Programmable Interrupt Controller 8259A

Lecture 21

Interrupts

- Interrupts alter a program's flow of control
  - Behavior is similar to a procedure call
    - Some significant differences between the two
- Interrupt causes transfer of control to an *interrupt service routine* (ISR)
    - ISR is also called a *handler*
- When the ISR is completed, the original program resumes execution
- Interrupts provide an efficient way to handle unanticipated events

Interrupts versus Procedures

Interrupts

- Initiated by both *software* and *hardware*
- Can handle *anticipated* and *unanticipated* internal as well as external events
- ISRs or interrupt handlers are memory resident
- Use numbers to identify an interrupt service
- (E)FLAGS register is saved automatically

Procedures

- Can only be initiated by *software*
- Can handle *anticipated* events that are coded into the program
- Typically loaded along with the program
- Use meaningful names to indicate their function
- Do not save the (E)FLAGS register

How Are Hardware Interrupts Triggered

- Non-maskable interrupt is triggered by applying an electrical signal to the MNI pin of Pentium
  - Processor always responds to this signal
  - Cannot be disabled under program control
- Maskable interrupt is triggered by applying an electrical signal to the INTR (INTerrupt Request) pin of Pentium
  - Pentium recognizes this interrupt only if IF (interrupt enable flag) is set
  - Interrupts can be masked or disabled by clearing IF

How Does the CPU Know the Interrupt Type?

- Interrupt invocation process is common to all interrupts
  - Whether originated in software or hardware
- For hardware interrupts, CPU initiates an interrupt acknowledge sequence
  - CPU sends out interrupt acknowledge (INTA) signal
  - In response, interrupting device places interrupt type number on the data bus
  - CPU uses this number to invoke the ISR that should service the device (as in software interrupts)

Interrupts on 8086

- When an interrupt occurs the CPU completes the current instruction and then:
  - Disables the maskable interrupt (This prevents the interrupt from itself being interrupted
  - Saves the IP, CS code segment register, and Flags register on the stack.
  - Jumps to an address found in memory locations 4xN, where N is the number of the interrupt.
  - Executes an ISR found at that address
  - At the end of the ISR executes an iret instruction which pops IP, CS and the Flags register off the stack, restoring the CPU to the status it had before the interrupt occurred.

8086 hardware interrupt inputs

- 8086 has two hardware interrupt inputs:-
  - NMI
    - used to attend the power failure problems
  - INTR
    - used with an external device 8259 to handle interrupts from multiple sources
  - 8086 responds differently on INTR interrupt
  - While NMI is reserve, only one interrupt input is left to handle all other interrupt types “INTR”
- Typical system has more than one device that can interrupt --- keyboard, hard disk, floppy, etc.
- Use a special chip 8259 to prioritize the interrupts and forward only one interrupt to the CPU
  - The interrupt input type is sent to 8086 from an external hardware “8259A Priority Interrupt Controller”

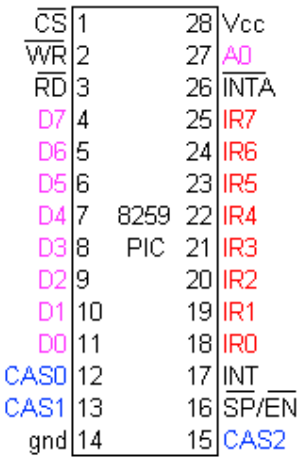
Interrupt handling with 8259A

- 8259A accepts interrupts
- 8259A determines priority
- 8259A signals 8086 (raises INTR line)
- CPU Acknowledges
- 8259A puts correct vector on data bus
- CPU processes interrupt
  - Each interrupt line has a priority
  - Higher priority lines can interrupt lower priority lines
- 8259 can service up to eight hardware devices
  - Interrupts are received on IRQ0 through IRQ7

