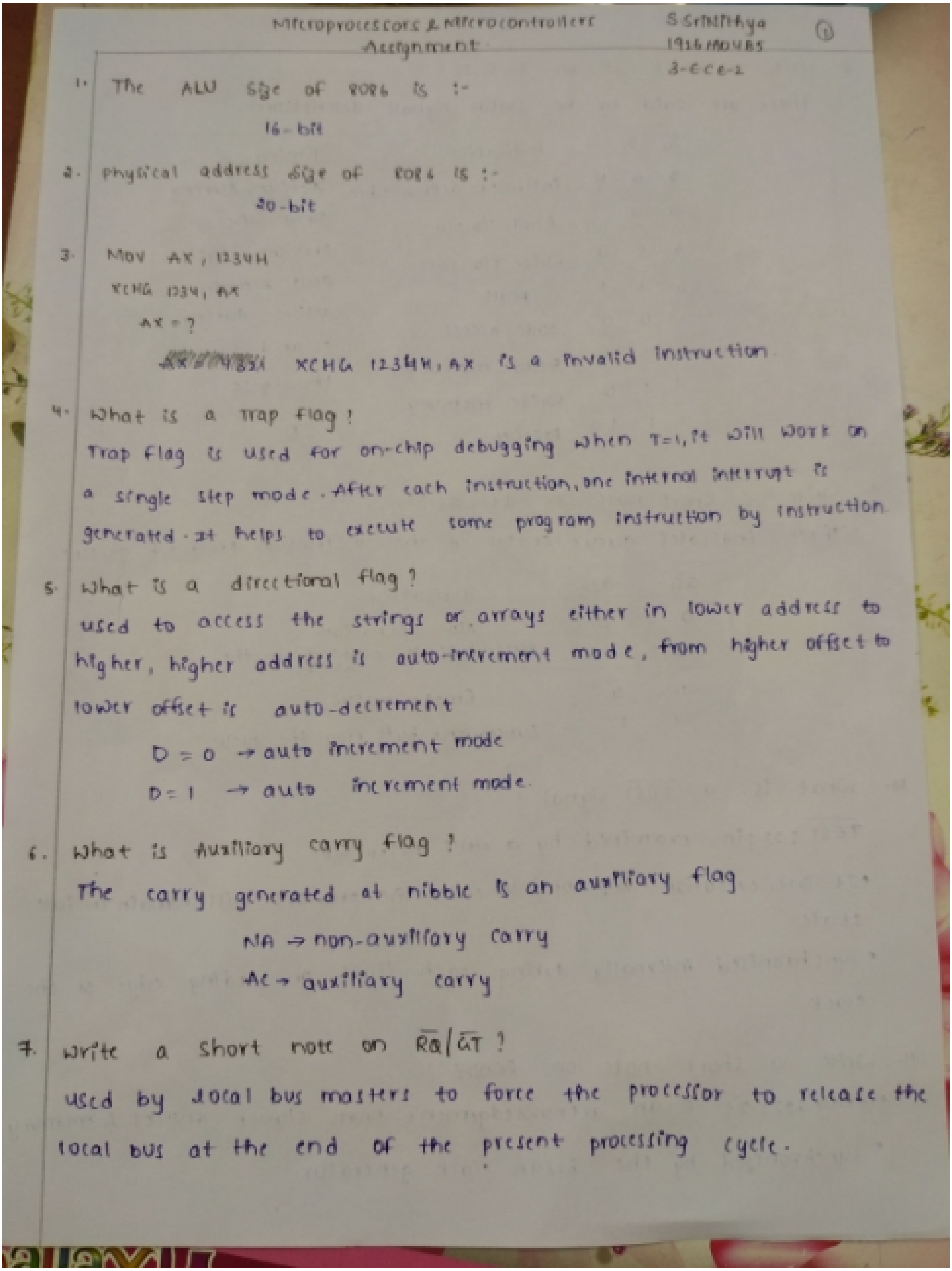


Microprocessors & Microcontrollers
Assignment

S.Srinithya
1916 MD485
3-ECE-2

1

1. The ALU size of 8086 is :-

16-bit

2. physical address size of 8086 is :-

20-bit

3. MOV AX, 1234H

XCHG 1234, AX

AX = ?

