Chapter-I

Short type questions [2 marks each]

Q-1) **Define Microprocessor and its application. [S-18(Q1-a)]**
Ans:-

Microprocessor:- It is a semiconductor device which takes digital data, processes according to the instruction and gives the result in digital form within small interval of time.

It is used in computer as CPU, automatic testing product, speed control of motors, traffic light control, light control of furnaces, vehicle and military system etc.

Medium type questions [5 marks each]

Q-1) **Distinguish between Microprocessor and Microcomputer and give it applications. [S-17(Q1-b)], [W-17(Q1-a)], [S-19(Q1-b)]**
Ans:-**Microprocessor:-**

- It is CPU on a single chip. It includes the ALU, small memory, register arrays on a single chip. It is of 8-bit, 16-bit or 32-bit type processor.
- It is used in instructions, automatic testing product, speed control of motors, traffic light control, light control of furnaces, etc.
- Its clock frequency is about 1GHz.
- Its cost is high.

Microcomputer:-

- It is designed using a microprocessor as its CPU, along with input device, output device and memory. It can be designed using 8-bit, 16-bit or 32-bit microprocessor.
- It is a computer that can perform specific application as arithmetic and logical operations.
- It is used in business and education including word processing, spreadsheets and database management, etc.
- Its clock frequency is up to 50MHz.
- Its cost is low.

Long type questions [7 marks each]

Q-1) **Explain Evolution of microprocessor. [S-18(Q1-c)]**
Ans:-

- The first microprocessor was evolved in the year 1971 by Intel Corporation as INTEL-4004. Its word length was 4-bit, memory addressing capacity is 1KB. In the same year it developed to its enhanced version INTEL-4040. After that other microprocessor such as Toshiba's T3471, Rockwell International's PPS-4 was made.

- In 1972 Intel introduced the first 8-bit microprocessor INTEL-8008 but due to slow processing in 1973 it introduced a more powerful 8-bit microprocessor 8080. But this processor use two power supply, so it is not useful.

- In 1975 by using one supply Intel developed a improved version of Intel 8080i.e.Intel 8085. Its memory addressing capacity is 64 KB and has 40 pins and has clock frequency is 3 to 6 MHz Other companies are ZILOG's Z80and Z800, national semiconductor's NSC800, Motorola's MC6800 etc.

- In the year 1978 Intel introduced a 16-bit microprocessor i.e. Intel 8086.Its memory addressing capacity is 1 MB and has 40 pins and clock frequency 5 to 10 MHz Other 16 bit microprocessor are Intel 80186, Intel 8088, ZILOG's Z8000, Motorola's 68000, 68010.

- In the year 1980 Intel introduced a 32-bit microprocessor but it was not become popular so in year 1985 Intel introduced a powerful 32-bit microprocessor i.e. 80386.Its memory addressing capacity is 4GB real and 64 TB virtual and has 132 pins and its clock frequency is 20 to 33 MHz Other 32 bit microprocessor are Intel 80486, Pentium pro, Pentium ii etc…..Other 32 bit microprocessor companies are Motorola's 68020, 68030, Cyrix's 586,686, national semiconductor's 32032,32332, ZILOG's Z80000 etc.

- A number of 64 bit microprocessor has also been developed, Ex-Sun's spark and ultra spark, AMD's claw hammer etc.

- In the year 2001 Intel develop a 64 bit Epic processor i.e. Intel Itanium which has 64 address lines and its clock frequency is about 733 MHz to 1.5 GHz.

Chapter-II

Short type questions [2 marks each]

Q-1) **Why data bus in microprocessor is bidirectional? [S-16(Q1-a)]**
Ans:-The data bus of the microprocessor is bidirectional because microprocessor can read/write data from/to memory and I/O devices.

Q-2) **What do you mean by program counter? [S-16 (Q2-a)], [S-19(Q1-a)]**
Ans:-A program counter (PC)is a register in µP that contains the address of the instruction being executed at the current time.

Q-3) **Write the various hardware interrupt of 8085 µP in ascending order. [S-17(Q6-a)], [S-15 (Q4-a)], [W-17 (Q7-a)]**
Ans:-Interrupts caused by I/O devices are called hardware interrupts. The hardware interrupts in ascending order are – (1) TRAP (2) RST 7.5 (3) RST 6.5 (4) RST 5.5 and (5) INTR.

Q-4) **What is the purpose of CLK signal of 8085 and name θ types of clock circuits? [S-17(Q2-a)]**
Ans:-The CLK is a output signal for user which is used as the system clock for peripheral and devices interfaced with the microprocessor. Different types of clock circuit are – (1) Single-phase clock (2) Two-phase clock (3)4-phase clock (4) Clock multiplier. The clock cycle is 320ns, but for 8085AH-2 version it is 200ns.

Q-5) **What is the function of ALE and where it is play an important role? [S-17(Q4-a)]**
Ans:-ALE is address latch enable signal. It is used to demultiplex the multiplexed lower order address/data bus. It is important during the read or write operation of the microprocessor.

Q-6) **Which are sixteen bit registers of 8085? [S-17(Q3-a)]**
Ans:-The stack pointer (SP) and program counter (PC) are sixteen bit registers of 8085 µP.

Q-7) **Define Architecture. [S-18(Q2-a)]**
Ans:-The data bus of the microprocessor is bidirectional because microprocessor can read/write data from/to memory and I/O devices.

Q-8) **What is the function of S_0 & S_1 pins of 8085 µP? [W-18(Q3-a)]**
Ans:- So and S1 pins of 8085 µP provide status signal.

S0	S1	Status
0	0	The circuit is in the Halt(HLT) state
1	0	The circuit is in the write(WR) state
0	1	The circuit is in the read (RD) state
1	1	OPCODE is fetched

Q-9) **Write two examples of GPR & SPR of 8085 µP with their sizes? [S-19(Q1-j)]**
Ans:- The two examples of GPR of 8085 µP are Register B, Register C. The size of GPR is 8-bit. The two examples of SPR of 8085 µP are SP and PC each size of 16-bit.

Medium type questions [5 marks each]

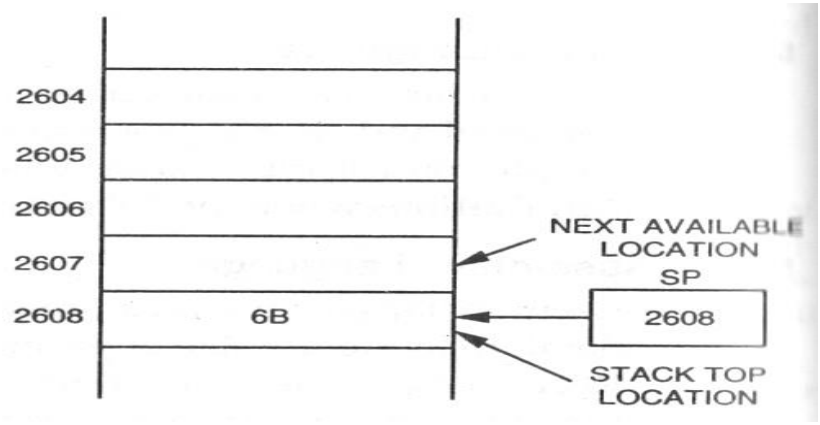
Q-1) **Describe three state registers and concept of multiplexing. [S-17(Q2-b)]**

Ans:- Almost all devices used in µP based system, use tri-state logic which are- HIGH(1), LOW(0) and OPEN(high impedance)ckt. The High and Low are normal logic levels used for data/address or control signals. The 3rd state is used to isolate the device from the system. The 3-state devices normally remain a High impedance state.

Microprocessor consists of 8-bit wide address and data bus. Both the bus can transfer 8-bits of data at a time. But address of µP is 16-bit long and data is 8-bit long. So, 8MSB of address is transferred through address bus and the 8LSB is through the data bus along with data. So these buses are multiplexed or joined.

Q-2) **With necessary diagram explain the operation of stack pointer and program counter. [S-15 (Q6-b)], [W-17(Q2-a)], [W-18 (Q7-b)]**

Ans:-



Stack Pointer:-The special register which points to a memory location in a stack, is called stack pointer. The beginning of the stack is defined by loading a 16-bit address in the SP. The stack pointer's contents are automatically adjusted to point to the stack. The stack is defined by initializing the SP. The SP is initialized by LXI SP,XXXXH instruction where XXXXH is the specified memory location.

Program Counter:- The special register that point to memory address from which the next byte is to be fetched is called program counter. When a byte is fetched, the PC is incremented by one to point to the next memory location.

Q-3) **State and explain stack, stack pointer and stack top. [S-18 (Q2-b)],[S-16 (Q5-b)], [S-15 (Q5-c)]**

Ans:-Stack is an area of memory identified by the programmer for temporary storage of information. Stack works on the principle of LIFO structure. The information is written on the stack by PUSH operation and read from the stack by POP operation. The data to be stored or retrieved to/from the stack will always be in double bytes. Stack is used to store data in register pairs BC,DE,HL or PSW.

The stack is implemented with the help of special memory pointer register called as the stack pointer. The stack pointer's contents are automatically adjusted to point to the stack. The stack is defined by initializing the SP. The SP is initialized by LXI SP,XXXXH instruction where XXXXH is the specified memory location.

The memory location that is currently pointed by the stack pointer is called stack top.

Q-4) **Draw the different bits of the flag register of Intel 8085 µP and explain the function of each flag with example. [S-15 (Q3-b)], [S-19(Q1-c)]**

Ans:-The 8085 µP contains 5 flip flops to serve as status flags. These are – Carry Flag (CS), Parity Flag (P), Auxiliary Carry Flag (AC), Zero Flag (Z) and Sign Flag (S).

Each flip-flop holds 1-bit by certain condition arises during arithmetic and logic operations.

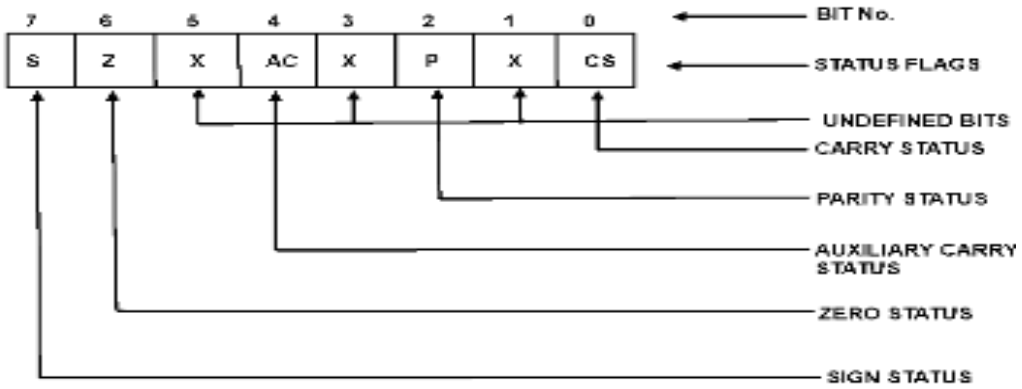


Figure 2: Status Flags of Intel 8085

1) **The carry flag(CS):-** This flag is set whenever there has been a carry out of, or a borrow into, the higher order bit of the result. The flag is used by the instructions that add and subtract multibyte numbers. It sets to-1 if carry out from MSB bit on addition or borrow into MSB bit on subtraction else reset to-0 if no carry out or borrow into MSB bit.

2) **The parity flag(P):-**This flag is set whenever the result of an arithmetic or logical operation has even parity, an even number of 1 bits. If parity is odd, this flag is cleared.

3) **The auxiliary carry flag(AC):-**This flag is set whenever there has been a carry out of the lower nibble into the higher nibble or a borrow from higher nibble into the lower nibble of an 8 bit quantity, else AC flag is reset. This flag is used by decimal arithmetic instructions.

4) **The zero flag(Z):-**This flag is set, when the result of an arithmetic or logical operation is zero, else it is reset.

5) **The sign flag(S):-**This flag is set to 1, if the result of an arithmetic or logical operation is negative. If the result is positive, it is set to 0. This flag has its significance only when signed arithmetic or logical operation is performed.

Q-5) **Define different register of 8085 and distinguish between GPR and SPR.**
[S-15 (Q7-b)], [S-19(Q1-b)]

Ans:-The 8085 µP has six different registers. They are one 8-bit Accumulator (ACC or Reg-A), Six 8-bit general purpose registers (Reg-B, C, D, E, H and L), One 16-bit stack pointer (SP), One 16-bit program counter (PC), Instruction register and Temporary register.

ACC:-It is associated with ALU which hold one of the operands of the operation and the final result of the operation.

GPR:-These general purpose registers are used to hold data like any other registers. The general purpose registers in 8085 processors are B, C, D, E, H and L. Each register can hold 8-bit data. Apart from the above function these registers can also be used to work in pairs to hold 16-bit data.They can work in pairs such as B-C, D-E and H-L to store 16-bit data. The H-L pair works as a memory pointer.A memory pointer holds the address of a particular memory location. They can store 16-bit address as they work in pair.

PC:-It stores the address of the next instruction to be executed. In other words the program counter keeps track of the memory address of the instructions that are being executed by the microprocessor and the memory address of the next instruction that is going to be executed.Microprocessor increments the program whenever an instruction is being executed, so that the program counter points to the memory address of the next instruction that is going to be executed.

SP:-Itis also a 16-bit register which is used as a memory pointer. Stack pointer maintains the address of the last byte that is entered into stack. Each time when the data is loaded into stack, Stack pointer gets decremented. Conversely it is incremented when data is retrieved from stack.

Instruction Register:-It holds the opcode of the instruction which is being decoded and executed.

Temporary Register:-This register acts as a temporary memory during the arithmetic and logical operations. Unlike other registers, this register can only be accessed by the microprocessor and it is completely inaccessible to programmers.

DISTINGUISH BETWEEN GPR & SPR:-

The GPR is general purpose register which can hold either data or instructions. These are dedicated memory storage areas inside the CPU and are used for carrying out immediate instructions, passing data or receiving immediate results from a function. These are B-C, D-E and H-L register pairs. These are user accessible.

The SPR is special purpose register that hold program state. These are stack pointer and program counter. These can't be used in pair form. These are not user accessible.

