CST 309 - MPMC 🐱-2 Important Questions

https://vivekkj.in & https://www.instagram.com/derick__davies/

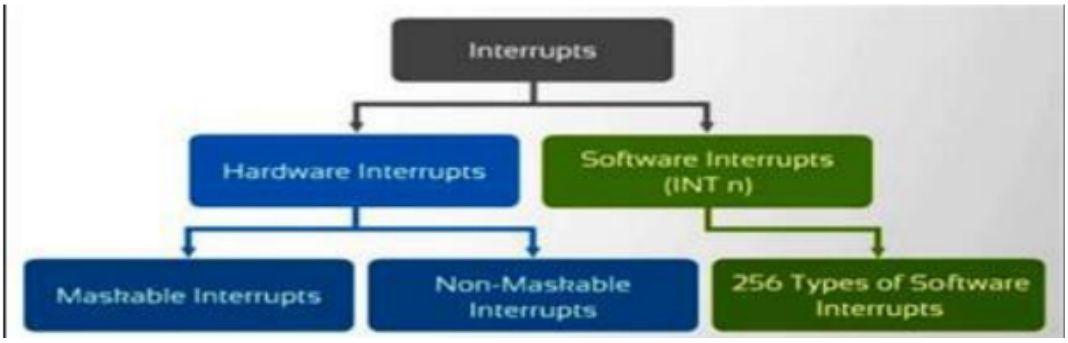
PART - A

1. **Explain how the INT n instruction finds the starting address of its interrupt service routine in IVT.** **(Dec 2021) (Module 3)**

When an INT n instruction is executed:

1. The interrupt type or vector number 'n' is sent to the CPU via data bus.
2. The contents of the flag register are pushed onto the stack.
3. The contents of the Code segment (CS) register and instruction pointer (IP) are pushed onto the stack.
4. The new values of IP and CS for the Interrupt Service Routine (ISR) is fetched from the Interrupt Vector Table (IVT) by:
    - loading IP value from word location (n*4) of the IVT
    - loading CS value from word location (n*4 + 2) of the IVT
5. The ISR is executed.
6. 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.** **(Dec 2021) (Module 3)**



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 (Eg: memory parity or bus faults)

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:
    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

The following is the answer in the answer scheme provided, please verify the same:

- **Ans:** TYPE 0 interrupts represents division by zero situation.
- TYPE 1 interrupt represents single-step execution during the debugging of a program.
- TYPE 2 interrupt represents non-mask able NMI interrupt.
- TYPE 3 interrupt represents break-point interrupt.
- TYPE 4 interrupt represents overflow interrupt.

3. **What is an interrupt vector table? Explain its structure in 8086** **(Dec 2022) (Module 3)**

See Qn. 14(b) for detailed answer

4. **Write notes on the following based on 8086:** **(Dec 2022) (Module 3)**
   a. software interrupt
   b. hardware interrupt
   c. nested interrupt

Hardware Interrupts:

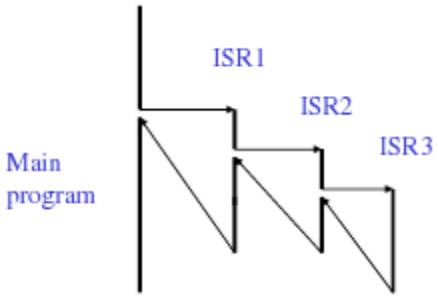
- Hardware interrupts are generated by an external hardware module. (Eg: key-press)
- It can be divided into two:
  3. Maskable: These interrupts can be masked (disables) or enabled by the processor
  4. Non-Maskable: These are high priority interrupts which cannot be ignored by the processor (Eg: memory parity or bus faults)

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:

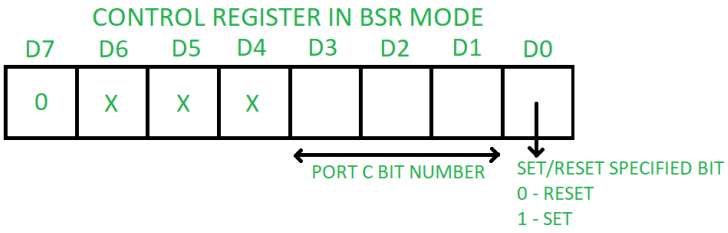
- Nested interrupts in 8086 refer to situations where an interrupt occurs while the processor is already handling another interrupt.
- This can lead to a chain of interrupt service routines.



5. **Interpret the mode and configurations of 8255 after its control word register is loaded with 86H.** **(Dec 2021) (Module 4)**

(86)H = (0101 0110)B

- The most significant bit (D7) is 0, which implies the 8255 operates in Bit Set-Reset (BSR) Mode.
- This mode is used to set or reset the bits of port C only, and is selected when the most significant bit (D7) in the control register is 0.
- The configuration in BSR mode for 8255 is as follows:



- The bits D6, D5 and D4 do not introduce any effects on its operation irrespective of their bit values in BSR mode.
- Here D3D2D1 = 011 = 4 (in decimal) which implies that PC4 bit is to be either set or reset.
- Since D0 = 0, PC4 will be reset to 0.

6. **Explain the features of 8257 DMA controller.** **(Dec 2021) (Module 4)**

(Any 6 points would be sufficient)

- It has four channels which can be used over four I/O devices.
- Each channel has a 16-bit address and 14-bit counter.
- Each channel can transfer data up to 64kb.
- Each channel can be programmed independently.
- Each channel can perform read transfer, write transfer and verify transfer operations.
- It generates MARK signal to the peripheral device that 128 bytes have been transferred.
- It requires a single-phase clock.
- Its frequency ranges from 250Hz to 3MHz.
- It operates in 2 modes, i.e., Master mode and Slave mode.

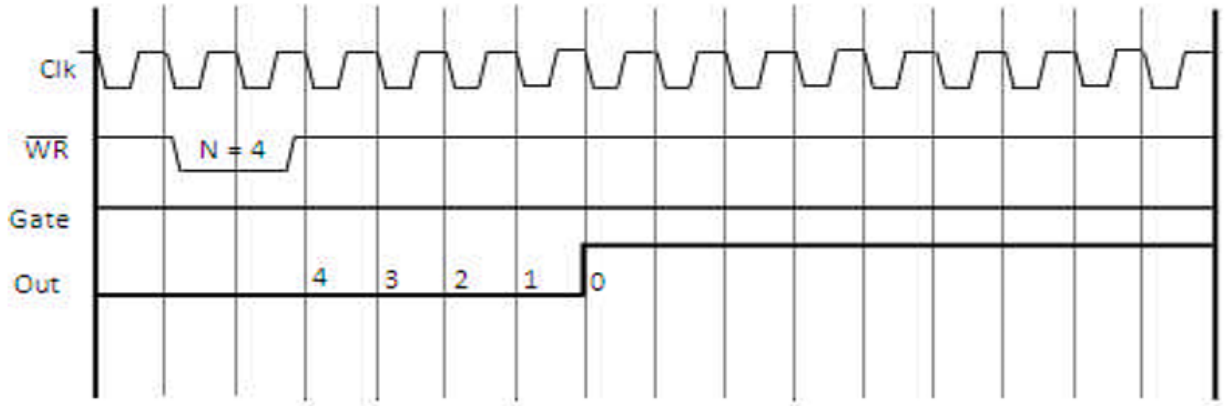
7. **Write the function of the following control signals in 8255.** **(Dec 2022) (Module 4)**
RD, WR, Ao, Ai, RESET, CS

- **CS**: Chip Select is an active low pin which selects the chip and enables communication between 8255 and CPU.

- **WR**: WR (write) is an active low pin that enables the microprocessor to write into a selected I/O port or control register.
- **RD**: RD (Read) is an active low pint that enables the microprocessor to read data from the selected I/O port of 8255.
- **A0 and A1**: These pins work in conjunction with CS to select the port as shown:

CS	A_1	A_0	Result
0	0	0	PORT A
0	0	1	PORT B
0	1	0	PORT C
0	1	1	Control Register
1	X	X	No Selection

8. Draw and explain the operational waveform of 8254 in MODE 0 operation. (Dec 2022) (Module 4)



- Mode 0 is used for event counting.
- This mode is selected if (D3 D2 D1) of the control word register is (0 0 0)
- When this mode is selected, the OUT pin is initially low.
- The GATE pin is made high, so counting is enabled.
- The count value is loaded, and the counting starts after the next high to low transition of the clock pulse.
- During counting, the OUT pin remains low.
- On terminal count (0), the OUT pin goes high, and it remains high until a new count is written into the counter.
- During counting, if the gate goes low then counting is terminated and the current count is latched till the GATE goes high again.

