SPOS

Mitali Srivastav
Div: 03
RNo: 875

1.

Title: Pass I of two pass assembler

Objectives:
- To study assembler
- To understand algorithm of pass 1.
- To implement assembler pass I using programming language.

Outcomes: Understand pass I of two assembler
- Implement pass I using JAVA

Software Requirements: 64 bit Machine
4 GB or 8 GB RAM
500 GB or 1 TB HDD

Theory: An assembler is a program that takes basic computer instruction that corresponds to basic machine operation that the computer can perform. For eg, "Load" instruction causes the processor to move a string of bits from a location in the processor memory to a special holding place called a register. Assuming the processor has atleast eight register each numbered the following instruction would have value at memory location 8000 into register.

- The programmer can write a program with a sequence of assembler instruction.
- This sequence is known as source code.
- Assembler takes each program statement in the

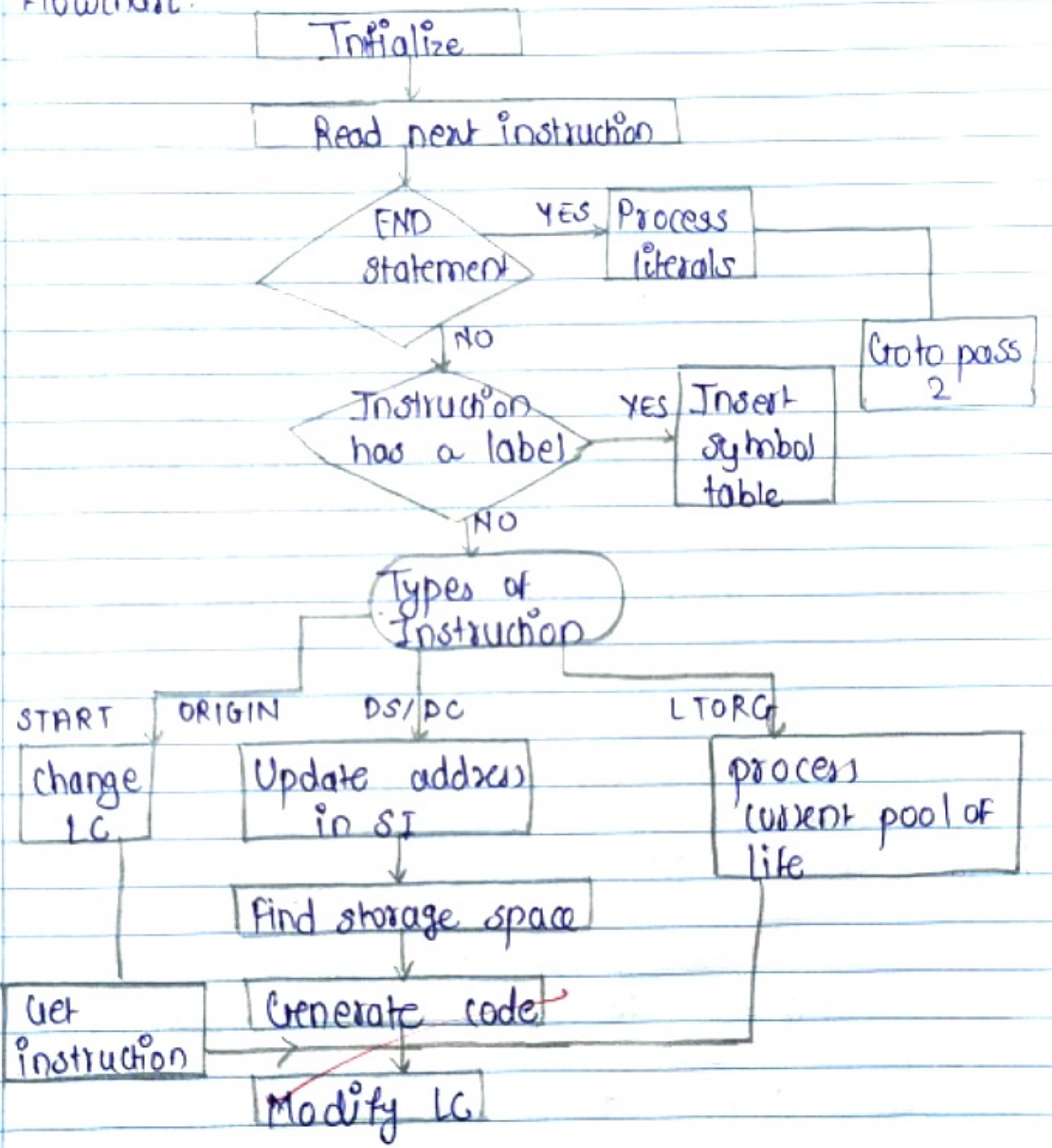
source program and generate a bit stream or pattern.

Algorithm for pass I:

1) Initialize LC=0 (default), pooltable ptr=1, pooltab[1]=1, tab-ptr=1
   Read source file
2) While next statement is not END
   a. If string = label then enter it is symbol table
   b. If string = START then LC = operand value of START
   c. If EQU then address = value of <add spec> corred symbol table entry for label to address
   d. If DS then
      - code = code of DS
      - size = size of memory area required by DC/DS.
   e. If an IS then
      - code = machine code from optab
      - LC = LC + length of instruct^{n}

   i: this literal = literal in operand
   ii: LITTAB [littab-p+1] = this literal
   iii: littab ptr = littab-p+r+1
   else
      this entry = symbol table entry.
3) Processing of END statement
   a. Performance of step 2f
   b. Generate IC
   c. Write all table & IC on to file.

