CS1203-SYSTEM SOFTWARE

UNIT I-INTRODUCTION

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 with out needing to know how the machine works internally.

   Eg: operating system, assembler, loader.

2. Give some applications of operating system.
   - 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.

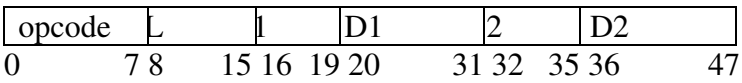
4. Define loader.

   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.

6. Draw SS instruction format.



   It is a 6 byte instruction used to move L+I bytes data fro the storage location1 to the storage location2.

   Storage location1 = D1+[B1]

   Storage location2 = D2+[B2]

   Eg: MOV 60,400(3),500(4)

7. Give any two difference between base relative addressing and program counter relative addressing used in SIC/XE.

Base relative addressing	PC relative addressing
Target address is calculated using the formula address = Displacement + [B] B-base register	The target address is calculated using the formula address = Displacement + [PC] PC-program counter
Displacement lies between 0 to 4095	Displacement lies between –2048 to 2047

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.

10. List out any two CISC and RISC machine.

CISC –Power PC, Cray T3E

RISC – VAX,Pentium Pro architecture

11. Following is a memory configuration:

Address	Value	Register R
1	5	5
5	7	
6	5	

What is the result of the following statement?

ADD 6(immediate) to R (indirect)

Here 6 is the immediate data and the next value is indirect data.ie the register contains the address of the operand. Here the address of the operand is 5 and its corresponding value is 7.

6 + [R] = 6+ [5] = 6+ 7 =13

12. Following is a memory configuration:

Address	Value	Register R
4	9	6
5	7	
6	2	

What is the result of the following statement?

SUB 4(direct) to R (direct)

Here one operand is in the address location 4(direct addressing) and the next operand is in the register(register direct).

The resultant value is 9 –6 =3.

13. What is the name of X and L register in SIC machine and also specify its use.

A-accumulator

Used for arithmetic operation.ie in the case of arithmetic operations one operand is in the accumulator,and other operand may be a immediate value,registre operand or memory content.The operation given in the instruction is performed and the result is stored in the accumulator register.

L-linkage register

It is used to store the return address in the case of jump to subroutine (JSUB) instructions.

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	R2

15. Consider the instructions in SIC/ XE programming

10	1000	LENGTH	RESW	4
20	-----	NEW	WORD	3

What is the value assign to the symbol NEW.

In the line 10 the address is 1000 and the instruction is RESW 4.It reserves 4 word (3 x 4 =12) area for the symbol LENGTH.hence 12 is added to the LOCCTR.

Thus the value of the symbol NEW is 1000+12 =100C.

16. What is the difference between the instructions LDA # 3 and LDA THREE?

In the first instruction immediate addressing is used. Here the value 3 is directly loaded into the accumulator register.

In the second instruction the memory reference is used. Here the address (address assigned for the symbol THREE) is loaded into the accumulator register.

17. Differentiate trailing numeric and leading separate numeric.

The numeric format is used to represent numeric values with one digit per byte. In the numeric format if the sign appears in the last byte it is known as the trailing numeric. If the sign appears in a separate byte preceding the first digit then it is called as leading separate numeric.

18. What are the addressing modes used in VAX architecture?

Register direct, register deferred, auto increment and decrement, program counter relative, base relative, index register mode and indirect addressing are the various addressing modes in VAX architecture.

19. How do you calculate the actual address in the case of register indirect with immediate index mode?

Here the target address is calculated using the formula

T.A =(register) + displacement.

20. Write the sequence of instructions to perform the operation BETA = ALPHA + 1 using SIC instructions.

```
            LDA     ALPHA
            ADD     ONE
            STA     BETA
            ….      ….
ALPHA       RESW    1
BETA        RESW    1
ONE         RESW    1
```

21. Write the sequence of instructions to perform the operation BETA = ALPHA+5 using SIC/XE instructions.

```
LDA         ALPHA
            ADD     #1
            STA     BETA
            ….      ….
ALPHA       RESW    1
BETA        RESW    1
```

22. What is the use of TD instruction in SIC architecture?

The test device (TD) instruction tests whether the addressed device is ready to send or receive a byte of data.The condition code is set to indicate the result of this test. Setting of < means the device is ready to send or receive, and = means the device is not ready.

UNIT II-ASSEMBLERS

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.

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 line10 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.

9.Differentiate absolute expression and relative expression.

