Module III

Interrupt Mechanism of x86 & interfacing of chips

Interrupts and it's need

- Interrupts are useful for efficient data transfer between processor and peripheral
- When a peripheral is ready for data transfer it interrupts the processor by sending interrupts signal
- On receiving interrupts signal, the processor suspends the current program execution, save the status in stack and execute I.S.R(Interrupts Service routine)
- At the end of ISR the processor status is restored from stack and processor restores it's normal operation

There are 3 sources of interrupts

- i) Hardware interrupts applied through INTR and NMI pins
- ii) Software interrupts provided using 'INT n' instructions
- iii)Interrupts produced due to certain conditions in 8086 while executing an instruction for *eg:Divide by '0' if we made an attempt to divide operand by zero the program execution is automatically interrupted.Such conditional interrupts are called exceptions.*

Classification of InterruptsGenerally classified in to 3 types :-

- **i) Hardware and software interrupts**
- **ii) Vectored and non-vectored interrupts**
- **iii)Maskable and Non-maskable interrupts**
  - **Hardware Interrupts:**Interrupts initiated by external hardware by sending appropriate signals to the interrupt pin
  - 8086 has two interrupt pins , **INTR & NMI**

Software Interrupts

- ➔ Software interrupts are program instructions
- ➔ These interrupts are inserted at desired location in program (INT instructions)
- ➔ 8086 has 256 types of software interrupts (ranging from 0 to 255)
- ➔ All 256 types of interrupts including predefined interrupts can be inserted using 'INT n' instruction

Vectored Interrupt

- When an interrupt signal is received by the processor,if the program automatically transfers to predefined specific address then the interrupt is called vectored interrupt
- Predefined address in which ISR is stored is called vector address

- Vector address is predefined by the manufacturer of processor
- All 8086 interrupts are vectored interrupts
- Vector address of 8086 is obtained from a vector table

Maskable Hardware Interrupts

- Maskable interrupts are those which can be either accepted or rejected by processor
- Rejecting an interrupt is referred to as masking or disabling
- Accepting an interrupt is referred to as unmasking or enabling
- All hardware interrupts inserted through INTR pin are maskable
- Set I.F = 1,to enable INTR interrupt

Non-Maskable Interrupt

- The interrupts whose request has to be definitely accepted (or can't be rejected) is called Non-Maskable interrupt
- So upon receiving N.M.I the processor has to accept it definitely suspend the current program and should execute I.S.R
- Interrupts initiated through NMI pin and all software interrupts are non-maskable

INTEL Dedicated or Predefined Interrupts

- There are 256 interrupts
- INTEL has defined first 5 interrupts as dedicated for special functions ,they are called predefined interrupts (type 0 to type 4)
- Type 5 to type 31 are reserved by intel
- Type 32 to type 255 are available as hardware and software interrupts

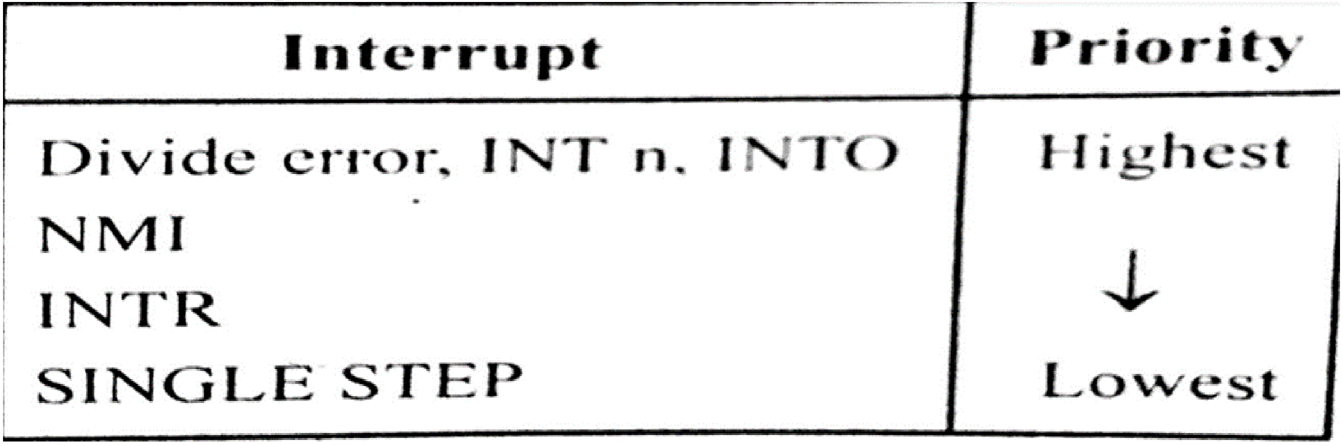
• Type-0 – Division by zero	• These interrupts are non-maskable.So the ISR indicates • error condition when an invalid division occurs, • The ISR is stored in vector table