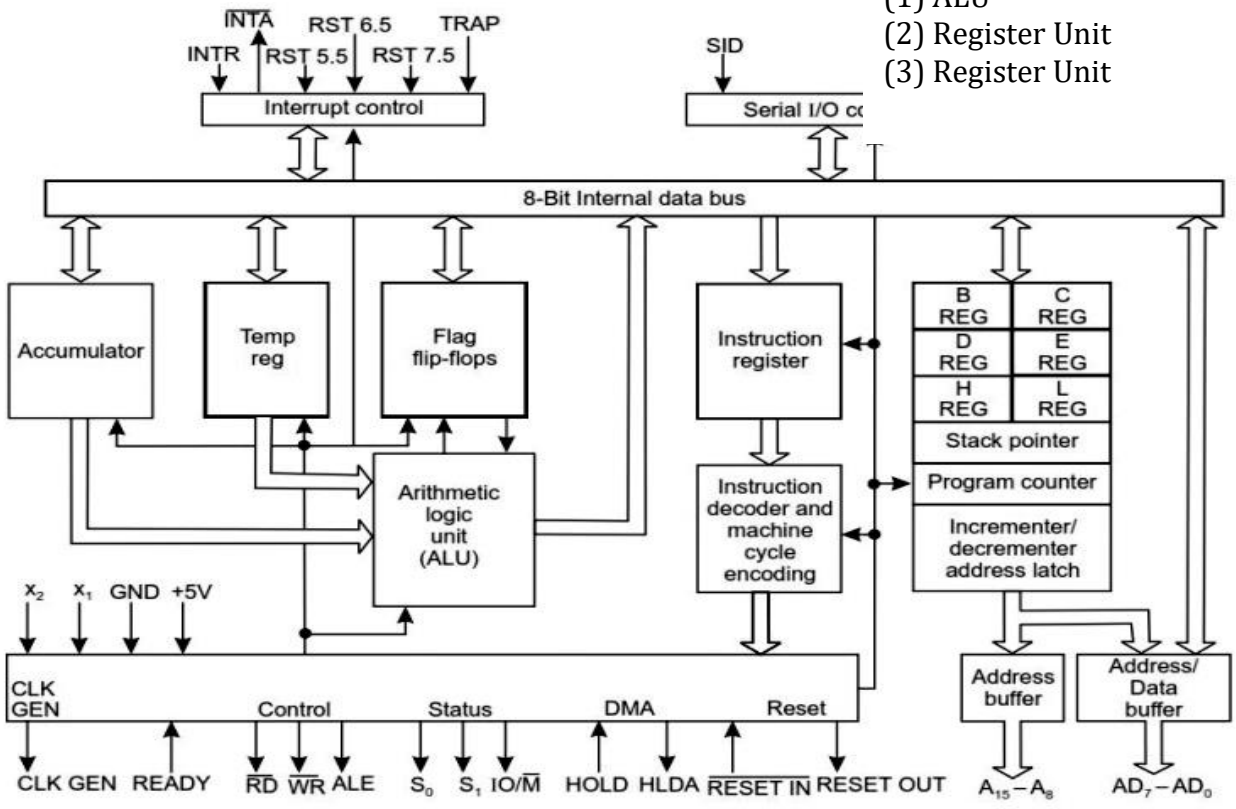
GPR	SPR
I. GPR stands for General Purpose Register.	I. SPR stands for Special Purpose Register.
II. These are 8 bit resisters.	II. These are 16 bit registers.
III. These are used to store the intermediate result.	III. These are used to store memory address.
IV. There are six general purpose register i.e. B, C, D, E, H, L.	IV. There are two special purpose register in 8085 i.e. stack pointer and program counter.
V. It can be used in pair form to store 16 bit data or address i.e. (B-C)(D-E)(H-L).	V. It cannot be used in pair form.
VI. These registers are user accessible.	VI. It is not user accessible.

Long type questions [7 marks each]

Q-1) **Draw the architecture of Intel 8085 µP through a block diagram and explain the function of each block. [S-17(Q1-c)], [S-15 (Q1-c)]**

Ans:-Intel 8085 µP is the most popular 8-bit microprocessor. Its architecture consists of three sections ----

(1) ALU
(2) Register Unit
(3) Register Unit



Arithmetic and logic unit: This is the unit where all the arithmetic operations like addition, subtraction, multiplication and division and logical operations like AND, OR, Ex-OR, complement, compare etc. are performed. It includes the accumulator, the temporary register, the arithmetic and logic circuits and five flags. The temporary registers are used to store data temporarily during an arithmetic/logical operation. The result is stored in the accumulator and the flags are set or reset according to the result of the operation.

Register unit: The micro processor 8085 consists of different types of registers (a) Accumulator (b) General purpose register (c) Flag register (d) Temporary register (e) Program counter (f) Stack pointer

Accumulator:- It is an 8-bit register which is used for performing all the arithmetic's and logical operations. Accumulator is also known as 'register A'. During any mathematical or logical operation one of the operands should be present in the accumulator. The final result is also stored in the accumulator.

General purpose registers:- There are six 8-bit general purpose registers B, C, D, E, H & L. General purpose registers are used for temporary storage of data and intermediate results while the processor is executing the program. Two eight bit registers can be combined for handling 16-bit data. Combination of two 8-bit registers is known as pair. Valid register pairs are B-C, D-E, H-L. The H-L pair is used to address memories. Flag register: It is an 8-bit register in which five flip flops are used for checking condition. These flip flops are called flags. Each of these flags can have the value either one or zero to indicate certain condition after arithmetic and logical operation. The five flags present in 8085 µP are: (a) Sign flag (b) Zero flag (c) Auxiliary carry flag (d) Parity flag (e) Carry flag.

Temporary register:- Register W and Z are known as temporary registers. They are used by the µP for storing the data temporarily during execution of a program. They are 8-bit registers and are not accessible to the user.

Program Counter:- It is a 16-bit register which holds the address of the instructions. Initially it indicates towards the starting address of the program but after the first instruction is fetched the program counter automatically gets incremented by one and points towards the next instruction. This process continues till the end of the program.

Stack pointer: The stack pointer is a 16-bit register which basically serves two purposes--

(a) Points towards the stack memory. Initially it indicates the beginning of the stack memory. Whenever something is added to the stack, the stack pointer is decremented and whenever something is removed from the stack the stack pointer is incremented. Hence the stack pointer always points to the top of the stack.

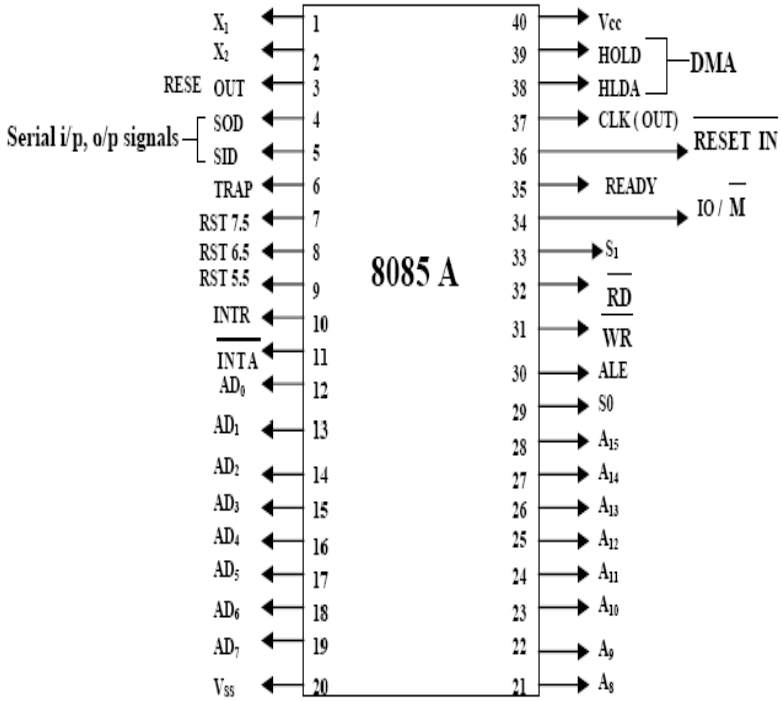
(b) Stack pointer also points towards the memory location where the µPhas to go after attending an interrupt or a subroutine; therefore it acts as a bookmark.

Timing and control unit: The timing and control unit generates timing and control signals which are necessary for the execution of instruction. It controls data flow between CPU and peripherals. It provides status, control and timing signals which are required for the operation of memory and I/O devices.

Q-2) **Draw the pin diagram of Intel 8085 µP and explain the function of each pin.**

[S-18 (Q2-c)], [S-16(Q1-c)], [W-17(Q1-c)]

Ans:-Intel 8085 µP is a 40 pin I.C. package fabricated on a single LSI chip. The pin diagram of Intel 8085 µP is as follows-



The pin configuration of 8085 µP can be classified into six groups namely:

1- High order Address bus
2- Multiplexed Address/Data bus
3- Control and status signals
4- Power supply and frequency signals
5- Externally initiated signals
and 6- Serial I/O ports

(1) <u>High order address bus</u>: The address bus is a group of 16-lines generally identified as **A_0-A_{15}**. The address bus is split into two segments **AD_0-AD_7** and **A_8-A_{15}** .The buses from **AD_0-AD_7**are low order address buses and **A_8-A_{15}**buses are high order address buses.

(2) <u>Multiplexed Address/Data bus</u>: The data bus is a group of 8-lines used for transfer of data. These lines are bidirectional i.e. data flows in both the directions from µP to peripherals and vice-versa. They are represented as **AD_0-AD_7**because they serve dual purpose. They are used as low-order address bus as well as data bus and therefore they are known as multiplexed address/data bus.

(3) Control and status signals: This group of signals includes two control signals, three status signals and one special signal. These signals are as follows:

***Control signal*:** - $\overline{RD}$ and $\overline{WR}$ are the two control signals which indicate that the data is to be read from or written into a selected memory or I/O location. Both are active low signals.

***Status signal*:-** IO/$\overline{M}$, S_1 and S_0 are the three status signals. IO/$\overline{M}$ is used to differentiate between I/O and memory operations. When it is high it indicates I/O operation and when it is low it indicates memory operation. S_1 and S_0 are also status signals but are rarely used in small systems.

Special signal:-ALE (Address Latch Enable) is a special signal used for demultiplexing address and data bus (**AD_0-AD_7**). It is a positive going pulse generated every time a machine cycle begins and so long as it remains positive it indicates that the bits on **AD_0-AD_7**are address bits.

(4) <u>Power supply and clock frequency</u>: The power supply required for 8085 µP is +5V. As shown in pin diagram V_{CC} is connected to +5V and V_{SS} is connected to the ground of the power supply. The µP operates on frequency of 3MHz, therefore an oscillator of frequency 6MHz is connected between pin no. 1

and 2 as the frequency is internally divided by two. CLK (OUT) i.e. Pin no. 37 is used as a system clock for other devices.

(5) Externally initiated signals: There are certain operations which can be initiated by external devices (or signals). For these externally initiated operations there are individual pins assigned on the microprocessor chip. Interrupts are also considered as externally initiated signal. Here is a brief explanation of interrupts and other externally initiated signals.

Interrupts: - The 8085 has five interrupt signals that can be used to interrupt a program execution. These interrupts are --

(a) TRAP:-The interrupt with highest priority. It is non-maskable and vectored interrupt i.e. the µP has to attend this interrupt immediately.

(b) RST 7.5, RST 6.5, RST 5.5:-These are known as Restart interrupts and have lower priority than TRAP but have higher priority than the INTR interrupt. They are vectored and maskable interrupts. Among the three the priority order is RST7.5>RST6.5>RST5.5

(c) INTR: It is a general purpose interrupt. It is maskable and non-vectored interrupt and has the least priority.

Reset:- When RESET pin is activated, the µP suspends all the internal operations and the program counter is cleared. Now the program execution can again begin at the zero memory address.

Ready:-The Ready signal is used to synchronize slower peripherals with the microprocessor. If the signal at READY pin is low the microprocessor enters into a wait state.

Hold:-This signal is used by the external devices to request the microprocessor for using the buses. When this signal is activated the µP leaves is control over the buses and makes them free for the peripherals to use.

$\overline{INTA}$:-Interrupt Acknowledge. This signal is used to acknowledge an interrupt.

HLDA:-Hold Acknowledge. This signal is used to acknowledge the HOLD request.

$\overline{RESET\ IN}$:-This is an active low signal. When it is activated the buses are tristated, the program counter is cleared and the microprocessor is reset.

RESET OUT: This signal indicates that the microprocessor is being reset. It can be used to reset other devices also.

(6) Serial I/O Ports: To send and receive data serially microprocessor has two pins 'SID' and 'SOD' by using these pins the µP can communicate with other µP and peripheral devices. In serial transmission, data bits are sent over a single line, one bit at a time. SID is Serial Input Data and SOD stands for Serial Output Data.

SID (Input) - On execution of the RIM instruction the data on this line is loaded into the seventh bit of the accumulator.

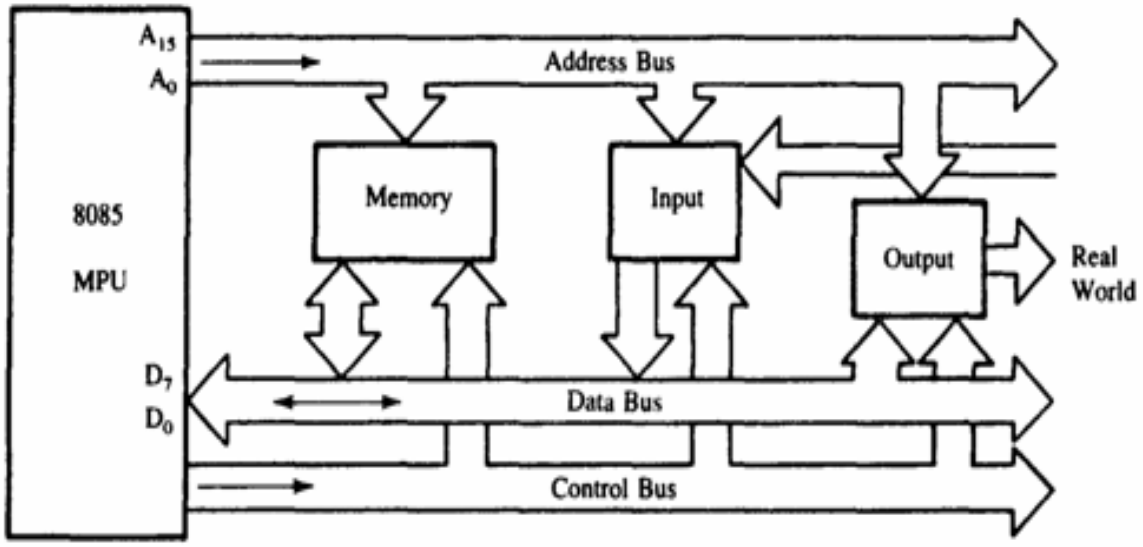
SOD(output)- When SIM instruction is executed the 7th bit of the accumulator is output on SOD line.

Q-3) **Draw the BUS structure of Intel 8085 µP and explain the function of address bus, data bus,**

control bus and system bus.[S-17(Q2-c)], [W-17(Q2-b)], [W-18(Q2-b)]

Ans:-Intel 8085 µP units are connected by set of lines. These sets of communication lines are called buses. The buses used in 8085 µP are-

(1) Address bus (2) Data bus (3) Control bus



Address bus: Address bus is a unidirectional group of 16 lines (i.e.) bits flow in one direction from the µP to the peripheral devices. The 8085 µP with its 16 address lines is capable of addressing 216=65536 (64K) memory locations. The µP performs 4-basic functions—Memory read, Memory write, I/O read and I/O write. The address must be present on the address bus as long as the read/write operation is not complete.

Data bus: Data bus carries data in binary form, between microprocessor and peripheral devices as well as memory. It is a group of 8-bits and is bidirectional. Data bus also carries instructions from memory to the microprocessor. Size of the bus therefore limits the number of possible instructions.

Control bus: The control bus is combination of various single lines that carry control signals. The control lines are not group of lines like address and data bus but are individual lines. Microprocessor generates specific control signals for every operation it performs. It is used for transmitting and receiving the synchronized control signals between µP and other devices.

System bus: To provide more modularity with reduced cost, memory and I/O buses were sometimes combined into a single unified system bus.