9. Differentiate between Microprocessors and Microcontrollers. (Dec 2021) (Module 5)

Microprocessor	Microcontroller
CPU is standalone while RAM, ROM, I/O, Timer are separate	CPU, RAM, ROM, I/O and Timer are all on a single chip
Designer can decide on the amount of ROM, RAM, and I/O Ports	Fixed amount of on-chip ROM, RAM, I/O Ports

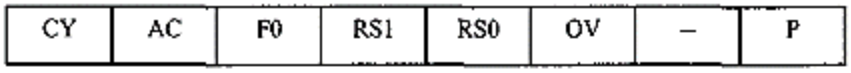
Expansive	Not Expansive
Versatile	Single Purpose
General Purpose (generally used in personal computers.)	Special Purpose (used in washing machines, and air conditioners.)
Expensive	Cheaper

10. List the IO ports available in 8051. **(Dec 2021) (Module 5)**

- There are total of 4 ports for I/O operation in 8051 microcontroller
- The four ports are P0,P1,P2, and P3. Each use 8 pins, to make the port 8 bit ports.
    - Port 0: Used as i/o pins and serves as lower order addresses in external memory interfacing
    - Port 1: Used as i/o pins
    - Port 2: Used as i/o pins and serves as higher order addresses in conjunction with port 0.
    - Port 3: Used as i/o pins + performing dual functions like TXD, INT0, INT1, WR, RD etc.
- When '0' is written to a port, it becomes an o/p port
- To use as an i/p port '1' must sent to the port

11. Draw and explain the format of program status word in 8051. **(Dec 2022) (Module 5)**

The Program Status Word (PSW) is a key register in the 8051 microcontroller that holds various status bits indicating the current state of the processor. The PSW is an 8-bit register, and its format is as follows:



CY	PSW.7	Carry flag.
AC	PSW.6	Auxiliary carry flag.
F0	PSW.5	Available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1.
RS0	PSW.3	Register Bank selector bit 0.
OV	PSW.2	Overflow flag.
–	PSW.1	User-definable bit.
P	PSW.0	Parity flag. Set/cleared by hardware each instuction cycle to indicate an odd/even number of 1 bits in the accumulator.

RS1	RS0	Register Bank	Address
0	0	0	00H - 07H
0	1	1	08H - 0FH
1	0	2	10H - 17H
1	1	3	18H - 1FH

12. Write an assembly language program for 8051 to compute x to the power n where both x and n are 8-bit numbers given by user and the result should not be more than 16 bits.

(Dec 2022) (Module 5)

(Will ask sir tomorrow)

PART - B

Module - 3

13. a) Explain the stack structure of 8086. **(4) (Dec 2021)**

- 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.

Suppose, a main program is being executed by the processor. At some stage during the execution of the program, all the registers in the CPU may contain useful data. In case there is a subroutine CALL instruction at this stage, there is a possibility that all or some of the registers of the main program may be modified due to the execution of the subroutine. This may result in loss of useful data, which may be avoided by using the stack. At the start of the subroutine, all the registers' contents of the main program may be pushed onto the stack one by one. After each PUSH operation SP will be modified as already explained before. Thus all the registers can be copied to the stack. Now these registers may be used by the subroutine, since their original contents are saved onto the stack. At the end of the execution of the subroutine, all the registers can get back their original contents by popping the data from the stack. The sequence of popping is exactly the reverse of the pushing sequence. In other words, the register or memory location that is pushed into the stack at the end should be popped off first.

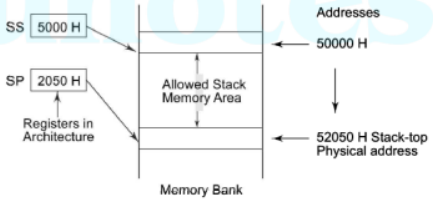
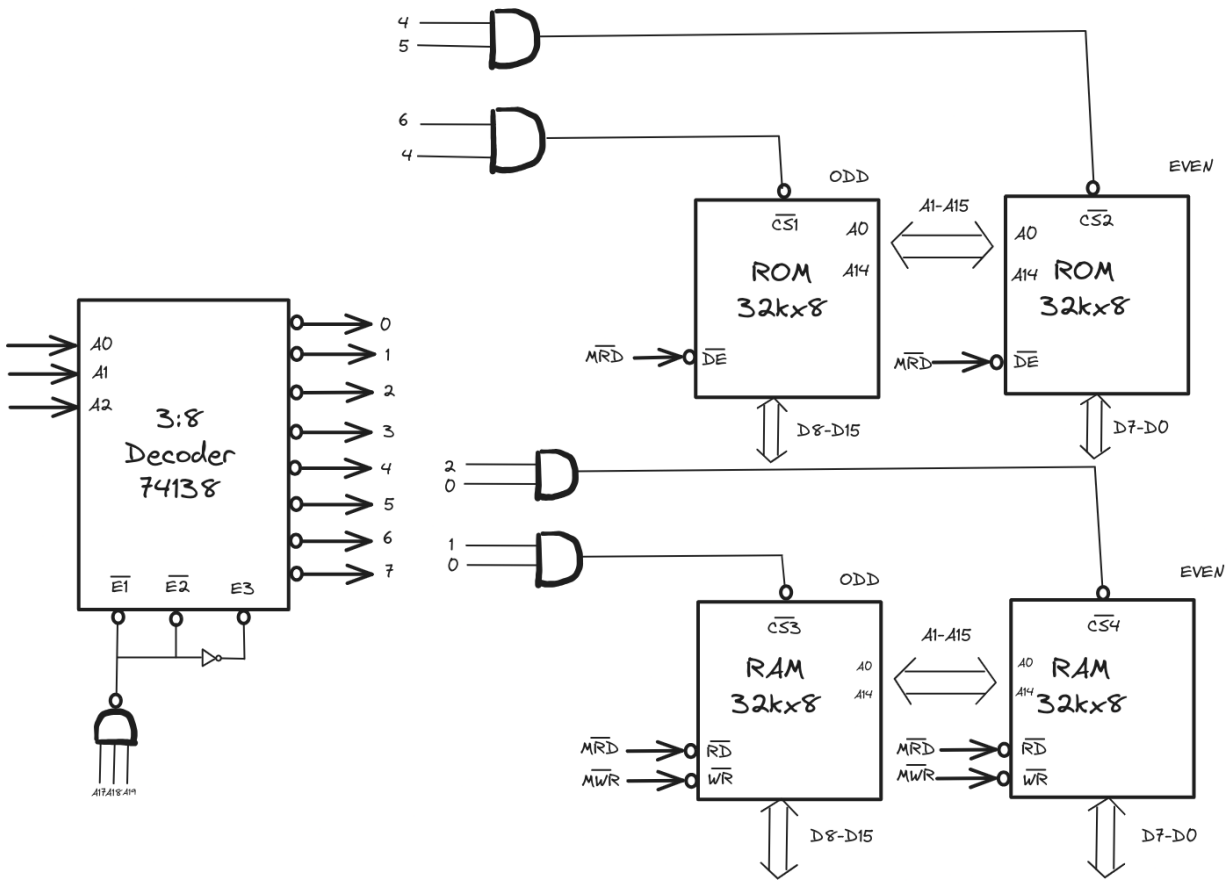


Fig. 4.1 *Stack-top Address Calculation*

b) Interface two 32K X 8 EPROMS and two 32K X 8 RAM chips with 8086, microprocessor and draw the suitable circuit showing their interfacing. **(10) (Dec 2021)**

2x32 K = 64 => 2^{10} . 2^6 = 16 pins needed

Address	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
EPROM (64Kx8) $FFFFF_H$ - $F0000_H$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RAM (64Kx8) $EFFFF_H$ - $E0000_H$	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



14. a) Draw and explain the internal architecture of 8259. **(8) (Dec 2021)**

See Qn. 16(b) for detailed answer

b) State the purpose of Interrupt Vector Table of 8086 and explain its structure. (6) (Dec 2021)

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). Since 4-bytes are required for storing starting addresses of ISPs, the table can hold 256 Interrupt procedures. The starting address of an ISP is often called the Interrupt Vector or Interrupt Pointer. Therefore, the table is referred to as Interrupt Vector Table.

