Unit 6
Basic I/O, Memory R/W and Interrupt Operations

Memory Read/Write Operation

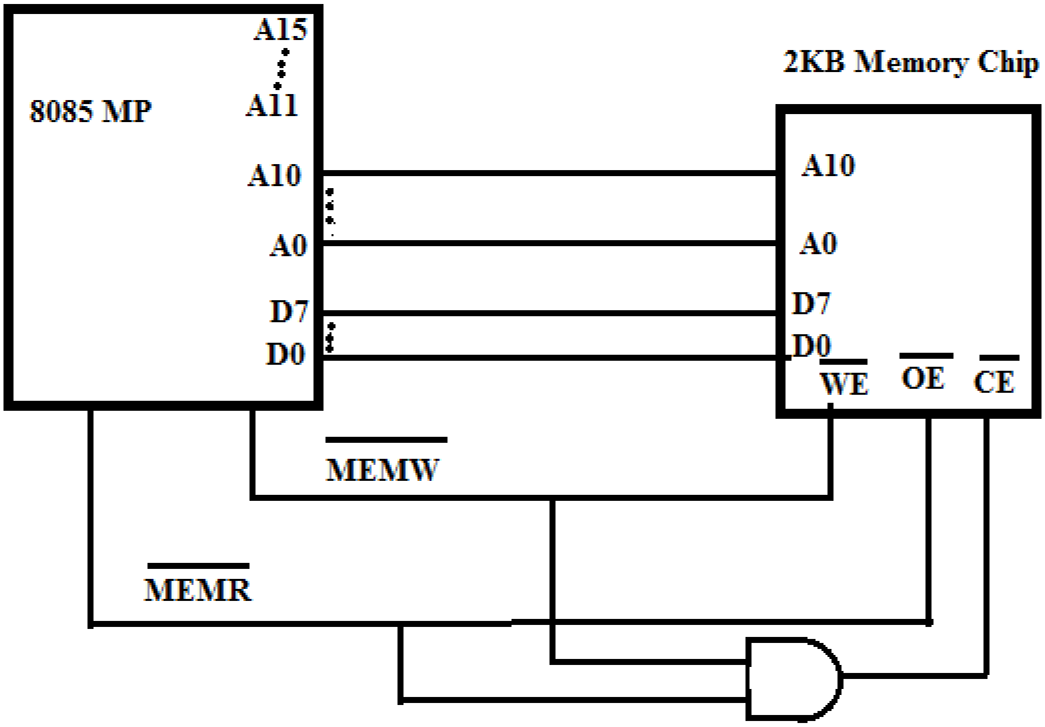


Fig: Memory chip interfacing with microprocessor

→ In the above figure, the lower address lines A0 – A10 of MP are connected to A0 – A10 pins of memory (2K) and D0 – D7 data lines of MP are connected to D0 – D7 of memory.

→ The memory read control signal MEMR(bar) and memory write control signal MEMW(bar) are connected to OE(bar) and WE(bar) pins of memory for read and write operation respectively.

→ When MP issued MEMW(bar) control signal to memory, then memory stores the data available in data lines into the addressed location. This process is known as memory write operation.

→ When MP issued MEMR(bar) control signal to memory, then memory places the data stored in addressed location towards the processor. This process is known as memory read operation.

Direct Memory Access (DMA)

