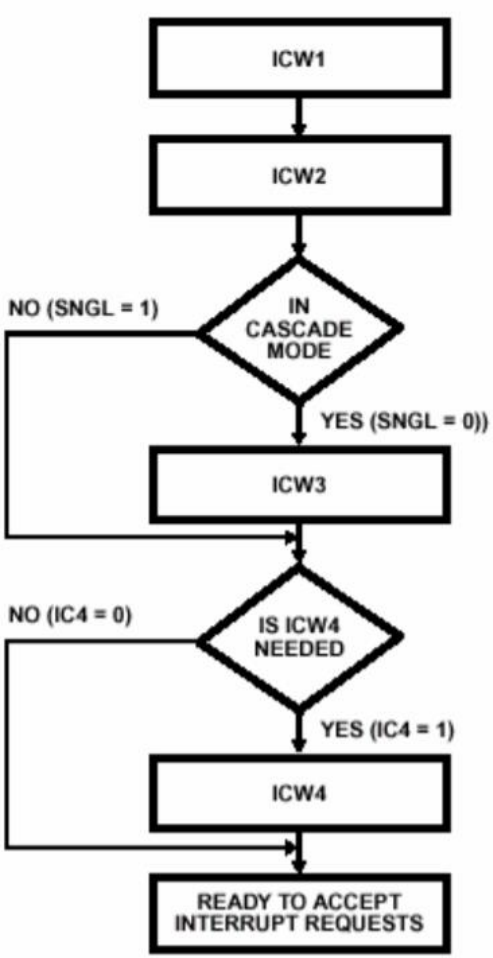

U1
21
22
24
18
31
30
23
17
33
19
RESET
READY
INTA/QS1
INTR
HOLD/GT1
HLDA/GT0
TEST
NMI
MN/MX
CLK
AD[0..15]
A[16..19]
ALE/QS0
BHE
DT/R/S1
DEN/S2
RD
WR/LOCK
M/IO/S0
25
34
27
26
32
29
28
8086
LOAD_SEG=0x0800


8259A-Master
INTR
INT
INTA
INTA
D0–D7
SEL 20–21
CS
+5V
SP/EN
Address bus
A0
CAS0
CAS1
CAS2
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
IRQ0 Timer 0
IRQ1 Keyboard
IRQ2 Cascade
IRQ3 COM2
IRQ4 COM1
IRQ5 LPT2
IRQ6 Floppy controller
IRQ7 LPT1
INT
INTA
D0–D7
SEL A0–A1
CS
SP/EN
Address bus
A0
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
IRQ8
IRQ9
IRQ10
IRQ11
IRQ12
IRQ13 Coprocessor
IRQ14 Hard disk controller
IRQ15
8259A-Slave

8259A PIC

- 8259A basit mikroişlemcili sistemlerde tek olarak kullanılacaksa $SP/\overline{EN}$ ucu 1 olarak ayarlanır (master)
- 8259A, Initialization command words (ICWs) ve Operation command words (OCWs) kullanılarak ayarlanır

Means, in non-buffered mode, the $\overline{SP}/\overline{EN}$ pin of an 8259 is used to specify whether the 8259 is to operate as a master or as a slave, and in the buffered mode, the $\overline{SP}/\overline{EN}$ pin is used as an output to enable the data bus buffer of the system.



Two types of command words are provided to program the 8259:

1) The initialization command words (ICW)

2) The operational command words (OCW)

Fully Nested Mode is entered.

ICW3 and ICW4 are optional

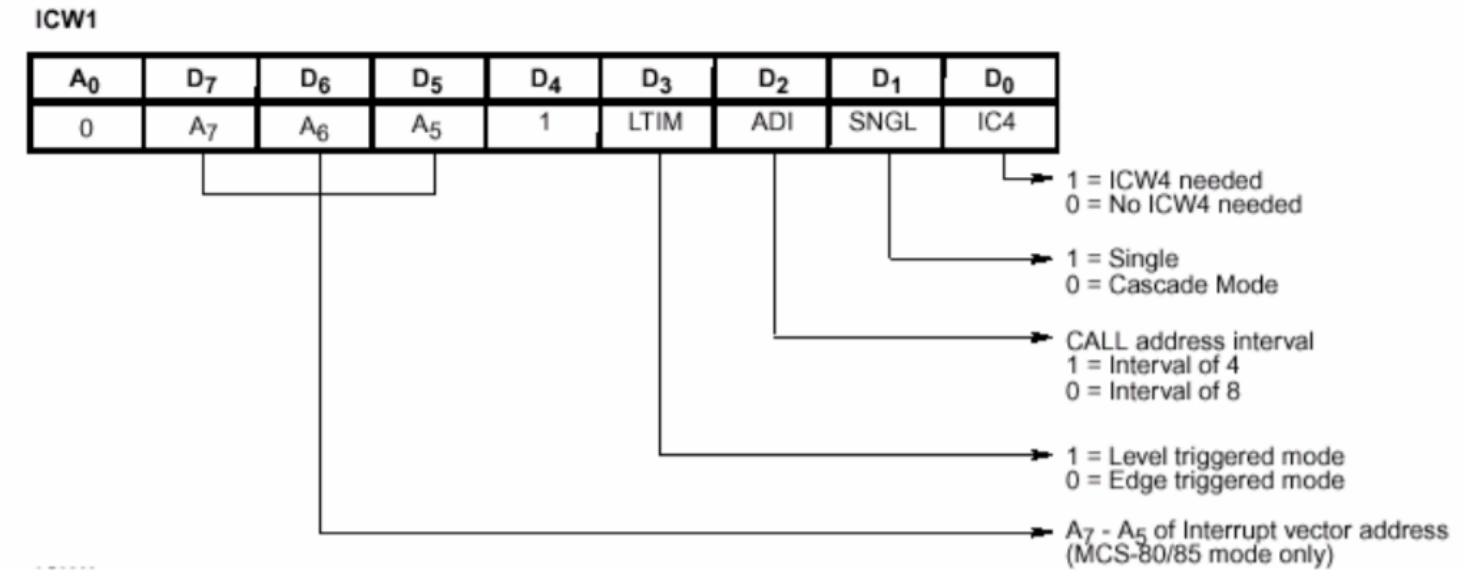
It is not possible to modify just one ICW. Whole ICW sequence must be repeated

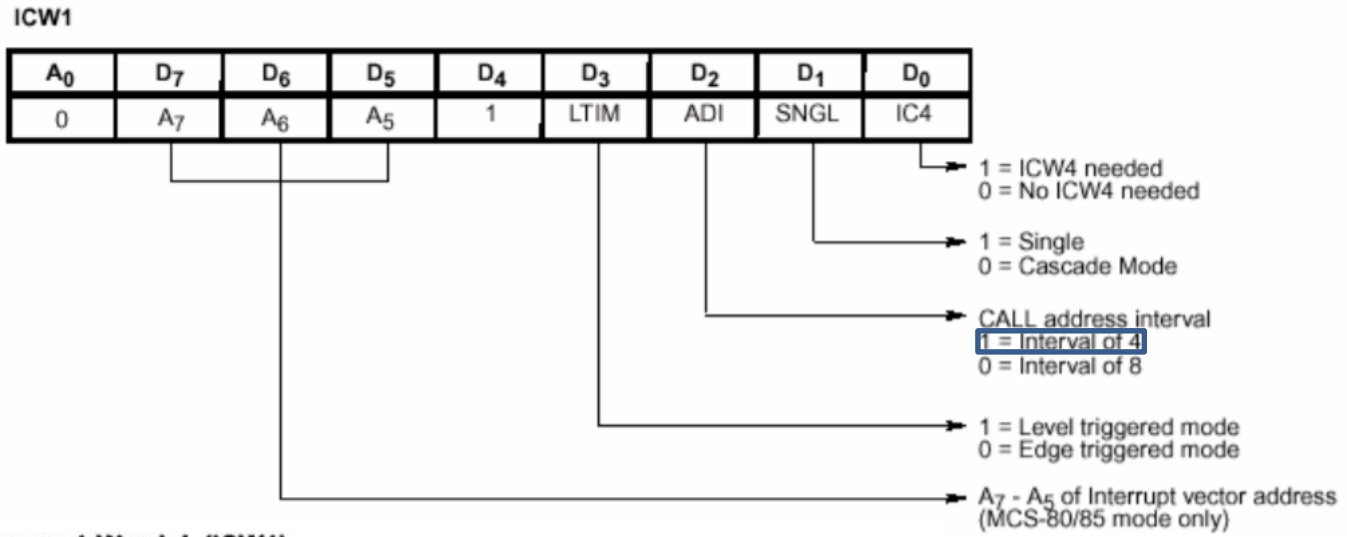
ICW1

AD0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	LTIM	0	SGNL	IC4
	0 for x86				1 for Level Trigger 0 for Edge Trigger		1=single 0=Cascade	1=IC4 needed 0=no IC4 needed

