UNIVERSITY INSTITUTE of COMPUTING

Asia's Fastest Growing University

System Software
CAT-204

Design By:

Prof. Pawandeep Sharma

A.P

Chandigarh University-Gharuan

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Syllabus

UNIT-I

Introduction to System Software: Machine Structure, evolution of operating system, machine language.

Assembler: Elements of Assembly Language Programming, General design procedure, design of a Two Pass Assemblers, A Single Pass Assemblers Design.

Table Processing: Searching & Sorting.

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Syllabus

UNIT-II

Macro and Macro Processors: Macro instructions, Features of a macro Facility: macro Instruction arguments, Conditional macro expansion, Macro calls within macros, Macro instruction defining macros, Advanced Macro Facilities, Implementation of simple macro processor, Two-pass algorithm, Implementation of macro calls within macros, Implementation within an assembler.

Linkers – Translated linked and load time addresses, relocation and linking concepts, Design of a linker, self relocating programs.

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Syllabus

UNIT-III

Loaders: Loader scheme, absolute loaders, Subroutine linkages, Relocating loaders, Direct linking loaders, binders, linking loaders, overlays, Dynamic Binders, Design of an Absolute Loader, Design of a Direct-Linking Loader. Compilers: Phases of Compiler Construction, Symbol Table, Top-down and bottom-up Parsing, Operator-Precedence Parsing, LR Parsers, Code Generation and Code Optimization, Memory management, Design & other issues.

The machine language

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Meaning

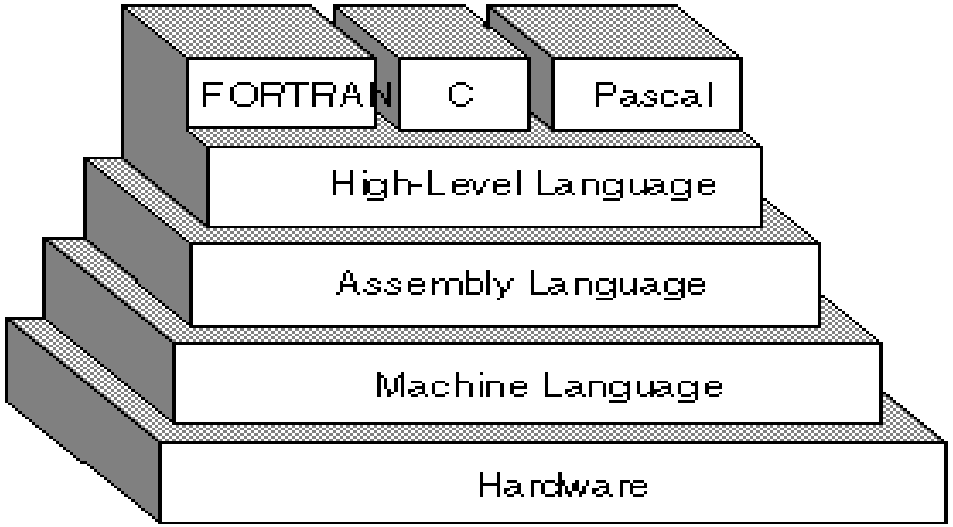
- Machine language is the basic low-level programming language designed to be recognized by a computer.
- Actually the language is written in a binary code of 0s and 1s that represent electric impulses or off and on electrical states respectively.
- A group of such digits is called an instruction and it is translated into a command that the central processing unit or CPU understands.

Why Humans Don't Use Machine Language?

- While easily understood by computers, machine languages are almost impossible for humans to use because they consist entirely of numbers.
- Programmers, therefore, use either a high-level programming language or an assembly language.
- An assembly language contains the same instructions as a machine language, but the instructions and variables have names instead of being just numbers.

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

The hierarchy of languages



UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Machine Language Instructions

- The instructions in the machine language are organized in patterns of 0s and 1s in various lengths such as 16, 24, 32, and 64 digits or bits, representing specific tasks such as storing or transferring data.
- An instruction is made up of two parts: the operator(opcode) and the operand.
- The first few bits of an instruction are the "operator or opcode," whose role is to specify the kind of operation that needs to be performed.
- The rest of the bits are the "operand," whose role is to indicate the location where the operation is to be performed.
- For instance, a binary opcode such as the 000001 could be an instruction to store the contents of the accumulator in a given memory address. The whole instruction could look like this: 00000100011100000000000100000010.

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

example

Example: L 2,924(0, 1)

OP R1, D2(X2, B2)