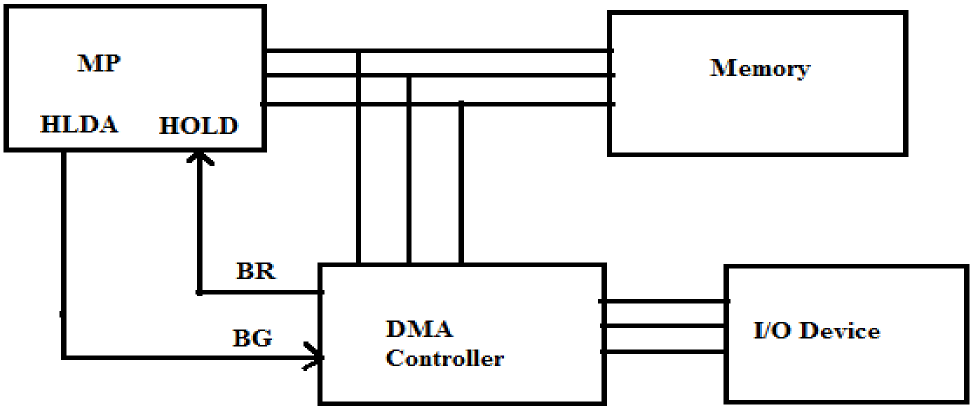


Fig: DMA

- → DMA is a process of communication for data transfer between memory and input/output, controlled by an external circuit called DMA controller, without involvement of CPU.
- → 8085 MP has two pins HOLD and HLDA which are used for DMA operation.
- → First, DMA controller sends a request by making Bus Request (BR) control line high. When MP receives high signal to HOLD pin, it first completes the execution of current machine cycle, it takes few clocks and sends HLDA signal to the DMA controller.
- → After receiving HLDA through Bus Grant (BG) pin of DMA controller, the DMA controller takes control over system bus and transfers data directly between memory and I/O without involvement of CPU. During DMA operation, the processor is free to perform next job which does not need system bus.
- → At the end of data transfer, the DMA controller terminates the request by sending low signal to HOLD pin and MP regains control of system bus by making HLDA low.