8259 Programmable Interrupt Controller

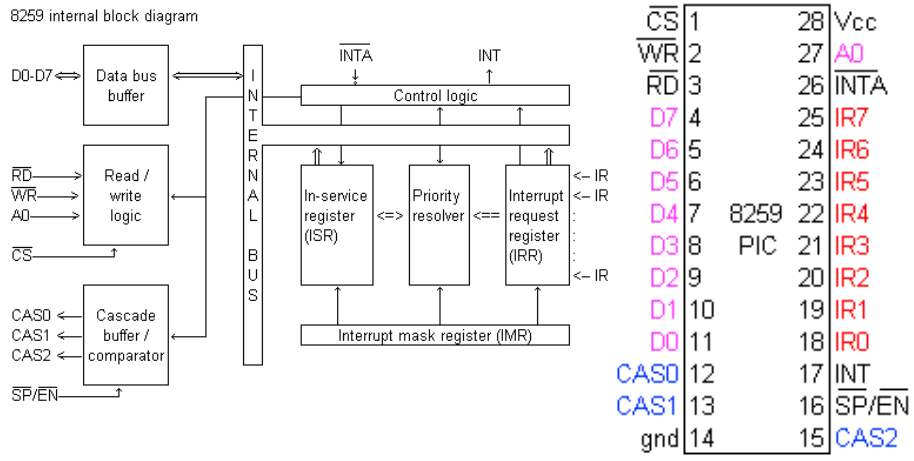
- 8259 can be programmed to assign priorities in several ways
  - Fixed priority scheme is used in the PC
    - IRQ0 has the highest priority and IRQ7 lowest
- 8259 has two registers
  - Interrupt Command Register (ICR)
    - Used to program 8259
  - Interrupt Mask Register (IMR)

8259 Pin layout

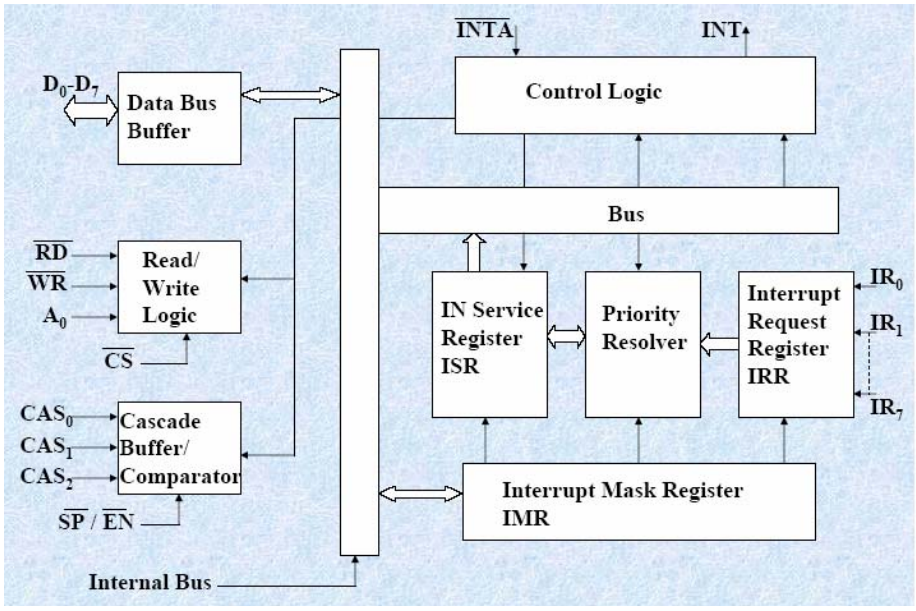


D0-D7	Bi-directional, tristated, buffered data lines. Connected to data bus directly or through buffers
RD-bar	Active low read control
WR-bar	Active low write control
A0	Address input line, used to select control register
CS-bar	Active low chip select
CAS0-2	Bi-directional, 3 bit cascade lines. In master 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.

