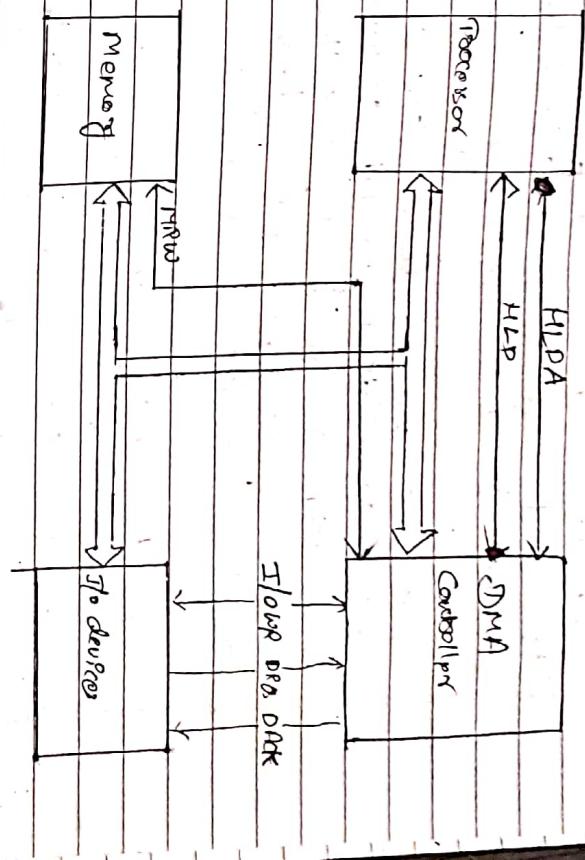


Chapters

PAGE : 1 / 1
DATE : 1 / 1

Explain the sequence of events that occurs during DMA operation



PAGE : 1 / 1
DATE : 1 / 1

Q. DMA

Direct memory access (DMA) is a feature of computer system that allows transfer of hardware subsystem to access main memory without involving the CPU is called direct memory access (DMA).

By using DMA technique a large amount of data is transferred between memory and the peripheral without generally impeding CPU Performance.

During the transfer of data the CPU has no access to control its own memory buses.

The DMA takes the control over the buses of CPU to manage the transfer of data directly without involving the CPU. DMA allows the CPU to do other high level processing on that time the DMA performs other works by the taking buses of CPU to the I/O devices directly from memory to the I/O devices or peripheral.

Fig: DMA Controllers Data Transfer

1 whenever an I/O device want to transfer the data to or from memory, it sends the DMA request (DRQ) to the DMA Controller. DMA controller accepts these DRQ and ask the CPU to hold for a few clock cycles by sending of the Hold request (HLI)

2- CPU receives the hold request (HLD) from DMA controller and ~~then~~ releases the bus and sends the hold acknowledgement (HLDA) to DMA controller.

3- After receiving the hold acknowledgement I/O device (DAC) that the data transfer can be performed and DMA controller takes the charge of the system bus and transfers the data to its own memory.

4- When the data transfer is completed, the processor that the task of the data transfer is finished and the processor can take control over the bus again and start processing where it has left.