ICW1

AD0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	LTIM	0	SGNL	IC4
	0 for x86				1 for Level Trigger 0 for Edge Trigger		1=single 0=Cascade	1=IC4 needed 0=no IC4 needed





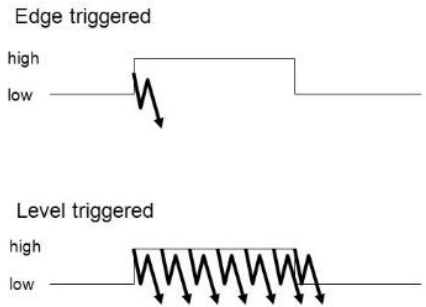
Initialization Command Word 1 (ICW1)

Fig. 8.8 shows the Initialization Command Word 1 (ICW1).

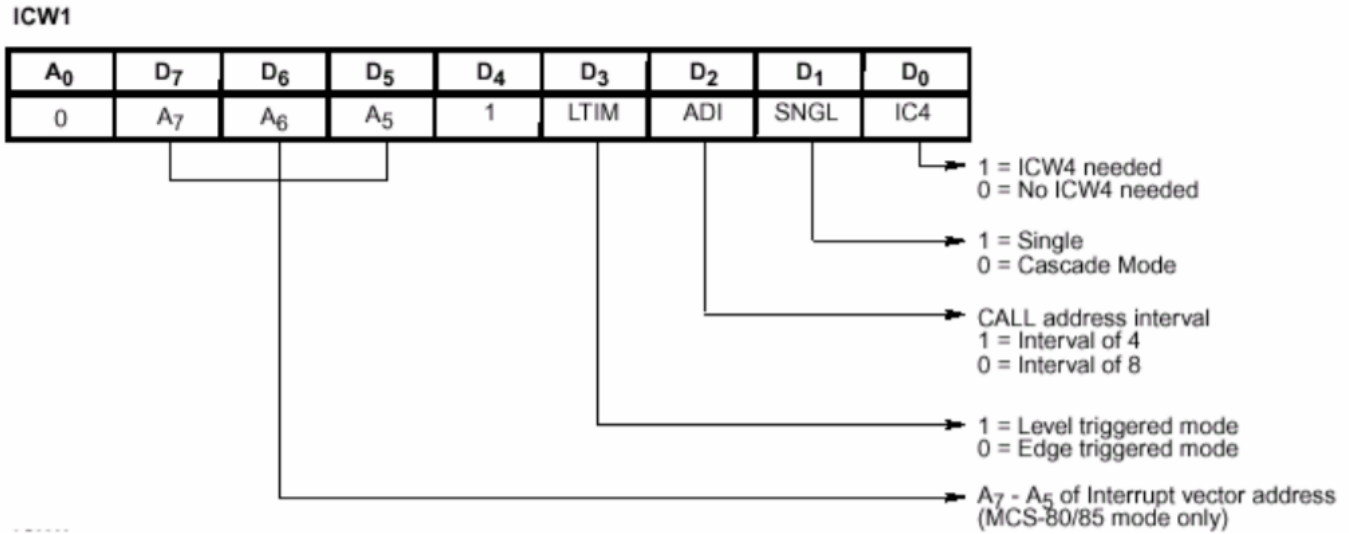
A write command issued to the 8259 with $A_0 = 0$ and $D_4 = 1$ is interpreted as ICW1, which starts the initialization sequence.

It specifies

1. Single or multiple 8259As in the system.
2. 4 or 8 bit interval between the interrupt vector locations.
3. The address bits A_7 - A_5 of the CALL instruction.
4. Edge triggered or level triggered interrupts.
5. ICW4 is needed or not.

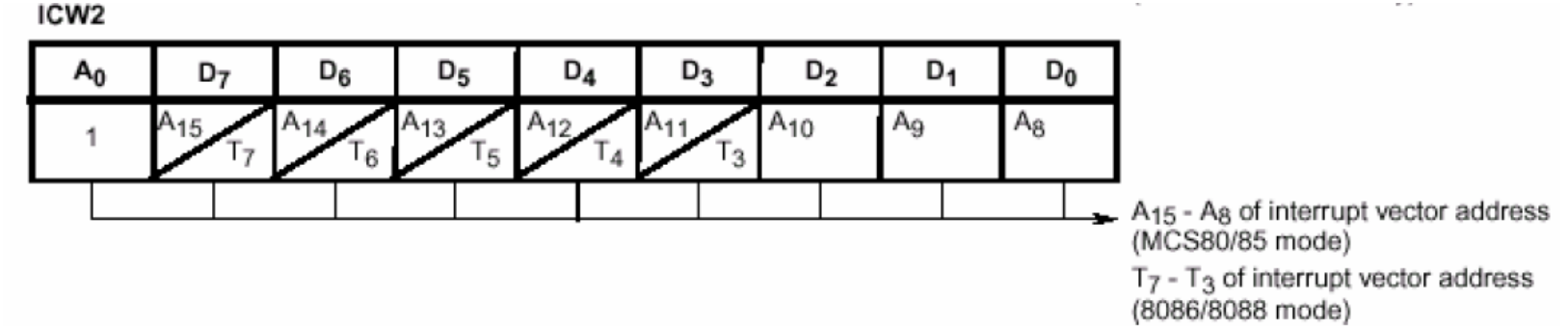


- Edge triggered interrupts signal as a one shot event!
- Level triggered interrupts are signaled as long as line is raised



What value should be written to ICW1 in order to configure the 8259 so that ICW4 needed, the system is going to use multiple 8259s and its inputs are level sensitive?

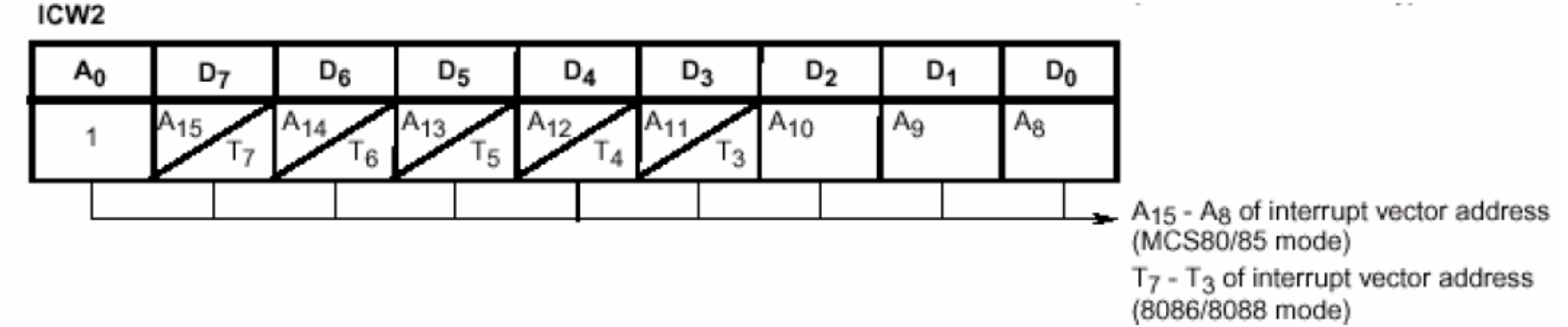
ICW2



AD0	D7	D6	D5	D4	D3	D2	D1	D0
1	T7	T6	T5	T4	T3	T2	T1	T0
	T7=T0 is the assign to IR0, Vector address for ISR							

