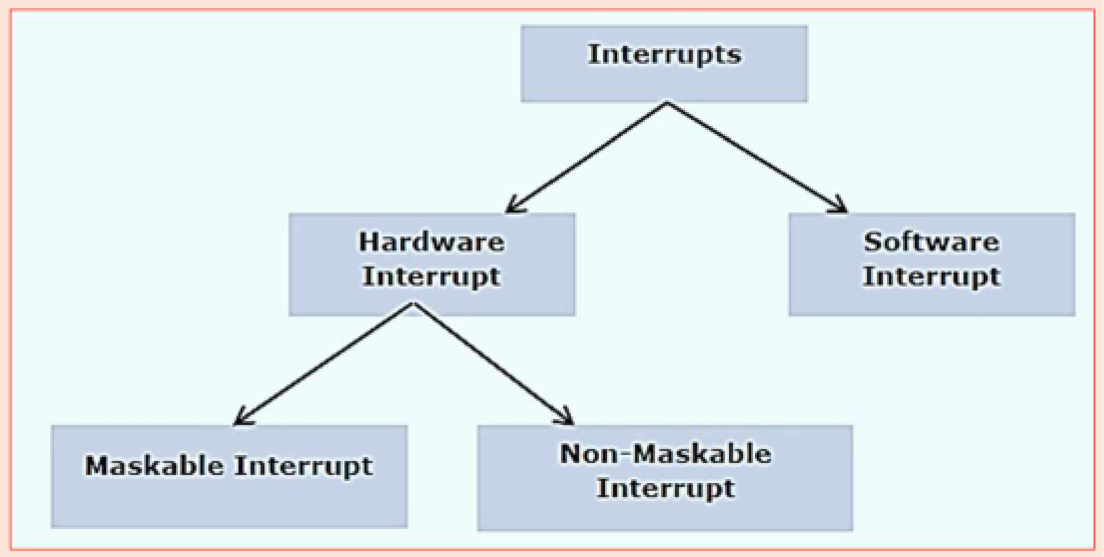
Microprocessors Module 3 Questions

Part A

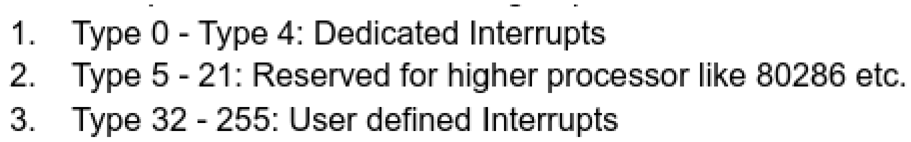
1. Explain how the INT n instruction finds the starting address of its interrupt service routine in IVT.

- The INT n instruction is an assembly language instruction to generate a software interrupt. This instruction causes the processor to temporarily halt its current execution and transfer control to a specific interrupt service routine (ISR) or handler.
- The vector number "n" in the `INT n` instruction represents the specific interrupt or exception that the instruction is intended to invoke
- The following steps are done to find the starting address
  - The interrupt type or vector number 'n' is sent to the CPU via data bus.
  - The contents of the flag register are pushed onto the stack.
  - The contents of the Code segment (CS) register and instruction pointer (IP) are pushed onto the stack.
  - The new values of IP and CS for the Interrupt Service Routine (ISR) is fetched from the Interrupt Vector
    - loading IP value from word location (n x 4) of the IVT
    - loading CS value from word location (n x 4 + 2) of the IVT
  - The ISR is executed.
  - The content of flag register, CS and IP is popped from the stack and control is returned to resume the execution.

2. Classify various categories of interrupts available in 8086.



- Hardware Interrupts
  - Hardware interrupts are generated by an external hardware module. (Eg: key-press)
  - It can be divided into two:
    1. Maskable: These interrupts can be masked (disables) or enabled by the processor
    2. Non-Maskable: These are high priority interrupts which cannot be ignored by the processor
- Software Interrupts:
  - Software interrupts are generated by software instructions
  - There are 256 interrupts each of which is invoked using INT n, where n is the type number.
  - These interrupts can be divided to 3 groups



1. Type 0 - Type 4: Dedicated Interrupts
2. Type 5 - 21: Reserved for higher processor like 80286 etc.
3. Type 32 - 255: User defined Interrupts