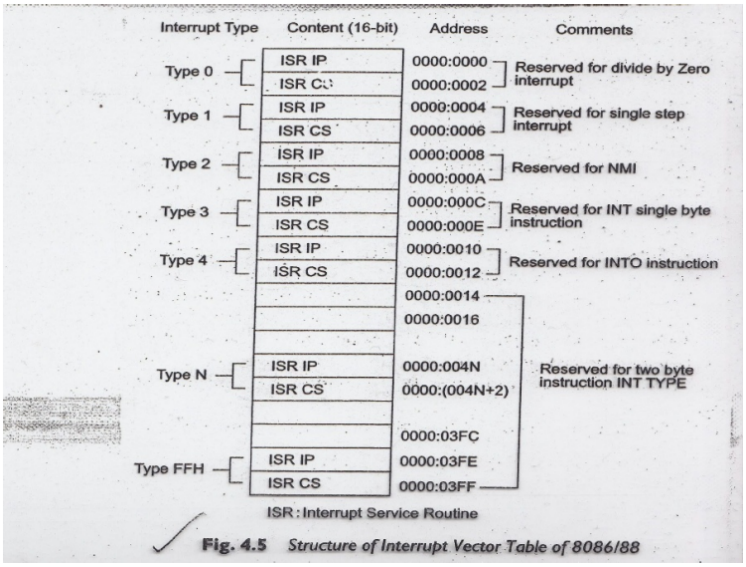


Fig. 4.5 *Structure of Interrupt Vector Table of 8086/88*

15. a) Explain the interrupt cycle of 8086. (8) (Dec 2022)

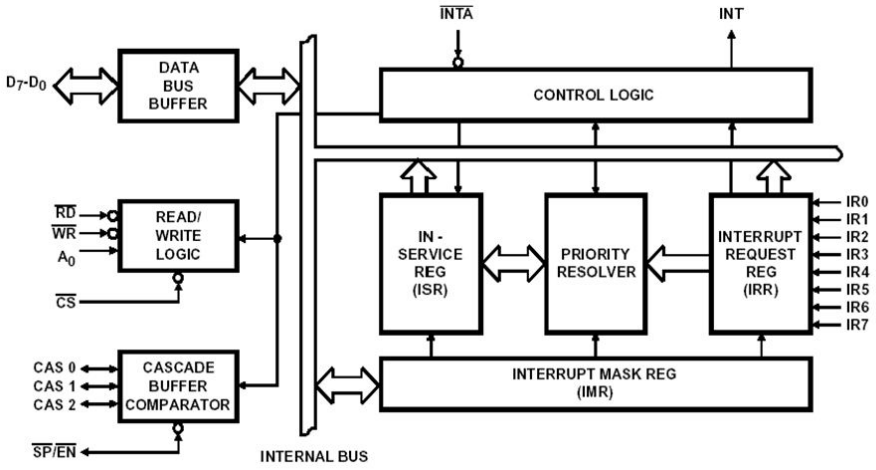
1. External interface sends an interrupt signal, to the Interrupt Request (INTR) pin, or an internal Interrupt occurs.
2. The CPU finishes the present instruction (for a hardware interrupt) and sends Interrupt acknowledge (INTA) to the hardware interface.
3. The interrupt type N is sent to the Central Processor Unit (CPU) via the Data bus from the hardware interface.
4. The contents of the flag registers are pushed onto the stack.
5. Both the interrupt (IF) and Trap (TF) flags are cleared. This disables the INTR pin and the trap or Single-step feature.
6. The contents of the code segment register (CS) are pushed onto the Stack.
7. The contents of the instruction pointer (IP) are pushed onto the Stack.
8. 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.

9. 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

b) Differentiate maskable and non-maskable interrupts in 8086. **(6) (Dec 2022)**

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

16. Draw the architectural block diagram of 8259A and explain the function of each Block.
(14) (Dec 2022)



- 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)
    - Interrupt Mask Register (IMR)
    - In-Service Register (ISR)

      (All are 8 bit registers)
    - Priority Resolver
    - Data Bus Buffer
    - Read/Write Logic
    - Cascade buffer / comparator
    - Control Logic
- 8259 will function in two modes:
    - Single mode ← Only one 8259
    - Cascaded mode
- All the internal functional units of 8259 are connected using 8 bit internal bus
- <u>Interrupt Request Register (IRR)</u>

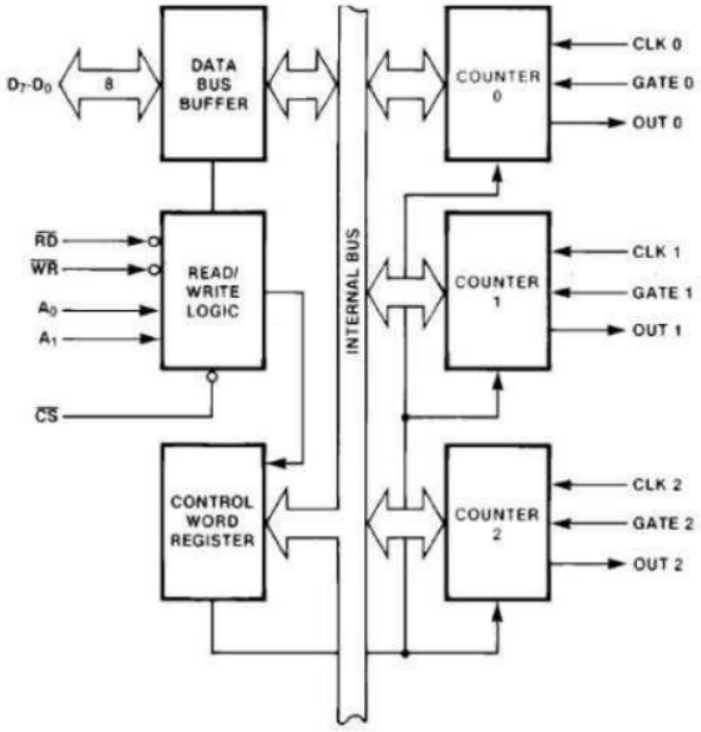
- In single mode, 8 different I/O devices (Maximum) can be connected with 8 interrupt request pins (IR_0-IR_7)
- In cascaded mode, there are one master 8259 and 8 slave 8259's are connected together
- In cascaded mode, the INT pin of each slave 8259's are connected with master 8259's interrupt request pin
- In single mode, if any of the I/O devices is asserting/generating an interrupt signal, the respective bit inside Interrupt Request Register (IRR) will be set.
- In default mode (without programming), IR_0 is having the highest priority, IR_1 is having the second-highest priority. Like that IR_7 is having the least priority. This priority can be changed using program (Writing command to 8 bit command register)

- 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.
- Priority Resolver
  - Priority resolver unit will decide which interrupt needs to be prioritized
  - The inputs to priority resolver are IMR, IRR, and ISR
- In-service Register
  - In-service register stores the status of the interrupt which is currently being executed
- Control Logic
  - Control logic sends interrupt signal to the 8086 microprocessor
  - INT pin of 8259 is connected INTR pin of 8086 microprocessor. INTA pin of 8086 microprocessor is connected with INTA pin of 8259
- Data Bus Buffer
  - Data bus buffer holds the vector no. of different interrupts before transferring to 8086 through D_0-D_7 (Data bus). It also holds the commands sent by 8086 microprocessor
- Read/Write Logic
  - Read/write logic block initiates different read and write operations (Command transfer - transferring the command, transferring from processor, transferring vector address)
- 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 (CAS_0, CAS_1, CAS_2)

Module - 4

17. a) Explain the 8254 programmable timer and its operation modes with a neat block diagram. **(9)(Dec 2021)**

(There is confusion regarding whether the block diagram is referring to the internal architecture or operational mode diagrams)



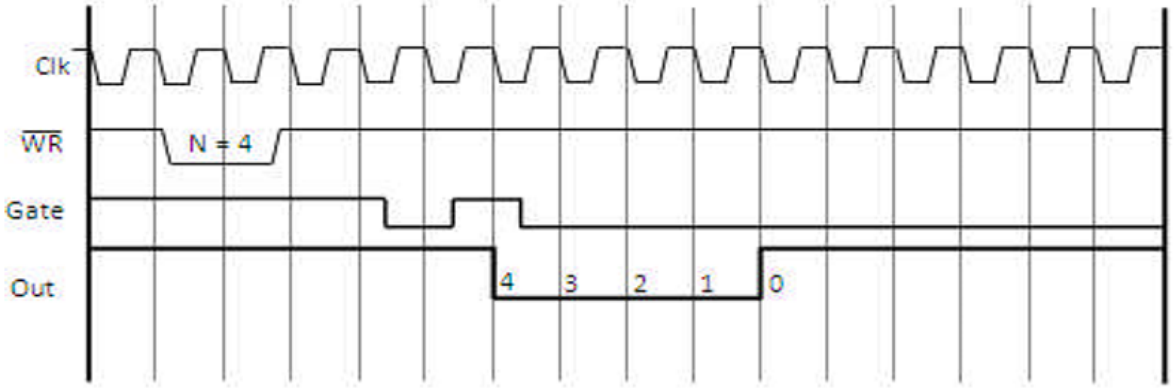
- The 8254 is a Programmable Interval Timer designed for microprocessors to perform timing and counting functions using three 16-bit registers.
- Each Counter has 3 pins: Clock, Gate and OUT.
- It has 3 independent 16-bit down counters.
- These 2 counters can be programmed for either binary or BCD Count.
- It has a powerful command - READ BACK - which allows the user to check the count value and the current mode.
- To operate a counter:
    1. A 16-bit count is loaded into its register.
    2. Counting starts after the next high to low transition of the clock pulse.
    3. Counter decrements for every clock pulse until it reaches Terminal Count (0).
    4. Once TC is reached, OUT signal is generated to interrupt the CPU.

