

[2065]

Section A

- Q) Why addressing modes are required in microprocessor? Discuss different types of addressing modes with suitable example.

The different ways in which a processor can access data are referred to as its addressing modes. In assembly language statements, the addressing mode is indicated in the instruction itself. In each statement programmer has to specify 3 things

- i) operation to be performed;
- ii) Address of source data,
- iii) Address of destination of result.

Addressing mode are simply required in microprocessor for the ease of programming leading towards efficient and fast processing as it directly effect the timing and steps to access an operand from register or memory.

The different types of addressing modes are :

- i) Register Addressing mode
- ii) Immediate Addressing mode
- iii) Direct Addressing mode
- iv) Register Indirect Addressing mode
- v) Implied Addressing mode

X.
(P.T.O)

i) Register Addressing mode:

- It is most common form of data addressing
- Transfers a copy of a byte/word from source register to destination register

instruction	source	Destination
-------------	--------	-------------

MOV A, B register B register A

- It is carried out with 8 bit registers A, B, C, D, E, H & L
- It is important to use register of same size.
- 8 bit registers should not be mix with 16-bit register

Example:

MOV A, B ; Copy B into A.

ii) Immediate Addressing mode:-

- It implies that the data immediately follows the hexa-decimal opcode in the memory.
- It transfers source immediate byte/word of data in destination register or memory location.

Instruction	Source	Destination
-------------	--------	-------------

MVI C, 3AH DATA 3AH REGISTER C

Example:

MVI A, 90 ; Copy 90 into A

MVI BX1 H, 1234H ; Copy 1234H into H & L pair

iii) Direct Addressing mode:

- Here, the address of the data is defined in the instruction itself.

Instruction	Source	Destination
-------------	--------	-------------

CDA 2000H memory location 2000H REGISTER A

Example:

LHLD 1000H ; Copies the content of 1000H address memory to L and 1000H memory to H

LDA 2000H ; Copies the content of 2000H memory to Accumulator.

JMP 4000H ; CALL 5000H

4) Register indirect Addressing mode

- It allows data to be addressed at any memory location through an address held in any of the H pair, B pair and D pair register.
- It transfer byte/word between a register and a memory location addressed.

Instruction	Source	Destination
MOV C, M	ASSUME HL = 1000H & M is Content of 1000H add.	REGISTER C

Example:

MOV C, M ; Copies the word content of memory location address by HL pair to register C.

STAX B ; Copy A into the memory location addressed by B pair.

5) Implied Addressing Mode:

- The addressing mode of certain instructions is implied by the instruction's function

Instruction	Destination
STC	Carry flag

Example: CMC : Complement carry flag

STC : Set carry flag & DAA : Decimal Adjust accum
ulator

3) Write a program in 8-bit Microprocessor to multiply two 16 bit numbers and store in the memory location starting from 3500H. Save the carry bits in the location starting from 3600H.

LHLD 3500H ; loads the content of memory address 3500H to HL register pair
XCHG ; Exchange content of HL with BC register pair
LDA 3502H ; load content of 3502 to accumulator
LXI H, 0000H ; Set HL pair to 0000H
MUL C, 08 ; Transfers 08H to register p C
LOOP1 DADH ; Add register to HL pair
RAL ; Rotate accumulator left
JNC LOOP2 ; If no carry, jumps to LOOP2
+
DAD D ; Add register pair to HL register pair

LOOP2 DCR B ; Decrement to B register
JNZ LOOP1 ; If no zero, jump to LOOP1 to

SHLD 3500H ; Stores the content of HL register pair to 3500H

HLT

Section 'B'

1. Write an assembly language program to subtract two 16-bit numbers.

```
LHLD 40D1H ; Get 1st 16-bit number in HL pair
XCHG ; Exchange data with DL pair
LHLD 40D2H ; Get 2nd 16-bit number in HL pair
MOV A, E ; 
SUB L ; Subtract lower byte of 2nd number with accum
MOV L, A
MOV A, D
SBB H ; Subtract higher
MOV H, A
SHLD 4804H ; Stores 16-bit result in memory location
              4804H
HLT
```

Differentiate between PUSH and POP operations. Write a program to illustrate the use of PUSH operation.

PUSH operation:

- It is used to copy the contents of the register pair designated in the operand onto the stack.
- The stack pointer register is decremented and the contents of high-order register (B, D, H, A) are copied into that location.
- Stack pointer is again decremented and contents of the low-order registers (C, E, L, flags) are copied to that

location

example: PUSH B , PUSH A

POP operation:

- The contents of the memory location pointed out by the stack pointer register are copied to low-order registers (C, E, L, status flag) of operand.
- Stack pointer is incremented by 1 and the contents of that memory location are copied to high-order register (B, D, H, P) of the operand.
- Stack pointer register is again incremented by 1.

Example: POP H , POP A

Program

Program to print sum of two numbers

• MODEL SMALL

• STACK

• DATA

VAL1 DB 22

VAL2 DB 33

• CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV AX, 0

MOV AL, VAL1

RAM

ADD AX, 3030H

```

PUSH AX
POP AX
MOV DL, AH
MOV DH, AL
MOV DH, 02H
INT 21H

MOV DL, GH
MOV AH, 02H
INT 21H
MOV AX, 4C00H
INT 21H
MAIN ENDP
END MAIN

```

Write an assembly

- 7) What are the function of I/O interface? Explain it with example.

The function of I/O interface is the communication between the internal system & hardware device or I/O device.

The method of communication are explained below:
Parallel communication:

When a word of n bits is to transmitted in parallel each bit is transmitted on a separate line the start along with a common ground line with respect to which the status of each line is measured.

Parallel data transmission is impractical over long distance because of prohibitive cost of installing a large number of lines.