Chapter-III

Short type questions [2 marks each]

Q-1) **Define Opcode and Operand. [S-16 (Q5-a)]**

Ans:-The opcode or the operation code is the first part of an instruction which specifies the task to be performed by the µP.

The operand is the second part of the instruction which contains the data to be operated on.

Q-2) **What is difference between the two instruction MOV and MVI? [S-16 (Q6-a)], [S-15 (Q1-a)]**

Ans:-MOV is used to transfer the content of register/memory to memory/register. It is used for normal transfer of data. Ex- MOV r,M
MVI is used to transfer data directly to register/memory. It is used to transfer immediate data. Ex- MVI B, 57H

Q-3) **What are the function of DAA instruction and where this instruction is not used? [S-15(Q3-a)]**

Ans:-**DAA** is the decimal adjust accumulator. It operates on the result of ADD, ADC, etc instruction and gives the final result in decimal system. It is not used after SUB or SBB instruction for decimal subtraction.

Q-4) **What an instruction consists of ?[S-17(Q1-a)]**

Ans:-An instruction consists of two part- Operation code(Opcode) and Operand. The op-code is the command and the operand is an object to be operated on.

Q-5) **What is difference between SUB B and CMP B instruction of Intel 8085? [S-16 (Q1-a)], [W-17(Q3-a)], [S-19 (Q1-h)]**

Ans:-

SUB B- It subtracts the content of register-B from the content of the accumulator, and the result is placed in the accumulator.

CMP B- It is used to compare register with accumulator. The content of register B is subtracted from the content of the accumulator and status flags are set according to the result. But the result is discarded. The content of the accumulator remains unchanged.

Q-6) **Which instruction are used for subroutine program and what is cast statement of subroutine program? [S-16 (Q4-a)]**

Ans:-The CALL and RET instructions are used for subroutine program.

Q-7) **Which instruction are used for the operation of stack and subroutine? [S-15(Q7-a)]**

Ans:-LXI rp, data 16 and SPHL addr with PUSH & POP instructions are used for the operation of stack. The CALL and RET instructions are used for subroutine program.

Q-8) **Name different types of assembler directive of Intel 8085. [S-15(Q6-a)]**

Ans:-Assembler directives are instructions to the assembler concerning the program being assembled. They are not translated into machine code or assigned any memory locations. The different assembler directives are- ORG, END, EQU, DB, DW and DS.

Q-9) **What do you mean by self assembler and cross assembler? [S-16 (Q1-a)], [W-17 (Q4-a)]**

Ans:-**Self Assembler**- A self-assembler or resident assembler is an assembler which runs on the microcomputer for which it produces object codes(machine codes).

Cross Assembler- A cross assembler is an assembler that runs on a computer other than that for which it produces object codes.

Q-10) **Define addressing and why it is required?[S-17(Q5-a)]**
Ans:-The various techniques of specifying data/operand for instructions are called as addressing. It is required as each instruction needs certain data on which it has to operate.

Q-11) **What is the function of SBI and CPI? [W-18(Q1-a)]**
Ans:- The function of SBI and CPI is-
SBI:- Subtract immediate data from accumulator with borrow. The data and carry status are subtracted from the content of accumulator.

CPI:- Compare immediate data with accumulator. The 2nd byte of the instruction is data, which is subtracted from the content of accumulator. The status flags are set as per the result but accumulator is unchanged.

Q-12) **What is the difference between RLC and RAL? [W-18(Q4-a)]**
Ans:- The difference between RLC and RAL is-
RLC:- Rotate accumulator left. The content of accumulator is rotated left by one bit. Only CS flag is affected. The 7th bit moved to carry as well as to the zero bit of accumulator.
RAL:- Rotate accumulator left through carry. The content of accumulator is rotated left one bit through carry. The 7th bit of accumulator is moved to carry and the carry bit moved to zero bit of accumulator.

Q-13) **Name two instruction of Intel 8085 µP to clear the content of accumulator. [S-19(Q2-i)]**
Ans:- The two instructions to clear the content of accumulator are-
MVI A,00 and ANI 00

Medium type questions [5 marks each]

Q-1) **Describe different addressing modes available in 8085 µP and give explanation of each. [S-16(Q2-b)], [S-15 (Q1-b)], [W-17(Q3-b)], [S-19 (Q2-a)]**
Ans:-8085 µP has four addressing modes- Direct addressing, Register addressing, Register indirect addressing and Immediate addressing.
Direct addressing- In this mode of addressing the address of the operand/data is given in the instruction itself. Ex- STA 2400H.

Register addressing- In this addressing mode the operand is in one of the general purpose registers or accumulator. The opcode specifies the address of the registers in addition to the operation to be performed. Ex- MOV A, B.

Register Indirect addressing- In this mode of addressing the address of the operand is specified by a register pair. Ex- LXI H, 2500H
MOV A, M
Here, MOV A, M is the register indirect addressing.

Immediate addressing- In this mode the operand is specified within the instruction itself. Ex- MVI A,05.

Q-2) **What function is performed by each of the following instruction- MOV A, B ; LDAX ; LDA ; CMA ; AND B ; DCR C. [S-16(Q3-b)]**

Ans:-**MOV A, B** –Moves the content of register B to register A.

LDAX – Load accumulator indirect. Load the content of the memory location, whose address is in the register pair, is loaded into the accumulator.

LDA – The content of the memory location, whose address is specified by the 2nd and 3rd bytes of the instruction; is loaded into the accumulator.

CMA – Complement the accumulator. 1's complement of the content of the accumulator is obtained, and the result is placed in the accumulator.

AND B – The content of register B is logically ANDed with the content of accumulator and the result is placed in the accumulator.

DCR C – The content of register C is decremented by one.

Q-3) **Explain 1-byte, 2-byte and 3-byte instruction of 8085 with example? [W-17(Q7-b)]**

Ans:- **One-byte instructions –**

In 1-byte instruction, the opcode and the operand of an instruction are represented in one byte. The length of these instructions is 8-bit; each requires one memory location. The mnemonic is always followed by a letter (or two letters) representing the registers

Ex- MOV B, A; ADD B; CMP

Two-byte instructions –

Two-byte instruction is the type of instruction in which the first 8 bits indicates the opcode and the next 8 bits indicates the operand. This type of instructions need two bytes to store the binary codes. The mnemonic is always followed by 8-bit (byte) data.

EX- MVI A, 32H; MVI B, F2H

Three-byte instructions –

Three-byte instruction is the type of instruction in which the first 8 bits indicates the opcode and the next two bytes specify the 16-bit address. The low-order address is represented in second byte and the high-order address is represented in the third byte. These instructions would require three memory locations to store the binary codes. The mnemonic is always followed by 16-bit (or adr).

EX- LDA 2050H; JMP 2085H

Long type questions [7 marks each]

Q-1) **Explain different types of Instructions of 8085 with example. [S-17(Q3-c)]**

Ans:-

The instructions of 8085 µP is categorised into 4 groups as –

1) **Data transfer group** –
   (a) ***MOV–*** The content of register/memory is moved to memory/register. Ex- MOV B,A; MOV B,M; MOV M,C
   (b) ***MVI–*** Move the immediate data to register/memory. Ex- MVI A,05; MVI M,08
   (c) ***LXI–*** This instruction loads 16-bit immediate data into register pair. Ex- LXI H,2500H
   (d) ***LDA–*** The content of memory location, whose address is specified by the 2nd and 3rd bytes of the instruction is loaded into the accumulator. Ex- LDA 2400H

2) **Arithmetic group** –

(a) ***ADD-*** The content of register/memory location is added to the content of accumulator. The sum is placed in the accumulator. Ex- ADD B; ADD M

(b) ***ADC-*** The content of register/memory location and carry status are added to the content of the accumulator. The sum is placed in the accumulator. Ex- ADC B

(c) ***SUB-*** The content of register/memory location is subtracted from the content of accumulator and the result is placed in the accumulator. Ex- SUB C; SUB M

3) Logical group –

(a) ***ANA-*** The content of register/memory location is ANDed with the accumulator and result is placed in the accumulator. Ex- ANA C

(b) ***ORA-*** The content of register/memory location is ORed with the content of accumulator. The result is placed in accumulator. Ex- ORA B; ORA 2500H

4) Branch group –

(a) ***JMP-*** The program jumps to the instruction specified by the address unconditionally. Ex- JMP 2601H

(b) ***CALL-*** The program jumps to subroutine starting at address specified by the label. Ex- CALL 2600H

Chapter-IV

Short type questions [2 marks each]

Q-1) **Write different types of assembler directives of Intel 8085 µP . [S-19 (Q1-g)]**

Ans:- The assembler directives are used to direct the assembler to take an action. The different types of assembler directives of 8085 are- ORG (Origin), END, EQU (equate), DB (define byte), DW (define word) and DS (define storage).

Medium type questions [5 marks each]

Q-1) **Ten 8-bit nos are stored starting from memory location 4200H. Find the greatest of the array and stored it of a memory location 4300H. Write an assembly language for 8085 µP.**
[S-16(Q4-b)], [W-18(Q2-c)]

Ans:-***Program***-

As there is ten numbers in series; the count = 10 and is placed in memory location 2500H.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2000	21,FF,41		LXI	H,41FF H	Address for Count in reg pair
2003	4E		MOV	C, M	Count in register C
2004	23		INX	H	Address of 1st number
2005	7E		MOV	A, M	Get 1st number in Acc.
2006	0D		DCR	C	Decrement the count
2007	23	LOOP	INX	H	Address of next number
2008	BE		CMP	M	Compare next no with previous max. number
2009	D2,0D,20		JNC	AHEAD	No. Larger is in ACC. Go to label AHEAD
200C	7E		MOV	A, M	Yes, get larger no in ACC.
200D	0D	AHEAD	DCR	C	Decrement the count
200E	C2,07,20		JNZ	LOOP	
2011	32,00,43		STA	4300H	Store result in 4300H
2014	76		HLT		Stop.

Q-2) **Write a completely assembly language program to find the decimal subtraction of two 8-bit numbers and store the result in memory location 4080H. [S-15(Q2-b)], [W-17(Q7-c)]**

Ans:-***Program***- Let the two 8-bit numbers are 98 stored in memory location 2501H and 38 stored in memory location 2502H. Since, the DAA can't used after SUB or SBB; the 2nd number is converted into 10's complement.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2000	21,02,25		LXI	H,2502H	Get address of 2nd no in HL pair
2003	3E,99		MVI	A,99	Place 99 in accumulator
2005	96		SUB	M	9's complement of 2nd no
2006	3C		INR	A	10's complement of 2nd no
2007	2B		DCX	H	Get address of 1st number
2008	86		ADD	M	Add 1st no and 10's

					complement of 2nd no.
2009	27		DAA		Decimal adjustment
200A	32,80,40		STA	4080H	Store result in 4080H
200D	76		HLT		Stop

Q-3) **Explain basic assembler directives and write a program to add two 8-bit numbers and store the result (8-bit) using assembler directives. [S-17(Q3-b)]**

Ans:- Assembler directives are instructions to the assembler concerning the program being assembled. They are not translated into machine code or assigned any memory locations.

Program- Let two numbers are 49 & 56 which are placed in two consecutive memory locations 2501H and 2502H. The sum is stored in memory location 2503H.

Memory address	*Machine code*	*Mnemonics*	*Operands*	*Comments*
2000	21,01,25	LXI	H,2501H	Get the address of 1stno in H-L pair
2003	7E	MOV	A,M	The 1st number in accumulator
2004	23	INX	H	Increment the content of H-L pair
2005	86	ADD	M	Add 2nd number to 1st number
2006	23,03,25	STA	2503H	Store the result in 2503H
2009	76	HLT		Stop

Q-4) **Write a program to transfer 10 nos. data stored in consecutive memory location from 8000H/8086H to 9000H/8257H using 8085 instruction sets. [S-17(Q5-b)], [S-19(Q-6)]**

Ans:-

Program-

As there is ten numbers in series; the count = 10 and is placed in memory location 7FFFH.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2400	21,FF,7F 21,85,80		LXI	H,7FFFH H,8085H	Get memory address of count
2403	4E		MOV	C,M	Count in register C
2404	23		INX	H	Source address of data in H-L pair
2405	11,00,90 11,57,82		LXI	D,9000H D,8257H	Destiny address of data in D-E pair
2408	7E	LOOP	MOV	A,M	Data from source address to Acc
2409	EB		XCHG		Destiny address in H-L pair
240A	77		MOV	M,A	Data to destiny address
240B	EB		XCHG		Source address in H-L pair
240C	23		INX	H	Source address of next data
240D	13		INX	D	Destiny address of next data
240E	0D		DCR	C	Decrement the count
240F	C2,08,24		JNZ	LOOP	Jump to label LOOP
2412	76		HLT		Stop

Long type questions [7 marks each]

Q-1) **Write a program to transfer 16 memory location from one memory location to other memory location in user area using 8085 instruction. [S-16(Q2-c)]**

Ans:-***Program***- As there is 16 numbers in series; the count = 16 and is placed in memory location 2000H.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2400	21,00,20		LXI	H,2000H	Get memory address of count
2403	4E		MOV	C,M	Count in register C
2404	23		INX	H	Source address of data in H-L pair
2405	11,01,22		LXI	D,2201	Destiny address of data in D-E pair
2408	7E	LOOP	MOV	A,M	Data from source address to Acc
2409	EB		XCHG		Destiny address in H-L pair
240A	77		MOV	M,A	Data to destiny address
240B	EB		XCHG		Source address in H-L pair
240C	23		INX	H	Source address of next data
240D	13		INX	D	Destiny address of next data
240E	0D		DCR	C	Decrement the count
240F	C2,08,24		JNZ	LOOP	Jump to label LOOP
2412	76		HLT		Stop

Q-2) **Write delay subroutine program of 10ms in 8085 μP based system whose clock frequency is 3MHz. (Give also suitable flow chart).[S-16(Q7-c)]**

Ans:-***Program***-

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2000	01,3D		LXI	B, 3D H	Load register pair B-C with 16-bit count.
2002	0B	LOOP	DCX	B	Decrement the count
2003	78		MOV	A,B	Move content of register B to accumulator
2004	B1		ORA	C	Check if B and C are zero
2005	C2		JNZ	LOOP	If B-C is not zero, jump to LOOP
2006	C9		RET		Return to main program

Instruction	States
LXI B	10
DCX B	6
MOV A,B	4
ORA C	4
JNZ	7/10
RET	10

Now, T_1 = 24 x T xNo. of 100 ps (N) sec
= 24 x0.333 x 10^{-6} N
and T_0 = 10 xT(for LXI B) = 3.333 x10^{-6} sec
Total Delay = $T_1 + T_0$
= 0.5 msec + 24 x0.333 x 10^{-6} N + 3.333 x10^{-6}
=> 0.5 x100 μs = { (24 x0.333 x N) + 3.333} 10^{-6}sec
=> 500 sec = (7.992 N + 3.333) sec
=> N (no. of loops) = (500 – 3.333)/7.992
= $(62.14)_{10}$ = $(3E)_H$; So, N-1 = $(3D)_H$

Q-3) **Write a program to find smallest number in the array using 8085 instruction. [S-17(Q5-c)], [S-15 (Q3-c)]**

Ans:-***Program***- As there is array of numbers in series (say 03 nos); the count =03 is placed in memory location 2500H. The numbers are stored in memory location 2501 to 2503H. Result will be stored in memory location 2504H.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2000	21,00,25		LXI	H,2500 H	Address for Count in H-L pair
2003	4E		MOV	C, M	Count in register C
2004	23		INX	H	Address of 1st number
2005	7E		MOV	A, M	Get 1st number in Acc.
2006	0D		DCR	C	Decrement the count
2007	23	LOOP	INX	H	Address of next number
2008	BE		CMP	M	Compare next no with previous smallest number. Is prev. Smallest < the next
2009	D2,0D,20		JNC	AHEAD	Yes,smaller is in ACC. Go to label AHEAD
200C	7E		MOV	A, M	No, get next no in ACC.
200D	0D	AHEAD	DCR	C	Decrement the count
200E	C2,07,20		JNZ	LOOP	
2011	32,04,25		STA	2504H	Store result in 2504H
2014	76		HLT		Stop.