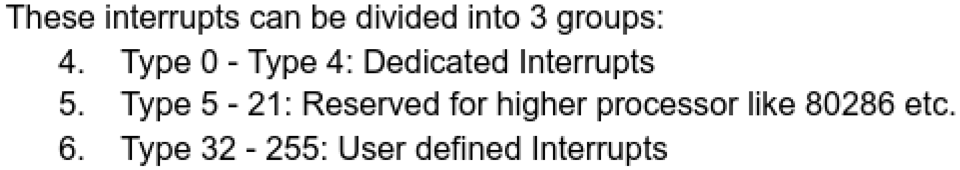
3. Write notes on the following based on 8086: a. software interrupt b. hardware interrupt c. nested interrupt

Hardware Interrupts

- Hardware interrupts are generated by an external hardware module. (Eg: key-press)
- Divided to two
  - Maskable: These interrupts can be masked (disables) or enabled by the processor
  - Non-Maskable: These are high priority interrupts which cannot be ignored by the processor

Software Interrupts

- Software interrupts are generated by software instructions and operate similarly to jump or branch instruction
- There are 256 interrupts each of which is invoked using INT n, where n is the type number.

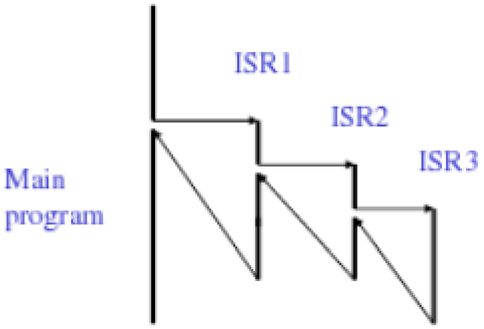


These interrupts can be divided into 3 groups:

4. Type 0 - Type 4: Dedicated Interrupts
5. Type 5 - 21: Reserved for higher processor like 80286 etc.
6. Type 32 - 255: User defined Interrupts

Nested Interrupts

- Nested interrupts in 8086 refer to situations where an interrupt occurs while the processor is already handling another interrupt.
- This can lead to a chain of interrupt service routines.