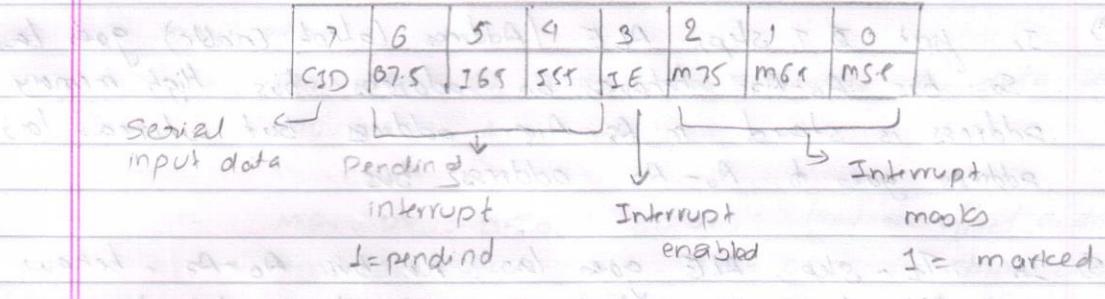
Serial communication:

In serial data transmission, each bit of the word is sent in succession, one at a time over a single pair of wires. A parallel to serial converter is used to convert the incoming parallel data to serial form and then the data is sent out with least significant bit. So first & most significant bit D7 coming last of all. If the bit rate is retained after the parallel to serial data transmission will be n times more than the time taken in parallel data transmission.

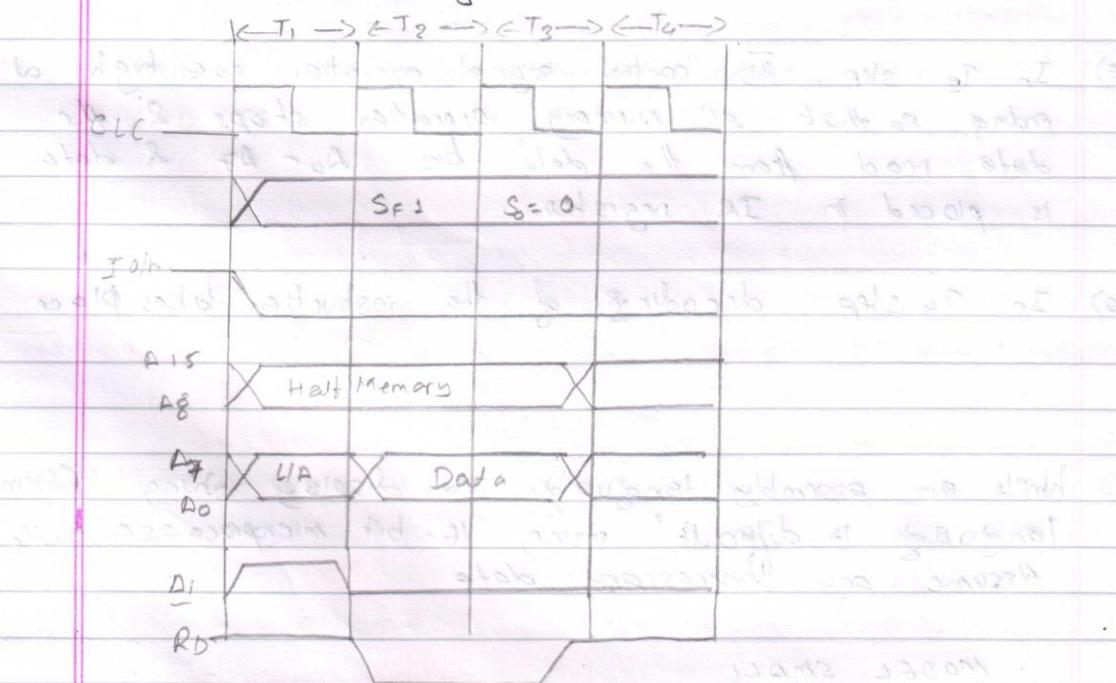
Why is interrupt required? Draw the block diagram of interrupt handlers & explain it.

An interrupt is a signal that a peripheral device sends to the central processor in order to request attention. In response to an interrupt, the processor stops what it is currently doing & execute a service routine. The response to microprocessor is directly controlled by the microprocessor.

(P.T.O)



Draw the timing diagram for ADD C & explain it.



The timing diagram of ADD C is explained below

i) Since, it is memory operation

$$I_0/M = 0 \quad \& \quad S_0 = 0 \quad \therefore S_1 = 1$$

- 2) In first T_1 step, ALE (Address Latch Enable) goes low so, bus $A_0 - A_7$ behaves as address bus. High memory address is stored in $A_8 - A_{15}$ addresses but whereas low address goes to $A_0 - A_7$ address bus.
- 3) In T_2 -step, ALE goes low, no bus $A_0 - A_7$, behaves as data bus. Now content C is stored in data bus.
- 4) In T_2 -step to T_3 step \bar{RP} control signal goes low show that the data is read from the data bus $A_0 - A_7$.
- 5) In T_3 step, \bar{RD} control signal operation goes high at ending so that all memory operation stops & the data read from the data bus $A_0 - A_7$ & data is placed in IR register.
- 6) In T_4 step, decoding of the instruction takes place

Write an assembly language to display string "Assembly language is difficult" using 16-bit microprocessor code
Assume any necessary data

- MODEL SMALL
- STACK
- DATA
- CODE

STRING DB 'Assembly language is difficult \$'

MAIN PROC

MOV AX, @ DATA Initialize the data segment
MOV DS, AX

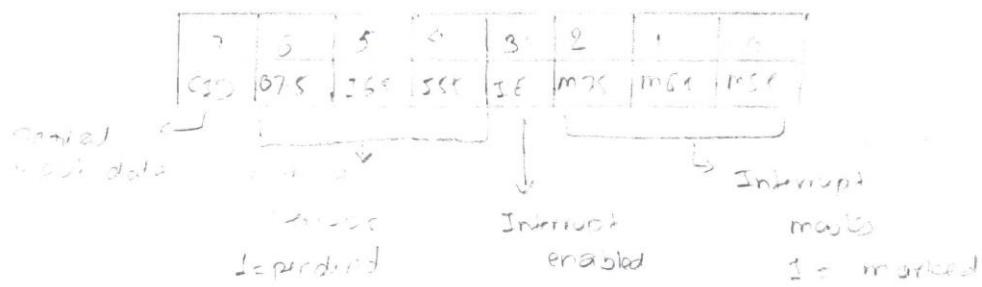
MOV DX, OFFSET STRING ; Load the offset address into DX
MOV AH, 09H ; AH = 09H for string display until
INT 21H ; DOS interrupt function