Q-4) **Design a time delay by using register pair. Find the maximum time delay using one register. [S-15(Q5-b)], [W-17 (Q5-b)]**

Ans:-***Program***-

Labels	*Mnemonics*	*Operands*	*Comments*
	LXI	D, FFFF	Get FFFF in register pair D-E
LOOP	DCX	D	Decrement count
	MOV	A, D	Move content of register D to accumulator
	ORA	E	Check if D and E are zero
	JNZ	LOOP	If D-E is not zero, jump to Loop
	RET		Return

Maximum delay using one register:-

Instruction	States
MVI B, 10H	7 X 1
DCR B	4 X 16
JNZ	10 X 15 + 7 X 1
RET	10 X 1

Total States = 7x1+4x16+(10x15+7x1)+10x1
= 7+64+150+7+10 = 238
Time for one state for 8085 is 320ns
Delay time = $238 \times 320 \times 10^{-9}$ second
= 0.238×0.320 millisecond
= 0.07616 millisecond

For maximum delay reg. B is loaded by FF (255 decimal).
So, Max.m delay using one register is –
=7x1+4x255+(10x254+7x1)+10x1
= 7+1020+2540+7+10
= 3584 states
=$3584 \times 320 \times 10^{-9}$ second
=1.116889 millisecond (Ans)

Q-5) **Write an assembly language program to find the greatest number between two 8-bit numbers. The two numbers are to be stored in memory location 8501, 8502 and the output will be stored in memory location 8503H.[W-17(Q3-c)]**

Ans:- ***Program-***

The two 8-bit numbers are stored in memory location 8501H and 8502H. The result is stored in the memory location 8503H.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2000	21,01,85		LXI	H,8501 H	Address of first no in H-L pair
2003	4E		MOV	A, M	Get 1st number in Acc.
2004	23		INX	H	Address of second no in H-L pair
2005	BE		CMP	M	Compare second no with first number. Is 2nd number > 1st number?
2006	D2,0A,20		JNC	AHEAD	No, larger is in ACC. Go to label AHEAD
2009	7E		MOV	A, M	Yes, get second no in ACC.
200A	32,03,85		STA	8503H	Store result in 8503H
200D	76		HLT		Stop.

Q-6) **Write a program to find largest number in an data array of in data stored in memory location with example. [W-18(Q2-c)]**

Ans:-***Program-*** As there is array of numbers in series (say 03 nos); the count =03 is placed in memory location 2500H. The numbers are stored in memory location 2501 to 2503H. Result will be stored in memory location 2450H.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2000	21,00,25		LXI	H,2500 H	Address for Count in H-L pair
2003	4E		MOV	C, M	Count in register C
2004	23		INX	H	Address of 1st number
2005	7E		MOV	A, M	Get 1st number in Acc.
2006	0D		DCR	C	Decrement the count
2007	23	LOOP	INX	H	Address of next number
2008	BE		CMP	M	Compare next no with previous maximum number. Is next number > prev.maximum?
2009	D2,0D,20		JNC	AHEAD	No, larger is in ACC. Go to label AHEAD
200C	7E		MOV	A, M	Yes, get larger no in ACC.
200D	0D	AHEAD	DCR	C	Decrement the count
200E	C2,07,20		JNZ	LOOP	
2011	32,50,24		STA	2450H	Store result in 2450H
2014	76		HLT		Stop.

Data
2500- 03
2501- 98
2502- 75
2503- 99

Result
2450- 99

Q-7) **Write a complete assembly language program to decimal subtraction of two 8-bit data present in two consecutive memory location and store the result in next memory location. [S-19(Q2-c)]**

Ans:-***Program-*** Let two the decimal 8-bit numbers are placed in memory location 2501H & 2502H. The result will be stored in memory location 2503H. DAA instruction cannot be used after SUB or SBB instruction for decimal subtraction. Hence, the 2nd number is converted into 10's complement. Then it is added to the 1st number.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2000	21,02,25		LXI	H,2502 H	Get address of 2nd no in H-L pair
2003	3E, 99		MVI	A, 99	Place 99 in accumulator
2005	96		SUB	M	9's complement of 2nd number
2006	3C		INR	A	10's complement of 2nd no
2007	2B		DCX	H	Get address of 1st number
2008	86		ADD	M	Add 1st number and 10's complement of 2nd number
2009	27		DAA		Decimal adjustment
200A	32,03,25		STA	2503H	Store result in 2503H
200D	76		HLT		Halt or Stop

Q-8) **Write a complete ALP to add two 8-bit numbers in memory where sum is 16-bit and store the result in another memory location. [W-18(Q5-b)]**

Ans:-***Program-*** Let the numbers are placed in memory location 2501H to 2502H. The result will be stored in memory location 2450H & 2451H. The count is in 2500H and the initial value of the sum is made 00.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
2400	21,00,25		LXI	H,2500 H	Address of count in H-L pair
2403	4E		MOV	C, M	Count in register C
2404	3E, 00		MVI	A, 00	LSB of sum=00(initial value)
2406	47		MOV	B, A	MSB of sum=00(initial value)
2407	23	LOOP	INX	H	Address of next data
2408	86		ADD	M	Prev. Sum + next data
2409	D2,0D,24		JNC	AHEAD	Is carry? No, go to AHEAD
240C	04		INR	B	Yes, Add carry to MSB of sum
240D	0D	AHEAD	DCR	C	Decrement count
240E	C2,07,24		JNZ	LOOP	Is C=0? No, jump to LOOP
2411	32,50,24		STA	2450H	Store LSBs of the sum
2414	78		MOV	A, B	Get MSB of sum into ACC.
2415	32,51,24		STA	2451H	Store MSBs of the sum
2418	76		HLT		Halt

Q-4) **Design a time delay by one register. Find the maximum time delay using one register. [W-18 (Q4-b)], [S-19 (Q2-d)]**

Ans:-***Program-*** *The delay time depends on the number loaded in the register. Ex- 10 in register B.*

Memory address	*Machine codes*	*Labels*	*Mnemonics*	*Operands*	*Comments*
FC00	06,10		MVI	B,10H	Get 10 in register B
FC02	05	LOOP	DCR	B	Decrement register B
FC03	C2,02,FC		JNZ	LOOP	Is content of B is zero? No, jump to Loop. Yes, go ahead
FC06	C9		RET		Return

Maximum delay using one register:-

Instruction	States
MVI B, 10H	7 X 1
DCR B	4 X 16
JNZ	10 X 15 + 7 X 1
RET	10 X 1

Total States = 7x1+ 4x16+ (10x15+ 7x1)+ 10x1 = 7+64+150+7+10 = 238

Time for one state for 8085 is 320ns

Delay time = $238 \times 320 \times 10^{-9}$ second

= 0.238x0.320 millisecond

= 0.07616 millisecond

For maximum delay reg. B is loaded by FF (255 decimal).

So, Maximum delay using one register is –

= 7x1+4x255+(10x254+7x1)+10x1 = 7+1020+2540+7+10 = 3584 states

= $3584 \times 320 \times 10^{-9}$ second =1.116889 millisecond (Ans)

Chapter-V

Short type questions [2 marks each]

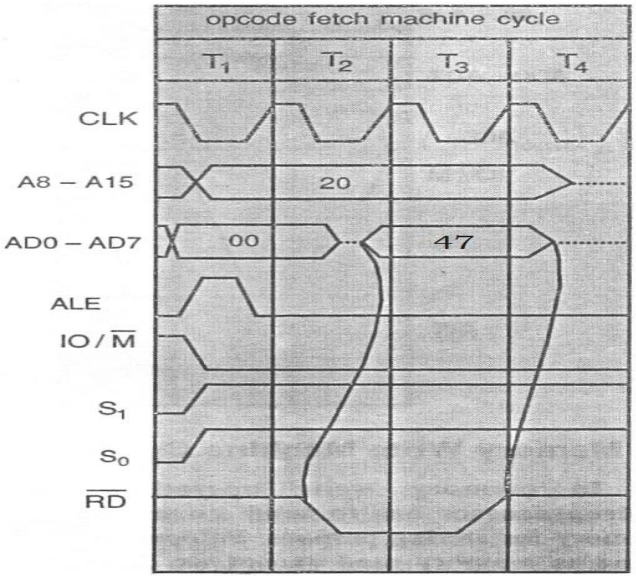
Q-1) **Name different machine cycles with their functions for INR M instruction. [S-19 (Q1-d)]**
Ans:- The instruction INR M has 3 machine cycles-

M1- Opecode fetch, M2- Memory read and M3- Memory write

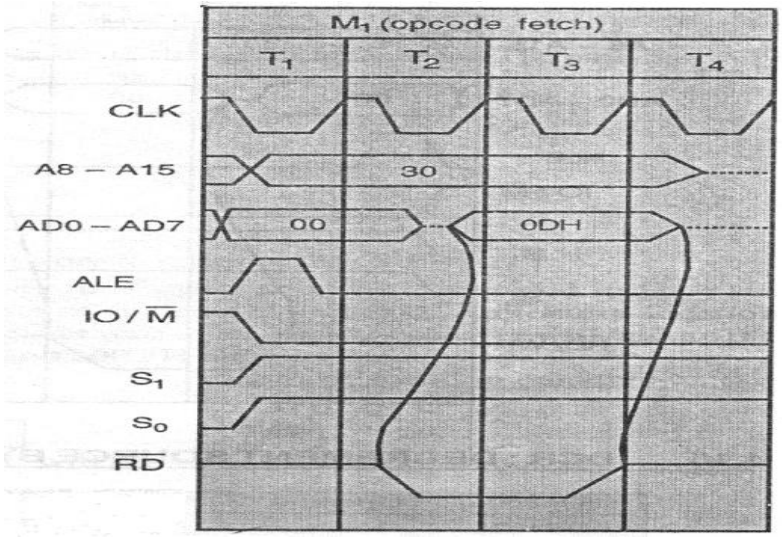
Medium type questions [5 marks each]

Q-1) **Draw the timing diagram for the instruction MOV B,A instruction of 8085 µP. [S-16 (Q1-b)]**
Ans:- The timing diagram for instruction MOV B,A is as follows-

The instruction MOV B,A is an one byte instruction and the opcode for this instruction in 47. So, it needs only one machine cycle which is the opcode fetch cycle.



Q-2) **Draw a neat sketch for the timing diagram for the DCR instruction.[S-15(Q4-b)]**
Ans:- The timing diagram for instruction DCR is as follows-



Q-3) **Define T-state, Fetch cycle, Machine cycle and Instruction Cycle. [S-17 (Q4-b)], [W-18 (Q5-c)]**

Ans:-**T-State**:- It is defined as one subdivision of operation performed in one clock period. These subdivisions are internal states synchronized with the system clock & each T-state is equal to one clock period.

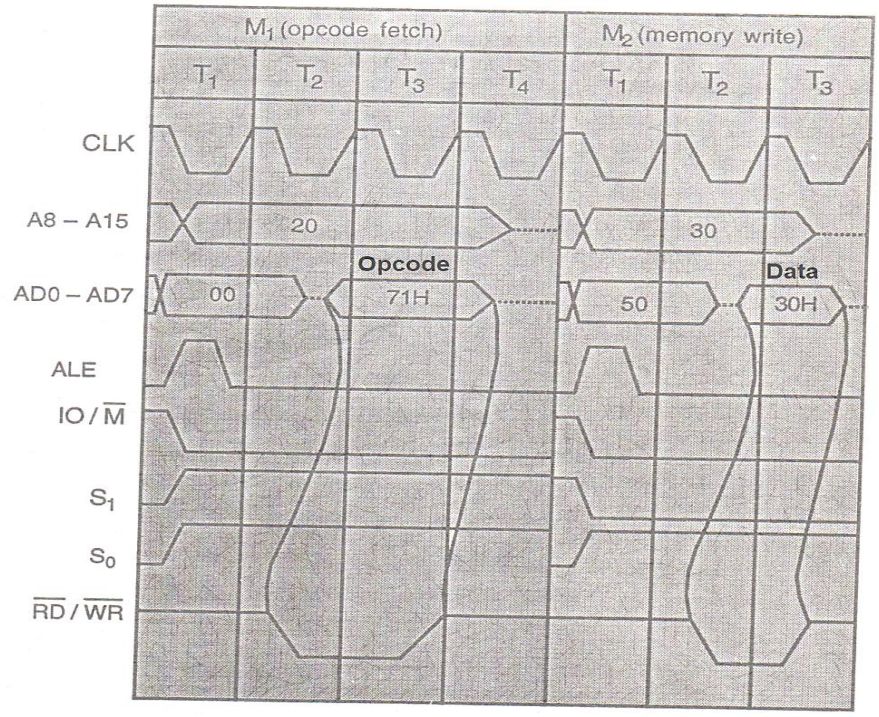
Fetch cycle:- It is the time required to fetch an opcode from a particular memory location. A fetch cycle consists of 3 T-states.

Machine cycle:- It is a time required by µP to complete the operation of accessing either memory or I/O device, called machine cycle.

Instruction cycle:- The steps required by CPU to fetch and execute an instruction is called Instruction cycle. The first machine cycle is opcode fetch cycle of the instruction cycle.