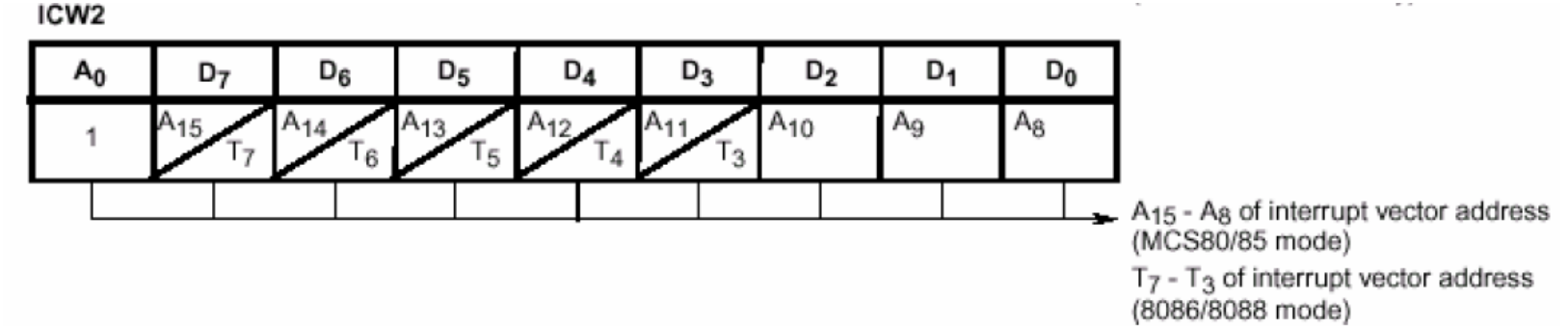
A write command following ICW1, with A0 = 1 is interpreted as ICW2. This is used to load the high order byte of the interrupt vector address of all the interrupts.

ICW2



What should be programmed into register ICW2 if type number output on the bus is to range from F0h to F7h

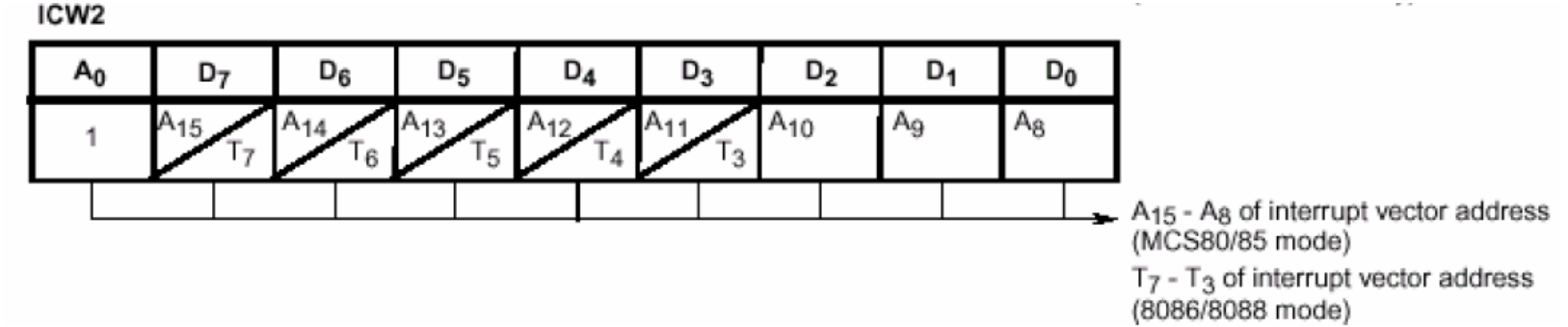
ICW2



What should be programmed into register ICW2 if type number output on the bus is to range from F0h to F7h

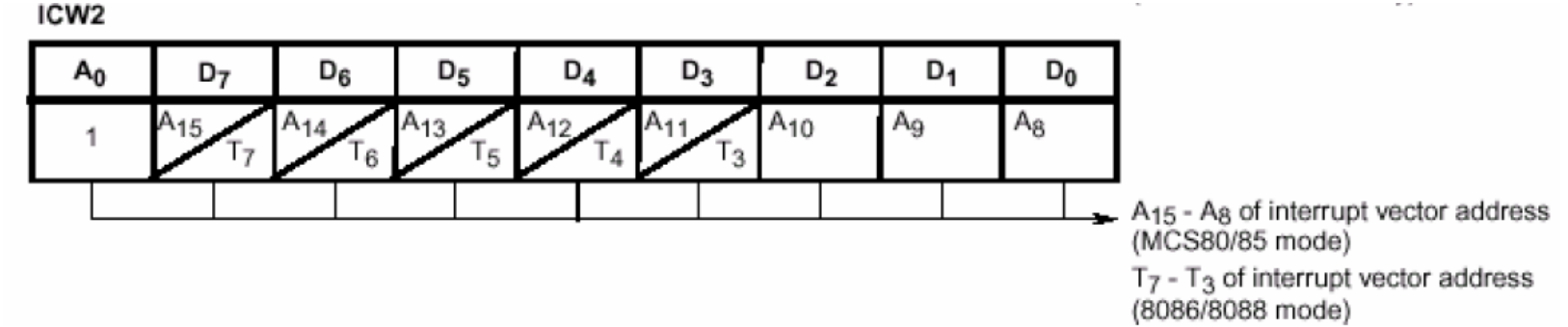
11110000b = F0h

ICW2



Suppose IR6 is set to generate the value of 6E. Generate the addresses for the other interrupts.

ICW2



Suppose IR6 is set to generate the value of 6E. Generate the addresses for the other interrupts.

IR7 = 6F
IR6 = 6E
IR5 = 6D
IR4 = 6C
IR3 = 6B
IR2 = 6A
IR1 = 69
IR0 = 68

ICW3 (MASTER DEVICE)

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0

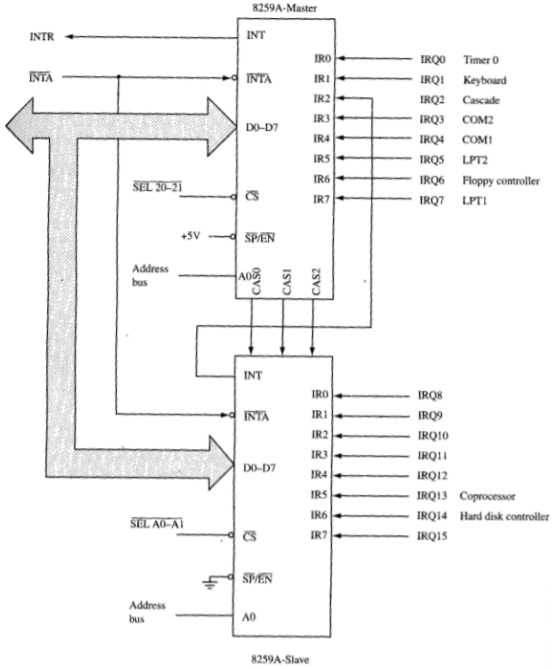
1 = IR input has a slave
0 = IR input does not have a slave

ICW3 (SLAVE DEVICE)

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	ID_2	ID_1	ID_0

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



Initialization Command Word 3 (ICW3)

ICW3 is required only if there is more than one 8259 in the system and if they are cascaded. An ICW3 operation loads a slave register in the 8259. The format of the byte to be loaded as an ICW3 for a master 8259 or a slave is shown in the Fig. 8.10. For master, each bit in ICW3 is used to specify whether it has a slave 8259 attached to it on its corresponding IR (Interrupt Request) input. For slave, bits D_0-D_2 of ICW3 are used to assign a slave identification code (slave ID) to the 8259.