MOV AX, 4C00H ; End Request with AH = 4CH
or AX = 4C00H

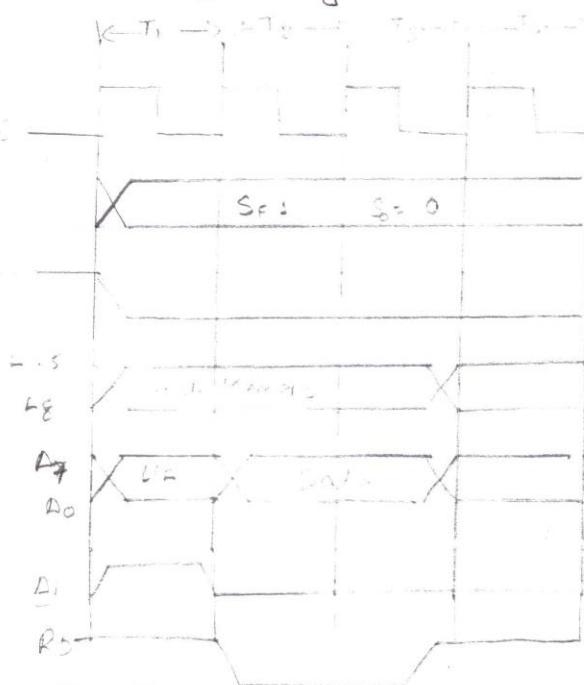
INT 21H

MAIN ENDP ; End Procedure

END MAIN ; End Program



Draw the timing diagram for ADD C & explain it.



- The timing diagram of ADD C is explained below
- Since, it is memory opera~~ta~~
 $I_0/M = 0$ & $S_0 = 0 \rightarrow S_1 = 1$

2066

* Section 'A'

Q 1. [2070 : 01]

Q 2 f

2. Explain the application of flags in microprocessor. Discuss different type of flags with suitable.

The

Flags represents the status of the arithmetic and logical operation. flip-flop concepts are used to represent the status of arithmetic and logical operation in flags. It is 8-bit registers that consists of many types.

i) Sign flag

It is used for indicating the sign of the data in the accumulator. The sign flag is set if negative (1 if negative) & sign flag is reset if positive (0 if positive)

ii) zero flag :

It is set if result obtained after an operation is 0. It is set following an increment or decrement operation of that register.

iii) Parity flag

It is set if parity is even. It is cleared if the parity is odd.

Write a program in 8-bit Microprocessor to store 60H, BAH, 7CH and 10H in the memory location starting from 2000H, add those data and store the result in 3000H and carry flag in 5001H. Explain all the steps.

MVI A, 60H ; Moves 60H to accumulator
STA 2000H ; Stores content of accumulator to address 2000H
MVI A, BAH ; Move BAH to accumulator
STA 2001H ; Stores content of accumulator to address 2001H
MVI A, 7CH ; Move 7CH to accumulator
STA 2002H ; Stores content of accumulator to address 2002H
MVI A, 10H ; Moves 10H to accumulator.
STA 2003H ; Stores content of accumulator to address 2003H
MVI B, 04H ; Move 04H to register B
LXI H, 2004H ; Load 2004H to HL pair register & content to memory (M) register

MOV A, M ; Copies content of memory to accumulator

L1: LXI H ; Increment to the HL pair by 1.

ADD M ; Adds the content of register memory to that of accumulator & stores result in accumulator.

JNC L2 ; If there is no carry then jumps to skip L2.

INR E ; Increment given to E register

L2 ; DCR B ; Decrement to B register

JNZ L1 ; Jump to L1 until B becomes zero.

STA 300H ; stores the result in the address 300H

MOV A, E ; Copies the content of E register to accumulator

STA 5001H ; stores the content of accumulator to address 5001H

HLT ; Stops the execution of program

Section 'B'

Explain about fetch operation and timing diagram.

The process of opcode fetch operation requires minimum 4-clock cycles T_1 , T_2 , T_3 and T_4 and is 1st machine cycle (M_1) of every instruction.

Example :

Fetch a byte 41H stored at memory location 2105H.

The first byte of an instruction is its opcode. The program counter keeps the memory address of the next instruction to be executed in the beginning of fetch cycle. The process of opcode fetch operation requires minimum 4-clock cycles T_1 , T_2 , T_3 & T_4 and is the 1st machine cycle (M_1) of every instruction. The fetch part of the instruction is the same for every instruction. The control unit puts the contents of the program counter (PC) on the address bus. In the execute cycle of the instruction, the control unit examines the opcode and as per interpretation further memory read or write operations are performed depending upon whether additional information/data are required or not. At the same time, the control unit sends the contents of the accumulator to the ALU and performs the additional operations. The results of addition operation is passed to the accumulator over writing the previous contents. On the completion of one instruction, the PC is automated to point next memory

location to execute the subsequent instruction.

The necessary steps which are called carried in a machine cycle can be represented graphically. Such graphical representation is called timing diagram.

Q.No. 7. [2009 Q. 7]

8. What do you mean by interrupt? Explain in detail about software interrupt?

An interrupt is a signal that a peripheral board sends to the central processor in order to request attention. In response to an interrupt the processor stops what it is currently doing and executes a service routine. When the execution of the service routine is terminated, the original process may resume its previous operation. The interrupt is initiated by an external device and is asynchronous, meaning that it can be initiated at any time without reference to system clock. These interrupts are primarily issued on: initiation of I/O operation, completion of I/O operation and occurrence of hardware or software errors.

Software Interrupt:

External and internal interrupt are initiated from signal that occur in the hardware of the CPU. A software interrupt is initiated by executing an instruction. Software interrupt is a special call instruction that behaves like an interrupt rather than a sub-routine call. It can be used by the

programmer to initiate an interrupt procedure at any desired point in the program. The most common use of software interrupt is associated with a supervisor call instruction. This instruction provides means for switching from a CPU user mode to the supervisor mode. In 8085, the instruction like $\text{RST } 0, \text{ RST } 1, \text{ RST } 2, \text{ RST } 3, \dots$ etc causes a software interrupt.