If the result of the expression is an absolute value (constant) then it is known as absolute expression.,

Eg : BUFEND – BUFFER

If the result of the expression is relative to the beginning of the program then it is known as relative expression.label on instructions and data areas and references to the location counter values are relative terms.

Eg: BUFEND + BUFFER

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.Aftre 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 .

14.What are the use of base register table in AIX assembler?

A base register table is used to remember which of the general purpose registers are currently available as base registers and also the base addresses they contain.
.USING statement causes entry to the table and .DROP statement removes the corresponding table entry.

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.

16.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 – PASS1

18. What is meant by machine independent assembler features?

The assembler features that does not depends upon the machine architecture are known as machine independent assembler features.
Eg: program blocks,Literals.

19. How the register to register instructions are translated in assembler?

In the case of register to register instructions the operand field contains the register name.During the translation first the object code is converted into its corresponding machine language equivalent with

the help of OPTAB.Then the SYMTAB is searched for the numeric equivalent of register and that value is inserted into the operand field.

Eg: 125 1036 RDREC CLEAR X B410

B4-macine equivalent of the opcode CLEAR
10-numeric equivalent of the register X.

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 with in 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 in to 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(111111111100) 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 the 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.

Pass1: assigns address to all external symbols

Pass2: 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 direct 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 into the memory.

15. Give the difference between linking loader and linkage editors.

g loader	ge editor
location and linking is performed each time ogram is loaded	luces a linked version of a program and is written in a file for later execution
he loading can be accomplished in a single	asses are required

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.

19. What is shared and private data?

The data divided among processing element is called shared data. If the data is not shared among processing elements then it is called private data.