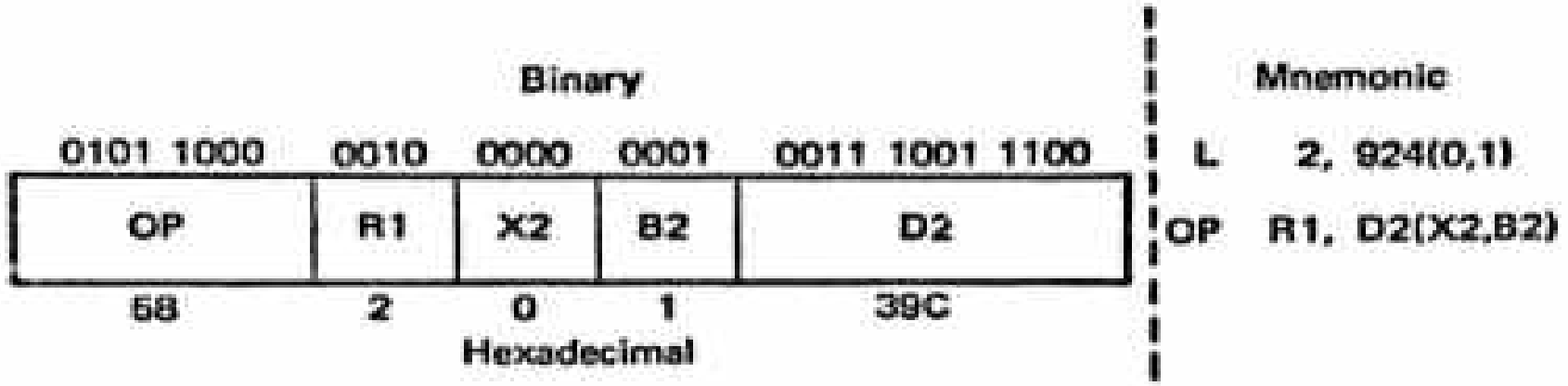


FIGURE 2.6 Mnemonic form of machine language

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Machine Language vs High Level Language

- A machine language is designed in terms of execution by the hardware of the computer. A high-level language is an easy, reliable, and efficient way to express the creativity of a programmer into commands that a computer will understand.
- The task of programming using binary code would be complex as compared to HLL.
- A slight change in machine language, e.g. the change of a bit, may affect the whole instruction sequence.
- Instructions written in a machine language can be very lengthy. This can easily result in errors during programming. What's more, the whole process could become very time-consuming and costly.

References

BOOKS:-

- System Programming, Dhamdhare, Chapter 3.
- https://www.youtube.com/watch?v=VG9VopzV_T0
- http://whatis.techtarget.com/definition/system-software
- http://searchdatacenter.techtarget.com/definition/assembler
- http://www.icse.s5.com/notes/m2.html

Queries???

UNIVERSITY INSTITUTE *of*
COMPUTING
Asia's Fastest Growing University

Thank You

UNIVERSITY INSTITUTE of COMPUTING

Asia's Fastest Growing University

System Software
CAT-204

Design By:

Prof. Pawandeep Sharma

A.P

Chandigarh University-Gharuan

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Syllabus

UNIT-I

Introduction to System Software: Machine Structure, evolution of operating system, machine language.

Assembler: Elements of Assembly Language Programming, General design procedure, design of a Two Pass Assemblers, A Single Pass Assemblers Design.

Table Processing: Searching & Sorting.

UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

Syllabus

UNIT-II

Macro and Macro Processors: Macro instructions, Features of a macro Facility: macro Instruction arguments, Conditional macro expansion, Macro calls within macros, Macro instruction defining macros, Advanced Macro Facilities, Implementation of simple macro processor, Two-pass algorithm, Implementation of macro calls within macros, Implementation within an assembler.

Linkers – Translated linked and load time addresses, relocation and linking concepts, Design of a linker, self relocating programs.

Syllabus

UNIT-III

Loaders: Loader scheme, absolute loaders, Subroutine linkages, Relocating loaders, Direct linking loaders, binders, linking loaders, overlays, Dynamic Binders, Design of an Absolute Loader, Design of a Direct-Linking Loader. Compilers: Phases of Compiler Construction, Symbol Table, Top-down and bottom-up Parsing, Operator-Precedence Parsing, LR Parsers, Code Generation and Code Optimization, Memory management, Design & other issues.

Elements of Assembly Language

Statement Format

[Label] <Opcode> <operand specification>[,<operand specification>...]

- [..] indicates optional
- Label associates with memory word generated

<operand Specification> syntax is:-

<symbolic name> [± <displacement>][(<index register>)]

The Mnemonic Codes

Instruction opcode	*Assembly mnemonic*	*Remarks*
00	STOP	Stop execution
01	ADD	Perform addition
02	SUB	Perform subtraction
03	MULT	Perform multiplication
04	MOVER	Move from memory to register
05	MOVEM	Move from register to memory
06	COMP	Compare and set condition code
07	BC	Branch on condition
08	DIV	Perform division
09	READ	Read into register
10	PRINT	Print contents of register

Assembly Language Statements

- Imperative Statements
- Declaration Statements
- Assembler Directives

1. Imperative Statements

- Indicates the action to be performed during execution
- Translates into machine instruction
- Eg. Arithmetic operations etc.

2. Declaration Statements

- Syntax:-
  - [Label] DS <constant>
  - [Label] DC ‘<value>’
- Example:-
  - A DS 1
  - G DS 200
  - ONE DC ‘1’

3. Assembler Directive

- Instructs assembler to perform an action
  - START <constant>
  - END [<operand specification>]

Constants

Assembly language uses two types of constants:

- Immediate Operand
- Literals

Constants

Immediate Operands :

The format for using immediate operand is :

ADD AREG,5

Here AREG is register & 5 is immediate operand.

Literals:

The format for using Literal is : =<‘VALUE’>

i.e. ADD A,=‘5’

Here A is operand& 5 is literal.

References

BOOKS:-

•System Programming, Donovan, Chapter 3.

•https://www.youtube.com/watch?v=VG9VopzV_T0

• http://whatis.techtarget.com/definition/system-software

• http://searchdatacenter.techtarget.com/definition/assembler

•http://www.icse.s5.com/notes/m2.html

Queries???

UNIVERSITY INSTITUTE *of*
COMPUTING
Asia's Fastest Growing University

Thank You