Real Time System

Third Level

Lecture Fifteen

8259 Programmable Interrupt Controller (PIC)

RealTime Systems.
Dr. Osama Abbas Hussein

Goals
Up-on completing this lecture, the student should be able to:

1- Identify the concepts behind PIC
2- Comprehend the concepts interrupt driven data transfer modes.

Interrupt Driven Data Transfer Mode

The interrupt driven data transfer mode is the best method of data transfer for effectively utilizing the processor time. In this mode, the processor first initiates the I/O device for data transfer. After initiating the device, the processor will continue the execution of instructions in the program. Also at the end of an instruction the processor will check for a valid interrupt signal. If there is no interrupt then the processor will continue the execution. When the I/O device is ready, it will interrupt the processor. On receiving an interrupt signal, the processor will complete the current instruction execution and saves the processor status in stack. Then the processor calls an interrupt service routine (ISR) to service the interrupted device. At the end of ISR the processor status is retrieved from stack and the processor starts executing its main program. The sequence of operations for an interrupt driven data transfer mode is shown in figure below.

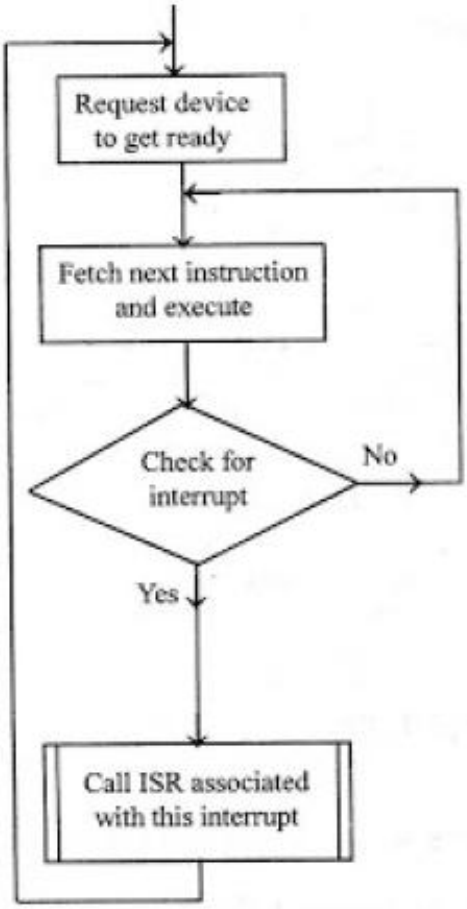


Fig (a) : Main program execution sequence

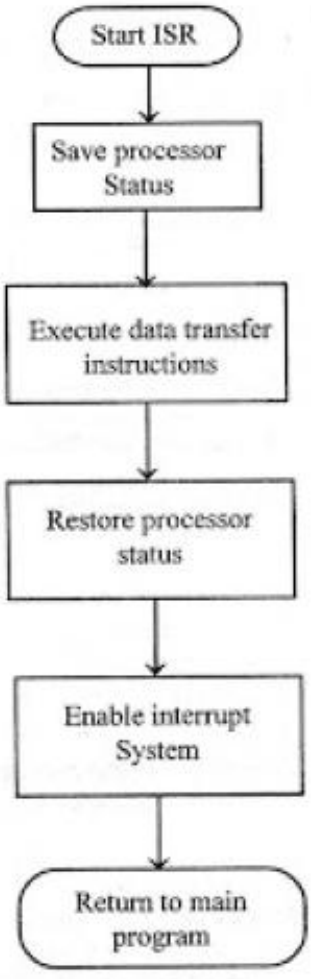


Fig (b) : ISR execution sequence

This is for one I/O device. What about many I/O devices?