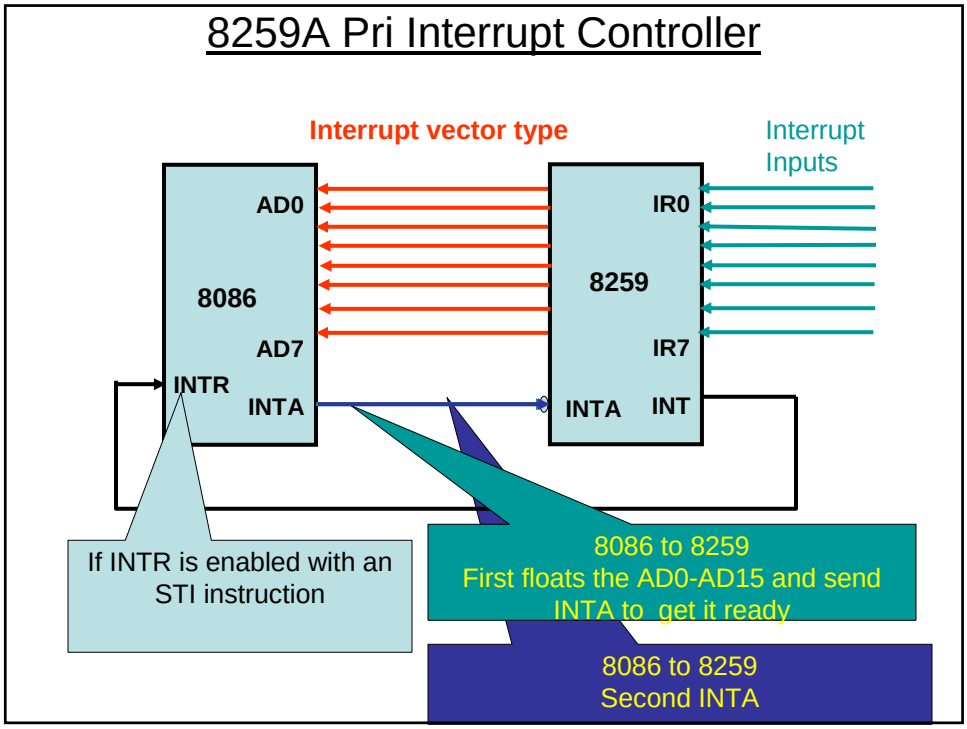
8259 Block Diagram



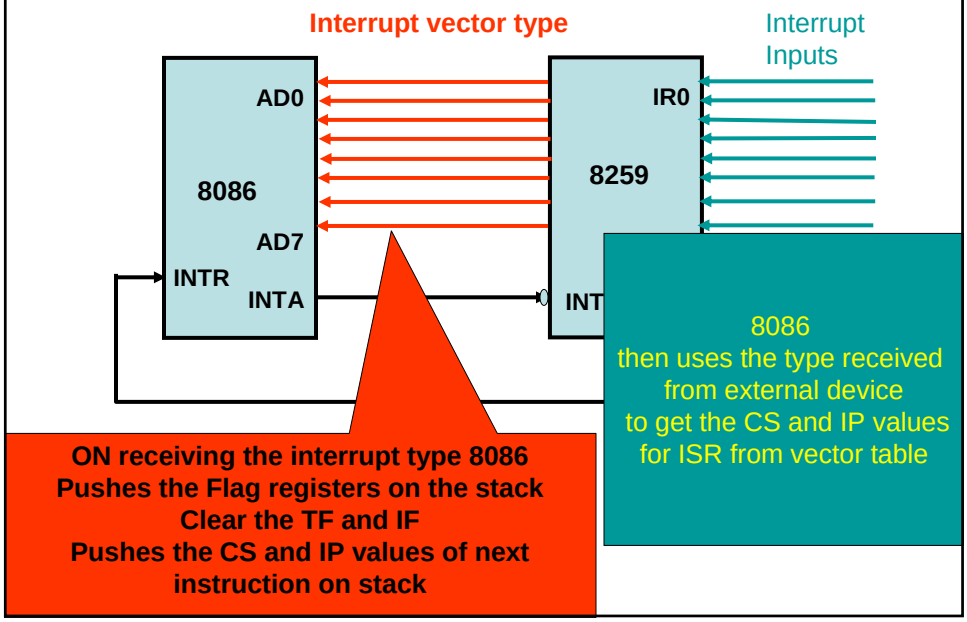
Internal Structure – 8259A



8259A Pri Interrupt Controller



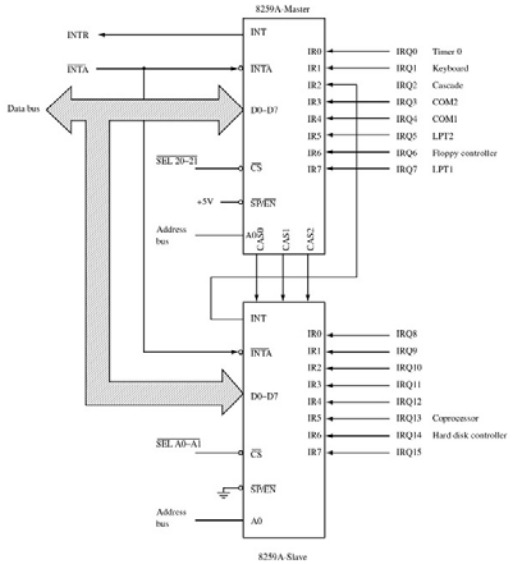
8259A Pri Interrupt Controller



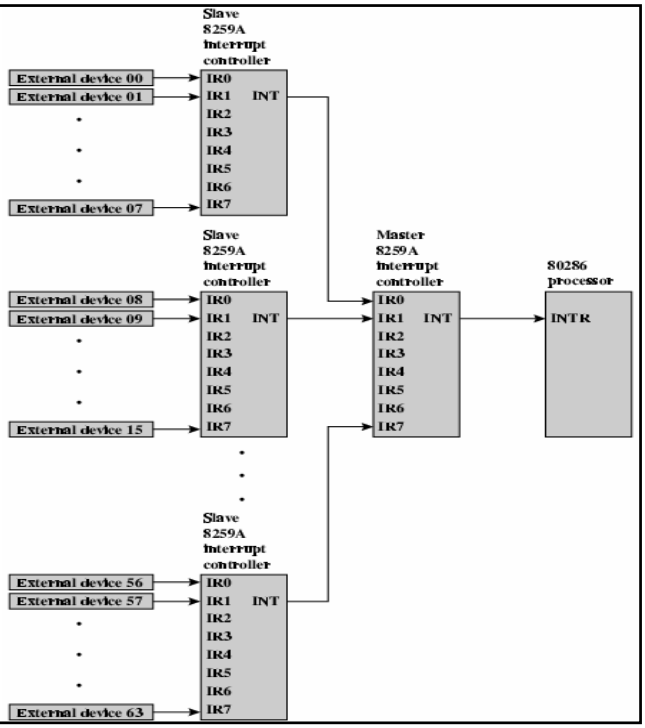
8259A working

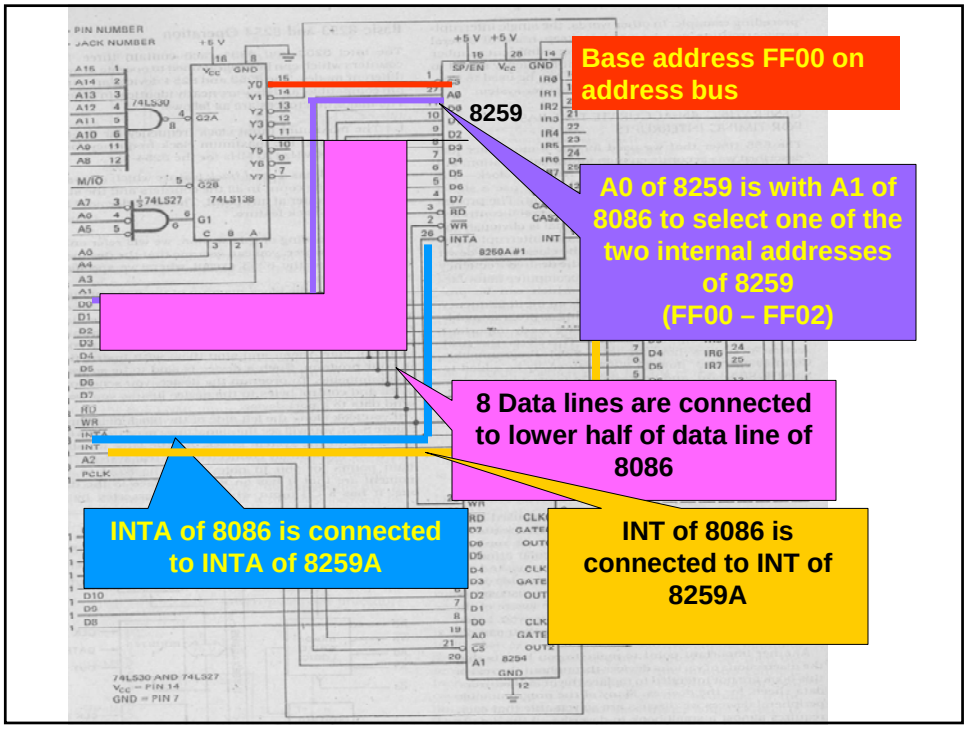
- If 8086 interrupt flag is set and INTR input receives a high signal, then 8086 will:-
  - Send out two interrupt ack pulses on its INTA pin to the INTA pin of 8259A
  - The INTA pulses tells the 8259A to send the desired interrupt type on data bus
  - Multiply the interrupt type by 4 to get the address
  - Push the flag on the stack
  - Clear IF and TF
  - Push the return address on the stack
  - Get the starting address of interrupt procedure
  - Execute the interrupt service procedure