• Type-I – Single step interrupt	• Execution of one instruction byone instruction is known as • single step. • When the TF(Trap Flag) is set 1 8086 automatically generates type-1 interrupt
• Type-2 Non-Maskable Interrupt NMI	• On receiving hiigh signal in NMI Pin.Usually Type-2 interrupts is used to save processor status.When a power failure occurs the rectifier which converts AC to DC has large filter capacitor which can retain dc power up to 50 milli seconds,that time will be sufficient to run ISR by type-2 interrupt and to save program data
• Type-3 – Breakpoint interrupt	• Used to implement break point function.It is useful to debug a program by executing the program part by part, User can insert INT 3 instructions at desired location

- Type 4 Interrupt-Overflow Interrupt

When the signed result of an arithmetic operation on two signed numbers is too large to fit in the destiation Register,an overflow error occurs.8086 sets overflow flag to 1.then INTO INSTn performs type 4 interrupt

Priorities of Interrupts in 8086



Interrupt	Priority
Divide error, INT n, INTO	Highest
NMI	
INTR	↓
SINGLE STEP	Lowest

Interrupt Vector Table

- ➔ For each and every interrupt in 8086 the system designer has written I.S.R and stored them in memory
- ➔ The system designer has created an Interrut vector table for this purpose
- ➔ The size of Interrupt vector table is 1kb with memory starting range 00000H to 003FFH
- ➔ Interrupt vector table consists of adresses of all 256 interrupt types

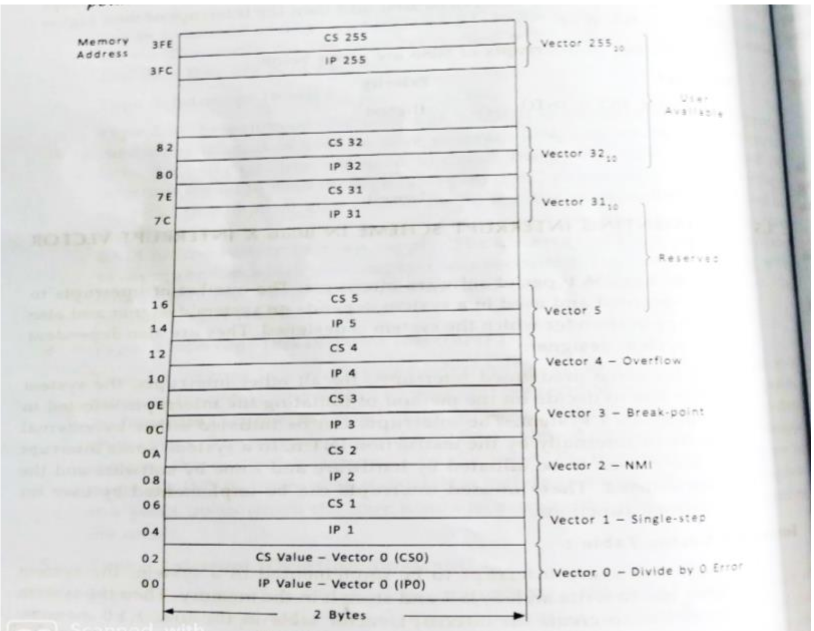


Fig. 3.1 Interrupt Vector Table of 8086

Servicing an interrupt by 8086If an interrupt request is deducted, then 8086 microprocessor respondsto interrupt by performing the following operations:

1.SP is decremented by 2 and contents of flag register are pushed to stack.

2.Interrupt s/m is disabled by clearing the interrupt flag(IF)

3.The single step trap flag is disabled by clearing the trap flag(TF)

4.SP is decremented by 2 and contents of CS are pushed to stack.

5.SP is decremented by 2 and contents of IP are pushed to stack.

6.Processor runs an Interrupt acknowledge cycle to get interrupt type number

7.20 bit physical address of ISS is given by:

Physical address of ISS =(16*contents of CS)+(Contents of IP)

8.The processor executes the ISS to service the interrupt.