8085 Processor has only 5 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.

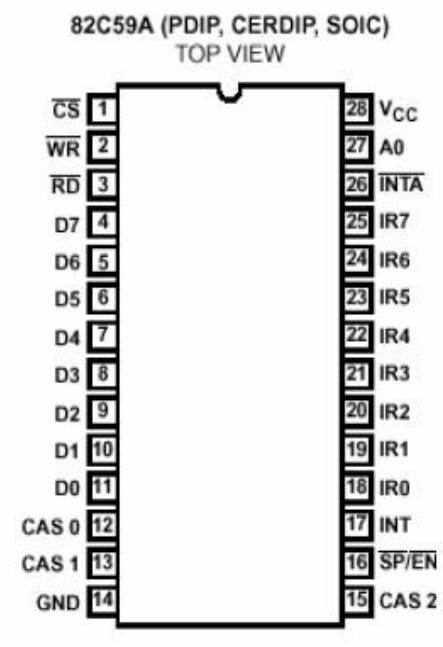
8259 Programmable Interrupt Controller (PIC):-

- Able to handle a number of interrupts at a time.
- Takes care of a number of simultaneously appearing interrupt requests along with their types and priorities.
- Compatible with 8-bit as well as 16-bit processors (The 8259A is a programmable interrupt controller designed to work with Intel microprocessor 8080 A, 8085, 8086, 8088).
- One 8259 can accept 8 interrupt requests and allow one by one to processor INTR pin.
- The Programmable Interrupt Controller (PlC) functions as an overall manager in an Interrupt-Driven system. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced, and issues an interrupt to the CPU based on this determination.

Features of 8259

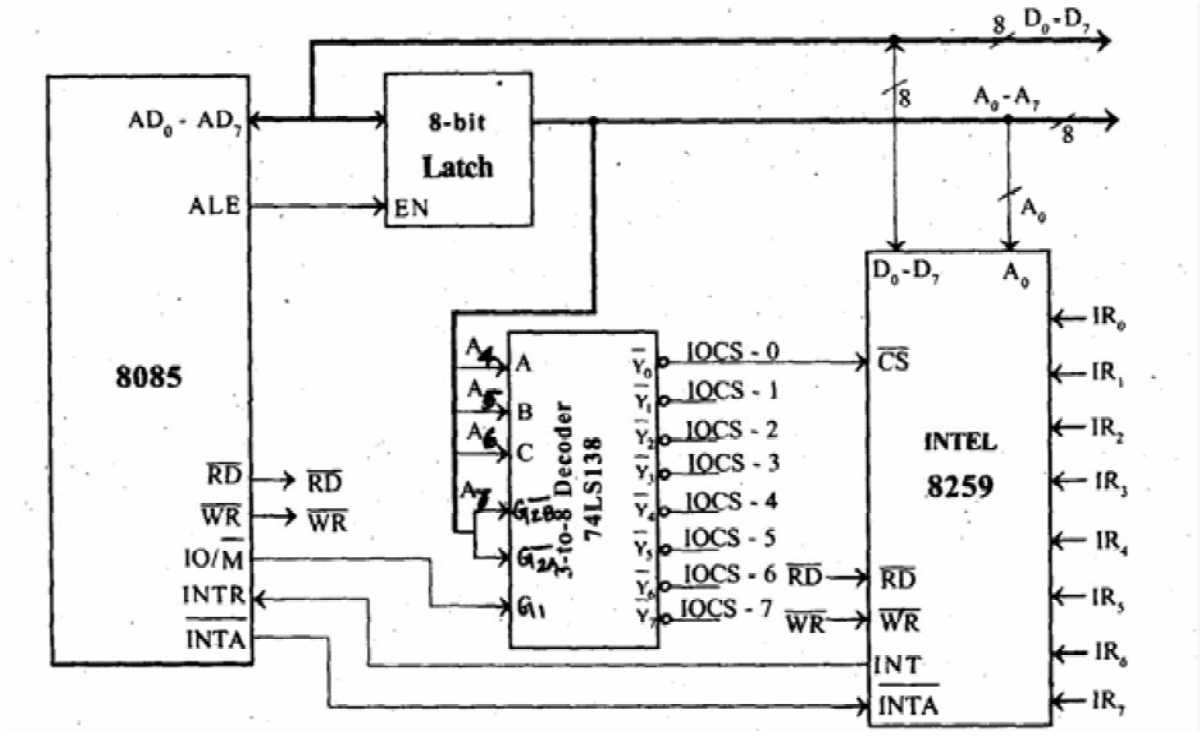
- Programmed to work with 8085 & 8086 (Compatible with 8-bit as well as 16-bit processors).
- It manages 8 interrupts according to the instructions written into its control registers.
- The priorities of interrupts are programmable.
- 8259 can be programmed to accept either level triggered or edge triggered
- The interrupts can be masked or unmasked individually.
- The 8259s can be cascaded to accept a maximum of 64 interrupts.