Q-4) **Draw the timing diagram for the instruction MOV M, D of 8085 µP.**
[W-17 (Q4-b)], [W-18 (Q5-c)]

Ans:- The timing diagram for instruction MOV M,D is as follows-

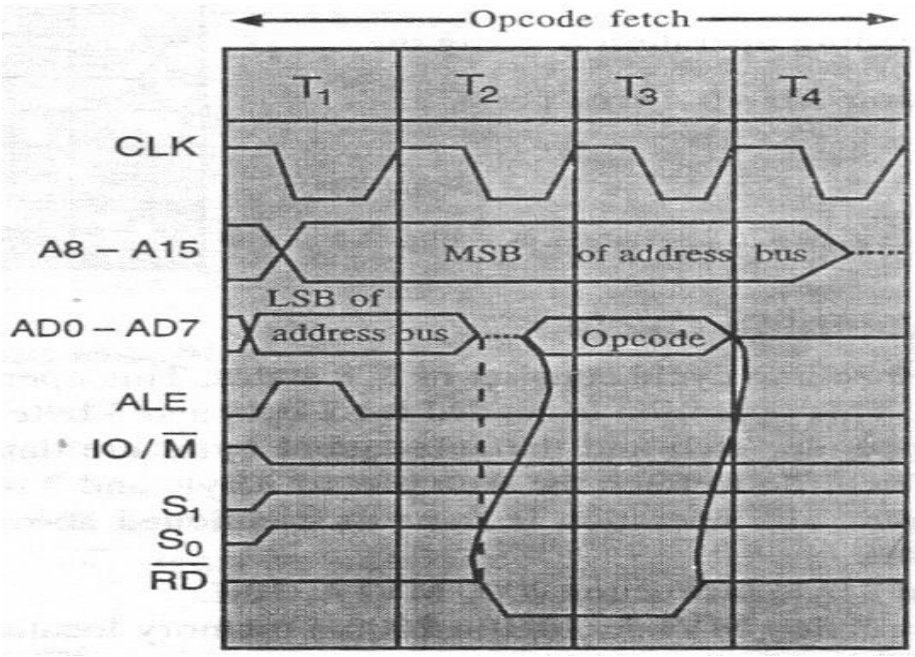


Timing diagram for MOV M, D

Long type questions [7 marks each]

Q-1) **Draw the op-code fetch machine cycle of 8085 and discuss. [S-17(Q4-c)]**

Ans:- The timing diagram for Op-code fetch machine cycle is as follows-

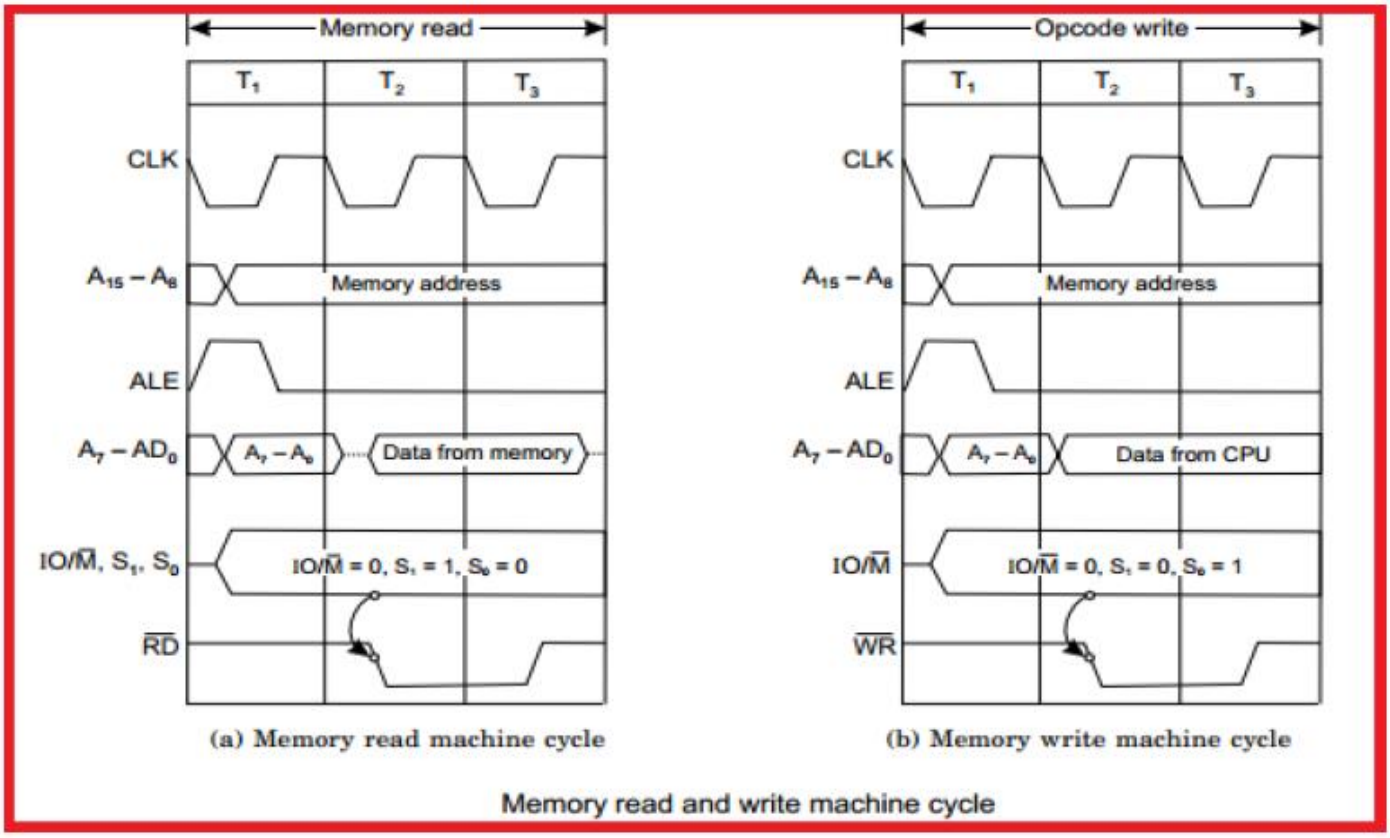


In a fetch cycle the µP fetches the opcode of an instruction from the memory. T_1, T_2, T_3 and T_4 are consecutive four clock cycles. The µP issues a low IO/M signal to indicate that it wants to make communication with the memory. Again µP sends out high S_0 and S_1 signals to indicate that it is going to perform fetch operation.

During the first clock cycle (T_1), the µP sends out the address of the memory location where the opcode is available. The 16-bit memory address is sent through the address bus A and address/data bus AD. The 8 MSBs of the memory address are sent over the A bus, and 8 LSBs over the AD bus. Since the AD bus is used in time multiplexed mode, therefore, it has to be made available to carry data during T_2 and T_3. To accomplish this µP sends an ALE signal to latch the 8-LSBs of the memory address either in the memory or an external latch so that the complete 16-bit memory address may be available in the subsequent clock cycles.16-bit memory address is needed by the memory to obtain the opcode from the given memory address. During T_2, AD bus becomes ready to carry data. In T_2 the µP makes RD low. Now memory gets the opcode from the specified memory location and places it on the data bus. During T_3, the opcode is placed in the instruction register, IR which is within the µP. The memory is disabled when RD goes high during T_3. The fetch cycle is completed by T_3. The opcode is decode in T_4. The µP examines READY signal in T_2. If it is high, the µP enters into T_3 state. If it is low, the µP inserts a wait state in between T_2 and T_3.

Q-2) **Draw the timing diagram for memory read, memory write, I/P read, I/P write machine cycle. [S-1(Q5-c)]**

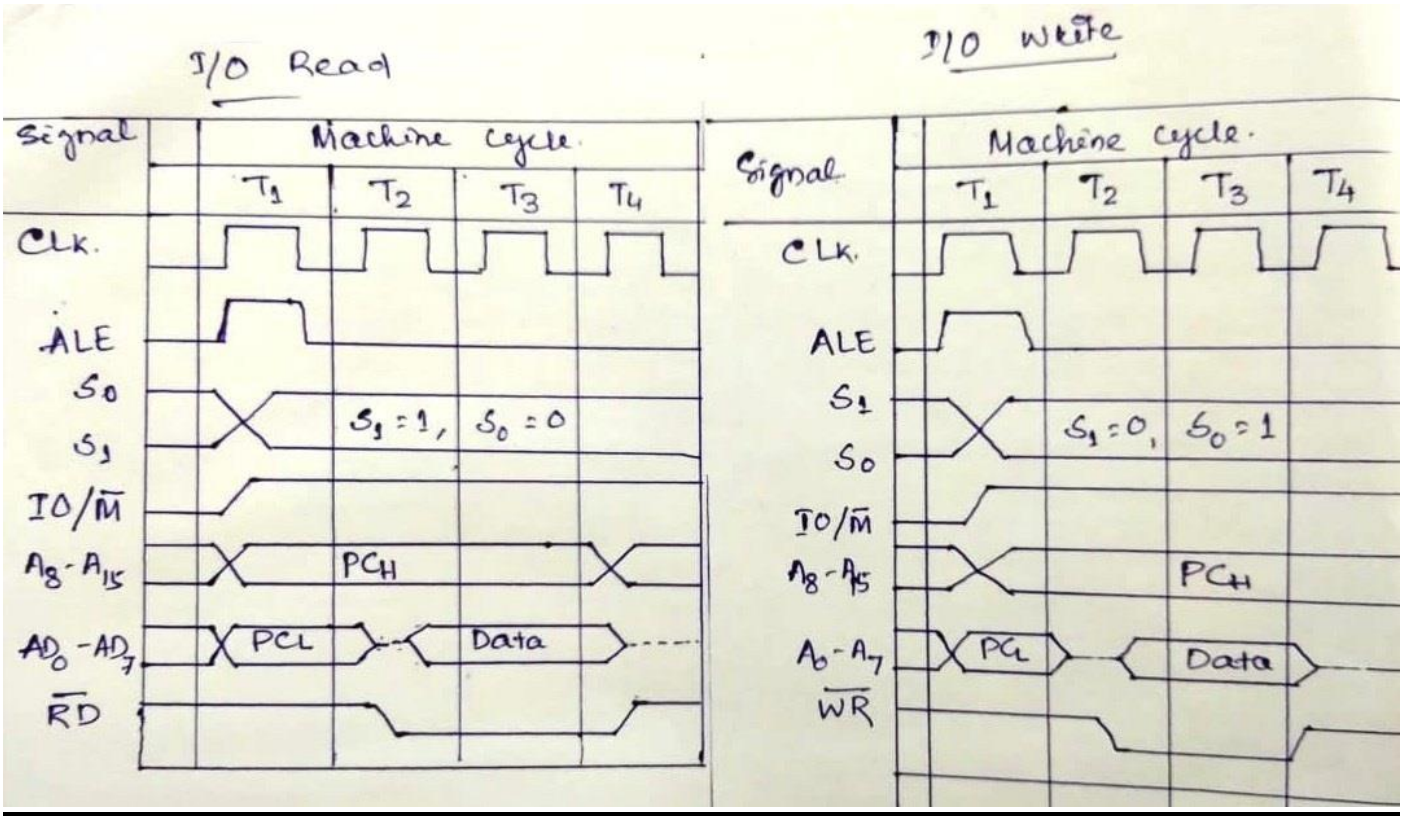
Ans:- The timing diagram for instruction memory read is as follows-



(a) Memory read machine cycle

(b) Memory write machine cycle

Memory read and write machine cycle


I/O Read
Signal
Machine cycle.
T1
T2
T3
T4
CLK.
ALE
S0
S1
IO/M
A8-A15
AD0-AD7
RD
S1 = 1, S0 = 0
PCH
PCL
Data
I/O Write
Signal
Machine cycle.
T1
T2
T3
T4
CLK.
ALE
S1
S0
IO/M
A8-A15
A0-A7
WR
S1 = 0, S0 = 1
PCH
PCL
Data

Chapter-VI

Short type questions [2 marks each]

Q-1) **How many modes can 8255 operate and name them?[S-16 (Q1-a)]**

Ans:- The 8255 can operate in three modes. They are-
(i) Mode 0- Simple input/output
(ii) Mode 1- Strobed input/output
(iii) Mode 2- Bidirectional port

Q-2) **What is the function of USART? [S-17(Q7-a)]**

Ans:- The Intel 8251 is a programmable communication interface which is called universal synchronous/asynchronous receiver/transmitter. It is used to transmit/receive serial data. It accepts data in parallel format from the µP and converts them into serial data for transmission and vice-versa.

Q-3) **Write the full form of PPI, PIC, DMA and USART. [S-16 (Q1-a)], [W-17 (Q4-a)], [W-18 (Q7-a)]**
Ans:- The full form of --

PPI:- It is Programmable Peripheral Interface (Intel 8255)

PIC:- It is Programmable Interrupt Controller (Intel 8259)

DMA:- It is Direct Memory Access (Intel 8257/8237A)

USART:- It is Universal Synchronous/Asynchronous Receiver/Transmitter (Intel 8251)

Q-4) **What do you mean by DMA techniques? Which pins of 8085 belongs to this group. [S-16 (Q1-a)]**
Ans:- The bulk of data transfer from fast I/O devices to memory or its vice-versa through the accumulator is a time consuming process. For such a situation the direct memory access technique (DMA) is required. In DMA technique, data are directly transferred from an I/O device to RAM or from RAM to an I/O device.

Q-5) **What do you mean by Delay routine? [S-16 (Q7-a)]**
Ans:- Delay routine is a program which is used to provide the desired delay in industrial control before issuing the control signal by the microcomputer. To generate delay a few registers of the µP are loaded with desired numbers and then decremented to zero.

Q-6) **What are the different methods of interfacing of I/O devices to 8085 based system? [S-15 (Q2-a)]**
Ans:- The different methods of interfacing of I/O devices to 8085 based systems are – (i) Memory mapped I/O scheme
(ii) I/O mapped I/O scheme

Q-7) **Name different pins come under DMA and Serial data transfer control group? [S-19 (Q1-f)]**
Ans:- The pins come under DMA are pin no. 38(HOLD) & 39(HLDA).

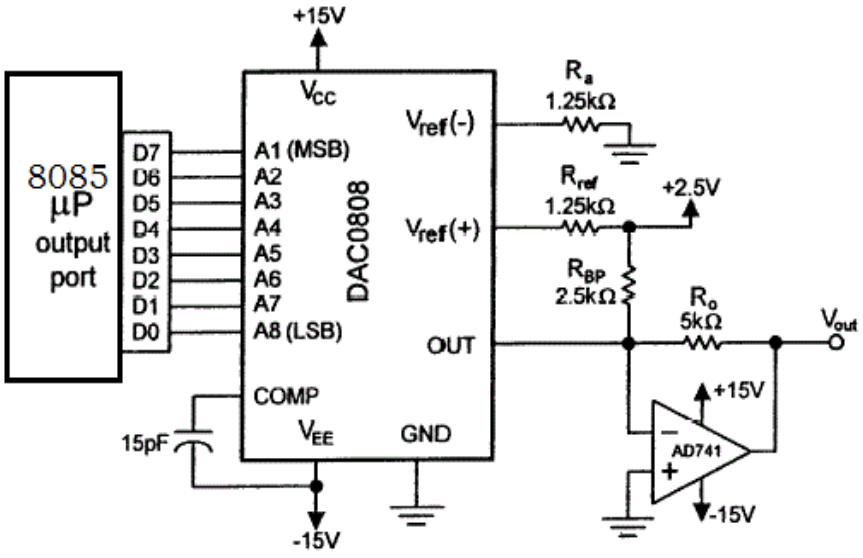
The pins come under serial data transfer control group are pin 4(SOD) & 5(SID).

Medium type questions [5 marks each]

Q-1) **Describe briefly about DAC and explain how they interface with 8085 µP with neat block diagram. [W-17(Q3-b)]**

Ans:- Digital to Analog Conversion or DAC is a device which is used to convert a digital code into an analog signal. DAC allows the digital control of machines, equipment, household appliances. The data can be converted to clean digital form using gates which are designed to be on or off depending on the value of the incoming signal. Data in clean binary digital form can be converted to an analog form by using a summing amplifier.

The interfacing of DAC 0808 with 8085 µP is shown below. The PPI 8255 is used as parallel port to send the digital data to DAC. From pin 5 to pin 12 of DAC 0800 are for digital input. Pin 5 is MSB, 12 is LSB. These are connected to Port B of 8255.