DMA controller 8237 Interfacing

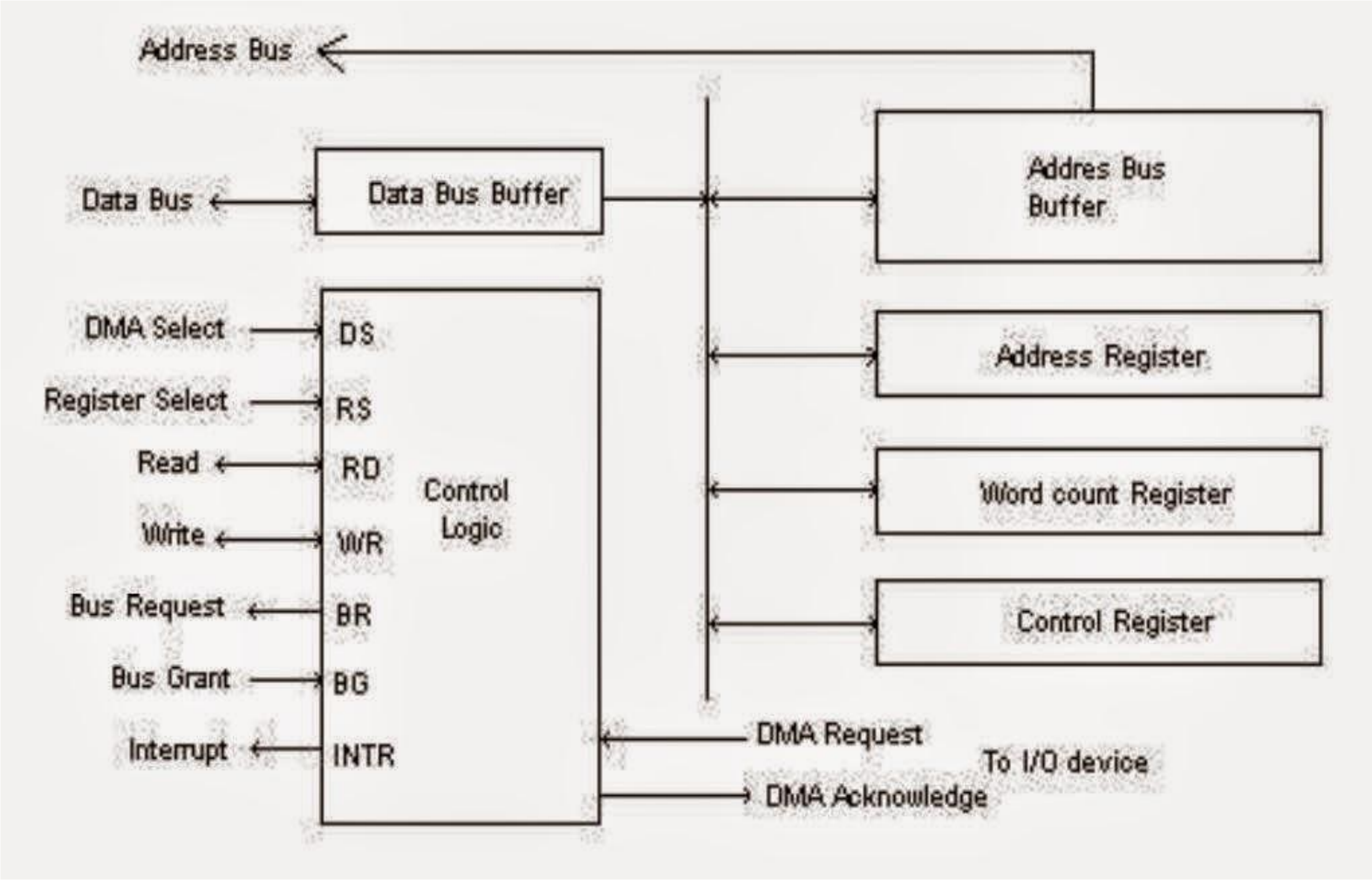


Fig: DMA Controller

- → Figure shows the block diagram of a typical DMA controller. The unit communicates with the MP via the data bus and control lines.
- → The registers in the DMA are selected by the MP through the address bus by enabling the DS (DMA select) and RS (Register Select) inputs. The RD (read) and WR (write) inputs are bidirectional.
- → When the bus grant (BG) input is 0, the MP can communicate with the DMA registers through the data bus to read from or write to the DMA registers. When BG=1, the processor does not have control over the system buses and the DMA can communicate directly with the memory by specifying an address in the address bus and activating the RD or WR control.
- → The DMA controller has three registers: an address register, a word count register and a control register.
- → The address register contains an address to specify the desired location in memory. The address bits go though bus buffers into the address bus. The address register is incremented after each word that is transferred to memory.
- → The word count register holds the number of words to be transferred. The register is decremented by one after each word transfer and internally tested for zero.
- → The control register specifies the mode of transfer.

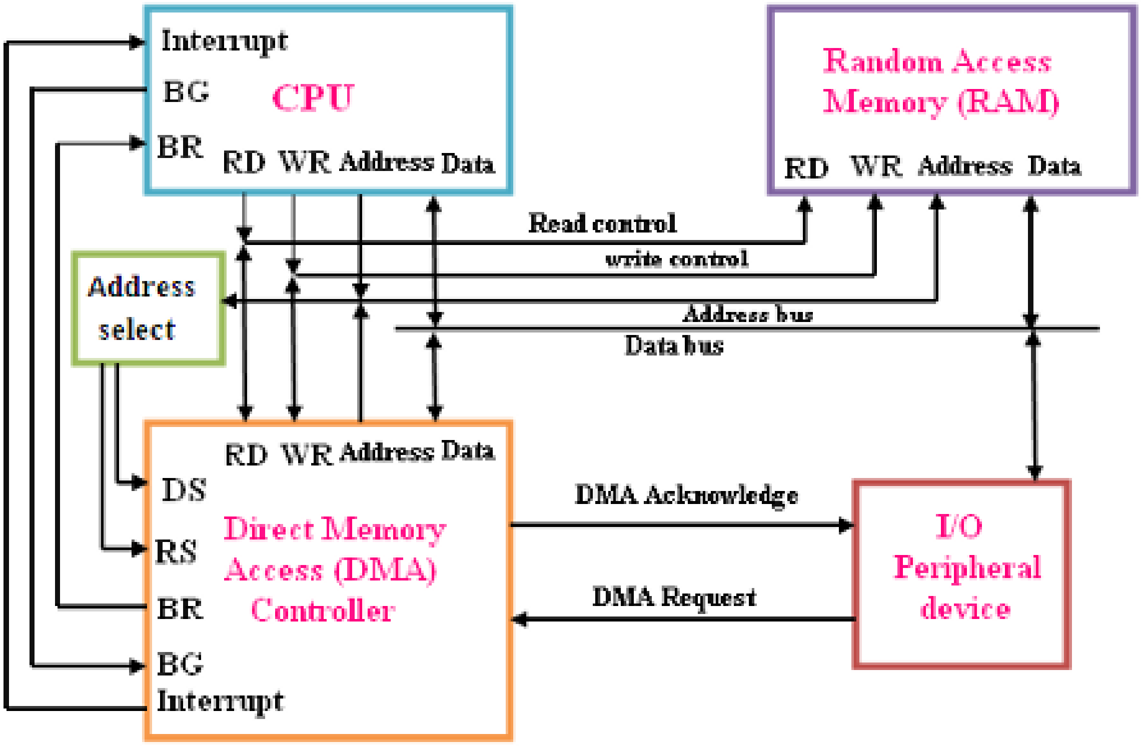


Fig: DMA Transfer

Interrupt

→ Interrupt is a process where an external device can get the attention of the microprocessor. The process starts from the I/O device. An interrupt is considered to be an emergency signal that may be serviced. The Microprocessor may respond to it as soon as possible.