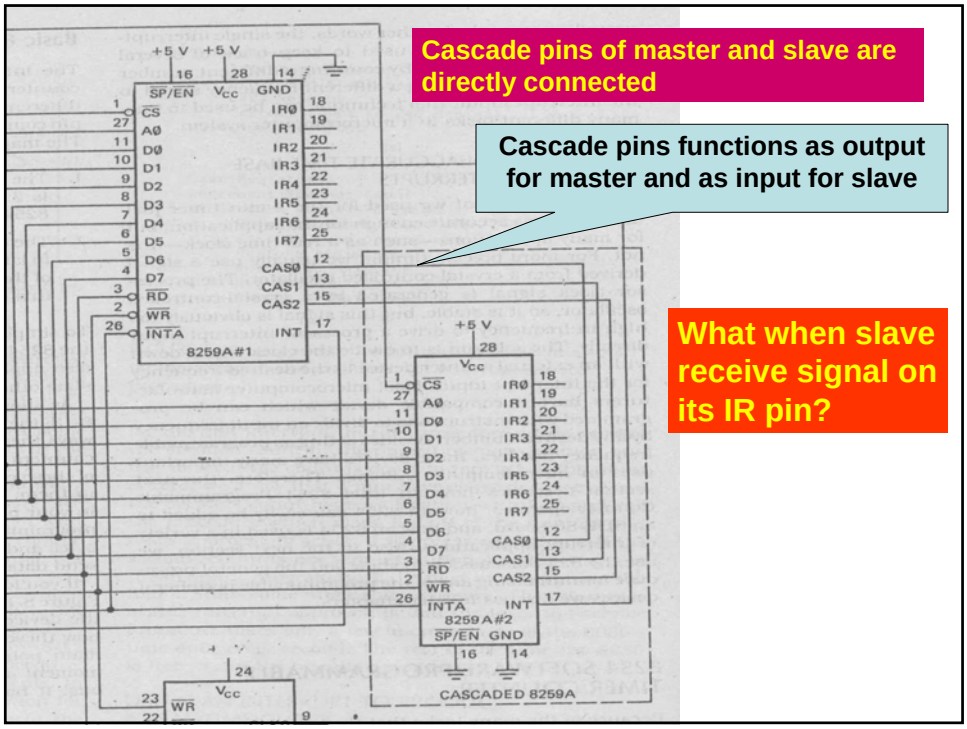
Example of two cascaded PICs

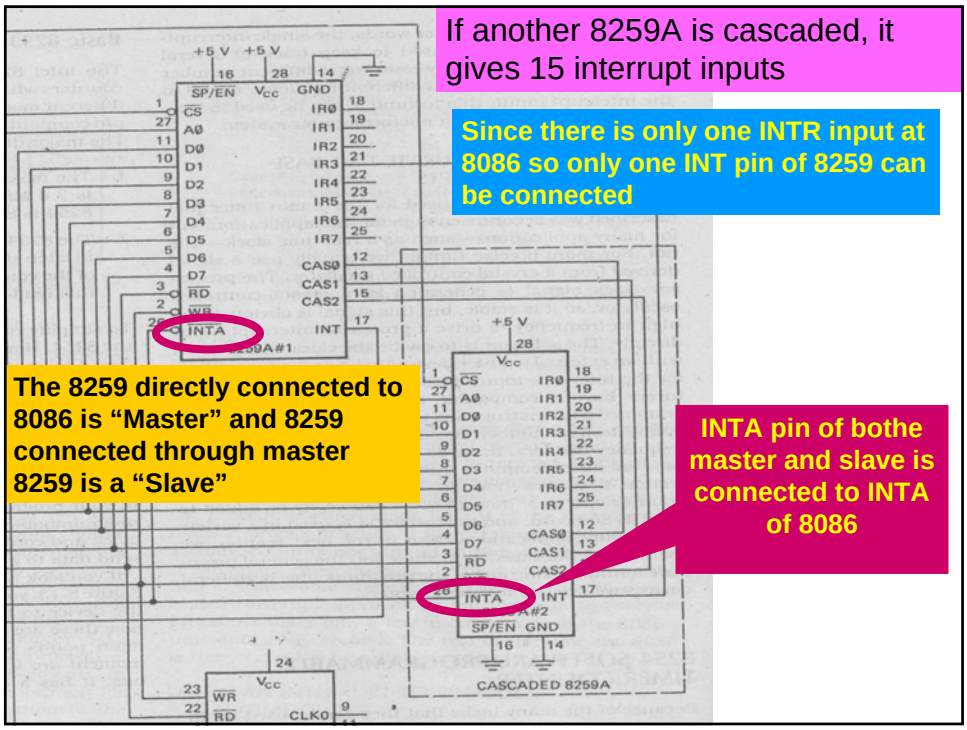


Multiple Interrupts using 8259A Interrupt Controller






Cascade pins of master and slave are directly connected
Cascade pins functions as output for master and as input for slave
What when slave receive signal on its IR pin?
8259A#1
8259A#2
CASCADED 8259A


If another 8259A is cascaded, it gives 15 interrupt inputs
Since there is only one INTR input at 8086 so only one INT pin of 8259 can be connected
The 8259 directly connected to 8086 is “Master” and 8259 connected through master 8259 is a “Slave”
INTA pin of bothe master and slave is connected to INTA of 8086
CASCADED 8259A

Initializing and programming 8259A

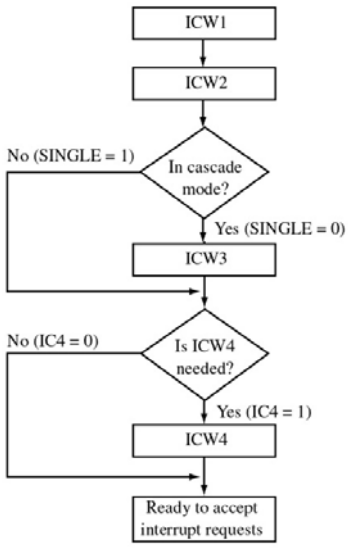
- 8259 is programmed by initialization and Operation command word
  - Initialization command word (ICW) are programmed before 8259 is able to function
    - ICW dictates the basic mode of operation
  - Operation Command word (OCW) are used during the normal course of operation, to make 8259 function properly

Internal Registers 8259A

- **Priority Resolver :**
  - This unit determines the priorities of the interrupt requests appearing simultaneously.
    - The highest priority is selected and stored into the corresponding bit of ISR during INTA pulse.
    - The IR0 has the highest priority while the IR7 has the lowest one, normally in fixed priority mode.
  - The priorities however may be altered by programming the 8259A in rotating priority mode.
- **• Interrupt Mask Register (IMR)** :
  - This register stores the bits required to mask the interrupt inputs.
    - IMR operates on IRR at the direction of the Priority Resolver.

ICWs

- There are four ICWs for 8259A
- When 8259 is first powered
  - ICW1 and ICW2 are sent
  - ICW3 is only needed when ICW1 is programmed for cascade operation (SNLG =1)
  - ICW4 will be programmed for 8086 operation



