INTERFACING AND WORKING OF A "SINGLE" 8259

A single 8259 can accept 8 interrupts.
Whenever a device interrupts 8259, 8259 will interrupt the **µP on INTR pin.**
Hence, first the **INTR** signal of the µP **should be enabled** using the **STI** instruction**.**
8259 is **initialized** by giving **ICW1** and **ICW2** (compulsory) and **ICW4** (optional).
Note that **ICW3** is **not given** as Single 8259 is used. OCWs are given if required.

Once 8259 is **initialized**, the **following sequence** of events takes place when one or more **interrupts occur** on the IR lines of the 8259.

1) The **corresponding bit** for an interrupt is **set in IRR**.

2) The **Priority Resolver checks** the 3 registers:
   IRR (for highest interrupt request)
   IMR (for the masking Status)
   InSR (for the current level serviced)
   and **determines** the **highest priority** interrupt.
   It **sends** the **INT** signal **to** the **µP**.

3) The **µP finishes** the **current instruction** and **acknowledges** the interrupt by **sending** the **first** $\overline{\textbf{INTA}}$ **pulse**.

4) On receiving the first $\overline{\textbf{INTA}}$ signal, the **corresponding bit** in the **InSR** is **set** (indicating that now this interrupt is in service) and the **bit** in the **IRR** is **reset** (to indicate that the request is accepted).
   For doubts contact Bharat Sir on 98204 08217
   8259 now **prepares** to send the Vector number **N** to the µP on the data bus.

5) The **µP sends the second** $\overline{\textbf{INTA}}$ **pulse** to 8259.

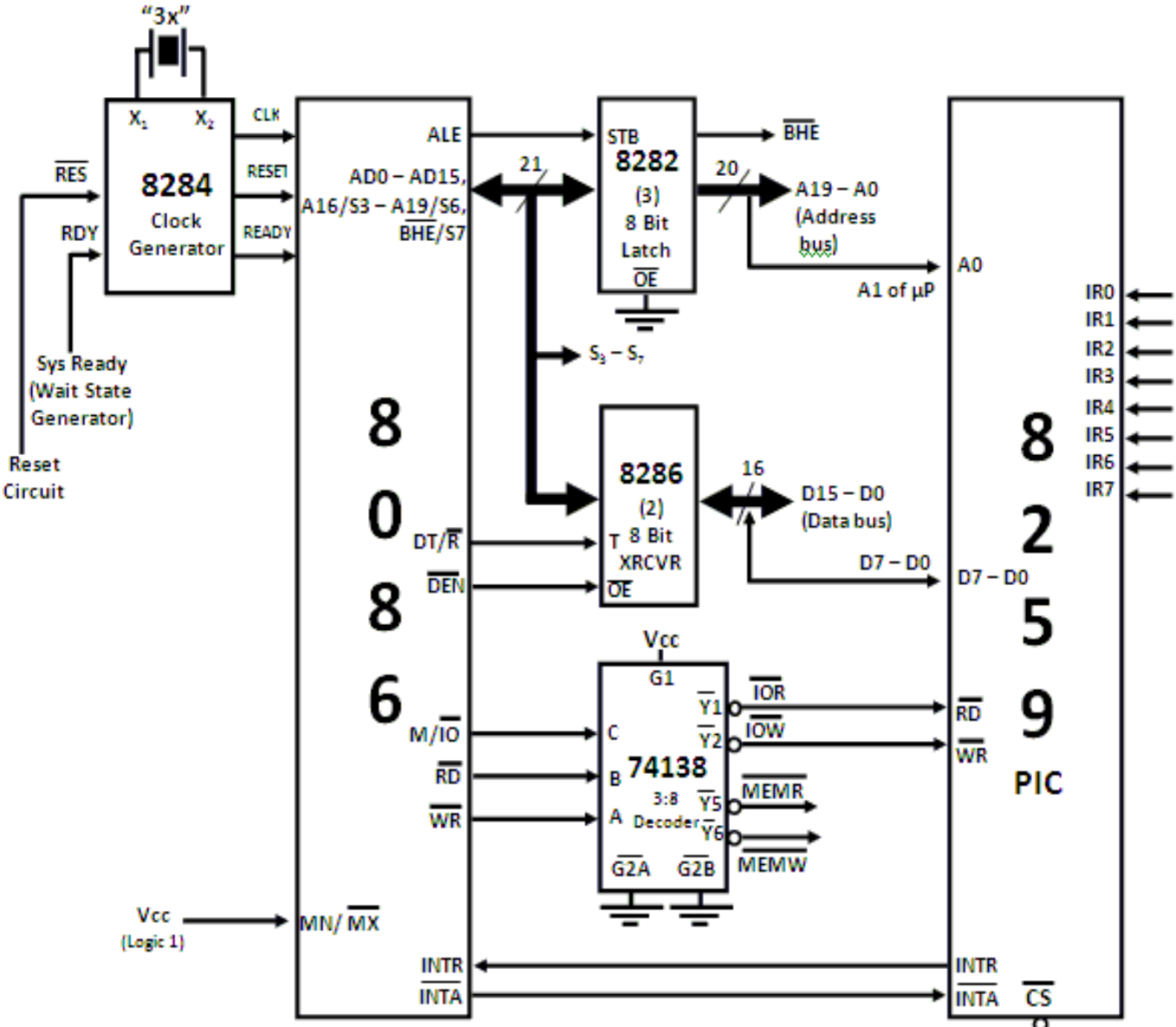
6) In response to the 2nd $\overline{\textbf{INTA}}$ pulse, **8259 sends the one byte Vector Number N to µP**.