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
```

UNIT IV (MACRO PROCESSORS)

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.

7.What are the statements used for conditional macro expansion?

IF-ELSE-ENDIF statement

WHILE-ENDW statement

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.

9.Consider the macro definition

#Define DISPLAY(EXPR) Printf("EXPR = %d\n",EXPR)

Expand the macro instruction DISPLAY (ANS)

Ans.: Printf ("EXPR = %d\n", ANS)

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?

Pass1: 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.

18.How the nested macro calls are executed?

The execution of nested macro call follows the LIFO rule.In case of nested macro calls the expansion of the latest macro call is completed first.

19.Mention the tasks involved in macro expansion.

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

UNIT V (TEXT EDITORS)

1.Define interactive editor?

An interactive editor is a computer program that allows a user to create and revise a target document. The term document includes objects such as computer programs, text, equations, tables, diagrams, line art, and photographs any thing that one might find on a printed page.

2. What are the tasks performed in the editing process?

4 tasks

1. select the part of the target document to be viewed and manipulated.
2. Determine how to format this view on-line and how to display it.
3. Specify and execute operations that modify the target document.
4. Update the view appropriately.

3. What are the three categories of editor's devices?

1. Text device/ String devices
2. Button device/Choice devices
3. Locator device

4.What is the function performed in editing phase?

In the actual editing phase, the target document is created or altered with a set of operations such as insert, delete, replace, move and copy.

5.Define Locator device?

Locator devices are two-dimensional analog-to-digital converters that position a cursor symbol on the screen by observing the user's movement of the device. The most common such devices for editing applications are the mouse and the data tablet.

6.What is the function performed in voice input device?

Voice-input devices, which translate spoken words to their textual equivalents, may prove to be the text input devices of the future. Voice recognizers are currently available for command input on some systems.

7.What are called tokens?

The lexical analyzer tracks the source program one character at a time by making the source program into sequence of atomic units is called tokens.

8.Name some of typical tokens.

Identifiers, keywords, constants, operators and punctuation symbols such as commas and parentheses are typical tokens.

9. What is meant by lexeme?

The character that forms a token is said to be a lexeme.

10.Mention the main disadvantage of interpreter.

The main disadvantage of interpreter is that the execution time of interpreted program is slower than that of a corresponding compiled object program.

11.What is meant by code optimization?

The code optimization is designed to improve the intermediate code, which helps the object program to run faster and takes less space.

12. What is error handler?

The error handler is used to check if there is an error in the program. If any error, it should warn the programmer by instructions to proceed from phase to phase.

13.Name some of text editors.

- line editors
- stream editors
- screen editors
- word processors
- structure editors

14.What for debug monitors are used?

Debug monitors are used in obtaining information for localization of errors.

15.Mention the features of word processors.

- moving text from one place to another
- merging of text
- searching
- word replacement

16.What are the phases in performing editing process?

a. Traveling phase
b. Filtering phase
c. Formatting phase
d. Editing phase

17.Define traveling phase.

The phase specifies the region of interest. Traveling is achieved using operations such as next screenful, bottom, find pattern.

18.Filtering phase.

The selection of what is to be viewed and manipulated in given by filtering.

19.Editing phase

In this phase, the target document is altered with the set of operations such as insert, delete, replace, move and copy.

20.Define user interface?

User interface is one, which allows the user to communicate with the system in order to perform certain tasks. User interface is generally designed in a computer to make it easier to use.

21.Define input device?

Input device is an electromechanical device, which accepts data from the outside world and translates them into a form, which the computer can interpret.

22.Define output devices

Output devices the user to view the elements being edited and the results of the editing operations.

23.What are the methods in Interaction language of a text editor?

a. Typing –oriented or text command oriented method
b. Function key interfaces
c. menu oriented method

24. Define interactive debugging systems

An interactive debugging system provides programmers with facilities that aid in the testing and debugging of programs.

1. Debugging functions and capabilities
2. Relationship with other parts of the system
3. User interface criteria.

25.Define editor structure.

The command language processor accepts input from the users input devices and analyzes the tokens and syntactic structure of the commands.

26.Give the components of editor structure

4 components

a. Editing component
b. Traveling component
c. Viewing component
d. Display component

27.What are the basic types of computing environments used in editors functions?

Editor's function in three basic types of computing environments

i.Time sharing
ii.Stand-alone
iii.Distributed

UNIT 1

1. Explain about the Cray T3E architecture

Memory
Size-64 MB to 2 GB and 64 bit virtual address space.
Basic unit-byte
2byte-word,4 byte-long word,8 byte –quad word
Registers
-32 general purpose registers(R0-R31) –64 bits
-32 floating point registers(F0-F31)
-some status and control registers.
Data Format
-integers-word,long word,quad word integer format
-negative numbers-2's complement form
-characters-ASCII code
-two different floating point datas
Instruction format
-5 different formats-32 bitds length
-6 bits opcode field
-some instructions having additional field is used(functional field)
Addressing modes
-immediate
-register direct
-PC relative mode(used for branch instructions)
-register indirect with displacement(load &store instructions in subroutine)
Instruction sets
-130 machine instructions
-only load & store instructions refers memory
I/O devices