Program:-

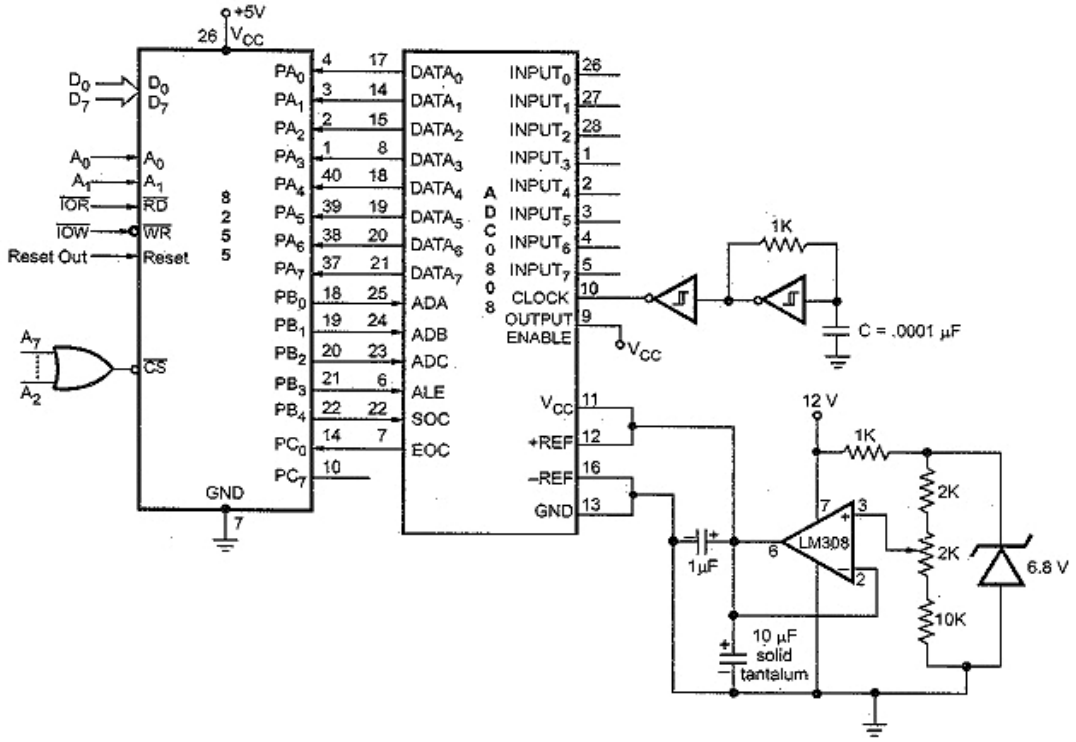
Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
FC00	3E,98		MVI	A, 98H	Get control word for 8255
FC02	D3, 0B		OUT	0B	Initialise ports
FC04	3E, 80		MVI	A, 80	Get 80 for digital input to DAC
FC06	D3, 09		OUT	09	Input 80 to DAC through Port B
FC08	76		HLT		Stop

When chip select of DAC is enabled then DAC will convert digital input value given through portliness PB0-PB7 to analog value. The analog value o/p from DAC is a current quantity. The current quantity is converted to voltage using OPAMP base CVC. The standard o/p voltage will be 10V when FF is inputted, will be5V when 80 is inputted and will be 0V when 00 is inputted.

For bipolar operation, connect pin 2 of DAC to non-inverting terminal of OPAMP. This common point is earthed through a 5K Ohm resistor. Then the o/p voltage will be -10V when 00 is inputted, will be0V when 80 is inputted and will be +10V when FF is inputted.

Q-2) **Describe briefly about ADC and explain how they interface with 8085 μP with neat block diagram. [S-16(Q6-b)]**

Ans:- Analog to Digital Conversion or ADC is a device which is used to convert a 8-bit analog signal to digital. It uses successive approximation method for conversion.



The device starts conversion when a signal is applied through the WR pin. The input analog signal is given through pin Vin+ and Vin-. When both CS and RD are low then digital output is available Do-D7 output lines. When INTR the conversion goes low then is over. The clock frequency of conversion is about 100KHz to 800KHz. Reference voltage is given through pin Vref, when it is left open then Vref=Vcc, otherwise half of the maximum analog input must be given.

Q-3) **Distinguish between memory mapping and I/O mapping with their application. [S-16 (Q7-b)]**

Ans:-

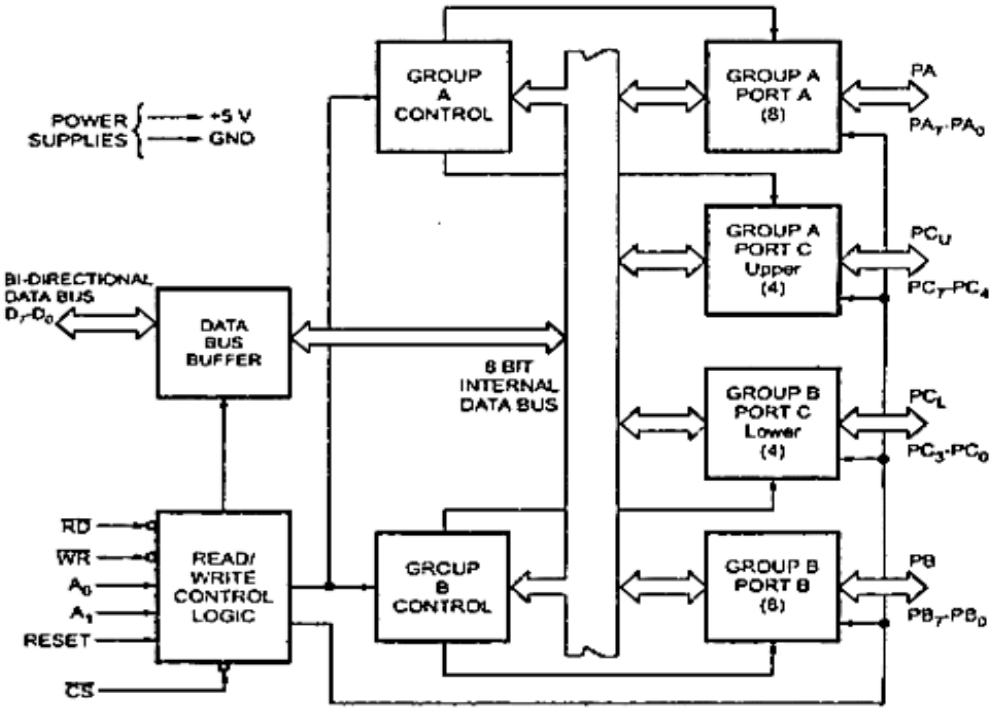
In the interfacing of the microprocessor, I/O is a general-purpose port used by processor or controller to handle peripherals Connectedtoit.

I/O mapped I/Os have a separate address space from the memory. The total addressed capacity is the number of I/Os connected and the memory connected. Separate I/O-related instructions are used to access I/Os. A separate signal is used for addressing an I/O device.

Memory-mapped I/Os share the memory space with external memory. So, total addressed capacity is memory connected only. It is considered to be underutilization of resources if this processor supports I/O-mapped I/O. The instructions used to access I/Os are same as that used for memory. There is only one address space.

Q-4) **Describe the internal architecture of 8255 interfacing chip with neat block diagram. [S-17 (Q6-b)]**

Ans:- Figure shows the internal block diagram of 8255A. It consists of data bus buffer, control logic and Group A and Group B controls.



Block Diagram of 8255

1. **Data Bus Butter**: This tri-state bi-directional buffer is used to interface the internal data lilts of 8255 to the system data bus. Input or Output instructions executed by the CPU either Read date from or Write data into the buffer. Output data from the CPU to the ports or control register, and input data to the CPU from the ports or status register are all passed through the buffer.
2. **Control Logic**: The control logic block accepts control bus signals as well as inputs from the address bus, and issues commands to the individual group control blocks (Group A control and Group B control). It issues appropriate enabling signals to access the required data/control words or status word. The input pins for the control logic section are described here.
3. **Group A and Group B Controls**: Each of the Group A and Group B control blocks receives control words from the CPU and issues appropriate commands to the ports associated with it. The Group A control block controls Port A and PC_7-PC_4 while the Group B control block controls Port B and PC_3-PC_0.
4. **Port A**: This has an 8-bit latched and buffered output and an 8-bit input latch. It can be programmed in three modes: mode 0, mode 1 and mode 2.
5. **Port B**: This has an 8-bit data I/O latch/ buffer and an 8-bit data input buffer. It can be programmed in mode 0 and mode 1.
6. **Port C**: This has one 8-bit unlatched input buffer and an 8-bit output latch/buffer. Port C can be spitted into two parts and each can be used as control signals for ports A and B in the handshake mode. It can be programmed for bit set/reset operation.

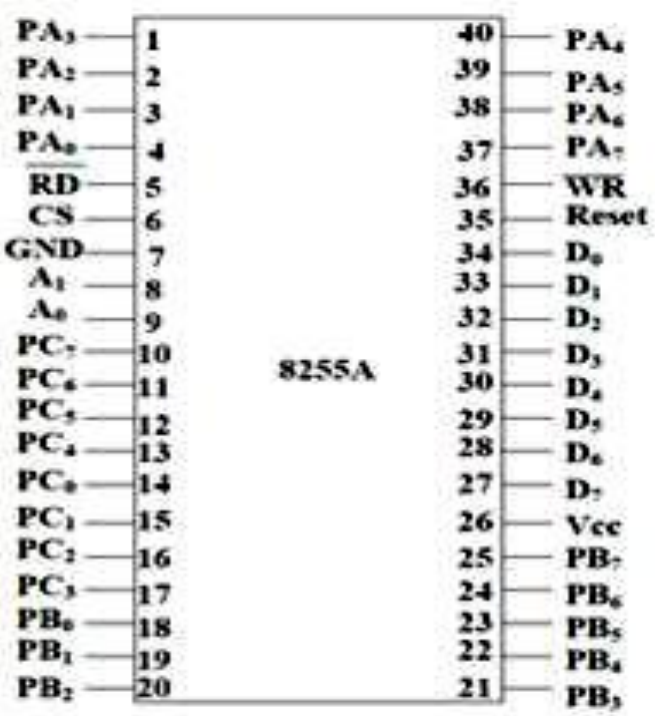
Long type questions [7 marks each]

Q-1) **Draw the internal block diagram of 8255A chip and explain the function of each pin. [S-18 (Q6-c)], [S-15 (Q4-c)], [S-16 (Q4-c)], [W-17(Q2-c)], [W-18 (Q6-c)] [S-19 (Q5)]**

Ans:-

A programmable peripheral interface is a multiport device. It has three 8-bit ports, namely Port A, Port B and Port C. The Port C is divided into two 4-bit ports. Each port can be programmed either as an input port or an output port. The pin diagram of Intel 8255A is given below. It is a 40 pin IC package. It operates on a single 5V dc supply.

PA_0 and PA_7:- These are 8 pins of port-A
PB_0 and PB_7:- These are 8 pins of port-B
PC_0 and PC_3:- These are 4 lower pins of port-C
PC_4 and PC_7:- These are 4 upper pins of port-C



$\overline{CS}$(Chip Select):- It is a chip select signal. The LOW status of this signal enables communication between the CPU and 8255.

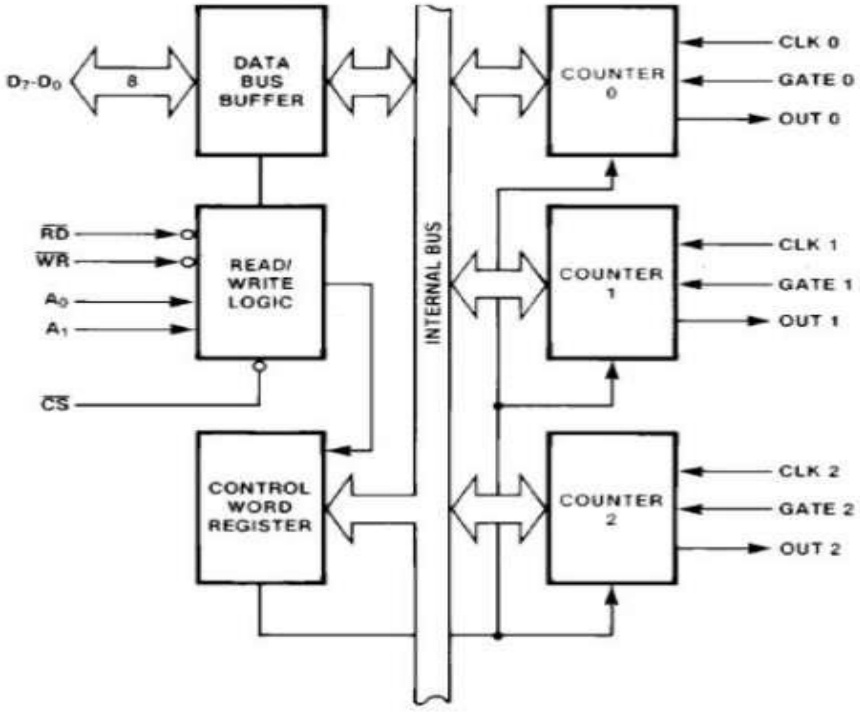
$\overline{RD}$ (Read):- When $\overline{RD}$ goes LOW the 8255 sends out data or status information to the CPU on the data bus. In other words it allows the CPU to read data from the input port of 8255.

$\overline{WR}$ (Write):- When $\overline{WR}$ goes LOW the CPU writes data or control word into 8255. The CPU writes data into the output port of 8255 and the control word into the control word register.

A_0 and A_1:- The selection of input port and control word register is done using A_0 and A_1 in conjunction with $\overline{RD}$ and $\overline{WR}$. A_0 and A_1 are normally connected to the least significant bits of the address bus.

Q-2) **What do you mean by Universal timer 8253? Explain each block with suitable block diagram. [S-16 (Q5-c)]**

Ans:- The architecture of 8253 looks as follows –



In the above figure, there are three counters, a data bus buffer, Read/Write control logic, and a control register. Each counter has two input signals - CLOCK & GATE, and one output signal - OUT.

Data Bus Buffer:-It is a tri-state, bi-directional, 8-bit buffer, which is used to interface the 8253 to the system data bus. It has three basic functions –

- Programming the modes of 8253
- Loading the count registers
- Reading the count values

Read/Write Logic:-It includes 5 signals, i.e. RD, WR, CS, and the address lines A_0 & A_1. In the peripheral I/O mode, the RD and WR signals are connected to IOR and IOW, respectively. In the memory mapped I/O mode, these are connected to MEMR and MEMW.

Address lines A_0 & A_1 of the CPU are connected to lines A_0 and A_1 of the 8253/54, and CS is tied to a decoded address. The control word register and counters are selected according to the signals on lines A_0 & A_1.

A_1	***A_0***	***Result***
0	0	Counter 0
0	1	Counter 1
1	0	Counter 2
1	1	Control Word Register

Control Word Register:-

This register is accessed when lines A_0 & A_1 are at logic 1. It is used to write a command word, which specifies the counter to be used, its mode, and either a read or write operation. Following table shows the result for various control inputs.

A1	*A0*	*RD*	*WR*	*CS*	*Result*
0	0	1	0	0	Write Counter 0
0	1	1	0	0	Write Counter 1
1	0	1	0	0	Write Counter 2
1	1	1	0	0	Write Control Word
0	0	0	1	0	Read Counter 0
0	1	0	1	0	Read Counter 1
1	0	0	1	0	Read Counter 2
1	1	0	1	0	No Operation
X	X	1	1	0	No Operation
x	x	x	x	1	No Operation

Counters:-Each counter consists of a single, 16 bit-down counter, which can be operated in either binary or BCD. Its input and output is configured by the selection of modes stored in the control word register. The programmer can read the contents of any of the three counters without disturbing the actual count in process.