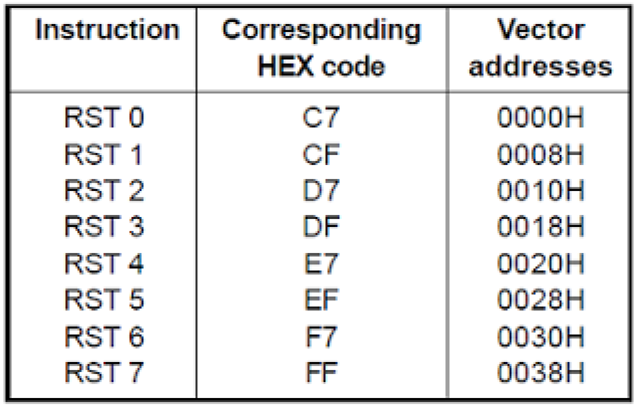
→ When the Microprocessor receives an interrupt signal, it suspends the currently executing program and jumps to an Interrupt Service Routine (ISR) to respond to the incoming interrupt. Each interrupt will most probably have its own ISR.

→ Responding to an interrupt may be immediate or delayed depending on whether the interrupt is maskable or non-maskable and whether interrupts are being masked or not. There are two ways of redirecting the execution to the ISR depending on whether the interrupt is vectored or non-vectored.

- **Vector Interrupt:** In this type of interrupt, Processor knows the address of Interrupt. In other word processor knows the address of interrupt service routine.
  The examples of vector interrupt are RST 7.5, RST 6.5, RST 5.5, TRAP.
- **Non-Vector Interrupt:** In this type of interrupt, Processor cannot know the address of Interrupt. It should give externally. In the device will have to send the address of interrupt service routine to processor for performing Interrupt.
  The example of Non-vector interrupt is INTR.

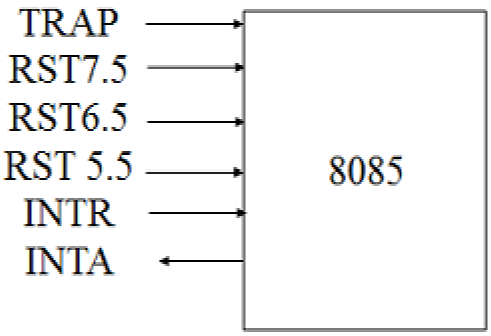
→ When a device interrupts, it actually wants the MP to give a service which is equivalent to asking the MP to call a subroutine. This subroutine is called ISR (Interrupt Service Routine).

Software Interrupt: It is an instruction based Interrupt which is completely controlled by software. That means programmer can use this instruction to execute interrupt in main program.
There are eight software interrupts available in 8085 microprocessor. See the example with their hex code and vector address.



Instruction	Corresponding HEX code	Vector addresses
RST 0	C7	0000H
RST 1	CF	0008H
RST 2	D7	0010H
RST 3	DF	0018H
RST 4	E7	0020H
RST 5	EF	0028H
RST 6	F7	0030H
RST 7	FF	0038H

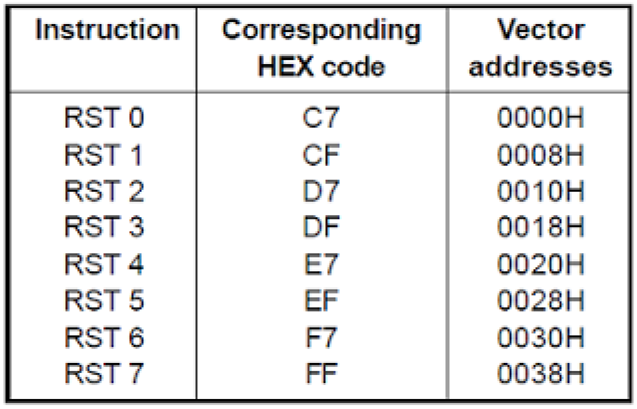
Hardware Interrupt: As name suggests it is interrupt which can get the interrupt request in hardware pin of microprocessor 8085. There are mainly six dedicated pins available for interrupt purpose.