8254 can be operated in 6 different modes:

1. **Mode 0 - Interrupt on Terminal Count**

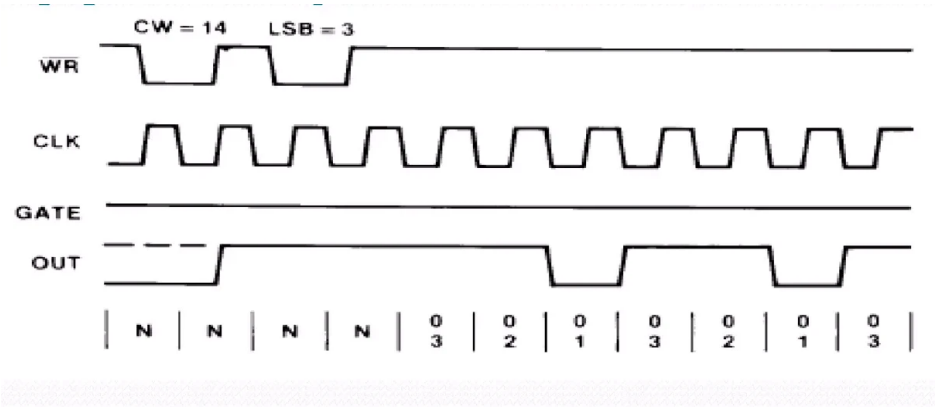
- Mode 0 is used for event counting.
- This mode is selected if (D3 D2 D1) of the control word register is (0 0 0)
- When this mode is selected, the OUT pin is initially low.
- The GATE pin is made high, so counting is enabled.
- The count value is loaded, and the counting starts after the next high to low transition of the clock pulse.
- During counting, the OUT pin remains low.
- On terminal count (0), the OUT pin goes high, and it remains high until a new count is written into the counter.
- During counting, if the gate goes low then counting is terminated and the current count is latched till the GATE goes high again.

2. **Mode 1 - Hardware Retriggerable one shot**



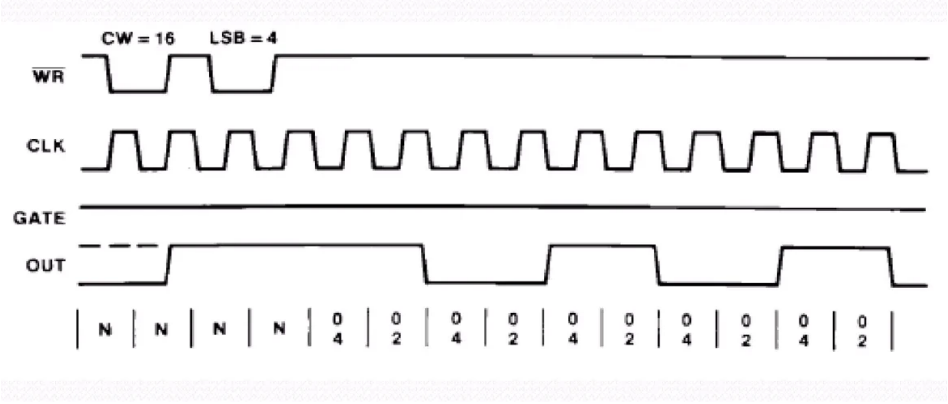
- It can be used as a mono stable multi-vibrator.
- This mode id selected if (D3 D2 D1) of CWR is (0 0 1).
- When this mode is selected OUT pin is initially high.
- The count value is loaded. Counting begins ONLY when a rising edge is applied to the GATE. This will act as a trigger. OUT pin goes low and remains low during counting.
- On Terminal Count (TC) the OUT pin goes high, and remains high.
- During counting if GATE is made low, it has no effect on the Counting.
- The Counter can be re-triggered by applying a rising edge on the GATE. This would Restart the counting, and hence re-trigger it.

3. **Mode 2 - Rate Generator**



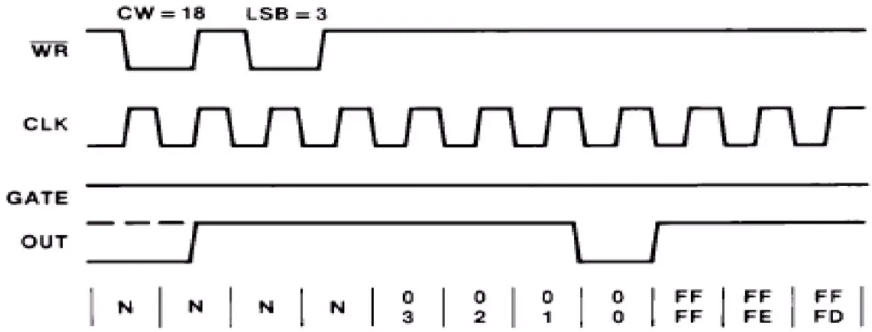
- Also know as divide by N counter.
- OUT pin is initially high.
- If N is loaded as count values then after N - 1 cycle, the output becomes low only for one clock cycle.
- The counter is then reloaded, the OUT pin is set high and counting starts once more.

4. **Mode 3 - Square Wave Generator**



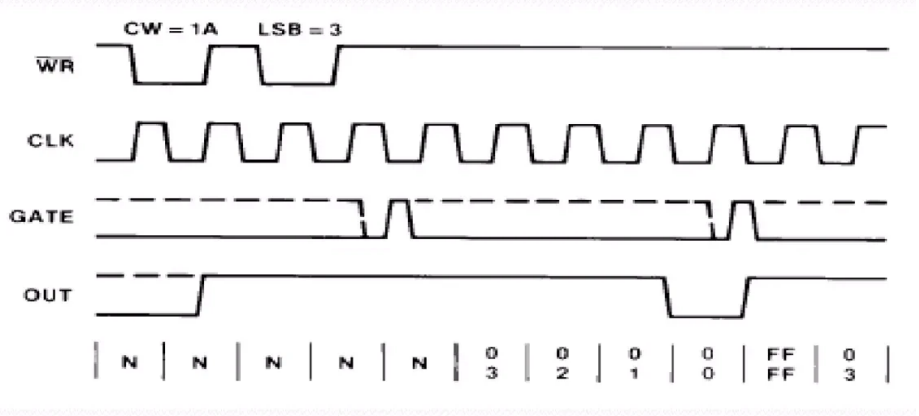
- Similar to Mode 2 except the OUT remains low for half of the timer period and high for the other half of the period.
- If the count loaded is odd, the first clock cycle decrements it by 1 resulting in an even count.
- This process is repeated continuously, generating a square wave.

5. **Mode 4 - Software Triggered Mode**



- Intially OUT remains high.
- On last count, the output goes low for one clock cycle and then goes HIGH. This low pulse can be used as a strobe.
- The count is then reloaded again during next clock pulse.
- The count is latched when the GATE signal goes low.

6**. Mode 5 - Hardware Triggered Mode**

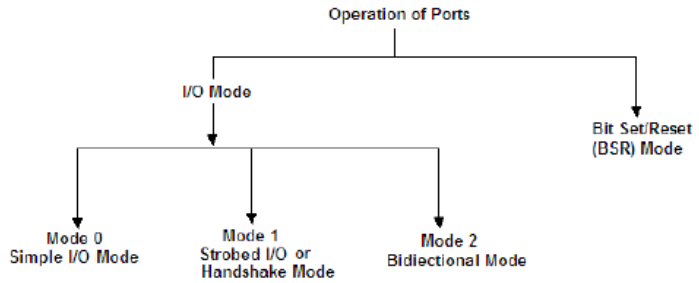


- Similar to mode 4 except that the counting is initiated by a signal at the gate input.
- This means it is hardware triggered.

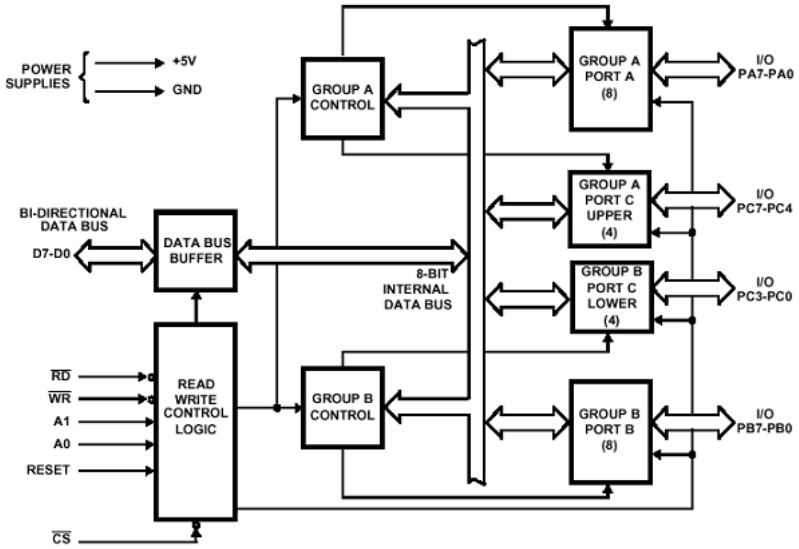
b) Explain different modes of operation of 8255 PPI. **(5)(Dec 2021)**
(See Qn. 19 for detailed answer)

