

1) Briefly explain FCFS.

FCFS stands for First-come, first-served. It is one type of scheduling algorithm. In this scheme, the process that requests the CPU first is allocated the CPU first.

Implementation is managed by a FIFO queue.

2) What is RR scheduling algorithm?

RR (round-robin) scheduling algorithm is primarily aimed for time-sharing systems. A circular queue is a setup in such a way that the CPU scheduler goes around that queue, allocating CPU to each process for a time interval of up to around 10 to 100 milliseconds.

3) What are necessary conditions which can lead to a deadlock situation in a system?

Deadlock situations occur when four conditions occur simultaneously in a system: Mutual exclusion; Hold and Wait; No preemption; and Circular wait.

Banker's algorithm is one form of deadlock-avoidance in a system. It gets its name from a [banking](#) system wherein the bank never allocates available cash in such a way that it can no longer satisfy the needs of all of its customers.

4) What factors determine whether a detection-algorithm must be utilized in a deadlock avoidance system?

One is that it depends on how often a deadlock is likely to occur under the implementation of this algorithm. The other has to do with how many processes will be affected by deadlock when this algorithm is applied.

5) State the main difference between logical from physical address space.

Logical address refers to the address that is generated by the CPU. On the other hand, physical address refers to the address that is seen by the memory unit.

6) What is the basic function of paging?

Paging is a memory management scheme that permits the physical address space of a process to be noncontiguous. It avoids the considerable problem of having to fit varied sized memory chunks onto the backing store.

7) What is fragmentation?

Fragmentation is memory wasted. It can be internal if we are dealing with systems that have fixed-sized allocation units, or external if we are dealing with systems that have variable-sized allocation units.

8) What are the different types of CPU registers in a typical operating [system design](#)?

- Accumulators
- Index Registers
- Stack Pointer
- General Purpose Registers

9) What is an Assembler?

An assembler acts as a translator for low-level language. Assembly codes written using mnemonic commands are translated by the Assembler into machine language.

10) What is preemptive multitasking?

Preemptive multitasking allows an operating system to switch between software programs. This, in turn, allows multiple programs to run without necessarily taking complete control over the processor and resulting in system crashes.

11) What is the difference between process and program?

A program while running or executing is known as a process.

12) What is the use of paging in operating system?

Paging is used to solve the external fragmentation problem in operating system. This technique ensures that the data you need is available as quickly as possible.

13) What are the four necessary and sufficient conditions behind the deadlock?

These are the 4 conditions:

- 1) **Mutual Exclusion Condition:** It specifies that the resources involved are non-sharable.

2) **Hold and Wait Condition:** It specifies that there must be a process that is holding a resource already allocated to it while waiting for additional resource that are currently being held by other processes.

3) **No-Preemptive Condition:** Resources cannot be taken away while they are being used by processes.

4) **Circular Wait Condition:** It is an explanation of the second condition. It specifies that the processes in the system form a circular list or a chain where each process in the chain is waiting for a resource held by next process in the chain.

14) What is a thread?

A thread is a basic unit of CPU utilization. It consists of a thread ID, program counter, register set and a stack.

15) What is FCFS?

FCFS stands for First Come, First Served. It is a type of scheduling algorithm. In this scheme, if a process requests the CPU first, it is allocated to the CPU first. Its implementation is managed by a FIFO queue.

16) What is fragmentation?

Fragmentation is a phenomenon of memory wastage. It reduces the capacity and performance because space is used inefficiently.

17) How many types of fragmentation occur in Operating System?

There are two types of fragmentation:

- **Internal fragmentation:** It is occurred when we deal with the systems that have fixed size allocation units.
- **External fragmentation:** It is occurred when we deal with systems that have variable-size allocation units.

18) What is semaphore?

Semaphore is a protected variable or abstract data type that is used to lock the resource being used. The value of the semaphore indicates the status of a common resource.

There are two types of semaphore:

- Binary semaphores
- Counting semaphores

19) What is a binary Semaphore?

Binary semaphore takes only 0 and 1 as value and used to implement mutual exclusion and synchronize concurrent processes.

.....

1. Define system software.

It consists of variety of programs that supports the operation of the computer. This software makes it possible for the user to focus on the other problems to be solved without needing to know how the machine works internally.