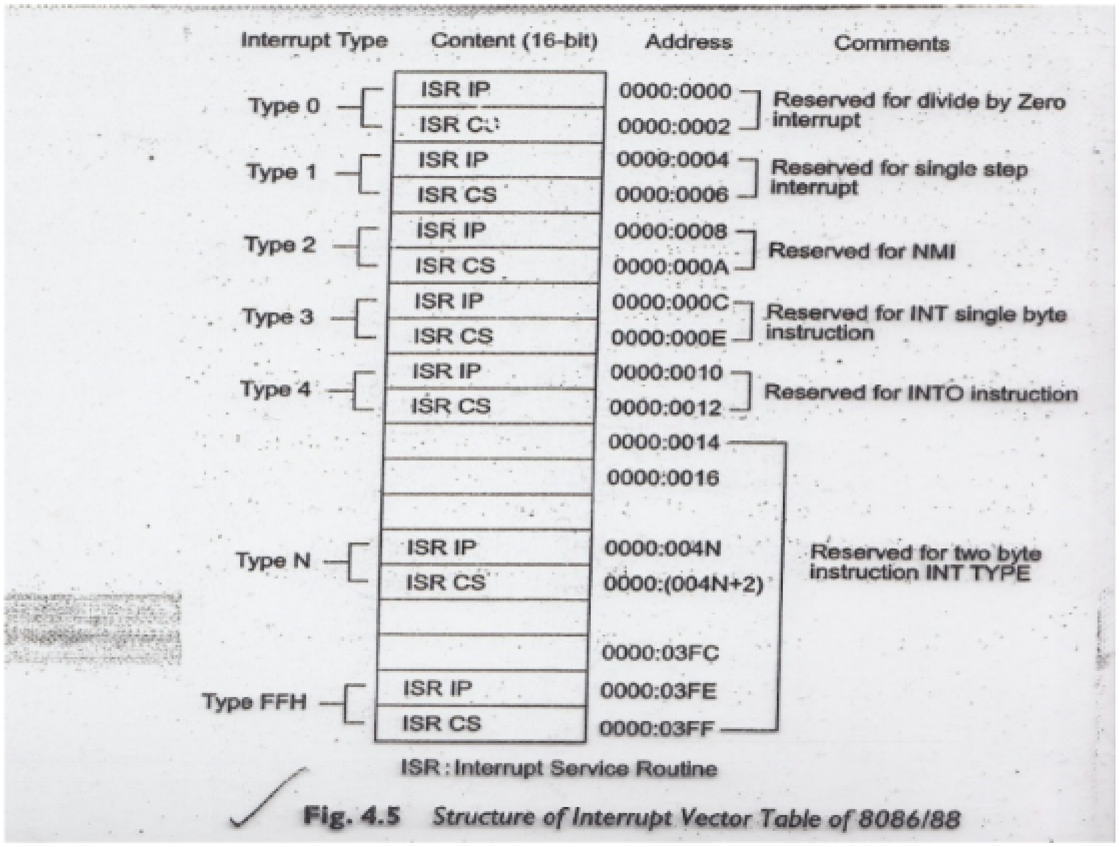
Part B

1. Explain the stack structure of 8086.

- Stack is a block of memory that may be used for temporarily for storing contents of a register inside the CPU.
- The stack is a block of memory that is accessed using SP and SS registers. The stack works in LIFO (Last In First Out) manner
- The stack pointer is a 16 bit register that contains the offset of the address that lies in the stack segment.
- The stack segment has maximum 64 Kbytes locations, and thus may overlap with other segments.
- The stack segment register contains the base address of the stack in the memory.
- The stack segment register and stack pointer register together address the stack top.
- Each push operation decrements the stack pointer
- while each pop operation increments the stack pointer

2. State the purpose of Interrupt Vector Table of 8086 and explain its structure

- For every interrupt, there must be an interrupt service routine (ISR), or interrupt handler. When an interrupt is invoked, the microprocessor runs the interrupt service routine.
- For every interrupt, there is a fixed location in memory that holds the address of its ISR. The group of memory locations set aside to hold the addresses of ISRs is called the interrupt vector table.
- When an interrupt occurs, the microprocessor stops execution of current instruction. It transfers the content of the program counter (CS and IP) into stack.
- After this, it jumps to the memory location specified by Interrupt Vector Table (IVT). After that the code(ISR) written on that memory area will execute
- The interrupt vector (or interrupt pointer) table is the link between an interrupt type code and the procedure that has been designated to service interrupts associated with that code.
- 8086 supports a total 256 types, i.e., 00H to FFH.
    - The first 1k Byte of memory of 8086 (00000 to 003FF) is set aside as a table for storing the starting addresses of Interrupt Service Procedures(ISP).



Interrupt Type	Content (16-bit)	Address	Comments
Type 0	ISR IP	0000:0000	Reserved for divide by Zero interrupt
	ISR CS	0000:0002	
Type 1	ISR IP	0000:0004	Reserved for single step interrupt
	ISR CS	0000:0006	
Type 2	ISR IP	0000:0008	Reserved for NMI
	ISR CS	0000:000A	
Type 3	ISR IP	0000:000C	Reserved for INT single byte instruction
	ISR CS	0000:000E	
Type 4	ISR IP	0000:0010	Reserved for INTO instruction
	ISR CS	0000:0012	
		0000:0014	Reserved for two byte instruction INT TYPE
		0000:0016	
Type N	ISR IP	0000:004N	
	ISR CS	0000:(004N+2)	
		0000:03FC	
Type FFH	ISR IP	0000:03FE	
	ISR CS	0000:03FF	