ICW3 (MASTER DEVICE)

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0

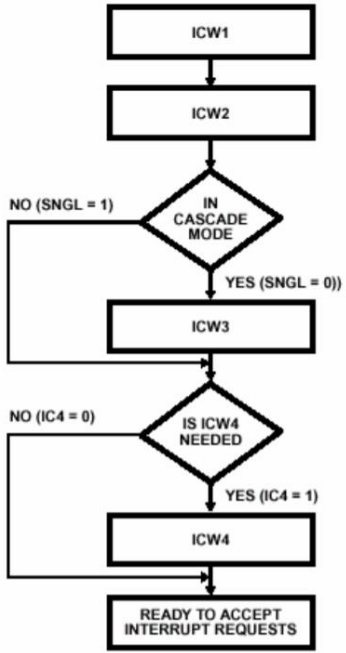
1 = IR input has a slave
0 = IR input does not have a slave

ICW3 (SLAVE DEVICE)

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	ID_2	ID_1	ID_0

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



Initialization Command Word 3 (ICW3)

ICW3 is required only if there is more than one 8259 in the system and if they are cascaded. An ICW3 operation loads a slave register in the 8259. The format of the byte to be loaded as an ICW3 for a master 8259 or a slave is shown in the Fig. 8.10. For master, each bit in ICW3 is used to specify whether it has a slave 8259 attached to it on its corresponding IR (Interrupt Request) input. For slave, bits D_0-D_2 of ICW3 are used to assign a slave identification code (slave ID) to the 8259.

ICW4 **It is loaded only if the D_0 bit of ICW1 (IC 4) is set.**

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	SFNM	BUF	M/S	AEOI	µPM

1 = 8086/8088 mode
0 = MCS-80/85 mode

1 = Auto EOI
0 = Normal EOI

0	X	- Non buffered mode
1	0	- Buffered mode slave
1	1	- Buffered mode master

1 = Special fully nested moded
0 = Not special fully nested mode

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.

ICW4

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	SFNM	BUF	M/S	AEOI	µPM

It specifies.

1) Whether to use special fully nested mode or non special fully nested mode.
2) Whether to use buffered mode or non buffered mode.
3) Whether to use Automatic EOI or Normal EOI
4) CPU used, 8086/8088 or 80810.

End of Interrupt (EOI)

The IS bit can be reset by an End of Interrupt command issued by the CPU, usually just before exiting from the interrupt routine.

Automatic End of Interrupt (AEOI)

If the AEOI mode is set, the 8259 will perform a non-specific EOI on its own on the trailing edge of the third $\overline{\text{INTA}}$ pulse. The AEOI mode can only be used for a master 8259 and not for a slave.

Fully nested mode:

- This is a general purpose mode where all IR's are arranged in highest to lowest.
- IR0 highest and IR7 lowest.

Special Fully Nested Mode:

- Used in more complicated systems.
- Similar to, normal nested mode.
- When an interrupt request from a certain slave is in service, this slave can further send requests to the master.
- The master interrupts the CPU only.

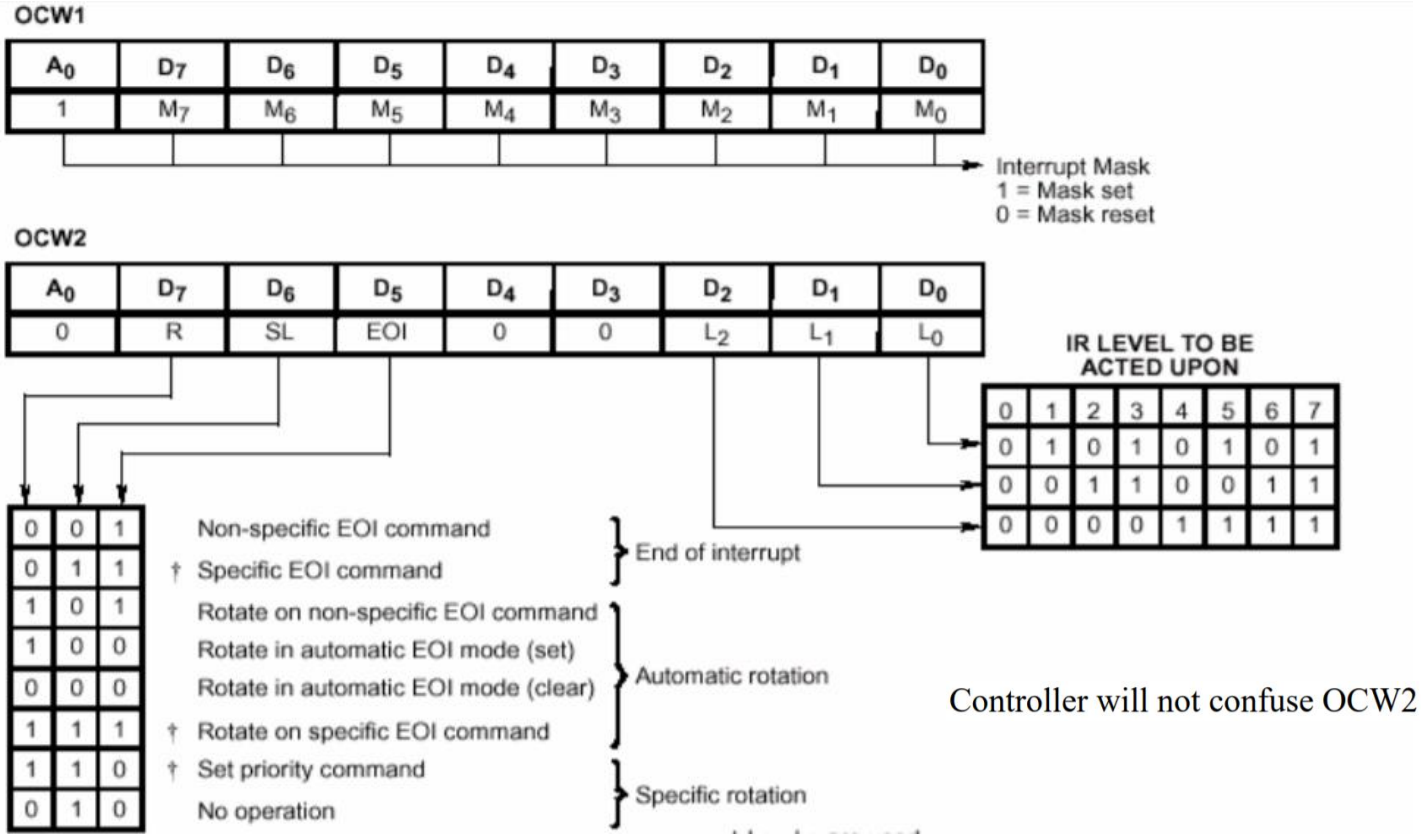
Operation Command Word 1 (OCW1)

A Write command to the 8259 with $A_0 = 1$ (after ICW2) is interpreted as OCW1. OCW1 is used for enabling or disabling the recognition of specific interrupt requests by programming the IMR.

Operation Command Word 2 (OCW2)

A Write command with $A_0 = 1$ and $D_4\ D_3 = 00$ is interpreted as OCW2. The R(Rotate), SL (Select-Level), EOI bits control the Rotate and End Of Interrupt Modes and combinations of the two. Fig. 8.13 shows the Operation Command Word format. $L_2 - L_0$ are used to specify the interrupt level to be acted upon when the SL bit is active.

OCW1 is used to access the contents of the IMR. A READ operation can be performed to the IMR to determine the present setting of the mask. Write operations can be performed to mask or unmask certain bits.



Controller will not confuse OCW2 with ICW1 since D4 = 1

AUTOMATIC ROTATION MODE:

- In this mode, a device after being serviced, receives the lowest priority.