XCHG 1234H, AX is a invalid instruction.

4. What is a Trap flag !

Trap flag is used for on-chip debugging when T=1, it will work on a single step mode. After each instruction, one internal interrupt is generated. It helps to execute some program instruction by instruction.

5. What is a directional flag ?

used to access the strings or arrays either in lower address to higher, higher address is auto-increment mode, from higher offset to lower offset is auto-decrement

D = 0 → auto increment mode

D = 1 → auto increment mode.

6. What is Auxiliary carry flag ?

The carry generated at nibble is an auxiliary flag

NA → non-auxiliary carry

AC → auxiliary carry

7. write a short note on $\overline{RQ}/\overline{GT}$?

used by local bus masters to force the processor to release the local bus at the end of the present processing cycle.

8. write a short note on S_0, S_1, S_2 ?

These are said to be status signals description

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	indication	Timing
0	0	0	interrupt acknowledge	Active during T_4 of the previous cycle and remain active during T_1 and T_4 of the cycle.
0	0	1	Read I/o port	
0	1	0	write I/o port	
0	1	1	Halt	
1	0	0	code access	
1	0	1	Read memory	
1	1	0	write memory	
1	1	1	Passive	T_3

9. write a short note on QS_0, QS_1 ?

These indicates queue status of the instruction pre-fetch queue.

$\overline{QS_1}$	$\overline{QS_0}$	indication
0	0	No operation
0	1	first byte of opcode from the queue
1	0	Empty queue
1	1	Subsequent byte from the queue

10. What is a $\overline{TEST}$ signal ?

$\overline{TEST}$: 23-pin. examined by a WAIT instruction.

- If low, execution will continue, else, the processor will remain in idle state.
- synchronised internally during each clock on leading edge of the clock.

11. write a short note on READY ?

- READY : 22 : It is an acknowledgement from slower devices & memory.
- synchronized by the 8284A clock generator

12. Write a short note on M/IO ?

- M/I/O : 28 , 1: memory operation and 0: I/O operation
- Gets activated during previous T_4 and remains active until current T_4
- Tristated during local bus $\overline{HLDA}$.

13. What is $\overline{LOCK}$ signal ?

- $\overline{LOCK}$: 29 : Indicates the other system bus masters will be prevented from gaining the system bus.
- Activated by LOCK instruction prefix and remains active until the completion of next instruction.
- Tristated during local bus $\overline{HLDA}$.

14. What is $\overline{BHE}$ signal ?

- $\overline{BHE}$ is active low(0) during T_1 for read, write and interrupt acknowledges cycles.
- Status information is available during T_2, T_3, T_4.
- Becomes tristated during 'hold'.

15. What is $\overline{DEN}$ signal ?

- $\overline{DEN}$: 26 : indicates the availability of the valid address on the $AD_0 - AD_{15}$.
- Active from middle of T_2 until the middle of T_4.
- Tristated during local bus 'HLDA'

16. What is $\overline{INTA}$ signal ?

- $\overline{INTA}$: 24 : used as a read strobe for interrupt acknowledgement
- Goes low when the interrupt accepted by the processor.
- Remain low during T_2, T_3 & T_4.

17. Write the format of register-register instruction.

- It is 2 bytes long,
- first byte specifies the opcode and length of the operation is specified by W-bit.
- second byte has register operands and R/M field.