18. a) With a neat diagram, describe the architecture of 8255 PPI. **(8)(Dec 2021)**

- 8255 PPI is designed by Intel corporation for interfacing I/O devices (Peripheral devices) with microprocessor (8085,8086,8088)
- Basically 8255 will function in two modes:
  - BSR Mode
  - I/O Mode



- BSR mode is also known as bit reset mode is using for setting or resetting port C bits
- I/O Mode is also known as Input/Ouput mode will function in 3 different modes:
    - Mode 0
    - Mode 1
    - Mode 2
- In mode 0, all ports (Port A, Port B, Port C) are using for input/output data transfer without handshaking. It is also known as basic I/O mode
- In mode 1, only port A, B are using for input/output data transfer. Port C pins are using for handshaking operation. The upper port C bits (PC_4 - PC_7) are using for handshaking operation of Port A. The lower port C bits (PC_0 - PC_3) are using for handshaking operation of port B. It is also known as strobed I/O mode
- In mode 2, only port A is using for I/O data transfer (Bi-directional data transfer) and port C pins are using for handshaking operation of port A. It is also known as strobed bidirectional I/O mode



- In 8255 IC,there are 40 pins. 24 pins are using for port A, port B, and port C (PA_0 - PA_7, PB_0 - PB_7, PC_0 - PC_7). There are 3 internal latches/buffers for each port. There is one control register that controls the different operations of 8255
- All internal latches and control registers are 8 bits
- The D_7 bit of control register will decide the different operations of 8255 (BSR Mode and I/O mode)
  If D_7 bit = 0, then 8255 will function in BSR mode
  If D_7 bit = 1, then 8255 will function in I/O mode
- D_3-D_6 bits of control word are using for mode selection and I/O data transfer direction of port A
- D_0-D_2 bits of control word are using for mode selection and I/O data transfer direction of port B
- D_0-D_3 pins of control word are using for direction of data transfer of port C (Mode 0)

b) Give the registers available in 8257 DMA Controller. Explain their functions. (6)(Dec 2021)

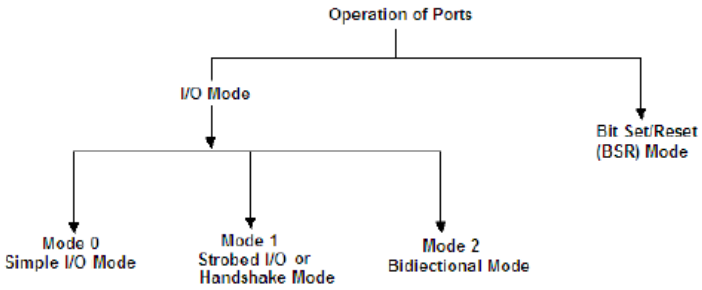
The 8257 performs DMA operation over four independent DMA channels with the following Registers.

1. **DMA Address Register**
   Each DMA channel has one DMA address register. The function of this register is to store the address of the starting memory location, which will be accessed by the DMA channel. The device that wants to

transfer data over a DMA channel, will access the block of the memory with the starting address stored in the DMA Address Register.

2. **Terminal Count Registers**
   Each of the four DMA channels of 8257 has one terminal count register (TC). This 16-bit register is used for ascertaining that the data transfer through a DMA channel cease or stops after the required number of DMA cycles.
   After each DMA cycle, the terminal count register content will be decremented by one and finally it becomes zero after the required number of DMA cycles are over. The bits 14 and 15 of this register indicate the type of the DMA operation (transfer).
3. **Mode Set Register**
   The mode set register is used for programming the 8257 as per the requirements of the system.
   The function of the mode set register is to enable the DMA channels individually and also to set the various modes of operation. The bits Do-D3 enable one of the four DMA channels of 8257.If the TC STOP bit is set, the selected channel is disabled after the terminal count condition is reached, and it further prevents any DMA cycle on the channel. If the TC STOP bit is programmed to be zero, the channel is not disabled, even after the count reaches zero and further request are allowed on the same channel. The autoload bit, if set, enables channel 2 for the repeat block chaining operations, without immediate software intervention between the two successive blocks. The extended write bit, if set to '1', extends the duration of MEMW and IOW signals by activating them earlier, which is useful in interfacing the peripherals with different access times.
4. **Status register**
   The lower order 4-bits of this register contain the terminal count status for the four individual channels. If any of these bits is set, it indicates that the specific channel has reached the terminal count condition. The update flag is not affected by the read operation. This flag can only be cleared by resetting 8257. The update flag is set every time the channel 2 registers are loaded with contents of the channel 3 registers. It is cleared by the completion of the first DMA cycle of the new block. This register can only read.

19. Explain the different modes of operation of 8255 in detail. **(14)(Dec 2022)**



Modes of Operation

8255A has three different operating modes -

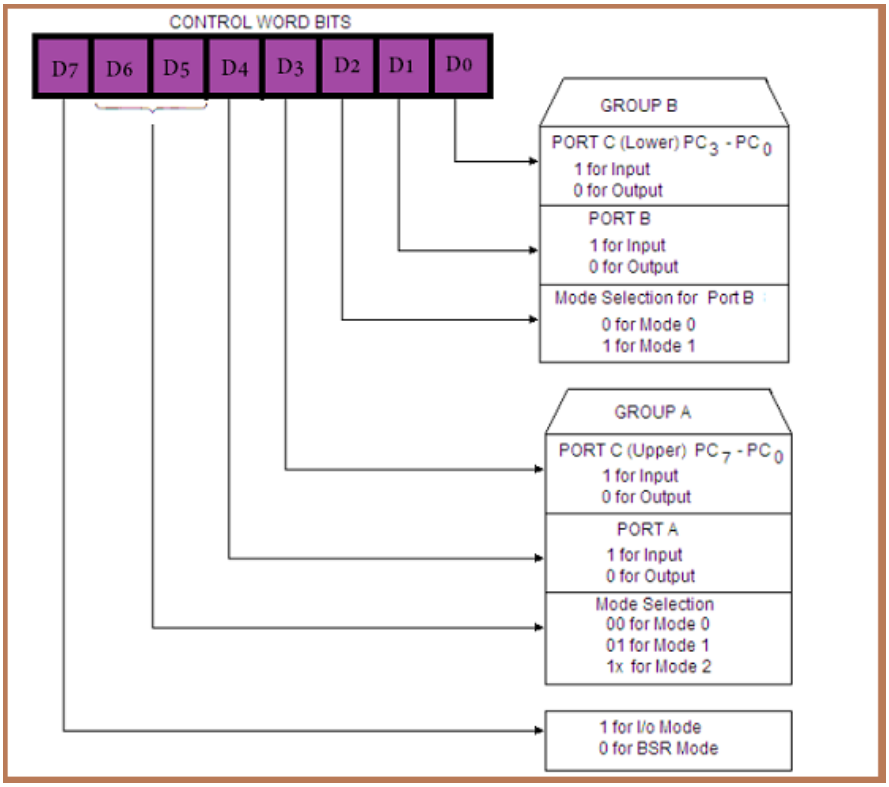
1. **INPUT/OUTPUT MODE**

There are three basic modes of operation than can be selected by the system software:

- Mode 0 -Basic Input/Output
- Mode 1 -Strobed Input/Output
- Mode 2 - Bi-directional Bus

Mode 0 :

- In this mode, Port A and B is used as two 8-bit ports and Port C as two 4- bit ports.
- Each port can be programmed in either input mode or output mode.
- Ports do not have interrupt capability.
- Port C is divided into two nibbles - upper and lower. Both upper and lower can now be programmed to use as input or output.
- Mode 0 is also known as input/output mode (1/0 mode).



Mode 1:

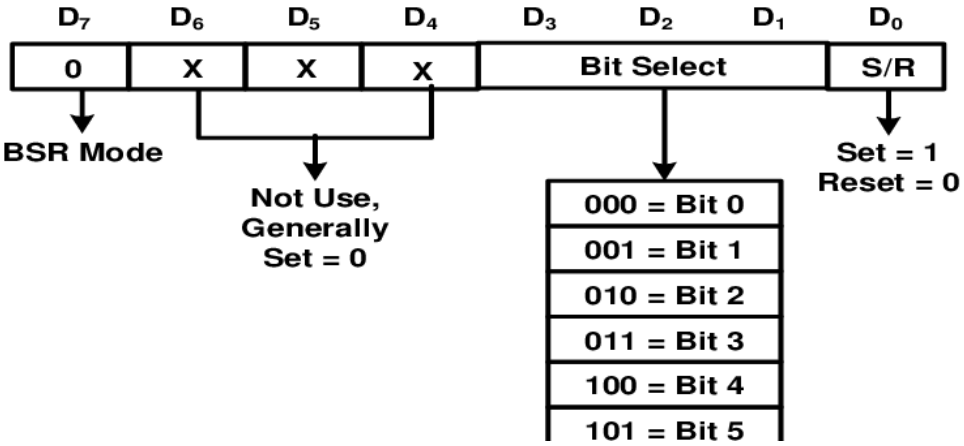
- In mode 1 each group can be used as 8-bit input or output data bus and the remaining 4-bits (of Port C) are used as handshaking and interrupt control signals.
- Port A is used with port C upper three bits and port B is used with port c lower 3 bits. The remaining 2 bits of port C can be used as control signals.