IR_0	IR_1	IR_2	IR_3	IR_4	IR_5	IR_6	IR_7
4	5	6	7	0	1	2	3

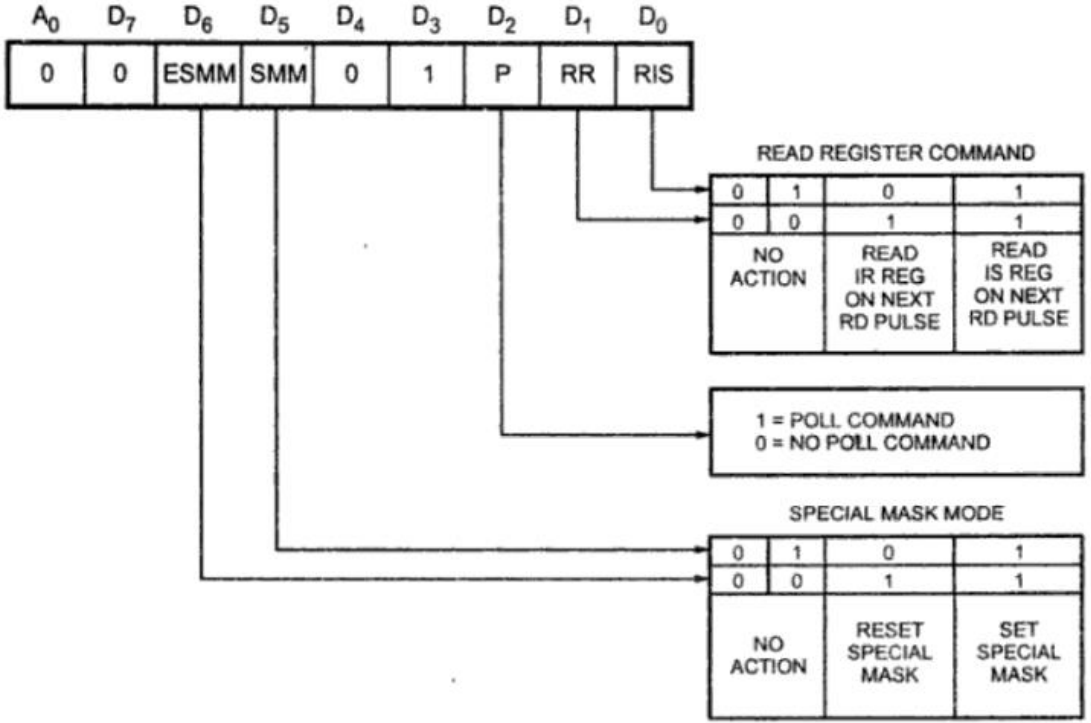
SPECIFIC ROTATION MODE:

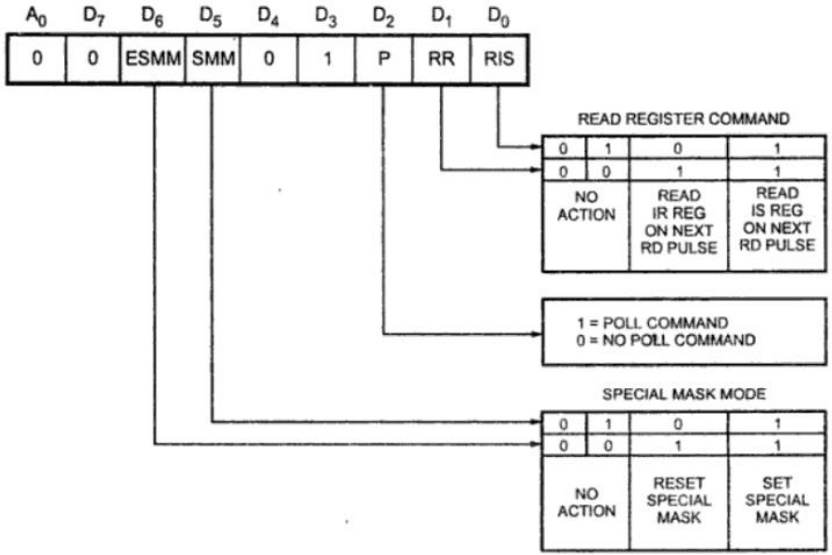
- Similar to automatic rotation mode, except that the user can select any IR for the lowest priority, thus fixing all other priorities.

IR_0	IR_1	IR_2	IR_3	IR_4	IR_5	IR_6	IR_7
5	6	7	0	1	2	3	4

Operation Command Word 3 (OCW3)

OCW3 is used to read the status of the registers, and to set or reset the Special Mask and Polled modes. Fig. 8.14 shows format operational command word 3.





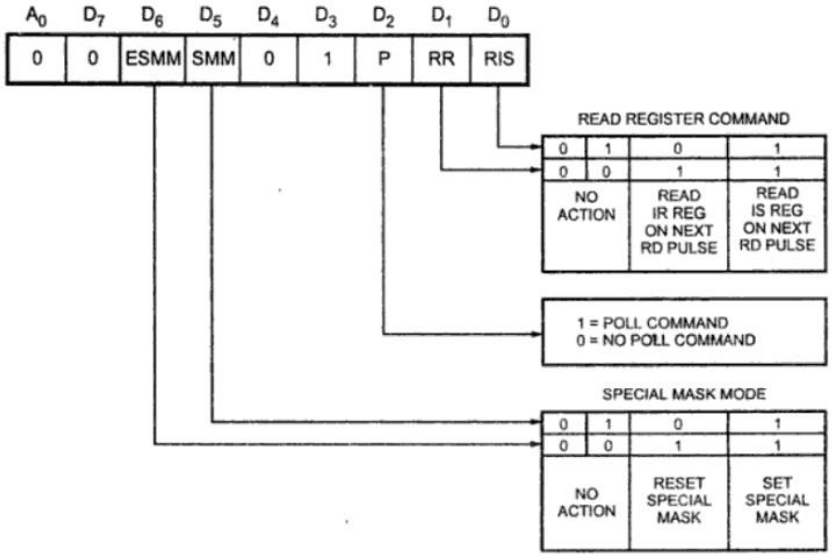
8259 Status Read Operations

The status of the Interrupt Request Register, the In-Service Register, and the Interrupt Mask Register of the 8259 may be read by issuing appropriate Read commands as described below.

IRR Status Read

An OCW3 with RR (Read Register) = 1 and RIS (Read ISR) = 0 set up the 8259 for a status read of the Interrupt Request Register.

When the 8259 is not in the Polled mode, after it is sets up for an IRR status read operation, all Read commands with $A_0 = 1$ cause the 8259 to send the IRR status word.

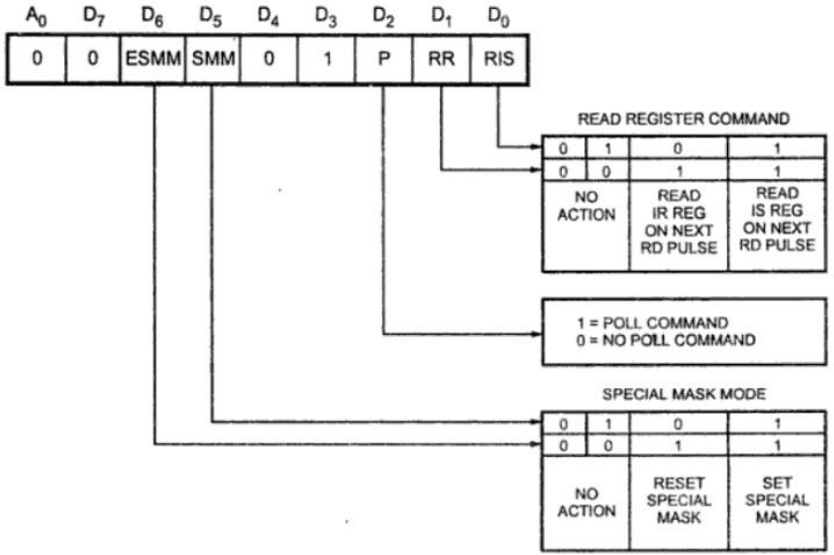


ISR Status Read

An OCW3 with RR = 1 and RIS = 1 sets up the 8259 for a status read of the In-Service Register. A subsequent read command issued to the 8259 will cause the 8259 to send the contents of the ISR onto the data bus.