Eg: operating system, assembler, loader.

2. Give some applications of OS.

-to make the computer easier to use

-to manage the resources in computer

-process management

-data and memory management

-to provide security to the user.

Operating system acts as an interface between the user and the system

Eg: windows, linux, unix, dos

3. Define compiler and interpreter.

Compiler is a set of program which converts the whole high level language program to machine language program.

Interpreter is a set of programs which converts high level language program to machine language program line by line.

No intermediate storage is needed

4. Define loader.

Loader is a set of program that loads the machine language translated by the translator into the main memory and makes it ready for execution.

5. What is the need of MAR register.

MAR (memory address register) is used to store the address of the memory from which the data is to be read or to which the data is to be written.

8. Define indirect addressing.

In the case of immediate addressing the operand field gives the memory location. The word from the given address is fetched and it gives the address of the operand.

Eg: ADD R5, [600]

Here the second operand is given in indirect addressing mode. First the word in memory location 600 is fetched and which will give the address of the operand.

9. Define immediate addressing.

In this addressing mode the operand value is given directly. There is no need to refer memory. The immediate addressing is indicated by the prefix „#“.

Eg: ADD #5

In this instruction one operand is in accumulator and the second operand is a immediate value the value 5 is directly added with the accumulator content and the result is stored in accumulator.

14. What are the instruction formats used in SIC/XE architecture? Give any one format. Format 1 (1 byte), Format 2 (2 bytes), Format 3 (3 bytes) & Format 4(4 bytes) Are the different instructions used in SIC/XE architecture.

Format 2:

8

4

4

OPCODE

R1

UNIT II-ASSEMBLERS

2 Marks

1. Define the basic functions of assembler.

*translating mnemonic operation codes to their machine language equivalents.

*Assigning machine addresses to symbolic labels used by the programmer.

2. What is meant by assembler directives. Give example.

These are the statements that are not translated into machine instructions, but they provide instructions to assembler itself.

Also called pseudo instructions

example START,END,BYTE,WORD,RESW and RESB.

3 .What is forward references?

It is a reference to a label that is defined later in a program.

Consider the statement

```
10 1000 STL RETADR
```

```
80 1036 RETADR RESW 1
```

The first instruction contains a forward reference RETADR.If we attempt to translate the program line by line,we will unable to process the statement in line 10 because we do not know the address that will be assigned to RETADR .The address is assigned later(in line 80) in the program.

4.What are the three different records used in object program?

The header record,text record and the end record are the three different records used in object program.

The header record contains the program name,starting address and length of the program. Text record contains the translated instructions and data of the program.

End record marks the end of the object program and specifies the address in the program where execution is to begin.

5.What is the need of SYMTAB(symbol table) in assembler?

The symbol table includes the name and value for each symbol in the source program,together with flags to indicate error conditions.Some times it may

contains details about the data area. SYMTAB is usually organized as a hash table for efficiency of insertion and retrieval.

6. What is the need of OPTAB(operation code table) in assembler?

The operation code table contain the mnemonic operation code and its machine language equivalent. Some assemblers it may also contains information about instruction format and length.OPTAB is usually organized as a hash table,with mnemonic operation code as the key.

7.What are the symbol defining statements generally used in assemblers?

* „EQU“-it allows the programmer to define symbols and specify their values directly.The general format is symbol **EQU** value

* „ORG“-it is used to indirectly assign values to symbols.When this statement is encountered the assembler resets its location counter to the specified value.

The general format is **ORG** value

In the above two statements value is a constant or an expression involving constants and previously defined symbols.

8.Define relocatable program.

An object program that contains the information necessary to perform required modification in the object code depends on the starting location of the program during load time is known as relocatable program.

10. Write the steps required to translate the source program to object program.

- Convert mnemonic operation codes to their machine language equivalents.
Convert symbolic operands to their equivalent machine addresses
- Build the machine instruction in the proper format.

- Convert the data constants specified in the source program into their internal machine representation
- Write the object program and assembly listing.

11 .What is the use of the variable LOCCTR(location counter) in assembler?

This variable is used to assign addresses to the symbols. LOCCTR is initialized to the beginning address specified in the START statement. After each source statement is processed the length of the assembled instruction or data area to be generated is added to LOCCTR and hence whenever we reach a label in the source program the current value of LOCCTR gives the address associated with the label.