Mode 2:

- In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1.
- Port A uses five signals from Port C as handshake signals for data transfer. The remaining three signals from Port C can be used either as simple I/0 or as handshake for port B.

2. **BIT SET/RESET (BSR) MODE:**

- In this mode only Port B can be used (as an output port).
- Each line of port C (PC0 - PC7) can be set/reset by suitably loading the command word register.
- The figure shows the control word format in BSR mode. This mode is selected by making D7='0'.
- D1, D2, D3 are used to select a particular port C bit whose value may be either set or reset.
- D0 is used for bit set/reset. When D0 = 1, the port C bit selected is SET and when D0 = 0, the port C bit is RESET.



20. Draw and explain the internal architecture of 8257. **(14)(Dec 2022)**

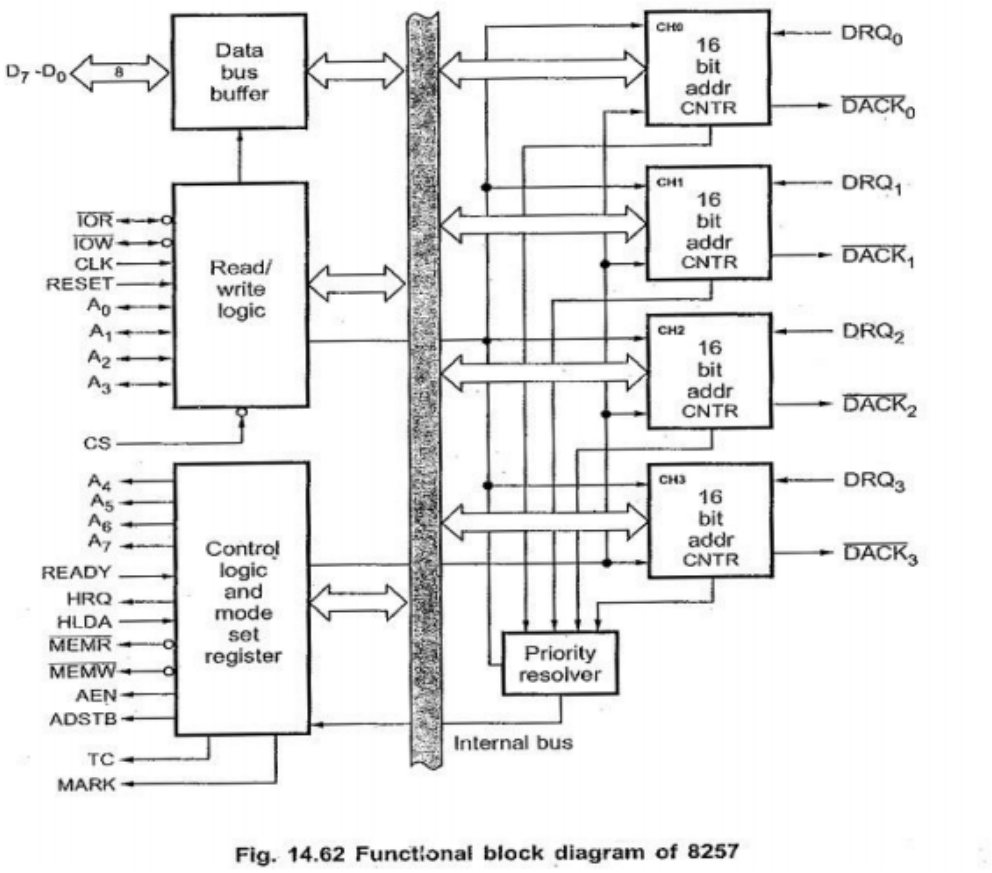


Fig. 14.62 Functional block diagram of 8257

DRQ0−DRQ3: These are the four individual channel DMA request inputs, which are used by the peripheral devices for using DMA services. When the fixed priority mode is selected, then DRQ0 has the highest priority and DRQ3 has the lowest priority among them.

DACKo − DACK3: These are the active-low DMA acknowledge lines, which updates the requesting peripheral about the status of their request by the CPU. These lines can also act as strobe lines for the requesting devices.

Do − D7: These are bidirectional, data lines which are used to interface the system bus with the internal data bus of DMA controller. In the Slave mode, it carries command words to 8257 and status word from 8257. In the master mode, these lines are used to send the higher byte of the generated address to the latch. This address is further latched using ADSTB signal.

IOR: It is an active-low bidirectional tri-state input line, which is used by the CPU to read internal registers of 8257 in the Slave mode. In the master mode, it is used to read data from the peripheral devices during a memory write cycle.

IOW: It is an active low bi-direction tri-state line, which is used to load the contents of the data bus to the 8-bit mode register or upper/lower byte of a 16-bit DMA address register or terminal count register. In the master mode, it is used to load the data to the peripheral devices during DMA memory read cycle.

CLK: It is a clock frequency signal which is required for the internal operation of 8257.

RESET: This signal is used to RESET the DMA controller by disabling all the DMA channels.

A_0 - A_3: These are the four least significant address lines. In the slave mode, they act as an input, which selects one of the registers to be read or written. In the master mode, they are the four least significant memory address output lines generated by 8257.
CS: It is an active-low chip select line. In the Slave mode, it enables the read/write operations to/from 8257. In the master mode, it disables the read/write operations to/from 8257.

A_4 - A_7: These are the higher nibble of the lower byte address generated by DMA in the master mode.

READY: It is an active-high asynchronous input signal, which makes DMA ready by inserting wait states.

HRQ: This signal is used to receive the hold request signal from the output device. In the slave mode, it is connected with a DRQ input line 8257. In Master mode, it is connected with HOLD input of the CPU.

HLDA: It is the hold acknowledgement signal which indicates the DMA controller that the bus has been granted to the requesting peripheral by the CPU when it is set to 1.

MEMR: It is the low memory read signal, which is used to read the data from the addressed memory locations during DMA read cycles.

MEMW: It is the active-low three state signal which is used to write the data to the addressed memory location during DMA write operation.

ADST: This signal is used to convert the higher byte of the memory address generated by the DMA controller into the latches.

AEN: This signal is used to disable the address bus/data bus.

TC: It stands for ‘Terminal Count’, which indicates the present DMA cycle to the present peripheral devices.

MARK: The mark will be activated after each 128 cycles or integral multiples of it from the beginning. It indicates the current DMA cycle is the 128th cycle since the previous MARK output to the selected peripheral device.

V_{cc}: It is the power signal which is required for the operation of the circuit.

8257 consists of five functional blocks:

a) Data bus buffer
b) Control logic
c) Read/write logic
d) Priority Resolver

e) DMA channels

Data Bus Buffer:
8-bit Tristate, bidirectional buffer interfaces the internal bus of 8257 with the external system bus under the control of various control signals.

Read/Write Logic:
In the slave mode, the read/write logic accepts the I/O Read or I/O Write signals, decodes the Ao-A3 lines and either writes the contents of the data bus to the addressed internal register or reads the selected register depending upon whether IOW or IOR signal is activated. In master mode, the read/write logic generates the IOR and IOW signals to control the data flow to or from the selected peripheral.

Control Logic:
The control logic controls the sequences of operations and generates the required control signals like AEN, ADSTB, MEMR, MEMW, TC and MARK along with the address lines A4 - A7, in master mode.

Priority Resolver:
The priority resolver resolves the priority of the four DMA channels depending upon whether normal priority or rotating priority is programmed.

Module - 5

21. a) Explain the Internal RAM organization of 8051 with a neat diagram. (8)(Dec 2021)

- 8051 supports 64 Kbytes of external data memory whose map starts at 0000H and ends at FFFFH. This external data memory can be accessed under the control register DPTR, which stores the addresses for external data memory accesses.
- 8051 generates RD and WR signals during external data memory accesses. The chip select line of the external data memory may be derived from the address lines, as in the case of other microprocessors.