8259A PIC- Pin Diagram:-



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

Interfacing 8259 with 8085 Microprocessor:-



- It requires two internal address and they are A =0 or A = 1.
- The data bus lines D0-D7 of the 8085 processor are connected to D0-D7 of 8259.
- The address line A0 of the 8085 processor is connected to A0 of 8259 to provide the internal address.

- The 8259 require one chip select signal. Using 3-to-8 decoder generates the chip select signal for 8259.
- The address lines A4, A5 and A6 are used as input to decoder.
- The address line A7 is used as logic low enable for decoder.
- The address lines A1, A2 and A3.
- The I/O addresses of 8259 are shown in table.

	Binary Address								Hexa address
	Decoder input/ enable			Input to address pin of 8259					
	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
For A_0 of 8259 to be zero	0	0	0	0	x	x	x	0	00
For A_0 of 8259 to be one	0	0	0	0	x	x	x	1	01

Note : *Don't care "x" is considered as zero.*

Working of 8259 with 8085 Microprocessor:-

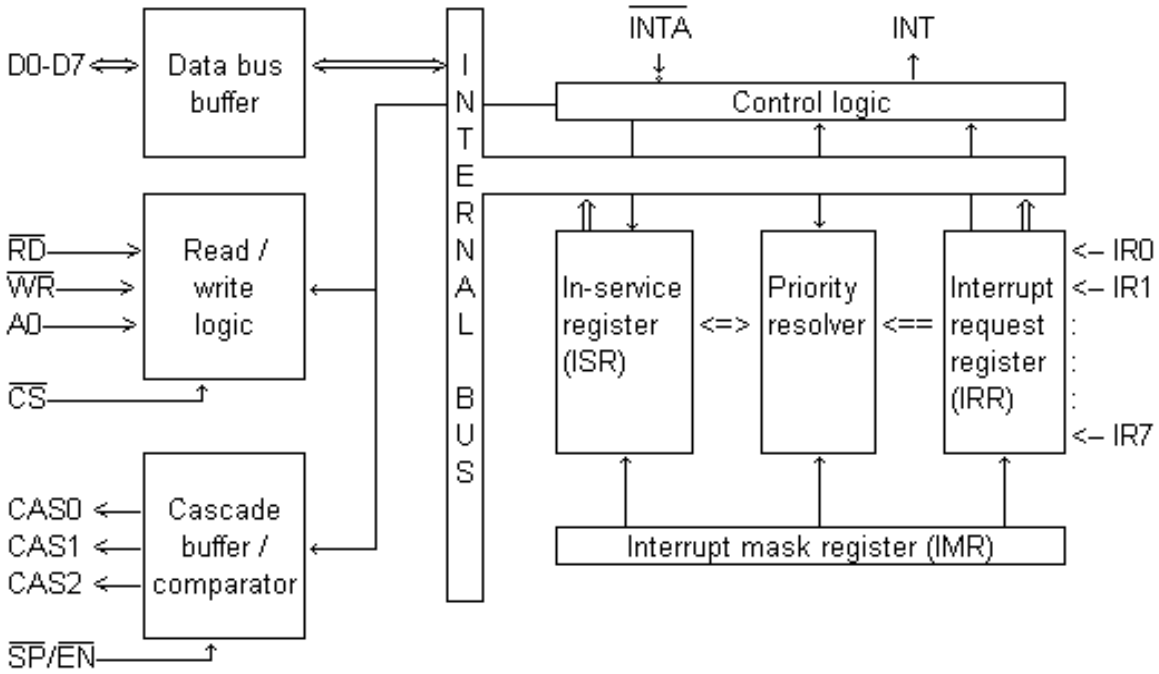
- First the 8259 should be programmed by sending Initialization Command Word (ICW) and Operational Command Word (OCW).
- These command words will inform 8259 about the following:
  - Type of interrupt signal (Level triggered / Edge triggered).
  - Type of processor (8085/8086).
  - Masking of interrupts.
  - Priority of interrupts.
  - Type of end of interrupts.
- Once 8259 is programmed it is ready for accepting interrupt signal. When it receives an interrupt through any one of the interrupt lines IR0-IR7 it checks for its priority and also checks whether it is masked or not.
- If the previous interrupt is completed and if the current request has highest priority and unmasked, then it is serviced.
- For servicing this interrupt the 8259 will send INT signal to INTR pin of 8085.
- In response it expects an acknowledge INTA (low) from the processor.
- When the processor accepts the interrupt, it sends three INTA (low) one by one.