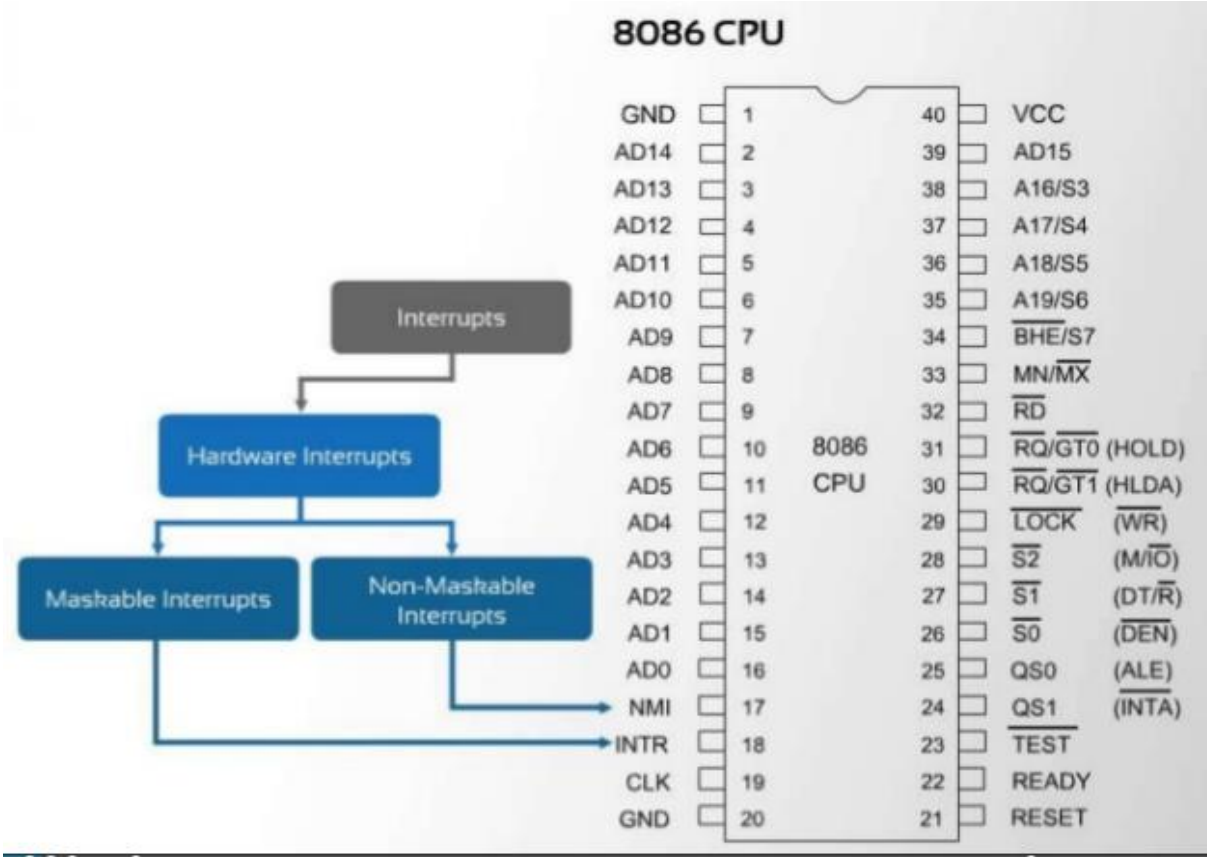
9.The ISS will be terminated by IRET.

10.Then the stack top is poped to IP,CS and flag register one by one word. After every pop operation, SP is incremented by 2.

11.Thus at the end of ISS, Processor will execute normal program execution from the instruction where it was suspended.