Those are TRAP, RST 7.5, RST 6.5, RST 5.5, INTR, INTA (It is not an Interrupt pin but it is used to send acknowledgement of the Interrupt request getting from other interrupt pin.)

8085 Interrupt Pins and Interrupt Priority

There are five interrupt pins in 8085 and one interrupt acknowledge (INTA) pin.

<table>
<tr><th>Pin No.</th><th>Name</th><th>Type</th><th>Priority</th><th>ISR Location</th></tr>
<tr><td>6</td><td>TRAP</td><td>Vectored</td><td>Highest</td><td>CALL 0024H (3-byte call)</td></tr>
<tr><td>7</td><td>RST 7.5</td><td>Vectored</td><td rowspan="3">↓</td><td>CALL 003CH (3-byte call)</td></tr>
<tr><td>8</td><td>RST 6.5</td><td>Vectored</td><td>CALL 0034H (3-byte call)</td></tr>
<tr><td>9</td><td>RST 5.5</td><td>Vectored</td><td>CALL 002CH (3-byte call)</td></tr>
<tr><td>10</td><td>INTR</td><td>Non-Vectored</td><td>Lowest</td><td>RST (Restart instructions) – 1 byte call

<table>
<tr><th>Instruction</th><th>Corresponding HEX code</th><th>Vector addresses</th></tr>
<tr><td>RST 0</td><td>C7</td><td>0000H</td></tr>
<tr><td>RST 1</td><td>CF</td><td>0008H</td></tr>
<tr><td>RST 2</td><td>D7</td><td>0010H</td></tr>
<tr><td>RST 3</td><td>DF</td><td>0018H</td></tr>
<tr><td>RST 4</td><td>E7</td><td>0020H</td></tr>
<tr><td>RST 5</td><td>EF</td><td>0028H</td></tr>
<tr><td>RST 6</td><td>F7</td><td>0030H</td></tr>
<tr><td>RST 7</td><td>FF</td><td>0038H</td></tr>
</table></td></tr>
</table>

→ Pin 6 to pin 10 interrupts have the priorities from highest to lowest in decreasing order.

→ Priority means which interrupt gets the acknowledgement first if more than one are interrupting the microprocessor.

Maskable and Non-Maskable Interrupt

Maskable interrupts: An interrupt which can be disabled by software that means we can disable the interrupt by sending appropriate instruction, is called a maskable interrupt.
RST 7.5, RST 6.5, and RST 5.5 are the examples of Maskable Interrupt.

Non-Maskable interrupts: As name suggests we cannot disable the interrupt by sending any instruction is called Non Maskable Interrupt.
TRAP interrupt is the non-maskable interrupt for 8085. It means that if an interrupt comes via TRAP, 8085 will have to recognize the interrupt we cannot mask it.

Vectored and Polled Interrupt

Vectored Interrupt

In a computer, a vectored interrupt is an I/O interrupt that tells the part of the computer that handles I/O interrupts at the hardware level that a request for attention from an I/O device has been received and also identifies the device that sent the request.