- R/M specifies the another operand as whether register/memory

opcode	destination operand	source operand

18. Write the format of register to/from with no displacement instruction

D_7 opcode D_0	D_7 D_6 MOD	D_3 D_4 D_5 REG	D_2 D_1 D_0 R/M

19. Write the format of register to/from with displacement instruction

D_7 opcode D_0	D_7 D_6 MOD	D_3 D_4 D_5 REG	D_2 D_1 D_0 R/M	D_7 lower byte of displacement D_0	D_7 upper byte of displacement D_0

4 byte instruction format

20. Write the format of immediate operand to register instruction.

D_7 opcode D_0	D_7 D_6 11	D_3 D_4 D_5 opcode	D_2 D_1 D_0 R/M	D_7 lower byte data D_0	D_7 upper byte data D_0

four byte immediate mode

21. Write the format of immediate operand to memory with 16-bit displacement instruction.

- It is 6 bytes long

D_7 opcode D_0	D_7 D_6 MOD	D_3 D_4 D_5 REG	D_2 D_1 D_0 R/M	D_7 lower byte of displacement D_0	D_7 upper byte of displacement D_0

D_7 lower byte data D_0	D_7 upper byte data D_0

22. Write a short note on 8086 Immediate addressing mode with an example.

- It is a part of instruction
- Immediate data can not be a destination operand

Eg: MOV AX, 1234H
MOV DL, 77H

23. Write a short note on 8086 direct addressing mode with an example.

- 16-bit memory address (offset) or an I/O address is directly specified

Eg: MOV AX, [1234H]
MOV DL, [1234H]
IN 80H
MOV [1234H], DL

24. write an Short note on 8086 register addressing mode with an example.

- If the data is a part of any register/content of any register it is said to be register addressing mode.

```
Eg: MOV  DI, AX
    MUL  BL
    DIV  CX
```

25. write a Short note on 8086 register indirect addressing mode with an example.

the offset address is a part of register or any of the GPR's

```
Eg: MOV  AX, [BX]
    ADD  [SI] ,[CX]
```

26. write a short note on 8086 indexed addressing mode with an example

- offset of the operand stored in any of the indexed registers.
- DS is the default segment register for SI and DI.
- In string instructions DS and ES are default segment registers for SI and DI, respectively.

```
Eg: MOV  AX, [SI]
    MOV  DL, [DI]
```

27. write a short note on 8086 register relative addressing mode with an example.

- effective address is formed by adding an 8-bit/16-bit with the displacement with contents of any of registers BX, BP, SI and DI

```
Eg: MOV  AX, 50H[BX]
    mov  10H[SI], BL
```

28. write a short note on 8086 based indexed addressing mode with an example.

- effective address is formed by adding content of a base register to the content of index register
- DS is the default segment register for SI and DI

- DS and ES are default segment Registers

  Eg: MOV AX, [BX][SI]

  MOV [BX][DI], DX

29. Write a Short note on 8086 relative based indexed addressing modes with an example ?

- Effective address is formed by adding an 8-bit/16bit displacement with the sum contents of any of the registers BX/BP and any of the index registers SI/DI

Eg: MOV AX, 50H[BX][SI]

ADD 50H[BX][SI], BP

30. Write a Short note on 8086 intrasegment addressing mode with an example.

- Intrasegment Direct :- Address lies in the same segment and appears directly in intruction as an immediate displacement value.

  Eg: JMP Short Label.

- Intrasegment Indirect :- Address lies in the same segment and passed to instruction indirectly.

  Eg: JMP [BX]

  JMP [BX+5000H]

31. Write a short note on 8086 Intersegment addressing mode with an example.

- Intersegment direct :- Address to which the control is to be transferred in a different segment.

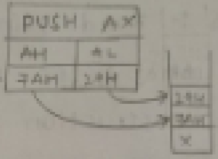
  Eg: JMP 5000H : 2000H

- Intersegment indirect :- Address lies in the different segment and is passed to the instruction indirectly.