IMR Status Read

A Read command issued to the 8259 with $A_0 = 1$ (with $\overline{RD}$, $\overline{CS} = 0$) causes the 8259 to put out the contents of the Interrupt Mask Register. OCW3 is not required for a status read of the IMR.



e) Poll Mode :

In this mode the INT output is not used. The microprocessor checks the status of interrupt requests by issuing poll command. The microprocessor reads contents of 8259A after issuing poll command. During this read operation the 8259A provides polled word and sets ISR bit of highest priority active interrupt request FORMAT.

I	X	X	X	X	W_2	W_1	W_0

$I = 1 \rightarrow$ One or more interrupt requests activated.

$I = 0 \rightarrow$ No interrupt request activated.

$W_2\ W_1\ W_0 \rightarrow$ Binary code of highest priority active interrupt request.

OCW1 & OCW2 & OCW3

AD0	D7	D6	D5	D4	D3	D2	D1	D0
1	M7	M6	M5	M4	M3	M2	M1	M0
	Mi: Interrupt mask, 1=mask set, 0=mask reset							

AD0	D7	D6	D5	D4	D3	D2	D1	D0
0	R	SL	EOI	0	0	L2	L1	L0
	Rotate	Specific	EOI			IR Level to be acted Upon (0-7)		

AD0	D7	D6	D5	D4	D3	D2	D1	D0
0	0	ESMM	SMM	0	1	P	RR	RIS
		Special Mask Mode				Poll Command	Register Read Mode	

8259A Ayarlama – ICW'ler

CS	A0	Initialization
0	0	ICW1
0	1	ICW2,ICW3,ICW4
1	X	Not Address

8259A Ayarlama – OCW'ler

CS	A0	Operation Command Word
0	0	OCW2, OCW3
0	1	OCW1
1	X	Not Address

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

2. The register that stores all the interrupt requests in it in order to serve them one by one on a priority basis is
a) Interrupt Request Register
b) In-Service Register
c) Priority resolver
d) Interrupt Mask Register

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

2. The register that stores all the interrupt requests in it in order to serve them one by one on a priority basis is

a) Interrupt Request Register

b) In-Service Register

c) Priority resolver

d) Interrupt Mask Register

View Answer

Answer: a

Explanation: The interrupts at IRQ input lines are handled by Interrupt Request Register internally.

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

3. The register that stores the bits required to mask the interrupt inputs is
a) In-service register
b) Priority resolver
c) Interrupt Mask register
d) None

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

3. The register that stores the bits required to mask the interrupt inputs is
a) In-service register
b) Priority resolver
c) Interrupt Mask register
d) None
View Answer

Answer: c
Explanation: Also, Interrupt Mask Register operates on IRR(Interrupt Request Register) at the direction of the Priority Resolver.

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

4. The interrupt control logic
a) manages interrupts
b) manages interrupt acknowledge signals
c) accepts interrupt acknowledge signal
d) all of the mentioned

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

4. The interrupt control logic
a) manages interrupts
b) manages interrupt acknowledge signals
c) accepts interrupt acknowledge signal
d) all of the mentioned
View Answer

Answer: d
Explanation: The interrupt control logic performs all the operations that are involved within the interrupts like accepting and managing interrupt acknowledge signals, interrupts.

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

5. In a cascaded mode, the number of vectored interrupts provided by 8259A is

a) 4

b) 8

c) 16

d) 64

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

5. In a cascaded mode, the number of vectored interrupts provided by 8259A is

a) 4

b) 8

c) 16

d) 64

View Answer

Answer: d

Explanation: A single 8259A provides 8 vectored interrupts. In cascade mode, 64 vectored interrupts can be provided.

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

6. When the PS(active low)/EN(active low) pin of 8259A used in buffered mode, then it can be used as a
a) input to designate chip is master or slave
b) buffer enable
c) buffer disable
d) none

ref:

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

6. When the PS(active low)/EN(active low) pin of 8259A used in buffered mode, then it can be used as a
a) input to designate chip is master or slave
b) buffer enable
c) buffer disable
d) none

View Answer

Answer: b
Explanation: When the pin is used in buffered mode, then it can be used as a buffer enable to control buffer transreceivers. If it is not used in buffered mode, then the pin is used as input to designate whether the chip is used as a master or a slave.

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

9. In the application where all the interrupting devices are of equal priority, the mode used is
a) Automatic rotation
b) Automatic EOI mode
c) Specific rotation
d) EOI

Microprocessors Questions and Answers – Programmable Interrupt Controller 8259A

9. In the application where all the interrupting devices are of equal priority, the mode used is
a) Automatic rotation
b) Automatic EOI mode
c) Specific rotation
d) EOI

View Answer

Answer: a
Explanation: The automatic rotation is used in the applications where all the interrupting devices are of equal priority.

ref: https://www.sanfoundry.com/microprocessors-mcqs-programmable-interrupt-controller-8259a/

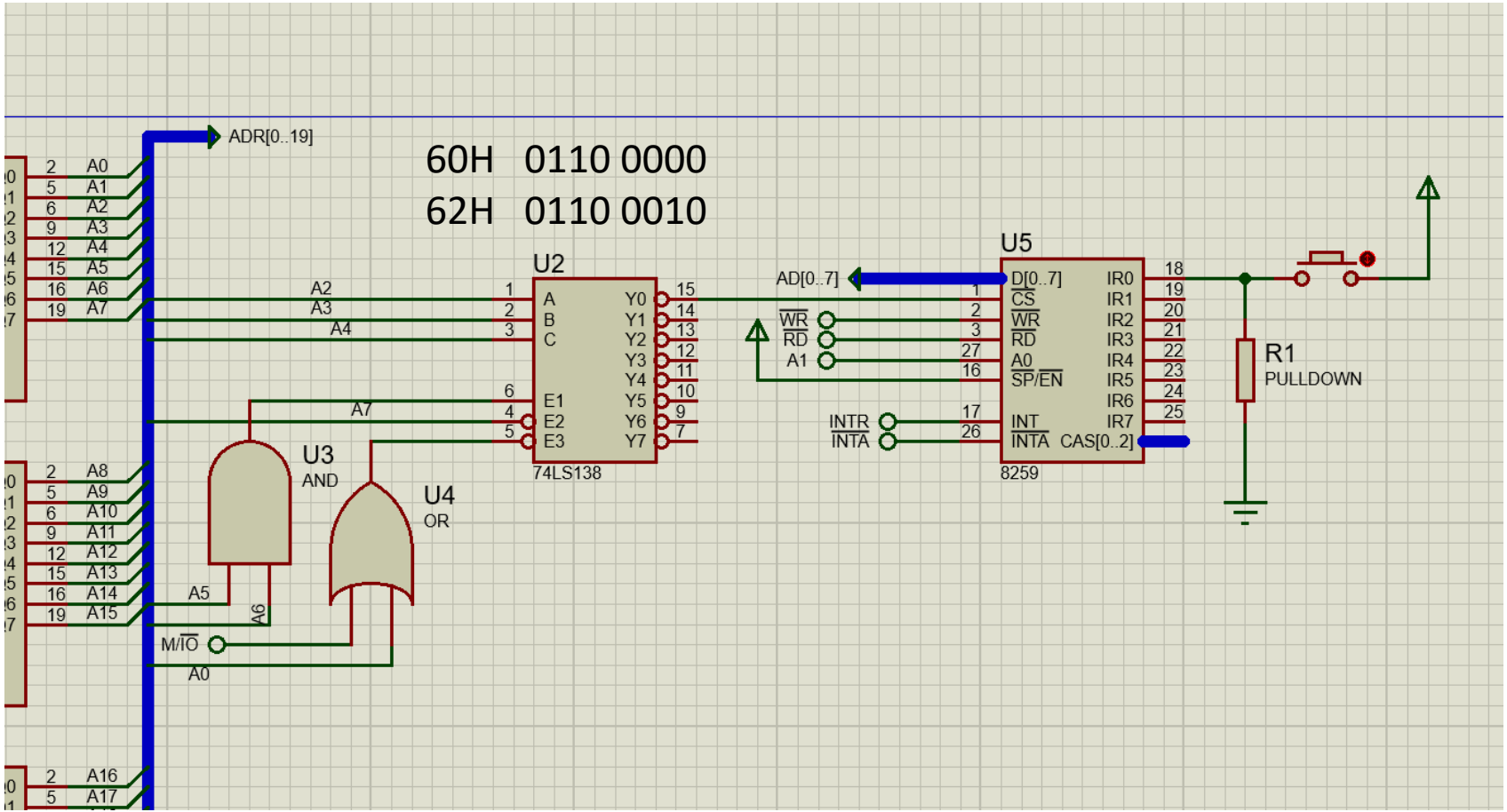
60H Adresinden itibaren ardışık çift adreslere 8259 yerleştiriliyor.
8259'un IR0 ucuna pull-down dirençli bir buton bağlı.