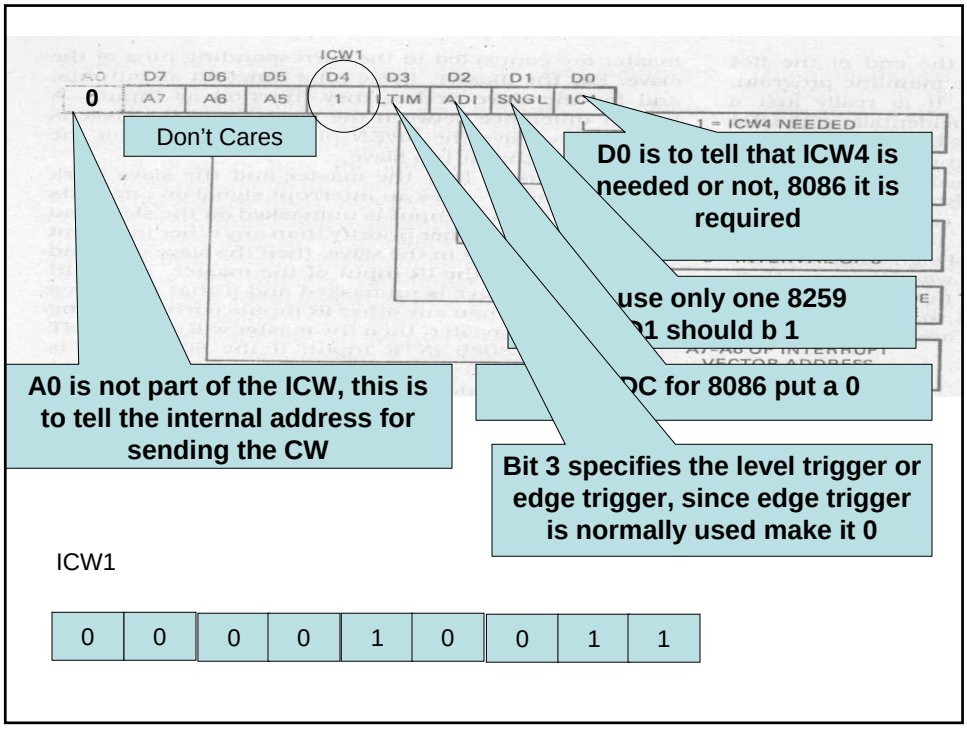
ICWs

How can the 8259 make a distinction between ICW2, ICW3, and ICW4 when they are sent to the same address?

This is one of the functions of ICW1. D0, the LSB of ICW1, will tell the 8259 if it should look for ICW4 or not.

In a similar manner, if D1 is high it knows that the system is configured in slave mode and it should not expect any ICW3 in the initialization sequence.

The initialization sequence must always start with ICW1, followed by ICW2, and finally the last one, if needed.

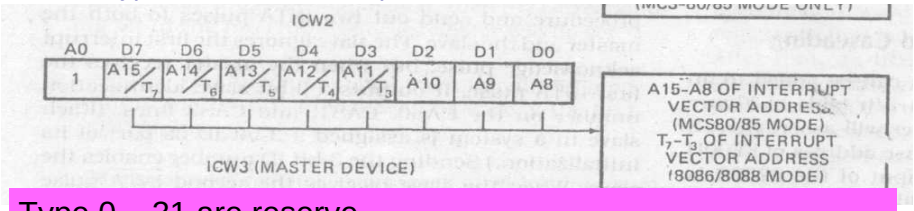


ICW1

- D0: IC4: 0=no ICW4, 1=ICW4 required
  D1: SNGL: 1=Single PIC, 0=Cascaded PIC
  D2: ADI: Address interval.
  - Used only in 8085, not 8086. 1=ISR's are 4 bytes apart (0200, 0204, etc) 0=ISR's are 8 byte apart (0200, 0208, etc)
- D3: LTIM: level triggered interrupt mode: 1=All IR lines level triggered. 0=edge triggered
- D4-D7: A5-A7: 8085 only.
  - ISR address lower byte segment. The lower byte is

ICW2

- Used to tell the 8259A the type number to send in response to an interrupt signal on the IR0 input
  - Interrupt signal to some other IR, 8259 will automatically add the number of the IR input to the base number and send it to 8086 as type number for that input

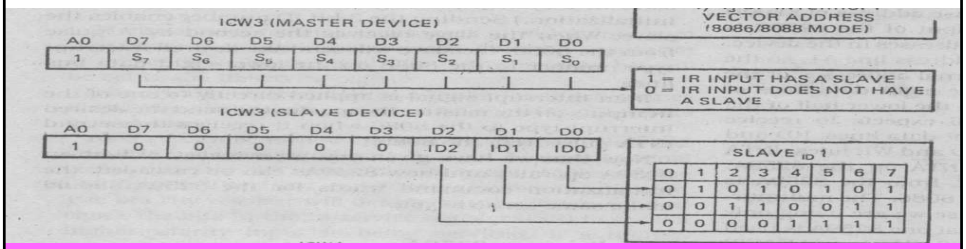


Type 0 – 31 are reserve

Type 32 (00100000h), is the lowest type number for IR0

  - Higher byte of ISR address (8085), or 8 bit vector address (8086).