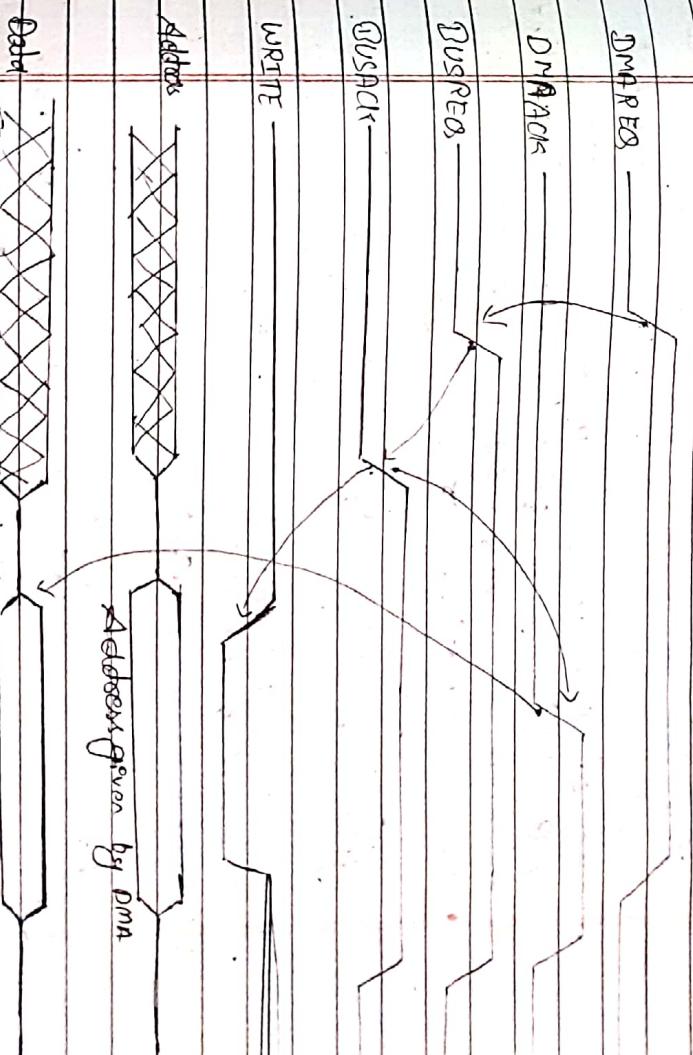


Fig: DMA Timing Diagram

- Many hardware systems use DMA including disk drives controllers, graphics cards, network cards, routers and sound card.
- DMAs also use for intra-chip data transfer in multi-core processor.

Computers that have DMA channels can transfer data to and from devices with much less CPU overhead than computer without DMA channels.

Interrupt

Interrupt : Interrupts are the signals generated by the external devices in response to an interrupt. The process stop what it is currently doing and make some sense when the execution of the source program is terminated.

The original process may resume its previous operations. Interrupts are important because they give the user better control over the computer.

Types of Interrupts

- 1) External interrupts
- 2) Internal interrupts
- 3) Software interrupts

External Interrupts

- a) Power interrupt
- b) Hardware interrupt

Internal Interrupts

- c) External interrupt
- d) Internal interrupt

External interrupt :- External interrupt are initiated via the interrupter's interrupt pins.

by external source. The device, timing device, circuit monitoring the power supply etc.

These are two types of external interrupt.

External interrupt :- In simple way

maskable interrupt is those type of interrupt which we can disable by writing some instruction in to the program.

Non-maskable interrupt : It is a interrupt we can't disable or ignore by writing instruction. This interrupt has higher priority over the maskable interrupt.

Internal interrupt : An internal interrupt is a type of interrupt that occurs from a specific event within the processor such as the occurrence of an error due to the memory.

by zero which produces an internal interrupt called divide by zero interrupt.

Software interrupt : Software interrupt is

initiated by executing an instruction. It can be used by the programmer to initiate an interrupt procedure at any desired point. The most common use of software is associated with a supervisor call instruction. It goes there are 8 software interrupt.

They are.

Vector and Non-Vectored Interrupts

Vectored Interrupt

In vectored interrupt the CPU or the processor actually know the address of the interrupt service routine where it is located or stored in the memory to execute the ISR. All it needs is the interrupt source sends its unique vector via a data bus and through its I/O device interface to the CPU. The CPU takes the vector, checks the interrupt table in memory and carries out correct ISR into that device. And precease the program flow where the interrupt initially occurred.

Vectored Interrupt
are those which have fixed vector addresses. Starting address of sub-routine and offset preceding these program control is transferred to stack address.

Non-Vectored Interrupt

are those in which vector address is not predefined. The interrupting device gives the address of sub-routine. These interrupts INT 10H is the only non-vectored interrupt in 8088 microprocessor.