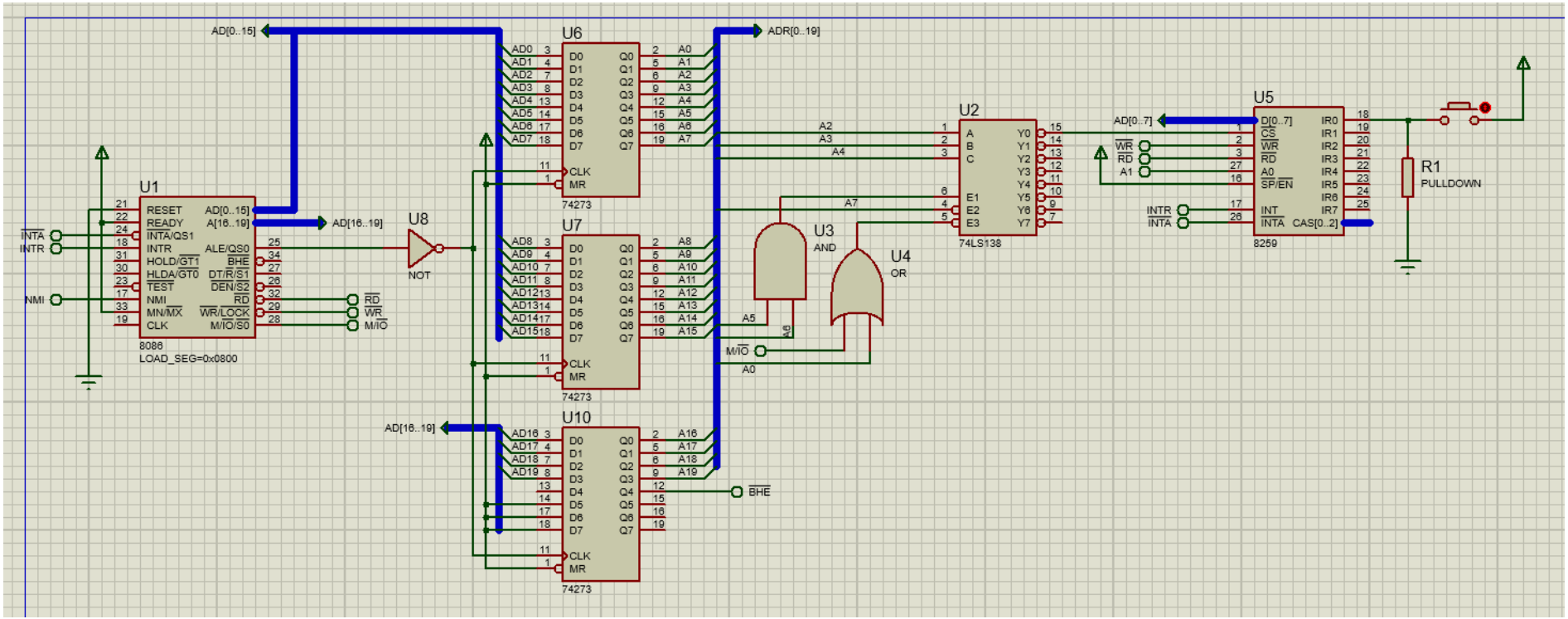
1.) Butona basıldığında 40H tipinde kesme tetiklenmeli.
2.) 40H tipindeki kesme ile butona kaç kez basıldığı sayılmalı.

60H Adresinden itibaren ardışık çift adreslere 8259 yerleştiriliyor.
8259'un IR0 ucuna pull-down dirençli bir buton bağlı.

1.) Butona basıldığında 40H tipinde kesme tetiklenmeli.
2.) 40H tipindeki kesme ile butona kaç kez basıldığı sayılmalı.

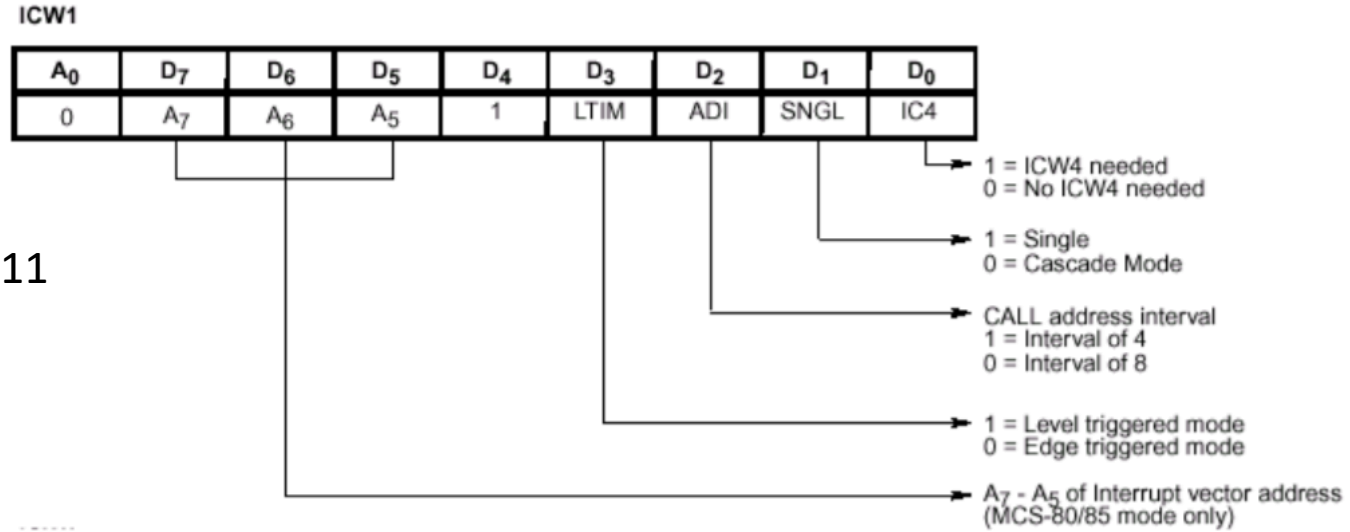
60H 0110 0000
62H 0110 0010


60H 0110 0000
62H 0110 0010
ADR[0..19]
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
A17
M/IO
U2
74LS138
U3
AND
U4
OR
U5
8259
AD[0..7]
D[0..7]
CS
WR
RD
A0
SP/EN
INT
INTA
CAS[0..2]
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
R1
PULLDOWN