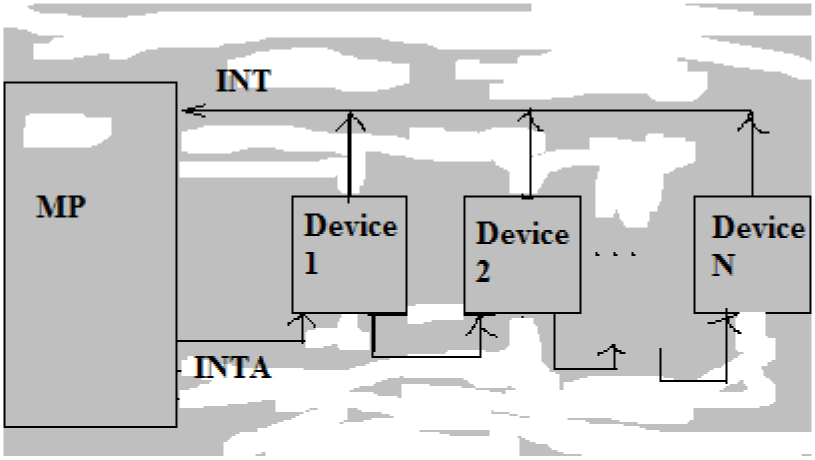


Fig: Vectored Interrupt

The device is connected in a chain as shown in figure above for setting up the priority systems. Suppose that one or more devices interrupt the processor at a time. In response, the processor saves its current status and then generates an interrupt acknowledge (INTA) signal to the highest priority device, which is device1, in this case. If this device has generated the interrupt, it will accept the INTA signal from the processor; otherwise, it will pass INTA on to the next device until INTA is accepted by the interrupting device.

Polled Interrupt

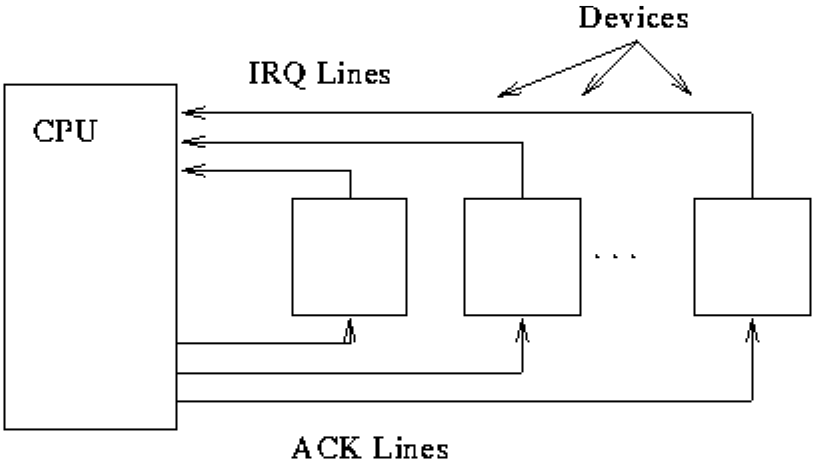


Fig: Polled Interrupt

In a computer, a polled interrupt is a specific type of I/O interrupt that notifies the part of the computer containing the I/O interface that a device is ready to be read or otherwise handled but does not indicate which device. The interrupt controller must poll (send a signal out to) each device to determine which one made the request.

Polled interrupts are handled using mostly software and are therefore slower compared to vectored (hardware) interrupts. The processor responds to an interrupt by executing one general service routine for all devices. The priority of these devices is determined by the order in which the routine polls each device. Once the processor determines the source of interrupt, it branches to the service routine for that device. The typical configuration of the polled interrupt is shown in figure above.

As shown in figure, several external devices (Device1, Device2,…….., Device N) are connected to a single interrupt line (INT) of the processor. When one or more devices activate the INT line high, the processor saves the content of the PC and other registers and then branches to an address defined by the manufacturer of the processor. The user can write a program at this address in order to poll each device starting with highest priority device in order to find the source of the interrupt.

Polled interrupts are very simple. But for a large number of devices, the time required to poll each device may exceed the time to service the device.

Programmable Interrupt Controller: The 8259A

The 8259A is a programmable interrupt-managing device, specifically designed for use with the interrupt signals (INTR/INT) of the 8085 MP.

The 8259A block diagram includes control logic, registers for interrupt requests, priority resolver, cascade logic, and data bus. The registers manage interrupt requests; the priority resolver determines their priority. The cascade logic is used to connect additional 8259A devices.