ICW3



ICW3 is required if a slave is used in the system

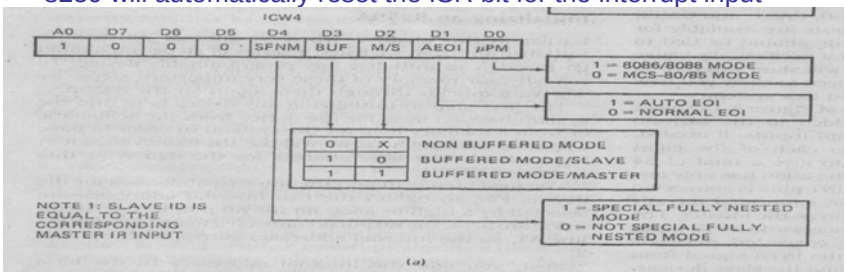
ICW 3 is required to tell the that at which IR input the slave is connected

CAS0 – CAS2 are used as three bit ID for each slave

A0=1
Master mode: 1 indicates slave is present on that interrupt, 0 indicates direct interrupt
Slave mode: ID3-ID2-ID1 is the slave ID number. Slave 4 on IR4 has ICW3=04h (0000 0100)

ICW4

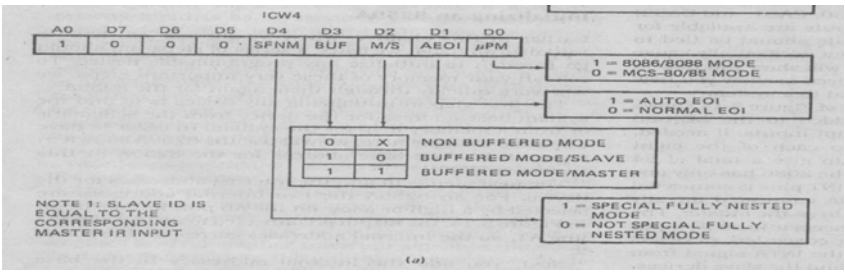
- The purpose of ICW 4 is to tell 8259 that it working is for x86 family
  - D0 as 1 for specifying 8086 system
  - D1 is used for EOI (end of interrupt) bit, if this bit is set in ICW4 the 8259 will automatically reset the ISR bit for the interrupt input



  - SFNM: 1=Special Fully Nested Mode, 0=FNM
  - M/S: 1=Master, 0=Slave
  - AEOI: 1=Auto End of Interrupt, 0=Normal
  - Mode: 0=8085, 1=8086

ICW4

- The purpose of ICW 4 is to tell 8259 that it working is for x86 family
  - D0 as 1 for specifying 8086 system
  - D1 is used for EOI (end of interrupt) bit, if this bit is set in ICW4 the 8259 will automatically reset the ISR bit for the interrupt input



A0=1
SFNM: 1=Special Fully Nested Mode, 0=FNM
M/S: 1=Master, 0=Slave
AEOI: 1=Auto End of Interrupt, 0=Normal
Mode: 0=8085, 1=8086

Operating Modes of 8259

- The 8259A can be programmed in different modes by setting or resting the appropriate bits of the ICW .
- The different modes of operation of 8259A are explained as the following.
  - **Fully Nested Mode** :
    - This is the default mode of operation of 8259A.
      - IR0 has the highest priority and IR7 has the lowest one.
    - When interrupt request are noticed, the highest priority request is determined and the vector is placed on the data bus.
      - The corresponding bit of ISR is set and remains set till the microprocessor issues an EOI command

Modes of Operation

- **Special Fully Nested Mode** :
  - This mode is used where cascading is used and the priority has to be programmed in the master using ICW4.
    - when an interrupt request from a slave is in service, this slave can further send request to the master,
      - if the requesting device connected to the slave has higher priority than the one being currently served.
  - In this mode, the master interrupt the CPU only when the interrupting device has a higher or the same priority than the one current being served.

Modes of Operation

- **End of Interrupt (EOI)** :
  - The ISR bit can be reset either with AEOI bit of ICW4 or by EOI command, issued before returning from the interrupt service routine.
- There are two types of EOI commands specific and non-specific.
  - When 8259A is operated in the modes that preserve fully nested structure, it can determine which ISR bit is to be reset on EOI.
  - When non-specific EOI command is issued to 8259A it will be automatically reset the highest ISR bit out of those already set.

Modes of Operation

- **Automatic Rotation** :
  - This is used in the applications where all the interrupting devices are of equal priority.
    - an interrupt request IR level receives priority after it is served while the next device to be served gets the highest priority in sequence.
      - Once all the device are served like this, the first device again receives highest priority.

Modes of Operation

- **Automatic EOI Mode** :
  - Till AEOI=1 in ICW4, the 8259A operates in AEOI mode.
    - In this mode, the 8259A performs a non-specific EOI operation at the trailing edge of the last INTA pulse automatically.
  - This mode should be used only when a nested multilevel interrupt structure is not required with a single 8259A.