Interrupt	Vector address
TRAP (4)	0020H
RST 5:5	0021H
RST 6:6	0022H
RST 7:7	0023H

for Software Interrupts vector address are given by

Interrupt	Vector Address
RST 0	0000H
RST 1	0008H
RST 2	0010H
RST 3	0018H
RST 4	0020H
RST 5	0028H
RST 6	0030H
RST 7	0038H

Ques

"Interrupt based I/O is efficient compared to Polling I/O" Justify.

- * Pollled Interrupt
- * Chained (Vectorized) Interrupt

Interrupt Priority

When interrupt is a signal or condition that causes the processor to stop the current execution (Save status of execution) and do the instruction like do the source the signal or condition. The signal and condition is come from external devices and internal device send the signal to the processor to stop the current execution.

The data transfers between the CPU and peripherals are initiated by the CPU. CPU don't transfer the data data unless the peripherals are ready.

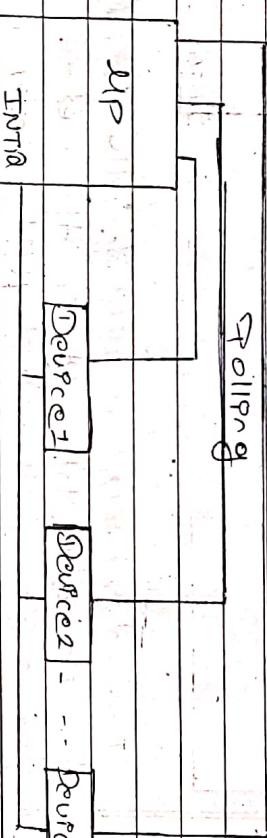
Main task of

Interrupt system is to identify the source of interrupt.

There is a possibility to suspend device send request for the communication simultaneously. Then the system must also decide which device to serve first.

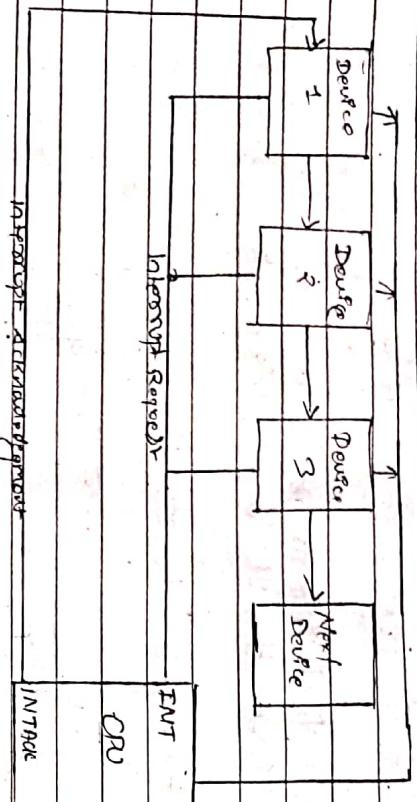
High priority interrupt are serve first then the lower priority interrupt. Device with high transfer speed such as memory has higher priority and slower device such as keyboard receive low priority.

There are mainly two ways to solving multiple interrupts. These are,



Pollled Interrupts are handle by using software which is slower than hardware. The processor generate the common interrupt source routine for all the devices. Several devices are connected to a single line INTR of MP. When INTR goes up the processor saves the contents of PC and registers and then branches to an address defined by the manufacturers of the processor the user can write a program to find the source of the interrupt by starting the polled from highest priority device.

* Chained Interrupt

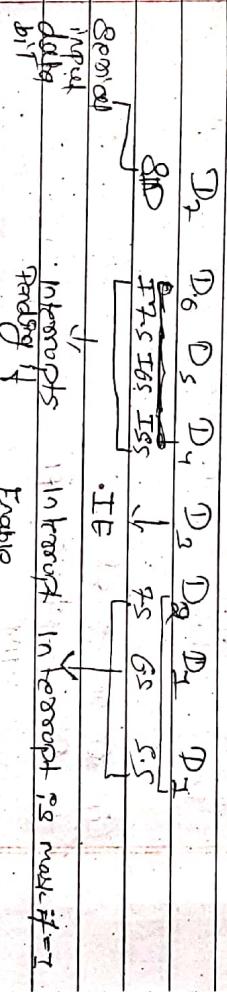


Second highest priority go on.