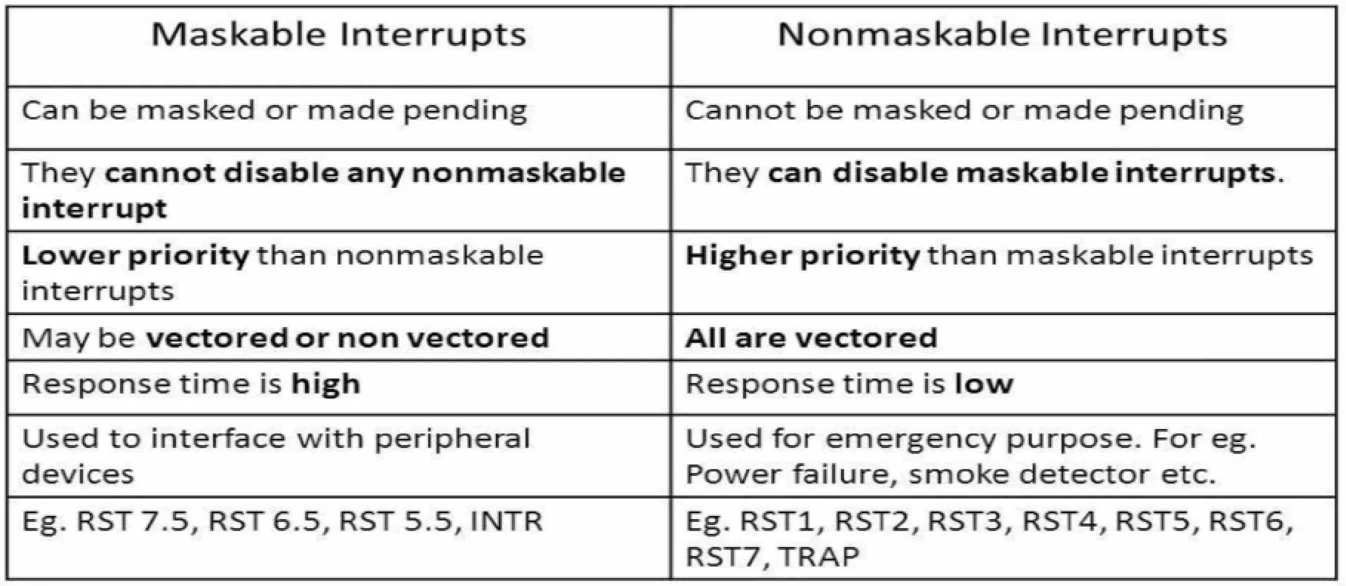
ISR : Interrupt Service Routine

Fig. 4.5 *Structure of Interrupt Vector Table of 8086/88*

3. Explain the interrupt cycle of 8086.

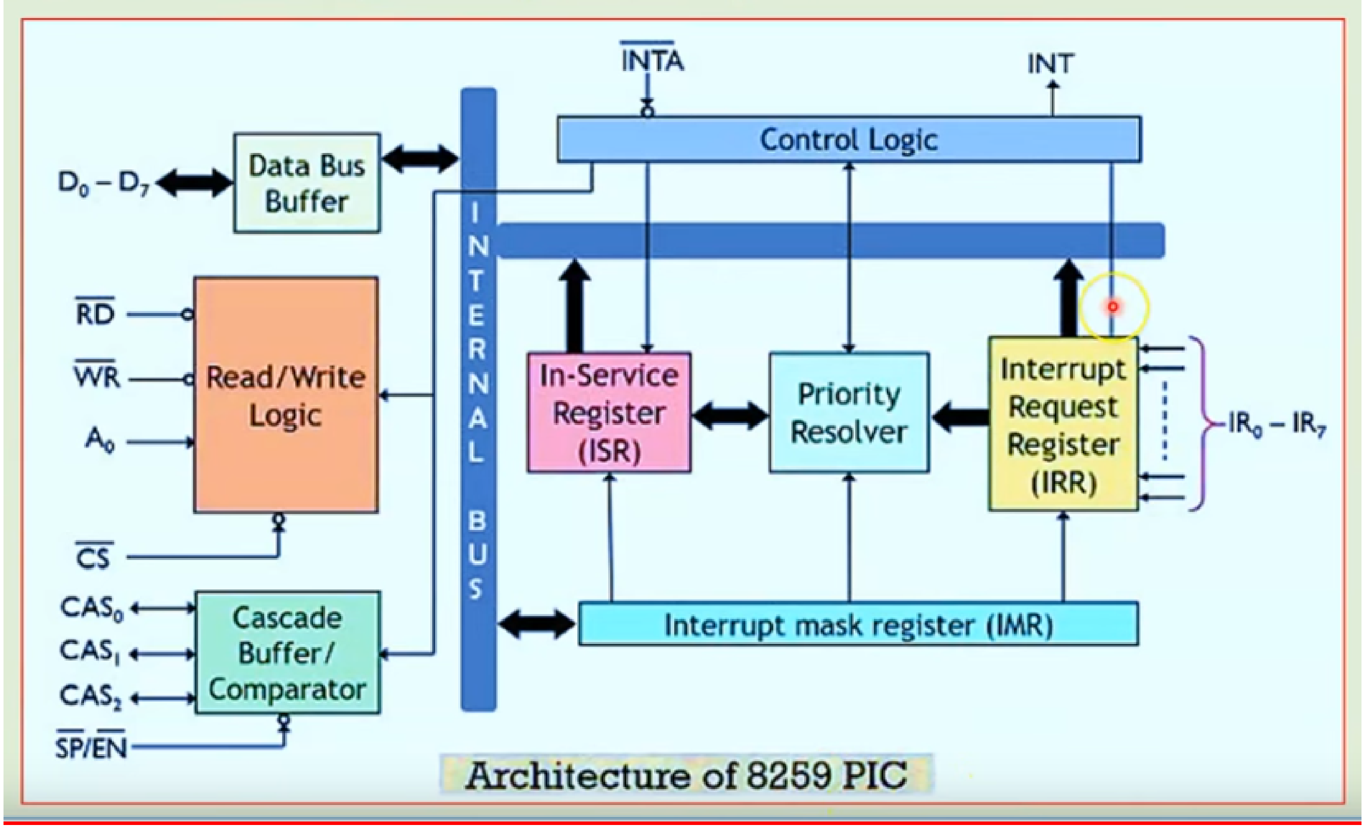
- External interface sends an interrupt signal, to the Interrupt Request (INTR) pin, or an internal Interrupt occurs.
- The CPU finishes the present instruction (for a hardware interrupt) and sends Interrupt acknowledge (INTA) to the hardware interface.
- The interrupt type N is sent to the Central Processor Unit (CPU) via the Data bus from the hardware interface.
- The contents of the flag registers are pushed onto the stack.
- Both the interrupt (IF) and Trap (TF) flags are cleared.
- The contents of the code segment register (CS) are pushed onto the Stack.
- The contents of the instruction pointer (IP) are pushed onto the Stack.
- The interrupt vector contents are fetched, from word location (4 x N) and then placed into the IP and from (4 x N+2) into the CS so that the next instruction executes at the interrupt service procedure addressed by the interrupt vector
- While returning from the interrupt-service routine by the Interrupt Return (IRET) instruction, the IP,CS and Flag registers are popped from the Stack and return to their state prior to the interrupt