12. Define load and go assembler.

One pass assembler that generate their object code in memory for immediate execution is known as load and go assembler. Here no object programmer is written out and hence no need for loader.

13 .What are the two different types of jump statements used in MASM assembler.

Near jump

A near jump is a jump to a target in the same segment and it is assembled by using a current code segment CS.

Far jump

A far jump is a jump to a target in a different code segment and it is assembled by using different segment registers .

15. Differentiate the assembler directives RESW and RESB.

RESW –It reserves the indicated number of words for data area. Eg:

10 1003 THREE RESW 1

In this instruction one word area(3 bytes) is reserved for the symbol THREE. If the memory is byte addressable then the address assigned for the next symbol is 1006.

RESB –It reserves the indicated number of bytes for data area. Eg: 10 1008
INPUT RESB 1

In this instruction one byte area is reserved for the symbol INPUT .Hence the address assigned for the next symbol is 1009.

1 6. Define modification record and give its format

This record contains the information about the modification in the object code during program relocation. the general format is

Col 1 M

Col 2-7 starting location of the address field to be modified relative to the beginning of the program

Col 8-9 length of the address field to be modified in half bytes.

17. Write down the pass numbers(PASS 1/ PASS 2) of the following activities that occur in a two pass assembler:

- a. Object code generation
- b. Literals added to literal table
- c. Listing printed
- d. Address location of local symbols

Answer:

- a. Object code generation - PASS 2
- b. Literals added to literal table – PASS 1 c. Listing printed – PASS2
- d. Address location of local symbols – PASS 1

20. What is meant by external references?

Assembler program can be divided into many sections known as control sections and each control section can be loaded and relocated independently of the others. If the instruction in one control section need to refer instruction or data in another control section, the assembler is unable to process these references in normal way. Such references between control are called external references.

21 .Define control section.

A control section is a part of the program that maintain its identity after assembly; each control section can be loaded and relocated independently of the others.

Control sections are most often used for subroutines. The major benefit of using control sections is to increase flexibility.

22. What is the difference between the assembler directive EXTREF and EXTDEF.

EXTDEF names external symbols that are defined in a particular control section and may be used by other sections.

EXTREF names external symbols that are referred in a particular control section and defined in another control section.

23. Give the general format of define record.

This record gives information about external symbols that are defined in a particular control

section. The format is

Col 1 D

Col 2-7 name of external symbol defined in this control section Col 8-13 relative address of the symbol within this control section Col 14-73 name and relative address for other external symbols.

24. Give the use of assembler directive CSECT and USE

CSECT - used to divide the program into many control sections

USE – used to divide the program into many blocks called program blocks

25. What is the use of the assembler directive START.

The assembler directive START gives the name and starting address of the program. The format is PN START 1000

Here PN – name of the program

1000 – starting address of the program.

UNIT III (LOADERS AND LINKERS)

1. What are the basic functions of loaders.

Loading – brings the object program into memory for execution

Relocation – modifies the object program so that it can be loaded at an address different from the location originally specified

Linking – combines two or more separate object programs and also supplies the information needed to reference them.

2. Define absolute loader

The loader, which is used only for loading, is known as absolute loader. e.g. Bootstrap loader

3. What is meant by bootstrap loader?

This is a special type of absolute loader which loads the first program to be run by the computer. (usually an operating system)

4. What are relative (relocative) loaders?

Loaders that allow for program relocation are called relocating (relocative) loaders.

5. What is the use of modification record?

Modification record is used for program relocation. Each modification record specifies the starting address and the length of the field whose value is to be altered and also describes the modification to be performed.

6. What are the 2 different techniques used for relocation?

Modification record method and relocation bit method.

7. Relocation bit method

If the relocation bit corresponding to a word of object code is set to 1, the program's starting address is

to be added to this word when the program is relocated.

Bit value 0 indicates no modification is required.

8. Define bit mask

The relocation bits are gathered together following the length indicator in each text record and which is called as bit mask. For e.g. the bit mask FFC(1111111100) specifies that the first 10 words of object code are to be modified during relocation.

9. What is the need of ESTAB.

It is used to store the name and address of each external symbol. It also indicates in which control section the symbol is defined.