  Eg: JMP [2000H]

32. Write a short note on PUSH and POP instructions.

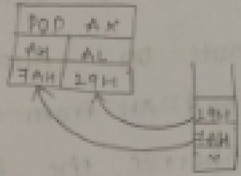
- PUSH :- push on to the stack
- Stack pointer is decremented by 2, no flags get affected
- Higher byte is pushed first and then the lower byte
- SS:SP points to top of the stack



Eg: PUSH AX
PUSH DS

- POP :- Pop from the stack
- Lower byte is popped first and then higher byte
- SP is incremented by 2, no flags get affected.

Eg: POP AX
POP DS



33. Write a note on IN and OUT instructions.

- IN :- input of the port

Eg: IN AL, 03H
IN AX, DX

- OUT :- output of the port
- port address may be direct | specified in DX
- only content of AL/AX can be written on output port.

Eg: OUT 03H, AL
OUT DX, AX

34. Write a note on XLAT instructions.

- XLAT : Translate.
- finds the codes using loop-up table in case of code conversion problems

```
Eg: MOV AX, SEG table
    MOV DS, AX
    MOV AL, code
    MOV BX, offset table
    XLAT
```

35. Write a note on LEA and LDS/LES Instructions.

- LEA :- Load effective Address

  Eg: LEA BX, ADR

  LEA SI, ADR [BX]

- LDS/LES :- Load pointer to DS/ES

- Loads the DS/ES registers and the specified destination register in the instruction with content of memory location specified as source in the instruction.

  Eg: LDS BX, 5000H

  LES BX, 5000H

36. Write a note on LAHF and SAHF Instructions.

- LAHF :- Load AH from lower byte of flag
- used to observe the status of all conditional code flags (Except OF)
- SAHF :- Store AH from lower byte of flag.
- used to sets/resets all the conditional code flags (except OF) depending on the corresponding bit value in AH register.

37. Write a note on PUSHF and POPF instructions.

- PUSHF :- push flags to stack (16-bit)
- POPF :- pop stack-content to flags (16-bit).

38. Write a note on IMUL and IDIV instructions.

- IMUL :- signed byte or word multiplication
- multiplies an signed byte/word by signed byte in AL or signed word in AX register, respectively.
- upper byte of 8-bit is stored in AH, & lower byte is stored in AL
- Upper byte of 16-bit is stored in DX & lower byte is stored in AX
- The unsigned higher bits of the result are filled with sign bit and CF, AF are cleared.

  Eg: IMUL BL

  IMUL CX

- IDIV :- signed division
- divides an unsigned word or double word by 8-bit/16-bit operand.
- dividend must be in AX for 16-bit in (DX)(AX) for 32-bit
- 16-bit division result : Quotient stored in AL, remainder in AH
- 32-bit division result : Quotient stored in AX, remainder in DX

39. Write a note on CBW and CWD instructions.

- CBW :- convert signed byte to word
- converts a signed byte in AL register into a signed word
- result stored in AX register.
- upper byte is filled with SIGN bit
- CWD :- convert signed word to double word
- copies the sign bit of AX and fill DX register with it
- doesn't affect any flags.

40. Write a note on REP instruction

- REP :- Repeat instruction prefix
- used as a prefix to other string manipulation instruction.
- instruction prefixed by REP executed until CX becomes ZERO.

41. Write a note on CALL instruction.

- CALL :- unconditional call
- calls a subroutine from the main program.

NEAR CALL :- The subroutine present in the same segment within the displacement of ±32KB, addressing mode is intrasegment.

FAR CALL :- the subroutine present in any other segment than the current. Then addressing mode is intersegment.

42. Write a note on RET instruction.

- RET :- Return from the procedure.
- must be executed at the end of the procedure.
- Store values of CS, IP and flags in the stack are retrieved into the corresponding registers and the execution of the main program continues.