7) Now the **µP multiplies N x 4**, to get the values of CS and IP **from the IVT**.

8) **In** the **AEOI Mode** the **InSR bit is reset** at this point, **otherwise** it **remains set until** an **appropriate EOI** command is given at the End of the ISR.

9) The **µP pushes** the contents of **Flag** Register, **CS**, **IP**, into the **Stack, Clears IF and TF** and **transfers program to** the **address** of the **ISR**.. *#Please refer Bharat Sir's Lecture Notes for this ...*

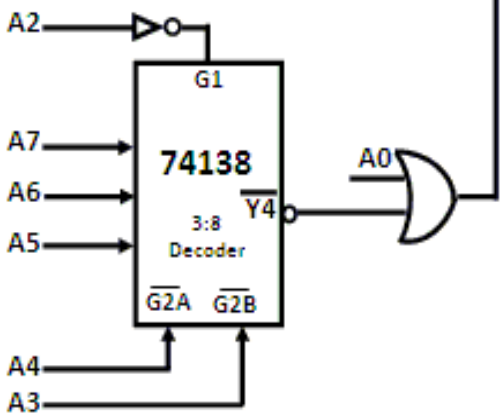
10) **The ISR thus begins**.



I/O Map of 8259 at 80H

	A7	A6	A5	A4	A3	A2	A1	A0	
ICW1	1	0	0	0	0	0	0	0	80H
ICW2	1	0	0	0	0	0	1	0	82H

Chip Selection
Internal Selection
Bank Selection



Interfacing and Working of "CASCADED" 8259

When **more than one 8259s** are connected to the µP, it is called as a **Cascaded configuration**.
A Cascaded configuration **increases** the **number of interrupts** handled by the system.
As the **maximum** number of **8259s** interfaced can be **9** (1 Master and 8 Slaves) the **Maximum** number of **interrupts** handled can be **64**.

The **master 8259** has $\overline{\textbf{SP}}$ / $\overline{\textbf{EN}}$ **= +5V** and the **slave** has $\overline{\textbf{SP}}$ / $\overline{\textbf{EN}}$ **= 0V**.

Each slave's INT output is **connected** to the **IR input** of the **Master**.
The **INT** output **of** the **Master** is **connected** to the **INTR** input of the **µP**.
The **master addresses** the individual **slaves through** the **CAS_2, CAS_1, CAS_0 lines** connected from the master to each of the slaves.
First the **INTR** signal of the µP **should be enabled** using the **STI** instruction**.**
Each 8259 (Master or Slave) **has** its **own address** and **has to be initialized separately** by giving ICWs as per requirement.

When an **interrupt request** occurs **on** a **SLAVE**, the events are performed:

1) The **slave 8259 resolves** the **priority** of the interrupt and **sends** the **interrupt to** the **master** 8259.

2) The **master resolves** the **priority** among its slaves and **sends** the **interrupt to** the **µP**.

3) The **µP finishes the current instruction** and **responds** to the interrupt **by sending 2** $\overline{\textbf{INTA}}$ **pulses**.

4) **In response to the first** $\overline{\textbf{INTA}}$ **pulse** the **following events occur**:

   i. The master **sends** the **3-bit salve identification number** on the **CAS lines**.

   ii. The **Master sets** the **corresponding bit** in **its InSR**.