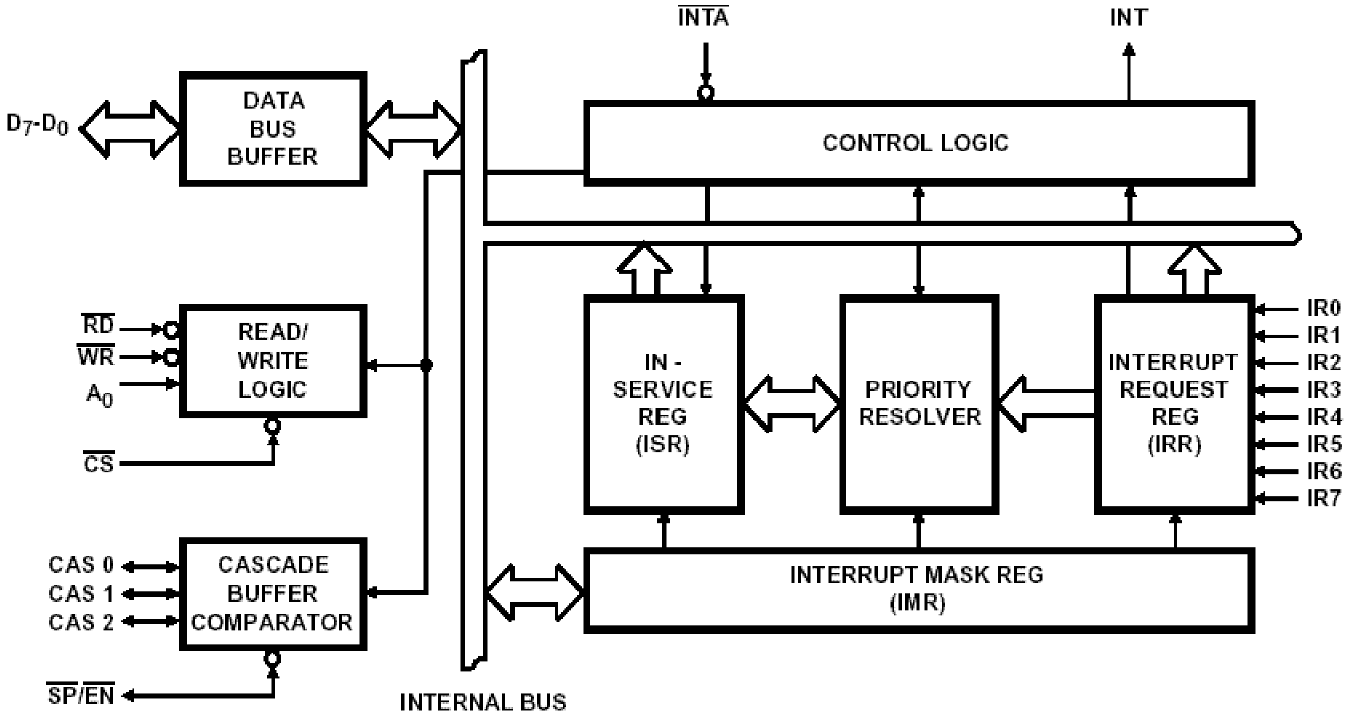


Fig: Block Diagram of 8259A PIC

The following steps take place during the operation of 8259A:

i) One or more interrupt request lines go high requesting the service.

ii) The 8259A resolves the priorities and sends an INT signal to the MP.

iii) The MP acknowledges the interrupt by sending INTA(bar).

iv) After the INTA(bar) has been received, the op-code for the call instruction (CDH) is placed on the data bus.

v) Because of the CALL instruction, the MP sends two more INTA(bar) signals.

vi) At the first INTA(bar), the 8259A places the low-order 8-bit address on the data bus and at the second INTA(bar), it places the high-order 8-bit address of the interrupt vector. This completes the 3-byte CALL instruction.

vii) The program sequence of the MP is transferred to the memory location specified by the CALL instruction.

Priority modes

i) Fully Nested mode

→ IR0 has the highest priority and the following IR1, IR2, IR3…… etc. have the decreasing priorities.

ii) Automatic rotation mode

→ First priority changes to the last after its service.

iii) Specific rotation mode

→ This is user selectable or programmable, which means priority can be selected by programming.

Features

i) It manages 8 interrupt requests.

ii) It can vector an interrupt request anywhere in the memory map through program control without additional hardware for restart instructions. However, all 8 requests are spaced at the interval of either 4 locations or 8 locations.

iii) It can solve 8 levels of interrupt priorities in a variety of modes.

iv) With cascading additional 8259A devices, the priority scheme can be expanded to 64 levels.

v) The 8259A has the abilities such as reading the status and changing the interrupt mode during a program execution.

vi) It can mask each interrupt request individually.

vii) It can be set up to work with either the 8085 MP mode or the 8086/8088 MP mode.