10. What is the use of the variable PROGADDR.

It gives the beginning address in memory where the linked program is to be loaded. The starting address is obtained from the operating system.

11. Write the two passes of a linking loader.

Pass 1: assigns address to all external symbols

Pass 2: it performs actual loading, relocation and linking.

12. Define automatic library search.

In many linking loaders the subroutines called by the program being loaded are automatically fetched from the library, linked with the main program and loaded. This feature is referred to as automatic library search.

13. List the loader options INCLUDE & DELETE. The general format of INCLUDE is

INCLUDE program_name (library name). This command directs the loader to read the designated object

program from a library and treat it as the primary loader input. The general format of DELETE

command is DELETE Csect-name

It instructs the loader to delete the named control sections from the sets of programs loaded.

14. Give the functions of the linking loader.

The linking loader performs the process of linking and relocation. It includes the operation of automatic library search and the linked programs are directly loaded.

16. Define dynamic linking.

If the subroutine is loaded and linked to the program during its first call (run time), then it is called as dynamic loading or dynamic linking.

17. Write the advantage of dynamic linking.

- a) it has the ability to load the routine only when they are needed
- b) The dynamic linking avoids the loading of entire library for each execution

18. What is meant by static executable and dynamic executable?

In static executable, all external symbols are bound and ready to run. In dynamic executables some symbols are bound at run time.

20. Write the absolute loader algorithm.

Begin

Read Header record

Verify program name and length Read first text record

While record type != „E“ do

Begin

Moved object code to specified location in memory Read next object program record

End

Jump to address specified in End record

Ktunotes.in

1 .Define macro processor.

Macro processor is system software that replaces each macroinstruction with the corresponding group of source language statements. This is also called as expanding of macros.

2. What do macro expansion statements mean?

These statements give the name of the macroinstruction being invoked and the arguments to be used in expanding the macros. These statements are also known as macro call.

3. What are the directives used in macro definition?

MACRO - it identifies the beginning of the macro definition

MEND - it marks the end of the macro definition

4.What are the data structures used in macro processor?

DEFTAB – the macro definitions are stored in a definition table ie it contains a macro prototype and the statements that make up the macro body.

-NAMTAB – it is used to store the macro names and it contains two pointers for each macro instruction which indicate the starting and end location of macro definition in DEFTAB.it also serves as an index to DEFTAB

ARGTAB – it is used to store the arguments during the expansion of macro invocations.

5.Define conditional macro expansion.

If the macro is expanded depends upon some conditions in macro definition (depending on the arguments supplied in the macro expansion) then it is called as conditional macro expansion.

6.What is the use of macro time variable?

Macro time variable can be used to store working values during the macro expansion. Any symbol that begins with the character & and then is not a macro instruction parameter is assumed to be a macro time variable.

8.What is meant by positional parameters?

If the parameters and arguments were associated with each other according to their positions in the macro prototype and the macro invocation statement, then these parameters in macro definitions are called as positional parameters.

10.What are known as nested macro call?

The statement in which a macro calls on another macro,is called nested macro call. In the nested macro call, the call is done by outer macro and the macro called is the inner macro.

11. How the macro is processed using two passes? Pass 1: processing of definitions

Pass 2: actual-macro expansion.

12. Give the advantage of line by line processors.

-it avoids the extra pass over the source program during assembling

-it may use some of the utility that can be used by language translators so that can be loaded once.

13. What is meant by line by line processor

This macro processor reads the source program statements, process the statements and then the output lines are passed to the language translators as they are generated, instead of being written in an expanded file.

14. Give the advantages of general-purpose macroprocessors.

*The programmer does not need to learn about a macro facility for each compiler.

*Overall saving in software development cost and a maintenance cost

15. What is meant by general-purpose macro processors?

The macro processors that are not dependent on any particular programming language, but can be used with a variety of different languages are known as general purpose macro processors. Eg. The ELENA macro processor.

16. What are the important factors considered while designing a general purpose macroprocessors?

-comments

-grouping of statements tokens

-syntax used for macro definitions

17.What is the symbol used to generate unique labels?

\$ symbol is used in macro definition to generate unique symbols.Each macro expansion the \$

symbol is replaced by \$XX,where XX is the alpha numeric character.