(ii) ; Program to print string

• MODEL SMALL

• STACK

• DATA

STRING DB 'I want to know more about up\$'

• CODE

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV DX, OFFSET STRING

MOV AH, 09H

INT 21H

MOV AX, 4C00H

MAIN ENDP

END MAIN

c) Differentiate between data and address bus with suitable example

Address Bus:

- Address bus is a part of the computer system that is dedicated for specifying an address.
- Address bus is unidirectional
- The width of the address bus is specified by the processor that is addressed by the system
- The address bus consists of 16, 20, 14 or 32 parallel lines

Data Bus:

- Data bus simply carries data
- Data bus is bi-directional
- The width of the data bus is determined by the size of the individual memory block
- The data bus consists of 16, 8, or 32 parallel lines.

Q. 8 [2065 Q. 8]

Input Port: It includes two input ports numbered as 1 & 2. It also consists of hexadecimally keyboard encoder connected to port 1, sends READY signal to bit 0 & of port 2.

Program Counter: It has 16 bits, therefore it can count from $PC = 0000\ 0000\ 0000\ 00$ to $PC = 1111\ 1111\ 1111\ 1111$. Its job is to send to the memory address of the next instructions to be fetched and executed. A low resets the PC before each computer run so that the data processing starts with the instruction stored in memory location 0004.

AMAR & Memory: During the fetch cycle, the MAR receives 16-bit addresses from PC. The two-state MAR output then addresses memory location. The memory has 8k ROM containing a program called monitor that initializes the computer on power up, interrupt the keyboard inputs & so on. The rest of memory is 64k RAM with addresses from 0000H to FFFFH.

MDR (Memory Data Register): It is 8-bit buffer register. Its output sets up the RAM. It receives data from the bus before a write operation and it sinks data to the bus after a read operation.

IR (Instruction register): It is part of control unit. Many read operation performed by computer to fetch an instruction from memory, this places the contents of the addressed memory location on W-bus. Its contents

split into 2 nibbles. SPP 2 uses 8-bits for opcode which can accommodate 256 instruction.

Control Sequence : It produces the control words or micro-instructions that co-ordinate and direct the rest of the computer. It has more the hardware since SPP 2 has bigger instruction set. Microinstruction determines how the registers react to the next +ve clock edge.

Accumulator : A buffer register that stores intermediate answers during the computer run. It has two outputs two-stack & 3-state. Two stack output goes to ALU & 3-state to W-bus. Hence, the 8-bit word in the accumulator continuously drives the ALU, but this same word appears on the bus only when En is active.

ALU & Flags : It includes arithmetic and logic operations. It has 4 or more control bits that determine the arithmetic or logic operation performed on words A & B. Flag is a flip-flop that keeps track of changing condition during a computer. SPP 2 has 2 flags : sign and zero flags.

TMP .B & C registers : TMP or (Temporary register) is used instead of register B to hold data being added or subtracted which allows us more freedom.

in using the B register. Beside B & TMP, SAP 2 has a register C which gives us more flexibility in moving data during the computer run.

Output Ports : It consists two outputs numbered as 3²⁴. Its contents of the accumulator can be loaded into port 3, which drives a hexadecimal display. The contents can also be seen through port 0.

Comparison between SAP 1 and SAP 2 are as follows :

- i) Bidirectional register
- ii) flags
- iii) Large instruction set having jump, call & loop
- iv) Standard input output devices in SAP 2
- v) High memory capacity (64k) in SAP 2
- vi) Address bus 16 bit in SAP 2
- vii) Separate MAR & MDR in SAP 2.
- viii) Complex programming possible in SAP 2
- ix) Introduction of ALU in SAP 2
- x) Serial as well as parallel Z/O in SAP 2.

2)

2065 - (Selhan D - Q No 3)

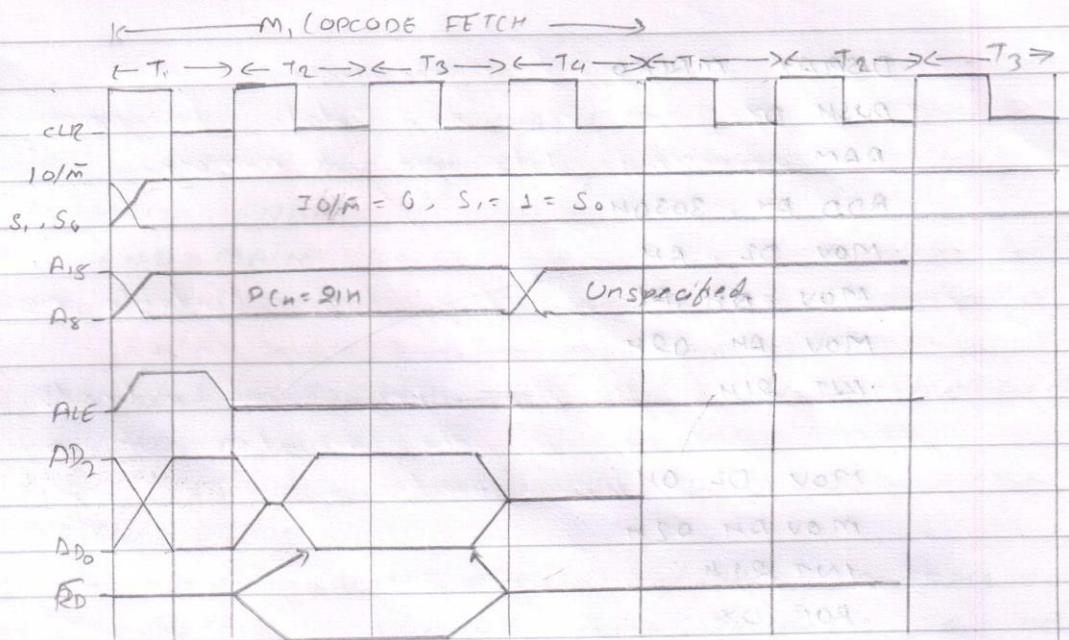
Explain

Section B

Explain execute operation & timing diagram with suitable example:

The opcode fetched from the memory goes to IR. From the IR it goes to decoder with decode restriction which decodes instructions. The instruction is decoded execution begins. If the operand is in general purpose register, execution begins. If the operand is in general purpose register, execution is performed, immediately i.e. in one clock cycle. If an instruction contains data or operand address data. In some instruction write operation is performed. In write cycle data are sent from the CPU to the memory or an O/P device. In some cases execute cycle may involve one or more read or write cycle or both.