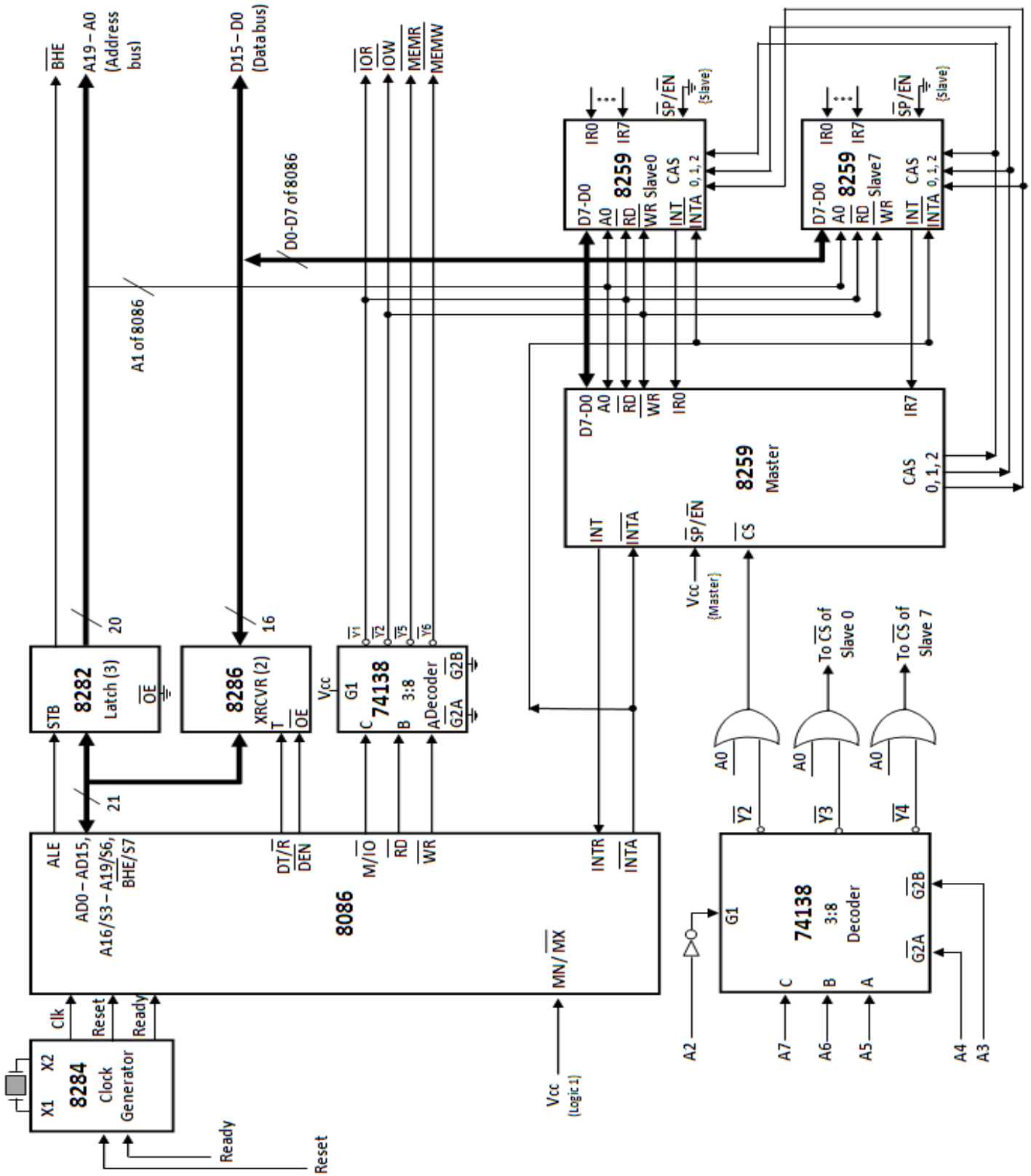
   iii. The **Slave identifies** its number on the **CAS lines** and **sets** the **corresponding bit in its InSR**.

5) In **response to** the **second** $\overline{\textbf{INTA}}$ pulse the **slave** places **Vector Number N** on the data bus.

6) **During** the **2nd** $\overline{\textbf{INTA}}$ pulse the **InSR bit** of the **slave** is **cleared** in **AEOI mode**, **otherwise** it is **cleared by** the **EOI command** at the end of the ISR.

7) The **µP pushes** the contents of **Flag** Register, **CS**, **IP**, into the **Stack, Clears IF and TF** and **transfers program to** the **address** of the **ISR**.. *#Please refer Bharat Sir's Lecture Notes for this ...*

8) **The ISR thus begins**.


BHE
A19 – A0 (Address bus)
D15 – D0 (Data bus)
IOR
IOW
MEMR
MEMW
D0-D7 of 8086
A1 of 8086
20
16
21
8282 Latch (3)
STB
OE
8286 XRCVR (2)
T
OE
74138 3:8 Decoder
G1
G2A
G2B
A
B
C
Y1
Y2
Y5
Y6
Vcc
8259 Master
8259
D7-D0
A0
RD
WR
IR0
IR7
INT
INTA
SP/EN
CS
CAS 0, 1, 2
Slave0
Slave7
{Slave}
Vcc {Master}
To CS of Slave 0
To CS of Slave 7
8086
ALE
AD0 – AD15,
A16/S3 – A19/S6, BHE/S7
DT/R
DEN
M/IO
RD
WR
INTR
INTA
MN/MX
Vcc (Logic 1)
8284 Clock Generator
X1
X2
Clk
Reset
Ready
A2
A3
A4
A5
A6
A7

I/O map		A7	A6	A5	A4	A3	A2	A1	A0	I/O
8259	ICW1	**0**	**1**	**0**	0	0	0	**0**	0	**40**
Master	ICW2	**0**	**1**	**0**	0	0	0	**1**	0	**42**
8259	ICW1	**0**	**1**	**1**	0	0	0	**0**	0	**60**
Slave 0	ICW2	**0**	**1**	**1**	0	0	0	**1**	0	**62**
8259	ICW1	**1**	**0**	**0**	0	0	0	**0**	0	**80**
Slave 7	ICW2	**1**	**0**	**0**	0	0	0	**1**	0	**82**