19.Mention the tasks involved in macro expansion.

x identify the macro calls in the program the values of formal parameters are identified x maintain the values of expansion time variables declared in a macro x expansion time control flow is organized x determining the values of sequencing symbols x expansion of a model statement is performed

20. How to design the pass structure of a macro assembler?

To design the structure of macro-assembler, the functions of macro preprocessor and the conventional assembler are merged. After merging, the functions are structured into passes of the macro assembler.

.....

1) How many bit 8086 processor is?

ANSWER: 16-bit processor

2) What is the size of data bus?

ANSWER: 16-bit

3) What is the size of address bus?

ANSWER: 20-bit

4) What is the maximum addressing capacity of 8086?

ANSWER: The maximum memory capability of 8086 is 1MB

5) What are the functional units of 8086?

ANSWER: 8086 has two independent functional units because of that the processor speed is more. The Bus interface unit and Execution unit are the two functional units.

6) What is the function of BIU?

ANSWER: Bus interface unit is responsible for transferring the data addresses on the buses necessary for execution unit.

7) What is the function of EU?

ANSWER: Execution Unit receives program instruction codes and data from BIU, executes these instructions and store the result in general registers.

9) What is the function of AX?

ANSWER: It is accumulator register and used for I/O, arithmetic and logical

operations by default. $AX (16\text{-bit}) = AL (8\text{-bit}) + AH (8\text{bit})$

10) What is the function of BX?

ANSWER: It is base register. It is used to store base address and is the only register used as offset address pointer. $BX (16\text{-bit}) = BL (8\text{-bit}) + BH (8\text{bit})$

11) What is the function of CX?

ANSWER: It is counter register. CX register is used as default counter.

$CX \text{ (16-bit)} = CL \text{ (8-bit)} + CH \text{ (8bit)}$

12) What is the function of DX?

ANSWER: It is data register. It is where the data is stored. Whenever an interrupt is called to print the data, it prints the data stored in data register.

$DX \text{ (16-bit)} = DL \text{ (8-bit)} + DH \text{ (8bit)}$

13) How much data is read at once and where it is stored?

ANSWER: 1 byte of data is read at once and is stored in AL register.

14) How physical address is generated?

ANSWER: Physical address is generated using Memory paging Unit.

15) Which are the pointers used in 8086?

ANSWER: IP (Instruction pointer): 16-bit register that stores the offset address of next instructions to be executed.

BP (Base pointer): 16-bit register that stores the offset address of the data or parameters within the stack.

SP (Stack pointer): 16-bit register that stores the offset address of the top most element of the stack.

16) Which is the default pointer for CS/ES?

ANSWER: DI register (Destination index register)

17) Which is the default pointer for DS?

ANSWER: SI register (source index register)

18) Use of CS register?

ANSWER: It contains upper 4 nibbles of starting address of code segment. Used for normal programming purpose. It is not directly referenced

19) Use of DS register?

ANSWER: It contains upper 4 nibbles of starting address of data segment. Used while computing 20-bit physical address of data.

20) Use of SS register?

ANSWER: It contains upper 4 nibbles of starting address of stack segment. This

Register permits the implementation of stack in memory. It is not directly referenced.

21) Use of ES register?

ANSWER: It contains upper 4 nibbles of starting address of extra segment. Used only during string manipulation.

22) What is the size of instruction queue in 8086?

ANSWER: 6 byte

23) Which are the registers present in 8086?

ANSWER: General purpose registers (AX, BX, CX, and DX)

Index registers (SI and DI)

Segment registers (CS, DS, SS and ES)

Pointer registers (IP, BP and SP)

Flag register

24) How many segments are there?

ANSWER: there are 4 segments

Code segment

Data segment

Stack segment

Extra segment

25) When is extra segment used?

ANSWER: Extra segment is used when data segment is not sufficient enough for the program.

26) What is the size of each segment?

ANSWER: 64 k bytes

27) What do you mean by .model?

ANSWER: Selects the programming model

28) Difference between tiny, small, medium, and huge.

29) What do you mean by directives?

ANSWER: Directives (Also called pseudo- operations) indicate how an operand or section of program is to be processed by the assembler.