-I/O operations are performed through multiple ports.
-one or more I/O channels are used.

2. Write in detail about Pentium Pro architecture.

Memory
-physical memory-basic unit byte-2 bytes word-4 bytes double word
-virtual memory-memory is a combination of segments.
-segments may some times divided into pages of equal size
Registers
-8 general purpose registers-32 bits
-EAX,EBX,ECX,EDX-data registers &EBP.ESP.EDI,ESI-address registers
-EIP –acts like a program counter
-FLAGS-gives status information
-6 segment registers(CS,DS,ES,FS,GS&SS)-16 bits
-CS-contains the address of currently executable instruction
-SS-address of stack segment
Data
integers-word,long word,quad word integer format
-negative numbers-2's complement form
-characters-ASCII code
-packed and unpacked decimals are also used
-3 different floating point datas(single precision –24 bits 1S,7E,24F,
double precision-64 bits 1S,10E,53E,extended precision-80 bits (1S,15E,64F)
Instruction format
-3fields-flag field(gives the type of operation)-opcode field(gives the operation to be performed 1 or 2 byte)-operand field
Addressing modes
-immediate mode
-register mode
-direct mode-Target address = base register + C[X]*(scale factor)+displacement.

this formula gives 8 different addressing modes.

-relative mode(Target address = displacement + C[EIP])

Instruction Set

-Depends on the type of operand-3 types(register-register ,register-memory, memory-memory instructions)

-Depends upon operation-arithmetic, data transfer, control, string manipulation and bit manipulation instructions,

Input &Output

-1 byte is transferred at a time

-EAX acts as a accumulator

-input device to EAX,EAXto output device.

3. Explain the architecture of SIC/XE machine.

Draw neatly the Block diagram

Explain each block separately

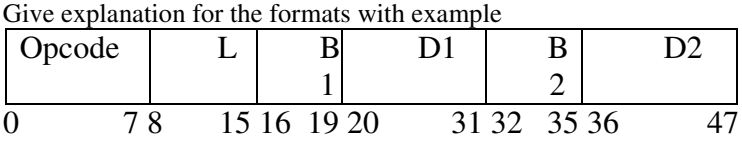
4. Explain the General structure of IBM 370 system with a neat sketch and also explain the different instruction formats used in it. Give one example for each instruction format.

Draw the general structure IBM 370

Mention all the instruction formats

SS instruction format

Give explanation for the formats with example



It is a 6 byte instruction used to move L+I bytes data fro the storage location1 to the storage location2.

Storage location1 = D1+[B1]

Storage location2 = D2+[B2] Eg: MOV 60,400(3),500(4)

5. Mention the differences between SIC and SIC/XE.

SIC	SIC/XE
Here only five registers are used. A,X,L,SW and PC	Here there are nine registers. A,X,L,SW ,PC,B,S,T and F
There is no floating point hardware	Floating point hardware is used
Only one instruction format is used	Four different type of instruction formats
Two different addressing modes are used	Here there are many addressing modes

6. Explain Ultra sparc architecture

SPARC-Scalable Processor Architecture

a.)Memory

-8 bit bytes,byte address

-2 consecutive bytes form half word

- 4 bytes –word,8 bytes –double word

-virtual address space-2^64 byte

b.) Registers:

-use 32 registers (r0-r31)

-r0 to r7-global

-general purpose register 32 bit long

-floating point computation using FPU

c.)Data Formats
-stored as 8,16,32,64 bit binary numbers
-2's complement for negative values
-big endian
d.)Instruction formats:
format 1,format 2,format 3
e.)Addressing modes
immediate,register direct mode
f.) Instruction set:
pipelined
conditional move instruction
g.)Input/Output:
load & store instruction.

UNIT-II

1. Explain in detail about basic assembler functions.
   - A simple SIC assembler
   - Assembler Algorithm
   - Data structures

2. Explain about the machine-Dependent Assembler features.
   - Instruction formats
   - Addressing modes
   - Program Relocation

3. Discuss in detail about the machine-Independent Assembler features.
   - Literals
   - Symbol-Defining Statements
   - Expressions
   - Program blocks
   - Control sections and Program Linking

4. Explain in detail about the assembler Design options.
   - One-pass Assembler
   - Multi-pass Assembler

5. Discuss in detail about MASM Assembler
   - Classes
   - Data Segments
   - Near jump
   - Far jump problem
   - Segments
   - MASM directives

UNIT-III

1. Explain in detail about basic loader functions.
   - Design of an Absolute Loader
   - A simple Bootstrap loader

2. Explain about Machine-Dependent Loader Features.
   - Relocation
   - Program Linking
   - Algorithm
   - Data structures

3. Discuss in detail about Machine-independent Loader features
   - Automatic Library Search
   - Loader Options

4. Explain about the Loader Design Options.
   - Linkage Editor
   - Dynamic linking
   - Bootstrap loaders

5. Explain in detail about MS-DOS Linker
   - MS-DOS Assemblers and Compilers
   - MS-DOS LINK
   - MS-DOS Object modules

UNIT-IV

1. Explian in detail about the basic Macro Processor functions.
   - Macro Definition
   - Macro expansion
   - Algorithm
   - Data Structures

2. Discuss in detail about the Machine-independent macro processor features.
   - Concatenation of Macro parameters
   - Generation of unique labels
   - conditional macro Expansion
   - Keyword macro parameters

3.Explain about Macro Processor Design options
   - Recursive Macro Expansion
   - General purpose Macro Procesors
   - Macroprocessing within Language Translators

4. Explain in detail about MASM Macro Processor
   - Conditional assembly statements
   - MASM macro
   - Conditional statements

5. Explain in detail about ANSI C macro Language
   - Macro definitions with parenthesis
   - Macro expansion with parenthesis
   - Conditional compilation statements
   - Debugging statements

UNIT-V

1. Explain in detail about the following

i) Editing process

ii) User Interface

Editing process:

Tasks
   - Select the part of the target document
   - Determine how to format
   - Specify and execute operations
   - Update

UI-Conceptual model
   - Operations on numbered sequence
   - Manipulate portions of the plane
   - Concerned i/p devices

2. Explain about the editor structure.

Diagram

Explanation

3. Discuss in detail about debugging functions and capabilities.

Different levels

- Procedure
- Branch
- Individual Instructions
- Examples

4. Explain in detail about the following

i)Relationships with other parts of the system

ii)User Interface criteria

i.Requirement-Always be available

- Debugging
- Application development time
- Production environment
- Coordinate with existing and future language compilers and interpreters

ii. simple organization

- full screen displays and windowing systems
- command language should be clear
- On-line help facility

5. Explain about various software tools.

- Text editors
- Debugging systems