* Read Instruction Mask (RIM)

- * Read interrupt mask.
- * 1 byte instruction
- * This is multipurpose instruction used to read the status of interrupt F.S, G.S and S.P. data input bit.

* This instruction loads eight bit data of accumulator with following interrupt operations

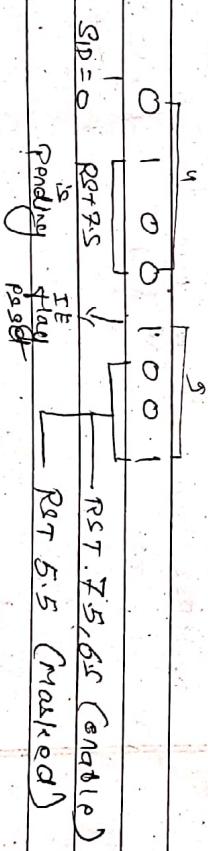


Chained interrupt is used for the handling the priority interrupt. In this method, the device which is highest priority is placed first position followed by lower priority devices. All the devices connected on serial method.

In this method when the

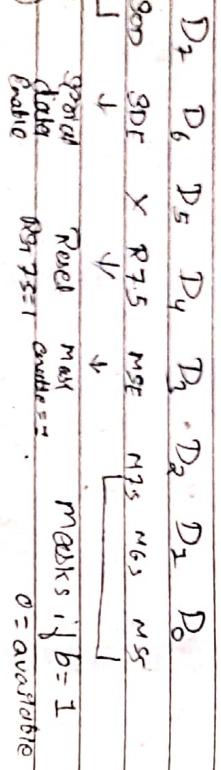
I/O devices generate interrupt then INT becomes high (1), then CPU doesn't know which device generate the interrupt so CPU make INTRACK high(1)-Then if the device of highest priority generate thus interrupt then it consume thus INTRACK signal and goes (0) for other devices. Then vectored address provided by this device before checking the ISR if highest priority device doesn't generate the interrupt when it check

* Accumulator data A = 49 H



SET Interrupt mask (SM) instruction

- This is 1 byte instruction
- The instruction is used to implement interrupts (RET F.S, B.S, S.S) and general data output
- This instruction implements following data of accumulator (read instruction of accumulator)



By ~~microprocessor~~ ~~microcontroller~~

SM

SM = I



Op code: 00001110
Reset Mask is enable
L.RST F.S, B.S (mask)
L.RST S.S (enable)

- It can solve eight levels of interrupt priorities in a variety of modes.