4. Differentiate maskable and non-maskable interrupts in 8086.



Maskable Interrupts	Nonmaskable Interrupts
Can be masked or made pending	Cannot be masked or made pending
They **cannot disable any nonmaskable interrupt**	They **can disable maskable interrupts**.
Lower priority than nonmaskable interrupts	**Higher priority** than maskable interrupts
May be **vectored or non vectored**	**All are vectored**
Response time is **high**	Response time is **low**
Used to interface with peripheral devices	Used for emergency purpose. For eg. Power failure, smoke detector etc.
Eg. RST 7.5, RST 6.5, RST 5.5, INTR	Eg. RST1, RST2, RST3, RST4, RST5, RST6, RST7, TRAP

5. Draw the architectural block diagram of 8259A and explain the function of each Block.



Architecture of 8259 PIC

- 8259 is an IC using for managing multiple interrupts for 8086 microprocessor
- It is also known as programmable interrupt controller or priority interrupt controller
- The following are the internal functional units of 8259 programmable interrupt controller(PIC):
  - Interrupt Request Register (IRR)
    - In single mode, 8 different I/O devices (Maximum) can be connected with 8 interrupt request pins IR0 - IR7
    - In cascaded mode, there are one master 8259 and 8 slave 8259’s are connected together
  - Interrupt Mask Register (IMR)
    - Based on the requirement, certain interrupts can be disabled (masked). In order to disable the different 8 interrupts, the respective bit of the IMR needs to be set.
  - In-Service Register (ISR)
    - In-service register stores the status of the interrupt which is currently being executed
  - Priority Resolver
    - Priority resolver unit will decide which interrupt needs to be prioritized
    - The inputs to priority resolver are IMR, IRR, and ISR
  - Data Bus Buffer
    - Data bus buffer holds the vector no. of different interrupts before transferring to 8086 through (Data bus). It also holds the commands sent by 8086

        microprocessor

    - Read/Write Logic
        - Read/write logic block initiates different read and write operations
    - Cascade buffer / comparator
        - Cascade pins of cascade buffer / comparator will function only in cascaded mode, In cascaded mode, the master 8259 send the ID of slave 8259 through CAS Pins (CAS0, CAS1, CAS2)
    - Control Logic
        - Control logic sends interrupt signal to the 8086 microprocessor
    -
- 8259 will function in two modes:
    - Single mode ⇽ Only one 8259
    - Cascaded mode