43. Write a note on INTN instruction.

- INTN :- Interrupt Type N
- 'N' points to one of the 256 interrupts defined in 8086 (00H - FFH)
- 'N' is multiplied by 4 and this is taken as offset address into IP register.
- Segment address will be 0000H stored in CS.
- IF must be set to execute the instruction.

Eg: INT 20H

offset = 20H * 4 = 80H.

44. Write a note on INTO instruction.

- INTO :- Interrupt on overflow
- executed when OF is set.
- CS and IP are taken from address 0000H : 0010H
- Similar to type 4 interrupt.

Eg: INT 04H

offset = 04H * 4 = 10H.

45. How does 8086 differentiate an 8-bit and a 16-bit operation?

- AX is an 16-bit general purpose register which can be used as 2, 8-bit general purpose registers.

46. In an 8086 instructions both the operands can be memory ~~operations~~ operands T|F :

False.

47. In an 8086 instruction, both the operands cannot be memory operands T|F :

True

48. We can use segment registers as operands for arithmetic and logic operations : T|F

False

49. Internal RAM size of an 8051 microcontroller is ?

128 bytes

50 Internal ROM size of an 8051 microcontroller is ?

4kbytes

51. Write PSW of 8051

CY	AC	F0	RS1	RS0	OV	–	P

52. How many register banks are available in 8051 RAM ?

4 Register banks

53. Write a short note on JZ & JNZ instructions.

- JZ :- Execution jumps to label specified in instruction only certain condition is true.

JZ : Transfers execution to label, if ZF = 1

- JNZ :- Execution jumps to label specified in instruction only certain condition is true.

JNZ : Transfers execution to label, if ZF = 0

54. What is the difference between long, short and Absolute calls ?

- Absolute call :- The Acall instruction calls a subroutine located at the specific address. The PC incremented twice to obtained the address and is stored on the stack which is incremented twice.

- Long call :- LACALL is a 3-byte instruction where the first byte represents the opcode and second and thired bytes are used to provide the address of target subroutine, available in 64KB.

~~•Short call :-~~

55. one machine cycle of 8051 has _____ oscillator periods ?

12 oscillator periods.

56. Write an example of nested looping in 8051 ?

- Repeating a sequence of instructions a certain number of times is called a loop. An instruction DJNZ Reg, label is used to perform a loop operation. In this instruction, both the registers decrement and decision to jump are combined into a single instruction. Registers can be any of R_0-R_7.

57. minimum machine cycles needed to execute an 8051 instruction.

$$t_{inst} = \frac{c \times 12}{f_{crystal}}$$

- minimum 12 machine cycles needed to execute 8051 instruction
- Each instruction of 8051 has 1,2,4 machine cycles to execute an instruction.

~~If c=1~~, then ~~tinst = 1/12 [illegible]~~

58. Find the machine cycle for a crystal frequency of 11.059MHZ.

$$f_{XTAL} = 11.0592 \text{ MHZ}$$

$$f_{mc} = \frac{f_{XTAL}}{12}$$

$$= \frac{11.0592}{12\ d}$$

$$\boxed{f_{mc} = 0.9216 \text{ mH}}$$

59. 8051 ports are bit addressable (T/F) :

True

60. Write a note on 8051 immediate addressing mode with an example.

- Data is a part of instruction itself
- When an immediate data move is executed, the PC is automatically incremented to point to the byte following the opcode byte in the program memory.
- Data found at that location is copied into destination address.

- # is the mnemonic for immediate data.

  Eg: MOV A, # DATA

  MOV reg, #DATA.

61. Write a note on 8051 direct addressing mode with an example.

- All 128 bytes of internal RAM can be addresses directly.
- Internal RAM uses address from 00H - 7FH
- The SFRs are existing from 80H to FFH

  Eg: MOV A, dptr

  mov addr, DPH

62. Write a note on 8051 indirect addressing mode with an example.

- uses a register to hold the actual address that will finally be used in the data move.
- The register itself is not the address, but the register content.
- Indirect addressing for mov opcodes uses Ro or R1, often called data pointers to hold the address of one of the data locations.