The necessary steps which are carried in a machine cycle can be represented graphically. Such graphical representation is called timing diagram.



Q8: Timing Diagram of opcode fetch

Write an assembly language program to add two 16-bit numbers (3467H & ALC0H)

```

· MODEL SMALL
· STACK
· DATA
    VAL1 DW 3467H
    VAL2 DW 0099H
    MSG DB 'The sum is $'
· CODE

```

DISPLAY MACRO

PUSH DX

POP AX

ADD AX, 3030H

MOV DL, AH

MOV DH, DL

MOV AH, 02H

INT 21H

MOV DL, 0H

MOV AH, 02H

INT 21H

POP DX

ENDM

MAIN PROC

MOV AX, @DATA

MOV DS, AX

MOV DX, OFFSET MSG

MOV AH, 09H

INT 21H

MOV AX, 6

MOV AX, VAL 1

MOV AX, VAL 2

MOV DX, 26

MOV CX, 100

DIV CX

DISPLAY

MOV AX, DX

```
    DISPLAY
    MOV AX, 4C00H
    INT 21H
MAIN ENDP
END MAIN
```

- 6 Differentiate between data and address bus with suitable example

Address Bus :

- Address bus is a part of the computer system that is dedicated for specifying an address
- Address bus are unidirectional
- The width of the address bus is specified by the processor that is address by the system
- The address bus consist of 16, 20, 24 or 32 parallel lines
- The number of memory location that the CPU can address is determined by the number of address line

Data Bus

- Data bus carries data
- Data bus are bi-directional
- The width of the data bus is determined by the size of the individual memory block
- The data bus consist of 16, 8 or 32

- The CPU can read data from memory or it can send data out to memory

10) Explain 3 types of flag with suitable example.

Flag is a flip flop which ~~read~~ indicates some condition produced by the execution of an instruction or controls certain operation.

Three types of flag are:

i) Sign flag:

- Used for indicating the sign of the data in accumulator.
- It is set if negative and reset if positive.

ii) Zero flag

- It is set if result obtained after an operation is 0.
- It is set following an increment or decrement operation of that register.

For ex:

$$\begin{array}{r}
 10110011 \\
 + 01001101 \\
 \hline
 100000000
 \end{array}$$

iii) Carry flag

- It is set if there is a carry or borrow from arithmetic operation.

e.g.:

$$\begin{array}{r} 1011 \ 0101 \\ - 1100 \ 1100 \\ \hline 1110 \ 1001 \end{array}$$

Borrow - 1

$$\begin{array}{r} 1011 \ 0101 \\ + 0110 \ 1100 \\ \hline 10010 \ 0001 \end{array}$$

Carry - 1

11) Why do we require serial communication? Explain with suitable example.

In serial communication each bit of the word is sent in succession one at a time over a single pair of wires. A parallel to series converter is used to convert the incoming parallel data to serial form and then the data is sent out with the least significant bit Do first and most significant bit D7 coming last of all if the bit rate is retained after the parallel to serial data transmission will be n times more than the time taken in parallel data transmission. If the word in the above example were to send serially the data on the channel will appear as in fig:

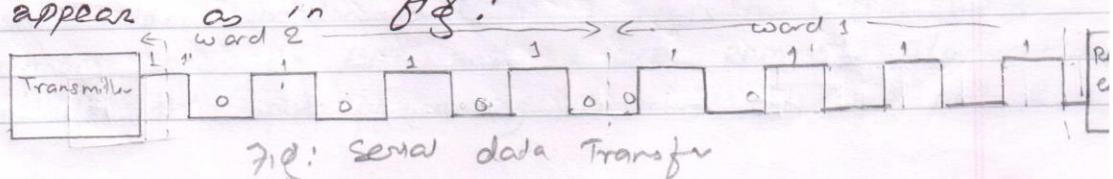


Fig: Serial data Transfer

Write an assembly language program to display string "I like programming in the assembly language". Using 16 bit microprocessor code. Assume any necessary data

```
· MODEL SMALL
· STACK 100H
· DATA
    STRING DB 'I like programming in the assembly language $'
· CODE
MAIN PROC
    MOV AX @DATA
    MOV DS, AX
    MOV DX, OFFSET STRING
    MOV AH, 09H
    INT 21H
    MOV AX, 4C00H
    INT 21H
MAIN ENDP
END MAIN
```

2069

What are the uses of flags in the microprocessor. Discuss different types of flags with examples.

A flag is a flip flop which indicates some condition by the execution of an instruction or control certain operation.

Uses of flags:

- Trap flag is used for single stepping through a program
- Interrupt flag is used to allow or prohibit the interruption of a program
- It is used to get result 16 bit in 8 bit system by using two register of 8 bit.

Eg: Using carry flag we can add two 8 bit nos and get result in 16 bit

$$\begin{array}{r} 1011 \quad 0101 \\ 0110 \quad 1100 \\ \hline \text{Carry} \rightarrow 1 \quad 0010 \quad 0001 \end{array}$$

- It is used for indicating the sign of the data in accumulator

Eg: Sign flag is set if -ve
Sign flag is reset if +ve.

Different types of flags are:

i) CONDITIONAL FLAGS.

a) Carry flag

- It is set if there is a carry or borrow for arithmetic operation

$$\begin{array}{r} 1011 \quad 0101 \\ 1100 \quad 1100 \\ \hline \text{Borrow} \rightarrow 1 \quad 11101001 \end{array}$$

b) Parity flag

- It is set if parity is even and reset if parity is odd

c) 1

c) Auxiliary carry flag

- It is set if there is a carry out of bits 3

d) Zero flag:

- It is set if result obtained after an operation is 0

- It is set following an increment or decrement operation of that register.

Eg:

$$\begin{array}{r} 1011 \quad 0011 \\ 0100 \quad 1101 \\ \hline 10000 \quad 0000 \end{array}$$

e) Sign flag:

- Used for indicating the sign of the data in accumulator

- Sign flag is set if negative & reset if positive

e) Overflow flag

2) CONTROL flag.

a) TF (single step trap flag)

- It is used for single stepping through a program.

b) IF (Interrupt flag)

- It is used to allow or prohibit the interruption of a program.

c) DF (string direction flag)

- Direction flag is used with string instruction.