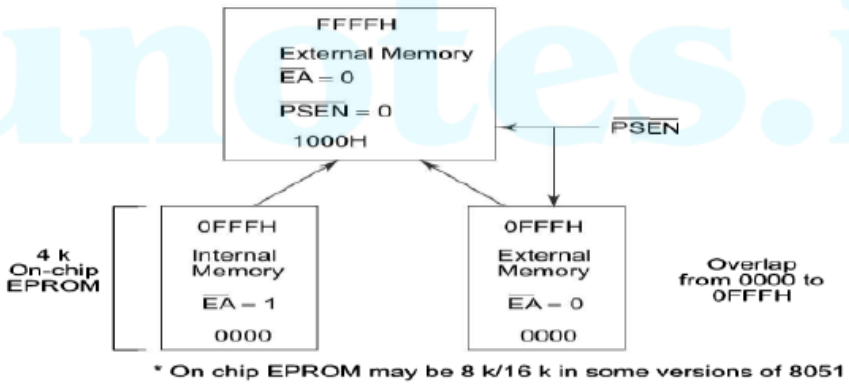
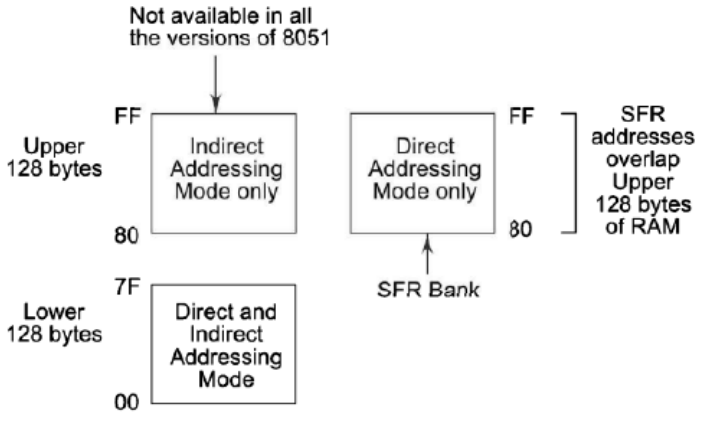


Fig. 17.5 *Program Memory Map of an 8051 System*

- Internal data memory of 8051 consists of two parts; the first is the RAM block of 128 bytes (256 bytes in case of some versions of 8051) and the second is the set of addresses from 80H to FFH, which includes the addresses allotted to the special function registers.
- The address map of the 8051 internal 128 bytes RAM starts from 00 and ends at 7FH. This RAM can be addressed by using direct or indirect mode of addressing.
- However, the special function register address map, i.e. from 80H to FFH is accessible only with direct addressing mode.

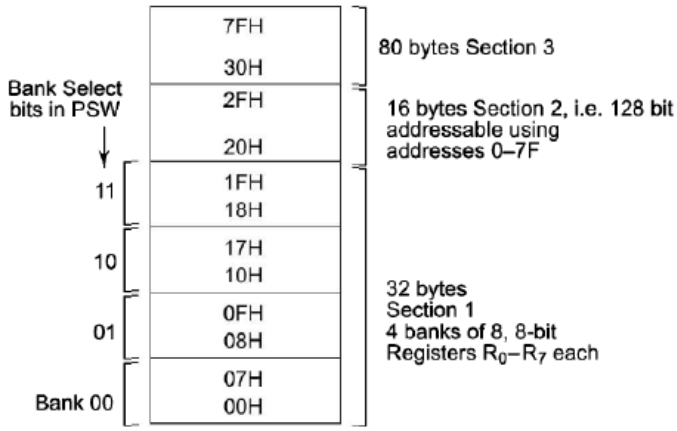


Internal Data Memory of 8051

- The lower 128 bytes of RAM whose address map is from 00 to 7FH is functionally organized in three sections.
    - First Section:
        - The address block from 00 to 1FH, i.e. the lowest 32 bytes which form the first section, is divided into four banks of 8-bit registers, denoted as bank 00,01,10 and 11.
        - Each of these banks contain eight 8-bit registers. The stack pointer gets initialized at address 07H, i.e. the last address of the bank 00, after reset operation.
        - After reset bank 0 is selected by default, but the actual stack data is stored from 08H onwards, i.e. bank 01, 10 and 11.
        - These bank addressing bits of the register banks are present in PSW, to select one of these banks at a time.
        - These banks are selected using RS1 and RS0 bits of the program status word.

RS1	RS0	Register Bank	Address
0	0	0	00H - 07H
0	1	1	08H - 0FH
1	0	2	10H - 17H
1	1	3	18H - 1FH

    - Second Section:
        - The second section extends from 20H to 2FH, i.e. 16 bytes, which is a bit-addressable block of memory, containing 16 × 8 = 128 bits.
        - Each of these bits can be addressed using the addresses 00 to 7FH.
        - Any of these bits can be accessed in two ways. In the first, its bit number is directly mentioned in the instruction, while in the second the bit is mentioned with its position in the respective register byte. For
        - example, the bits 0 to 7 can be referred directly by their numbers, i.e. 0 to 7 or using the notations 20.0 to 20.7 respectively.
    - Third Section:
        - The third block of internal memory occupies addresses from 30H to 7FH.
        - This block of memory is a byte addressable memory space.
        - In general, this third block of memory is used as stack memory.
        - All the internal data memory locations are accessed using 8-bit addresses under appropriate modes of addressing.



. 17.7 *Functional Description of Internal Lower 128 Bytes of RAM*

b) List any four addressing modes supported by 8051 microcontrollers, with one example each. **(6)(Dec 2021)**

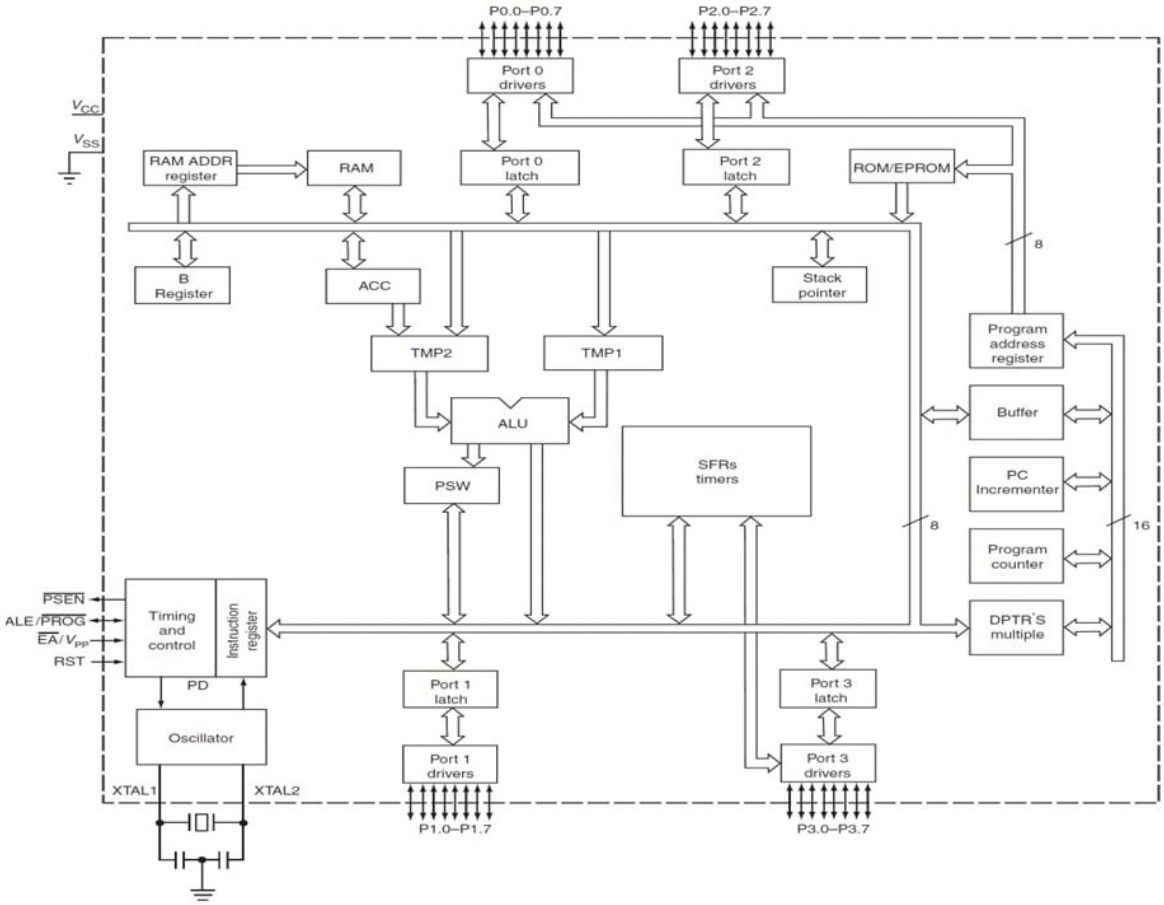
(Any 4 is needed, explaining all addressing modes below)

8051 supports 6 addressing modes:

1. **Immediate Addressing:** Data is immediately available in the instruction.
   For example -
   ADD A, #77 — Adds 77 (decimal) to A and stores in A
   MOV DPTR, #1000H — Moves 1000 (hexadecimal) to the data pointer

2. **Register Addressing:** This way of addressing accesses the bytes in the current register bank. Data is available in the register specified in the instruction. The register bank is decided by 2 bits of Processor Status Word (PSW).
   For example -
   ADD A, R0 — Adds content of R0 to A and stores in A
3. **Direct Addressing:** The address of the data is available in the instruction.
   For example -
   MOV A, 088H — Moves the content of SFR TCON to A (088H is the address of special function register TCON in 8051)
4. **Register Indirect Addressing:** The address of data is available in the R0 or R1 registers as specified in the instruction.
   For example -
   MOV A, @R0 — Moves the content of the address pointed by R0 to A
5. **Register-specific Addressing:** The operand is implicitly specified using one of the registers ie, such instructions operate only on a specific register.
   For example -
   RLA — This instruction rotates accumulator left
6. **Indexed Addressing:** Only program memory can be accessed using this mode. It is mainly used for look-up table manipulations.
   For example -
   MOVC A, @A+DPTR — DPTR has the base address of LUT and A has the relative (offset) address. The LUT content at the effective address obtained by adding the base address and relative address is moved to ACC (accumulator)

22. a) Explain the internal architecture of 8051 with a neat diagram. **(9)(Dec 2021)**



Accumulator: The Accumulator, as its name suggests, is used as a general register to accumulate the results of a large number of instructions. It can hold an 8-bit (1-byte) value.