  Eg: MOV @Rp, # data

  Mov @Rp, A

63. Write a note on 8051 register indirect addressing mode with an example.

- In which the address specifies which memory word or register contains not the operand but the address of the operand.

  Eg: Load R1, @100

64. Write a note on 8051 register addressing mode with an example.

- Registers are part of the ~~instruction~~ /opcode
- Registers A, DPTR, Ro-R7 may be part of the opcode mnemonic
- other registers may be addresses using direct addressing mode.

  Eg: MOV A, Ro

  MOV R5, DPH

65. Write a note on 8051 ADC instruction

- Intel 8051 don't have any in-built ADC, you have to interface it externally if required. It is an 8-bit successive approximation analog to digital converter.

66. Write a note on 8051 SBB instruction.

- In 8051 SUBB instruction subtracts the specified byte variable and the carry flag from the accumulator.. It will sets the CF if a borrow is required for bit 7 for result. If no borrow then CF is cleared, but no SBB instruction

67. Write a note on 8051 MUL instruction.

- The MUL instruction multiplies the unsigned 8-bit integer in the accumulator and the unsigned 8-bit integer in the B-register producing a 16-bit product. lower byte result is returned in accumulator, and higher byte is returned in B Register...

68. Write a note on 8051 CPL instruction.

- CPL instruction logically complements the value of the specified destination operand and stores the result back in the destination operand. Bits previously contains '1' will be changed to '0' and bits previously contains '0' will be changed to '1'.

69. Write a note on 8051 DIV instruction.

- DIV instruction divides the unsigned 8-bit integer in the accumulator by the unsigned 8-bit in register B. after the division, quotient is stored in accumulator and remainder is stored in B Register.

70. Write a note on 8051 MUL instruction.

- MUL instruction multiplies unsigned 8-bit integer in accumulator and unsigned 8-bit integer in B-Register producing a 16-bit product. Lower byte of product is returned in accumulator, and higher byte is returned in B-Register.

71. Write the value -128 in hexadecimal.

72. Write the value of -34h in hexadecimal.

73. Write a note on 8051 setb instruction.

- The setb instruction sets the bit operand to value 1, this can operate on the carry flag or any other directly addressable bit. no flags get affected.

74. Write the priority of the 8051 interrupts from high to low.

INT0 > Timer 0 > INT1 > Timer 1 > TI/RI

75. What is RxD signal ?

- RxD is present in Port 3.
- Port3.0 is RxD signal
- It is an serial data input (SBUF)

76. What is TxD signal ?

- TxD is present in Port 3.
- $P_3 \cdot 1$ is TxD signal
- It is an serial data output (SBUF)

77. What is $\overline{\text{INT0/1}}$ signal ?

- $\overline{\text{INT0/1}}$:- It is $P_3 \cdot 2$ pin in port 3
- It is an External interrupt 0 (TCON.1).

~~78. What is $\overline{EA}$ signal ?~~

~~- $\overline{EA}$:- It is an enable interrupt.~~
~~- cleared to 0 by program to disable all interrupts.~~

79. What is $\overline{PEN}$ signal ?

- $\overline{PEN}$:- It is stands for program store enable.
- It is output, active low-pin.
- This is used to read external memory.
- It is used to enable/disable the external memory interfacing.

80. Write a note on 8051 ORG directive.

- The origin (ORG) directive is used to indicate the beginning of the addresses the number that comes after ORG can be either in hex or in decimal if the number is not fdlowed by 'H' then it is in decimal and the assembler will convert it into hex some assombler use "ORG" instead of "ORG" for origin directive.

78. What is $\overline{EA}$ signal ?

- $\overline{EA}$ is pin no-31, which is used to fetch the code bytes from an external memory.
- Program can be stored either completely in internal ROM, complettly in external ROM, or combination of internal and external ROM

——— X ———