3) Q No. 3

≈ [2070 Q. 2]

Section 'B'

Write an assembly language program to multiply two numbers :

8DB1H 40

8DB3H 40

LHLD 8DB1H ; loads contents of 8DB1H to HL pair

XCHG ; Exchange the content of 8DB1H to HL to BC.

LDA 8DB3H ; loads the content of HL pair with BC register pair

LDA

LXI H 0000H ; Stores 0000H in HL pair

MVI B 08H ; Transfer 08H to B register.

LOOP: DAD H ; Adds HL pair register with BC pair

RAL ; Rotate accumulator left

JNC LABEL ; If no carry, jump to LABEL tag

DAD D ; Adds HL pair register with D pair

LABEL: DCR B ; Decrement to B register

JNZ LOOP ; If not zero, jumps to loop tag.

SHLD 8DB4H ; Stores the content of HL & register pair
to 8DB4H.

HLT

What are the function of I/O interface? Explain it with example.

The function of I/O interface is the communication between the internal system & hardware devices or I/O devices

The method of communication are:

i) Parallel Communication

ii) Serial Communication

i) Parallel Communication:

In serial When a word of n bits is to

transmitted in parallel each bit is transmitted on a

separate line along with a common ground line w.r.t. which the status of each line is measured.

Parallel data transmission is impractical over long distance because of prohibitive cost of installing a large number of lines

Serial Communication:

In serial data transmission, each bit of the word is sent in succession, one at a time over a single pair of wires. A parallel to serial converter is used to convert the incoming parallel data to serial form and then the data is sent out with least significant bit (LSB) and most significant bit (MSB) coming last of all. If the bit rate is retained after the parallel to serial conversion, the time taken to transmit a word in serial data transmission will be n times more than the time taken in parallel data transmission.

What do you mean by interrupt vector? Explain in detail about hardware interrupt.

External interrupts are initiated via the microprocessor's interrupt pins by external devices such as I/O devices, timing device, circuit monitoring the power supply etc. Cause of these interrupts may be I/O device requesting transfer of data, I/O device finished transfers of data, elapsed time of an event.

or power failure. Timeout interrupt may result from a program that is in an endless loop & thus exceeded its time allocation. They are also called hardware interrupts.

Hardware interrupt can be further divided into 2 types

i) Maskable interrupt :

A maskable interrupt is one which can be enabled or disabled by execution instruction such as EI (enable interrupt) & DI (Disable interrupt). No maskable interrupt are recognized by the processor when the interrupt is disabled.

ii) Non - maskable Interrupt

This type of interrupt cannot be enabled or disabled by instructions. This type has higher priority over the maskable interrupt.

ii) Write assembly language program to display a string "This is test program". using 16-bit microprocessor code.
Assume any necessary data

```
· MODEL PROC SMALL  
· STACK  
· DATA  
    STRING DB 'This is test program '$  
· CODE  
MAIN    PROC
```

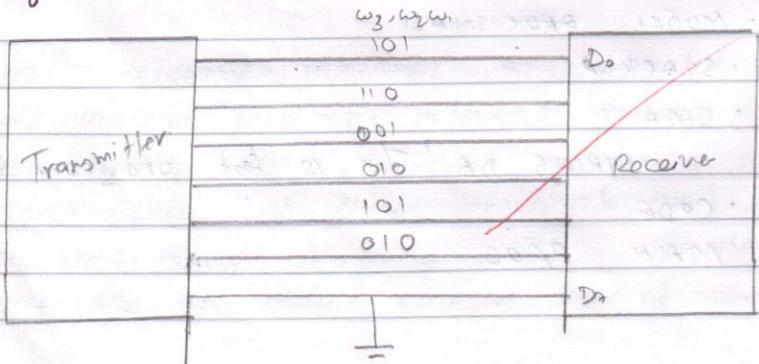
```

MOV AX, @DATA
MOV DS, AX
MOV DX, OFFSET STRING ; load offset string in
                        ; address DX
MOV AH, 09H ; AH = 09H for string display
              ; until $.
INT 21H ; DOS Interrupt function
MOV AX, 4C00H ; End request with AH=4CH
              ; or AX=4C00H
MAIN ENDP
END MAIN

```

Q Why parallel communication is required? Explain with reference to 8-bit system.

When a word of n bit is to be transmitted in parallel each bit is transmitted on a separate line along with a common ground line with respect to which the status of each line measured. Thus, a channel comprises of $(n+1)$ lines.



Here the time period required to transfer one word is equal to the time taken to transmit a bit. Parallel data transmission is impractical over long distance because of prohibitive cost of installing a large number of lines.