E.g.: DB, DW, BYTE, .CODE, .MODEL etc...

30) What is DD, DW, DB?

ANSWER: DD- data double word

DW- data word

DB-data byte

These are the directives used to declare a variable of type double word, word and byte respectively.

31) What is the function of 01h of int 21h?

ANSWER: read the data character by character.

32) What is the function of 02h of int 21h?

ANSWER: Print the data character by character.

33) What is the function of 09h of int 21h?

ANSWER: print the data as a string.

34) What is the function of 08h of int 21h?

ANSWER: read the data as a string.

35) What is the function of 4ch of int 21h?

ANSWER: Signifies the End of the DOS program.

36) What does int 21h signify?

ANSWER: Execute the interrupt

37) What is the size of flag register? Explain all

ANSWER: size of the flag register is 16-bit STATUS FLAGS

Carry flag: set when carry occurs after an operation, otherwise reset.

Parity flag: set if the result of an operation contains even number of 1 bits, otherwise reset.

Auxiliary carry flag: set if there is carry from lower nibble, otherwise reset.

Zero flag: set if the result of an arithmetic or logical operation is zero, otherwise reset.

Sign flag: If the MSB of the result is 1 then the flag is set, otherwise reset.

Overflow flag: set when carry out of MSB and carry into MSB is different, otherwise reset.

CONTROL FLAGS

Trap flag: To execute the program in a single step, this flag should be set.

Interrupt flag: If this flag is set, then interrupts are enabled else the interrupts are disabled.

Direction flag: This is used during string manipulation.

39) What is procedure?

ANSWER: procedure is also called functions or sub routines. A procedure is a collection of instructions that perform some specific activity when executed in a sequence.

42) What do you mean by assembler?

ANSWER: assembler is used to translate the high level language program to machine code.

Example: MASM

43) What do you mean by linker?

ANSWER: The linker is a program that allows the user to combine different objects into a single executable file.

44) What do you mean by loader?

ANSWER: It is a program that is used to assign specific addresses of each object code during loading.

45) What do you mean by emulator?

ANSWER: Emulators are used to test and debug the hardware and software of an external system.

46) What do you mean by debugger?

ANSWER: debugger is a program that allows the user to load the object program into the memory, execute the program in single step.

47) What do you mean by editor?

ANSWER: editor is a program that allows the user to create/edit a file containing the text, program or data.

48) What do you mean by compiler?

ANSWER: Compiler is used to translate the high-level language program into

machine code at a time. It doesn't require special instruction to store in a memory, it stores automatically. The Execution time is less compared to Interpreter.

49) List some stack related instructions

ANSWER: PUSH, POP

50) What do you mean by 20 dup (0)?

ANSWER: 20 memory locations are reserved and initialized to 0. Dup stands for duplicate.

52) Can you perform 32-bit operation with 8086?

ANSWER: Yes

56) What type of Queue in 8086?

ANSWER: FIFO

57) In string pointer which is the default source pointer?

ANSWER: SI

58) In string pointer which is the default destination pointer?

ANSWER: DI

60) What is MUL BX? Where is the result stored?

ANSWER: the value of BX is multiplied with the value of AL and stored in AL.

62) What is the number representation system you have used?

ANSWER: Hexadecimal

63) What is LEA?

ANSWER: Load Effective Address, load the address of the data that is to be printed to dx.

64) What is the difference between MUL and IMUL?

ANSWER: MUL instructions is used when we deal with unsigned numbers.

IMUL instruction is used when we deal with signed numbers.

65) What is difference between DIV and IDIV?

ANSWER: DIV instructions is used when we deal with unsigned numbers.

IDIV instruction is used when we deal with signed numbers.

66) What is difference between shift and rotate?

ANSWER:

Shift: move numbers to left or right within a register or memory location.

Rotate: position binary data by rotating the information in a register or memory location, either from one end to another or through the carry flag.

67) List some of the string related instructions?

ANSWER: MOVS, LODS, STOS, INS, OUTS, SCAS, CMPS

68) What do you mean by pipelining?

ANSWER: the feature of fetching the next instruction by BIU when the current instruction is executing by EU is called pipelining.

69) What is the speed of execution in 8086 processor?

ANSWER: 400ns or 2.5 MIPS

70) What is a Microprocessor?