Q-3) **Draw the functional diagram of 8259 and explain the function of each block. [S-15 (Q7-c)], [W-17 (Q6-c)], [S-19 (Q2-g)]**

Ans:- Figure below shows the internal block diagram of the 8259A.It includes eight blocks: data bus buffer, read/write logic, control logic, three registers (IRR, ISR and IMR), priority resolver, and cascade buffer.

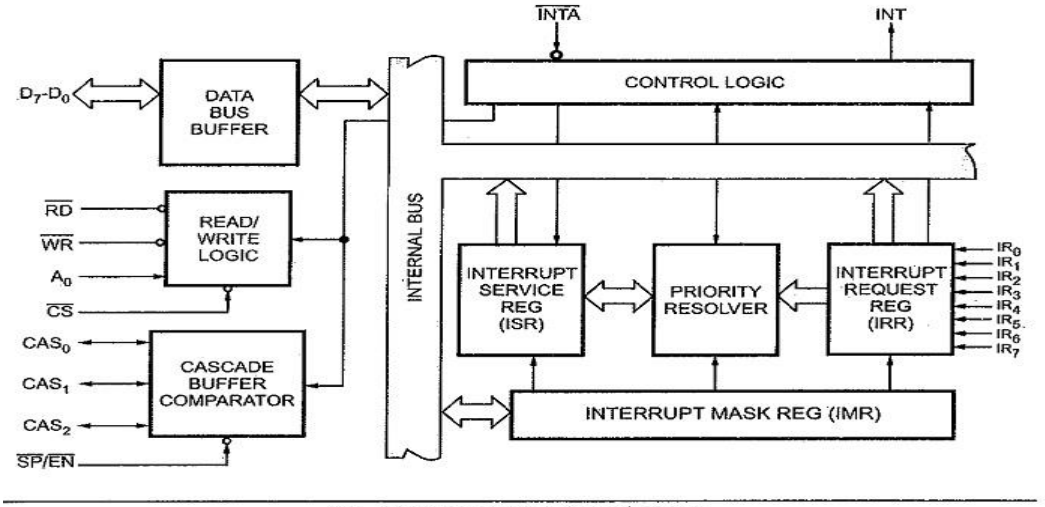


Fig. 14.71 Block diagram of 8259A

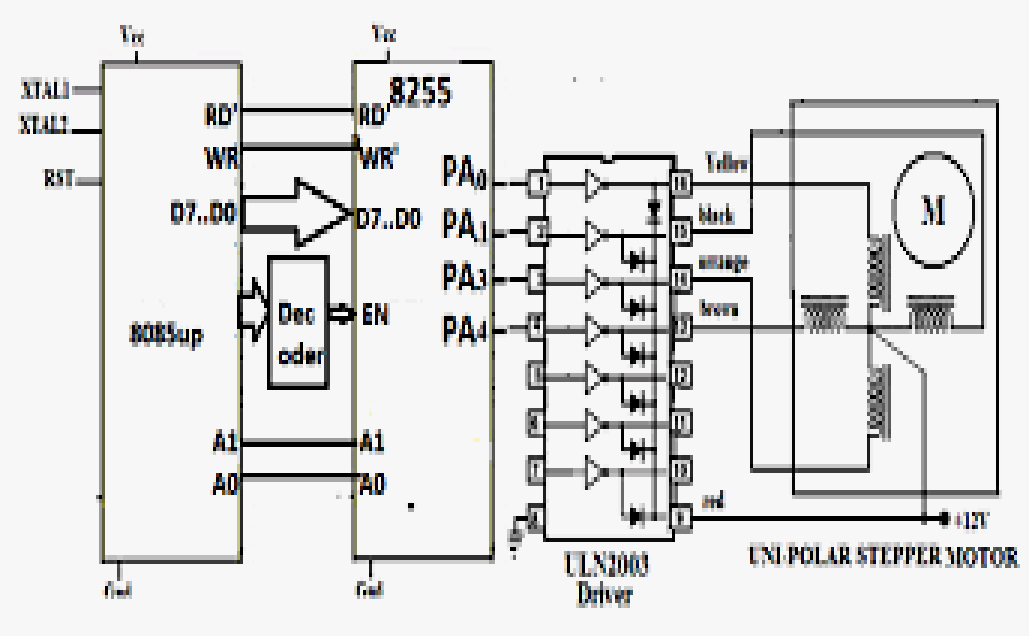
1. **Data Bus Buffer:** The data bus buffer allows the 8085 to send control words to the 8259A and read a status word from the 8259A. The 8•bit data bus buffer also allows the 8259A to send interrupt opcode and address of the interrupt service subroutine to the 8085.
2. **Read/Write Logic:** The RD and WR inputs control the data flow on the data bus when the device is selected by asserting its chip select (CS) input low.
3. **Control Logic:** This block has an input and an output line. If the 8259A is properly enabled, the interrupt request will cause the 8259A to assert its INT output pin high. If this pin is connected to the INTR pin of an 8085 and if the 8085 Interrupt Enable (IE) flag is set, then this high signal will cause the 8085 to respond INTR as explained earlier.
4. **Interrupt Request Register (IRR):** The IRR is used to store all the interrupt levels which are requesting the service. The eight interrupt inputs set corresponding bits of the Interrupt Request Register upon service request.
5. **Interrupt Service Register (ISR):** The Interrupt Service Register (ISR) stores all the levels that are currently being serviced.
6. **Interrupt Mask Register (IMR):** Interrupt Mask Register (IMR) stores the masking bits of the interrupt lines to be masked. This register can be programmed by an Operation Command Word (OCW). An interrupt which is masked by software will not be recognized and serviced even if it sets the corresponding bits in the IRR.
7. **Priority Resolver:** The priority resolver determines the priorities of the bits set in the IRR. The bit corresponding to the highest priority interrupt input is set in the ISR during the MITA input.
8. **Cascade Buffer Comparator:** This section generates control signals necessary for cascade operations. It also generates Buffer-Enable signals. As stated earlier, the 8259 can be cascaded with other 825% in order to expand the interrupt handling capacity to sixty-four levels. In such a case, the former is called a master, and the latter are called slaves. The 8259 can be set up as a master or a slave by the SP/ER pin.
9. **CAS_0 - CAS_2:** For a master 8259, the CAS_0 - CAS_2pins are output pins, and for slave 8259s, these are input pins. When the 8259 is a master (that is, when it accepts interrupt requests from other 8259s), the CALL opcode is generated by the Master in response to the first INTA. The vector address must be released by the slave 8259. The master sends an identification code of three-bits to select one out of the eight possible slave 8259s on the CAS_0 - CAS_2lines. The slave 8259s accept these three signals as inputs (on their CAS_0 - CAS_2 pins) and compare the code sent by the master with the codes assigned to them during initialization. The slave thus selected (which had originally placed an interrupt request to the master 8259) then puts the address of the interrupt service routine during the second and third INTA pulses from the CPU.
10. **SP/ER (Slave Program /Enable Buffer):** The SP/EN signal is tied high for the master. However it is grounded for the slave. In large systems where buffers are used to drive the data bus, the data sent by the 8259 in response to INTA cannot be accessed by the CPU (due to the data bus buffer being disabled). If an 8259 is used in the buffered mode (buffered or non-buffered modes of operation can be specified at the time of initializing the 8259), the SP/ER pin is used as an output which can be used to enable the system data bus

buffer whenever the data bus outputs of 8259 are enabled (i.e. when it is ready to send data). Thus, in non-buffered mode, the SP/EN pin of an 8259 is used to specify whether the 8259 is to operate as a master or as a slave, and in the buffered mode, the SP / EN pin is used as an output to enable the data bus buffer of the system.

Q-4) **What is stepper motor? Write with neat interfacing diagram and program for stepper motor control. [S-15 (Q6-c)], [S-16 (Q3-c)]**

Ans:- A stepper motor rotates in steps in response to digital pulse input. The shaft of the motor rotates in equal increments when a train of input pulses is applied. To control the direction and number of steps appropriate pulses are applied to the stator windings of the motor. There are two most common types of stepper motors- Permanent magnet type and Variable reluctance type.

Interfacing of stepper motor:-



The stepper motor does not rotate smoothly like AC or DC motor. It rotates from one position to the next. The stepper motors are designed to move by 0.9° to 30° in a single step for different type of operation, we can for different stepper motor like low step size and high step size stepper motor.

To rotate the stepper motor in clockwise direction by 360° and then the anticlock-wise direction by 180°. A 12V dc supply is used energies the poles. Pulses sent by the switch on rated voltage (12V dc) to the winding of the desired program. After energising one set of pole windings some delay is provided, then the power supply is switched on to the other set of pole windings. This delay time governs the speed of the motor.

Program:- The program is given to rotate the motor in the clock wise direction through 90°.

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*
2100	3E,80		MVI	A,80
2102	D3,23		OUT	23
2104	21,00,25	START	LXI	H,2500
2107	06, 04		MVI	B,04

2109	7E	REPEAT	MOV	A,M
210A	D3,20		OUT	20
210C	21,02,02		LXI	H,0202
210F	00	DELAY	NOP	
2110	1B		DCX	D
2111	7E		MOV	A,E
2112	B2		ORA	D
2113	C2,0F,21		JNZ	DELAY
2116	23		INX	H
2117	05		DCR	B
2118	C2,09,21		JNZ	REPEAT
211B	C3,04,21		JNZ	START
211E	76		HLT	

```
CODE DB 03H; Sequence for clock wise motor rotation
     DB 06H
     DB 0CH
     DB 09H.
```

Q-5) **Write a program to interface traffic light control system of only one side with desired delay for Red, Yellow and Green light using 8085 instruction sets. [S-17 (Q6-c)]**

Ans:- A traffic light control system is interfaced between 8085 and 8255. All ports of 8255 have been programmed as output ports. The control word to make all port output ports in Mode 0 operation is 80H. The connection of pins of the ports to LED have been made through buffers (7407).Positive logic has been used to switch on LEDs.

Program:- The program is given for Traffic Light Control

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
FC00	3E,80		MVI	A, 80H	Get control word for 8255
FC02	D3, 0B		OUT	0B	Initialize ports of 8255.2
FC04	3E,11	LOOP	MVI	A,11	
FC06	D3,0A		OUT	0A	Red is ON
FC08	D3,04		OUT	0A	
FC0A	CD,00,FD		CALL	DELAY 1	
FC0D	3E, 22		MVI	A,22	Yellow is ON
FC0F	D3,0A		OUT	0A	
FC11	CD,13,FD		CALL	DELAY 2	
FC14	3E,44		MVI	A,44	Green is ON
FC16	CD,00,FD		CALL	DELAY 1	
FC19	3E,22		MVI	A,22	Yellow is ON
FC1B	CD,13,FD		CALL	DELAY 2	
FC1E	C3,04,FC		JMP	LOOP	

DELAY-1

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*

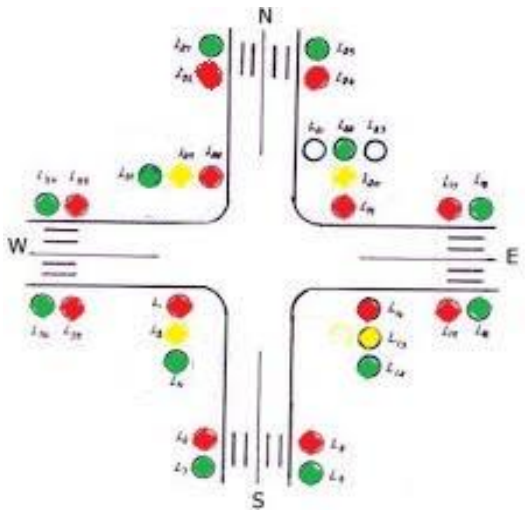
FD00	06,20		MVI	B,20H
FD02	0E,FF	G03	MVI	C,FF
FD04	16,FF	G02	MVI	D,FF
FD06	15	G01	DCR	D
FD07	C2,06,FD		JNZ	G01
FD0A	0D		DCR	C
FD0B	C2,04,FD		JNZ	G02
FD0E	05		DCR	B
FD0F	C2,02,FD		JNZ	G03
FD12	C9		RET	

DELAY-2

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*
FD13	06,10		MVI	B,10H
FD15	C3,02,FD		JMP	FD02

Q-6) **Design a traffic light controller program with a neat block diagram.**
[W-17 (Q4-c)], [W-18 (Q3-c)], [S-19 (Q4)]

Ans:- A traffic light control system is interfaced between 8255 and 8085. All ports of 8255 have been programmed as output ports. The connection of pins of the ports to LED have been made through buffers (7407). The control word to make o/p pins in Mode 0 operation is 80H. Positive logic has been used to switch on LEDs.



Program:- The program is given for Traffic Light Control

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*	*Comments*
FC00	3E,80		MVI	A, 80H	Get control word for 8255
FC02	D3,0B		OUT	0B	Initialize ports of 8255.2
FC04	3E,01	LOOP	MVI	A,01	
FC06	D3,09		OUT	09	Red is ON for South

FC08	D3,08		OUT	08	Red is ON for North
FC0A	3E,44		MVI	A,44	Green ON for East & West
FC0C	D3,04		OUT	0A	
FC0E	CD,00,FD		CALL	DELAY 1	
FC11	3E,22		MVI	A,22	Yellow ON for East & West
FC13	D3,0A		OUT	0A	
FC15	3E,02		MVI	A,02	
FC17	D3,09		OUT	09	Yellow ON for South
FC19	D3,08		OUT	08	Yellow ON for North
FC1B	CD,13,FD		CALL	DELAY 2	
FC1E	3E,11		MVI	A,11	
FC20	D3,0A		OUT	0A	Red is ON fro East & West
FC22	3E,04		MVI	A,04	
FC24	D3,08		OUT	08	Green is ON for North
FC26	D3,09		OUT	09	Green is ON for South
FC28	CD,00,FD		CALL	DELAY 1	
FC2B	3E,22		MVI	A,22	Yellow ON for East & West
FC2D	D3,0A		OUT	0A	
FC2F	3E,02		MVI	A,02	
FC31	D3,09		OUT	09	Yellow ON for South
FC33	D3,08		OUT	08	Yellow ON for North
FC35	CD,13,FD		CALL	DELAY 2	
FC38	C3,04,FC		JMP	LOOP	

DELAY-1

Memory address	*Machine code*	*Labels*	*Mnemonics*	*Operands*
FD00	06,20		MVI	B,20H
FD02	0E,FF	G03	MVI	C,FF
FD04	16,FF	G02	MVI	D,FF
FD06	15	G01	DCR	D
FD07	C2,06,FD		JNZ	G01
FD0A	0D		DCR	C
FD0B	C2,04,FD		JNZ	G02
FD0E	05		DCR	B
FD0F	C2,02,FD		JNZ	G03
FD12	C9		RET	