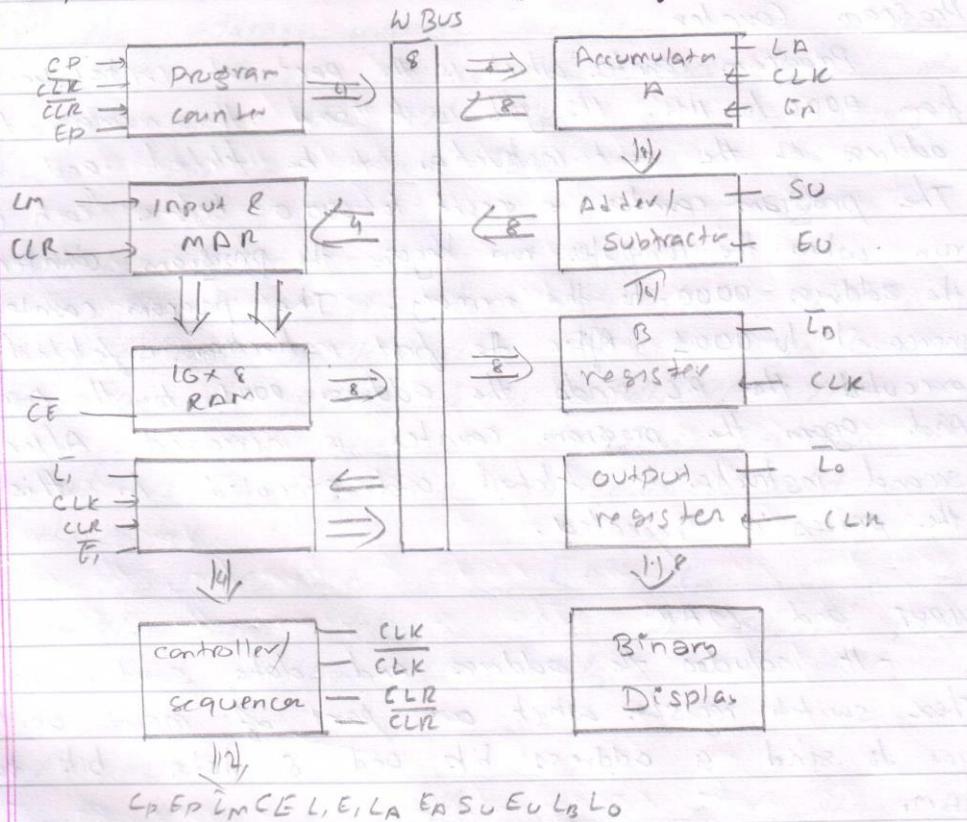
b) What is fetch & Execute cycle ? Explain with timing diagram.

[2066 & 2067 Q. No. 4]

A
S|9

2070

1) Explain with block diagram of SAP-1 computer.



SAP-1 Architecture

SAP-1 control unit consists of the program counter, the instruction register, and the controller-sequencer that produces the control word, the clear signals, and the clock signals. The SAP-1 ALU consists of an accumulator, an adder subtractor and a B register. The SAP-1 memory has the MAR and a 16x8 RAM. The I/O unit includes the input programming switch, the output port &

the binary display.

Program Counter

Program counter which is the part of control unit counts from 0000 to 1111. Its job is to send the memory the address of the next instruction to be fetched and executed. The program counter is reset to 0000 before each computer run when the computer run begins the program counter sends the address 0000 to the memory. The program counter is increased to 0001. After the first instruction is fetched and executed the PC sends the address 0001 to the memory. And again the program counter is increased. After the second instruction is fetched and executed. In this way the process is repeated.

INPUT and MAR

-It includes the address and data switch register. These switch registers which are part of input unit allow you to send 4 address bits and 8 data bits to the RAM.

During run, address in the program counter is latched into the MAR. A bit later the MAR applies this 4-bit address to RAM where a read operation is performed.

RAM

RAM is 16x8 slate. During a computer run, RAM receives 4-bit address from MAR and a read operation is performed. In this way the instruction or data word stored in the RAM is placed on the W-Bus.

Instruction register

- it is part of the control unit. It places the contents of the address memory locations on the W bus. At the same time IR is set up for loading on the next positive clock edge. The contents of the instruction register are split into two nibbles.

Controller sequencer.

The lower left block contains the controller sequencer. Before each computer run, a \overline{CLR} signal is sent to the PC to 0000 and to the IR. This resets PC to 0000 and writes at the last instruction in IR. The wires carrying the control word are called the control bus.

$$CON = CP EP \overline{IN} \overline{CE} \overline{L} \overline{E} \overline{I}_A EP SU EO \overline{I}_B \overline{I}_C$$

This word determines how the register will react to the next positive clock edge.

Accumulator:

It is a buffer register that stores the instruction intermediate answer during a computer run. It has two outputs, 2-state output goes directly to the adder-subtractor & the 3-state output goes to the W-bus.

Adder-Subtractor

SAP is one 2's complement adder-subtractor when S_0 is low the sum of Adder-subtractor is $S = A + B$. When S_0 is high, the content appears on the

the binary display.

Program Counter

Program counter which is the part of control unit counts from 0000 to 1111. Its job is to send the memory the address of the next instruction to be fetched and executed. The program counter is reset to 0000 before each computer run when the computer run begins the program counter sends the address 0000 to the memory. The program counter is increased to 0001. After the first instruction is fetched and executed the P.C sends the address 0001 to the memory. And again the program counter is increased. After the second instruction is fetched and executed. In this way the process is repeated.

INPUT and MAR

-It includes the address and data switch register. These switch registers which are part of input unit allow you to send 4 address bits and 8 data bits to the RAM.

During run, address in the program counter is latched into the MAR. A bit later the MAR applies this 4-bit address to RAM where a read operation is performed.

RAM

RAM is 16x8 state. During a computer run, RAM receives 4-bit address from MAR and a read operation is performed. In this way the instruction or data word stored in the RAM is placed on the W-Bus.

Instruction register

- It is part of the control unit. It places the contents of the address memory location on the W bus. At the same time IR is set up for loading on the next positive clock edge. The contents of the instruction register are split into two nibbles.

Controller sequencer.

The lower left block contains the controller sequencer. Before each computer run, a CLR signal is sent to the PC to 0000 and to the IR. This resets PC to 0000 and writes at the last instruction in IR. The 12 wires carrying the control word are called the control bus.

$$CON = CP EP \bar{IN} \bar{CE} \bar{L}_1 \bar{E}_1 \bar{L}_0 \bar{E}_0 S_U E_U \bar{L}_0 \bar{Z}_0$$

This word determines how the register will react to the next positive clock edge.

Accumulator:

It is a buffer register that stores the instruction intermediate answer during a computer run. It has two outputs, 2-state output goes directly to the adder-subtractor & the 3-state output goes to the W-bus.

Adder-Subtractor

SOP 1 or 2's complement adder-subtractor when S_0 is low the sum of Adder-subtractor is $S = A + B$. When S_0 is high, the content appears on the

w-bus $S = A + B'$

when E_0 is high, these can be appears on w-bus

B - register

- It is used in the arithmetic operation. A low I_E and the clock edge load the word on the w-bus into the B register. The two state output of the B register drives the adder-subtractor, supplying the number to be added or subtracted from the content of the accumulator.