Programmable Interrupt Controller(PIC)

- ➔ P.I.C is used to combine several source of interrupt in to one or more C.P.U's
- ➔ P.I.C will arrange interrupts in their order of priority and will feed interrupts to 8086



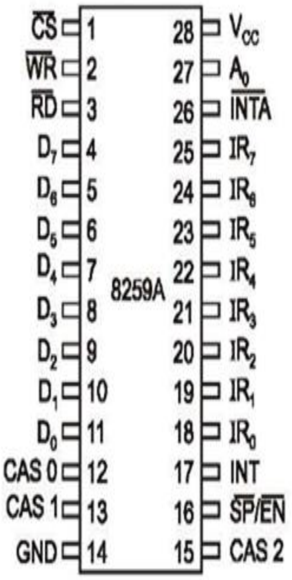
INTEL 8259

INTEL 8259 is a Programmable Interrupt Controller

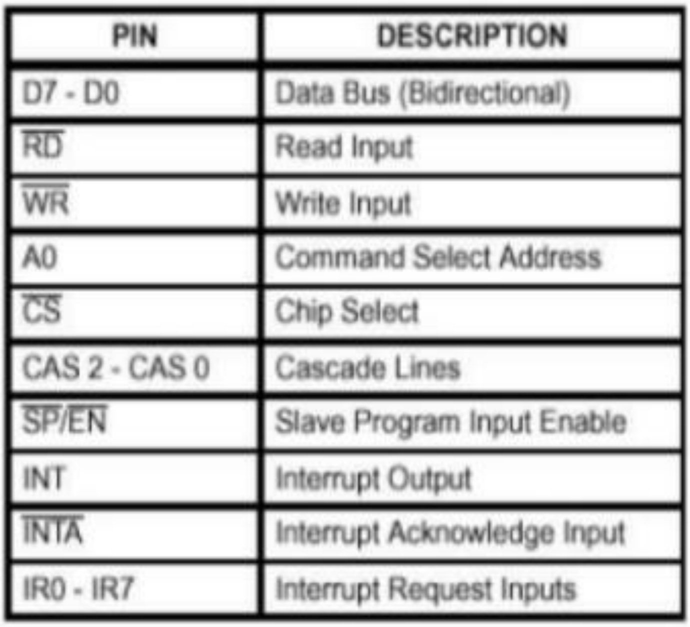
- ➔ It is used to expand the interrupts of 8085/8086

Features of 8259

- ➔ 8259 is available in 28 pin D.I.P
- ➔ 8259 can be programmed to work with 8086 or 8085
- ➔ One 8259 can accepts eight interrupt request and allow 1 by 1 to the processor
- ➔ The priorities of interrupts are programmable
- ➔ 8259 can provide the status of pending interrupts,masked interrupts and interrupts being served
- ➔ 8259 can operate in cascade mode and can accept a maximum of 64 interrupts
- ➔ 8259 can be programmed to accept level triggered or edge triggered interrupts

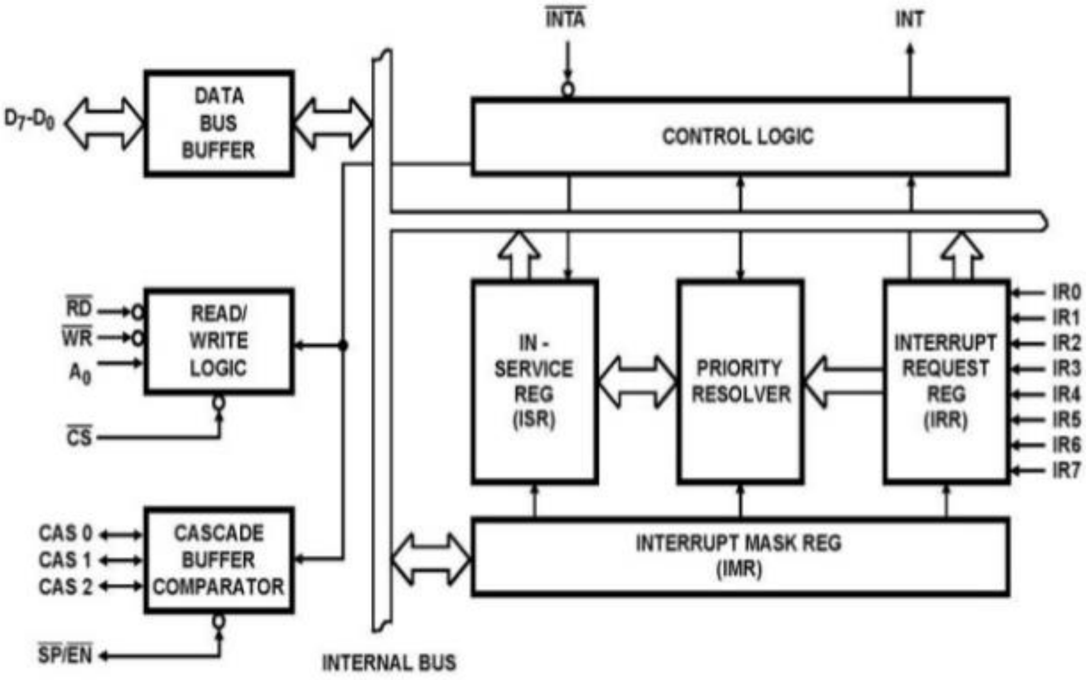


8259 pin diagram (*Source:* Intel Corporation)