8259A Programmable Interrupt Controller



Basic of 8259A

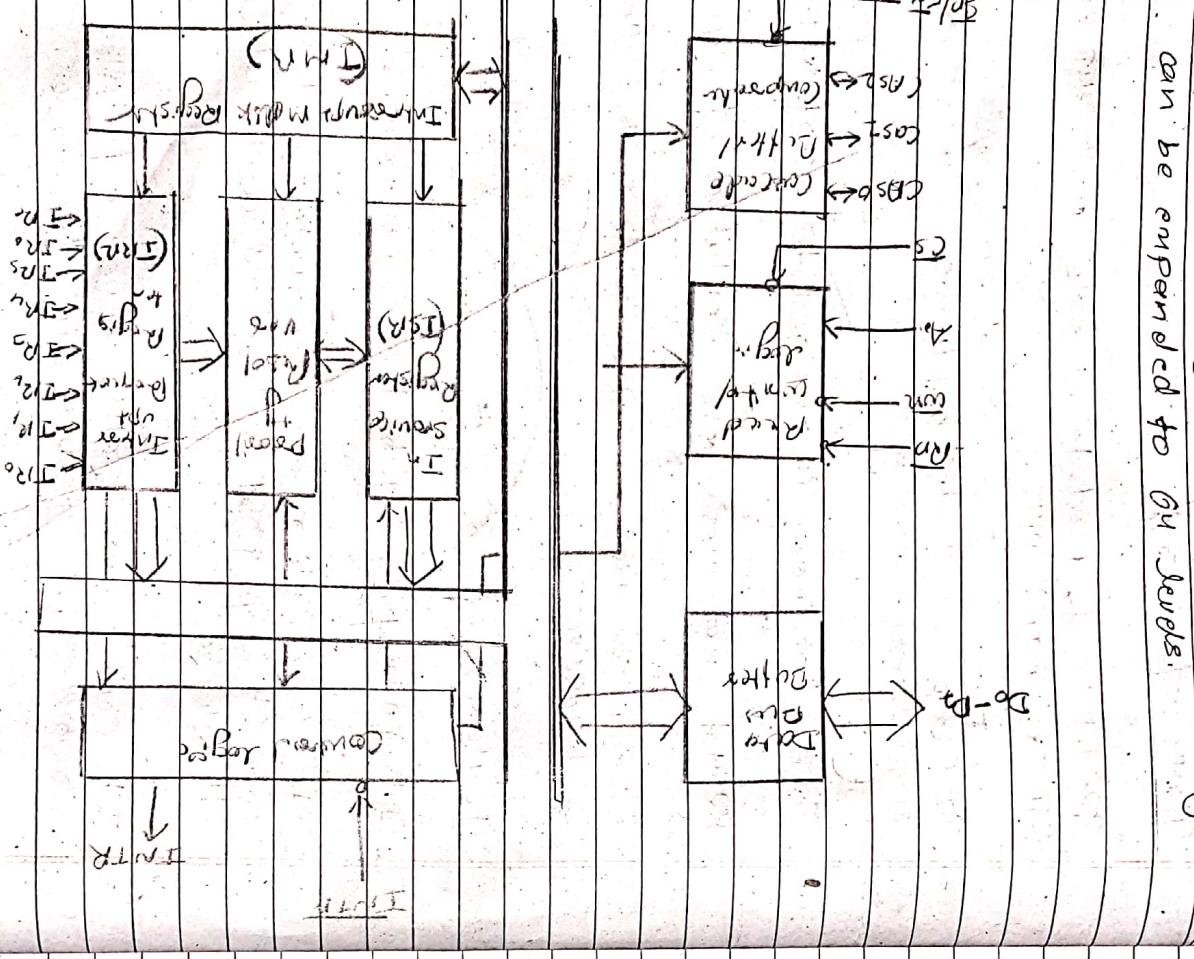
- 8259A is designed to operate with MPP 8085 with (INTR/INTA) pins.
- 8085 chip has limited hardware Interrupts with limited interrupt priority.
- Extra hardware is required to insert RET instruction
- 8259A overcomes these limitations of MPP. 8085.
- 8259A can be employed with 16-bit Intel MPP as 8086/8088.

Features of 8259A

- It manages eight interrupt requests.

- It can vector an interrupt anywhere on memory map through program control without additional hardware.

with additional 8259A devices, the priority scheme can be expanded to 64 levels.



Operation of 8259A

DATE: / /

- 2) Then the IEP acknowledges the interrupt by INTR.
- 3) After IEP sends INTR signal, opcode for CALL instruction is placed on data bus.
- 4) Due to CALL instruction IEP(8085) sends two more INTR sig. nls. during that
- 5) 1st INTR: 8259A placed I/O device address on data bus and at 2nd INTR, 8259A placed higher address on data bus.
- 6) Program sequence is now transferred to location specified by CHAL instruction and it will issue that particular interrupt.
- 7) After execution of that particular interrupt again program control will come back to original position to resolve the priority.