

# Assembler



# Introduction



- It translate assembly language program into machine code.
- It converts symbolic operands to m/c address.
- Build machine instructions
- Perform error checking
- Represent variables in symbolic names

# Fields of assembler



- It has 4 fields
  - Label
    - ✦ It is an identifier and optional field
  - Opcode
    - ✦ It contains mnemonic called operation code
  - Operand
    - ✦ It contains additional information that opcode requires like constant, address, immediate data register
  - Comments

Example: START: LDA #24H

# Elements of Assembly language programming



- Mnemonic operation codes- they are easier to remember and use than numeric operation code.
- Symbolic operands- A programmer can associate symbolic names with data or instruction and use them in assembly statements.
- Data declaration- Data can be declared in a variety of notations.

In assembly language each statement has 2 operands, 1<sup>st</sup> operand is always CPU register.



- 2<sup>nd</sup> operand refers to memory word by using a symbolic name and optional displacement.
- Example AREA refers to memory word with which the name AREA is associated
- AREA+5 refers to memory word that is 5 words away from the word with the name AREA (5 displacement)
- AREA(4) indexing the operand AREA with index register 4. So operand address is obtained by adding context of index register and address of AREA.

# Assembly language statements



- It consists of 3 statements:
  - Imperative statements: It indicates an action to be performed during execution of program. Each imperative statement is translate into one machine instruction.
  - Declaration statement: It focus on the what the problem and solution mechanism for language implementation.
  - Assembler Directives: It instructs the assembler to perform certain actions during assembling of the program. They can be used to declare variables, create storage space, to declare constant.

# Advantages



- Reduced errors
- Faster translation times
- Changes made easier and faster.

# Disadvantages



- Many instructions are required to achieve small task.
- Source program large and difficult to follow
- Require knowledge of processor architecture
- Programs are machine dependent



# Simple assembly scheme



- Following steps are used for design specification of any assembler program.
  - Find out information necessary to perform task.
  - Design suitable data structure
  - To obtain and maintain information determine
  - To perform the task by using recorded information determine the required processing.