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
$\overline{SP}/\overline{EN}$	Slave Program Input Enable
INT	Interrupt Output
$\overline{INTA}$	Interrupt Acknowledge Input
IR0 - IR7	Interrupt Request Inputs

Functional Block Diagram of 8259



- ➔ The functional block diagram of 8259 is as shown in figure
- ➔ It includes 8 functional blocks
- ➔ They are control logic,Read/Write logic,Data bus buffer,Interrupt Request Register(I.R.R),In-Service Register(I.S.R),Interrupt Mask Register(I.M.R),Priority Resolver(P.R) and Cascade buffer
- ➔ The IRR has 8 input lines for interrupts
- ➔ The IMR stores the interrupts line to be masked
- ➔ The ISR keeps track of interrupts that is being served
- ➔ PR sets the priority of interrupts
- ➔ Cascade buffer/comparator is used for cascade connection
- ➔ Data bus buffer is used to connect data lines D0-D7
- ➔ Read/Write logic is used for Read/write 8259

Programmable Peripheral Devices

Intel has developed number of peripheral devices that can be used with 8085/8086/8088 systems

- ➔ These peripheral devices are programmable so each peripheral devices performs specific functions

- ➔ Some of the Programmable Peripheral Devices are

i) Programmable Peripheral Interface-8255

ii) Keyboard/Display Controller-8279

iii)Interrupt Controller-8259

Data Communication

- ➔ In microprocessor based system data can be transferred by either serial / parallel data transfer scheme
- ➔ In parallel transfer scheme a group of data is transferred from one device to another
- ➔ For acheiving this a group of data lines are connected with processor and peripheral devices

Programmable Peripheral Interface-INTEL 8255

- ➔ INTEL 8255 is a Programmable Peripheral Interface device
- ➔ 8255 is used for implementing parallel data transfer between processor and slow peripheral devices such as LCD,DAC,ADC,7 Segment display
- ➔ It is available in 40 pin DIP
- ➔ 8255 has 3 ports

i)PORT A-8 Bit parallel port

ii)PORT B-8 Bit parallel port (Similiar to PORT A)

- ➔ iii)PORT C-Split in to two parts PART C Lower (PC0-PC3) and PORT C Upper(PC4-PC7)Ports are further grouped in to

i) GROUP A – Includes Port A ,Port C Upper & Control circuit

ii) GROUP B-Includes Port B,Port C Lower & Control circuit

Operating Modes of 8255

8255 has 3 operating modes :-

i)Mode-O

- ➔ In this mode all 3 ports can be programmed as either input or output port
- ➔ o/p are latched and i/p's are not latched
- ➔ In this mode ports does not have interrupting capability

ii)Mode-1

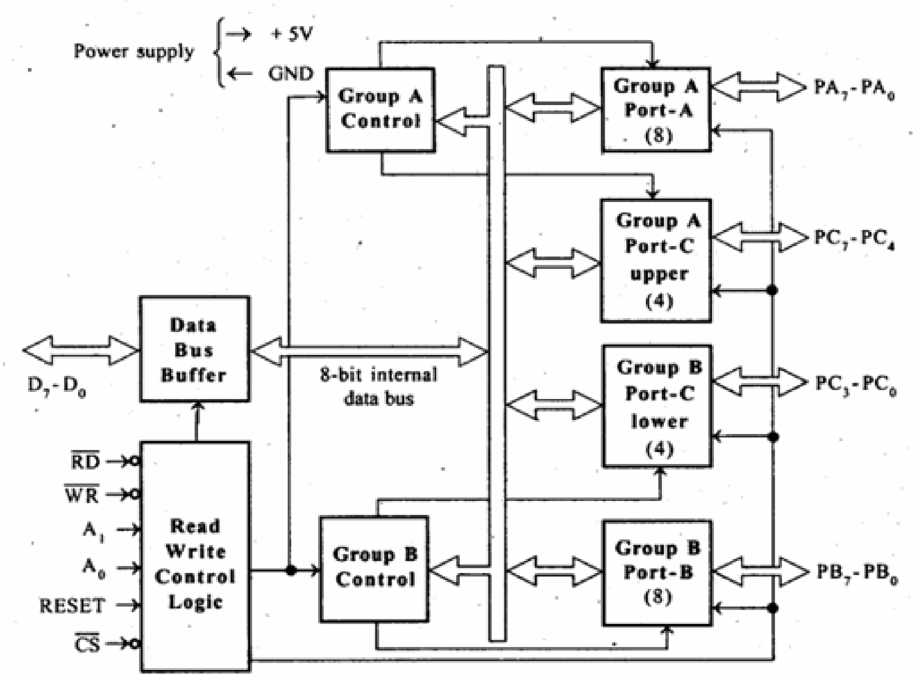
- ➔ In this mode only port A and port B can be programmed either as I/p or as O/p port
- ➔ Both i/p and o/p are latched