ANSWER: Microprocessor is a program-controlled device, which fetches the instructions from memory, decodes and executes the instructions. Most Micro Processors are single- chip devices.

72) What are the flags in 8086?

ANSWER: carry flag, Parity flag, Auxiliary carry flag, Zero flag, Overflow flag, Trace flag, Interrupt flag, Direction flag, and Sign flag.

73) What is the Maximum clock frequency in 8086?

ANSWER: 5 MHz is the Maximum clock frequency in 8086

74) What are the various segment registers in 8086?

ANSWER: Code, Data, Stack, Extra Segment registers in 8086

75) Logic calculations are done in which type of registers?

ANSWER: Accumulator is the register in which Arithmetic and Logic calculations are done.

76) How 8086 is faster than 8085?

ANSWER: Because of pipelining concept. 8086 BIU fetches the next instruction when EU busy in executing another instruction.

77) Which Segment is used to store interrupt and subroutine return address registers?

ANSWER: Stack Segment in segment register is used to store interrupt and subroutine return address registers.

78) What does microprocessor speed depend on?

ANSWER: The processing speed depends on DATA BUS WIDTH.

79) What is flag?

ANSWER: Flag is a flip-flop used to store the information about the status of a processor and the status of the instruction executed most recently.

80) Which Flags can be set or reset by the programmer and also used to control the operation of the processor?

ANSWER: Trace Flag, Interrupt Flag, Direction Flag.

81) In how many modes 8086 can be operated and how?

ANSWER: 8086 can be operated in 2 modes. They are Minimum mode if MN/MX pin is active high and maximum mode if MN/MX pin is ground.

82) What is the difference between min mode and max mode of 8086?

ANSWER: Minimum mode operation is the least expensive way to operate the 8086 microprocessor because all the control signals for the memory and I/O are generated by the microprocessor. In Maximum mode some of the control signals must be externally generated. This requires the addition of an external bus controller. It is used only when the system contains external coprocessors such as 8087 arithmetic coprocessor.

83) Which bus controller is used in maximum mode of 8086?

ANSWER: 8288 bus controller is used to provide the signals eliminated from the 8086 by the maximum mode operation.

84) What is stack?

ANSWER: Stack is a portion of RAM used for saving the content of Program Counter and general purpose registers.

85) Which Stack is used in 8086?

ANSWER: FIFO (First In First Out) stack is used in 8086. In this type of Stack the first stored information is retrieved first.

86) What is the position of the Stack Pointer after the PUSH instruction?

ANSWER: The address line is 02 less than the earlier value.

87) What is the position of the Stack Pointer after the POP instruction?

ANSWER: The address line is 02 greater than the earlier value.

88) What is interrupt?

ANSWER: Interrupt is a signal send by external device to the processor so as to request the processor to perform a particular work.

89) What are the various interrupts in 8086?

ANSWER: Maskable interrupts, Non-Maskable interrupts.

90) What is meant by Maskable interrupts?

ANSWER: An interrupt that can be turned off by the programmer is known as Maskable interrupt.

91) What is Non-Maskable interrupts?

ANSWER: An interrupt which can be never be turned off (i.e. disabled) is known as Non-Maskable interrupt.

92) Which interrupts are generally used for critical events?

ANSWER: Non-Maskable interrupts are used in critical events. Such as Power failure, Emergency shut off etc.

93) What is a SIM and RIM instruction?

ANSWER: SIM is Set Interrupt Mask. Used to mask the hardware interrupts.
RIM is

Read Interrupt Mask. Used to check whether the interrupt is Masked or not.

94) What is macro?

ANSWER: Macro is a set of instructions that perform a task and all the instructions defined in it is inserted in the program at the point of usage.

95) What is the difference between Macro and Procedure?

ANSWER: A procedure is accessed via a CALL instruction and a macro will inserted in the program at the point of execution.

96) What is meant by LATCH?

ANSWER: Latch is a D- type flip-flop used as a temporary storage device controlled by a timing signal, which can store 0 or 1. The primary function of a Latch is data storage. It is used in output devices such as LED, to hold the data for display

97) What is the disadvantage of microprocessor?

ANSWER: It has limitations on the size of data. Most Microprocessor does not support floating-point operations.