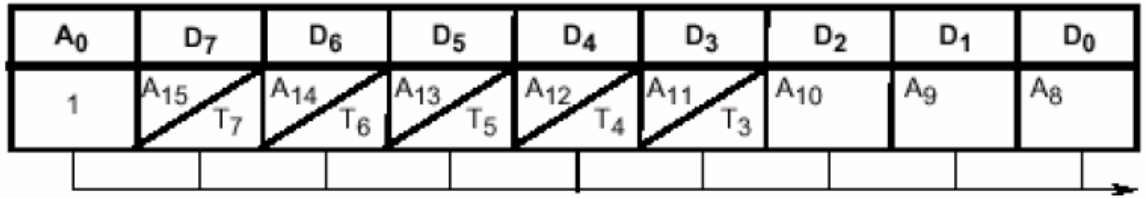

AD[0..15]
ADR[0..19]
AD[16..19]
U1
8086
LOAD_SEG=0x0800
RESET
READY
INTA/QS1
INTR
HOLD/GT1
HLDA/GT0
TEST
NMI
MN/MX
CLK
AD[0..15]
A[16..19]
ALE/QS0
BHE
DT/R/S1
DEN/S2
RD
WR/LOCK
M/IO/S0
INTA
INTR
NMI
U8
NOT
U6
U7
U10
74273
D0
D1
D2
D3
D4
D5
D6
D7
Q0
Q1
Q2
Q3
Q4
Q5
Q6
Q7
CLK
MR
U2
74LS138
A
B
C
E1
E2
E3
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
U3
AND
U4
OR
A5
A0
M/IO
BHE
U5
8259
D[0..7]
CS
WR
RD
A0
SP/EN
INT
INTA
IR0
IR1
IR2
IR3
IR4
IR5
IR6
IR7
CAS[0..2]
R1
PULLDOWN

```
49  MOV AL, 13H
50  OUT 60H, AL
51  MOV AL, 40H
52  OUT 62H, AL
53  MOV AL, 03H
54  OUT 62H, AL
55  STI
56  XOR AX, AX
57  ENDLESS:
```

0001 0011



ICW2




```
49  MOV AL,
50  OUT 60H, AL
51  MOV AL, 40H
52  OUT 62H, AL
53  MOV AL, 03H
54  OUT 62H, AL
55  STI
56  XOR AX, AX
57  ENDLESS:
```

0100 0000

```
49  MOV AL, 13H
50  OUT 60H, AL
51  MOV AL, 40H
52  OUT 62H, AL
53  MOV AL, 03H
54  OUT 62H, AL
55  STI
56  XOR AX, AX
57  ENDLESS:
```

0000 0011

ICW4

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	SFNM	BUF	M/S	AEOI	µPM

0	X	- Non buffered mode
1	0	- Buffered mode slave
1	1	- Buffered mode master

1 = 8086/8088 mode
0 = MCS-80/85 mode

1 = Auto EOI
0 = Normal EOI

1 = Special fully nested moded
0 = Not special fully nested mode

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

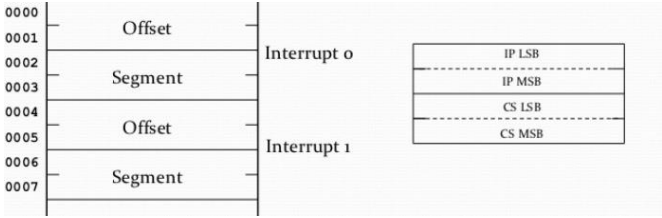
M/S is used to set the function of the 8259 when operated in buffered mode.
If M/S is set the 8259 will function as the MASTER.
If cleared will function as SLAVE.

```
;=====================================================================
STAK    SEGMENT PARA STACK 'STACK'
        DW 20 DUP(?)
STAK    ENDS

DATA    SEGMENT PARA 'DATA'
MYDAT    DB 20 DUP(0)
DATA    ENDS

CODE    SEGMENT PARA 'CODE'
        ASSUME CS:CODE, DS:DATA, SS:STAK

NEWINT  PROC FAR
        PUSH BP
        MOV BP, SP
        INC AX

        POP BP
        IRET
NEWINT  ENDP

START PROC FAR
        MOV AX, DATA
        MOV DS, AX


        XOR AX, AX
        MOV ES, AX
        MOV AL, 40H
        MOV AH, 4
        MUL AH
        MOV BX, AX
        LEA AX, NEWINT
        MOV WORD PTR ES:[BX], AX
        MOV AX, CS
        MOV WORD PTR ES:[BX+2], AX



MOV AL, 13H
OUT 60H, AL
MOV AL, 40H
OUT 62H, AL
MOV AL, 03H
OUT 62H, AL
STI
XOR AX, AX
ENDLESS:



JMP ENDLESS
RET
START ENDP
```

```
19  CODE    SEGMENT PARA 'CODE'
20          ASSUME CS:CODE, DS:DATA, SS:STAK
21
22  NEWINT  PROC FAR
23          PUSH BP
24          MOV BP, SP
25          INC AX
26
27          POP BP
28          IRET
29  NEWINT  ENDP
30
31  START PROC FAR
32          MOV AX, DATA
33          MOV DS, AX
34
35
36          XOR AX, AX
37          MOV ES, AX
38          MOV AL, 40H
39          MOV AH, 4
40          MUL AH
41          MOV BX, AX
42          LEA AX, NEWINT
43          MOV WORD PTR ES:[BX], AX
44          MOV AX, CS
45          MOV WORD PTR ES:[BX+2], AX
46
```

Base pointer and local variables

The base pointer is conventionally used to mark the start of a function's stack frame,



Given a Vector, where is the ISR address stored in memory ?

$$Offset = Type \times 4$$

- `LEA` means Load Effective Address
- `MOV` means Load Value

In short, `LEA` loads a pointer to the item you're addressing whereas MOV loads the actual value at that address.

The purpose of `LEA` is to allow one to perform a non-trivial address calculation and store the result [for later usage]

```
LEA ax, [BP+SI+5] ; Compute address of value

MOV ax, [BP+SI+5] ; Load value at that address
```

```
CODE    SEGMENT PARA 'CODE'
        ASSUME CS:CODE, DS:DATA, SS:STAK

NEWINT  PROC FAR
        PUSH BP
        MOV BP, SP
        INC AX

        POP BP
        IRET
NEWINT  ENDP

START PROC FAR
        MOV AX, DATA
        MOV DS, AX


        XOR AX, AX
        MOV ES, AX
        MOV AL, 40H
        MOV AH, 4
        MUL AH
        MOV BX, AX
        LEA AX, NEWINT
        MOV WORD PTR ES:[BX], AX
        MOV AX, CS
        MOV WORD PTR ES:[BX+2], AX

```

Review Questions

1. *What do you mean by interrupt ?*
2. *What is interrupt service routine ?*
3. *What are the sources of interrupts in 8086 ?*
4. *What is interrupt vector table ?*
5. *Draw and explain the IVT for 8086.*
6. *Briefly describe the conditions which cause the 8086 to perform each of the following types of interrupts : Type 0, Type 1, Type 2, Type 3 and Type 4.*
7. *Explain interrupt structure of 8086.*
8. *What are software interrupt ? How 8086 responds to software interrupts ?*
9. *Draw and explain the interrupt acknowledge cycle of 8086.*
10. *Describe the response of 8086 to the interrupt coming on pin.*
11. *What do you mean by interrupt priorities ?*
12. *State the interrupt priorities for 8086 interrupts.*
13. *What are advantages of using 8259 ?*
14. *List the features of 8259.*
15. *Explain the operating modes of 8259.*
16. *Draw and explain the interfacing of 8259 with 8086.*
17. *Draw and explain the interfacing of cascaded 8259s with 8086.*
18. *Explain the procedure of interrupt programming.*

ref: https://books.google.com.tr/books?id=OVYGy4NhymMC&pg=RA7-PA16&lpg=RA7-PA16&dq=call+address+intervals+8259&source=bl&ots=AxfAFQ_o8n&sig=ACfU3U2qrZLaXWMAPaZIz-A4vQHxhgca8A&hl=tr&sa=X&ved=2ahUKEwjRlc3I5qLpAhWZQUEAHUw6B7IQ6AEwBHoECAMQAQ#v=onepage&q&f=false