'B' Register: The "B" register is very similar to the Accumulator in the sense that it may hold an 8-bit (1-byte) value. The "B" register is only used by two 8051 instructions: MUL AB and DIV AB. Aside from the MUL and DIV instructions, the "B" register is often used as yet another temporary storage register.

Program Status Word: The PSW register contains program status information. It is an 8-bit flag register, out of 8-bits 6 bits are used and 2 bits are reserved. Out of 6 bits 4 bits are conditional bits and 2 bits are used for selecting the register bank.

Stack Pointer: The Stack Pointer register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

Data Pointer: The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its intended function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

Program Counter: The Program Counter (PC) is a 2-byte address (16 - bit) that tells the 8051 where the next instruction to execute is found in memory. When the 8051 is initialized PC always starts at 0000h and is incremented each time an instruction is executed.

Ports 0 to 3: P0, P1, P2, and P3 are the SFR latches of Ports 0, 1, 2, and 3, respectively. Writing a one to a bit of a port SFR (P0, P1, P2, or P3) causes the corresponding port output pin to switch high. Writing a zero causes the port output pin to switch low.

Serial Data Buffer: The Serial Buffer is actually two separate registers, a transmit buffer and a receive buffer. When data is moved to SBUF, it goes to the transmit buffer and is held for serial transmission. For serial reception, the data is received in receive buffer.

Timing and Control Unit: This unit derives all timing and control signals required for the internal operation of the chip. It also derives the control signals required for controlling the external system bus

PSEN (Program Store Enable): It is a control signal that enables external program (code) memory. It usually connects to an EPROM's Output Enable (OE) pin to permit the reading of program bytes.
ALE (Address Latch Enable): The 8051 uses ALE for demultiplexing the address and data bus. ALE enables address lines.
EA (External Access): The EA input signal is generally tied high (+5 V) or low (ground). If high, the 8051 executes programs from internal ROM. If low, programs execute from external memory only (and PSEN pulses low accordingly).
RST (Reset): The RST input is the master reset for the 8051. When this signal is brought high for at least two machine cycles, the 8051 internal registers are loaded with appropriate values for an orderly system start-up.
On-chip Oscillator Inputs: The 8051 features an on-chip oscillator. The nominal crystal frequency is 12 MHz.

b) State the name and purpose of any 6 special function registers (SFRs) of 8051 microcontroller.
(5)(Dec 2021)

1. P0 (Port 0):
   Purpose: P0 is an 8-bit bidirectional I/O port. It can be used to interface with external devices or as general-purpose I/O pins.
2. P1 (Port 1):
   Purpose: Similar to P0, P1 is an 8-bit bidirectional I/O port. It also has additional functions such as external interrupt and timer input.
3. P2 (Port 2):
   Purpose: P2 is an 8-bit bidirectional I/O port with additional control signals. It can be used for interfacing and controlling external devices.
4. P3 (Port 3):
   Purpose: P3 is an 8-bit bidirectional I/O port, and like P1 and P2, it can be used to interface with external devices. Additionally, P3 has specific control signals for external interrupts.
5. TCON (Timer Control):
   Purpose: TCON is the Timer Control register. It is used to control and monitor the status of Timer 0 and Timer 1, including their modes and interrupt flags.
6. TMOD (Timer Mode):
   Purpose: TMOD is the Timer Mode register. It is used to set the operating modes of Timer 0 and Timer 1. It allows configuration of timers for different counting and timing modes.

23. a) Explain the addressing modes of 8051 with examples. (10)(Dec 2022)
See Qn. 21(b), all addressing modes are explained there
b) Write an assembly language program for 8051 to perform addition of two 2×2 matrices.
(4)(Dec 2022)
(Will ask sir tomorrow)

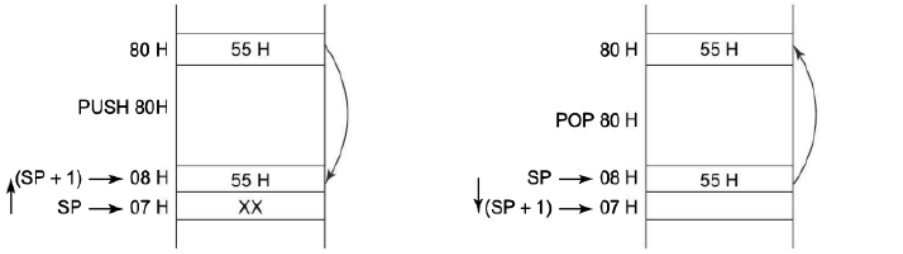
24. a) Explain the interrupt and stack structure of 8051. (10)(Dec 2022)

- There are five interrupt sources for the 8051, which means that they can recognize 5 different events that can interrupt regular program execution. Each interrupt can be enabled or disabled by setting bits of the IE (Interrupt enable) register. Likewise, the whole interrupt system can be disabled by clearing the EA (global interrupt enable) bit of the same register.
- **The five interrupt sources are:**
  - External interrupt 0 (IE0)
  - Timer 0 (TF0)
  - External interrupt 1 (IE1)
  - Timer 1 (TF1)
  - Serial port (R1 = T1)
- **Interrupt registers in 8051:** 8051 µC has 2 8-bit interrupt registers.

    - Interrupt enable register (IE): it is an 8-bit register. It is bit/byte addressable. It is used to enable and disable the function of interrupt.
    - Interrupt priority register (IP): It is an 8-bit register. It is bit/byte addressable. It is used to select the low or high-level priority of individual interrupts.
- **Interrupt Priorities**
- There is a priority list instructing the controller how to handle multiple interrupts.
- The priority list offers 3 levels of interrupt priority:
    - Reset - When a reset request arrives, everything is stopped and the microcontroller restarts.
    - Interrupt priority 1 - can be disabled by Reset only.
    - Interrupt priority 0 - can be disabled by both Reset and interrupt priority 1.
- The IP Register (Interrupt Priority Register) specifies which one of the existing interrupt sources has higher and which one has lower priority. Interrupt priority is usually specified at the beginning of the program. According to that, there are several possibilities:
    - If an interrupt of higher priority arrives while an interrupt is in progress, it will be immediately stopped and the higher priority interrupt will be executed first.
    - If two interrupt requests, at different priority levels, arrive at the same time then the higher priority interrupt is serviced first.
    - If both interrupt requests, at the same priority level, occur one after another, the one which came later has to wait until the routine being in progress ends.
    - If two interrupt requests of equal priority arrive at the same time then the interrupt to be serviced is selected according to the following priority list:
        - External interrupt INT0
        - Timer 0 interrupt
        - External Interrupt INT1
        - Timer 1 interrupt
        - Serial Communication Interrupt

8051 STACK

8051 stack operations are 8-bit wide i.e. in an operation using PUSH or POP instruction one byte of data is stored on to stack or retrieved from the stack. In case of internal 16 bit address push or pop to/from the stack, the operation is implemented byte by byte i.e. lower byte first followed by higher byte. The SP register is an 8-bit register and is initialized to internal RAM address 07H after reset. Obviously, the capabilities of stack in 8051 are limited compared to microprocessors. Fig. shows operation of PUSH instruction. The SP registers points to stack top. The stack top is always assumed to be preoccupied. So the SP is incremented first. Then 8 bit content of the 8-bit address provided as operand is pushed on to the stack memory address available in SP.



(a) Storing into Stack Memory *(b) Retrieving from Stack Memory*

Thus the PUSH instruction has following two steps.

1. Increment stack by 1.
2. Store 8-bit content of the 8-bit address specified in the instruction to the address pointed to by SP.

Complementarily, POP operation has the following two steps as shown in Fig.

1. Store the content of top of stack pointed to by SP register to the 8 bit memory specified in the instruction.
2. Decrement SP by 1.

b) Write an assembly language program for 8051 to find the transpose of a 2×2 matrix.

(4)(Dec 2022)

(Will ask sir tomorrow)