Computer Organization & Architecture (Assembly Language)

Dr. Sonu Lamba

Department of Computer Science and Engineering
The LNM Institute of Information Technology Jaipur

August 31, 2020

Introduction to Assembly Language

- Machine instructions are represented by patterns of 0s and 1s.
    - Such patterns are awkward to deal with when preparing programs
    - Thus, use symbolic names to represent the patterns.

Assembler

- What is an Assembler?

Assembler Directives

- Commands or instruction to the assembler while it translates a source program into an object program.

Macros

- Short sequences of text lines which are used frequently

The Assembly Process

- To prepare a source program: Text Editor
- Assembler translates source programs into object programs that comprise machine instructions.
    - Generates the binary encoding for the OP code and other instruction fields.
    - Recognizes directives that specify numbers and characters and directives that allocate memory space for data areas.
    - Address labels are assigned values based on their position relative to the beginning of an assembled program
    - Keeps track of all names and their corresponding values in a symbol table

Label	Opcode Operands	Length	ILC
ONE:	MOV R1, X	5	100
	MOV R2, Y	5	105
TWO:	MOV R3, Z	5	110
	CMP R1,R0	2	112
	BZ NEXT	4	116
	SET MACRO		
	ADD R1, R2	2	118
	ADD R1, R3	2	120
	ENDM		
	MUL R1, R2	4	124
	SET		
NEXT:	JMP ONE	5	

Two-pass Assembler

- Forward Referencing Problem: a problem arises when a name appears as an operand before its value is defined
- Commonly-used Solution: to have the assembler scan through the source program twice
  1. First Pass
     - Creates the symbol table
     - Store macros and expand them
     - For address labels, determines the value of each name from its position relative to the start of the source program
  2. Second Pass
     - Looks up each name it encounters in the symbol table
     - Substitutes the corresponding numerical value