- In response to first, second and third INTA (low) signals, the 8259 will supply CALL opcode, low byte of call address and high byte of call address respectively. Once the processor receives the call opcode and its address, it saves the content of program counter (PC) in stack and loads the CALL address in PC and start executing the interrupt service routine stored in this call address.

Functional block diagram of 8259:-

- It has eight functional blocks. They are
  - Control logic
  - Read Write logic
  - Data bus buffer
  - Interrupt Request Register (IRR)
  - In-Service Register (ISR)
  - Interrupt Mask Register (IMR)
  - Priority Resolver (PR)
  - Cascade buffer.

8259 internal block diagram



Control logic

For servicing interrupt the 8259 will send INT signal to INTR pin of 8085.

When the processor accepted the interrupts, it sends 3 INTA (low) one by one. In response:

- 1st INTA(low) - 8259 supply CALL opcode
- 2nd INTA(low)-low byte of call address
- 3rd INTA(low) -high byte of call address

Once the processor receive the call opcode and its address, it saves the content of program counter in stack and load the CALL address in PC and start executing the interrupt service routine stored in call address.

Read Write logic

The processor uses the RD (low), WR (low) and A0 to read or write 8259.

The 8259 is selected by CS (low).

Data bus buffer

The data bus and its buffer are used for the following activities.

- The processor sends control word to data bus buffer through D0-D7.
- The processor read status word from data bus buffer through D0-D7.
- From the data bus buffer the 8259 call opcode and address through D0-D7 to the processor.

Interrupt Request Register (IRR)

The IRR has eight input lines (IR0-IR7) for interrupts. When these lines go high, the request is stored in IRR. It registers a request only if the interrupt is unmasked.

Normally IR0 has highest priority and IR7 has the lowest priority. The priorities of the interrupt request input are also programmable.

Interrupt Mask Register (IMR)

The interrupt mask register (IMR) stores the masking bits of the interrupt lines to be masked. The relevant information is send by the processor through OCW.

In-Service Register (ISR)

The in-service register keeps track of which interrupt is currently being serviced.

Priority Resolver (PR)

The priority resolver examines the interrupt request, mask and in-service registers and determines whether INT signal should be sent to the processor or not.

Cascade buffer

The cascade buffer/comparator is used to expand the interrupts of 8259.

```
Summary:
1- There is a need to handle many interrupts at a time. This is the case for PIC
2- 8259 is such a device whose registers can be configured and dealt with just to do
that.


Questions:
1- What is a PIC device ?
2- How does the 8259 operate in a design?
```