- ➔ Port C pins are used for handshake signals (each ports uses 3 pins of port c) iii) Mode-2
- ➔ Port A will be a bidirectional port (ie both read and write operation is possible)
- ➔ Only Port A can be programmed in mode-2
- ➔ 5 pins of Port C is used for handshake signals
- ➔ Port B remains in either Mode-0 or mode-1

iii) Mode-2

- Port A will be a bidirectional port (ie both read and write operation is possible)
- Only Port A can be programmed in mode-2
- 5 pins of Port C is used for handshake signals
- Port B remains in either Mode-0 or mode-

Block Diagram of 8255



RD (Read) : Enables read operation

WR (Write) : Enables write operation

RESET : Clears control register and set all ports in the input mode

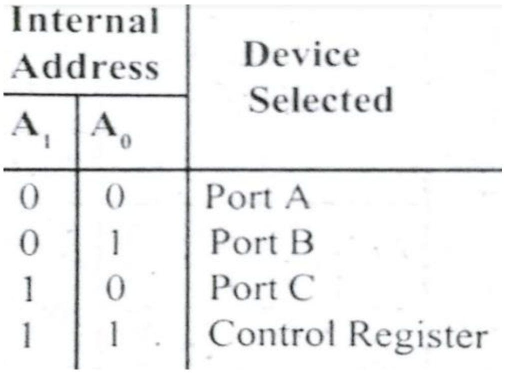
CS,A0,A1 : These are device select signals,A0 and A1 is connected to two adress lines of processor

RD (Read) : Enables read operation

WR (Write) : Enables write operation

RESET : Clears control register and set all ports in the input mode

A0,A1 : These are device select signals,A0 and A1 is connected to two adress lines of processor



Internal Address		Device Selected
A_1	A_0	
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Control Register

Keyboard/Display Controller-INTEL 8279

The INTEL 8279 is developed for interfacing keyboard and display devices.

Working : Intel's 8279 scans the Keyboard to identify if any key has been pressed and sends the code of the pressed key to the CPU. It also transmits the data received from the CPU, to the display device.

The features of 8279 are :-

- ➔ Simultaneous keyboard and display operation
- ➔ 16 x 8 Display RAM
- ➔ Scanned keyboard mode
- ➔ Scanned sensor mode
- ➔ Strobed input entry mode
- ➔ Mode programmable from CPU
- ➔ Available in 40 pin DIP
- ➔ 8 x 8 keyboard FIFO
- ➔ Right or Left entry

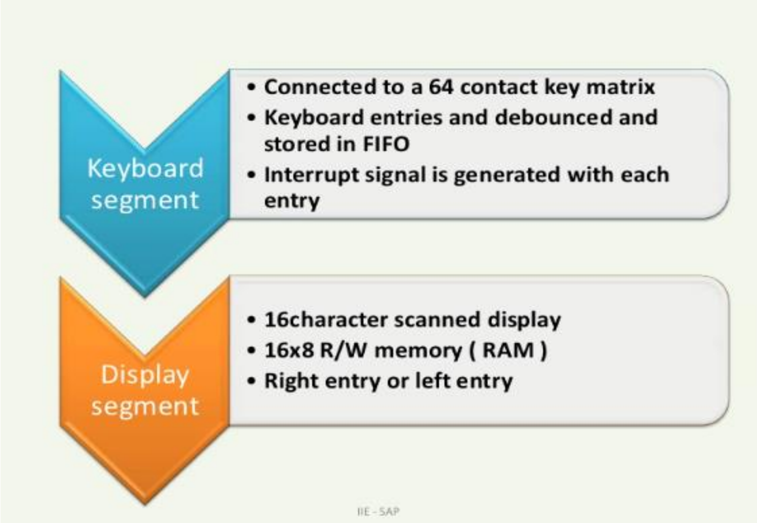
Why 8279 ?