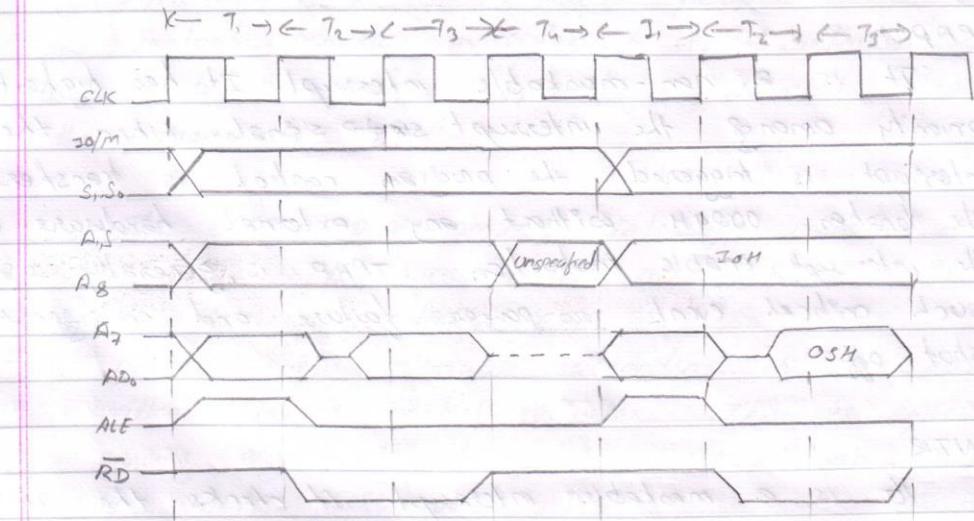
OUTPUT register

- At the end of the computer run, the accumulator contains the answer to the problem being solved. At this point, we need to transfer the answer to the outside world. This is where the output register is used.

Binary display:

The binary display is a row of 8 light emitting diodes (LEDs). Because each LED connects to one flip-flop of the output port the binary display shows us the contents of the output port.

- 3) Draw the timing diagram of MVI A, 05H instruction and explain it.



Timing diagram for MVI A, 05H

The MVI A, 36H instruction requires 2-machine cycles (M_1 & M_2). M_1 requires 4 states & M_2 requires 3-state total of 7-state as shown in fig. States of signals Z0/M - S₁ & S₀ specifies the 1st machine cycle as the opcode fetch.

In T₂ state the RD line to goes low, and the data 06H from memory location 1000H are placed on the data bus. The fetches cycle becomes complete in T₃ state. During T₄ state the contents of the bus are unknown.

7. Explain the function of the following signals:

TRAP

INTR

TRAP:

It is a non-maskable interrupt. It has highest priority among the interrupt signals. When the interrupt is triggered the program control is transferred to the location 0024H without any external hardware or the interrupt enable instruction. TRAP is generally used for such critical events as power failure and emergency shut off.

INTR

It is a maskable interrupt. It checks the INTR line (when interrupt enable flip flop is enabled using EI instruction) during the execution of each instruction, when the microprocessor is executing a program. If the line is high and the interrupt is enabled, the microprocessor completes the current instruction, disabled the interrupt enable flip flop and sends a INTR signal. The processor does not accept any interrupt request until the interrupt flip flop is enabled.

Discuss the importance of interrupt based system. Explain how interrupt controller (8259) can be used to handle interrupts

Interrupt is a signal that a peripheral board send to the central processor in order to request attention. The response to an interrupt request is directed or controlled by the microprocessor.

The 8259 A programmable interrupt controller designed to work with Intel microprocessor 8085, 8086 & 8088. The 8259 A interrupt controller can:

- 1) Manage eight interrupts according to the instruction written into its control registers. This is equivalent to providing eight interrupt pins on the processor in place of one INTR (8085) pins.
- 2) Resolve eight levels of interrupt priorities in a variety of modes, as fully nested mode, automatic rotation mode and specific rotation mode.
- 3) Mask each interrupt request individually.
- 4) Read the status of pending interrupts in service interrupt and masked interrupts
- 5) Be expanded to 64 priority level by cascading additional 8259 As

Observe the following program and write the content of Accumulator register B and flags after execution of each instruction.

MVI A, 45H

MVI B, 66H

ADD B

ANI 63H

HLT

	Accumulator	Register B	cy	F	P	Ac	Z	S
MVI A, 45H	45H	0	0	0	0	0	0	0
MVI B, 66H	66H	66H	0	0	0	0	0	0
ADD B	11H		1	0	0	0	0	0
ANI 63H	74H		1	0	0	0	0	0
HLT								

What are the various register in 8086 microprocessor? Explain the function of each register.

The register in 8086 are.

- 1) General Purpose register
- 2) Flag Register
- 3) Pointer Register
- 4) Index register

1) General Purpose Registers

- The execution unit (EU) has 8 general purpose registers labeled AH, AL, BH, BL, CH, CL, DH & DL.
 - These registers can be used individually for temporary storage of 8-bit data.
 - The AL register is also called accumulator.
 - Certain pairs of these general purpose registers can be used together to store 16 bit words.
 - Acceptable register pairs are AH & AL, BH & BL, CH & CL, DH & DL where they referred to as the AX register, BX register, CX register, DX register respectively
- AX \Rightarrow Accumulator Register CX \Rightarrow Count register
BX \Rightarrow Base register DX \Rightarrow Data register

2) FLAG Register:

- It is a flip-flop which indicates some condition produced by the execution of an instruction or controls certain operations of the EU.
- A 16 bit flag register in the EU contains 9 active flags.

- Among 9 flag there are 6 conditional flag and 3 control flag. The control flag like TF (Trap flag) is used for single stepping through program. IF (Interrupt enable flag) is used to allow or prohibit the interruption of the program. DF (Direction flag) is used for with string instruction?

3) Pointer register:

- 16 bit pointer registers are IP, SP & BP respectively. SP & BP are located in EU whereas IP is located in BIU.

3.1. STACK Pointer:

SP register provides an offset value, which when associated with the SS register

3.2. Base pointer:

BP can also combined with DI and SI as a base register for special addressing

4) INDEX Register

- 16-bit index registers are SI & DI

4.1. Source Index register

- 16-bit SI Register is required for some string handling operations

- SI is associated with the DS register

4.2. Destination Index register

- 16-bit DI register is also required for some string operations

- DI is associated with ES register