DELAY-2

FD13	06,10		MVI	B,10H
FD15	C3,02,FD		JMP	FD02

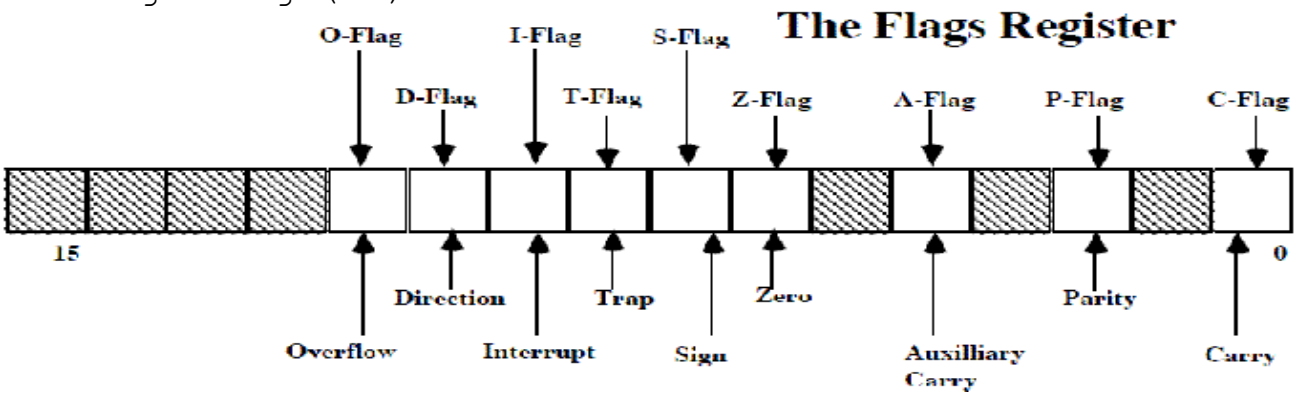
Chapter-VII

Short type questions [2 marks each]

Q-1) **Name different types of flag register available in 8086µP. [ ]**

Ans:-The different types of flag register of 8086 microprocessor are –

1. Carry Flag (CF)
2. Parity Flag (PF)
3. Auxiliary Flag (AF)
4. Zero Flag (ZF)
5. Sign Flag (SF)
6. Trap Flag (TF)
7. Interrupt Enable Flag (IF)
8. Direction Flag (DF)
9. Over flow Flag (OF)



Medium type questions [5 marks each]

Q-1) **Explain different addressing modes of 8086 µP with example. [S-17 (Q7-b)], [S-19 (Q1-e)]**

Ans:- The different ways in which a source operand is denoted in instruction is known as addressing modes. There are 8 different addressing modes in 8086 programming-

Immediate addressing– The addressing mode in which the data operand is a part of the instruction itself is known as immediate addressing mode. Ex- MOV CX,4929H; ADD AX,2387H; MOV AL,FFH

Register addressing– It means that the register is the source of an operand for an instruction. Ex- MOV CX,AX; ADD BX,AX

Direct addressing– The addressing mode in which the effective address of the memory location is written directly in the instruction.
Ex- MOV AX,[1592H]; MOV AL,[0300H]

Register indirect addressing– This addressing mode allows data to be addressed at any memory location through an offset address held in any of the following registers: BP,BX,DI& SI. EX- MOV AX,[BX]; ADD CX,[BX]

Based addressing– In this mode of addressing, the offset address of the operand is given by the sum of contents of the BX/BP registers and 8-bit/16-bit displacement. EX-MOV DX,[BX+04]; ADD CL,[BX+08]

Indexed addressing– In this mode of addressing, the operands offset address is found by adding the contents of SI or DI register and 8-bit/ 16-bit displacements. Ex- MOV BX,[SI+16]; ADD AL,[DI+16]

Based-Indexed addressing– In this mode of addressing, the offset address of the operand is computed by summing the base register to the contents of an index register. Ex- ADD CX,[AX+SI]; MOV AX,[AX+DI]

Based indexed with displacement addressing– In this mode of addressing, the operand's offset is computed by adding the base register contents. An Index registers contents and 8/16-bit displacement.
Ex- MOV AX,[BX+DI+08]; ADD CX,[BX+SI+16]

Q-2) **Draw and explain different bits of flag registers of 8086 µP. [W-17 (Q6-b)], [W-18 (Q6-b)]**
Ans:- The different

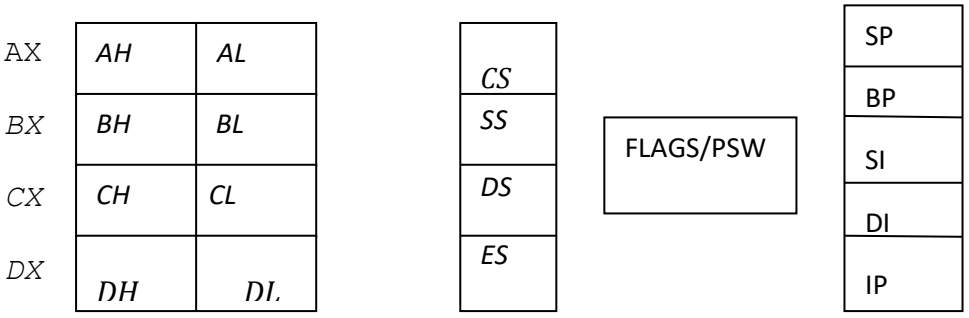
Long type questions [7 marks each]

Q-1) **Give the schematic diagram of various register organisation of 8086 µP and explain the function briefly. [S-16 (Q6-c)]**

Ans:-**Register Organization of 8086:-**

All the registers of 8086 are 16-bit registers. The general purpose registers can be used as either 8-bit registers or 16-bit registers.

The register set of 8086 can be categorized into 4 different groups. The register organization of 8086 is shown in the figure.



General data registers Segment Registers Pointers and index Registers

General Data Registers *:*

The registers *AX, BX,CX*and *DX* are the general purpose 16-bit registers.

- *AX* is used as 16-bit accumulator. The lower 8-bit is designated as *AL* and higher 8-bit is designated as *AH*. *AL* can be used as an 8-bit accumulator for 8-bit operation.All data register can be used as either 16 bit or 8 bit.
- *BX* is a 16 bit register, but *BL* indicates the lower 8-bit of *BX* and *BH* indicates the higher 8-bit of *BX*.
- The register *CX* is used default counter in case of string and loop instructions.The register *BX* is used as offset storage for forming physical address in case of certain addressing modes.
- *DX* register is a general purpose register which may be used as an implicit operand or destination in case of a few instructions.

Segment Registers**:**
The 8086 architecture uses the concept of segmented memory. 8086 able to address to address a memory capacity of 1 megabyte and it is byte organized. This 1 megabyte memory is divided into 16 logical segments. Each segment contains 64 Kbytes of memory.

There are four segment register in 8086

- Code segment register (CS)
- Data segment register (DS)
- Extra segment register (ES)
- Stack segment register (SS)

Code segment register (CS): is used from addressing memory location in the code segment of the memory, where the executable program is stored.

Data segment register (DS): points to the data segment of the memory where the data is stored.

Extra Segment Register (ES) : also refers to a segment in the memory which is another data segment in the memory.

Stack Segment Register (SS): is used from addressing stack segment of the memory. The stack segment is that segment of memory which is used to store stack data.

While addressing any location in the memory bank, the physical address is calculated from two parts:-

- The first is segment address, the segment registers contain 16-bit segment base addresses, related to different segment.
- The second part is the offset value in that segment.

The advantage of this scheme is that in place of maintaining a 20-bit register for a physical address, the processor just maintains two 16-bit registers which is within the memory capacity of the machine.

***Pointers and Index Registers*.**

The pointers contain offset within the particular segments.
The pointer register *IP* contains offset within the code segment.
The pointer register *BP* contains offset within the data segment.
He pointer register *SP* contains offset within the stack segment.

The index registers are used as general purpose registers as well as for offset storage in case of indexed, base indexed and relative base indexed addressing modes.

The register *SI* is used to store the offset of source data in data segment.

The register *DI* is used to store the offset of destination in data or extra segment.

The index registers are particularly useful for string manipulation.

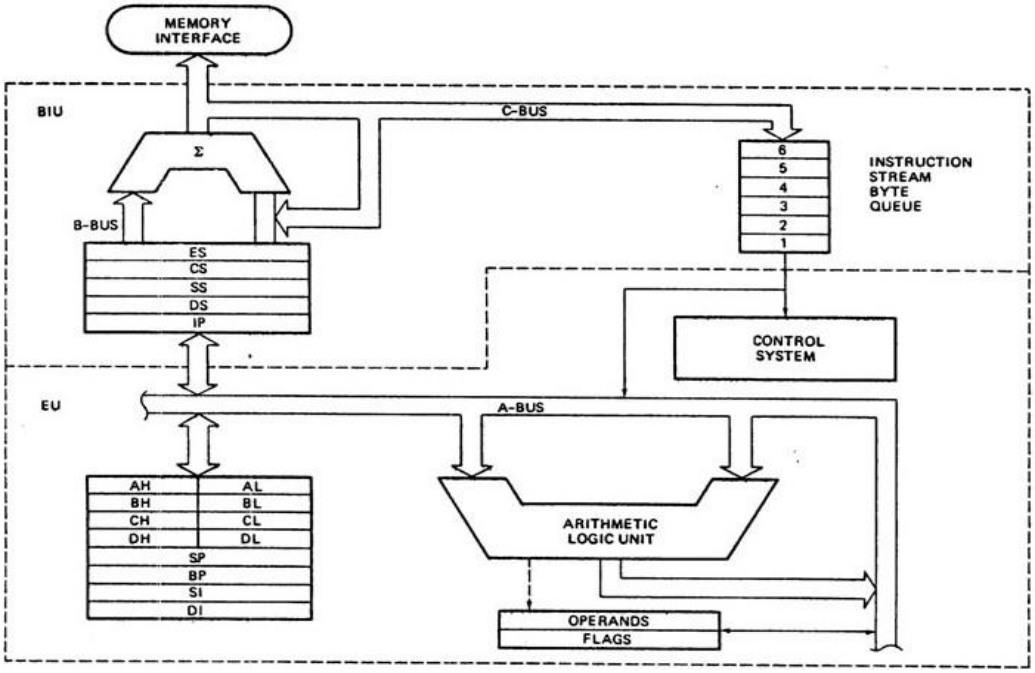
Flag Register:

The 8086 flag register contents indicate the results of computation in the *ALU*. It also contains some flag bits to control the *CPU* operations.

Q-2) **Draw the functional block diagram of internal architecture of 8086 µP and explain the function of each block briefly. [S-15 (Q2-c)], [W-17 (Q5-c)], [W-18 (Q7-c)], [S-19 (Q7)]**

Ans:-The 8086 CPU is divided into two independent functional units:

- Bus Interface Unit (BIU)
- Execution Unit (EU)



Bus Interface Unit (BIU)

The function of BIU is to:

- Fetch the instruction or data from memory.
- Write the data to memory.
- Write the data to the port.
- Read data from the port.

Instruction Queue

To increase the execution speed, BIU fetches as many as six instruction bytes ahead to time from memory.
All six bytes are then held in first in first out 6 byte register called instruction queue.
Then all bytes have to be given to EU one by one.
This pre fetching operation of BIU may be in parallel with execution operation of EU, which improves the speed execution of the instruction.

Execution Unit (EU)

The functions of execution unit are:

- To tell BIU where to fetch the instructions or data from.
- To decode the instructions.
- To execute the instructions.

The EU contains the control circuitry to perform various internal operations. A decoder in EU decodes the instruction fetched memory to generate different internal or external control signals required to perform the operation. EU has 16-bit ALU, which can perform arithmetic and logical operations on 8-bit as well as 16-bit.

General Purpose Registers of 8086:-

These registers can be used as 8-bit registers individually or can be used as 16-bit in pair to have AX, BX, CX, and DX.

AX Register: AX register is also known as accumulator register that stores operands for arithmetic operation like divided, rotate.
BX Register: This register is mainly used as a base register. It holds the starting base location of a memory region within a data segment.
CX Register: It is defined as a counter. It is primarily used in loop instruction to store loop counter.
DX Register: DX register is used to contain I/O port address for I/O instruction.

Segment Registers:-
Additional registers called segment registers generate memory address when combined with other in the microprocessor. In 8086 microprocessor, memory is divided into 4 segments as follow:
Code Segment (CS): The CS register is used for addressing a memory location in the Code Segment of the memory, where the executable program is stored.
Data Segment (DS): The DS contains most data used by program. Data are accessed in the Data Segment by an offset address or the content of other register that holds the offset address.
Stack Segment (SS): SS defined the area of memory used for the stack.
Extra Segment (ES): ES is additional data segment that is used by some of the string to hold the destination data.

Flag Registers of 8086:-
Flags Register determines the current state of the processor. They are modified automatically by CPU after mathematical operations, this allows to determine the type of the result, and to determine conditions to transfer control to other parts of the program. 8086 has 9 flags and they are divided into two categories: Conditional Flags&Control Flags

Conditional Flags
Conditional flags represent result of last arithmetic or logical instruction executed. Conditional flags are as follows:

Carry Flag (CF): This flag indicates an overflow condition for unsigned integer arithmetic. It is also used in multiple-precision arithmetic.

Auxiliary Flag (AF): If an operation performed in ALU generates a carry/barrow from lower nibble (i.e. D0 – D3) to upper nibble (i.e. D4 – D7), the AF flag is set i.e. carry given by D3 bit to D4 is AF flag. This is not a general-purpose flag, it is used internally by the processor to perform Binary to BCD conversion.

Parity Flag (PF): This flag is used to indicate the parity of result. If lower order 8-bits of the result contains even number of 1's, the Parity Flag is set and for odd number of 1's, the Parity Flag is reset.

Zero Flag (ZF): It is set; if the result of arithmetic or logical operation is zero else it is reset.

Sign Flag (SF): In sign magnitude format the sign of number is indicated by MSB bit. If the result of operation is negative, sign flag is set.

Overflow Flag (OF): It occurs when signed numbers are added or subtracted. An OF indicates that the result has exceeded the capacity of machine.

Control Flags

Control flags are set or reset deliberately to control the operations of the execution unit. Control flags are as follows:

Trap Flag (TP):

It is used for single step control.
It allows user to execute one instruction of a program at a time for debugging.
When trap flag is set, program can be run in single step mode.

Interrupt Flag (IF):

It is an interrupt enable/disable flag.
If it is set, the maskable interrupt of 8086 is enabled and if it is reset, the interrupt is disabled.
It can be set by executing instruction sit and can be cleared by executing CLI instruction.

Direction Flag (DF):

It is used in string operation.
If it is set, string bytes are accessed from higher memory address to lower memory address.
When it is reset, the string bytes are accessed from lower memory address to higher memory address.

Q-3) **Write an ALP to evaluate x (y + z), where x = 10H, y = 20H and z = 30H using 8086 instruction sets. [S-17 (Q7-c)]**

Ans:-**Program**:-

```
START: MOV AX, 5000H ; Initialising DS with value
       MOV DS, AX    ; 5000H
       MOV AL,20H    ; Get value of Y into Acc.
       MOV CL,30H    ; Get value of z in Reg. C
       ADD AL,CL     ; Add the content of Reg.C with the content of Acc.

       MOV CL,10H    ; Get value of x in Reg. C
       MUL CL        ; Multiply the content of Reg. C with the content
       of accumulator.
       MOV SI,4000H  ; Source index address 4000H is moved to SI
       MOV [SI],AX   ; Move the result from accumulator to SI
       HLT           ; Stop the program
```