Flowchart:



Conclusion: Thus we have performed pass I of two pass assembler.

27/09/22

Mitali Shrivastav
Div: 03
RNo: 875

Title: Pass II of two pass assembler

Objectives: To study pass II of two pass assembler
To implement pass II using JAVA

Problem Statement:
Implement pass II of two pass assem-bler for pseudo machine in JAVA using objed oriented feature

Outcomes: Study pass II of two pass assembler understand algorithm of pass II.

Software Requirement: Operating System
JDK
Eclipse

Hardware Requirement: 64 bit machine
4GB or 8GB RAM
500 GB or TTB HDD.

Theory: Algorithm for pass II
1. Code_area_address = address of code area, pooltable-ptr=1 LC=0
2. While next statement is not END
a. clean machine buffer
b. If an LTORG statement then
· Process literal LITTAB [POOLTAB [Pooltab_ptr]... LITTAB [littab_ptr+1] -1 similar to processing of constant in DC statement
· size = size of memory area required for literal

- pooltable_ptr = pooltab_ptr + 1

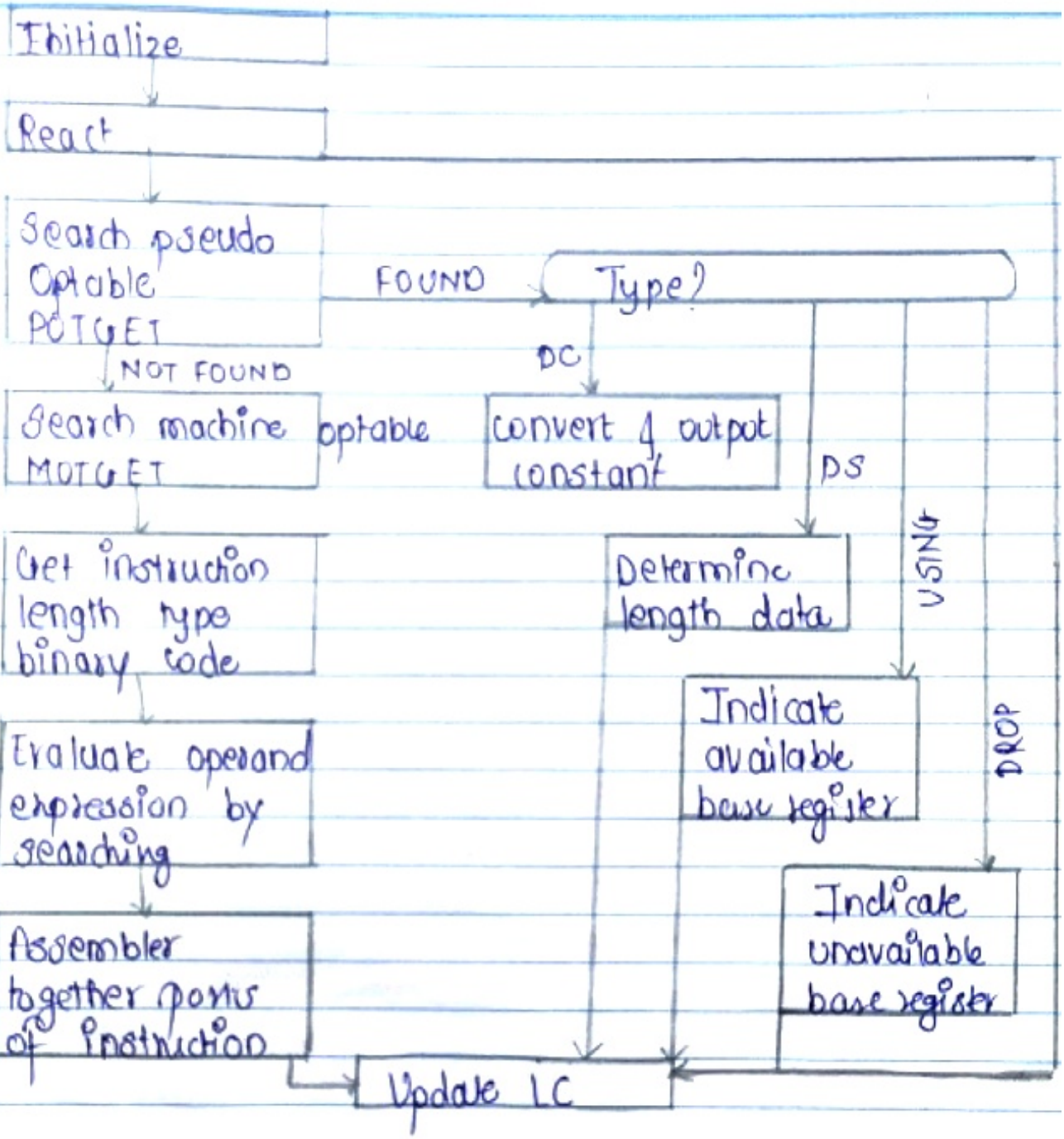
c. IF START statement then LC = value specified in operand field size = 0.

d. IF DS then assemble the constant in machine code buffer.

e. IF IS then
- Get operand address from SYMTAB
- ASSEMBLE instruction to machine code
- size = size of instruction

f. IF size ≠ 0 then
- move content of machine code buffer to address of code area at LC.
- LC = LC + size

3. Processing of END statement
- perform step 2b & 2f
- perform error handling
- Write code to output file.



Conclusion: Hence we have studied pass II assembler

Done
07/09/22