Actually we can interface keyboard/display using 8255 but disadvantage are :

i) CPU have to refresh the display and to check the status of keyboard periodically for any key entry

ii)Thus a considerable amount of time is wasted which reduces operating speed

While INTEL'S 8279 can simultaneously drive keyboard and display ,thus cpu becomes free for its routine task



Keyboard segment

- Connected to a 64 contact key matrix
- Keyboard entries and debounced and stored in FIFO
- Interrupt signal is generated with each entry

Display segment

- 16character scanned display
- 16x8 R/W memory (RAM)
- Right entry or left entry

IIE - SAP

Block Diagram of 8279

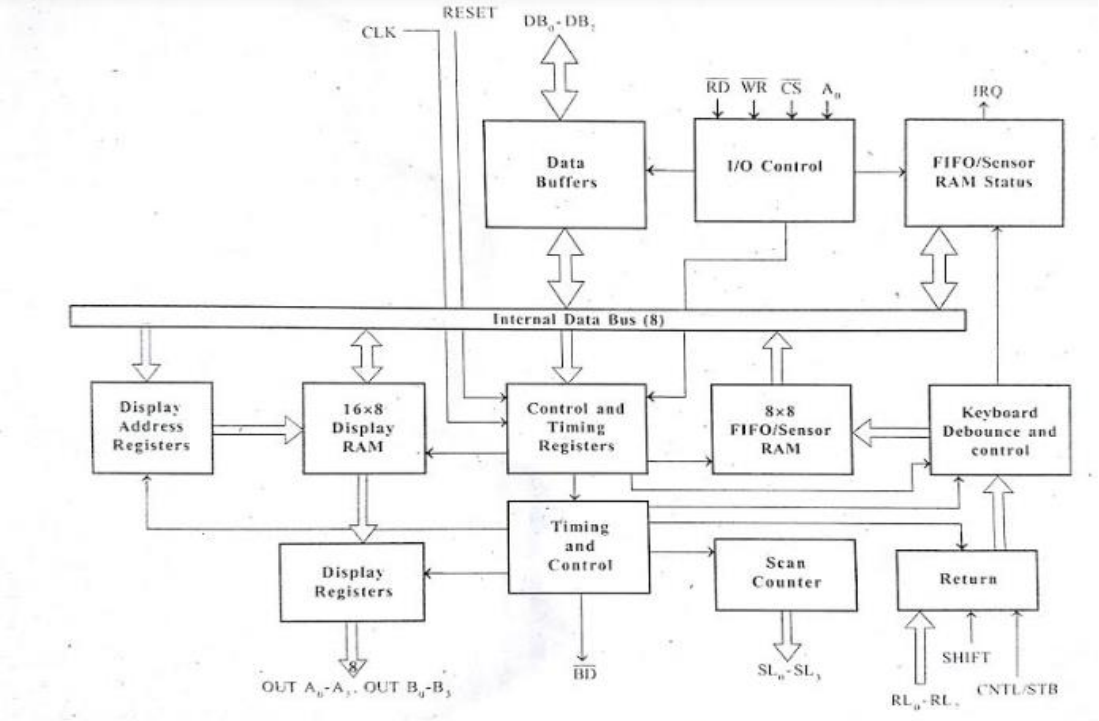


Fig 7.42 : Block diagram of 8279.

The block diagram of 8279 contains four major sections.These are

1.Keyboardsection 2.Display section 3.Scan section 4.Processor interface section

1.**Keyboard section**:This section has eight lines of RL0-RL7 that can be connected to a 8 columns of a keyboard setion and two additional lines for SHIFT and CNTL/STB

2.**Display section**:This section has 8 ouput lines.They are divided into two groups as A0-A3 and B0-B3.These lines can be used separately or combined.

3.**Scan section**:This section has scan counter and four san lines SL0-SL3

4.**Processor interfacing section**:It includes 8 bidirectional data lines DB0-DB7 to transfer data on IRQ(Interrupt Request Line) and six lines for i…